EXP. 10: INSTALL HADOOP 2.X AND CONFIGURE THE NAME NODE AND DATA NODE.

AIM:

PROCEDURE:

Step 7 - Modify Hadoop config files
//Hadoop Environmental variable setting – The following files will be modified

1. ~/.bashrc
2. /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/hadoop-env.sh
3. /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/core-site.xml
4. /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/hdfs-site.xml
5. /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/yarn-site.xml
6. /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml.template

$ sudo nano ~/.bashrc

// Add the following lines at the end of the file

```
export JAVA_HOME=/usr/lib/jvm/java-8-oracle export
HADOOP_HOME=/usr/local/hadoop/hadoop-2.7.2
export PATH=$PATH:$HADOOP_HOME/bin export
PATH=$PATH:$HADOOP_HOME/sbin export
HADOOP_MAPRED_HOME=$HADOOP_HOME export
HADOOP_COMMON_HOME=$HADOOP_HOME export
HADOOP_HDFS_HOME=$HADOOP_HOME export
YARN_HOME=$HADOOP_HOME
HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-D.java.library.path=$HADOOP_HOME/lib"
export PATH=$PATH:/usr/local/hadoop/hadoop-2.7.2/bin
```

// Configure Hadoop Files

$ cd /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/

$ sudo nano hadoop-env.sh

// Add following line in hadoop-env.sh – Set JAVA variable in Hadoop

```
# The java implementation to use. export
JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

// Create datanode and namenode

$ sudo mkdir –p /usr/local/hadoop_tmp/hdfs/namenode

$ sudo mkdir –p /usr/local/hadoop_tmp/hdfs/datanode

// Changing ownership to hadoop_tmp

$ sudo chown –R hduser:hadoop /usr/local/hadoop_tmp

// Edit hdfs-site.xml

$ sudo nano hdfs-site.xml

// Add the following lines between <configuration> …… </configuration>

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
</configuration>
```

// Edit core-site.xml

$ sudo nano core-site.xml

// Add the following lines between <configuration> …… </configuration>

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

// Edit yarn-site.xml

$ sudo nano yarn-site.xml

// Add the following lines between <configuration> …… </configuration>

```
<configuration>
<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.Shuffle-Handler</value>
</property>
</configuration>
```

// Edit mapred-site.xmsudo

$ cp /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml

$ sudo nano mapred-site.xml

// Add the following lines between <configuration> …… </configuration>

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Step-8 – Format Hadoop File System
$ cd /usr/local/hadoop/hadoop-2.7.2/bin
$ hadoop namenode -format

Step 9 - Start Hadoop
$ cd /usr/local/hadoop/hadoop-2.7.2/sbin

// Starting dfs services

$ start-dfs.sh

// Starting mapreduce services

$ start-yarn.sh

$ jps

Step 10 - Check Hadoop through web UI
Go to browser type http://localhost:8088 – All Applications Hadoop Cluster

Go to browser type http://localhost:50070 – Hadoop Namenode

Step 11 - Stop Hadoop

$ stop-dfs.sh

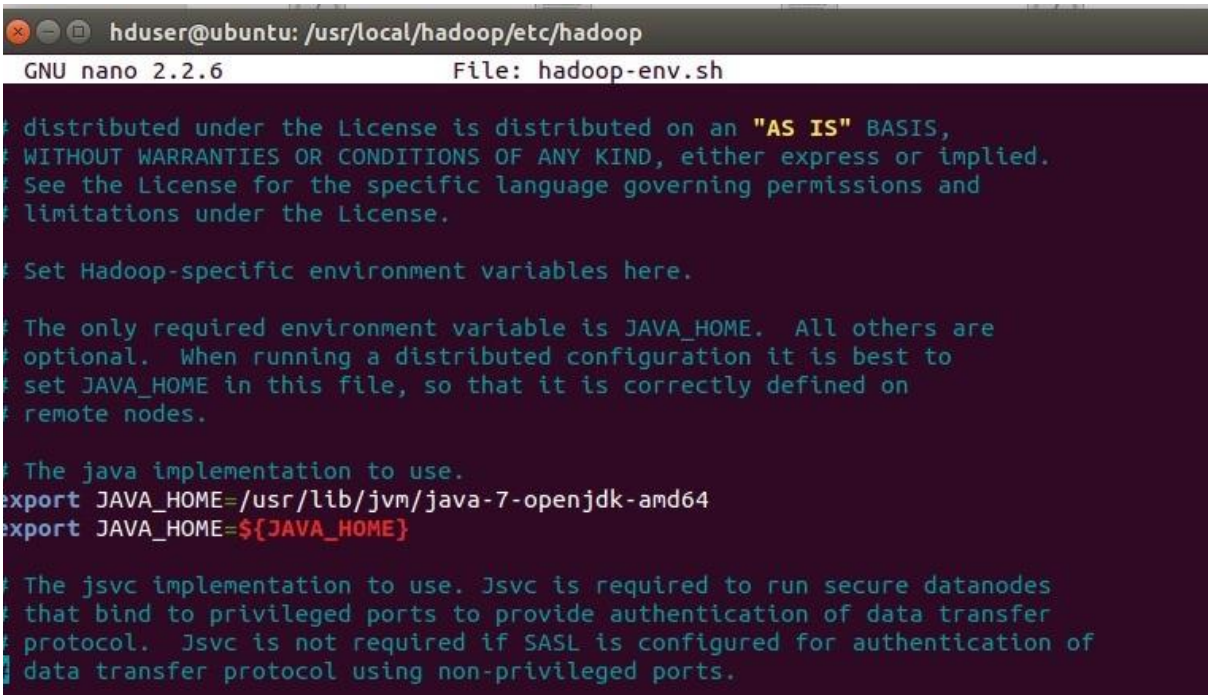
$ stop-yarn.sh

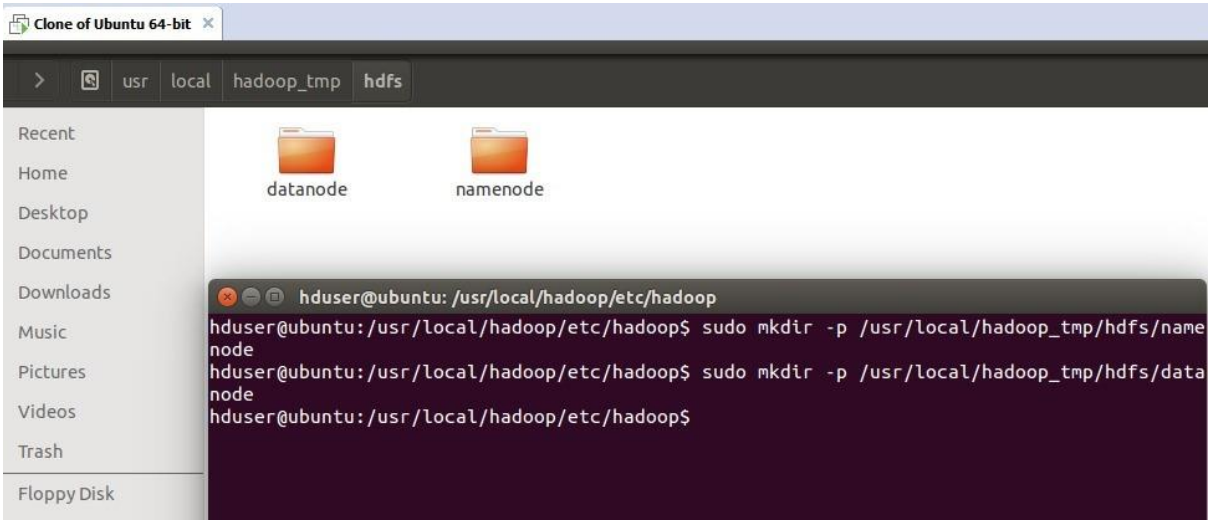
IMPLEMENTAION:

Clone of Ubuntu 64-bit

```
  GNU nano 2.2.6                                File: /home/hduser/.bashrc

# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi


#HADOOP VARIABLES START
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
#HADOOP VARIABLES END
```

```
hduser@ubuntu:/home$ cd ..
hduser@ubuntu:/$ cd usr
hduser@ubuntu:/usr$ cd local
hduser@ubuntu:/usr/local$ cd hadoop
hduser@ubuntu:/usr/local/hadoop$ cd etc
hduser@ubuntu:/usr/local/hadoop/etc$ cd hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ ls
capacity-scheduler.xml      httpfs-env.sh            mapred-env.sh
configuration.xsl           httpfs-log4j.properties  mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret  mapred-site.xml
core-site.xml               httpfs-site.xml          mapred-site.xml.template
hadoop-env.cmd              kms-acls.xml             slaves
hadoop-env.sh               kms-env.sh               ssl-client.xml.example
hadoop-metrics2.properties  kms-log4j.properties     ssl-server.xml.example
hadoop-metrics.properties   kms-site.xml             yarn-env.cmd
hadoop-policy.xml           log4j.properties         yarn-env.sh
hdfs-site.xml               mapred-env.cmd           yarn-site.xml
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$
```


hduser@ubuntu: /usr/local/hadoop/etc/hadoop
GNU nano 2.2.6 File: hadoop-env.sh
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
Set Hadoop-specific environment variables here.
The only required environment variable is JAVA_HOME. All others are
optional. When running a distributed configuration it is best to
set JAVA_HOME in this file, so that it is correctly defined on
remote nodes.
The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export JAVA_HOME=${JAVA_HOME}
The jsvc implementation to use. Jsvc is required to run secure datanodes
that bind to privileged ports to provide authentication of data transfer
protocol. Jsvc is not required if SASL is configured for authentication of
data transfer protocol using non-privileged ports.


Clone of Ubuntu 64-bit
usr
local
hadoop_tmp
hdfs
Recent
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Floppy Disk
datanode
namenode
hduser@ubuntu: /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/name
node
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/data
node
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$

RESULT: