

PRACTICAL JOURNAL IN

BIG DATA ANALYTICS

MODERN NETWORKING

EMBEDDED SYSTEMS

SUBMITTED BY

VRUSHALI PRAKASH ADAK

PRN NO.: 2020430041

**IN PARTIAL FULLFILMENT FOR THE DEGREE OF MASTER OF
SCIENCE IN INFORMATION TECHNOLOGY PART – I
SEMESTER II**

ACADEMIC YEAR

2023-2024

**B.N. BANDODKAR COLLEGE OF SCIENCE (AUTONOMOUS)
(AFFILIATED TO UNIVERSITY OF MUMBAI)**

THANE (W) - 400601, MAHARASHTRA

YEAR: 2023-2024

Vidya Prasarak Mandal's
**B. N. BANDODKAR COLLEGE OF SCIENCE
(AUTONOMOUS), THANE.**

(Affiliated to University of Mumbai)

NAAC REACCREDITED 'A' GRADE
Best College Award, University of Mumbai

माहिती व तंत्रज्ञान विभाग

दूरध्वनी क. २५३३ ६५०७



**Department of
Information Technology**

Tel. No. 2533 6507

Email : itbnb@vpmthane.org

CERTIFICATE

This is to certify that

Shri / Kum. _____

of M. Sc. (Information Technology) Part I Semester - II has completed the required number of experiments (Total =) signed herein, in this laboratory during the year 2023 – 2024.

Seal

Incharge
Department of Information
Technology

Principal
B. N. Bandodkar College of Science,
Thane

External Examiner

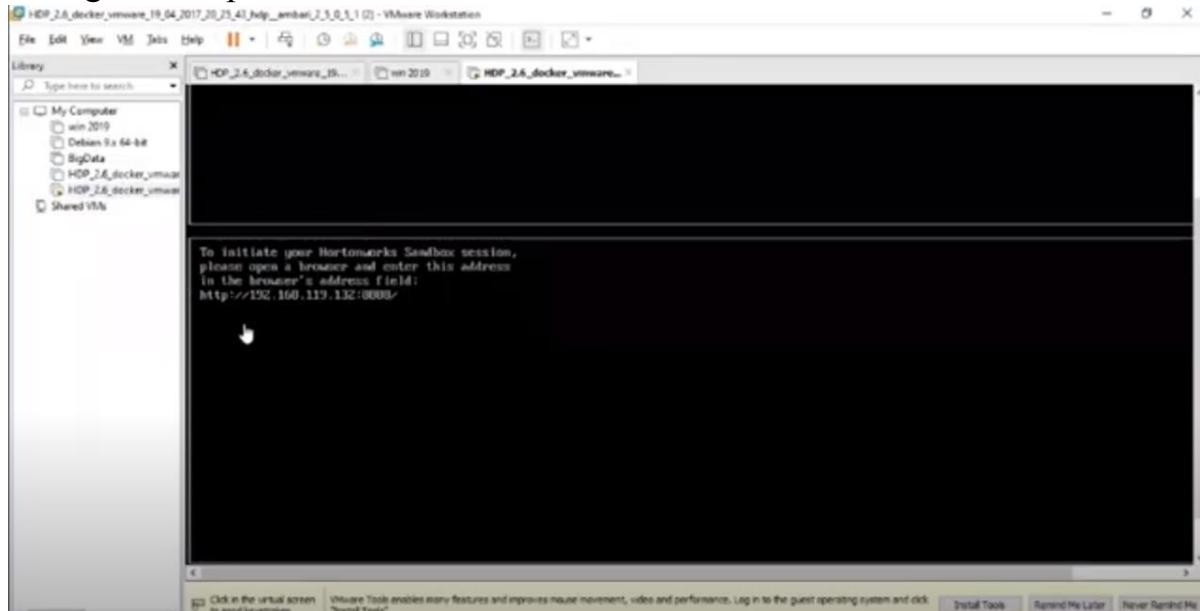
INDEX

Sr. No	Practical
1	Install, configure and run Hadoop and HDFS ad explore HDFS.
2	Implement word count / frequency programs using MapReduce
3	Implement an MapReduce program that processes a weather dataset.
4	Implement the program using Pig.
5	Implement the application in Hive.
6	Implement an application that stores big data in Hbase/ Python
7	Implement Decision tree classification techniques
8	Implement SVM classification techniques

Practical 1

Install, configure and run Hadoop and HDFS ad explore HDFS.

Download Virtual machine setup ie VMware setup (in which Hadoop is configured). Step 1: Load the server on VM ware workstation



Step 2: To enable admin login open shell and reset root

login. Open Terminal 192.168.119.132:4200

In Sandbox login enter root

And Password is Hadoop

And reset the password

```
root@sandbox.hortonworks.com's password:  
You are required to change your password immediately (root enforced)  
Last login: Wed Jun 30 14:50:19 2021 from 172.17  
.0.2  
Changing password for root.  
(current) UNIX password:  
New password:  
Retype new password:  
[root@sandbox ~]# █
```

Windows linux system and Hadoop system are different

When we type **ls** command it is executed in local system

```
(current) UNIX password:  
New password:  
Retype new password:  
[root@sandbox ~]# ls  
anaconda-ks.cfg  install.log.syslog  
blueprint.json    sandbox.info  
build.out        start_ambari.sh  
hdp              start_hbase.sh  
install.log :  
[root@sandbox ~]# █
```

When we type **hdfs dfs -ls** it will execute in Hadoop system directory

```
New password:  
Retype new password:  
[root@sandbox ~]# ls  
anaconda-ks.cfg  install.log.syslog  
blueprint.json   sandbox.info  
build.out        start_ambari.sh  
hdp             start_hbase.sh  
install.log  
[root@sandbox ~]# hdfs dfs -ls /  
[
```

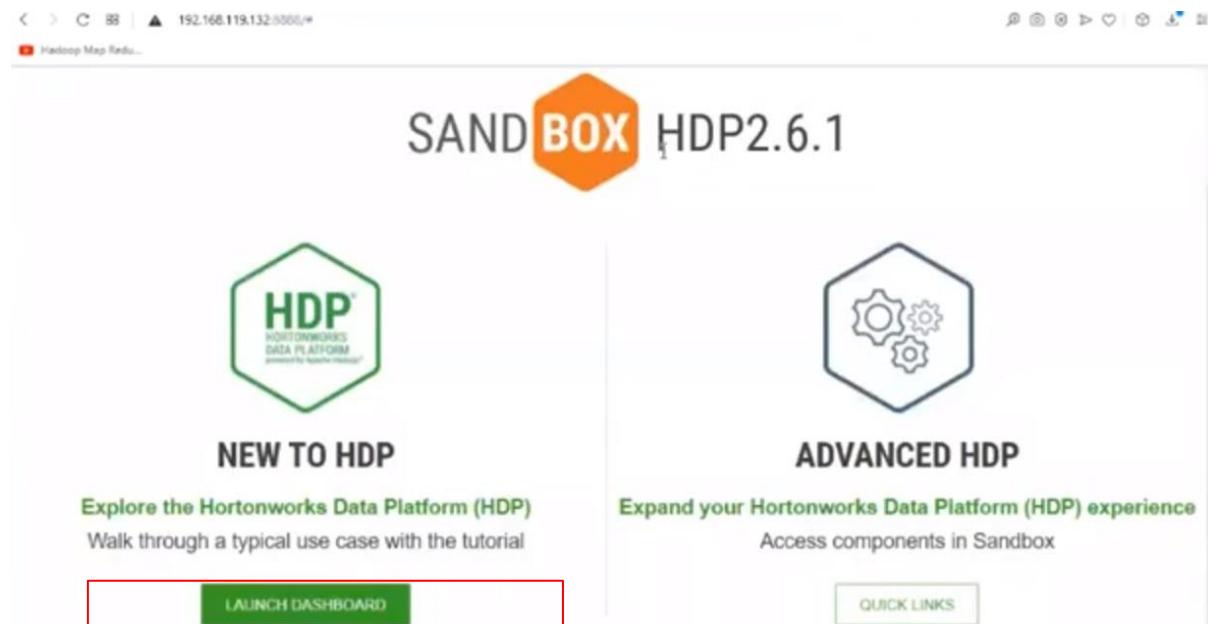
Step 3: Reset Admin account Password

```
[root@sandbox ~]# ambari-admin-password-reset  
Please set the password for admin:  
Please retype the password for admin:  
  
The admin password has been set.  
Restarting ambari-server to make the password change effective...  
  
Using python /usr/bin/python  
Restarting ambari-server  
Waiting for server stop...  
Ambari Server stopped  
Ambari Server running with administrator privileges.  
Organizing resource files at /var/lib/ambari-server/resources...  
Ambari database consistency check started...  
Server PID at: /var/run/ambari-server/ambari-server.pid  
Server out at: /var/log/ambari-server/ambari-server.out  
Server log at: /var/log/ambari-server/ambari-server.log  
Waiting for server start....[
```

Server listening on 8080 and shell login is complete.

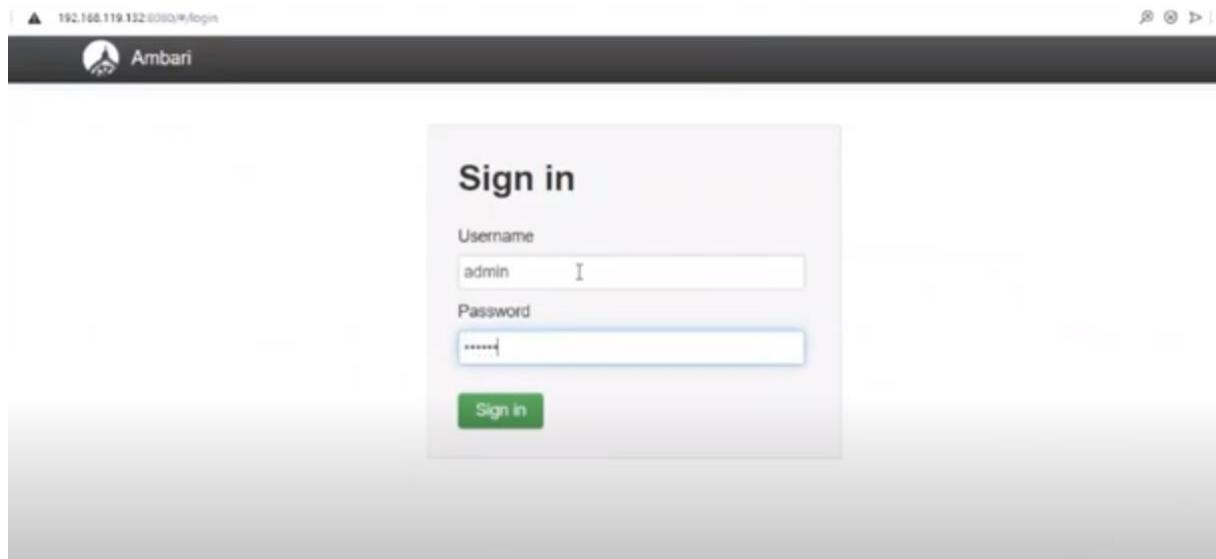
```
The admin password has been set.  
Restarting ambari-server to make the password change effective...  
  
Using python /usr/bin/python  
Restarting ambari-server  
Waiting for server stop...  
Ambari Server stopped  
Ambari Server running with administrator privileges.  
Organizing resource files at /var/lib/ambari-server/resources...  
Ambari database consistency check started...  
Server PID at: /var/run/ambari-server/ambari-server.pid  
Server out at: /var/log/ambari-server/ambari-server.out  
Server log at: /var/log/ambari-server/ambari-server.log  
Waiting for server start.....  
Server started listening on 8080  
  
DB configs consistency check: no errors and warnings were found.  
[root@sandbox ~]#
```

To use graphical user interface login to 192.168.119.132:4200



Click on Launch Dashboard Enter the username and password for admin login.

MSc IT Sem II



Below is the Hadoop server.

A screenshot of the Ambari Metrics Dashboard. The URL in the address bar is 192.168.119.132:8080/#/main/dashboard/metrics. The dashboard shows various metrics for HDFS, YARN, and other services. The left sidebar lists services: HDFS, YARN (selected), MapReduce2, Tez, Hive, HBase, Pig, Sqoop, Oozie, ZooKeeper, Falcon, Storm, Flume, Ambari Infra, and Ambari. The main area displays metrics like HDFS Disk Usage (n/a), DataNodes Live (n/a), HDFS Links (NameNode, Secondary NameNode, 1 DataNodes), Memory Usage (No Data Available), Network Usage (No Data Available), CPU Usage (No Data Available), Cluster Load (No Data Available), NameNode Heap (n/a), NameNode RPC (n/a), NameNode CPU WIO (n/a), and HBase Master Uptime (192.168.119.132:8080/#/main/services/VarHandle/summary).

To view file in HDFS click on HDFS and click on File view.

MSc IT Sem II

The screenshot shows the Ambari UI interface for managing a cluster. The top navigation bar includes tabs for Dashboard, Services, Hosts, Alerts, Admin, and a user icon. Below the navigation is a sidebar with various service icons. The 'HDFS' service is highlighted with a red box. To the right, there are two main sections: 'Summary' and 'Metrics'. The 'Summary' section contains status information for NameNodes, DataNodes, and NFSGateways, along with metrics like Disk Remaining, Total Files + Directories, Upgrade Status, and Safe Mode Status. A dropdown menu titled 'YARN Queue Manager' is open, with 'File View' also highlighted with a red box. Other options in the dropdown include Hive View, Hive View 2.0, Pig View, Storm View, Tez View, and Workflow Manager.

Commands:

- 1) To view root folder file from terminal use command hdfs dfs -ls /user and press enter.

It will display all the files in the root user that we see in UI(Screenshot 2).
ls: This command is used to list all the files

```
DB configs consistency check: no errors and warnings were found.
[root@sandbox ~]# hdfs dfs -ls /user
Found 13 items
drwxr-xr-x  - admin      hdfs          0 2017-04-19 19:09 /user/admin
drwxrwx---  - ambari-qa  hdfs          0 2017-04-19 18:48 /user/ambari-qa
drwxr-xr-x  - amy_ds    hdfs          0 2017-04-19 19:04 /user/amy_ds
drwxr-xr-x  - hbase     hdfs          0 2017-04-19 18:48 /user/hbase
drwxr-xr-x  - hcat      hdfs          0 2017-04-19 18:51 /user/hcat
drwxr-xr-x  - hive       hdfs          0 2017-04-19 19:08 /user/hive
drwxr-xr-x  - holger_gov hdfs          0 2017-04-19 19:05 /user/holger_gov
drwxrwxr-x  - livy      hdfs          0 2017-04-19 18:49 /user/livy
drwxr-xr-x  - maria_dev hdfs          0 2017-04-19 18:58 /user/maria_dev
drwxrwxr-x  - oozie     hdfs          0 2017-04-19 18:52 /user/oozie
drwxr-xr-x  - raj_ops   hdfs          0 2017-04-19 19:06 /user/raj_ops
drwxrwxr-x  - spark     hdfs          0 2017-04-19 18:49 /user/spark
drwxr-xr-x  - zeppelin  hdfs          0 2017-04-19 18:49 /user/zeppelin
[root@sandbox ~]#
```

The screenshot shows a file browser interface for HDFS. At the top, there are icons for refresh, search, and refresh. The path is shown as '/ > user'. A yellow box in the top right corner displays 'Total: 13 files or folders'. Below this is a table with the following columns: Name, Size, Last Modified, and Owner. The table lists three entries:

Name	Size	Last Modified	Owner
admin	--	2017-04-20 00:39	admin
ambari-qa	--	2017-04-20 00:18	ambari-qa
amy_ds	--	2017-04-20 00:34	amy_ds

2)

mkdir: To create a directory.

Create a folder in Hadoop directory. Type command hdfs dfs -mkdir /bigdatastest and enter. After it execute the command, we will see whether it is created folder in UI.

The terminal window shows the command being run:

```
[root@sandbox bigdata]# hdfs dfs -mkdir /bigdatastest
[root@sandbox bigdata]#
```

MSc IT Sem II

Name >	Size >	Last Modified >	Owner >
app-logs	--	2017-04-20 00:38	yarn
apps	--	2017-04-20 00:25	hdfs
ats	--	2017-04-20 00:18	yarn
bigdatatest	--	2021-06-30 20:31	root
demo	--	2017-04-20 00:33	hdfs
hdp	--	2017-04-20 00:18	hdfs

- 3) Create a file in local directory

Cat: Create a file.

Cat>>

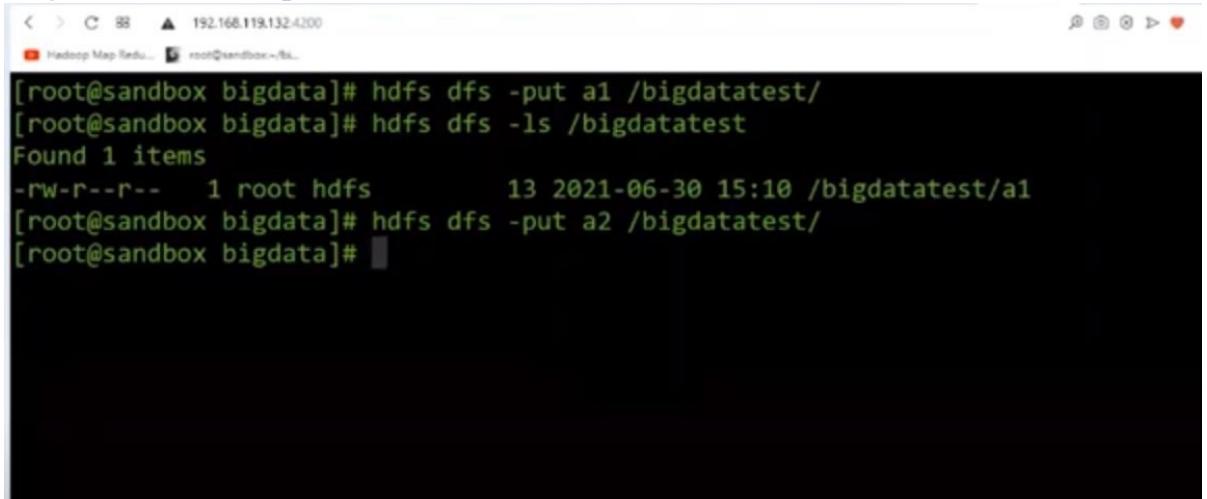
To terminate press ctrl+d

```
< > C ⌂ ▲ 192.168.119.132:4200
Hadoop Map Redu...
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-truncate [-w] <length> <path> ...]
[root@sandbox bigdata]# hdfs dfs -cat >>/bigdatatest/a1
-bash: /bigdatatest/a1: No such file or directory
[root@sandbox bigdata]# cat >>a1
hello world
[root@sandbox bigdata]# cat >>a2
ffffjhsf

gdsgdsghsd
gs
gs
hf
hf
h
f
[root@sandbox bigdata]# ls
a1 a2
[root@sandbox bigdata]# cat a1
```

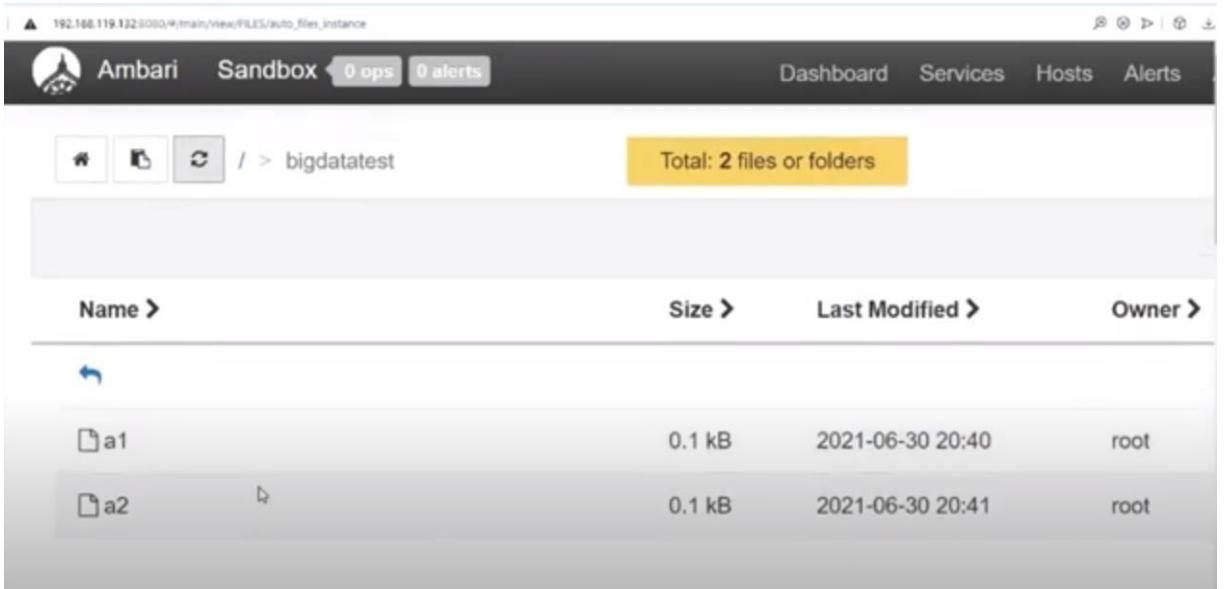
- 4) To upload files/directory to from local to HDFS

Put: to move a local file or directories into the distributed file system
Command: hdfs dfs -put a1 /bigdatatest/ and hdfs dfs -put a2 /bigdatatest/ will upload both the files.



```
< > C BB 192.168.119.132:4200
[root@sandbox bigdata]# hdfs dfs -put a1 /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -ls /bigdatatest
Found 1 items
-rw-r--r-- 1 root hdfs 13 2021-06-30 15:10 /bigdatatest/a1
[root@sandbox bigdata]# hdfs dfs -put a2 /bigdatatest/
[root@sandbox bigdata]#
```

Refresh the user interface we can see both the files.



The screenshot shows the Ambari UI with the "Sandbox" cluster selected. In the top navigation bar, there are tabs for Dashboard, Services, Hosts, and Alerts. Below the navigation bar, there is a breadcrumb trail showing the path: / > bigdatatest. A yellow callout box indicates "Total: 2 files or folders". The main content area displays a table of files and folders in the "/bigdatatest" directory. The table has columns for Name, Size, Last Modified, and Owner. There are two entries: "a1" and "a2". Both files are owned by "root" and have a size of 0.1 kB, with last modified dates of 2021-06-30 20:40 and 2021-06-30 20:41 respectively.

Name	Size	Last Modified	Owner
a1	0.1 kB	2021-06-30 20:40	root
a2	0.1 kB	2021-06-30 20:41	root

To download files/directories from hdfs to local
Get: To copy files/folders from hdfs store to local file system.
Command: hdfs dfs -get

/bigdatatest/a1 and hdfs dfs -get /bigdatatest/ a2 will upload both the files.

```
< > C 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bl...
Found 1 items
-rw-r--r-- 1 root hdfs 13 2021-06-30 15:10 /bigdatatest/a1
[root@sandbox bigdata]# hdfs dfs -put a2 /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/a1
get: `a1': File exists
[root@sandbox bigdata]# ls
a1 a2
[root@sandbox bigdata]# rm a2
rm: remove regular file `a2'? y
[root@sandbox bigdata]# rm a1
rm: remove regular file `a1'? y
[root@sandbox bigdata]# ls
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/a1
[root@sandbox bigdata]# ls
a1
[root@sandbox bigdata]# cat a1
hello world
[root@sandbox bigdata]#
```

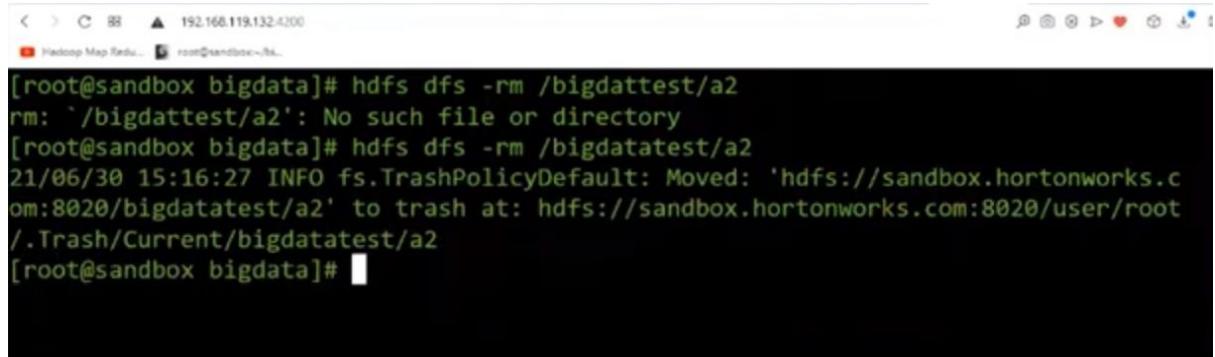
5) To remove file from local use rm command

```
< > C 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bl...
[root@sandbox bigdata]# hdfs dfs -put a1 /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -ls /bigdatatest
Found 1 items
-rw-r--r-- 1 root hdfs 13 2021-06-30 15:10 /bigdatatest/a1
[root@sandbox bigdata]# hdfs dfs -put a2 /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/a1
get: `a1': File exists
[root@sandbox bigdata]# ls
a1 a2
[root@sandbox bigdata]# rm a2 I
rm: remove regular file `a2'? y
[root@sandbox bigdata]# rm a1
rm: remove regular file `a1'? 
```

6) To remove file from Hadoop directory

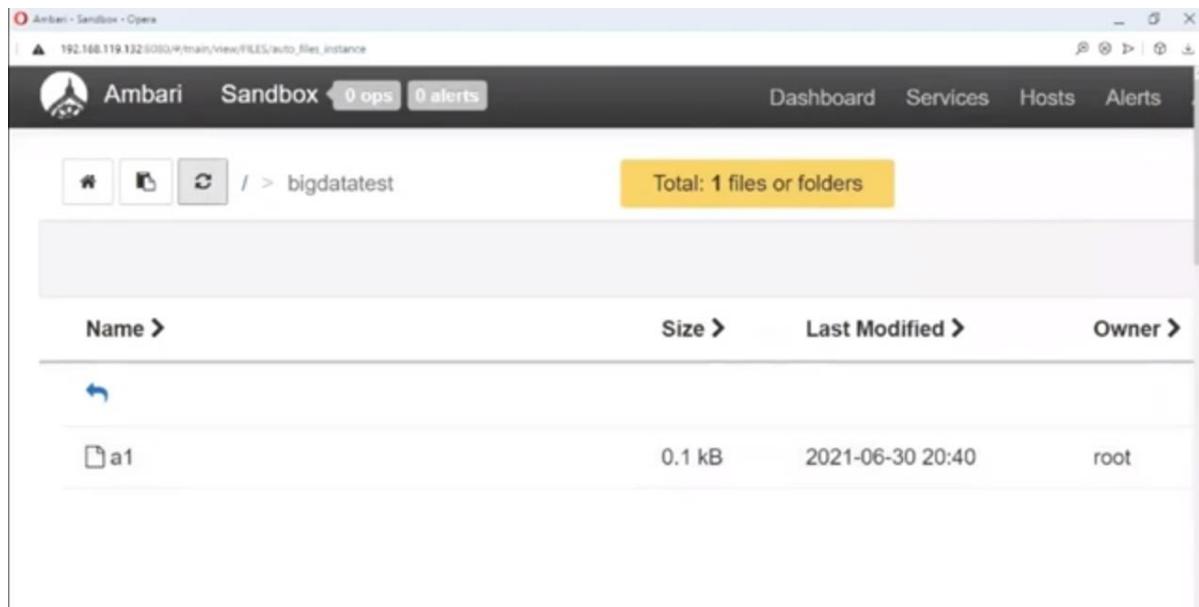
Command: hdfs dfs -rm a2 /gibdatatest/a2

MSc IT Sem II



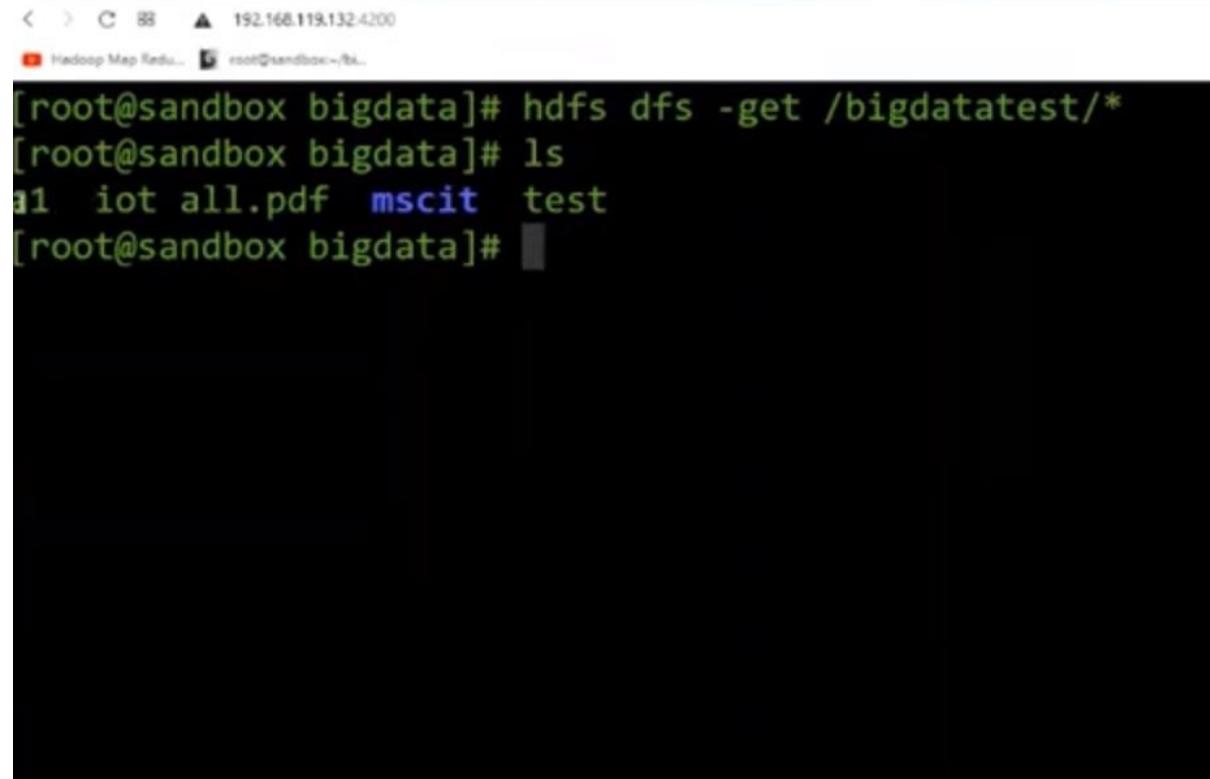
```
< > C ⌂ 192.168.119.132:4200
[root@sandbox bigdata]# hdfs dfs -rm /bigdattest/a2
rm: '/bigdattest/a2': No such file or directory
[root@sandbox bigdata]# hdfs dfs -rm /bigdattest/a2
21/06/30 15:16:27 INFO fs.TrashPolicyDefault: Moved: 'hdfs://sandbox.hortonworks.com:8020/bigdattest/a2' to trash at: hdfs://sandbox.hortonworks.com:8020/user/root/.Trash/Current/bigdattest/a2
[root@sandbox bigdata]#
```

Now refresh the UI and the file will be deleted.



The screenshot shows the Ambari Sandbox HDFS UI. The top navigation bar includes 'Ambari - Sandbox - Opera', the IP address '192.168.119.132:8080', and tabs for 'Dashboard', 'Services', 'Hosts', and 'Alerts'. Below the navigation is a search bar and a breadcrumb trail: 'Ambari' > 'Sandbox' > '0 ops' > '0 alerts' > '/ > bigdattest'. A yellow box indicates 'Total: 1 files or folders'. A table lists one file: 'a1' (Size: 0.1 kB, Last Modified: 2021-06-30 20:40, Owner: root). There are also icons for creating a new folder ('New Folder') and a trash can ('Delete').

- 7) To download all the files from hdfs to local Command: hdfs dfs -get /bigdattest/*



The screenshot shows a terminal window with the following session:

```
< > C 88 ▲ 192.168.119.132:4200
[Hadoop Map Redu... root@sandbox:~/bigdata]# hdfs dfs -get /bigdatatest/*
[root@sandbox bigdata]# ls
a1 iot all.pdf mscit test
[root@sandbox bigdata]#
```

8) Change user and directory and change user
only Command: su – hdfs (Change user and directory)
Su hdfs (change user only)

MSc IT Sem II

```
< > C B8 | ▲ 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bl...
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/*
[root@sandbox bigdata]# ls
a1 iot all.pdf mscit test
[root@sandbox bigdata]# hdfs dfs -put * /bigdatatest/x
put: unexpected URISyntaxException
put: `/bigdatatest/x': No such file or directory
[root@sandbox bigdata]# pwd
/root/bigdata
[root@sandbox bigdata]# su hdfs
[hdfs@sandbox bigdata]$ pwd
/root/bigdata
[hdfs@sandbox bigdata]$ exit
exit
[root@sandbox bigdata]# su -Ihdfs
[hdfs@sandbox ~]$ pwd
/home/hdfs
[hdfs@sandbox ~]$ 
```

Practical 2

Implement word count / frequency programs using MapReduce

Map Reduce as two component Map and Reduce.

Java program:

```
write program save as WordCount.java///////////
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import
org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordCount { public static class TokenizerMapper
extends Mapper<Object, Text, Text, IntWritable>{
private final static IntWritable one = new IntWritable(1);
private Text word = new Text();
public void map(Object key, Text value, Context context
) throws IOException, InterruptedException
{ StringTokenizer itr = new
StringTokenizer(value.toString()); while
(itr.hasMoreTokens()) ///"This is the output is the"
word.set(itr.nextToken());
context.write(word, one);
}
}
}
}
public static class IntSumReducer extends
Reducer<Text,IntWritable,Text,IntWritable> {
private IntWritable result = new IntWritable();
public void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException,
InterruptedException
{//is,3
int sum = 0;
for (IntWritable val : values)
{ sum += val.get();}
```

```
}

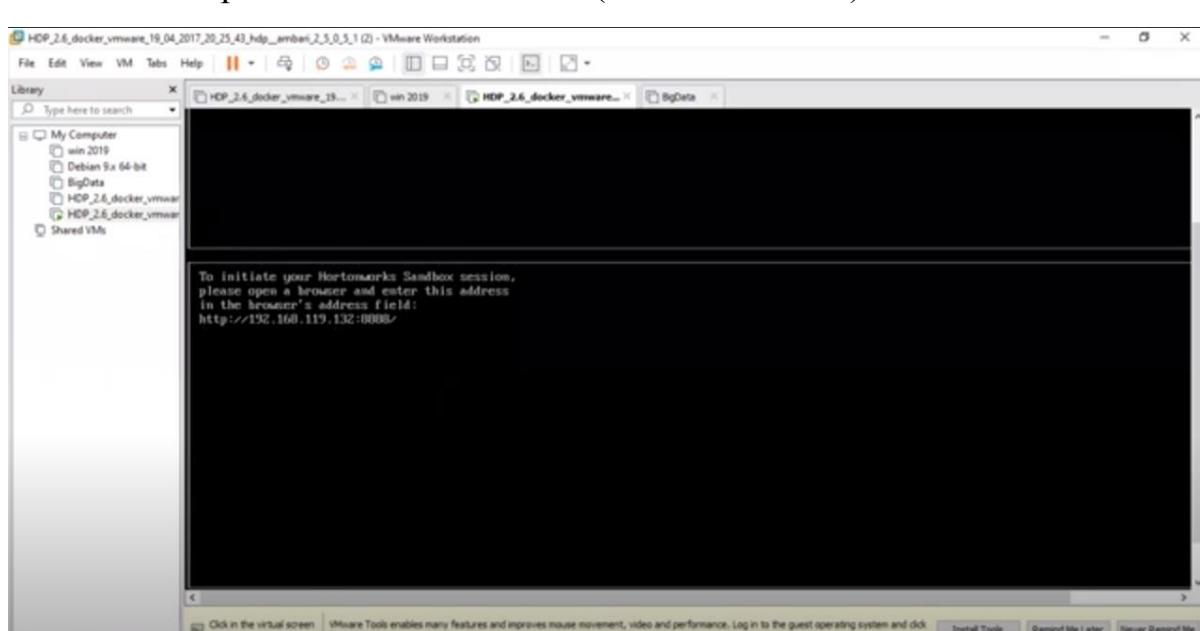
result.set(sum);
context.write(key, result);
}

}public static void main(String[] args) throws Exception
{ Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word count");
job.setJarByClass(WordCount.class);
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(args[0]));
FileOutputFormat.setOutputPath(job, new Path(args[1]));
System.exit(job.waitForCompletion(true)?0:1);
}
}

//////////Text File:
```

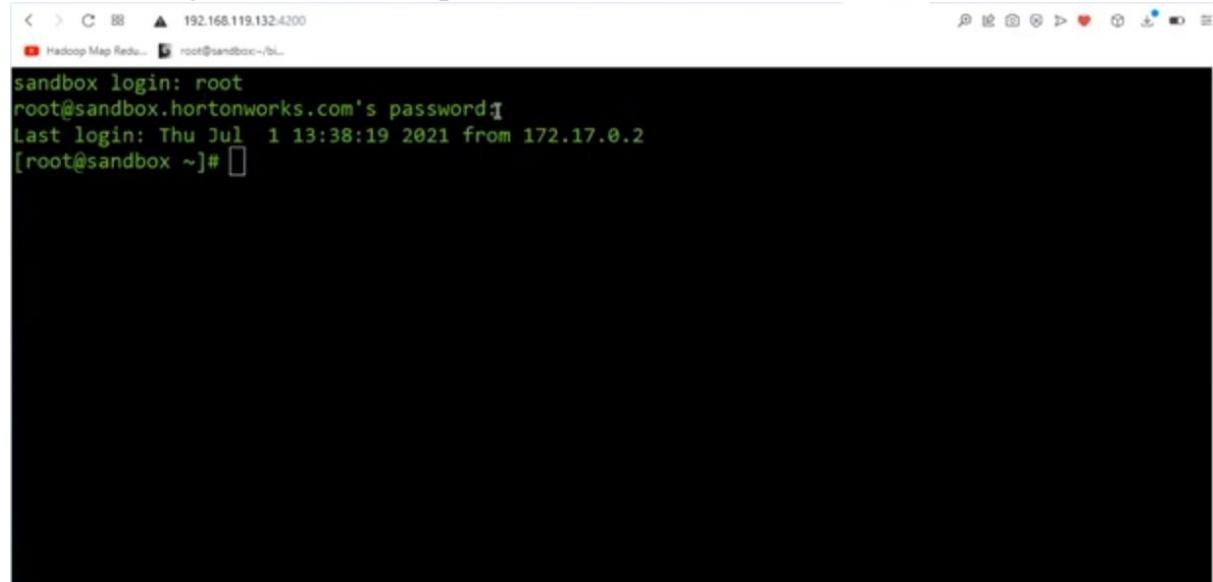
Hello World

This is the output is theStart the server (Horton Sandbox)



Open the terminal with 192.168.119.132/4200

Enter the login: root and the password and enter



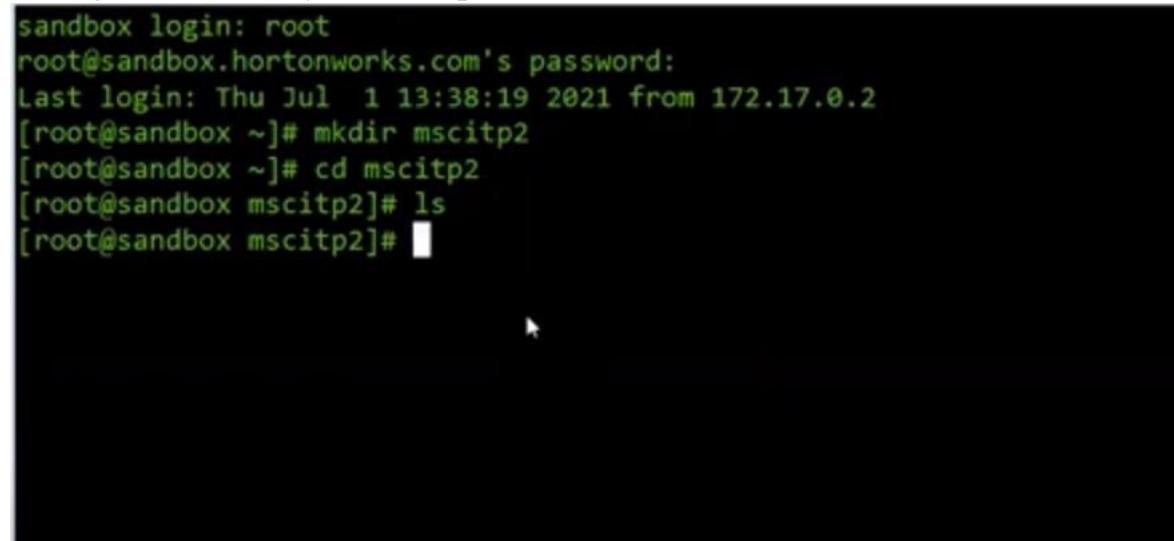
A screenshot of a terminal window titled "Hadoop Map Redu...". The window shows a root login session. The text in the terminal is:

```
192.168.119.132:4200
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]#
```

Create a folder in local directory.

Command: mkdir mscitp2

Change the directory cd mscitp2



A screenshot of a terminal window showing the creation of a folder and changing directory. The text in the terminal is:

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]#
```

Now create input file

Command: cat >> wordin.txt

Paste the text by right clicking on terminal

Hello World

This is the output is the

MSc IT Sem II

A screenshot of a terminal window titled "192.168.119.132:4200". The terminal shows a root shell on a sandbox host. A context menu is open over the command line, with the "Paste from browser" option highlighted. A small modal dialog box is overlaid on the terminal, containing the text "192.168.119.132:4200 says" and a text input field with the placeholder "Hello World\n>This is the output is the".

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]# cat >>wordin.txt
```

Copy
Paste
Paste from browser
Reset
✓ Unicode
Visual Bell
Onscreen Keyboard
Disable Alt Key
✓ Blinking Cursor
About...

A screenshot of a terminal window titled "192.168.119.132:4200". The terminal shows a root shell on a sandbox host. The command "cat >>wordin.txt" has been run, and the output "Hello World" is visible. Below the command line, the text "This is the output is the" is displayed, indicating the result of the paste operation.

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]# cat >>wordin.txt
Hello World
This is the output is the
```

To remove the extra space type command

vi wordin.txt

After removing the extra space check the content of the file

cat wordin.txt

MSc IT Sem II

```
< > C 88 ▲ 192.168.119.132:4200
Hadoop Map Redu... root@ sandbox:~/bl...
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]# cat >>wordin.txt
Hello World

This is the output is the
[root@sandbox mscitp2]# vi wordin.txt
[root@sandbox mscitp2]# cat wordin.txt
Hello World
This is the output is the
[root@sandbox mscitp2]#
```

Create another file wordcount.java

```
This is the output is the
[root@sandbox mscitp2]# vi wordin.txt
[root@sandbox mscitp2]# cat wordin.txt
Hello World
This is the output is the
[root@sandbox mscitp2]# cat >>WordCount.java
```

Paste the java code.

MSc IT Sem II

A screenshot of a terminal window titled '192.168.119.132:4200'. The terminal shows a Java code execution process. A context menu is open over the terminal window, with the 'Paste from browser' option highlighted. A tooltip window titled '192.168.119.132:4200 says' contains the Java code: 'new Path(args[1]).~> System.exit(job.waitForCompletion(true)?0:1);~>}'. Buttons for 'OK' and 'Cancel' are visible at the bottom of the tooltip.

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2018
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]# cat >>wordin.txt
Hello World
This is the output i
[root@sandbox mscitp2]# t.java
[root@sandbox mscitp2]#
```

Press control d to save the file

Check both the files create with command ls

A screenshot of a terminal window titled '192.168.119.132:4200'. The terminal shows the result of the 'ls' command, which lists two files: 'WordCount.java' and 'wordin.txt'. The terminal prompt '[root@sandbox mscitp2]#' is visible at the end.

```
[root@sandbox mscitp2]# ls
WordCount.java wordin.txt
[root@sandbox mscitp2]#
```

Now, to compile the java file

```
export HADOOP_CLASSPATH=$(hadoop classpath)
mkdir classes (To keep the compile files)
```

```
javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java
```

```
[root@sandbox mscitp2]# ls  
WordCount.java wordin.txt  
[root@sandbox mscitp2]# export HADOOP_CLASSPATH=$(hadoop classpath)  
[root@sandbox mscitp2]# ls  
WordCount.java wordin.txt  
[root@sandbox mscitp2]# mkdir classes  
[root@sandbox mscitp2]# javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java  
[root@sandbox mscitp2]#
```

Check class files are created with command ls classes

```
WordCount.java wordin.txt  
[root@sandbox mscitp2]# mkdir classes  
[root@sandbox mscitp2]# javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java  
[root@sandbox mscitp2]# ls classes  
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class  
[root@sandbox mscitp2]#
```

Now we have to bind all the class into single jar file with below command

```
jar -cvf WordCount.jar -C classes/ .
```

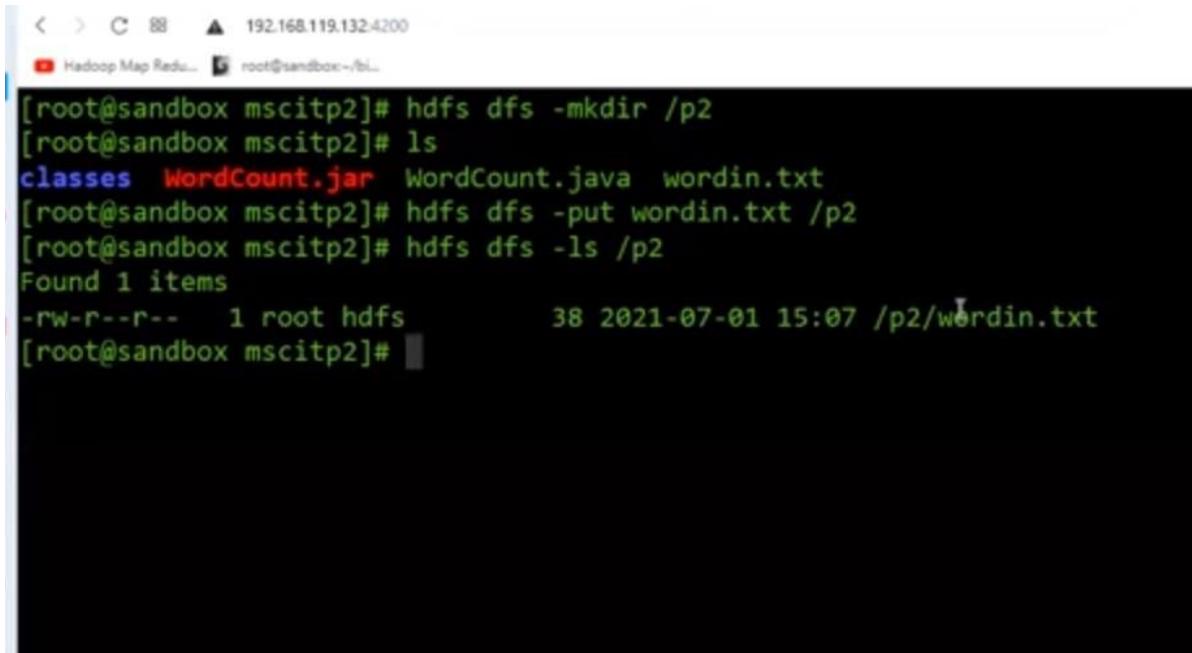
```
[root@sandbox mscitp2]# ls  
WordCount.java wordin.txt  
[root@sandbox mscitp2]# export HADOOP_CLASSPATH=$(hadoop classpath)  
[root@sandbox mscitp2]# ls  
WordCount.java wordin.txt  
[root@sandbox mscitp2]# mkdir classes  
[root@sandbox mscitp2]# javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java  
[root@sandbox mscitp2]# ls classes  
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class  
[root@sandbox mscitp2]# ls  
classes WordCount.java wordin.txt  
[root@sandbox mscitp2]# jar -cvf WordCount.jar -C classes/ .  
added manifest  
adding: WordCount$IntSumReducer.class(in = 1739) (out= 742)(deflated 57%)  
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 756)(deflated 56%)  
adding: WordCount.class(in = 1491) (out= 813)(deflated 45%)  
[root@sandbox mscitp2]#
```

Run ls command we can see jar file is created.

```
[root@sandbox mscitp2]# ls  
classes WordCount.jar WordCount.java wordin.txt  
[root@sandbox mscitp2]#
```

wordin.txt should be present in word directory of hdfs. So we need to upload wordin.txt file.

MSc IT Sem II

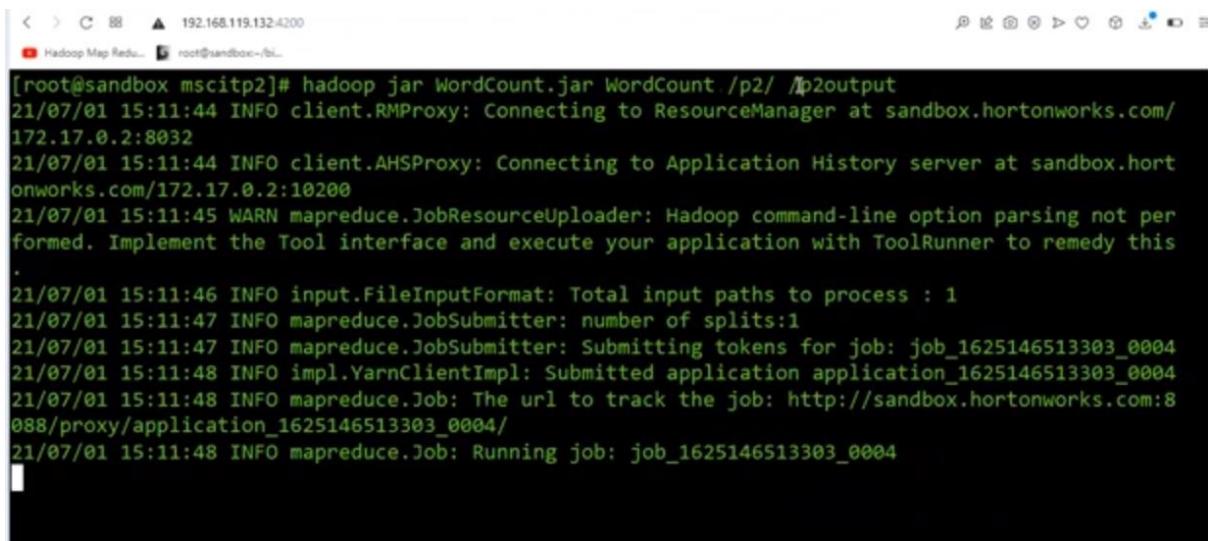


A screenshot of a terminal window titled "Hadoop Map Redu...". The IP address is 192.168.119.132:4200 and the user is root@sandbox. The terminal shows the following commands:

```
[root@sandbox mscitp2]# hdfs dfs -mkdir /p2
[root@sandbox mscitp2]# ls
classes WordCount.jar WordCount.java wordin.txt
[root@sandbox mscitp2]# hdfs dfs -put wordin.txt /p2
[root@sandbox mscitp2]# hdfs dfs -ls /p2
Found 1 items
-rw-r--r-- 1 root hdfs          38 2021-07-01 15:07 /p2/wordin.txt
[root@sandbox mscitp2]#
```

We need to put the final output p2output.

hadoop jar WordCount.jar WordCount /p2/ /p2output



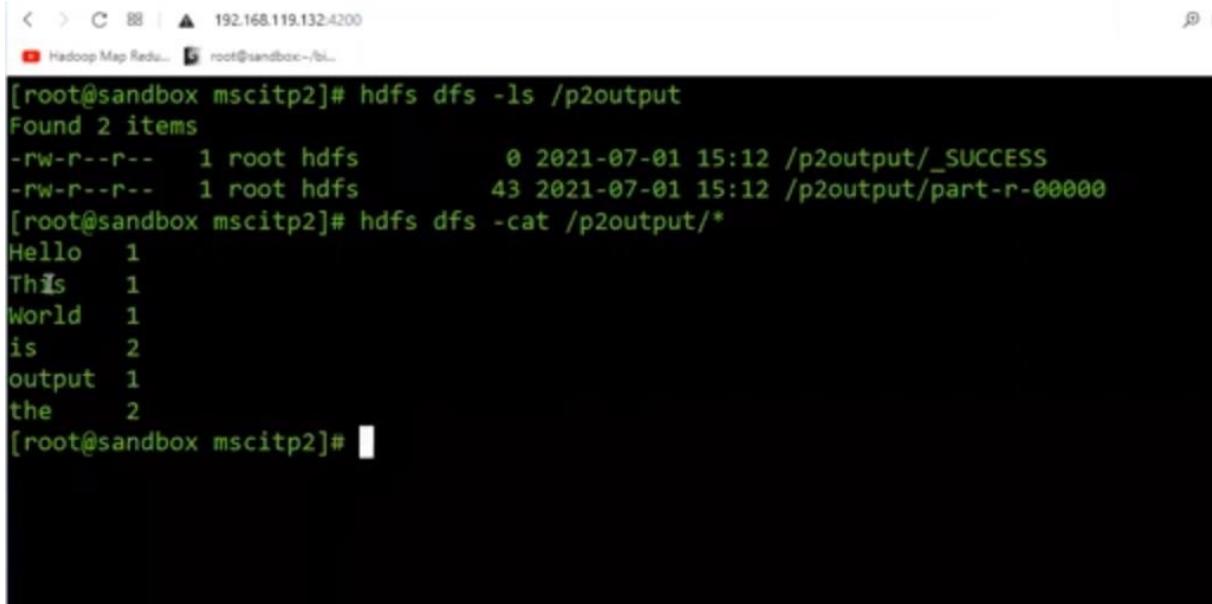
A screenshot of a terminal window titled "Hadoop Map Redu...". The IP address is 192.168.119.132:4200 and the user is root@sandbox. The terminal shows the following command and its output:

```
[root@sandbox mscitp2]# hadoop jar WordCount.jar WordCount /p2/ /p2output
21/07/01 15:11:44 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8032
21/07/01 15:11:44 INFO client.AHSProxy: Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
21/07/01 15:11:45 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
.
21/07/01 15:11:46 INFO input.FileInputFormat: Total input paths to process : 1
21/07/01 15:11:47 INFO mapreduce.JobSubmitter: number of splits:1
21/07/01 15:11:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1625146513303_0004
21/07/01 15:11:48 INFO impl.YarnClientImpl: Submitted application application_1625146513303_0004
21/07/01 15:11:48 INFO mapreduce.Job: The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1625146513303_0004/
21/07/01 15:11:48 INFO mapreduce.Job: Running job: job_1625146513303_0004
```

Print the content of the output file

Command: hdfs dfs -cat /p2output/*

MSc IT Sem II



The screenshot shows a terminal window with the following session:

```
[root@sandbox mscitp2]# hdfs dfs -ls /p2output
Found 2 items
-rw-r--r-- 1 root hdfs          0 2021-07-01 15:12 /p2output/_SUCCESS
-rw-r--r-- 1 root hdfs        43 2021-07-01 15:12 /p2output/part-r-00000
[root@sandbox mscitp2]# hdfs dfs -cat /p2output/*
Hello    1
This    1
World    1
is        2
output   1
the      2
[root@sandbox mscitp2]#
```

Ctrl + l to clear the screen.

vi filename.txt= this command will create/ open filename.txt

two modes of vi editor

- 1) Insert mode – i (press i key)
- 2) Command mode – esc key

:wq is to save and exit

Practical 3

Implement an MapReduce program that processes a weather dataset.

Java program:

```
MyMaxMin.java
///////////////
// importing Libraries
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import
org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import
org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;public class MyMaxMin {
    // Mapper

    /*MaxTemperatureMapper class is static
     * and extends Mapper abstract class
     * having four Hadoop generics type
     * LongWritable, Text, Text, Text.
    */
}
```

```
public static class MaxTemperatureMapper extends  
    Mapper<LongWritable, Text, Text, Text> {  
    public static final int MISSING = 9999;  
  
    @Override  
    public void map(LongWritable arg0, Text Value, Context context)  
        throws IOException, InterruptedException {  
  
        String line = Value.toString();  
  
        // Check for the empty line  
        if (!(line.length() == 0)) {  
  
            // from character 6 to 14 we have  
            // the date in our dataset  
            String date = line.substring(6, 14);  
            // similarly we have taken the maximum  
            // temperature from 39 to 45 characters  
            float temp_Max = Float.parseFloat(line.substring(39,  
45).trim());  
  
            // similarly we have taken the minimum  
            // temperature from 47 to 53 characters  
            float temp_Min = Float.parseFloat(line.substring(47,  
53).trim());  
            // if maximum temperature is  
            // greater than 30, it is a hot day  
        }  
    }  
}
```

```
if (temp_Max > 30.0) {  
  
    // Hot day  
    context.write(new Text("The Day is Hot Day :"  
+ date),  
                new  
Text(String.valueOf(temp_Max)));  
}  
// if the minimum  
temperature is  
  
// less than 15, it is a cold day  
if (temp_Min < 15) {  
  
    // Cold day  
    context.write(new Text("The Day is Cold Day  
:" + date),  
                new  
Text(String.valueOf(temp_Min)));  
}  
}  
}  
}  
} // Reducer  
  
/*MaxTemperatureReducer class is  
static and extends Reducer abstract class  
having four Hadoop generics type  
Text, Text, Text, Text.  
*/  
//The Day is Cold Day :20150101 ,-21.8  
  
public static class MaxTemperatureReducer extends
```

```
Reducer<Text, Text, Text, Text> {          /**
 * @method reduce
 * This method takes the input as key and
 * list of values pair from the mapper,
 * it does aggregation based on keys and
 * produces the final context.
 */
}

public void reduce(Text Key, Iterator<Text> Values, Context
context) throws IOException, InterruptedException {
    // putting all the values in
    // temperature variable of type String
    String temperature = Values.next().toString();
    context.write(Key, new Text(temperature));
} } /**

* @method main
* This method is used for setting
* all the configuration properties.
* It acts as a driver for map-reduce
* code.
*/
}

public static void main(String[] args) throws Exception {      // reads
the default configuration of the
    // cluster from the configuration XML files
    Configuration conf = new Configuration();
```

```
// Initializing the job with the
// default configuration of the cluster
Job job = new Job(conf, "weather example");

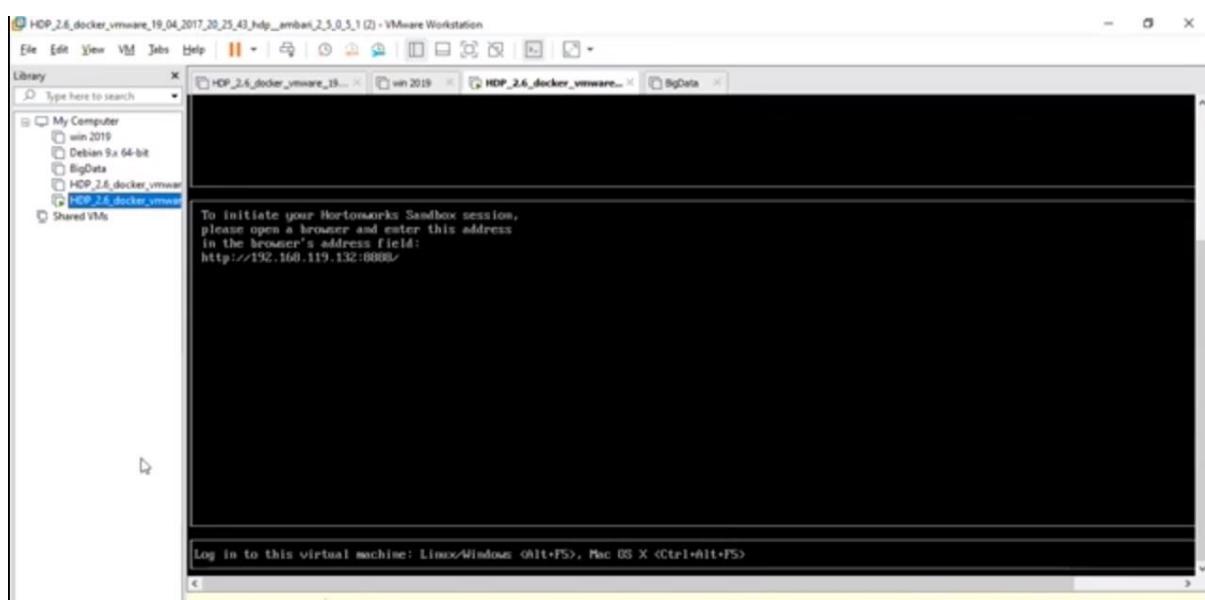
// Assigning the driver class name
job.setJarByClass(MyMaxMin.class);           // Key type coming
out of mapper
job.setMapOutputKeyClass(Text.class);

// value type coming out of mapper
job.setMapOutputValueClass(Text.class);        // Defining the
mapper class name
job.setMapperClass(MaxTemperatureMapper.class);

// Defining the reducer class name
job.setReducerClass(MaxTemperatureReducer.class); // Defining input Format class which is
                                                     // responsible to parse the dataset
                                                     // into a key value pair
job.setInputFormatClass(TextInputFormat.class);

// Defining output Format class which is
// responsible to parse the dataset
// into a key value pair
job.setOutputFormatClass(TextOutputFormat.class); // setting the second argument
                                                 // as a path in a path variable
Path outputPath = new Path(args[1]);           // Configuring the
input path
```

```
// from the filesystem into the job  
FileInputFormat.addInputPath(job, new Path(args[0])); //  
Configuring the output path from  
// the filesystem into the job  
FileOutputFormat.setOutputPath(job, new Path(args[1])); //  
deleting the context path automatically  
// from hdfs so that we don't have  
// to delete it explicitly  
OutputPath.getFileSystem(conf).delete(OutputPath);  
// flag value becomes false  
System.exit(job.waitForCompletion(true) ? 0 : 1); }  
}  
///////////Start the server
```



Open the terminal with 192.168.119.132/4200

Enter the login: root and the password and enter

MSc IT Sem II

A screenshot of a terminal window titled "192.168.119.132:4200". The title bar also shows "Hadoop Map Redu..." and "root@sandbox:~\$". The terminal prompt is "root@sandbox ~]#". The session starts with "sandbox login: root", followed by a password prompt "root@sandbox.hortonworks.com's password:", and ends with "Last login: Thu Jul 1 13:38:19 2021 from 172.17.0.2". The command history shows "[root@sandbox ~]#".

Create a folder in local directory.

Command: mkdir mscitp3

Change the directory cd mscitp3

A screenshot of a terminal window titled "192.168.119.132:4200". The title bar also shows "Hadoop Map Redu..." and "root@sandbox:~\$ Bulk Image Resize". The terminal prompt is "root@sandbox ~]#". The session starts with "sandbox login: root", followed by a password prompt "root@sandbox.hortonworks.com's password:", and ends with "Last login: Thu Jul 1 14:53:58 2021 from 172.17.0.2". The command history shows "[root@sandbox ~]# ls", "[root@sandbox ~]# mkdir mscitp3", and "[root@sandbox ~]# cd mscitp3".

Now create input file

Command: cat >> weatherin2.txt

Paste the weather dataset by right clicking on terminal

Ctrl d will save the file

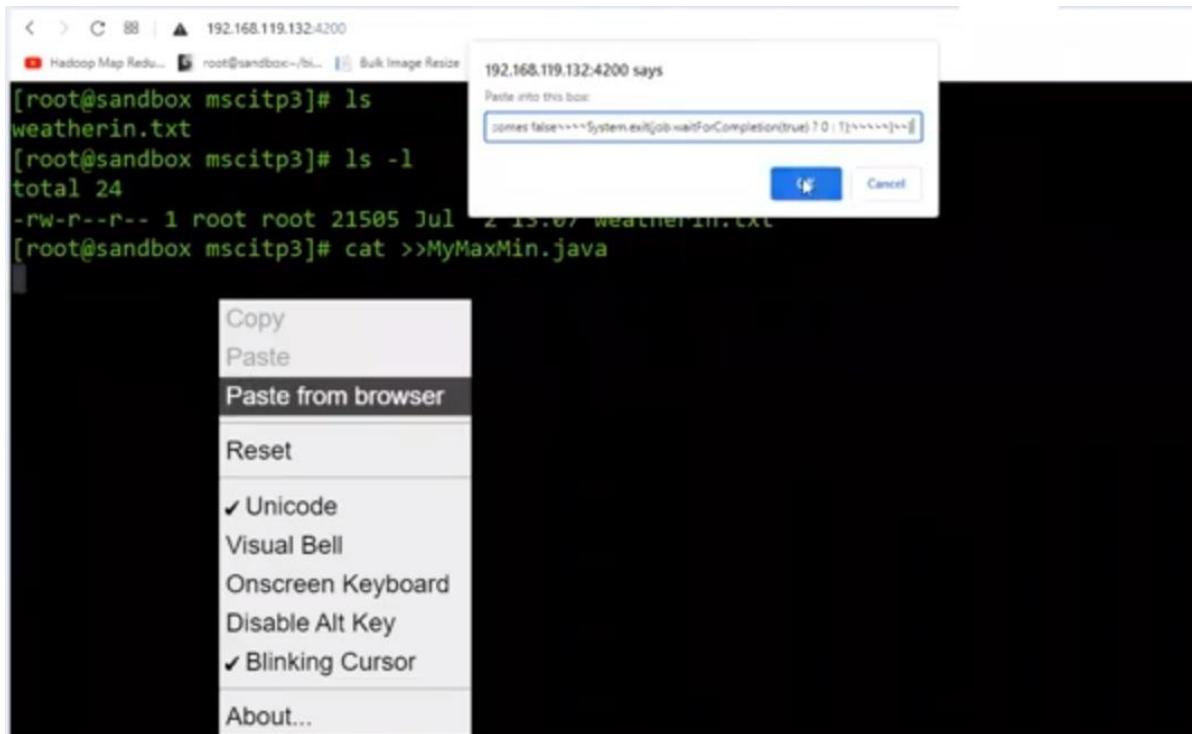
Run command ls to see the file.

Create java file

Command: cat >>MyMaxMin.java

MSc IT Sem II

Paste the java code and ctrl d to save the file



The screenshot shows a terminal window on a Linux system (Ubuntu) with the IP address 192.168.119.132:4200. The user has run the command 'ls' and then 'ls -l', both of which show a file named 'weatherin.txt'. The user then runs 'cat >>MyMaxMin.java' to append the contents of 'weatherin.txt' to a Java source file. A context menu is open over the terminal window, with the 'Paste from browser' option highlighted. A clipboard dialog box is overlaid on the terminal, containing the Java code: 'comes false\n***System.exit(job.waitForCompletion(true)) TO : T\n*****\n'. There are 'OK' and 'Cancel' buttons at the bottom of the clipboard dialog.

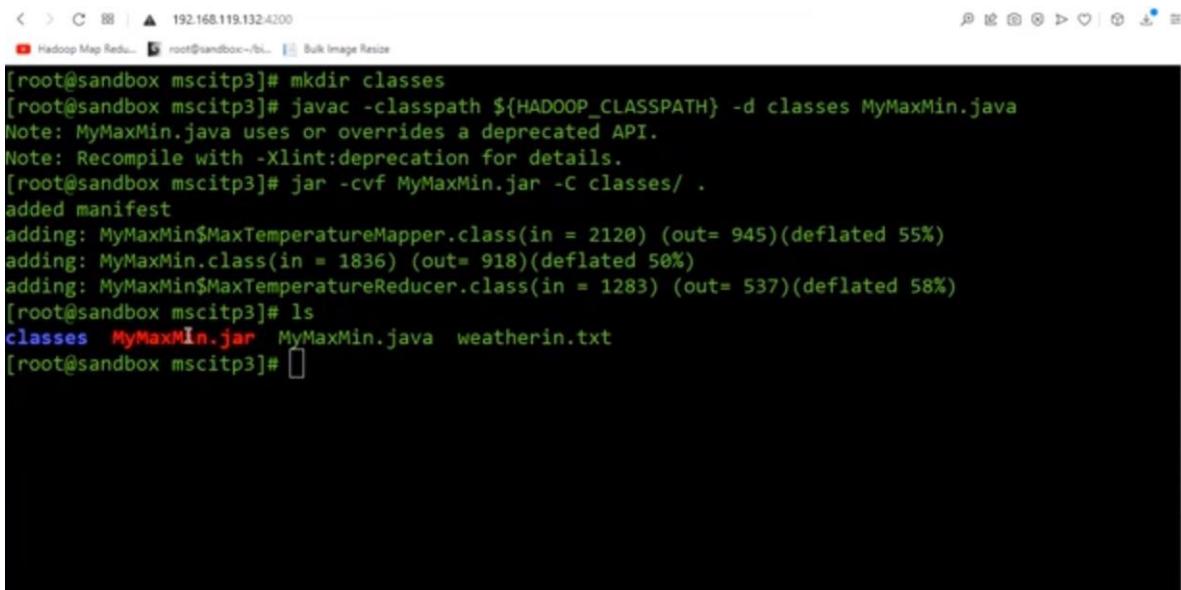
export HADOOP_CLASSPATH=\$(hadoop classpath) //compile and to create jar file

mkdir classes

javac -classpath \${HADOOP_CLASSPATH} -d classes MyMaxMin.java

After compile need to create a jar file

jar -cvf MyMaxMin.jar -C classes/ .



The screenshot shows a terminal window on a Linux system (Ubuntu) with the IP address 192.168.119.132:4200. The user runs 'mkdir classes', then 'javac -classpath \${HADOOP_CLASSPATH} -d classes MyMaxMin.java'. The terminal displays two warning messages: 'Note: MyMaxMin.java uses or overrides a deprecated API.' and 'Note: Recompile with -Xlint:deprecation for details.' The user then runs 'jar -cvf MyMaxMin.jar -C classes/ .' to create the jar file. The terminal shows the progress of the jar creation, including the addition of the 'MyMaxMin\$MaxTemperatureMapper.class' and 'MyMaxMin.class' files, and the 'MyMaxMin\$MaxTemperatureReducer.class' file. Finally, the user runs 'ls' to list the contents of the current directory, which includes the 'MyMaxMin.jar' file, 'MyMaxMin.java', and 'weatherin.txt'. The terminal ends with a blank line.

MSc IT Sem II

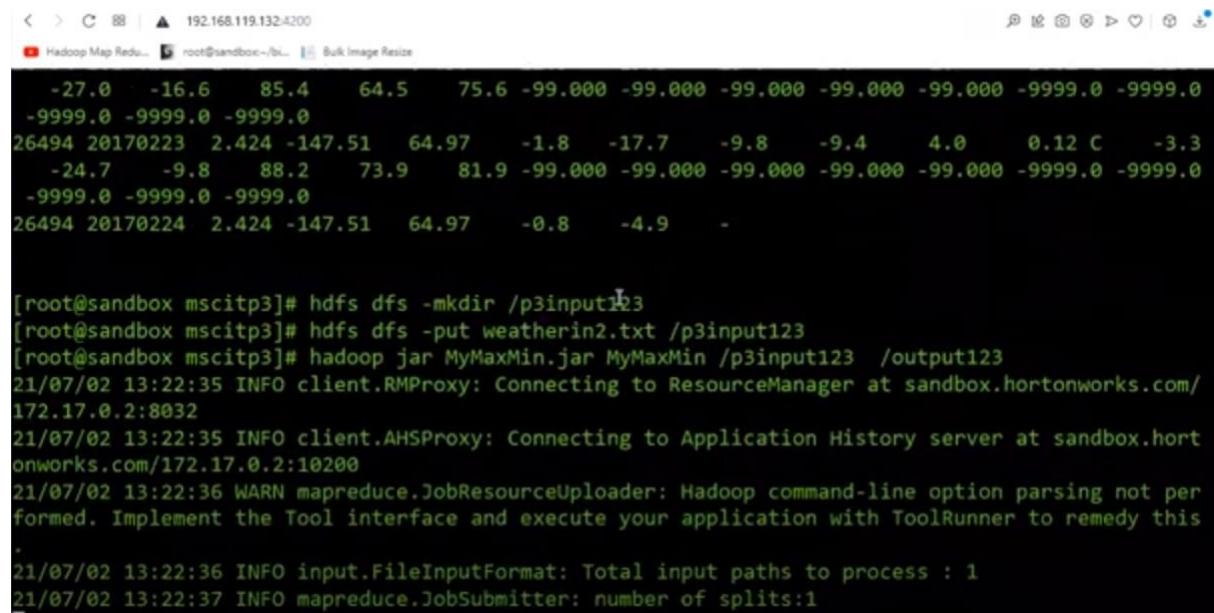
Now, put weatherin.txt in hdfs

Before that create a folder

Command: hdfs dfs -mkdir /p3input123

Then run command: hdfs dfs -put weatherin2.txt /p3input123

hadoop jar MyMaxMin.jar MyMaxMin /p3inputw /output123

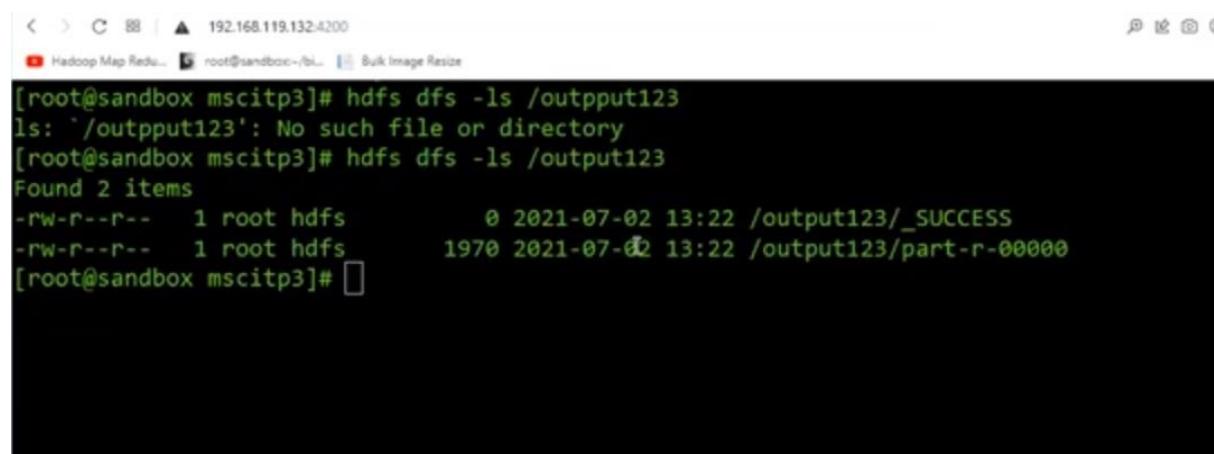


```
< > C 88 | ▲ 192.168.119.132:4200
[Hadoop Map Redu... root@sandbox:~/bi... Bulk Image Resize

-27.0 -16.6 85.4 64.5 75.6 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0
-9999.0 -9999.0 -9999.0
26494 20170223 2.424 -147.51 64.97 -1.8 -17.7 -9.8 -9.4 4.0 0.12 C -3.3
-24.7 -9.8 88.2 73.9 81.9 -99.000 -99.000 -99.000 -99.000 -99.000 -9999.0 -9999.0
-9999.0 -9999.0 -9999.0
26494 20170224 2.424 -147.51 64.97 -0.8 -4.9 -
[root@sandbox mscitp3]# hdfs dfs -mkdir /p3input123
[root@sandbox mscitp3]# hdfs dfs -put weatherin2.txt /p3input123
[root@sandbox mscitp3]# hadoop jar MyMaxMin.jar MyMaxMin /p3input123 /output123
21/07/02 13:22:35 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8032
21/07/02 13:22:35 INFO client.AHSProxy: Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
21/07/02 13:22:36 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this
.
21/07/02 13:22:36 INFO input.FileInputFormat: Total input paths to process : 1
21/07/02 13:22:37 INFO mapreduce.JobSubmitter: number of splits:1
```

Check outfile is created

Command: hdfs dfs -ls /output123

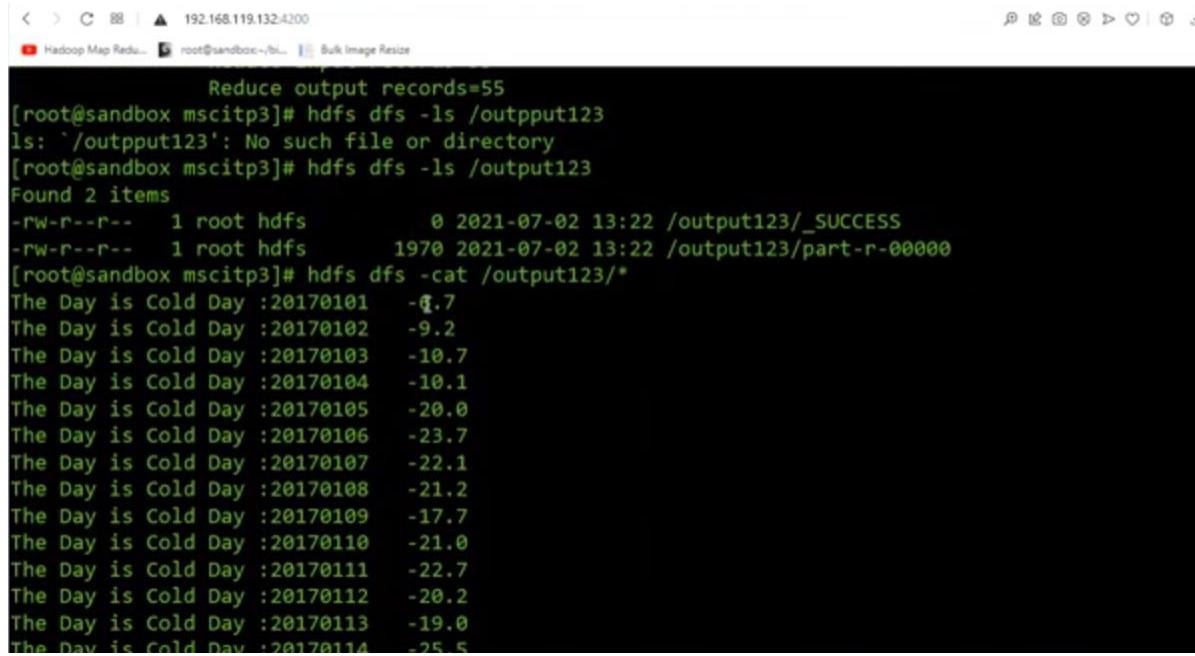


```
< > C 88 | ▲ 192.168.119.132:4200
[Hadoop Map Redu... root@sandbox:~/bi... Bulk Image Resize

[root@sandbox mscitp3]# hdfs dfs -ls /outpput123
ls: '/outpput123': No such file or directory
[root@sandbox mscitp3]# hdfs dfs -ls /output123
Found 2 items
-rw-r--r-- 1 root hdfs 0 2021-07-02 13:22 /output123/_SUCCESS
-rw-r--r-- 1 root hdfs 1970 2021-07-02 13:22 /output123/part-r-00000
[root@sandbox mscitp3]#
```

hdfs dfs -cat /output123/*

MSc IT Sem II



The screenshot shows a terminal window with the following session:

```
192.168.119.132:4200
Hadoop Map Redu... root@ sandbox mscitp3 Bulk Image Resize
Reduce output records=55
[root@ sandbox mscitp3]# hdfs dfs -ls /outpput123
ls: '/outpput123': No such file or directory
[root@ sandbox mscitp3]# hdfs dfs -ls /output123
Found 2 items
-rw-r--r-- 1 root hdfs 0 2021-07-02 13:22 /output123/_SUCCESS
-rw-r--r-- 1 root hdfs 1970 2021-07-02 13:22 /output123/part-r-00000
[root@ sandbox mscitp3]# hdfs dfs -cat /output123/*
The Day is Cold Day :20170101 -8.7
The Day is Cold Day :20170102 -9.2
The Day is Cold Day :20170103 -10.7
The Day is Cold Day :20170104 -10.1
The Day is Cold Day :20170105 -20.0
The Day is Cold Day :20170106 -23.7
The Day is Cold Day :20170107 -22.1
The Day is Cold Day :20170108 -21.2
The Day is Cold Day :20170109 -17.7
The Day is Cold Day :20170110 -21.0
The Day is Cold Day :20170111 -22.7
The Day is Cold Day :20170112 -20.2
The Day is Cold Day :20170113 -19.0
The Day is Cold Day :20170114 -25.5
```

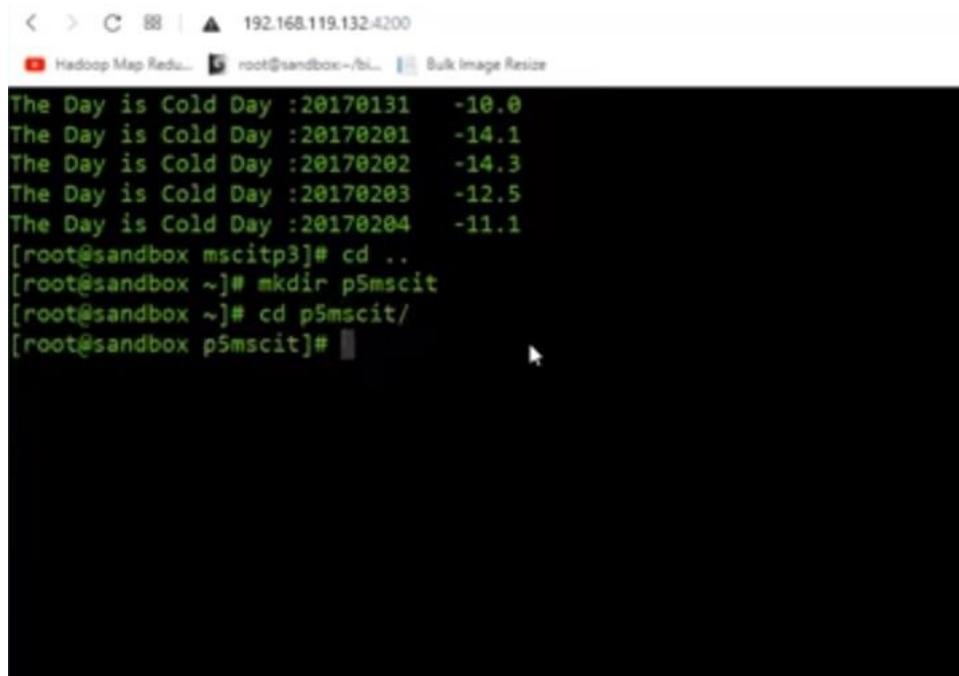
Practical 4

Implement the program using Pig.

Dataset:

```
001,Rajiv,Reddy,21,9848022337,Hyderabad  
002,siddarth,Battacharya,22,9848022338,Kolkata  
003,Rajesh,Khanna,22,9848022339,Delhi  
004,Preethi,Agarwal,21,9848022330,Pune  
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar  
006,Archana,Mishra,23,9848022335,Chennai  
007,Komal,Nayak,24,9848022334,trivendram  
008,Bharathi,Nambiayar,24,9848022333,Chennai  
#student.txtcreate a directory and get into that directory
```

Command: mkdir p5mscit



The screenshot shows a terminal window with the IP address 192.168.119.132:4200 at the top. The window title is "Hadoop Map Redu...". The terminal displays the following text:

```
The Day is Cold Day :20170131 -10.0  
The Day is Cold Day :20170201 -14.1  
The Day is Cold Day :20170202 -14.3  
The Day is Cold Day :20170203 -12.5  
The Day is Cold Day :20170204 -11.1  
[root@sandbox mscitp3]# cd ..  
[root@sandbox ~]# mkdir p5mscit  
[root@sandbox ~]# cd p5mscit/  
[root@sandbox p5mscit]#
```

Create a file

Command: cat >>student.txt

Right click and paste the text

MSc IT Sem II

The screenshot shows a terminal window with the IP address 192.168.119.132:4200. A context menu is open over a list of student records. The menu includes options like Copy, Paste, Paste from browser, Reset, Unicode (checked), Visual Bell, Onscreen Keyboard, and Disable Alt Key.

```
The Day is Cold Day :20170131 -10.0
The Day is Cold Day :20170201 -14.1
The Day is Cold Day :20170202 -14.3
The Day is Cold Day :20170203 -12.5
The Day is Cold Day :20170204 -11.1
[root@sandbox mscitp3]# cd ..
[root@sandbox ~]# mkdir p5mscit
[root@sandbox ~]# cd p5mscit/
[root@sandbox p5mscit]# cat student.txt
cat: student.txt: No such file or directory
[root@sandbox p5mscit]# cat >>student.txt
```

```
The Day is Cold Day :20170201 -14.1
The Day is Cold Day :20170202 -14.3
The Day is Cold Day :20170203 -12.5
The Day is Cold Day :20170204 -11.1
[root@sandbox mscitp3]# cd ..
[root@sandbox ~]# mkdir p5mscit
[root@sandbox ~]# cd p5mscit/
[root@sandbox p5mscit]# cat student.txt
cat: student.txt: No such file or directory
[root@sandbox p5mscit]# cat >>student.txt
001,Rajiv,Reddy,21,9848022337,Hyderabad
002,siddarth,Battacharya,22,9848022338,Kolkata
003,Rajesh,Khanna,22,9848022339,Delhi
004,Preethi,Agarwal,21,9848022330,Pune
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
006,Archana,Mishra,23,9848022335,Chennai
007,Komal,Nayak,24,9848022334,trivendram
008,Bharathi,Nambiayar,24,9848022333,Chennai
```

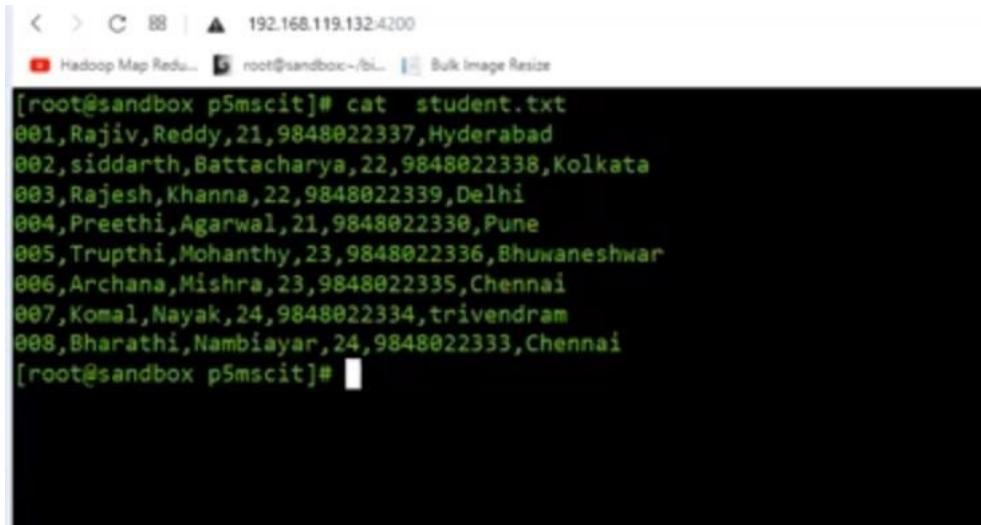
Remove the space with vi editor

Command: vi student.txt and press i for insert mode

After editing: wq and enter

Print the content and see the text

MSc IT Sem II

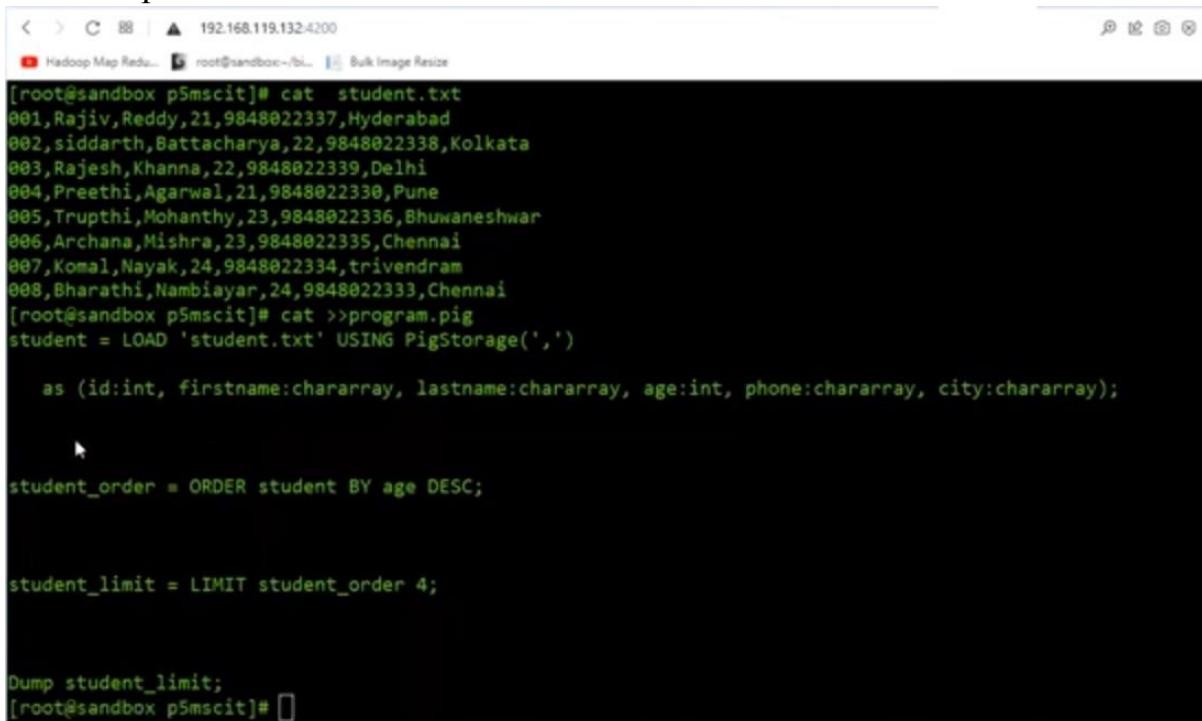


```
< > C 88 | ▲ 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bi... Bulk Image Resize
[root@sandbox p5mscit]# cat student.txt
001,Rajiv,Reddy,21,9848022337,Hyderabad
002,siddarth,Battacharya,22,9848022338,Kolkata
003,Rajesh,Khanna,22,9848022339,Delhi
004,Preethi,Agarwal,21,9848022330,Pune
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
006,Archana,Mishra,23,9848022335,Chennai
007,Komal,Nayak,24,9848022334,trivendram
008,Bharathi,Nambiyar,24,9848022333,Chennai
[root@sandbox p5mscit]#
```

Create a program file

```
//////////script start
student = LOAD 'student.txt' USING PigStorage(',')
    as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray,
city:chararray);

student_order = ORDER student BY age DESC;student_limit = LIMIT
student_order 4;Dump student_limit;
/////////script end
```



```
< > C 88 | ▲ 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bi... Bulk Image Resize
[root@sandbox p5mscit]# cat student.txt
001,Rajiv,Reddy,21,9848022337,Hyderabad
002,siddarth,Battacharya,22,9848022338,Kolkata
003,Rajesh,Khanna,22,9848022339,Delhi
004,Preethi,Agarwal,21,9848022330,Pune
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar
006,Archana,Mishra,23,9848022335,Chennai
007,Komal,Nayak,24,9848022334,trivendram
008,Bharathi,Nambiyar,24,9848022333,Chennai
[root@sandbox p5mscit]# cat >>program.pig
student = LOAD 'student.txt' USING PigStorage(',')
    as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray);

student_order = ORDER student BY age DESC;

student_limit = LIMIT student_order 4;

Dump student_limit;
[root@sandbox p5mscit]#
```

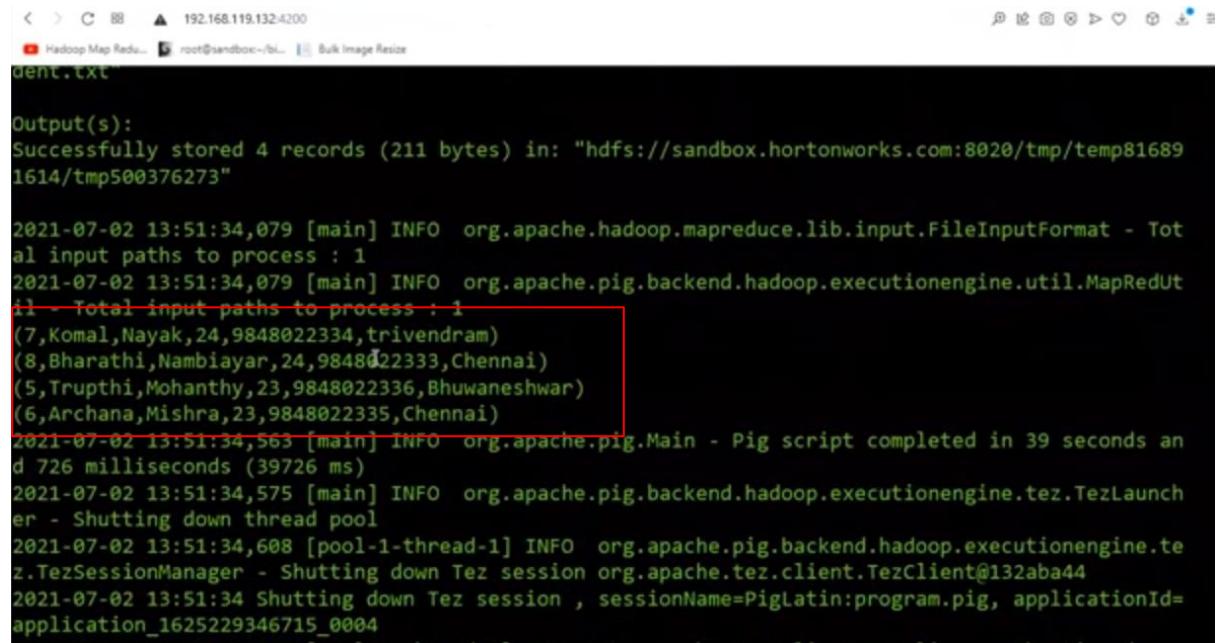
Upload student on hdfs

Command: hdfs dfs -put student.txt /user/root/

Run the pig program

```
[root@sandbox p5mscit]# hdfs dfs -put student.txt /user/root/
[root@sandbox p5mscit]# pig program.pig
```

Output:



```
< > C 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:/bl... Bulk Image Resize
student.txt

Output(s):
Successfully stored 4 records (211 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp816891614/tmp500376273"

2021-07-02 13:51:34,079 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2021-07-02 13:51:34,079 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
(7,Komal,Nayak,24,9848022334,trivendram)
(8,Bharathi,Nambiayar,24,9848022333,Chennai)
(5,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar)
(6,Archana,Mishra,23,9848022335,Chennai)
2021-07-02 13:51:34,563 [main] INFO org.apache.pig.Main - Pig script completed in 39 seconds and 726 milliseconds (39726 ms)
2021-07-02 13:51:34,575 [main] INFO org.apache.pig.backend.hadoop.executionengine.tez.TezLauncher - Shutting down thread pool
2021-07-02 13:51:34,608 [pool-1-thread-1] INFO org.apache.pig.backend.hadoop.executionengine.tez.TezSessionManager - Shutting down Tez session org.apache.tez.client.TezClient@132aba44
2021-07-02 13:51:34 Shutting down Tez session , sessionName=PigLatin:program.pig, applicationId=application_1625229346715_0004
```

Practical 5

Implement the application in Hive.

Dataset:

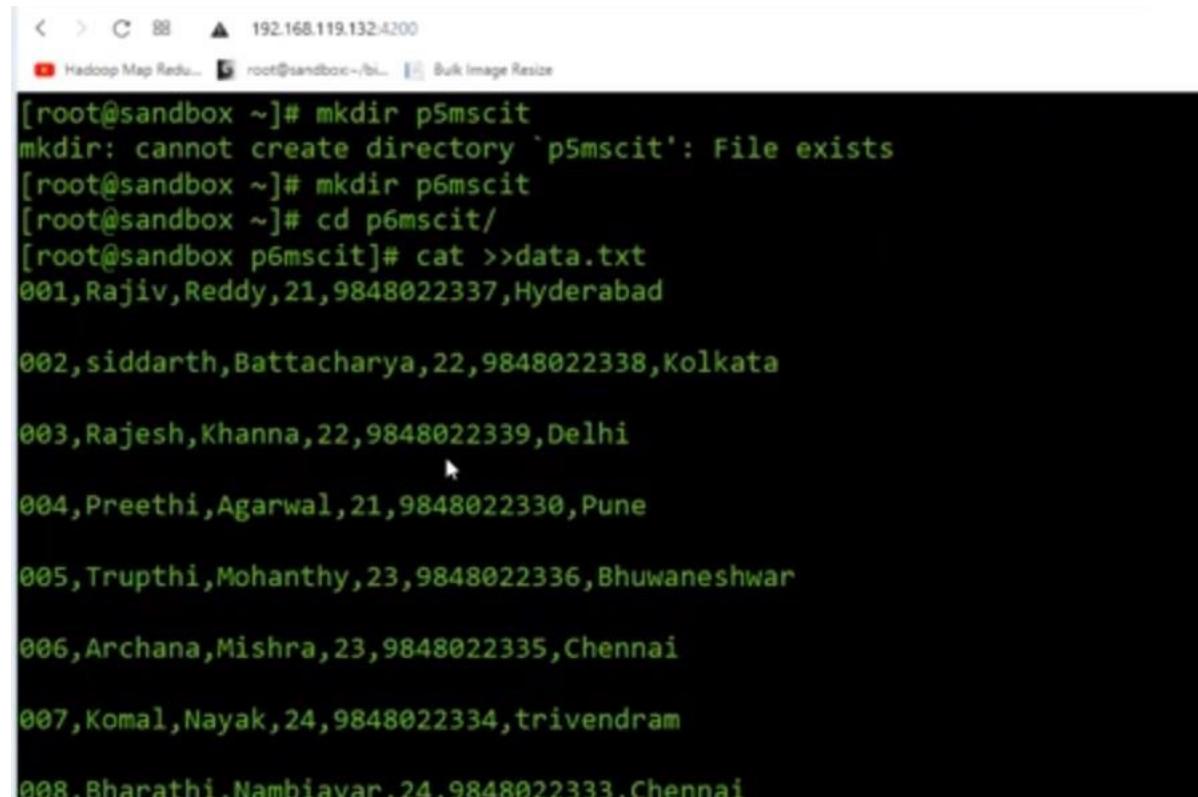
```
001,Rajiv,Reddy,21,9848022337,Hyderabad  
002,siddarth,Battacharya,22,9848022338,Kolkata  
003,Rajesh,Khanna,22,9848022339,Delhi  
004,Preethi,Agarwal,21,9848022330,Pune  
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar  
006,Archana,Mishra,23,9848022335,Chennai  
007,Komal,Nayak,24,9848022334,trivendram  
008,Bharathi,Nambiayar,24,9848022333,Chennai  
#student.txtcreate a directory and get into that directory
```

Command: mkdir p6mscit

Create a file

Command: cat >>data.txt

Right click and paste the text



```
< > C 192.168.119.132:4200  
Hadoop Map Redu... root@sandbox:~/bl... Bulk Image Resize  
[root@sandbox ~]# mkdir p5mscit  
mkdir: cannot create directory `p5mscit': File exists  
[root@sandbox ~]# mkdir p6mscit  
[root@sandbox ~]# cd p6mscit/  
[root@sandbox p6mscit]# cat >>data.txt  
001,Rajiv,Reddy,21,9848022337,Hyderabad  
  
002,siddarth,Battacharya,22,9848022338,Kolkata  
  
003,Rajesh,Khanna,22,9848022339,Delhi  
004,Preethi,Agarwal,21,9848022330,Pune  
  
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar  
  
006,Archana,Mishra,23,9848022335,Chennai  
  
007,Komal,Nayak,24,9848022334,trivendram  
  
008,Bharathi,Nambiayar,24,9848022333,Chennai
```

Remove the space with vi editor

MSc IT Sem II

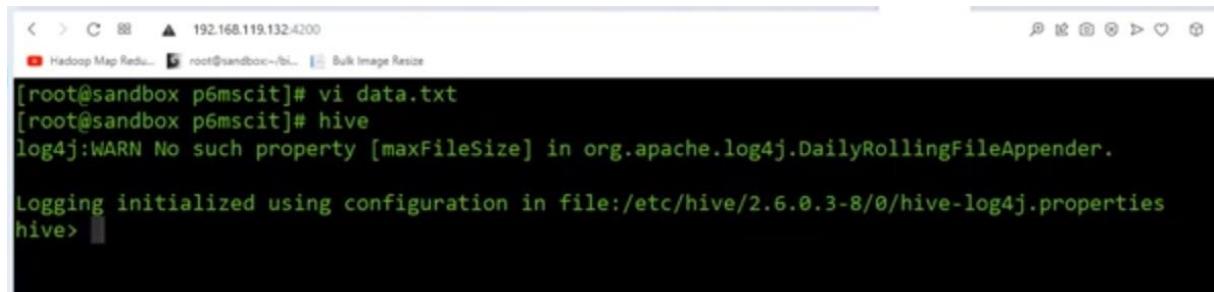
Command: vi student.txt and press i for insert mode

After editing: wq and enter

Print the content and see the text

Now start the hive terminal

Command: hive

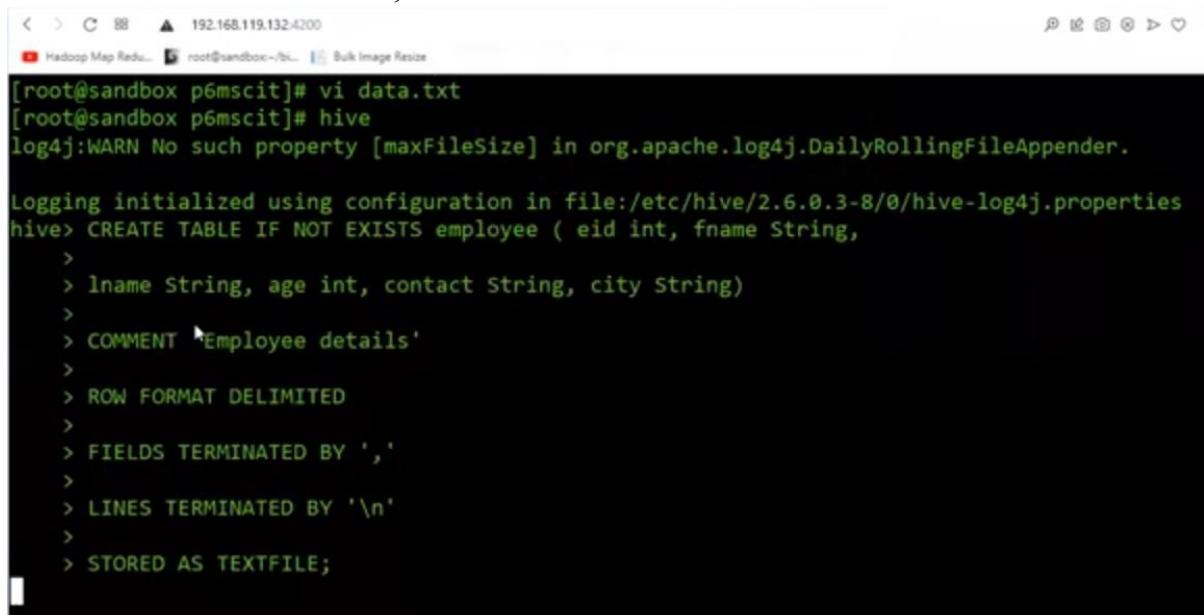


```
< > C 192.168.119.132:4200
[root@sandbox p6mscit]# vi data.txt
[root@sandbox p6mscit]# hive
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.0.3-8/0/hive-log4j.properties
hive>
```

Copy paste below command on hive and enter

```
CREATE TABLE IF NOT EXISTS employee ( eid int, fname String,
lname String, age int, contact String, city String)
COMMENT 'Employee details'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```



```
< > C 192.168.119.132:4200
[root@sandbox p6mscit]# vi data.txt
[root@sandbox p6mscit]# hive
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.0.3-8/0/hive-log4j.properties
hive> CREATE TABLE IF NOT EXISTS employee ( eid int, fname String,
> lname String, age int, contact String, city String)
> COMMENT 'Employee details'
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> LINES TERMINATED BY '\n'
> STORED AS TEXTFILE;
```

Run command: LOAD DATA LOCAL INPATH 'data.txt' OVERWRITE INTO

TABLE employee;

```
hive> LOAD DATA LOCAL INPATH 'data.txt' OVERWRITE INTO TABLE employee;
Loading data to table default.employee
Table default.employee stats: [numFiles=1, numRows=0, totalSize=339, rawDataSize=0]
OK
Time taken: 1.296 seconds
hive> ;
```

Run the command like select * from employee;

```
< > C 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:/bl... Bulk Image Resize
Loading data to table default.employee
Table default.employee stats: [numFiles=1, numRows=0, totalSize=339, rawDataSize=0]
OK
Time taken: 1.296 seconds
hive> select * from employee;
OK
1      Rajiv    Reddy   21      9848022337      Hyderabad
2      siddarth Battacharya 22      9848022338      Kolkata
3      Rajesh    Khanna   22      9848022339      Delhi
4      Preethi   Agarwal  21      9848022330      Pune
5      Trupthi   Mohanthy 23      9848022336      Bhuvaneshwar
6      Archana   Mishra   23      9848022335      Chennai
7      Komal     Nayak    24      9848022334      trivendram
8      Bharathi Nambiayar 24      9848022333      Chennai
Time taken: 0.188 seconds, Fetched: 8 row(s)
hive> select * from employee where age > 23;
OK
7      Komal     Nayak    24      9848022334      trivendram
8      Bharathi Nambiayar 24      9848022333      Chennai
Time taken: 0.551 seconds, Fetched: 2 row(s)
hive> ;
```

Practical 6

Implement an application that stores big data in Hbase/

Python What is HBase?

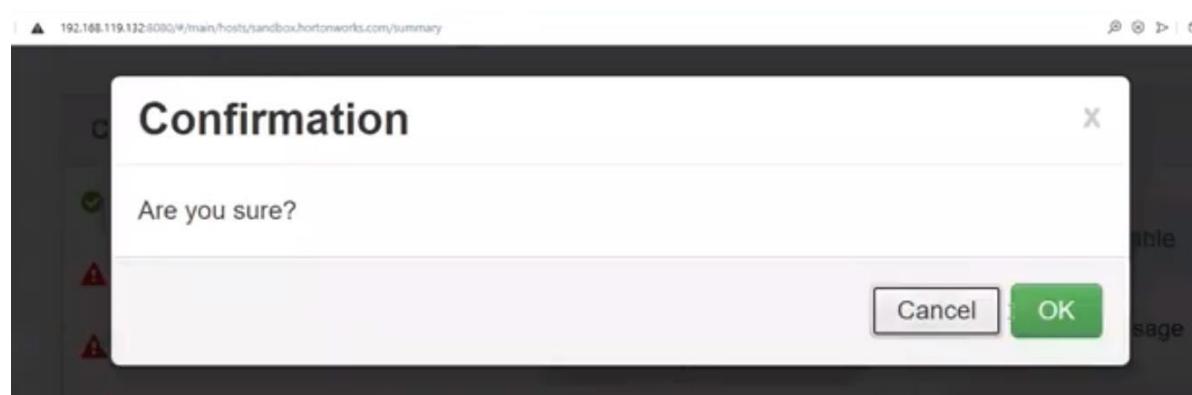
HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable. It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System. Go to GUI page and start the hbase service.

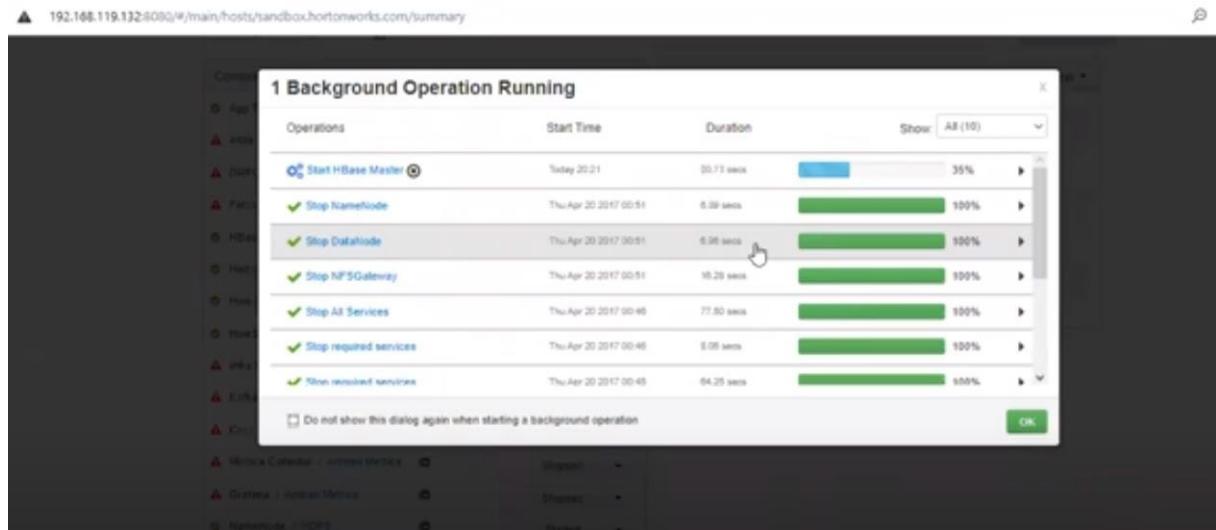
The screenshot shows the 'Components' section of the HDP 2.6.3 Cluster Summary UI. It lists several services with their current status:

Service	Status
App Timeline Server / YARN	Started
Atlas Metadata S... / Atlas	Stopped
DRPC Server / Storm	Stopped
Falcon Server / Falcon	Stopped
HBase Master / HBase	Stopped
History Server / MapReduce2	Started

A 'Start' button is visible for the HBase service, and it is currently being clicked, as indicated by the cursor icon. Below the table, there are two buttons: 'Turn Off Maintenance Mode' and another 'Start' button.

Click on OK to start the service.

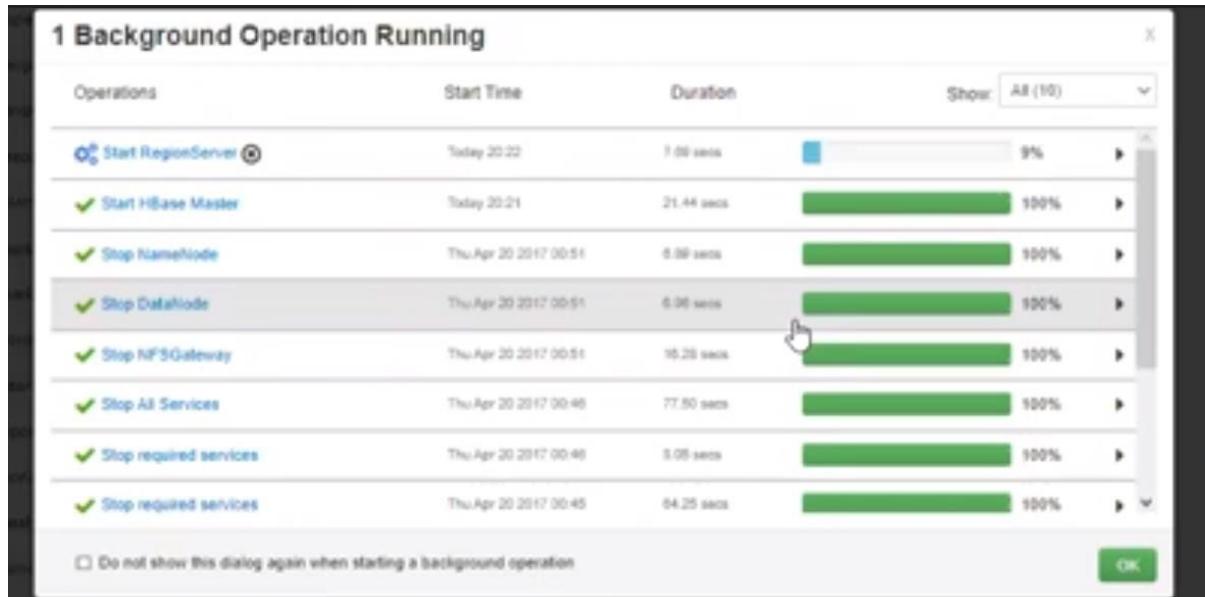




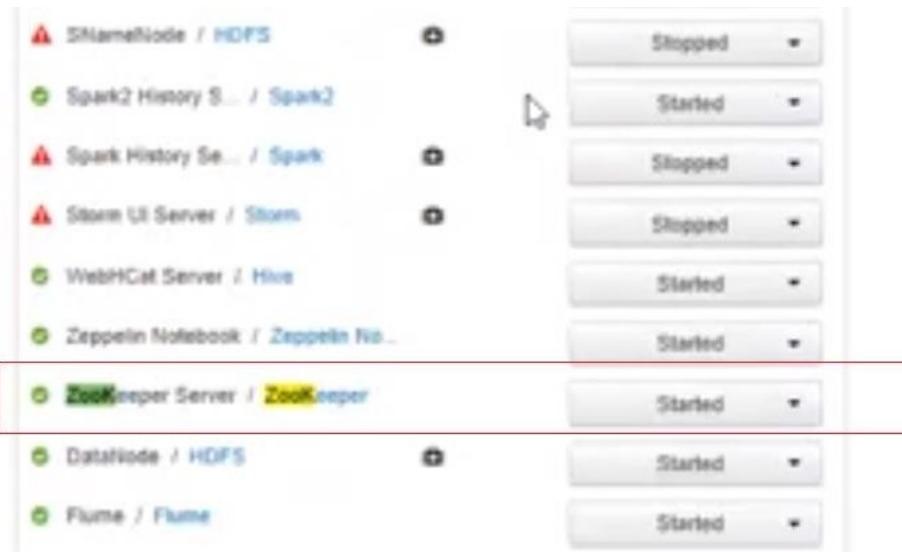
Now we must start region server.

The screenshot shows the Ambari Service Status page. The "RegionServer / HBase" service is highlighted with a red border and has a dropdown menu open over its status button. The menu options are: Start, Turn Off Maintenance Mode, and Go to... . The status button currently displays "Stopped".

Service	Status
ResourceManager / YARN	Started
NameNode / HDFS	Stopped
Spark2 History Server / Spark2	Started
Spark History Server / Spark	Stopped
Storm UI Server / Storm	Stopped
WebHCat Server / Hive	Started
Zeppelin Notebook / Zeppelin Notebooks	Started
ZooKeeper Server / ZooKeeper	Started
DataNode / HDFS	Started
Flume / Flume	Started
RegionServer / HBase	Stopped
Livy for Spark2 / Spark2	Started
Livy Server / Spark	Stopped
Metrics Monitor / Ambari Metrics	Stopped
NFSGateway / HDFS	Started



Check zookeeper server is started.



Check hbase and region server are started.

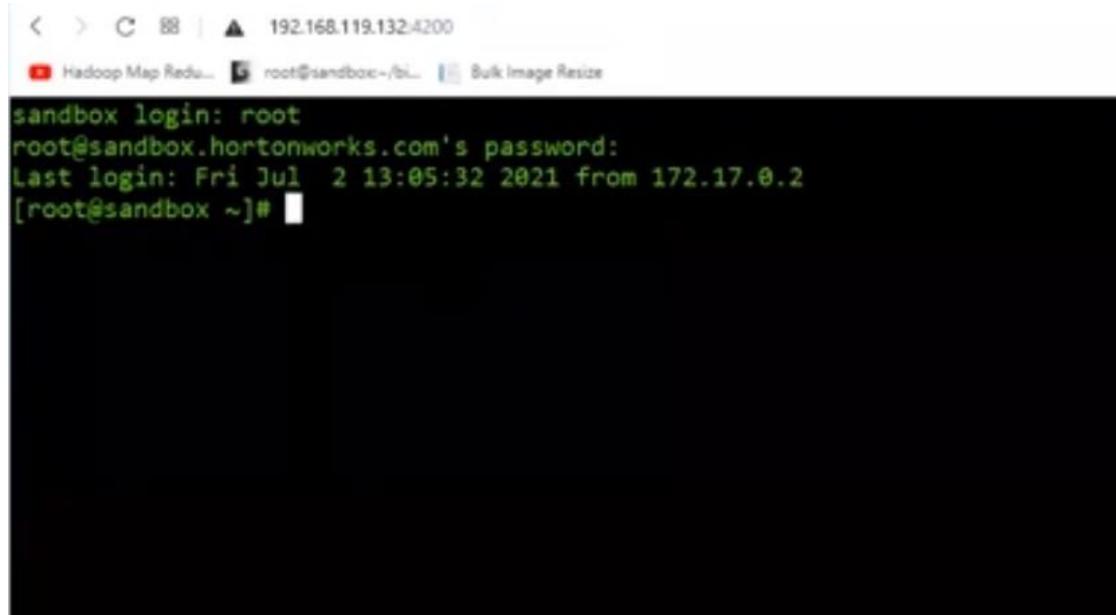
The screenshot shows the 'Components' section of the Hortonworks Data Platform UI. It lists the following components:

Component	Status
App Timeline Server / YARN	Started
Atlas Metadata Server / Atlas	Stopped
DRPC Server / Storm	Stopped
Falcon Server / Falcon	Stopped
HBase / HBase	Started
History Server / MapReduce2	Started
Hive Metastore / Hive	Started
WebHCat Server / Hive	Started
Zeppelin Notebook / Zeppelin Notebooks	Started
ZooKeeper Server / ZooKeeper	Started
DataNode / HDFS	Started
Flume / Flume	Started
RegionServer / HBase	Started
Livy for Spark2 / Spark2	Started
Livy Server / Spark	Stopped

Command: **whichapplication-name** gives directory in which application-name is installed.

Open the shell

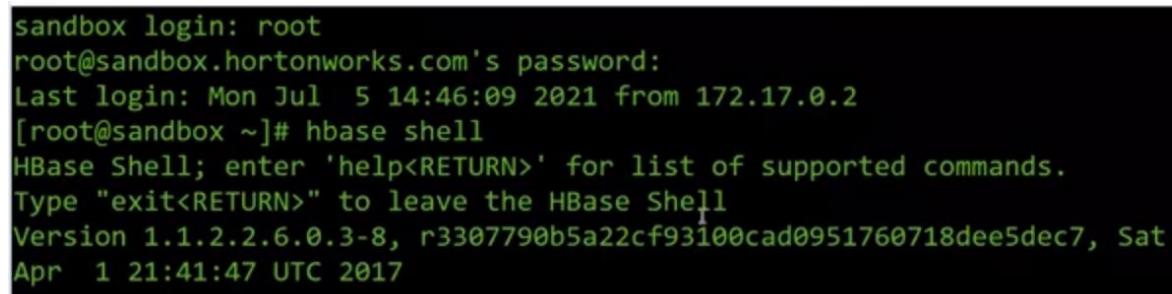
192.168.119.132:4200



The screenshot shows a terminal window with a black background and white text. At the top, there are several small icons: a left arrow, a right arrow, a circle, a square, a triangle pointing up, and a triangle pointing down. To the right of these icons, the IP address '192.168.119.132:4200' is displayed. Below the icons, there is a red status bar with the text 'Hadoop Map Redu...' and 'root@sandbox:/bi... Bulk Image Resize'. The main area of the terminal shows the following text:
sandbox login: root
root@ sandbox.hortonworks.com's password:
Last login: Fri Jul 2 13:05:32 2021 from 172.17.0.2
[root@sandbox ~]#

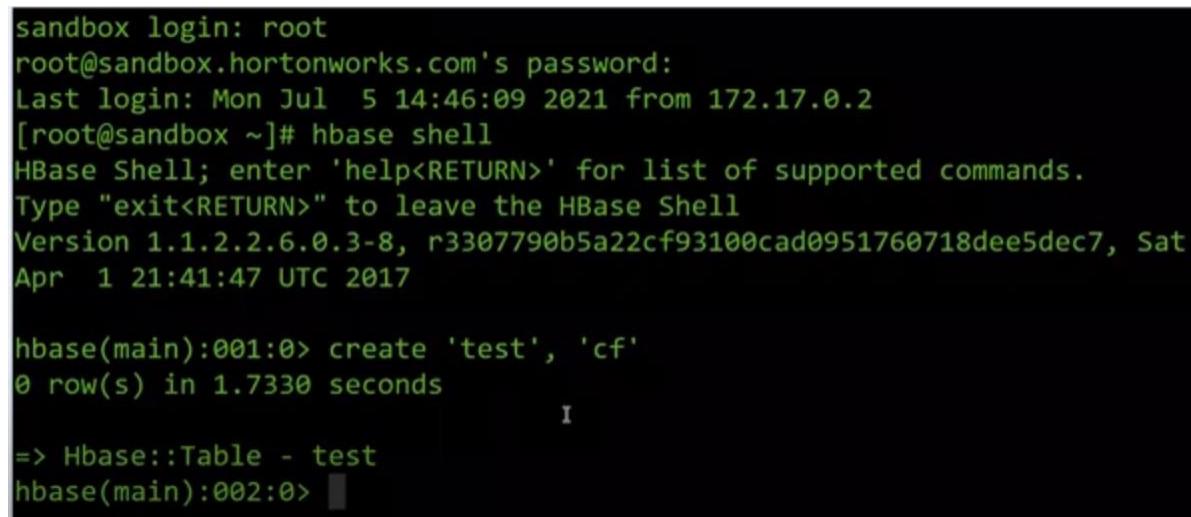
Command: hbase shell

It will start the server



The screenshot shows a terminal window with a black background and white text. The text is identical to the previous screenshot, except it includes the command 'hbase shell' and its output:
sandbox login: root
root@ sandbox.hortonworks.com's password:
Last login: Mon Jul 5 14:46:09 2021 from 172.17.0.2
[root@sandbox ~]# hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.0.3-8, r3307790b5a22cf93100cad0951760718dee5dec7, Sat
Apr 1 21:41:47 UTC 2017

Enter the command create 'test', 'cf' and it will create the table



The screenshot shows a terminal window with a black background and white text. The text is identical to the previous screenshots, but it includes the creation of a table:
sandbox login: root
root@ sandbox.hortonworks.com's password:
Last login: Mon Jul 5 14:46:09 2021 from 172.17.0.2
[root@sandbox ~]# hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.0.3-8, r3307790b5a22cf93100cad0951760718dee5dec7, Sat
Apr 1 21:41:47 UTC 2017

hbase(main):001:0> create 'test', 'cf'
0 row(s) in 1.7330 seconds
I
=> Hbase::Table - test
hbase(main):002:0>

Check the table is created with

command List- It will list all the tables created.

MSc IT Sem II

```
< > C 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bl... Bulk Image Resize
Apr 1 21:41:47 UTC 2017

hbase(main):001:0> create 'test', 'cf'
0 row(s) in 1.7330 seconds

=> Hbase::Table - test
hbase(main):002:0> list
TABLE           I
ATLAS_ENTITY_AUDIT_EVENTS
atlas_titan
iemployee
test
4 row(s) in 0.0740 seconds

=> ["ATLAS_ENTITY_AUDIT_EVENTS", "atlas_titan", "iemployee", "test"]
hbase(main):003:0>
```

If we want to see column description of a table.

Command- describe tablename

```
hbase(main):003:0> describe 'test'
Table test is ENABLED
test
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false',
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER',
COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true',
BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.1950 seconds

hbase(main):004:0>
```

Now, we have to put the values in table

Values:

put 'test', 'row1', 'cf:a', 'value1'

put 'test', 'row2', 'cf:b', 'value2'

put 'test', 'row3', 'cf:c', 'value3'

copy paste the data in shell.

MSc IT Sem II

```
< > C 88 | ▲ 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bl... Bulk Image Resize

'FOREVER', COMPRESSION => 'NONE', MIN VERSIONS => '0', BLO
rue', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.1950 seconds

hbase(main):004:0> put 'test', 'row1', 'cf:a', 'value1'
0 row(s) in 0.1930 seconds

hbase(main):005:0>
hbase(main):006:0* put 'test', 'row2', 'cf:b', 'value2'
0 row(s) in 0.0140 seconds

hbase(main):007:0>
hbase(main):008:0* put 'test', 'row3', 'cf:c', 'value3'
0 row(s) in 0.0340 seconds
```

We to display the records of table

Command: scan ‘test’

```
hbase(main):009:0> scan 'test'
ROW COLUMN+CELL
row1 column=cf:a, timestamp=1625496989589, value=value1
row2 column=cf:b, timestamp=1625496989697, value=value2
row3 column=cf:c, timestamp=1625496993087, value=value3
3 row(s) in 0.0620 seconds
```

Python: storage/retrieval

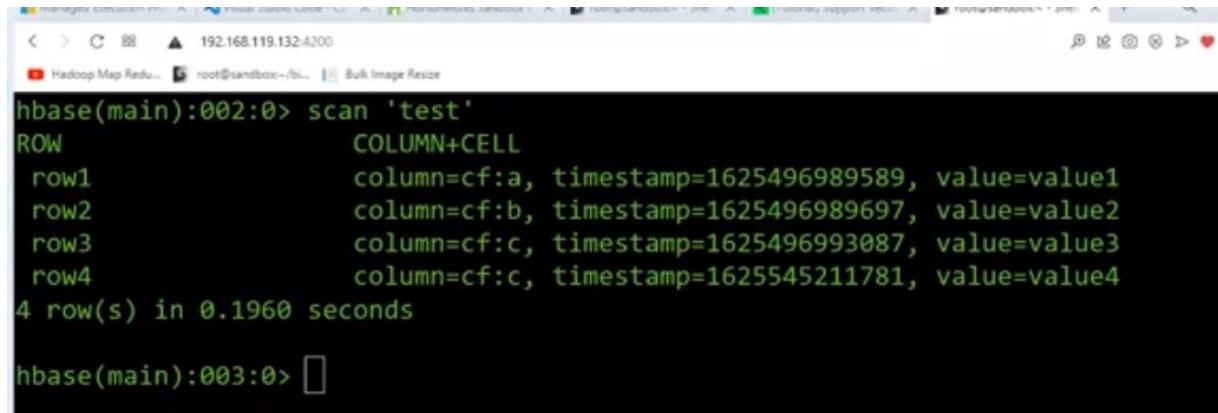
Start the service with command

Hbase thrift start -p 9090 –inforport 9095

```
< > C 88 | ▲ 192.168.119.132:4200
Hadoop Map Redu... root@sandbox:~/bl... Bulk Image Resize

sandbox login: root
root@ sandbox.hortonworks.com's password:
Last login: Tue Jul  6 13:22:05 2021 from 172.17.0.2
[root@sandbox ~]# hbase thrift start -p 9090 --infoport 9095
2021-07-06 14:52:38,870 INFO  [main] util.VersionInfo: HBase 1.1.2.2.6.0
.3-8
2021-07-06 14:52:38,873 INFO  [main] util.VersionInfo: Source code repos
itory git://c66-slave-ff632c10-5/grid/0/jenkins/workspace/HDP-parallel-c
entos6/SOURCES/hbase revision=3307790b5a22cf93100cad0951760718dee5dec7
2021-07-06 14:52:38,873 INFO  [main] util.VersionInfo: Compiled by jenki
ns on Sat Apr  1 21:41:47 UTC 2017
2021-07-06 14:52:38,873 INFO  [main] util.VersionInfo: From source with
checksum e816bb65a763f766331d511df40814e0
```

Create the table the way we did it in hbase and see the records using scan command



A screenshot of a terminal window titled "hbase(main):002:0>". The window shows the output of the command "scan 'test'". The output lists four rows: "row1", "row2", "row3", and "row4". Each row has three columns: "cf:a", "cf:b", and "cf:c". The values for "cf:a" are "value1", "value2", "value3", and "value4" respectively. The values for "cf:b" and "cf:c" are timestamped values. The total execution time is 0.1960 seconds. The window also shows the prompt "hbase(main):003:0>" at the bottom.

```
hbase(main):002:0> scan 'test'
ROW                                COLUMN+CELL
row1                               column=cf:a, timestamp=1625496989589, value=value1
row2                               column=cf:b, timestamp=1625496989697, value=value2
row3                               column=cf:c, timestamp=1625496993087, value=value3
row4                               column=cf:c, timestamp=1625545211781, value=value4
4 row(s) in 0.1960 seconds

hbase(main):003:0>
```

Create a program file

Import happybase as hb

```
conn=hb.connection('192.168.119.132', 9090)
```

```
print(conn.table('test').row('row1'))
```

```
print(conn.table('test').row('row2'))
```

```
print(conn.table('test').row('row3'))
```

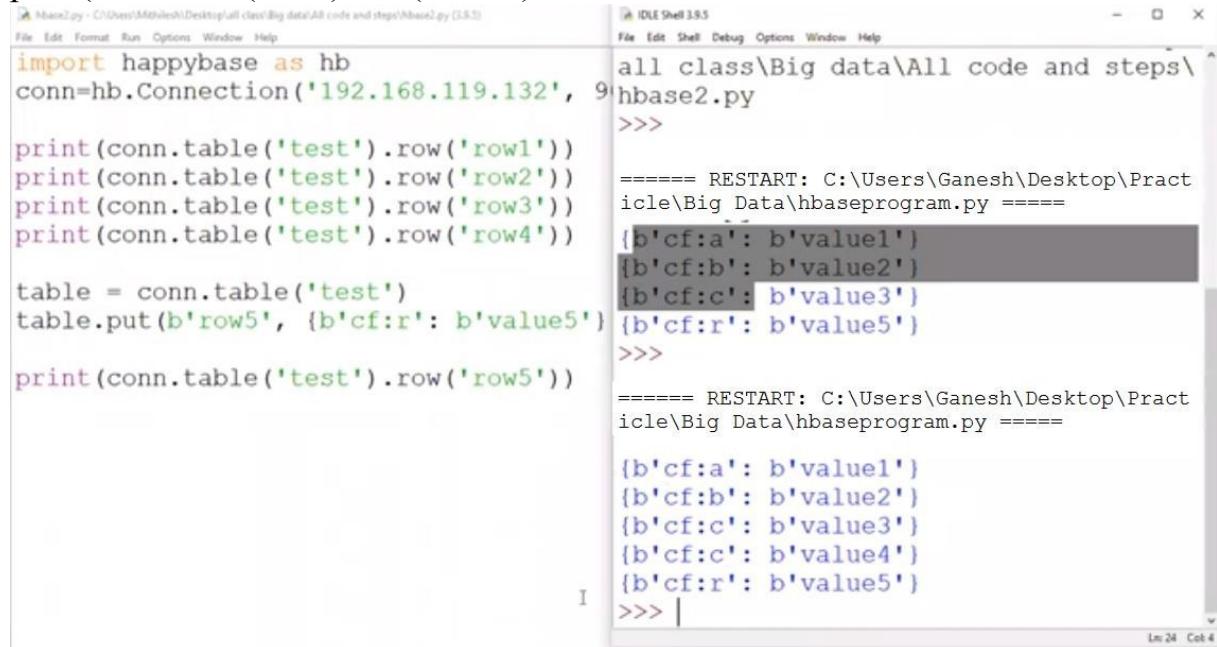
```
print(conn.table('test').row('row4'))
```

```
table = conn.table('test')
```

```
table.put(b'row5', {b'cf:r': b'value5'})
```

MSc IT Sem II

```
print(conn.table('test').row('row5'))
```



```
import happybase as hb
conn=hb.Connection('192.168.119.132', 9)

print(conn.table('test').row('row1'))
print(conn.table('test').row('row2'))
print(conn.table('test').row('row3'))
print(conn.table('test').row('row4'))

table = conn.table('test')
table.put(b'row5', {b'cf:r': b'value5'})

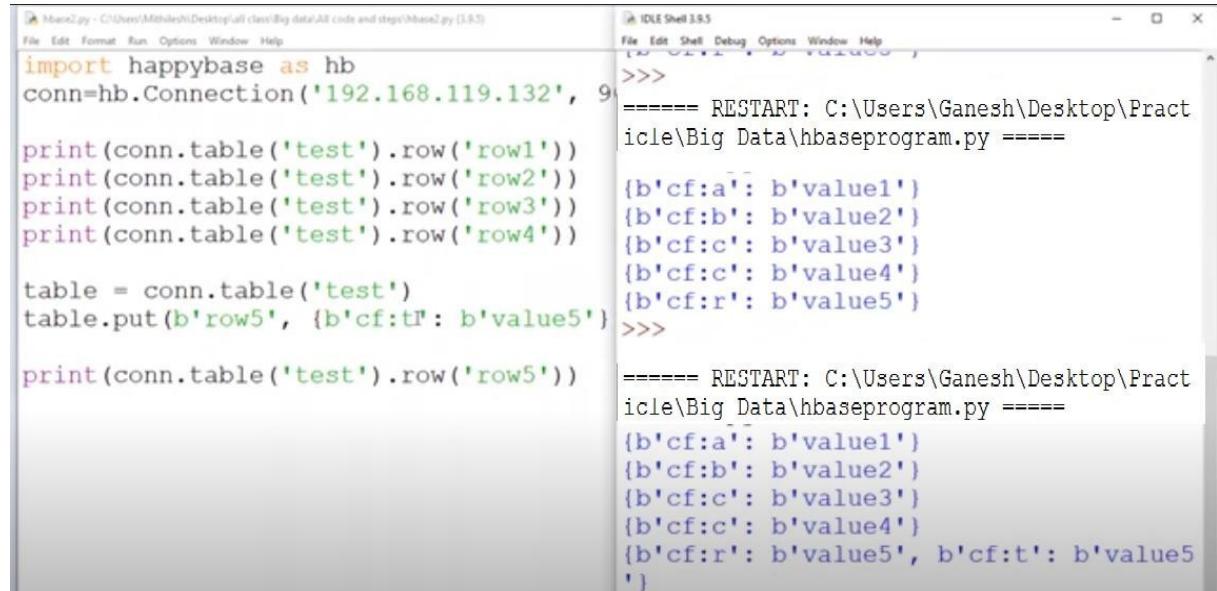
print(conn.table('test').row('row5'))
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
all class\Big data\All code and steps\hbase2.py
>>>
=====
RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\hbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:r': b'value5']
>>>
=====
RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\hbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5']
>>> |
```

Run a scan command on shell to display the values

```
hbase(main):004:0> scan 'test'
ROW           COLUMN+CELL
row1          column=cf:a, timestamp=1625496989589, value=value1
row2          column=cf:b, timestamp=1625496989697, value=value2
row3          column=cf:c, timestamp=1625496993087, value=value3
row4          column=cf:c, timestamp=1625545211781, value=value4
row5          column=cf:r, timestamp=1625583481042, value=value5
5 row(s) in 0.0320 seconds
```

Now, try with duplicate value at row 5 say value t



```
import happybase as hb
conn=hb.Connection('192.168.119.132', 9)

print(conn.table('test').row('row1'))
print(conn.table('test').row('row2'))
print(conn.table('test').row('row3'))
print(conn.table('test').row('row4'))

table = conn.table('test')
table.put(b'row5', {b'cf:t': b'value5'})

print(conn.table('test').row('row5'))
```

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
>>>
=====
RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\hbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5']
>>>
=====
RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\hbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5', b'cf:t': b'value5']
```

Run a scan command on shell to display the values

When there is unique value, it will create a record. If duplicate value it will not create a record

```
hbase(main):005:0> scan 'test'
ROW                                COLUMN+CELL
row1                               column=cf:a, timestamp=1625496989589, value=value1
row2                               column=cf:b, timestamp=1625496989697, value=value2
row3                               column=cf:c, timestamp=1625496993087, value=value3
row4                               column=cf:c, timestamp=1625545211781, value=value4
row5                               column=cf:r, timestamp=1625583481042, value=value5
row5                               column=cf:t, timestamp=1625583505297, value=value5
5 row(s) in 0.1320 seconds
```

Practical 7

Implement Decision tree classification techniques **Decision Trees (DTs)** are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Using the Iris dataset, we can construct a tree as follows:

+ Code + Text

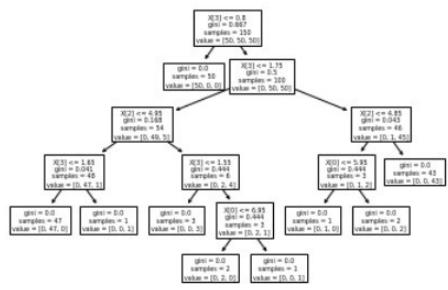


```
from sklearn.datasets import load_iris
from sklearn import tree
iris = load_iris()
X, y = iris.data, iris.target
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X, y)
```

Once trained, we can plot the tree with the **plot_tree** function:

[4] tree.plot_tree(clf)

```
[Text(167.4, 199.32, 'X[3] <= 0.8\n gini = 0.667\n samples = 150\nvalue = [50, 50, 50]'),
Text(141.15384615385, 163.0799999999998, 'gini = 0.0\nsamples = 50\nvalue = [50, 0, 0]'),
Text(193.15384615384616, 163.0799999999998, 'X[3] <= 1.75\n gini = 0.5\nsamples = 100\nvalue = [0, 50, 50]'),
Text(103.01538461538462, 126.83999999999999, 'X[2] <= 4.95\n gini = 0.168\nsamples = 54\nvalue = [0, 49, 5]'),
Text(51.50769230769231, 90.6, 'X[3] <= 1.65\n gini = 0.041\nsamples = 48\nvalue = [0, 47, 1]'),
Text(25.753846153846155, 54.35999999999985, 'gini = 0.0\nsamples = 47\nvalue = [0, 47, 0]'),
Text(77.26153846153846, 54.35999999999985, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(154.52307692307693, 90.6, 'X[3] <= 1.55\n gini = 0.444\nsamples = 6\nvalue = [0, 2, 4]'),
Text(128.76923076923077, 54.35999999999985, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
Text(180.27692307692308, 54.35999999999985, 'X[0] <= 6.95\n gini = 0.444\nsamples = 3\nvalue = [0, 2, 1]'),
Text(154.52307692307693, 18.11999999999976, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
Text(206.03076923076924, 18.11999999999976, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(283.2923076923077, 126.83999999999999, 'X[2] <= 4.85\n gini = 0.043\nsamples = 46\nvalue = [0, 1, 45]'),
Text(257.53846153846155, 90.6, 'X[0] <= 5.95\n gini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
Text(231.7846153846154, 54.35999999999985, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
Text(283.2923076923077, 54.35999999999985, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
Text(309.04615384615386, 90.6, 'gini = 0.0\nsamples = 43\nvalue = [0, 0, 43]')]
```



Practical 8

Implement SVM classification techniques

Support Vector Machines

Generally, Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes.

Loading data:

```
[1] #Import scikit-learn dataset library
    from sklearn import datasets

    #Load dataset
    cancer = datasets.load_breast_cancer()
```

Exploring data:

```
▶ # print the names of the 13 features
print("Features: ", cancer.feature_names)

# print the label type of cancer('malignant' 'benign')
print("Labels: ", cancer.target_names)

⇒ Features: ['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
Labels: ['malignant' 'benign']
```

Check the shape of the dataset using shape.

```
▶ # print data(feature)shape
cancer.data.shape

⇒ (569, 30)
```

Check top 5 records of the feature set.

```
[4] # print the cancer data features (top 5 records)
print(cancer.data[0:5])

[[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
 1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
 6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
 1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
 4.601e-01 1.189e-01]
[2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
 7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
 5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e+01
 2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.860e-01
 2.750e-01 8.902e-02]
[1.969e+01 2.125e+01 1.300e+02 1.203e+03 1.096e-01 1.599e-01 1.974e-01
 1.279e-01 2.069e-01 5.999e-02 7.456e-01 7.869e-01 4.585e+00 9.403e+01
 6.150e-03 4.006e-02 3.832e-02 2.058e-02 2.250e-02 4.571e-03 2.357e+01
 2.553e+01 1.525e+02 1.709e+03 1.444e-01 4.245e-01 4.504e-01 2.430e-01
 3.613e-01 8.758e-02]
[1.142e+01 2.038e+01 7.758e+01 3.861e+02 1.425e-01 2.839e-01 2.414e-01
 1.052e-01 2.597e-01 9.744e-02 4.956e-01 1.156e+00 3.445e+00 2.723e+01
 9.110e-03 7.458e-02 5.661e-02 1.867e-02 5.963e-02 9.208e-03 1.491e+01
 2.650e+01 9.887e+01 5.677e+02 2.098e-01 8.663e-01 6.869e-01 2.575e-01
 6.638e-01 1.730e-01]
[2.029e+01 1.434e+01 1.351e+02 1.297e+03 1.003e-01 1.328e-01 1.980e-01
 1.043e-01 1.809e-01 5.883e-02 7.572e-01 7.813e-01 5.438e+00 9.444e+01
 1.149e-02 2.461e-02 5.688e-02 1.885e-02 1.756e-02 5.115e-03 2.254e+01
 1.667e+01 1.522e+02 1.575e+03 1.374e-01 2.050e-01 4.000e-01 1.625e-01
 2.364e-01 7.678e-02]]
```

Target set:

Splitting Data:

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Split the dataset by using the function `train_test_split()`. you need to pass 3 parameters features, target, and `test_size`. Additionally, you can use `random_state` to select records randomly.

```
[7] # Import train_test_split function
from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target, test_size=0.3, random_state=109) # 70% training and 30% test
```

Generate Model:

Let's build support vector machine model. First, import the SVM module and create support vector classifier object by passing argument kernel as the linear kernel in SVC() function.

Then, fit your model on train set using fit() and perform prediction on the test set using predict().

```
[8] #Import svm model
from sklearn import svm

#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

#Train the model using the training sets
clf.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

Evaluating

the Model:

Let's estimate how accurately the classifier or model can predict the breast cancer of patients. Accuracy can be computed by comparing actual test set values and predicted values.



```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics

# Model Accuracy: how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.9649122807017544

Vidya Prasarak Mandal's
**B. N. BANDODKAR COLLEGE OF SCIENCE
(AUTONOMOUS), THANE.**

(Affiliated to University of Mumbai)

NAAC REACCREDITED 'A' GRADE
Best College Award, University of Mumbai

माहिती व तंत्रज्ञान विभाग

दूरध्वनी क. २५३३ ६५०७



**Department of
Information Technology**

Tel. No. 2533 6507

Email : itbnb@vpmthane.org

CERTIFICATE

This is to certify that

Shri / Kum. _____

of M. Sc. (Information Technology) Part I Semester - II has completed the required number of experiments (Total =) signed herein, in this laboratory during the year 2023 – 2024.

Seal

Incharge
Department of Information
Technology

Principal
B. N. Bandodkar College of Science,
Thane

External Examiner