**Author: Aaron Valoroso**

**Title: Setting up a cluster on CentOS 7**

**Date: August 20th, 2017**

**OS: CentOS 7**

**Setup:**

To start you will need to machines that are connected to one another either through a switch or on the same network. Now this can be applied to as many machines as you want but we will just focus on two for now. My first machine will be called Master and the second will be Client and they both will be virtual machines on VirtualBox. If you don’t know how to setup a virtual machine then look for my document on how to setup a virtual machine for your specific needs.

**Part 1:**

* Type: su
* Type: yum update
* Type: yum install wget - This should already be installed
* Type: yum install net-tools - This should already be installed
* Type: yum install gcc-c++ kernel-devel make
* Type: yum groupinstall “Development Tools”
* Type: yum install gcc-gfortran - This should already be installed
* Type: yum -y install openssh-server - This should already be installed
* Type: chkconfig sshd on
* Type: service sshd start
* Type: yum install nfs-utils - This should already be installed
* Type: yum install portmap - This should already be installed
* Type: cd /home/”username”/
* Type: mkdir mpich
* Type: cd /home/”username”/mpich
* Type: wget <http://www.mpich.org/static/downloads/3.2/mpich-3.2.tar.gz>
* Type: tar xfz mpich-3.2.tar.gz
* Type: cd mpich-3.2
* Type: ./configure --prefix=/usr/local
* Type: sudo make; sudo make install
* Type: reboot
* Type: su
* Type: systemctl stop firewalld
* Type: systemctl disable firewalld
* Type: sudo visudo
  + Under Root Privileges add mpiuser (tab) ALL=(ALL) (tab) ALL
* Type: sudo adduser mpiuser
* Type: passwd mpiuser
* Type: su - mpiuser
* Type: chkconfig nfs on
* Type: service rpcbind start
* Type: service nfs start
* Type: mkdir cloud
* Type: exit

**Part 2 on Master:**

* Type: sudo nano /etc/sysconfig/network-scripts/ifcfg-enp0s8
  + Make sure the ONBOOT is yes, and then change the BOOTPROTO from dhcp to static. Finally add IPADDR=(some ip address ex. 192.168.38.202), and add the NETMASK=(some netmask ex. 255.255.255.0) underneath the IPADDR.
* Type: hostnamectl set-hostname node#
* Type: sudo nano /etc/exports
  + /home/mpiuser/cloud \*(rw,sync,no\_root\_squash,no\_subtree\_check)
* Type: sudo exportfs –a

**Part 3 on Clients:**

* Type: sudo nano /etc/sysconfig/network-scripts/ifcfg-enp0s8
  + Make sure the ONBOOT is yes, and then change the BOOTPROTO from dhcp to static. Finally add IPADDR=(some ip address ex. 192.168.38.202), and add the NETMASK=(some netmask ex. 255.255.255.0) underneath the IPADDR.
* Type: hostnamectl set-hostname node#
* Type sudo mount –v (server enp0s8 IP):/home/mpiuser/cloud ~/cloud
* Type: sudo nano /etc/fstab
  + “server enp0s8 IP”:/home/mpiuser/cloud /home/mpiuser/cloud nfs

**Part 4 the SSHing:**

For the sake of simplicity go onto every mpiuser and original user on each machine and ssh into one of the other machines. Don’t do anything just exit. Doing this will create .ssh directory that we need in order to setup password less entry.

**Master**

* Do not be root for the following. The mpiuser on the master machine needs to ssh into each mpiuser user on the client machines and also to the user you created when you installed the operating system. This will create the .ssh directory in home.
  + Type: ssh-keygen –t rsa
    - Just hit enter for the following output, if you want to secure the key some more the pay attention to the following.
  + Type: cd ~/.ssh
  + Type: scp id\_rsa.pub [mpiuser@192.168.20.202](mailto:mpiuser@192.168.20.202):
    - Don’t forget the colon at the end of the command.
  + Type: scp id\_rsa.pub [node2@192.168.20.202](mailto:node2@192.168.20.202):

SSH into each client and move the file that we transferred over before hand into the .ssh directory on both users in the client machine. Then:

* + Type: cp id\_rsa.pub authorized\_keys
  + Type: rm id\_rsa.pub

Exit the client machine(s) and remove the id\_rsa.pub file from the mpiuser in the master’s .ssh directory. We do not need the public key anymore. The id\_rsa file without the extension is the private key, don’t remove this file.

**Clients**

* Do not be root for the following. Now we will repeat the same steps for each mpiuser on the client machines. But in this step we want to create the key’s again from the client and put them into the mpiuser on the master, and also the original user on the machine as well.
  + Type: ssh-keygen –t rsa
  + Type: cd ~/.ssh
  + Type: scp id\_rsa.pub [mpiuser@192.168.20.201](mailto:mpiuser@192.168.20.201):
    - Don’t forget the colon at the end of the command.
  + Type: scp id\_rsa.pub [node1@192.168.20.201](mailto:node1@192.168.20.201):
    - Don’t forget the colon at the end of the command.

SSH into the master machine mpiuser and on the original user, and move the file into the .ssh directory. The next step is tricky because from each client you will have to copy each individual key and copy it into the authorized\_keys file. Remember if it is not there then create the file and just each key on each new line.

* + Type: rm id\_rsa.pub

Exit the master machine and remove the id\_rsa.pub file from the mpiuser in the client’s .ssh directory. We do not need the public key anymore. The id\_rsa file without the extension is the private key, don’t remove this file.

**Other:**

You should now be able to ping either machine. If not, then use Google. I will have another document that will go over other related networking topics.