**# Nodes Setup for CEPH Lab**

A diagram of a computer server

Description automatically generated

1. Create 5 nodes with 1 GB RAM and 1 core CPU each. Let's denote them as follows:

\* ceph-controller

\* ceph-compute01

\* ceph-compute02

\* ceph-monitor

\* ceph-client

2. Attach 3 disks of 20 GB each to `ceph-compute01` and `ceph-compute02`.

**# Hostnames Configuration (Execute these commands on the respective nodes)**

3. On ceph-controller:

hostnamectl set-hostname ceph-controller

vim /etc/hosts

127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4

::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.15.138 ceph-monitor

192.168.15.139 ceph-controller

192.168.15.140 ceph-compute01

192.168.15.141 ceph-compute02

192.168.15.142 ceph-client

rsync /etc/hosts root@192.168.xx.xx:/etc/hosts

4. On ceph-compute01:

hostnamectl set-hostname ceph-compute01

5. On ceph-compute02:

hostnamectl set-hostname ceph-compute02

6. On ceph-monitor:

hostnamectl set-hostname ceph-monitor

7. On ceph-client:

hostnamectl set-hostname ceph-client

## Create FQDN Entry

8. Create FQDN Entry of hostname on all nodes with their respective IP:

Or we can use rsync commands

rsync /etc/hosts root@192.168.xx.xx :/etc/hosts

echo '192.168.xxx.xxx ceph-controller.hpcsa.cdac.in ceph-controller' >> /etc/hosts

echo '192.168.xxx.xxx ceph-compute01.hpcsa.cdac.in ceph-compute01' >> /etc/hosts

echo '192.168.xxx.xxx ceph-compute02.hpcsa.cdac.in ceph-compute02' >> /etc/hosts

echo '192.168.xxx.xxx ceph-monitor.hpcsa.cdac.in ceph-monitor' >> /etc/hosts

echo '192.168.xxx.xxx ceph-client.hpcsa.cdac.in ceph-client' >> /etc/hosts

Replace `192.168.xxx.xxx` with the respective IP addresses.

## Basic Node Configuration (Execute these commands on all nodes)

9. Disable Firewall on all nodes

systemctl stop firewalld && systemctl disable firewalld

--------------------------------------------------------------------------------------------------

**run all nodes**

10. Install chrony for time synchronization:

yum install chrony -y

chronyc sourcestats

11. Create a user for Ceph deployment and disable its password:

useradd cephadm && echo "cdac" | passwd --stdin cephadm

12. Allow cephadm user to run sudo commands without a password:

echo "cephadm ALL = (root) NOPASSWD:ALL" | sudo tee /etc/sudoers.d/cephadm

chmod 0440 /etc/sudoers.d/cephadm

13. Disable SELinux:

sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux

14. Install the EPEL repository:

sudo yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

----------------------------------------------------------------------------------------------------

15. Reboot each node:

reboot

**# Chrony Configuration (Execute these commands on all nodes except ceph-controller)**

16. Modify the chrony configuration:

vim /etc/chrony.conf

Comment all available servers and add the line `server ceph-controller`.

17. Restart chrony:

systemctl restart chronyd

**## Ceph Controller Configuration -------->> (18 to 35 run cmds in controller only)**

18. Install the Ceph release package and Ceph deployment tools:

sudo rpm -Uvh https://download.ceph.com/rpm-mimic/el7/noarch/ceph-release-1-1.el7.noarch.rpm

yum update -y && sudo yum install ceph-deploy python2-pip -y

19. Generate SSH keys for the cephadm user:

su - cephadm

ssh-keygen

Hit enter to accept the defaults.

20. Copy the SSH keys to the other nodes:

ssh-copy-id cephadm@ceph-compute01

ssh-copy-id cephadm@ceph-compute02

ssh-copy-id cephadm@ceph-monitor

ssh-copy-id cephadm@ceph-client

21. Create an SSH configuration file to specify the username to use when SSHing to the other nodes:

nano ~/.ssh/config

Add the following lines:

Host ceph-compute01

Hostname ceph-compute01

User cephadm

Host ceph-compute02

Hostname ceph-compute02

User cephadm

Host ceph-monitor

Hostname ceph-monitor

User cephadm

Host ceph-client

Hostname ceph-client

User cephadm

22. Adjust the permissions on the SSH configuration file:

chmod 644 ~/.ssh/config

23. Test the SSH configuration:

ssh cephadm@ceph-monitor

ssh cephadm@ceph-compute01

ssh cephadm@ceph-compute02

## Ceph Cluster Configuration (Execute these commands as the cephadm user on the Ceph controller)

24. Create a new directory for the Ceph cluster and navigate into it:

mkdir ceph\_cluster

cd ceph\_cluster

25. Create a new cluster:

ceph-deploy new ceph-monitor

26. Edit the Ceph configuration file:

nano ceph.conf

Add the line `public network = 192.168.xxx.xxx/24`, replace `192.168.xxx.xxx` with your local network.

Please note, you will need to replace placeholders like `192.168.xxx.xxx` with your actual values.

27. Install Ceph on the nodes:

ceph-deploy install ceph-controller ceph-compute01 ceph-compute02 ceph-monitor

**# Initialize the Monitor and Create the Ceph Storage Cluster**

28. Initialize the monitor:

ceph-deploy mon create-initial

29. Deploy the Ceph client admin keyring:

ceph-deploy admin ceph-controller ceph-compute01 ceph-compute02 ceph-monitor

30. Create Ceph manager daemons:

ceph-deploy mgr create ceph-compute01 ceph-compute02

31. List the disks available:

ceph-deploy disk list ceph-compute01 ceph-compute02

32. Prepare and activate the OSDs (Replace `/dev/sdb`, `/dev/sdc`, and `/dev/sdd` with the appropriate disk names):

ceph-deploy osd create --data /dev/sdb ceph-compute01

ceph-deploy osd create --data /dev/sdc ceph-compute01

ceph-deploy osd create --data /dev/sdd ceph-compute01

ceph-deploy osd create --data /dev/sdb ceph-compute02

ceph-deploy osd create --data /dev/sdc ceph-compute02

ceph-deploy osd create --data /dev/sdd ceph-compute02

33. Install Ceph on the client and distribute the admin keyring:

ceph-deploy install ceph-client

ceph-deploy admin ceph-client

## Verify the Ceph Storage Cluster

34. Check the health of the Ceph cluster and its detailed status:

sudo ceph health

sudo ceph health detail

sudo ceph -s

35. Create a Ceph storage pool:

sudo ceph osd pool create rbd 200 3 --->(replica)

## Ceph Client Configuration -------------->> run cmds in clients only

36. On the client, create a new block device, disable features that are not compatible with the kernel RBD driver, and map the block device:

rbd create disk01 --size 4096

rbd ls -l

modprobe rbd

rbd feature disable disk01 exclusive-lock object-map fast-diff deep-flatten

rbd map disk01

rbd showmapped

37. Create a filesystem on the new block device, create a mount point, and mount the block device:

mkfs.xfs /dev/rbd0

mkdir -p /mnt/mydisk

mount /dev/rbd0 /mnt/mydisk

df -h

dd if=/dev/rbd0 of=file1.txt bs=1024 count=220040