Hadoop2 Multi node cluster Setup

(On CentOS)

1. Assumptions:

a. Java is installed and JAVA_HOME=/usr/java/latest
b. Hadoop software is downloaded and is available at /home/<user>/Downloads/hadoop2
c. We have 3 VMs named master, slave1 and slave2
d. IP addresses: (replace the IPs with your IP addresses)

master	192.168.10.10
slave1	192.168.10.11
slave2	192.168.10.12

2. To be done on all machines (change the HOSTNAME value) as root

File: /etc/sysconfig/network	
master	HOSTNAME=master
slave1	HOSTNAME=slave1
slave2	HOSTNAME=slave2

3. Find out the IP address of your machine using 'ifconfig' command

```
$ifconfig          //it will list the IP of your machine
```

4. Update the /etc/hosts file with IP and hostname (On all Machines as root)

File: /etc/hosts	
master	192.168.10.10
slave1	192.168.10.11
slave2	192.168.10.12

5. Create common group and user account (on all machines as root)

Before starting the hadoop setup it is important to create a common user account and a group in all computers. Create the group training and user training:

Step1: Log in as root.

Step2: Create the group training using the groupadd utility followed by the name of the group, in this format:

```
#groupadd n training
```

Where n is an unused group ID greater than 100.

Step3: Create the user **training** using the useradd utility followed by the group (training) and user name (training) in this format:

```
#useradd -u n -g training  training
```

Step4: Create a password for the user training. To do this, use the passwd utility and the following command:

```
passwd training
```

6. Create passphrase less ssh (only on Master as user training)

```
$ssh-keygen –t rsa –P “”
```

Follow the on screen instructions and go by the defaults suggested (Press Enter key for the defaults)

```
$cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
$chmod 640 $HOME/.ssh/id_rsa.pub
$chmod 640 $HOME/.ssh/authorized_keys

Test the ssh login without passphrase
$ssh master
You should login without passphrase
```

7. The master log in must have password-less log in authorities to all slaves. (On Master)

```
training@master:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub training@slave1
training@master:~$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub training@slave2
```

8 . Create the necessary folders for hadoop installation (on all machines)

```
$sudo mkdir /u01
$sudo chown –R training:training /u01
$mkdir /u01/hadoop-work
$mkdir /u01/hadoop-work/tmp
$mkdir /u01/hadoop-work/name
$mkdir /u01/hadoop-work/data
```

9. Install hadoop(on all machines)

a. Untar the tarball and create a sym link

```
$tar zxvf $HOME/Downloads/hadoop2/hadoop-<version> -C /u01/
$ln –s /u01/hadoop-<version> /u01/hadoop
```

b. Update JAVA_HOME in /u01/hadoop/etc/hadoop/hadoop-env.sh (use vi editor)

```
vi /u01/hadoop/etc/hadoop/hadoop-env.sh

JAVA_HOME=/usr/java/latest
```

c. Update $HOME/.bashrc file as below, use vi editor)

```
vi $HOME/.bashrc

export HADOOP_PREFIX=/u01/hadoop

export HADOOP_MAPRED_HOME=$HADOOP_PREFIX

export HADOOP_COMMON_HOME=$HADOOP_PREFIX

export HADOOP_HDFS_HOME=$HADOOP_PREFIX

export YARN_HOME=$HADOOP_PREFIX

export PATH=$PATH:$HADOOP_PREFIX/bin:$HADOOP_PREFIX/sbin
```

d. Load .bashrc as

```
$source $HOME/.bashrc
```

e. **Edit $HADOOP_PREFIX/etc/hadoop/core-site.xml as follows:**

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/u01/hadoop-work/tmp </value>
</property>
</configuration>
```

f. **edit $HADOOP_PREFIX/etc/hadoop/hdfs-site.xml as follows:**

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.name.dir</name>
        <value>file:///u01/hadoop-work/name</value>
    </property>
    <property>
        <name>dfs.data.dir</name>
        <value> file:///u01/hadoop-work/data</value>
    </property>
</configuration>
```

g. **Create and update $HADOOP_PREFIX/etc/hadoop/mapred-site.xml as follows:**

```
$ mv mapred-site.xml.template  mapred-site.xml
$ vi mapred-site.xml

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
Configure $HADOOP_INSTALL/etc/hadoop/yarn-site.xml as follows:
<configuration>
<property>
        <name>yarn.resourcemanager.hostname</name>
        <value>master</value>
    </property>
<property>
   <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
    <name>yarn.nodemanager.aux-
services.mapreduce.shuffle.class</name>
   <value>org.apache.hadoop.mapred.ShuffleHandler</value>
```

```
</property>
</configuration>
```

h. Edit **$HADOOP_PREFIX/etc/hadoop/slaves file as follows (Only on Master Node):**
vi $HADOOP_PREFIX/etc/hadoop/slaves
enter the following lines (slave names)
master

```
slave1
slave2
```

i. **Format Namenode (only on master node)**

```
$ hdfs namenode -format
```

10. **Run hadoop daemons (from master node only)**

```
$start-dfs.sh
$start-yarn.sh
```

Check using "jps"
$jps

will show the following daemons
On Master
NamNode
DataNode
SecondaryNameNode
NodeManager
ResourceManager

On Slaves
DataNode
NodeManager

Your hadoop cluster of 3 nodes is completed.

Now you may submit jobs to resource manager from master node