Hadoop Installation:

Introduction:

Hadoop is a framework written in java for running applications on large clusters of hardware and incorporates features similar to those of the google file system and MapReduce computing paradigm. HDFS is a highly fault-tolerant distributed file system. HDFS is suitable for applications that have large data sets.

Hadoop installation can be done in three different modes (Standalone mode, Single Node Cluster mode and multi Node cluster mode) we have installed a single-node so that we can quickly perform simple operations using Hadoop MapReduce and the Hadoop Distributed File System(HDFS).

Prerequisites:

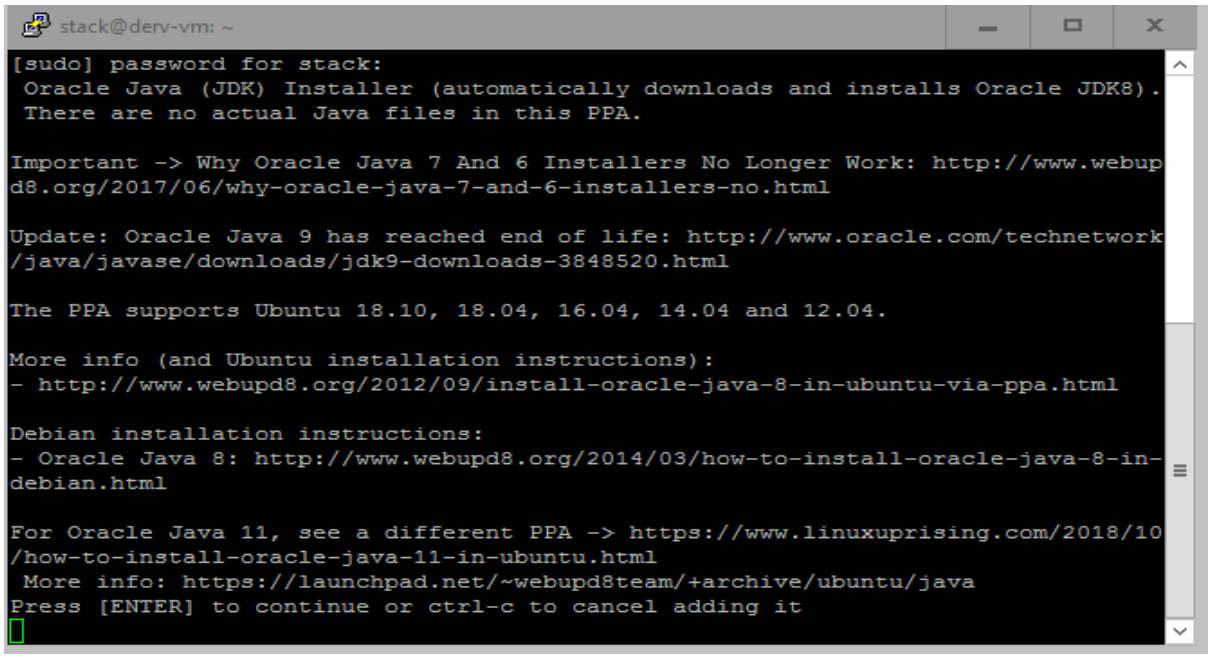
First we need to have Ubuntu server installed. I have installed Ubuntu server 16.04 with 2 processors running and set the ports to run the SSH.

I have connected Ubuntu server through SSH using putty and it is connected as the user stack which has sudo functions we need to load the java. So, java (JDK) installer because it automatically downloads and installs oracle JDK8. The following command is used to run the java using sudo.

Run Java8:

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

$ sudo apt-get update
And the following result is obtained



In the above screenshot we can see clearly that java (JDK) installer is automatically downloads and installs oracle JDKS one's the java is downloaded

it ask us the permission to continue or cancel the adding's after giving the permission we can see functions loaded which is shown in the below screenshot.

```
stack@derv-vm: ~
gpg: keyring `/tmp/tmpog6a7w7x/secring.gpg' created
gpg: keyring `/tmp/tmpog6a7w7x/pubring.gpg' created
gpg: requesting key EEA14886 from hkp server keyserver.ubuntu.com
gpg: /tmp/tmpog6a7w7x/trustdb.gpg: trustdb created
gpg: key EEA14886: public key "Launchpad VLC" imported
gpg: no ultimately trusted keys found
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
OK
stack@derv-vm:~$ sudo apt-get update
Get:1 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease [17.5 kB
]
Get:2 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main amd64 Package
s [1,556 B]
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Get:5 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main i386 Packages
 [1,556 B]
Get:6 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main Translation-e
n [928 B]
Get:7 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Fetched 347 kB in 0s (481 kB/s)
```

Once the update is done we ned to give the following command to install the oracle java8 installer.

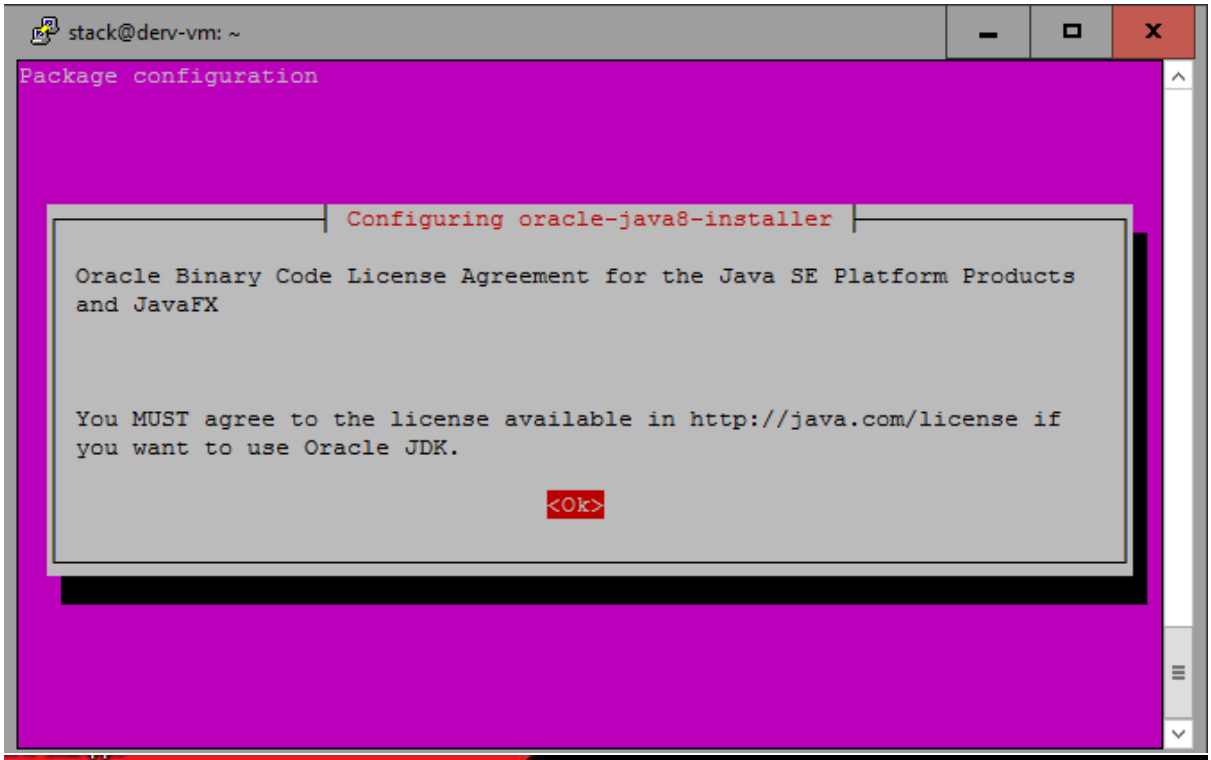
$ sudo apt-get install oracle-java8-installer

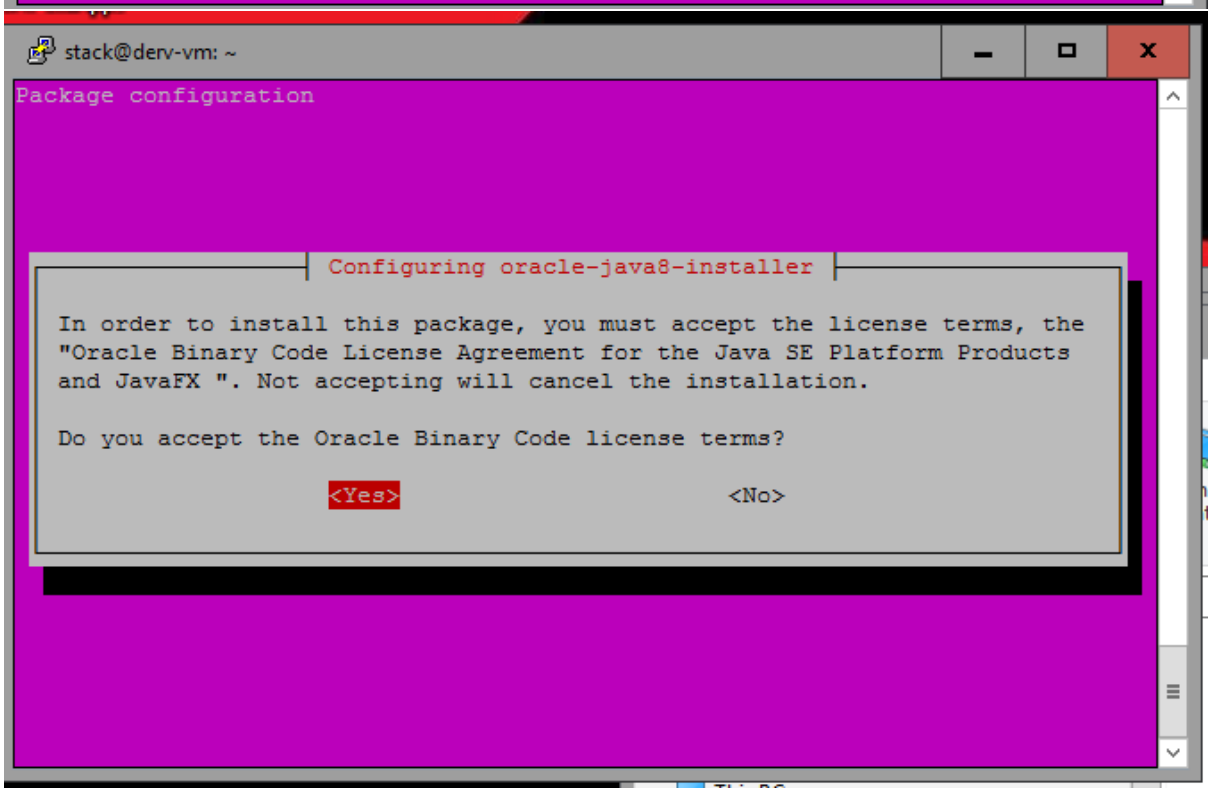
After running the above command it will read the packages and build the dependency tree and read the state information in the screenshot we can see that 11 newly installed, 25 not upgraded.

```
stack@derv-vm: ~
Try: sudo apt install <selected package>
stack@derv-vm:~$ sudo apt-get install oracle-java8-installer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils gsfonts gsfonts-x11 java-common libfontenc1 libxfont1
  oracle-java8-set-default x11-common xfonts-encodings xfonts-utils
Suggested packages:
  binutils-doc binfmt-support visualvm ttf-baekmuk | ttf-unfonts
  | ttf-unfonts-core ttf-kochi-gothic | ttf-sazanami-gothic ttf-kochi-mincho
  | ttf-sazanami-mincho ttf-arphic-uming firefox | firefox-2 | iceweasel
  | mozilla-firefox | iceape-browser | mozilla-browser | epiphany-gecko
  | epiphany-webkit | epiphany-browser | galeon | midbrowser
  | moblin-web-browser | xulrunner | xulrunner-1.9 | konqueror
  | chromium-browser | midori | google-chrome
The following NEW packages will be installed:
  binutils gsfonts gsfonts-x11 java-common libfontenc1 libxfont1
  oracle-java8-installer oracle-java8-set-default x11-common xfonts-encodings
  xfonts-utils
0 upgraded, 11 newly installed, 0 to remove and 25 not upgraded.
Need to get 6,520 kB of archives.
After this operation, 20.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

This PC

After that following pop-up screens we need to give the access just to agree the license.





Then we need to see JAVA_HOME has been set correctly using the following command we see the java_home.

$ java –version

Then it will displays the version and runtime environment and the java hotspot.

```
stack@derv-vm: ~
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for systemd (229-4ubuntu21.16) ...
Processing triggers for ureadahead (0.100.0-19) ...
Setting up oracle-java8-set-default (8u201-1~webupd8~1) ...
Setting up gsfonts (1:8.11+urwcyr1.0.7~pre44-4.2ubuntu1) ...
Setting up libfontenc1:amd64 (1:1.1.3-1) ...
Setting up libxfont1:amd64 (1:1.5.1-1ubuntu0.16.04.4) ...
Setting up x11-common (1:7.7+13ubuntu3.1) ...
update-rc.d: warning: start and stop actions are no longer supported; falling ba
ck to defaults
Setting up xfonts-encodings (1:1.0.4-2) ...
Setting up xfonts-utils (1:7.7+3ubuntu0.16.04.2) ...
Setting up gsfonts-x11 (0.24) ...
Processing triggers for libc-bin (2.23-0ubuntu11) ...
Processing triggers for systemd (229-4ubuntu21.16) ...
Processing triggers for ureadahead (0.100.0-19) ...
stack@derv-vm:~$ java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
stack@derv-vm:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
stack@derv-vm:~$
```

Adding a dedicated Hadoop system user:

Now, we need to create new user and new group were the Hadoop files need to be stored and run the Hadoop.

$ sudo addgroup Hadoop

Above command used to add the group named Hadoop and the below command is used to name the user name of the file.

$ sudo adduser --ingroup hadoop yaswanthuser

```
stack@derv-vm: ~
Setting up xfonts-encodings (1:1.0.4-2) ...
Setting up xfonts-utils (1:7.7+3ubuntu0.16.04.2) ...
Setting up gsfonts-x11 (0.24) ...
Processing triggers for libc-bin (2.23-0ubuntu11) ...
Processing triggers for systemd (229-4ubuntu21.16) ...
Processing triggers for ureadahead (0.100.0-19) ...
stack@derv-vm:~$ java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
stack@derv-vm:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
stack@derv-vm:~$ sudo adduser --ingroup hadoop yaswanthuser
Adding user `yaswanthuser' ...
Adding new user `yaswanthuser' (1001) with group `hadoop' ...
Creating home directory `/home/yaswanthuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yaswanthuser
Enter the new value, or press ENTER for the default
        Full Name []:
```

In the above screenshot we can see that the group Hadoop and user Yaswanthuser is created and the unix password is set.

```
stack@derv-vm: ~
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
stack@derv-vm:~$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
stack@derv-vm:~$ sudo adduser --ingroup hadoop yaswanthuser
Adding user `yaswanthuser' ...
Adding new user `yaswanthuser' (1001) with group `hadoop' ...
Creating home directory `/home/yaswanthuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yaswanthuser
Enter the new value, or press ENTER for the default
        Full Name []: yaswanth
        Room Number []: 4
        Work Phone []: 9491
        Home Phone []: 044
        Other []:
Is the information correct? [Y/n] y
stack@derv-vm:~$ groups yaswanthuser
yaswanthuser : hadoop
stack@derv-vm:~$
```

Then the sudo privileges is given to the yaswanthuser using the following command

$ sudo adduser yaswanthuser sudo

```
stack@derv-vm: ~
Done.
stack@derv-vm:~$ sudo adduser --ingroup hadoop yaswanthuser
Adding user `yaswanthuser' ...
Adding new user `yaswanthuser' (1001) with group `hadoop' ...
Creating home directory `/home/yaswanthuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for yaswanthuser
Enter the new value, or press ENTER for the default
        Full Name []: yaswanth
        Room Number []: 4
        Work Phone []: 9491
        Home Phone []: 044
        Other []:
Is the information correct? [Y/n] y
stack@derv-vm:~$ groups yaswanthuser
yaswanthuser : hadoop
stack@derv-vm:~$ sudo adduser yaswanthuser sudo
Adding user `yaswanthuser' to group `sudo' ...
Adding user yaswanthuser to group sudo
Done.
stack@derv-vm:~$
```

We can see in the above screenshot yaswanthuser to group sudo.

Hadoop Installation:

Now we start the download the Hadoop and extract the required files to Hadoop

$ wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz

Using the above link a zip file is downloaded and by using the below command we will unzip the Hadoop installation files.

$ tar xvzf hadoop-2.6.5.tar.gz

X tells the tar to extract the files

V list all the files one by one in the archive, Z tells the tar to uncompress the file, F give the file name to work with

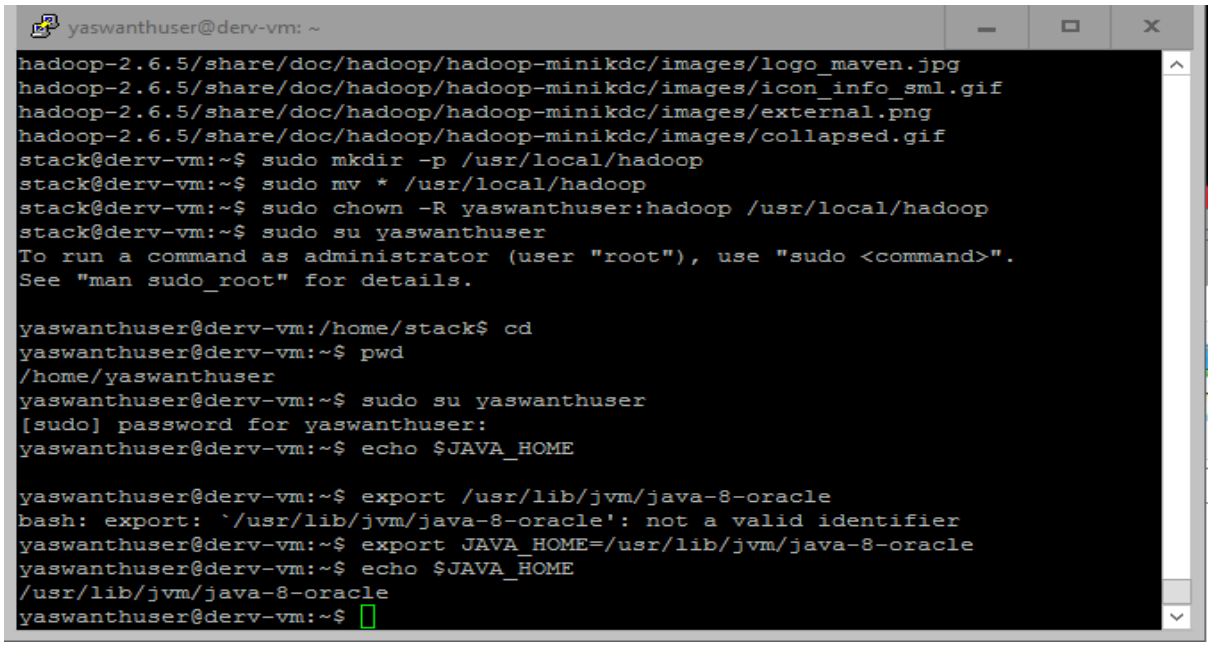
```
stack@derv-vm: ~
stack@derv-vm:~$ groups yaswanthuser
yaswanthuser : hadoop
stack@derv-vm:~$ sudo adduser yaswanthuser sudo
Adding user `yaswanthuser' to group `sudo' ...
Adding user yaswanthuser to group sudo
Done.
stack@derv-vm:~$ wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.6.5
/hadoop-2.6.5.tar.gz
--2019-04-01 12:46:08--  http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.
6.5/hadoop-2.6.5.tar.gz
Resolving mirrors.sonic.net (mirrors.sonic.net)... 157.131.0.16, 2001:5a8:601:2:
:bad:beef
Connecting to mirrors.sonic.net (mirrors.sonic.net)|157.131.0.16|:80... connecte
d.
HTTP request sent, awaiting response... 200 OK
Length: 199635269 (190M) [application/x-gzip]
Saving to: 'hadoop-2.6.5.tar.gz'

hadoop-2.6.5.tar.gz 100%[===================>] 190.39M  3.34MB/s    in 42s

2019-04-01 12:46:50 (4.59 MB/s) - 'hadoop-2.6.5.tar.gz' saved [199635269/1996352
69]

stack@derv-vm:~$
```

Now we need to move the installation files to newly created folder following command is used to perform the task.

$ sudo mv * /usr/local/Hadoop

$ sudo chown -R yaswanthuser:hadoop /usr/local/Hadoop

With the help of above command we can move all the Hadoop installation file to yaswanthuser



Now we will customise the environmental variables for that first we need to setup the .bashrc file to configure the home directory of java and Hadoop. .bashrc file is used to provide and set up variables functions and we can add any command in that file that can be typed in command prompt. Next step is important because all the installation should be done in the home/yaswanthuser

Next using echo we will find the JAVA_HOME file and copy the path.

$ echo $JAVA_HOME

After running the command I didn't get the path so I exported the path using command export and then run the same line.

Update .bashrc:

Then I opened the .bashrc nano editor and copy the following code in nano editor

#HADOOP VARIABLES START

export JAVA_HOME=/usr/lib/jvm/java-8-oracle # (this is the path we copied above)

export HADOOP_INSTALL=/usr/local/hadoop/hadoop-2.6.5

export PATH=$PATH:$HADOOP_INSTALL/bin

export PATH=$PATH:$HADOOP_INSTALL/sbin

export HADOOP_MAPRED_HOME=$HADOOP_INSTALL

export HADOOP_COMMON_HOME=$HADOOP_INSTALL

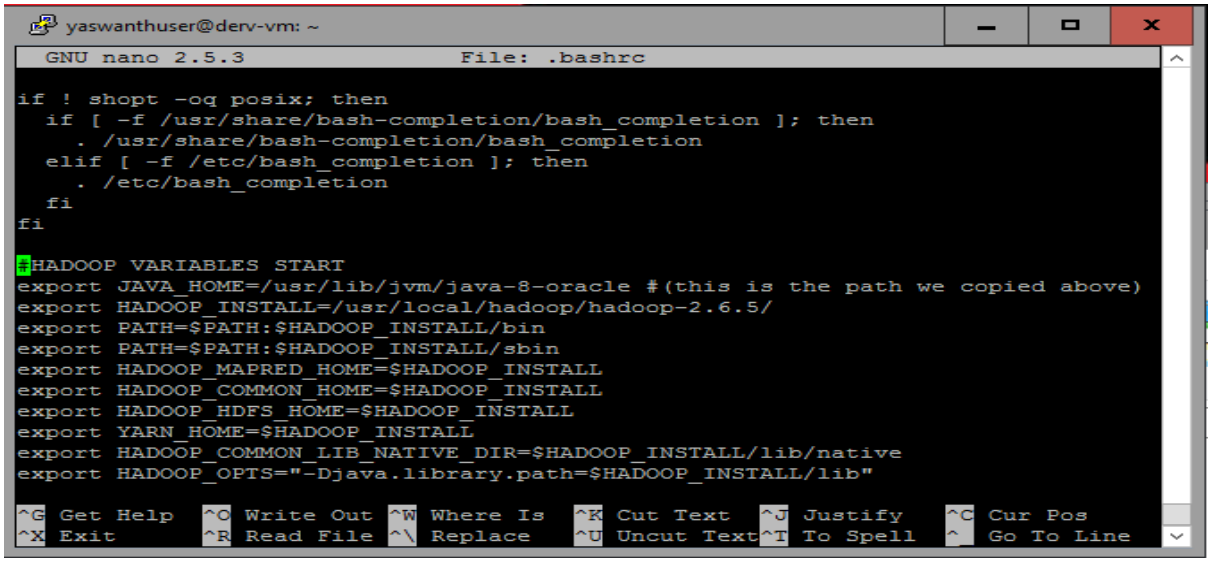
export HADOOP_HDFS_HOME=$HADOOP_INSTALL

export YARN_HOME=$HADOOP_INSTALL#(using YARN we can run the MapReduce)

export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native

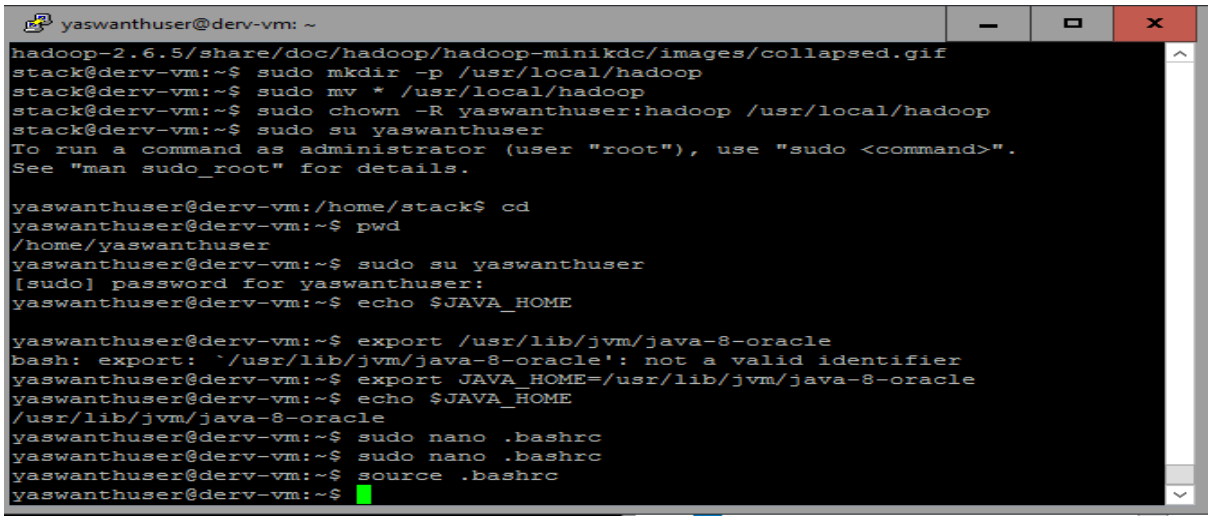
export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"

#HADOOP VARIABLES END



Whenever we edit the .bashrc file we need to refresh the .bashrc file using the following command

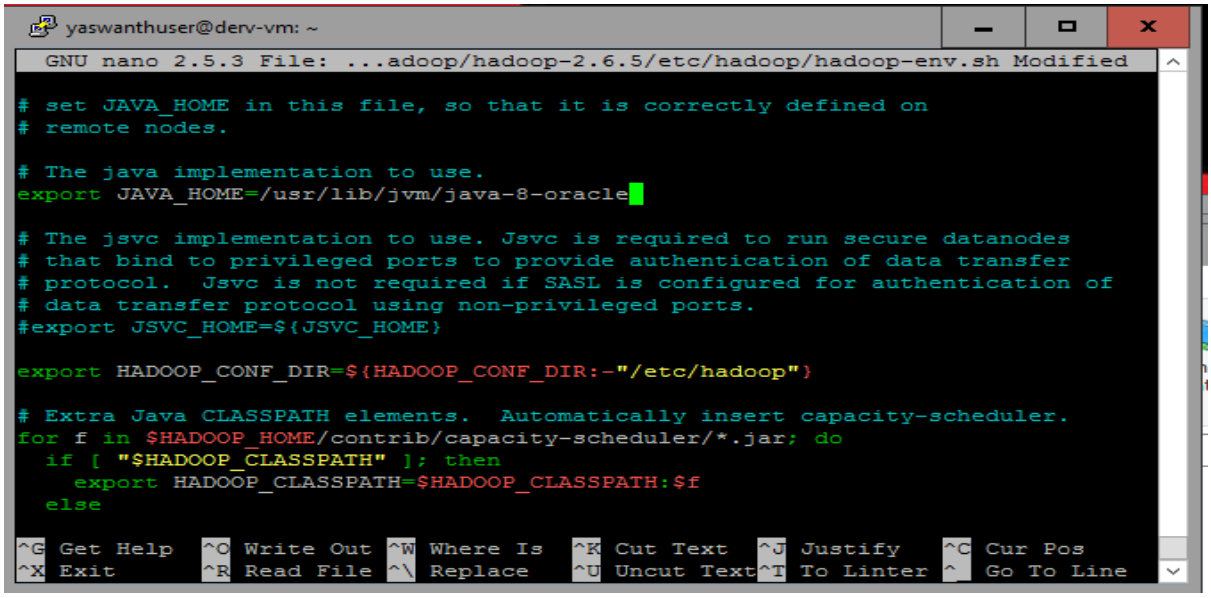
$ source .bashrc



Next we need to set the JAVA_HOME also in Hadoop_enviroment file and modify the java home modified java home file is showed in the below screenshot.

Configuration:

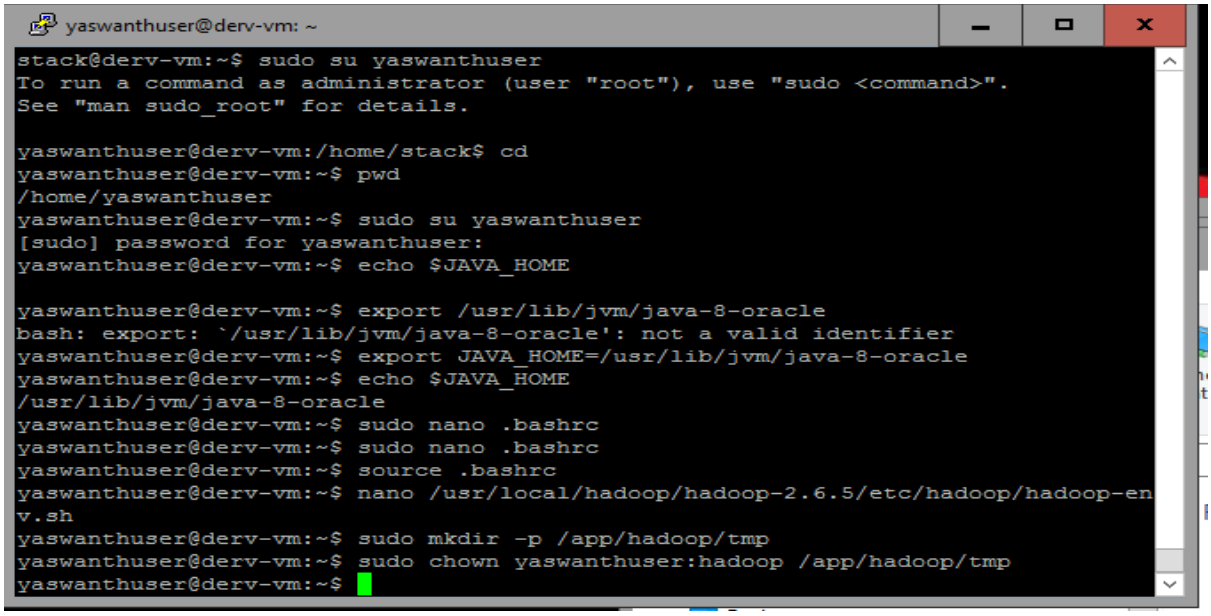
Daemon	Environment Variable
NameNode	HADOOP_NAMENODE_OPTS
DataNode	HADOOP_DATANODE_OPTS
Secondary NameNode	HADOOP_SECONDARYNAMENODE_OPTS
ResourceManager	YARN_RESOURCEMANAGER_OPTS
NodeManager	YARN_NODEMANAGER_OPTS
Map Reduce Job History Server	HADOOP_JOB_HISTORYSERVER_OPTS



Now, we create the temporary folder for Hadoop using mkdir(make directory)

$ sudo mkdir -p /app/hadoop/tmp

$ sudo chown yaswanthuser:hadoop /app/hadoop/tmp



Hadoop can also be run on a single-node in a pseudo distributed mode where each Hadoop daemon runs in a separate java process.

core-site.xml

$ nano /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/core-site.xml

```
yaswanthuser@derv-vm: ~
  GNU nano 2.5.3 File: ...adoop/hadoop-2.6.5/etc/hadoop/core-site.xml Modified

 <name>hadoop.tmp.dir</name>
 <value>/app/hadoop/tmp</value>
 <description>A base for other temporary directories.</description>
 </property>
 <property>
 <name>fs.default.name</name>
 <value>hdfs://localhost:54310</value>
 <description>The name of the default file system. A URI whose
 scheme and authority determine the FileSystem implementation. The
 uri's scheme determines the config property (fs.SCHEME.impl) naming
 the FileSystem implementation class. The uri's authority is used to
 determine the host, port, etc. for a filesystem.</description>
 </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
```

Now, we will configure the mapred-site.xml these to used to edit the file which drive the MapReduce process.

$ nano /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/mapred-site.xml

And configuration file is showed in the screenshot below.

```
yaswanthuser@derv-vm: ~
  GNU nano 2.5.3 File: ...oop/hadoop-2.6.5/etc/hadoop/mapred-site.xml Modified

  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. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
 <name>mapred.job.tracker</name>
 <value>localhost:54311</value>
 <description>The host and port that the MapReduce job tracker runs
 at. If "local", then jobs are run in-process as a single map
 and reduce task.
 </description>
 </property>
</configuration>

                              [ Cancelled ]
^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
```

Formatting the HDFS filesystem via the NameNode:

Then we will create the folders for namenode, datanode by making the directory using following commands.

$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode

$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode

```
yaswanthuser@derv-vm: ~

yaswanthuser@derv-vm:~$ export /usr/lib/jvm/java-8-oracle
bash: export: `/usr/lib/jvm/java-8-oracle': not a valid identifier
yaswanthuser@derv-vm:~$ export JAVA_HOME=/usr/lib/jvm/java-8-oracle
yaswanthuser@derv-vm:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-oracle
yaswanthuser@derv-vm:~$ sudo nano .bashrc
yaswanthuser@derv-vm:~$ sudo nano .bashrc
yaswanthuser@derv-vm:~$ source .bashrc
yaswanthuser@derv-vm:~$ nano /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/hadoop-en
v.sh
yaswanthuser@derv-vm:~$ sudo mkdir -p /app/hadoop/tmp
yaswanthuser@derv-vm:~$ sudo chown yaswanthuser:hadoop /app/hadoop/tmp
yaswanthuser@derv-vm:~$ nano /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/core-site
.xml
yaswanthuser@derv-vm:~$ cp /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/mapred-site
.xml.template /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/mapred-site.xml
yaswanthuser@derv-vm:~$ nano /usr/local/hadoop/hadoop-2.6.5/etc/hadoop/mapred-si
te.xml
yaswanthuser@derv-vm:~$ sudo mkdir -p /usr/local/hadoop_store/hdfs/namenode
yaswanthuser@derv-vm:~$ sudo mkdir -p /usr/local/hadoop_store/hdfs/datanode
yaswanthuser@derv-vm:~$ sudo chown -R yaswanthuser:hadoop /usr/local/hadoop_stor
e
yaswanthuser@derv-vm:~$
```

Now we need to configure the HDFS(Hadoop distributed file system) this file allows the specifications of the directories which can be used as Namenode and Datanode.

```
yaswanthuser@derv-vm: ~
  GNU nano 2.5.3 File: ...adoop/hadoop-2.6.5/etc/hadoop/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
```

Then we will format the new Hadoop file system by using the following command

$ hadoop namenode -format

```
yaswanthuser@derv-vm: ~
19/04/01 13:04:54 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 29
7.0 KB
19/04/01 13:04:54 INFO util.GSet: capacity      = 2^15 = 32768 entries
19/04/01 13:04:54 INFO namenode.NNConf: ACLs enabled? false
19/04/01 13:04:54 INFO namenode.NNConf: XAttrs enabled? true
19/04/01 13:04:54 INFO namenode.NNConf: Maximum size of an xattr: 16384
19/04/01 13:04:54 INFO namenode.FSImage: Allocated new BlockPoolId: BP-118907952
2-127.0.1.1-1554120294155
19/04/01 13:04:54 INFO common.Storage: Storage directory /usr/local/hadoop_store
/hdfs/namenode has been successfully formatted.
19/04/01 13:04:54 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/lo
cal/hadoop_store/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 using no
 compression
19/04/01 13:04:54 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/had
oop_store/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 of size 329 byt
es saved in 0 seconds.
19/04/01 13:04:54 INFO namenode.NNStorageRetentionManager: Going to retain 1 ima
ges with txid >= 0
19/04/01 13:04:54 INFO util.ExitUtil: Exiting with status 0
19/04/01 13:04:54 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at derv-vm.uni.mdx.ac.uk/127.0.1.1
************************************************************/
yaswanthuser@derv-vm:~$
```

In the above screenshot we can see that the Hadoop is installed successfully.

Now we start the Hadoop services we need to login as Hadoop user named yaswanthuser and open the Hadoop path and give the following command to start the Hadoop services.

Starting single-node cluster:

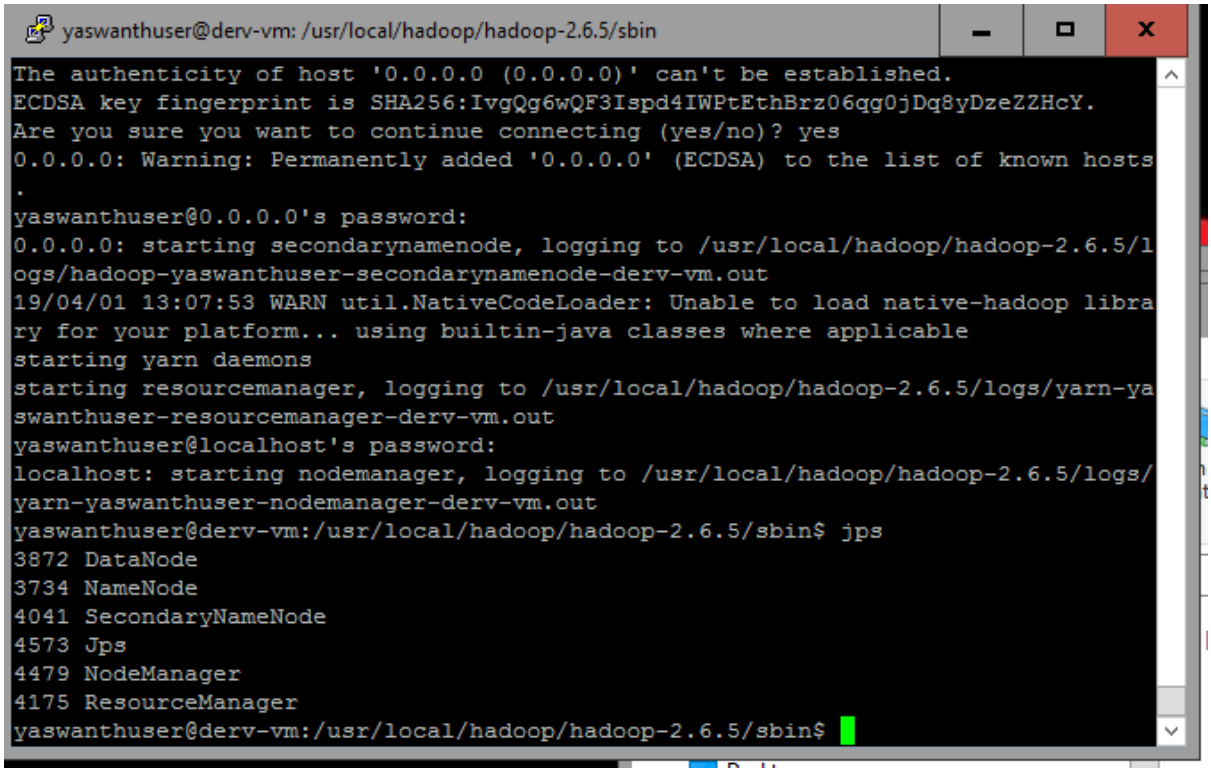
$ ./start-all.sh

```
yaswanthuser@derv-vm: /usr/local/hadoop/hadoop-2.6.5/sbin
/hdfs/namenode has been successfully formatted.
19/04/01 13:04:54 INFO namenode.FSImageFormatProtobuf: Saving image file /usr/lo
cal/hadoop_store/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 using no
 compression
19/04/01 13:04:54 INFO namenode.FSImageFormatProtobuf: Image file /usr/local/had
oop_store/hdfs/namenode/current/fsimage.ckpt_0000000000000000000 of size 329 byt
es saved in 0 seconds.
19/04/01 13:04:54 INFO namenode.NNStorageRetentionManager: Going to retain 1 ima
ges with txid >= 0
19/04/01 13:04:54 INFO util.ExitUtil: Exiting with status 0
19/04/01 13:04:54 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at derv-vm.uni.mdx.ac.uk/127.0.1.1
************************************************************/
yaswanthuser@derv-vm:~$ cd /usr/local/hadoop/hadoop-2.6.5/sbin
yaswanthuser@derv-vm:/usr/local/hadoop/hadoop-2.6.5/sbin$ sudo su yaswanthuser
yaswanthuser@derv-vm:/usr/local/hadoop/hadoop-2.6.5/sbin$ ./start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
19/04/01 13:06:56 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:IvgQg6wQF3Ispd4IWPtEthBrz06qg0jDq8yDzeZZHcY.
Are you sure you want to continue connecting (yes/no)? yes
```

After giving the start-all command we can see that namenode, datanode and secondarynamenode is started and we can check by giving the following command.

$ jps

Jps means (Java VM Process Status)



```
yaswanthuser@derv-vm: /usr/local/hadoop/hadoop-2.6.5/sbin
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:IvgQg6wQF3Ispd4IWPtEthBrz06qg0jDq8yDzeZZHcY.
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
.
yaswanthuser@0.0.0.0's password:
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/hadoop-2.6.5/l
ogs/hadoop-yaswanthuser-secondarynamenode-derv-vm.out
19/04/01 13:07:53 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/hadoop-2.6.5/logs/yarn-ya
swanthuser-resourcemanager-derv-vm.out
yaswanthuser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/hadoop-2.6.5/logs/
yarn-yaswanthuser-nodemanager-derv-vm.out
yaswanthuser@derv-vm:/usr/local/hadoop/hadoop-2.6.5/sbin$ jps
3872 DataNode
3734 NameNode
4041 SecondaryNameNode
4573 Jps
4479 NodeManager
4175 ResourceManager
yaswanthuser@derv-vm:/usr/local/hadoop/hadoop-2.6.5/sbin$
```

Stopping single-node cluster:

Conclusion:

The Single Node Cluster mode Hadoop is installed using the java oracle JDK and the new group and user is created to store or move the java files and the Hadoop installation file and HDFS and mapred and namenode xml files are configured and new Hadoop file system is formatted and finally Hadoop services are started with the single node cluster we can do Hadoop distributed file system and MapReduces mainly it provides high throughput access to application data and is suitable for applications that have large data sets.

Task 1: (MapReduce)

Introduction:

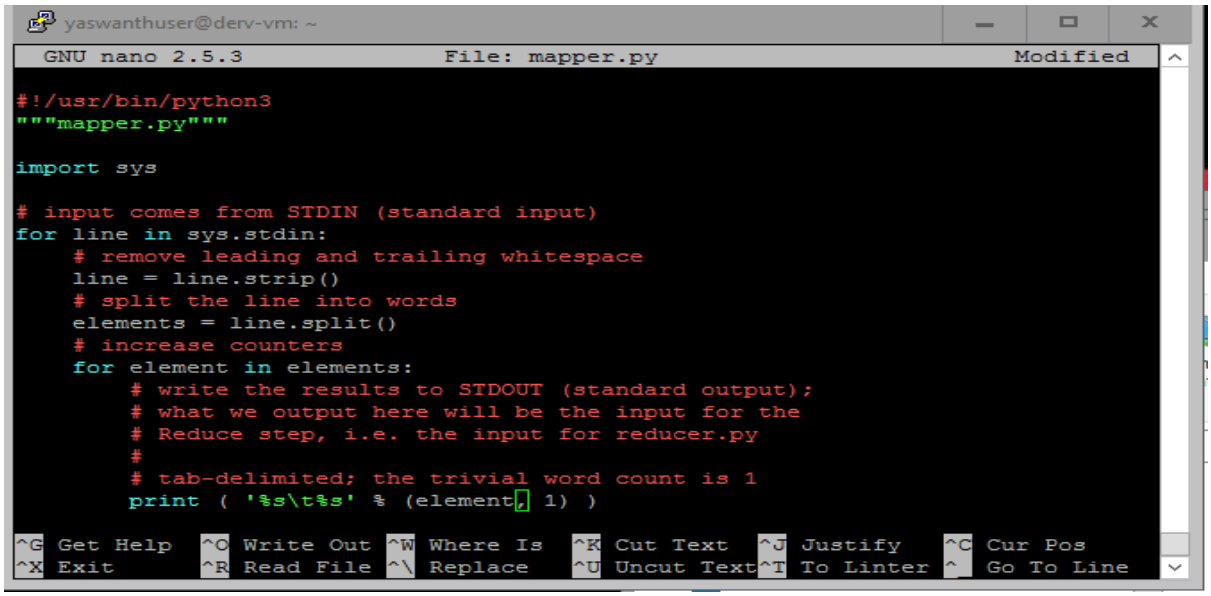
MapReduce is a frame work using which we can write applications to process huge amount of data on large clusters. MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce has two important tasks mapper and reducer.

Mapper:

The mapper job is to process the input data. The input data is in the form of file or directory stored in HDFS. The input file is passed to the mapper function line by line. The mapper processes the data and create several small chunks of data.

Run the MapReduce job:

Now, we do a MapReduce job that counts the total number of words in a document. First we write a mapper program in nano editor and save file name as mapper.py the program for mapper is shown in the screenshot. In the mapper.py program for loop is used to count the number of words.



Reducer:

The reducer is the combination of the shuffle stage and reduce stage the reducer job is to process the data that comes from the mapper. After the processing the new set of output will be stored in the HDFS.

The below screenshot is the reducer.py program reducer.py program takes the data from mapper.py and process the data from mapper.py.

```
yaswanthuser@derv-vm: ~
  GNU nano 2.5.3                File: reducer.py                    Modified

#!/usr/bin/python3
"""reducer.py"""

import sys

counting = 0

# input comes from STDIN
for elements in sys.stdin:
        # remove leading and trailing whitespace
        element, count = elements.split()
        count = int(count)
        counting += count

print( '%s' %(counting))

^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 Linter  ^_ Go To Line
```

Now first we check the MapReduce job in the Ubuntu server and then run in the Hadoop the following command is used to test MapReduce program in Ubuntu server.

$ echo "Yaswanth is very good" | python3 ./mapper.py | sort | python3 ./reducer.py

In the screenshot we can see the output 4 because the word count is 4.

```
yaswanthuser@derv-vm: ~
19/04/01 13:07:53 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/hadoop-2.6.5/logs/yarn-ya
swanthuser-resourcemanager-derv-vm.out
yaswanthuser@localhost's password:
localhost: starting nodemanager, logging to /usr/local/hadoop/hadoop-2.6.5/logs/
yarn-yaswanthuser-nodemanager-derv-vm.out
yaswanthuser@derv-vm:/usr/local/hadoop/hadoop-2.6.5/sbin$ jps
3872 DataNode
3734 NameNode
4041 SecondaryNameNode
4573 Jps
4479 NodeManager
4175 ResourceManager
yaswanthuser@derv-vm:/usr/local/hadoop/hadoop-2.6.5/sbin$ cd
yaswanthuser@derv-vm:~$ pwd
/home/yaswanthuser
yaswanthuser@derv-vm:~$ sudo nano mapper.py
yaswanthuser@derv-vm:~$ nano reducer.py
yaswanthuser@derv-vm:~$ echo "yaswanth is very good" | python3 ./mapper.py | sor
t | python3 ./reducer.py
4
yaswanthuser@derv-vm:~$
```

Before running the Mapreduce job in Hadoop we need to create a user directory in HDFS using the following command.

$ hadoop fs -mkdir /user

$ hadoop fs -mkdir /user/yaswanthuser

And copy all the program and text file to HDFS from Ubuntu server to HDFS by using following command

$ hadoop fs -put yastext.txt mapper.py reducer.py /user/yaswanthuser

```
yaswanthuser@derv-vm: ~
yaswanthuser@derv-vm:~$ nano reducer.py
yaswanthuser@derv-vm:~$ echo "yaswanth is very good" | python3 ./mapper.py | sor
t | python3 ./reducer.py
4
yaswanthuser@derv-vm:~$ hadoop fs -mkdir /user
19/04/01 13:21:41 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
yaswanthuser@derv-vm:~$ hadoop fs -mkdir /user/yaswanthuser
19/04/01 13:22:14 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
yaswanthuser@derv-vm:~$ nano yastext.txt
yaswanthuser@derv-vm:~$ hadoop fs -ls -R /
19/04/01 13:23:46 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
drwxr-xr-x   - yaswanthuser supergroup          0 2019-04-01 13:22 /user
drwxr-xr-x   - yaswanthuser supergroup          0 2019-04-01 13:22 /user/yaswant
huser
yaswanthuser@derv-vm:~$ find /usr/ -name 'hadoop-streaming*.jar'
/usr/local/hadoop/hadoop-2.6.5/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar
/usr/local/hadoop/hadoop-2.6.5/share/hadoop/tools/sources/hadoop-streaming-2.6.5
-test-sources.jar
/usr/local/hadoop/hadoop-2.6.5/share/hadoop/tools/sources/hadoop-streaming-2.6.5
-sources.jar
yaswanthuser@derv-vm:~$
```

Now we run the MapReduce job in Hadoop using the following command

$ hadoop jar /usr/local/hadoop/hadoop-2.6.5/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar -file /home/yaswanthuser/mapper.py -mapper mapper.py -file /home/yaswanthuseruser/reducer.py -reducer reducer.py -input /user/yaswanthuser/yastext.txt -output /user/yaswanthuser/Word2

```
yaswanthuser@derv-vm: ~
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=465313792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=960
        File Output Format Counters
                Bytes Written=5
19/04/01 13:39:17 INFO streaming.StreamJob: Output directory: /user/yaswanthuser
yaswanthuser@derv-vm:~$ hadoop fs -cat /user/hduser/Word2/part-00000
19/04/01 13:40:04 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
cat: `/user/hduser/Word2/part-00000': No such file or directory
yaswanthuser@derv-vm:~$ hadoop fs -cat /user/yaswanthuser/Word2/part-00000
19/04/01 13:40:18 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
129
yaswanthuser@derv-vm:~$
```

To read the output file the following command is used

$ hadoop fs -cat /user/yaswanthuser/Word2/part-00000

In the above screenshot we can see that output of MapReduce there are 129 words in yastext.txt file.

Conclusion:

In the task 1 MapReduce job is performed to count the number of words in a text document using python code and checked code first in Ubuntu server and then run the code in Hadoop before running code in Hadoop we need to create a user directory in HDFS and copy all the text file and programming file to HDFS user I wrote a mapper.py in nano editor by using the for loop and data from mapper is passed to reducer and output is stored in HDFS.

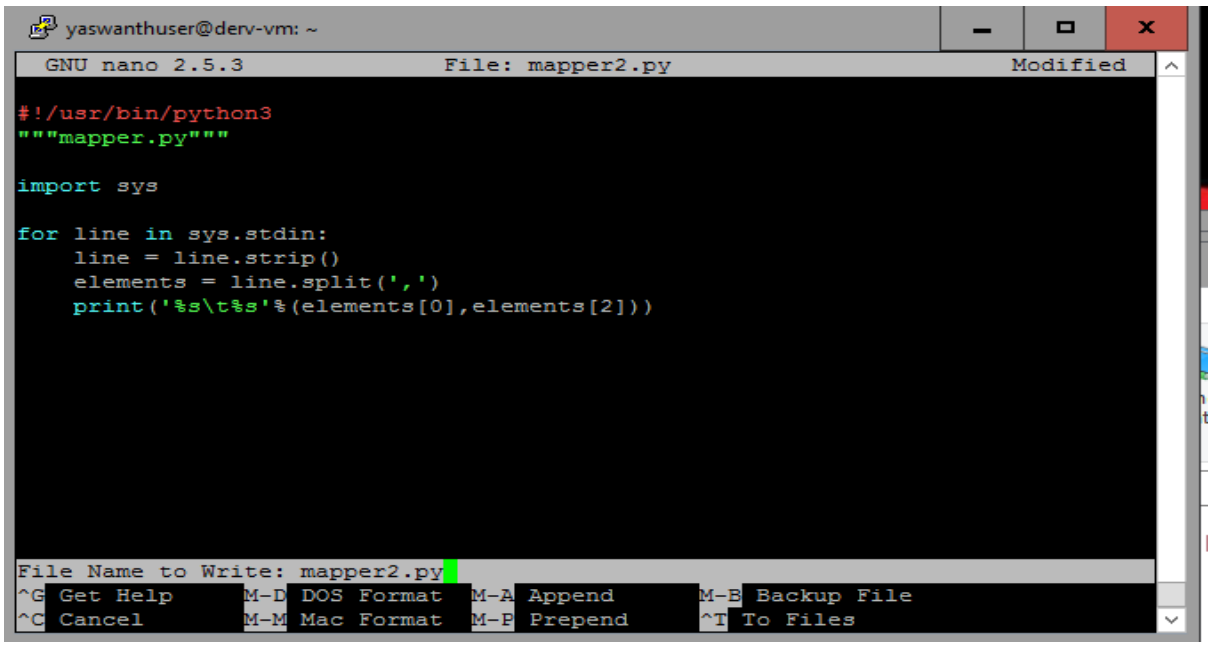
Task 2:

Introduction:

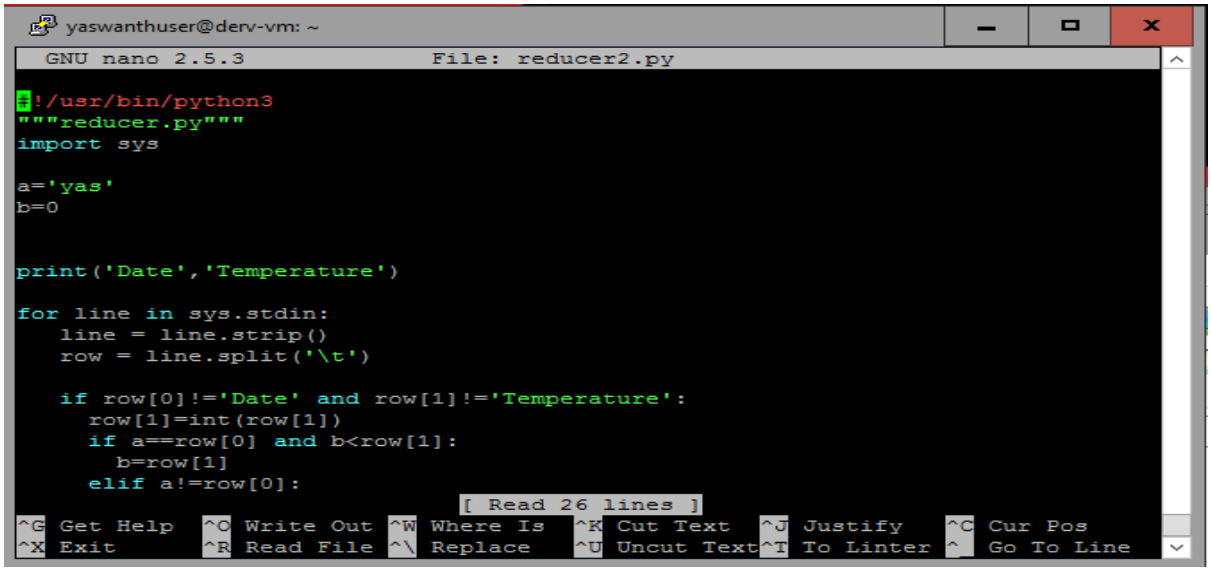
Hadoop MapReduce job is to be performed by reading .csv table having the structure ‘Date, Hour, Temperature’ and return the maximum temperature grouped by date. I saved the table name as file.csv and mapper as mapper2.py and reducer as reducer2.py

Mapper.py:

The below screenshot shows the python program for mapper and I used for loop in the mapper program.



In the reducer.py I used the if else statement to get the maximum temperature. Mapper send the data and reducer process the data and store in hdfs. The reducer.py program is mentioned in the following screenshot.



```
yaswanthuser@derv-vm: ~
  GNU nano 2.5.3              File: reducer2.py

for line in sys.stdin:
   line = line.strip()
   row = line.split('\t')

   if row[0]!='Date' and row[1]!='Temperature':
     row[1]=int(row[1])
     if a==row[0] and b<row[1]:
       b=row[1]
     elif a!=row[0]:
       if a!='yas':
         print(a,b)
       a=row[0]
       b=row[1]

print(a,b)

^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 Linter  ^_ Go To Line
```

Now we run the python code in the Ubuntu server by using following command and following output is obtained.

$ cat file.csv | python 3 ./mapper2.py | sort | python3 ./reducer2.py

And the following output is obatained.

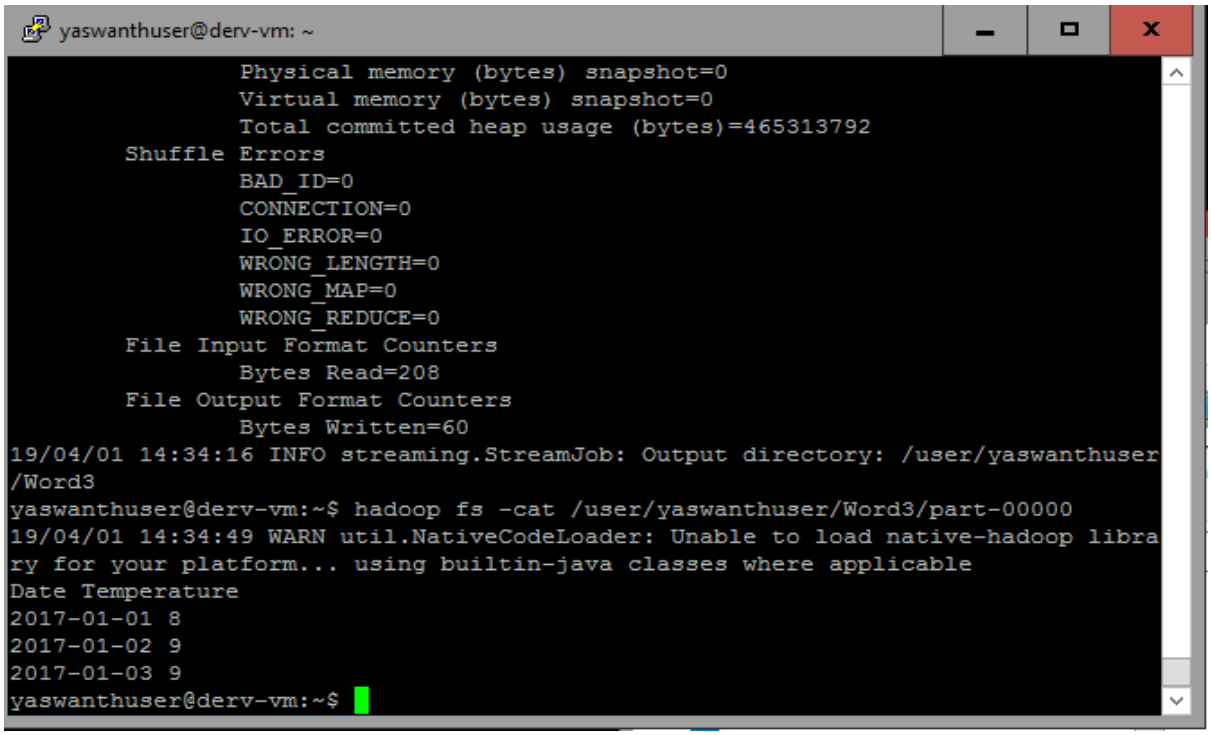
```
yaswanthuser@derv-vm: ~
yaswanthuser@derv-vm:~$ cat file.csv | python3 ./mapper2.py | sort | python3 ./r
educer2.py
Date Temperature
Traceback (most recent call last):
  File "./reducer2.py", line 21, in <module>
    print(d,t)
NameError: name 'd' is not defined
yaswanthuser@derv-vm:~$ nano reducer2.py
yaswanthuser@derv-vm:~$ nano reducer2.py
yaswanthuser@derv-vm:~$ cat file.csv | python3 ./mapper2.py | sort | python3 ./r
educer2.py
Date Temperature
2017-01-01 8
2017-01-02 9
2017-01-03 9
yaswanthuser@derv-vm:~$ nano reducer2.py
yaswanthuser@derv-vm:~$ nano reducer2.py
yaswanthuser@derv-vm:~$ cat file.csv | python3 ./mapper2.py | sort | python3 ./r
educer2.py
Date Temperature
2017-01-01 8
2017-01-02 9
2017-01-03 9
yaswanthuser@derv-vm:~$
```

Now we move the mapper and reducer programming file and table csv file to user HDFS before running the code in the Hadoop using the following command.

$ hadoop jar /usr/local/hadoop/hadoop-2.6.5/share/hadoop/tools/lib/hadoop-streaming-2.6.5.jar -file /home/yaswanthuser/mapper2.py -mapper mapper2.py -file /home/yaswanthuseruser/reducer2.py -reducer reducer2.py -input /user/yaswanthuser/file.csv -output /user/yaswanthuser/Word3

To see the output we need to give the following command.

$ hadoop fs -cat /user/yaswanthuser/Word3/part-00000



```
yaswanthuser@derv-vm: ~
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=465313792
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=208
        File Output Format Counters
                Bytes Written=60
19/04/01 14:34:16 INFO streaming.StreamJob: Output directory: /user/yaswanthuser
/Word3
yaswanthuser@derv-vm:~$ hadoop fs -cat /user/yaswanthuser/Word3/part-00000
19/04/01 14:34:49 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Date Temperature
2017-01-01 8
2017-01-02 9
2017-01-03 9
yaswanthuser@derv-vm:~$
```

Conclusion:

we have created the csv file and write the mapper and reducer program using python and check the mapper and reducer in the Ubuntu server if it works then move the file to Hadoop HDFS and run the mapper and reducer in the Hadoop and store the maximum temperature grouped by date.

Task 3:

In these task we need to create a two csv files and perform a MapReduce that receives both csv tables we write a mapper and reducer using python and run in ubuntu server and hadoop.

```
  GNU nano 2.5.3                                                    File: mapper3.py

#!/usr/bin/python3
"""mapper.py"""


import sys

for line in sys.stdin:
    line = line.strip()
    elements = line.split(',')

    sid=0
    name=0
    dob=0
    cid=0
    grade=0


    sid=elements[0]
    if len(elements[1])==7 and 'CSD' in elements[1]:
      cid=elements[1]

    else:
      name=elements[1]
    if '-' in elements[2]:
      dob=elements[2]
    else:
      grade=elements[2]

    print ('%s\t%s\t%s\t%s\t%s' % (sid,name,dob,cid,grade))

File Name to Write: mapper3.py
^G Get Help      M-D DOS Format     M-A Append         M-B Backup File
^C Cancel        M-M Mac Format     M-P Prepend        ^T To Files
```

In the above screenshot we can see the mapper program for the task done by python programing using if else statement. And in next screenshot we see the reducer program and the data for the reducer is processed from the mapper and reducer proces the data.

```
#!/usr/bin/python3
"""reducer"""
import sys

print ('StudentId\t Name\t CourseId\t Grade')

list1=[]
list2=[]

for line in sys.stdin:
     line = line.strip()
     row = line.split('\t')

     if row[0]!='StudentId' and row[2]>'1995-01-01':
       list1.append(row)
     if row[0]!='StudentId' and row[1]=='0':
       list2.append(row)

for row in list2:
    for row1 in list1:
       if row[0]==row1[0]:
         print('%s\t%s\t%s\t%s' % (row[0],row1[1],row[3],row[4]))
```

We will run the code in the Ubuntu server first and we will run the python program in Hadoop HDFS to run the MapReduce in Ubuntu server we will use following command.

$ cat file1.csv file2.csv | python 3 ./mapper3.py | sort | python3 ./reducer3.py

```
yaswanthuser@derv-vm: ~
/home/yaswanthuser
yaswanthuser@derv-vm:~$ ls
file1.csv  file.csv    mapper2.py  mapper.py    reducer3.py  yastext.txt
file2.csv  mapper1.py  mapper3.py  reducer2.py  reducer.py
yaswanthuser@derv-vm:~$ cat file1.csv file2. csv | ./mapper3.py | sort | ./reduc
er3.py
cat: file2.: No such file or directory
cat: csv: No such file or directory
StudentId        Name    CourseId        Grade
yaswanthuser@derv-vm:~$ cat file1.csv file2.csv | ./mapper3.py | sort | ./reduce
r3.py
StudentId        Name    CourseId        Grade
yaswanthuser@derv-vm:~$ nano file3.csv
yaswanthuser@derv-vm:~$ ls
file1.csv  file.csv    mapper2.py  mapper.py    reducer3.py  yastext.txt
file2.csv  mapper1.py  mapper3.py  reducer2.py  reducer.py
yaswanthuser@derv-vm:~$ nano file2.csv
yaswanthuser@derv-vm:~$ nano file1.csv
yaswanthuser@derv-vm:~$ cat file1.csv file2.csv | ./mapper3.py | sort | ./reduce
r3.py
StudentId        Name    CourseId        Grade
M912233 John    CSD8383 merit
M912233 John    CSD8481 distinction
yaswanthuser@derv-vm:~$
```

In the above screenshot we can see that MapReduce job is done in Ubuntu server now we run the program in Hadoop.

The following command is used to run command in Hadoop before that we need to move the files to Hadoop HDFS.

$ hadoop fs -cat /user/yaswanthuser/Word6/part-00000

```
yaswanthuser@derv-vm: ~
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=634466304
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=321
        File Output Format Counters
                Bytes Written=93
19/04/02 08:25:12 INFO streaming.StreamJob: Output directory: /user/yaswanthuser
/Word6
yaswanthuser@derv-vm:~$ hadoop fs -cat /user/yaswanthuser/Word6/part-00000
19/04/02 08:25:24 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
StudentId        Name    CourseId        Grade
M912233 John    CSD8481 distinction
M912233 John    CSD8383 merit
yaswanthuser@derv-vm:~$
```

Conclusion:

Two csv tables are created to perform MapReduce job using python and run the programs in both Ubuntu server and reducer and move all the csv and programming files to HDFS and run the MapReduce in Hadoop then the result is stored in HDFS.