MANUAL FOR MASTER

- The purpose of creating hadoop multimode cluster is to distribute the work and get the result in less time.
- The most important thing to know is ‘the distribution and final result given done by master node based on our given code’.
- The following steps must be followed to create a hadoop master node in multimode cluster.

STEP.1:   EDITING HOSTS FILE

- Go to terminal and login as hduser.
- Run the following code—   sudo gedit /etc/hosts
- Alter the file as per ua number of slaves by providing the number of slaves with corresponding IP Addresses(in the correct intendation as the system must recognize the address and slave names)

```
127.0.0.1   localhost
127.0.1.1   pc1-desktop   master
172.22.14.1     master
172.22.14.2     slave1
172.22.14.3     slave2
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- 172.22.14.1   master ->this is master node which we have mentioned as master
- 172.22.14.2   slave1->this is slave with corresponding ip address
- 172.22.14.3   slave2->this is another slave given as slave2
- Note that the name can be anything as per our convenience,but the ip address must be given according to the network.
- Save the file and close.

Step.2   EDITING MASTER AND SLAVE FILES

- Go to terminal and change the directory to /usr/local/hadoop/etc/hadoop
- For that use the following command— cd /usr/local/hadoop/etc/hadoop
- Check whether the masters and slaves file is exsits.
- If not create using following commands--- sudo gedit masters
- Edit the masters file. Do the following changes---

master

- Edit the slaves file--- sudo gedit slaves
- Do the following changes---

Master
slave1

slave2

- As master is slave to itself master also should be mentioned in slaves file or you cannot get the datanode for jps in master.
- The slave1 and slave2 is given for example purpose...it should be givenm according to your cluster.(as what you have choosen as slaves)

Step.3 CHANGING PERMISSIONS FOR DATANODE AND NAMENODE

- Create the namenode and datanode using following commands---

```
sudo mkdir /usr/local/hadoop_store/namenode
sudo mkdir /usr/local/hadoop_store/datanode
```

- Change the permissions using following commands—

```
sudo chown hduser:hadoop -R /usr/local/hadoop_store/namenode
sudo chown hduser:hadoop -R /usr/local/hadoop_store/datanode
```

Step.4 CONNECTING MASTER AND SLAVES

- Master should know that it is the master and it should be able to aquire the permissions to do its job. So for this we have to give the id permissions.
- Master should be able to connect to itself without password.
- For this use the following command-- `ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@master`
- Then, master should be able to know its slaves.
- master should be able to connect to slaves without password.
- So connect the slaves using following command(observe the end of the code the slave name must be changed as per your slaves)
- `ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@slave1`
- `ssh-copy-id -i $HOME/.ssh/id_rsa.pub hduser@slave2`
- The above commands are given as per our considered slaves i.e.slave1 and slave2.

Step.5 UPDATE CORE-SITE.XML FILE

- Go to the directory /usr/local/hadoop/etc/hadoop
- For that Use the following command— `cd /usr/local/hadoop/etc/hadoop`
- Open the core-site.xml using following code-- `sudo gedit core-site.xml`
- Paste these lines into <configuration> tag OR Just update it by replacing localhost with master

```
<property>
<name>fs.default.name</name>
<value>hdfs://master:54310</value>
</property>
```

- Save and exit.

Step.6 UPDATE HDFS-SITE.XML FILE

- Open the hdfs-site.xml file using following command-- sudo gedit hdfs-site.xml
- Paste/Update these lines into <configuration> tag

```
<property>
  <name>dfs.replication</name>
  <value>3</value>
</property>

(in masters remove datanode property if master is not a
slave.In slaves remove namenode property)

<property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_store/namenode</value>
 </property>


 <property>
   <name>dfs.datanode.data.dir</name>
   <value>file:/usr/local/hadoop_store/datanode</value>
 </property>
```

- Save the file and exit.

Step.7 UPDATE YARN-SITE.XML FILE

- Open the yarn-site.xml file using following command— sudo gedit yarn-site.xml
- paste/Update these lines into <configuration> tag

```
<property>
      <name>yarn.resourcemanager.resource-
tracker.address</name>
      <value>master:8025</value>
</property>
<property>
      <name>yarn.resourcemanager.scheduler.address</name>
      <value>master:8035</value>
</property>
<property>
      <name>yarn.resourcemanager.address</name>
      <value>master:8050</value>
</property>
```

- Save and exit.

Step.8 UPDATE mapred-site.xml FILE

- Open mapred-site.xml file using following command-- sudo gedit mapred-site.xml
- Paste/Update these lines into <configuration> tag

```
  <property>
      <name>mapreduce.job.tracker</name>
      <value>HadoopMaster:54311</value>
</property>
<property>
      <name>mapred.framework.name</name>
      <value>yarn</value>
</property>
```

Step.9 FORMATING NAMEODE

- By default hadoop may create fsimage files in the namenode so formatting namenode is required.
- Format namenode.
- Use the following command to format--   hadoop namenode -format

Prepared by

Mani Kiran Gadige