

Precursor Steps & Storage Node

In a basic HPC cluster, the head node is the orchestration unit and possibly the login portal for your end users. It's one of the most essential pieces to get working appropriately. On our head node setup we will be keep all of our configuration management, identity management, scheduler running, and also export our home file system and an applications file system to our compute nodes. In this section, we will accomplish some basic configuration steps on a head node and configure shared storage for the cluster.

1. Get a newly provisioned vanilla RHEL7 AWS VPC. The LCI instructors should give you credentials to log in using the same credentials used in the "Intro to EC2" session.
 - a. This image has the following extras
 - i. vim / emacs / nano editors
 - ii. gcc and gfortran
 - iii. clustershell installed (unconfigured)

2. Log in using SSH to the EC2 instance:

```
$ ssh -i <key>.pem ec2-user@<ip>
```

Or connect with PuTTY from earlier document

3. Copy PEM file to master node (execute this from your laptop).

```
$ scp -i <key>.pem <key>.pem ec2-user@<ip>:~/ssh/id.pem
```

PuTTY USERS: Download/Install/Run WinSCP

- a. Session -> New Session
- b. hostname -> <IP>
- c. username -> ec2-user
- d. click Advanced -> SSH -> Authentication
- e. in *Private Key File* Browse to your downloaded key PPK.
- f. Save your session just in case
- g. Connect to AWS account (Click Yes)
- h. Copy your .pem file over (not the PPK)

- i. Bottom right click the **6 hidden** to reveal hidden directories
- j. double click `.ssh`
- k. drag `studentXX.pem` (your number in place of XX) over to window
- l. Highlight it, press **F2** rename it to `id.pem`
- m. press **f9** Then remove read/write/execute access from group and other
- n. Feel free to close winscp and return to putty.

4. Clone the LCI Git repository to the master node

```
$ sudo yum install -y git
```

```
$ git clone https://github.com/WyoARCC/build-a-cluster.git
```

5. Set up a SSH key pairs for the `ec2-user` account and root. Utilize the `~/build-a-cluster/scripts/ssh_setup.sh` script. Comments for what is going on are in the script. Use an pager / editor to view the script.

```
$ cd build-a-cluster/scripts
```

```
$ bash ssh_setup.sh # type yes if needed
```

SSH to all of the nodes to load the fingerprints (SSH keyscan also does this)

```
$ ssh node02
```

```
# make sure you don't get permission denied
```

a. **If there are key permission issues:**

b.

```
$ rm /home/ec2-user/.ssh_finished_configured
```

c.

```
$ chmod 400 /home/ec2-user/.ssh/id.pem
```

d.

```
$ bash /home/ec2-user/build-a-cluster/scripts/ssh_setup.sh
```

type yes if needed

e. Retry the parent step

```
$ exit
```

```
$ ssh node03
```

```
# make sure you don't get permission denied
```

```
$ exit
```

```
...
```

```
$ ssh node05
```

```
# make sure you don't get permission denied
```

```
$ exit
```

You should try as the root user as well ;)

```
$ rm -f ~/.ssh/id.pem
```

6. Verify that SELinux is Disabled (This has already been done, but you should verify):

```
$ sestatus
```

```
$ sudo setenforce 0
```

Edit `/etc/selinux/config` and edit the value to be the following:

```
SELINUX=permissive
```

7. Configure firewall appropriately. We are using a VPC and won't worry about firewall for now. The firewall has not been installed. Often the firewalls are disabled for the internal network, but often enabled for login nodes and service nodes where access is allowed from outside the cluster.

```
$ sudo systemctl stop firewalld.service
```

```
$ sudo systemctl disable firewalld.service
```

8. Fix hostnames appropriately (`~/build-a-cluster/hostname_setup.sh`)

```
$ sudo hostname master
```

```
$ echo master | sudo tee /etc/hostname
```

```
$ sudo -i
```

```
# for i in {2..5}; do ssh "node0$i hostname node0$i && echo node0$i > /etc/hostname"; done
```

```
# exit
```

```
$ exec bash
```

The *PS1* prompt should change to indicate the new hostname.

9. Install and configure ClusterShell (installed, but needs configured). See configuration files in `~/build-a-cluster/etc/clusterShell`

```
$ sudo vim /etc/clusterShell/groups.d/local.cfg
```

(alternatively `/etc/clusterShell/groups.d/cluster.yaml`)

```
compute: node[02-05]
all: node[01-05]
```

```
$ clush -a hostname
```

```
$ clush -w @compute hostname
```

```
node02: node02
```

```
node04: node04
```

```
node03: node03
```

```
node01: master
```

```
node05: node05
```

```
$ clush -w master hostname
```

```
a. master: Host key verification failed.
```

```
b. clush: master: exited with exit code 255
```

```
c. $ ssh master
```

```
d. $ exit
```

```
e. $ sudo ssh master
```

```
f. # exit
```

```
$ clush -w master hostname
```

```
$ cluset -c @compute
```

Clush group descriptions and configuration file overview.

10. Make sure NFS services is installed:

```
$ sudo yum install nfs-utils
$ sudo systemctl enable nfs.service
$ sudo systemctl start nfs.service
```

11. Configure an applications directory:

```
$ sudo mkdir -p /exports/apps
```

Append the /etc/exports file to include an export of the directory

```
/exports/apps 10.0.0.0/28(ro,no_root_squash,sync)
```

Re-export the filesystems:

```
$ sudo exportfs -r
```

12. Bind mount the applications directory to the local system

a. Create viable point

```
$ sudo mkdir /apps
```

b. Append entry to the file system table (i.e., /etc/fstab)

```
$ sudo sh -c \  
'echo /exports/apps /apps none defaults,bind 0 0 >> /etc/fstab'
```

```
$ sudo cat /etc/fstab
```

c. Mount directory

```
$ sudo mount -a
```

d. Verify apps is mounted

```
$ mount | grep apps
```

13. Export the home directory (best practices have shared /home from /exports/home):

a. Create home exports directory

```
$ sudo mkdir /exports/dhome
```

b. Append the /etc/exports file to export the directory to the appropriate nodes

Content:

```
/exports/dhome 10.0.0.0/28(rw,no_root_squash, sync)
```

```
$ sudo vim /etc/exports
```

- c. Re-export the file system

```
$ sudo exportfs -r
```

14. Bind mount the new home file system on the head node

- a. Mount point (/dhome)

```
$ sudo mkdir /dhome
```

- b. Append entry to file system table (follow same steps as above)

- c. Mount directory (follow same steps as above)

- d. Verify mount exist (follow same steps as above)

15. Mount filesystems across the cluster using clush

```
$ sudo clush -w @compute yum install -y nfs-utils
```

```
$ sudo clush -w @compute mkdir /apps /dhome
```

```
$ sudo clush -w @compute \  
'echo master:/exports/apps /apps nfs defaults,noauto 0 0 >> /etc/fstab'
```

```
$ sudo clush -w @compute \  
'echo master:/exports/dhome /dhome nfs defaults,noauto 0 0 >>  
/etc/fstab'
```

```
$ sudo clush -w @compute mount /apps
```

```
$ sudo clush -w @compute mount /dhome
```