Head Node Setup

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.

NOTE: Steps 6, 7 have already been done for you, but we will walk through them to understand them better.

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:
    a. `$ ssh -i <key>.pem ec2-user@<ip>`
    b. `Or Connect with Putty from earlier document`
3. Copy PEM file to master node (execute this from your laptop).
    a. `$ scp -i <key>.pem <key>.pem ec2-user@<ip>:~/.ssh/id.pem`
    b. ***PuTTY USERS:*** **Download/Install/Run WinSCP**
        i. `Session -> New Session`
        ii. `hostname ->  <IP>`
        iii. `username -> ec2-user`
        iv. `click Advanced -> SSH -> Authentication`
        v. `in` *`Private Key File`* `Browse to your downloaded key` **`PPK.`**
        vi. `Save your session just in case`

vii. Connect to AWS account (Click Yes)

viii. Copy your **.pem** file over (**not the PPK**)

ix. Bottom right click the **6 hidden** to reveal hidden directories

x. double click .ssh

xi. drag student99.pem (your number in place of 99) over to window

xii. Highlight it, press **F2** rename it to **id.pem**

xiii. press **f9** Then remove read/write/execute access from group and other

xiv. Feel free to close winscp and return to putty.

4. Clone our Git repository to the master node

   a. `$ sudo yum install -y git`

   b. `$ 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.

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

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

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

   d. `$ ssh node02 # make sure you don't get permission denied`

      i. If there are key permission issues:

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

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

      iv. `bash /home/ec2-user/build-a-cluster/scripts/ssh_setup.sh # type yes if needed`

      v. Retry The Parent Step

   e. `$ exit`

f. `ssh node03 # make sure you don't get permission denied`

g. `$ exit`

h. `...`

i. `$ ssh node05 # make sure you don't get permission denied`

j. `$ exit`

k. You should try as the root user as well ;)

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

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

   a. `$ sestatus`

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

   ```
   SELINUX=disabled
   Reboot if necessary
   ```

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 turned off for the internal network, but often turned on for login nodes and service nodes where access is allowed from outside the cluster.

   a. `$ sudo systemctl stop firewalld.service`

   b. `$ sudo systemctl disable firewalld.service`

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

   a. `$ sudo hostname master`

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

   c. `$ sudo -i`

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

   e. `# exit`

   f. `$ exec bash # 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`

a. `sudo vim /etc/clustershell/groups.d/local.cfg` (alternatively /etc/clustershell/groups.d/cluster.yaml)
    i. compute: node[02-05]
    ii. all: node[01-05]

b. `$ clush -a hostname`

c. `$ clush -w @compute hostname`

    i. `node02: node02`

    ii. `node04: node04`

    iii. `node03: node03`

    iv. `node01: master`

    v. `node05: node05`

d. `$ clush -w master hostname`

    i. `master: Host key verification failed.`

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

    iii. `$ ssh master`

    iv. `$ exit`

    v. `$ sudo ssh master`

    vi. `# exit`

e. `$ clush -w master hostname`

f. `$ cluset -c @compute`

g. Clush group descriptions and configuration file overview.

10. Configure Puppet as a basic master server on the head node:

    a. Install Puppet server package on master node and Puppet on compute nodes

        i. `$ sudo yum install puppet-server`

        ii. `Make sure to hit yes.`

        iii. `$ sudo clush -w @compute yum install -y puppet`

iv. `$ sudo systemctl enable puppetmaster.service`

v. Add “puppet” entry to /etc/hosts

1. “puppet” should point to “master” (i.e., append “puppet” to line with “master” in it. Use a text editor like Vim.
2. `$ sudo vim /etc/hosts`
3. Copy to all compute nodes.
4. `$ sudo clush -w @compute --copy /etc/hosts`

vi. Edit `/etc/puppet/puppet.conf` with the appropriate DNS / hostfile names, `dns_alt_names = puppet,master,node01 in [main] section.`

vii. Generate the certificate for the puppet server and then hit CTRL + C

1. `$ sudo puppet master --verbose --no-daemonize`
2. CTRL+C after certificate generated
3. `$ sudo systemctl start puppetmaster.service`
4. `$ sudo systemctl status puppetmaster.service`
5. `$ sudo systemctl is-active puppetmaster.service`

viii. Start the puppet agents on the compute nodes:

1. `$ sudo clush -w @compute systemctl enable puppet.service`
2. `$ sudo clush -w @compute systemctl start puppet.service`

ix. Verify Puppet master is running and the compute node Puppet agents are running

1. `$ sudo systemctl status puppetmaster.service`
2. `$ sudo clush -w @compute -b systemctl is-active puppet.service`
3. `$ sudo clush -w @compute systemctl is-active puppet.service`
    a. Note order of output

x. Accept new Puppet Agent certs

1. `$ sudo puppet cert list`

2. `$ sudo puppet cert sign --all`

xi. Configure compute nodes as clients in `/etc/puppet/manifests/site.pp` (see `~/build-a-cluster/etc/puppet/manifests/site.pp`)

1. `$ sudo cp ~/build-a-cluster/etc/puppet/manifests/site.pp /etc/puppet/manifests/site.pp`
2. `$ cat /etc/puppet/manifests/site.pp`

xii. Run basic Puppet test

1. `$ sudo mkdir -p /etc/puppet/modules/test/{manifests,files}`
2. `$ sudo cp ~/build-a-cluster/etc/puppet/modules/test/manifests/init.pp /etc/puppet/modules/test/manifests/init.pp`
3. `$ sudo clush -w node02 puppet agent -t`
    a. If this fails.. try and see what setting may have been wrong.
    b. Then run:
    c. `bash ~/build-a-cluster/scripts/hostname_setup.sh`
    d. `bash ~/build-a-cluster/scripts/fix_puppet.sh`
    e. Retry parent step.

11. Configure a “Yum” repository to install software later which can be distributed in RPM format from our local master node.
    a. Install an Apache web server:
        i. `$ sudo yum install -y httpd`
    b. Install the yum-utils and createrepo packages on master node:
        i. `$ sudo yum install -y yum-utils createrepo`
    c. Navigate to the default web root location: /var/www/html and create a directory “hpc-repo”
        i. `$ cd /var/www/html`

ii. `$ sudo mkdir -p hpc-repo/Packages`

iii. `$ cd hpc-repo`

iv. `$ sudo createrepo $(pwd)`

d. Start the Apache web server and make sure it will run on reboot

i. `$ sudo systemctl start httpd.service`

ii. `$ sudo systemctl enable httpd.service`

Create a yum repo configuration file to house the packages in '`/etc/yum.repos.d/hpc.repo`' with the following content.

```
[hpc]
name=LCI HPC Build-A-Cluster
baseurl=http://master/hpc-repo/
enabled=1
gpgcheck=0
```

iii. `$ sudo yum repolist`

iv. `$ sudo clush -b -w @compute --copy /etc/yum.repos.d/hpc.repo`

v. `$ sudo clush -b -w @compute yum makecache fast`

12. Questions?