Network setup

As the root user execute the command "ifconfig" on each host. Take a note of ipaddress's of all machines I have a home LAN so my addresses are class C which might be in the format 192.168.192.x. For my cluster setup below are ip's configured for reverse DNS by adding the below entries to /etc/hosts file note the host should have FQDN eg server01.texas.com

```
192.168.192.22     server01.texas.com          server01
192.168.192.61     server03.texas.com          server03
```

Configure HOSTNAME by adding the hostname to the file /etc/sysconfig/network. Do the same to all the all nodes.

```
[root@server01 etc] # vi /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=server01.texas.com
```

Edit the ifcfg-eth0

```
[root@server01 ] vi /etc/sysconfig/network-scripts
DEVICE=eth0
HWADDR=00:15:17:12:F2:50
TYPE=Ethernet
UUID=6428569e-e729-4985-974d-e87fc81a8796
NM_CONTROLLED=yes
ONBOOT=yes
BOOTPROTO=dhcp
```

To rename the host without rebooting run the below command on every host in the cluster

```
[root@server01 etc]# hostname   server01.texas.com
```

Configuring Passwordless SSH for root user

Execute the below commands from the server01 Master Node: Just press "Enter" when prompted for a passphrase accept defaults. Two files will be created in the folder **/root/.ssh**

```
[root@server01 ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
```

when prompted for a passphrase. Two files will be created in the folder **/root/.ssh**

```
[root@server01 .ssh]#  ls -al
id_rsa  id_rsa.pub
```

Copy the contents of id_rsa.pub to authorized_keys and copy both the files to other nodes to all the remote-host's .ssh/authorized_key.

```
[root@server01 .ssh]# cat .ssh/id_rsa.pub | ssh root@192.168.192.18 'cat >>
.ssh/authorized_keys'
[root@server01 .ssh]# cat .ssh/id_rsa.pub | ssh root@192.168.192.61 'cat >>
.ssh/authorized_keys'
```

Test the use "ssh" to login from Server01 to other 3 servers of the clusters identified above. If Passwordless SSH is not configured, the command will prompt to Enter the Password of the other machine. The first attempt to logon with request for the password that's normal but thereafter it won't prompt for the password as shown below

```
[root@server01 ~]# ssh root@server03
The authenticity of host 'server03 (192.168.192.61)' can't be established.
RSA key fingerprint is ef:d6:b2:05:0f:17:22:3d:e5:be:74:e5:f3:72:20:63.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server03' (RSA) to the list of known hosts.
Last login: Sun Dec 27 04:13:22 2015 from server02.texas.com
[root@server03 ~]#
```

No prompt for password prompt

```
[root@server01 ~]# ssh root@server02
Last login: Sun Dec 27 11:16:56 2015 from server01.texas.com
[root@server02 ~]# exit
```

Copy the generated keys to all the hosts

NOTE: during the above first time connection you are prompted for the password of the remote machine.
But now below you can login to other machines directly using ssh. No password entry will be prompted.

```
[root@server01 .ssh]# ssh server03
Last login: Sun Nov 29 22:30:54 2015 from server01.texas.com
[root@server03 ~]# exit
logout
Connection to server03 closed.
[root@server01 .ssh]# ssh server02
Last login: Sun Nov 29 22:25:59 2015 from 192.168.192.22
[root@server02 ~]# exit
logout
Connection to server02 closed.
```

Disable firewall / iptables on all the cluster hosts

Execute the below command on all nodes to disable the firewall, type the following command as the root user to disable firewall for IPv6:

```
[root@server02 ~]#  service ip6tables stop
ip6tables: Setting chains to policy ACCEPT: filter          [OK]
ip6tables: Flushing firewall rules:                         [OK]
ip6tables: Unloading modules:                               [OK]
```

Disable on boot startup of iptables

```
[root@server02 ~]# chkconfig ip6tables off
[root@server02 ~]# service iptables status
iptables: Firewall is not running.
```

IPv4

```
[root@server01 etc] # service iptables save
iptables: Saving firewall rules to /etc/sysconfig/iptables:     [OK]
```

```
[root@server01 etc] # service iptables stop
iptables: Setting chains to policy ACCEPT: filter           [OK]
iptables: Flushing firewall rules:                          [OK]
iptables: Unloading modules:                                [OK]
[root@server01 etc]# chkconfig iptables off
```

Turn the iptables off

```
[root@server01 etc] #  chkconfig iptables off
```

Configure the sshd daemon to start on boot

```
[root@server01 ~]# chkconfig --level 345 sshd on
[root@server01 ~]# service sshd restart
Stopping sshd:                                  [  OK  ]
Starting sshd:                                  [  OK  ]
```

Disable SELinux

SELinux must be disabled for Ambari to function. To temporarily disable SELinux, run the following command on each host in your cluster:

```
[root@server01 ~]# setenforce 0
```

Permanently disabling SELinux so that on system reboot it does not restart edit the SELinux config and set SELINUX to disabled. on each host:

```
root@server01 ]# vi /etc/selinux/config
```

Add the below lines

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - SELinux is fully disabled.
SELINUX=disabled
# SELINUXTYPE= type of policy in use. Possible values are:
#    targeted - Only targeted network daemons are protected.
#    strict - Full SELinux protection.
SELINUXTYPE=targeted
```

Disable Transparent Huge Pages (THP)

This must be disabled on all the hosts otherwise the Ambari install will fail

```
[root@server02 ~]# echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
[root@server02 ~]# echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
```

To disable or make these changes persistent across reboots I add this to the bottom of my vi /etc/rc.local

```

#disable THP at boot time
if test -f /sys/kernel/mm/redhat_transparent_hugepage/enabled; then
echo never > /sys/kernel/mm/redhat_transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/redhat_transparent_hugepage/defrag; then
echo never > /sys/kernel/mm/redhat_transparent_hugepage/defrag
fi
```

To validate THP is disabled, I run the below three commands, or any variant you choose from here .

```
[root@server02 ~]# cat /sys/kernel/mm/redhat_transparent_hugepage/defrag
      always madvise [never]
 [root@server02 ~]#  cat /sys/kernel/mm/redhat_transparent_hugepage/enabled
       always madvise [never]
```

Configure the NTPD services

You must setup the NTPD server on CentOS to successfully implement Ambari follow the below steps to install and start the NTPD server. As the root user.

```
[root@server01 ~]# yum install ntp ntpdate
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
 * base: linuxsoft.cern.ch
 * epel: mirror.serverbeheren.nl
 * extras: linuxsoft.cern.ch
 * updates: linuxsoft.cern.ch
…
…
ntpdate.x86_64 0:4.2.6p5-5.el6.centos.2
Complete!
```

Turn on the service:

```
[root@server01 ~]#  chkconfig ntpd on
```

Synchronize the system clock with 0.pool.ntp.org server

```
[root@server01 bin]# ntpdate pool.ntp.org
 2 Jan 21:42:49 ntpdate[5101]: 31.3.135.236 rate limit response from server.
 2 Jan 21:43:56 ntpdate[5101]: step time server 212.51.144.44 offset 67.520940 sec
```

Check the NTPD services configure it start at system boot

```
[root@server01 bin]# chkconfig --list ntpd
ntpd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

Start the NTP server. The following will continuously adjusts system time from upstream NTP server. No need to run ntpdate:

```
[root@server01 ~]# /etc/init.d/ntpd start or
[root@server01 ~]#  service ntpd start
Starting ntpd:                                             [  OK  ]
```