

EX.NO: 09

REG.NO: 210701303

HADOOP SET UP SINGLE HADOOP CLUSTER AND SHOW THE PROCESS USING WEB UI

AIM:

To set-up one node Hadoop cluster.

PROCEDURE:

1. System Update
2. Install Java
3. Add a dedicated Hadoop user
4. Install SSH and setup SSH certificates
5. Check if SSH works
6. Install Hadoop
7. Modify Hadoop config files
8. Format Hadoop filesystem
9. Start Hadoop
10. Check Hadoop through web UI
11. Stop Hadoop

THEORY

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from a single server to thousands of machines, each offering local computation and storage.

HADOOP ARCHITECTURE

Hadoop framework includes following four modules:

Hadoop Common: These are Java libraries and utilities required by other Hadoop modules.

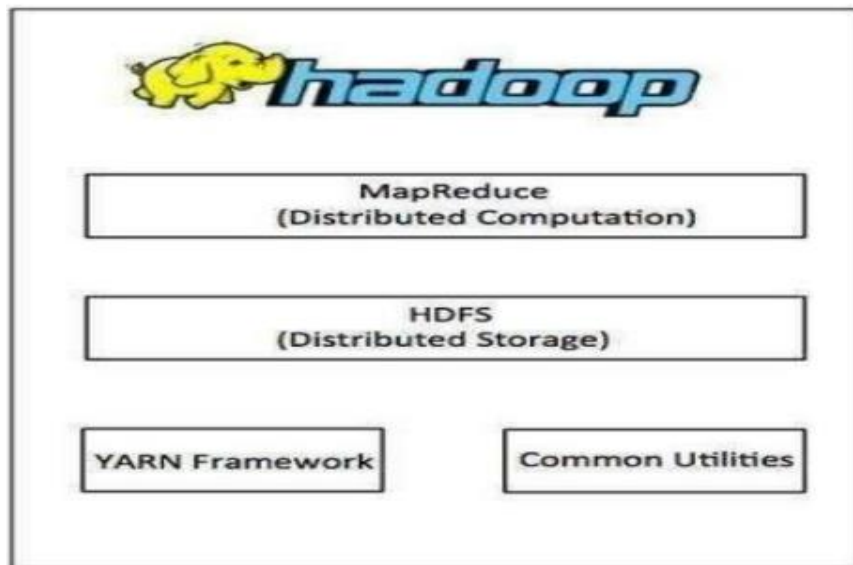
These libraries provide filesystem and OS level abstractions and contain the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management

Hadoop Distributed File System (HDFS): A distributed file system that provides highthroughput access to application data.

Hadoop MapReduce: This is a YARN-based system for parallel processing of large data sets.

We can use following diagram to depict these four components available in Hadoop Framework.



PROCEDURE

Step 1 – System Update

\$ sudo apt-get update

```
ubuntu@ubuntu-VirtualBox: ~$ sudo apt-get update
[sudo] password for ubuntu:
Hit http://in.archive.ubuntu.com wily InRelease
Get:1 http://security.ubuntu.com wily-security InRelease [65.9 kB]
Get:2 http://in.archive.ubuntu.com wily-updates InRelease [65.9 kB]
Get:3 http://security.ubuntu.com wily-security/main Sources [53.8 kB]
Hit http://in.archive.ubuntu.com wily-backports InRelease
Get:4 http://security.ubuntu.com wily-security/restricted Sources [2,854 B]
Get:5 http://security.ubuntu.com wily-security/universe Sources [13.9 kB]
Get:6 http://security.ubuntu.com wily-security/multiverse Sources [2,784 B]
Get:7 http://security.ubuntu.com wily-security/main amd64 Packages [172 kB]
Get:8 http://security.ubuntu.com wily-security/restricted amd64 Packages [10.9 kB]
Get:9 http://security.ubuntu.com wily-security/universe amd64 Packages [56.2 kB]
Get:10 http://security.ubuntu.com wily-security/multiverse amd64 Packages [6,248 B]
Get:11 http://security.ubuntu.com wily-security/main i386 Packages [169 kB]
Get:12 http://security.ubuntu.com wily-security/restricted i386 Packages [10.8 kB]
100% [Waiting for headers] [Waiting for headers] 73.8 kB/s 0s
```

Step 2 – Install Java and Set JAVA_HOME

//This first thing to do is to setup the webupd8 ppa on your system. Run the following command and proceed.

```
$ sudo apt-add-repository ppa:webupd8team/java
```

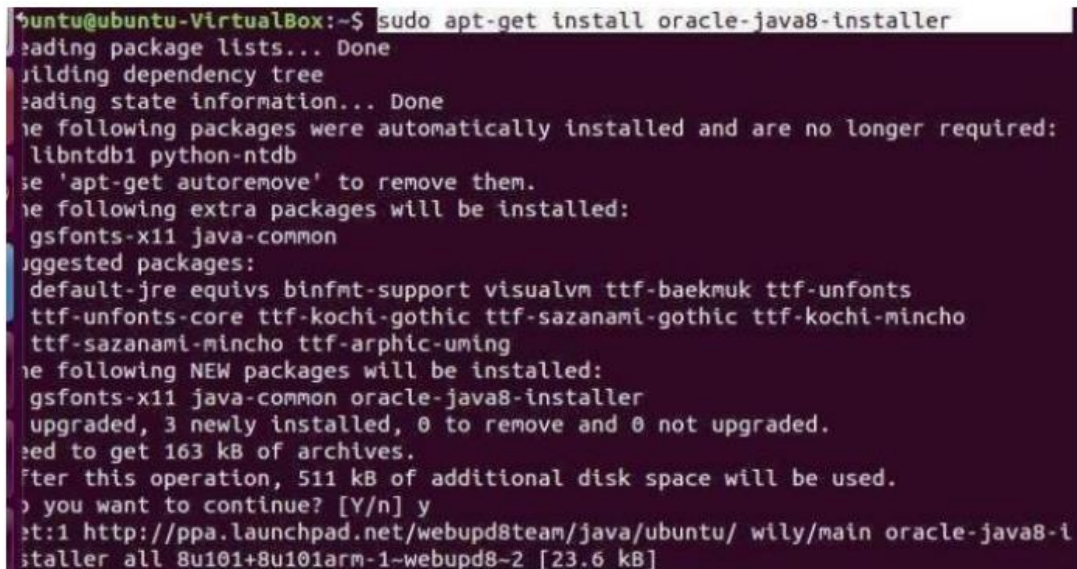
```
$ sudo apt-get update
```

//After setting up the ppa repository, update the package cache as well.

//Install the Java 8 installer

```
$ sudo apt-get install oracle-java8-installer
```

// After the installation is finished, Oracle Java is setup. Run the java command again to check the version and vendor.

A terminal window screenshot showing the command 'sudo apt-get install oracle-java8-installer' being executed. The output shows the package lists being read, the dependency tree being built, and state information being read. It lists packages that were automatically installed and are no longer required (libntdb1, python-ntdb) and suggests removing them with 'apt-get autoremove'. It also lists extra packages to be installed (gsfonts-x11, java-common) and suggested packages (default-jre, equivs, binfmt-support, visualvm, etc.). The NEW packages to be installed are gsfonts-x11, java-common, and oracle-java8-installer. The operation will upgrade 3 packages, install 3 new ones, and not upgrade 0. It will get 163 kB of archives and use 511 kB of additional disk space. The user is prompted to continue (Y/n) and responds with 'y'. The final output shows the source of the packages: 'http://ppa.launchpad.net/webupd8team/java/ubuntu/wily/main oracle-java8-installer all 8u101+8u101arm-1-webupd8-2 [23.6 kB]'.

OR

```
$ sudo apt-get install default-jdk
```

```
$ java -version
```

Step 3 – Add a dedicated Hadoop user

```
$ sudo addgroup hadoop
```

```
ubuntu@ubuntu-VirtualBox:~$ sudo addgroup hadoop
Adding group 'hadoop' (GID 1001) ...
Done.
```

\$ sudo adduser --ingroup hadoop hduser

```
ubuntu@ubuntu-VirtualBox:~$ sudo adduser --ingroup hadoop hduser
Adding user 'hduser' ...
Adding new user 'hduser' (1001) with group 'hadoop' ...
Creating home directory '/home/hduser' ...
Copying files from '/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
```

// Add hduser to sudo user group

\$ sudo adduser hduser sudo

```
ubuntu@ubuntu-VirtualBox:~$ sudo adduser hduser sudo
Adding user 'hduser' to group 'sudo' ...
Adding user hduser to group sudo
Done.
ubuntu@ubuntu-VirtualBox:~$
```

Step 4 – Install SSH and Create Certificates

\$ sudo apt-get install ssh

```
ubuntu@ubuntu-VirtualBox:~$ sudo apt-get install ssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libntdb1 python-ntdb
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libck-connector0 ncurses-term openssh-server openssh-sftp-server
  ssh-import-id
Suggested packages:
  rssh molly-guard monkeysphere
The following NEW packages will be installed:
  libck-connector0 ncurses-term openssh-server openssh-sftp-server ssh
  ssh-import-id
0 upgraded, 6 newly installed, 0 to remove and 8 not upgraded.
Need to get 661 kB of archives.
```

\$ su hduser

```
ubuntu@ubuntu-VirtualBox:~$ su hduser
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

\$ ssh-keygen -t rsa -P ""

```
hduser@ubuntu-VirtualBox:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:K/F5oNmAqhY02Axp0vzew4EnrN+UGgDTgxIiFPHpT7Q hduser@ubuntu-VirtualBox
The key's randomart image is:
+----[RSA 2048]----+
|=@o
|@.*.
|=* * o
|.o= *.+
|. .=.EooS
|...= *B +
|O. *+.= .
|o o .. .
|o
+-----[SHA256]-----+
```

// Set Environmental variables

\$ cat \$HOME/.ssh/id_rsa.pub >> \$HOME/.ssh/authorized_keys

```
hduser@ubuntu-VirtualBox:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

Step 6 – Install Hadoop

\$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.8.4/hadoop-2.8.4.tar.gz

// Extract Hadoop-2.8.4

\$ sudo tar xvzf hadoop-2.8.4.tar.gz

```
hduser@ubuntu-VirtualBox:~$ tar xvzf hadoop-2.7.2.tar.gz
```

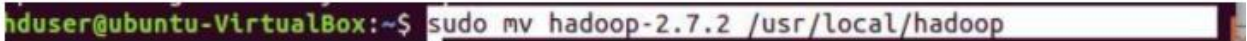
// Create a folder 'hadoop' in /usr/local

\$ sudo mkdir -p /usr/local/hadoop

```
hduser@ubuntu-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop
[sudo] password for hduser:
```

// Move the Hadoop folder to /usr/local/hadoop


```
$ sudo mv hadoop-2.8.4 /usr/local/hadoop
```



```
hduser@ubuntu-VirtualBox:~$ sudo mv hadoop-2.7.2 /usr/local/hadoop
```

// Assigning read and write access to Hadoop folder

```
$ sudo chown -R hduser:hadoop /usr/local/hadoop
```



```
hduser@ubuntu-VirtualBox:/usr/local/hadoop/hadoop-2.7.2$ sudo chown hduser:hadoop -R /usr/local/hadoop
hduser@ubuntu-VirtualBox:/usr/local/hadoop/hadoop-2.7.2$
```

Step 7 - Modify Hadoop config files

//Hadoop Environmental variable setting – The following files will be modified

1. ~/.bashrc
2. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/hadoop-env.sh
3. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/core-site.xml
4. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/hdfs-site.xml
5. /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/yarn-site.xml
6. /usr/local/hadoop/hadoop-2.8.4/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-2.8.4 export
```

```
PATH=$PATH:$HADOOP_HOME/bin
```

```
export PATH=$PATH:$HADOOP_HOME/sbin
```

```
export HADOOP_MAPRED_HOME=$HADOOP_HOME export
```

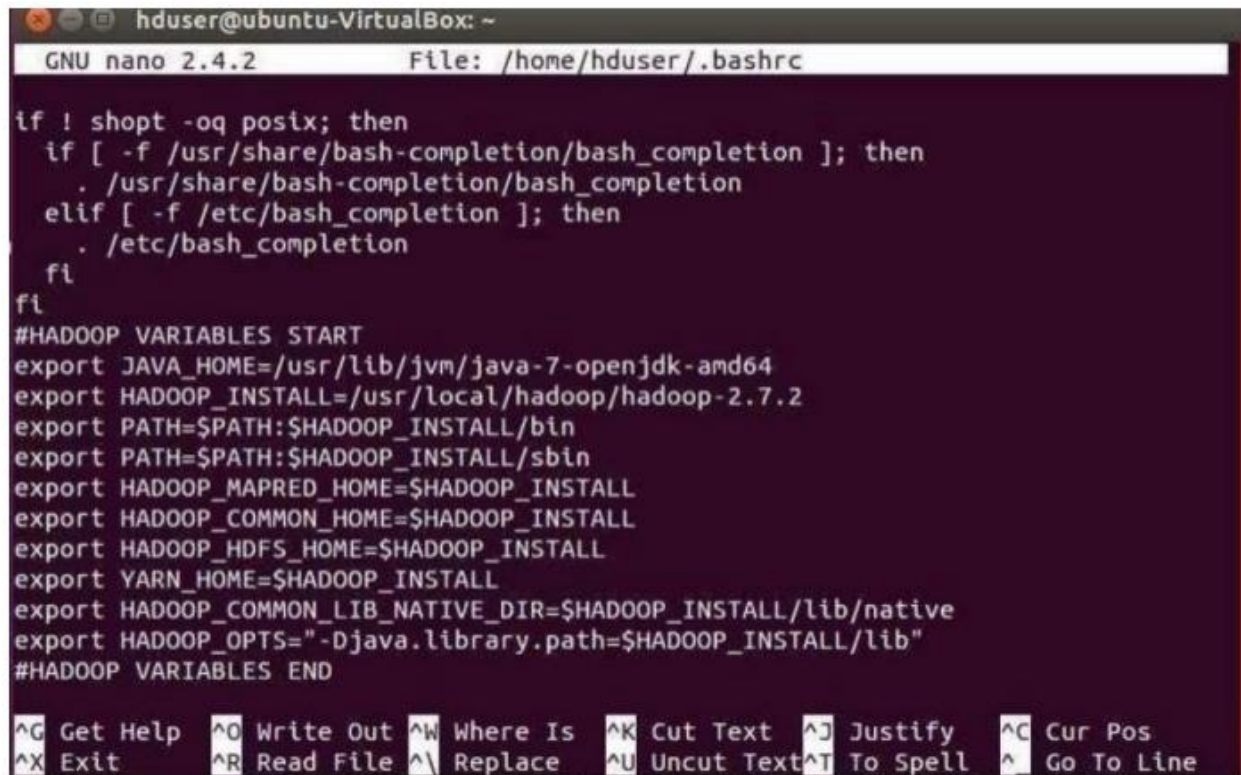
```
HADOOP_COMMON_HOME=$HADOOP_HOME export
```

```
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-2.8.4/bin
```

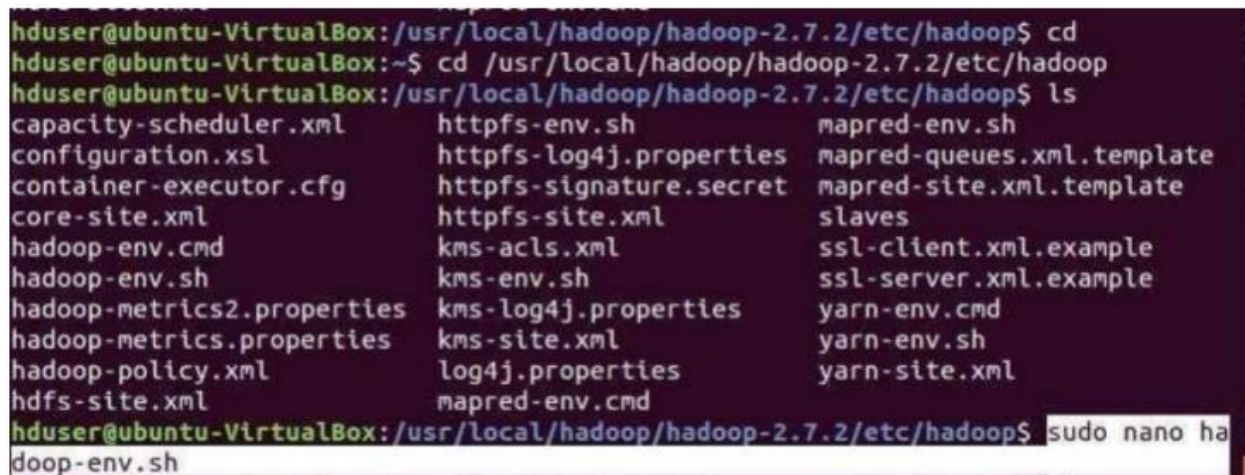


```
GNU nano 2.4.2 File: /home/hduser/.bashrc  
  
if ! shopt -oq posix; then  
  if [ -f /usr/share/bash-completion/bash_completion ]; then  
    . /usr/share/bash-completion/bash_completion  
  elif [ -f /etc/bash_completion ]; then  
    . /etc/bash_completion  
  fi  
fi  
  
#HADOOP VARIABLES START  
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64  
export HADOOP_INSTALL=/usr/local/hadoop/hadoop-2.7.2  
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  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

// Configure Hadoop Files

```
$ cd /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/
```

```
$ sudo nano hadoop-env.sh
```

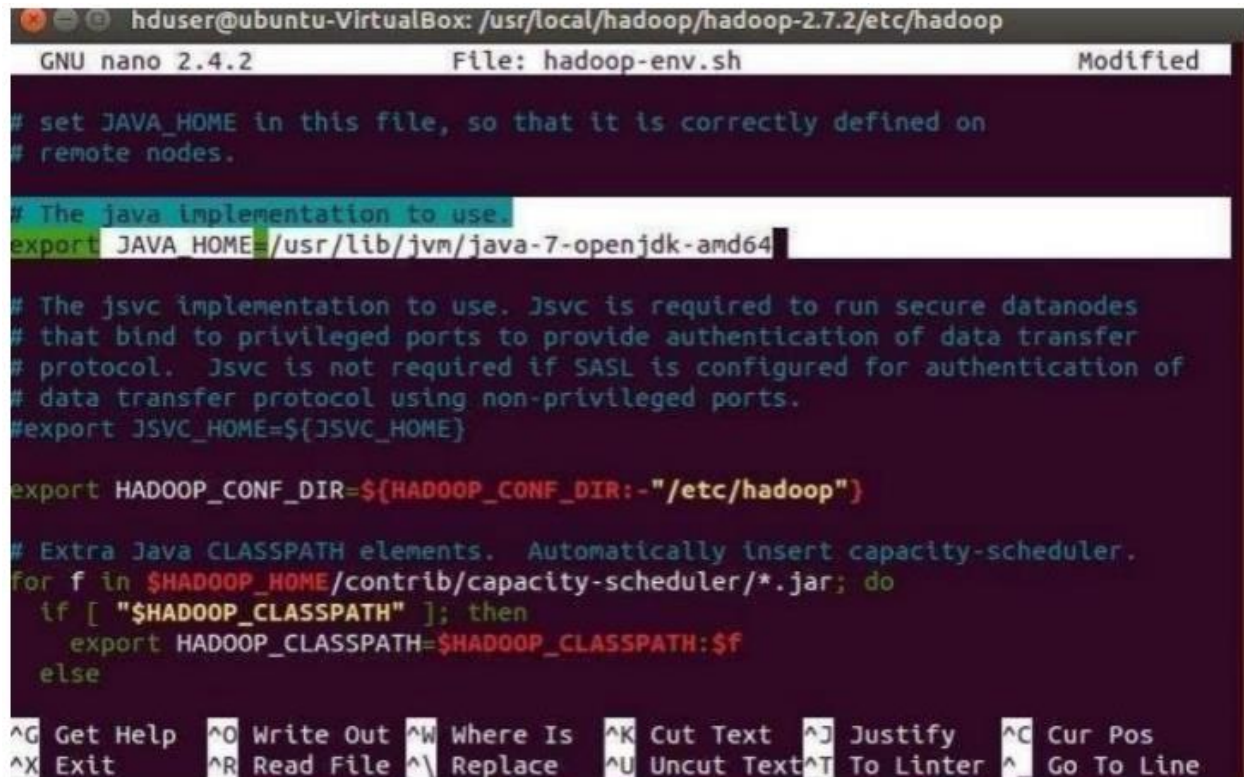


```
hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/etc/hadoop$ cd  
hduser@ubuntu-VirtualBox: ~$ cd /usr/local/hadoop/hadoop-2.7.2/etc/hadoop  
hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/etc/hadoop$ ls  
capacity-scheduler.xml      httpfs-env.sh               mapred-env.sh  
configuration.xml          httpfs-log4j.properties    mapred-queues.xml.template  
container-executor.cfg     httpfs-signature.secret    mapred-site.xml.template  
core-site.xml              httpfs-site.xml            slaves  
hadoop-env.cmd             kms-acls.xml               ssl-client.xml.example  
hadoop-env.sh              kms-env.sh                 ssl-server.xml.example  
hadoop-metrics2.properties kms-log4j.properties       yarn-env.cmd  
hadoop-metrics.properties kms-site.xml               yarn-env.sh  
hadoop-policy.xml          log4j.properties          yarn-site.xml  
hdfs-site.xml              mapred-env.cmd  
hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/etc/hadoop$ sudo nano ha  
doop-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



```
hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/etc/hadoop
GNU nano 2.4.2 File: hadoop-env.sh Modified

# 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

# 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.
#export JSVC_HOME=${JSVC_HOME}

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}

# Extra Java CLASSPATH elements. Automatically insert capacity-scheduler.
for f in $HADOOP_HOME/contrib/capacity-scheduler/*.jar; do
    if [ "$HADOOP_CLASSPATH" ]; then
        export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$f
    else

```

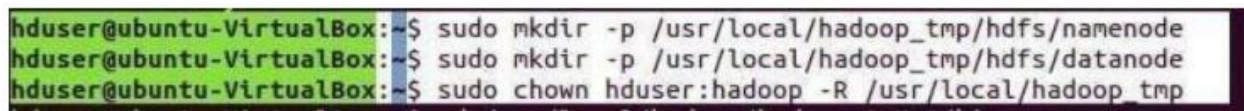
// Create datanode and namenode

\$ 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



```
hduser@ubuntu-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
hduser@ubuntu-VirtualBox:~$ sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
hduser@ubuntu-VirtualBox:~$ sudo chown hduser:hadoop -R /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>

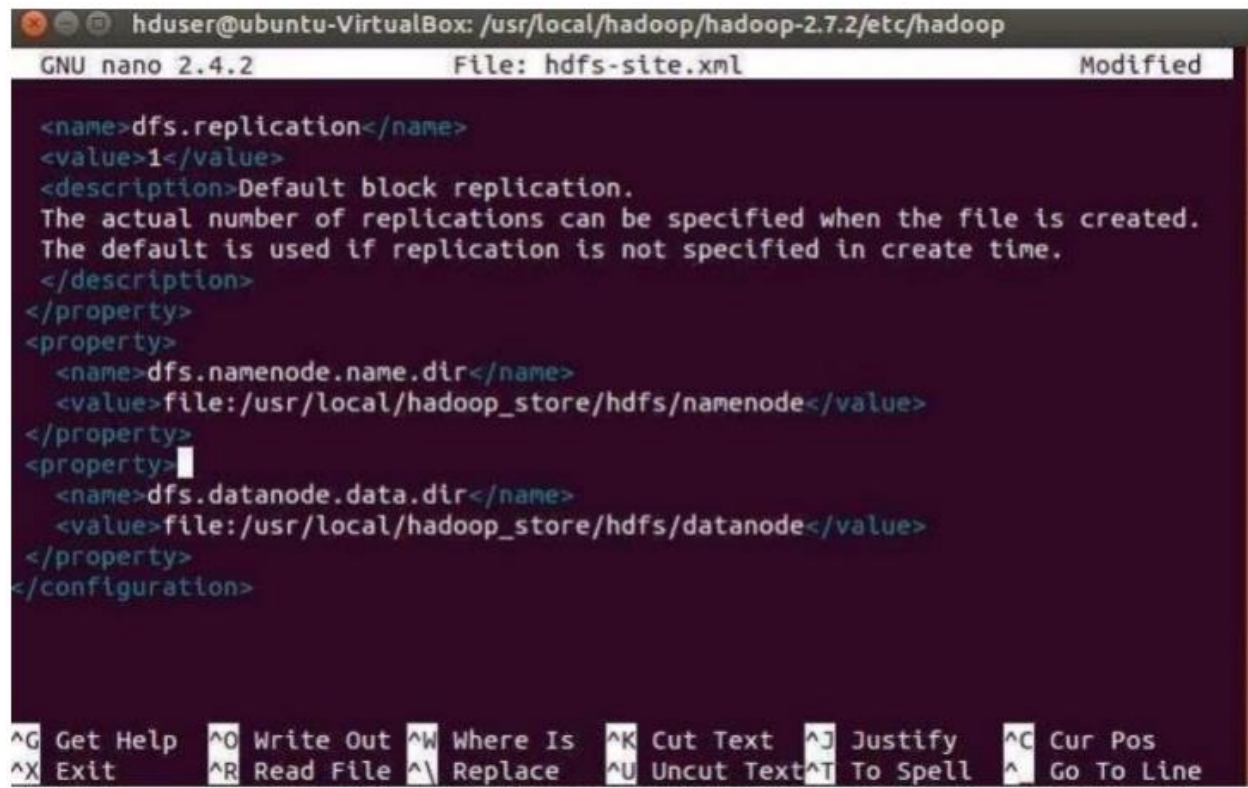
<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>

</property>

</configuration>
```



```
hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/etc/hadoop
GNU nano 2.4.2 File: hdfs-site.xml Modified

<name>dfs.replication</name>
<value>1</value>
<description>Default block replication.
The actual number of replications can be specified when the file is created.
The default is used if replication is not specified in create time.
</description>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/usr/local/hadoop_store/hdfs/namenode</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
</property>
</configuration>
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```
// 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.xml

$ cp /usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml.template

/usr/local/hadoop/hadoop-2.8.4/etc/hadoop/mapred-site.xml
```

```
hduser@ubuntu-VirtualBox:~$ 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-2.8.4/bin

\$ hadoop namenode -format

```
hduser@ubuntu-VirtualBox:~$ cd /usr/local/hadoop
hduser@ubuntu-VirtualBox:/usr/local/hadoop$ hadoop namenode -format
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

16/07/15 22:50:27 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
```

Step 9 - Start Hadoop

\$ cd /usr/local/hadoop/hadoop-2.8.4/sbin

// Starting dfs services

\$ start-dfs.sh

```

hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/sbin$ start-dfs.sh
16/07/15 22:55:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/hadoop-2.7.2/logs/hadoop-hduser-namenode-ubuntu-VirtualBox.out
localhost: starting datanode, logging to /usr/local/hadoop/hadoop-2.7.2/logs/hadoop-hduser-datanode-ubuntu-VirtualBox.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:+j+WF1JP500vL5mgcc7v9A/rU8jVQEHE8WFLmt2aEo8.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/hadoop-2.7.2/logs/hadoop-hduser-secondarynamenode-ubuntu-VirtualBox.out

```

// Starting mapreduce services

\$ start-yarn.sh

```

hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/sbin$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/hadoop-2.7.2/logs/yarn-hduser-resourcemanager-ubuntu-VirtualBox.out
localhost: starting nodemanager, logging to /usr/local/hadoop/hadoop-2.7.2/logs/yarn-hduser-nodemanager-ubuntu-VirtualBox.out

```

\$ jps

```

hduser@ubuntu-VirtualBox: /usr/local/hadoop/hadoop-2.7.2/sbin$ jps
12425 SecondaryNameNode
12609 ResourceManager
12733 NodeManager
13131 Jps
12205 DataNode
12080 NameNode

```

Step 10 - Check Hadoop through web UI

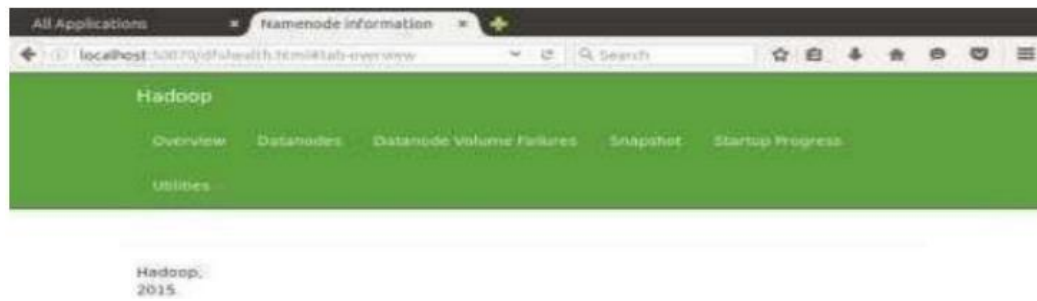
Go to browser type <http://localhost:8088> – All Applications Hadoop Cluster

The screenshot shows the Hadoop web UI at <http://localhost:8088/cluster>. The page features the Hadoop logo and a sidebar with navigation links: About, Nodes, Node Labels, Applications, NEW, SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, and Scheduler. The main content area is titled 'All Application' and contains two tables: 'Cluster Metrics' and 'Scheduler Metrics'.

Cluster Metrics									
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VC Total
0	0	0	0	0	0 B	8 GB	0 B	0	8

Scheduler Metrics							
Scheduler Type		Scheduling Resource Type		Minimum			
Capacity Scheduler		[MEMORY]		<memory:1024, vC			
ID	User	Name	Application Type	Queue	StartTime	FinishTime	State

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



Step 11 - Stop Hadoop

```
$ stop-dfs.sh
```

```
$ stop-yarn.sh
```

RESULT:

Thus the procedure to install single-node Hadoop is executed successfully.