Deploying OpenStack Cluster on CentOS 8 with packstack

Environment Description

5 CentOS 8.3 with external network access is needed. The following table describes the hardware requirements.

Role	hostname	IP address	CPU	RAM	Disk
Controller	controller	192.168.8.10	2vCPUs	8GB	100GB
Compute	compute01	192.168.8.20	4vCPUs	16GB	200GB
Compute	compute02	192.168.8.21	4vCPUs	16GB	200GB
Network	network	192.168.8.30	2vCPUs	8GB	100GB
NTP	ntp	192.168.8.40	1vCPU	2GB	20GB

Network configuration is described below

- Gateway IP Address: 192.168.8.1
- Subnet Address: 192.168.8.0/24
- Available IP Address: 192.168.8.50 - 150
- DNS Server: 192.168.8.1

Set Static IP Address

On all hosts

```
[root@192 ~]# nmcli connection modify ens33 ipv4.method manual ipv4.dns 192.168.8.1 ipv4.gateway 192.168.8.1 autoconnect yes ipv4.addresses <ip_addr>/24
[root@192 ~]# nmcli connection up ens33
```

Configure Hostname

Write `/etc/hosts`

```
# Static table lookup for hostnames.
# See hosts(5) for details.
#
127.0.0.1    localhost
::1          localhost

192.168.0.10 controller
192.168.0.20 compute01
192.168.0.21 compute02
192.168.0.30 network
192.168.0.40 ntp
```

Distribute `/etc/hosts` to all hosts

```
for server in {controller,compute01,compute02,network,ntp};do scp /etc/hosts root@$server:/etc/hosts;done
```

Set hostname

```
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "hostnamectl set-hostname $server";done
```

Disable Firewall, SELinux, NetworkManager

On all hosts

```
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "systemctl disable --now firewalld";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "systemctl mask firewalld";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "sed -i 's/SELINUX=enforcing/SELinux=disabled/' /etc/sysconfig/selinux";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "sed -i 's/SELINUX=enforcing/SELinux=disabled/' /etc/selinux/config";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "setenforce 0";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "dnf install -y network-scripts";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "systemctl disable --now NetworkManager";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "systemctl enable --now network";done
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "systemctl reboot";done
```

Enable NTP

On ntp, edit `/etc/chrony.conf`

```
# Allow NTP client access from local network.
allow 192.168.8.0/24
```

On other servers, edit `/etc/chrony.conf`

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
# pool 2.centos.pool.ntp.org iburst
pool ntp iburst
```

On all hosts

```
systemctl enable --now chronyd.service
systemctl restart chronyd.service
```

Verify

```
for server in {controller,compute01,compute02,network,ntp};do ssh root@$server "chronyc sources";done
```

Install OpenStack Repository

On all OpenStack attenders

```
for server in {controller,compute01,compute02,network};do ssh root@$server "dnf config-manager --enable powertools";done
for server in {controller,compute01,compute02,network};do ssh root@$server "dnf remove -y epel-release";done
for server in {controller,compute01,compute02,network};do ssh root@$server "sudo dnf install -y https://www.rdoproject.org/repos/rdo-release.el8.rpm";done
# Not needed but rdo project recommend to install
# for server in {controller,compute01,compute02,network};do ssh root@$server "sudo dnf install -y centos-release-openstack-victoria";done
for server in {controller,compute01,compute02,network};do ssh root@$server "dnf -y update";done
```

Install Packstack

On controller

```
dnf install -y openstack-packstack
```

Generate answer file

On controller

```
packstack \
--os-neutron-ml2-tenant-network-types=vxlan \
--os-neutron-l2-agent=openvswitch \
--os-neutron-ml2-type-drivers=vxlan,flat \
--os-neutron-ml2-mechanism-drivers=openvswitch \
--keystone-admin-passwd=redhat \
--nova-libvirt-virt-type=kvm \
--provision-demo=n \
--cinder-volumes-create=y \
--os-heat-install=y \
--os-swift-storage-size=20G \
--gen-answer-file /root/answers.txt
```

in this case, admin password is set to `redhat`, Edit `/root/answers.txt`

```
# Specify 'y' if you want to run OpenStack services in debug mode;
# otherwise, specify 'n'. ['y', 'n']
CONFIG_DEBUG_MODE=y

# Server on which to install OpenStack services specific to the
# controller role (for example, API servers or dashboard).
CONFIG_CONTROLLER_HOST=192.168.0.10

# List the servers on which to install the Compute service.
CONFIG_COMPUTE_HOSTS=192.168.0.20,192.168.0.21
```

```
# List of servers on which to install the network service such as
# Compute networking (nova network) or OpenStack Networking (neutron).
CONFIG_NETWORK_HOSTS=192.168.0.30
```

Deploy with answer file

on controller

```
packstack --answer-file /root/answers.txt --timeout=3000
```

Permit Non-admin User to View Instance Status

on controller, edit `/etc/nova/policy.json`, add

```
"os_compute_api:os-extended-server-attributes": ""
```

See also https://docs.openstack.org/nova/latest/configuration/policy.html

Add bash-completion

```
openstack complete | sudo tee /etc/bash_completion.d/osc.bash_completion >
/dev/null
```

Bridge Network to External

on network node, edit `/etc/sysconfig/network-scripts/ifcfg-ens33`

```
TYPE=OVSPort
NAME=ens33
DEVICE=ens33
ONBOOT=yes
TYPE=OVSPort
DEVICETYPE=ovs
OVS_BRIDGE=br-ex
```

create `/etc/sysconfig/network-scripts/ifcfg-br-ex`

```
DEVICE=br-ex
BOOTPROTO=none
ONBOOT=yes
TYPE=OVSBridge
DEVICETYPE=ovs
USERCTL=yes
PEERDNS=yes
IPV6INIT=no
IPADDR=192.168.8.10
PREFIX=24
GATEWAY=192.168.8.1
DNS1=192.168.8.1
```

restart network

```
systemctl restart network
```

Modify noVNC Address

on all compute nodes, edit `/etc/nova/nova.conf`

```
server_proxyclient_address=<compute_node_ip>
```

restart service

```
systemctl restart openstack-nova-compute.service
```

Test the deployment

on controller

```
source ~/keystonerc_admin
```

see all services

```
openstack service list
```

create private network

```
openstack network create private
```

create subnet for private network

```
openstack subnet create \
--network private \
--allocation-pool start=172.10.10.50,end=172.10.10.200 \
--dns-nameserver 223.5.5.5 \
--dns-nameserver 223.6.6.6 \
--subnet-range 172.10.10.0/24 private_subnet
```

create public network

```
openstack network create \
--provider-network-type flat \
--provider-physical-network extnet \
--external public
```

create subnet for public network

```
openstack subnet create \
--network public \
--allocation-pool start=192.168.8.50,end=192.168.8.150 \
--no-dhcp \
--subnet-range 192.168.8.0/24 public_subnet
```

create a router

```
openstack router create private_router
```

add router gateway

```
openstack router set --external-gateway public private_router
```

link router to subnet

```
openstack router add subnet private_router private_subnet
```

on network node, to show router information, run

```
ip netns show
```

to check connectivity, run

```
ip netns exec qrouter-<your_router_uuid_above> ping -c 1 baidu.com
```

on controller, fire up an instance

```
mkdir images
cd images
wget http://download.cirros-cloud.net/0.5.1/cirros-0.5.1-x86_64-disk.img
```

upload image

```
openstack image create \
--disk-format qcow2 \
--container-format bare --public \
--file ./cirros-0.5.1-x86_64-disk.img "Cirros-0.5.1"
```

list images

```
openstack image list
```

create a security group

```
openstack security group create permit_all --description "Allow all ports"
openstack security group rule create --protocol TCP --dst-port 1:65535 --remote-ip 0.0.0.0/0 permit_all
openstack security group rule create --protocol ICMP --remote-ip 0.0.0.0/0 permit_all
```

list security group

```
openstack security group list
```

upload a ssh key

```
openstack keypair create --public-key ~/.ssh/id_rsa.pub admin
```

list ssh keys

```
openstack keypair list
```

list flavors

```
openstack flavor list
```

launch instance

```
openstack server create \
--flavor m1.tiny \
--image "Cirros-0.5.1" \
--network private \
--key-name admin \
--security-group permit_all \
mycirros
```

list all servers

```
openstack server list
```

allocate a floating IP

```
openstack floating ip create public
```

list all floating IPs

```
openstack floating ip list
```

assign floating IP to instance

```
openstack server add floating ip mycirros 192.168.8.60
```

shell into instance

```
ssh cirros@192.168.8.60
```