LAB REPORT

Exercise-3

By

Ambuj Mishra

202116003

Big Data & Large-Scale Computing

DA-IICT, Gandhinagar

31-March-2022

Environment setup:

Before moving to the questions, we must first set up the hdfs environment to make all the nodes and resource managers running.

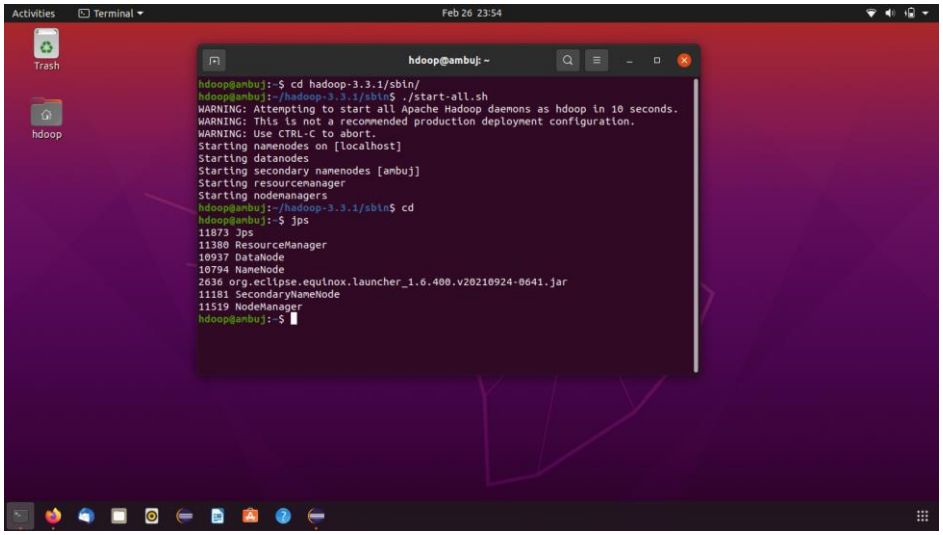


Figure 1: Starting the HADOOP environment

To run the java projects using .jar files on input data in hdfs file system, following commands are used:

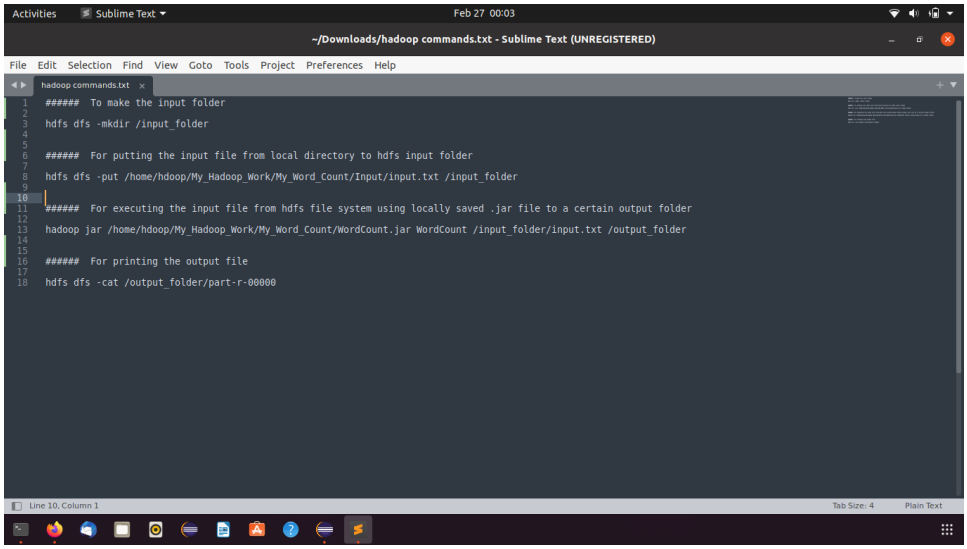


Figure 2: Commands to run a .jar file in hdfs file system

1st Question:

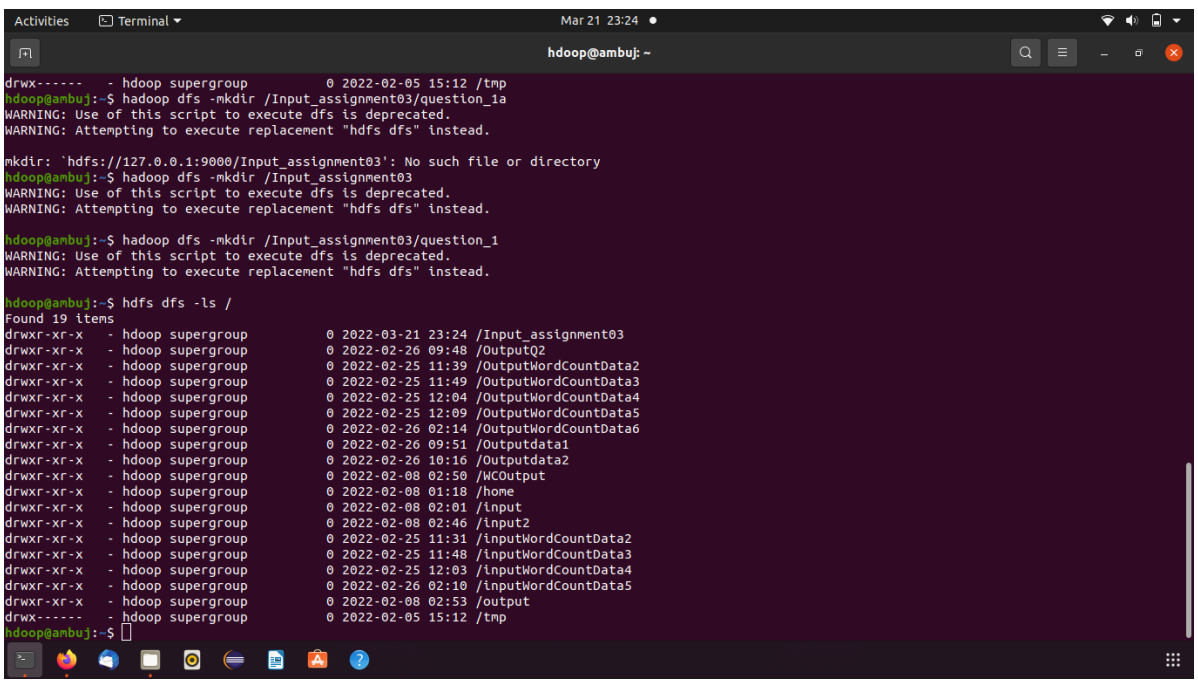
Part-(a)

Environment setup -

We have already set up the environment for Hadoop to work in pseudo-distributed mode. Now we'll run the following commands:

a.) mkdir -

First, we have listed the HDFS files and then we have created a new folder named "Input_assignment03" and inside this folder, we further created a subfolder 'question_1'.



```
drwx------   - hdoop supergroup          0 2022-02-05 15:12 /tmp
hdoop@ambuj:~$ hadoop dfs -mkdir /Input_assignment03/question_1a
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

mkdir: `hdfs://127.0.0.1:9000/Input_assignment03': No such file or directory
hdoop@ambuj:~$ hadoop dfs -mkdir /Input_assignment03
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

hdoop@ambuj:~$ hadoop dfs -mkdir /Input_assignment03/question_1
WARNING: Use of this script to execute dfs is deprecated.
WARNING: Attempting to execute replacement "hdfs dfs" instead.

hdoop@ambuj:~$ hdfs dfs -ls /
Found 19 items
drwxr-xr-x   - hdoop supergroup          0 2022-03-21 23:24 /Input_assignment03
drwxr-xr-x   - hdoop supergroup          0 2022-02-26 09:48 /OutputQ2
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 11:39 /OutputWordCountData2
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 11:49 /OutputWordCountData3
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 12:04 /OutputWordCountData4
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 12:09 /OutputWordCountData5
drwxr-xr-x   - hdoop supergroup          0 2022-02-26 02:14 /OutputWordCountData6
drwxr-xr-x   - hdoop supergroup          0 2022-02-26 09:51 /Outputdata1
drwxr-xr-x   - hdoop supergroup          0 2022-02-26 10:16 /Outputdata2
drwxr-xr-x   - hdoop supergroup          0 2022-02-08 02:50 /WCOutput
drwxr-xr-x   - hdoop supergroup          0 2022-02-08 01:18 /home
drwxr-xr-x   - hdoop supergroup          0 2022-02-08 02:01 /input
drwxr-xr-x   - hdoop supergroup          0 2022-02-08 02:46 /input2
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 11:31 /inputWordCountData2
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 11:48 /inputWordCountData3
drwxr-xr-x   - hdoop supergroup          0 2022-02-25 12:03 /inputWordCountData4
drwxr-xr-x   - hdoop supergroup          0 2022-02-26 02:10 /inputWordCountData5
drwxr-xr-x   - hdoop supergroup          0 2022-02-08 02:53 /output
drwx------   - hdoop supergroup          0 2022-02-05 15:12 /tmp
hdoop@ambuj:~$
```

Fig: Using mkdir to create folder structure

b.) copyFromLocal -

We have used 'copyFromLocal' command to put the file in HDFS from our local file system.

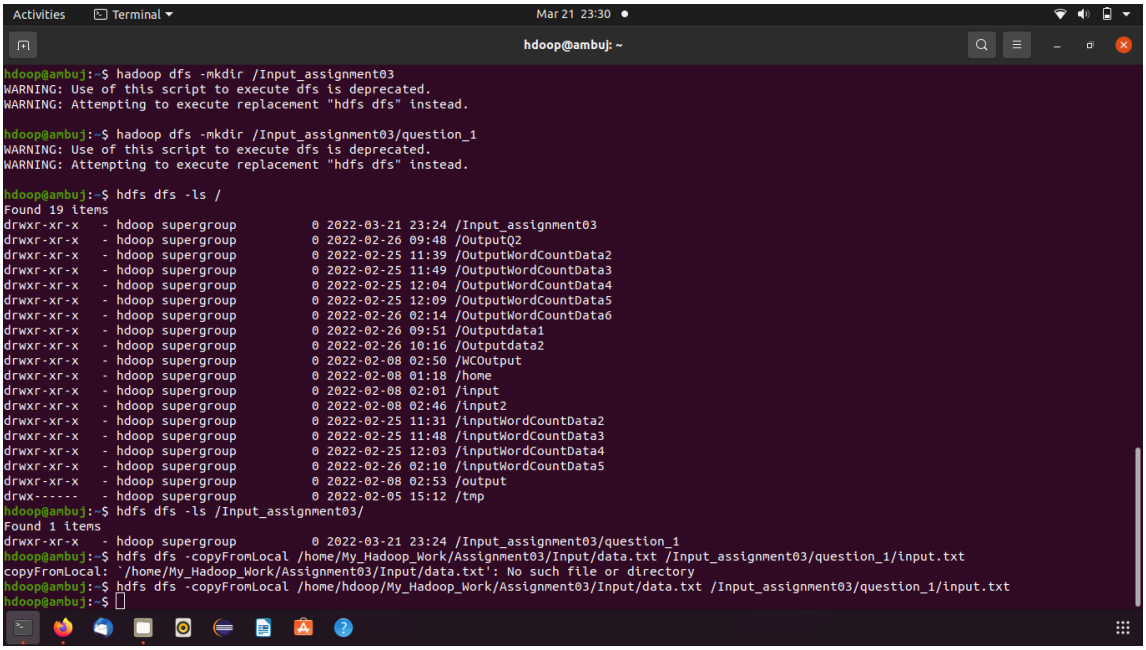


Fig: Using copyFromLocal to put the file in HDFS

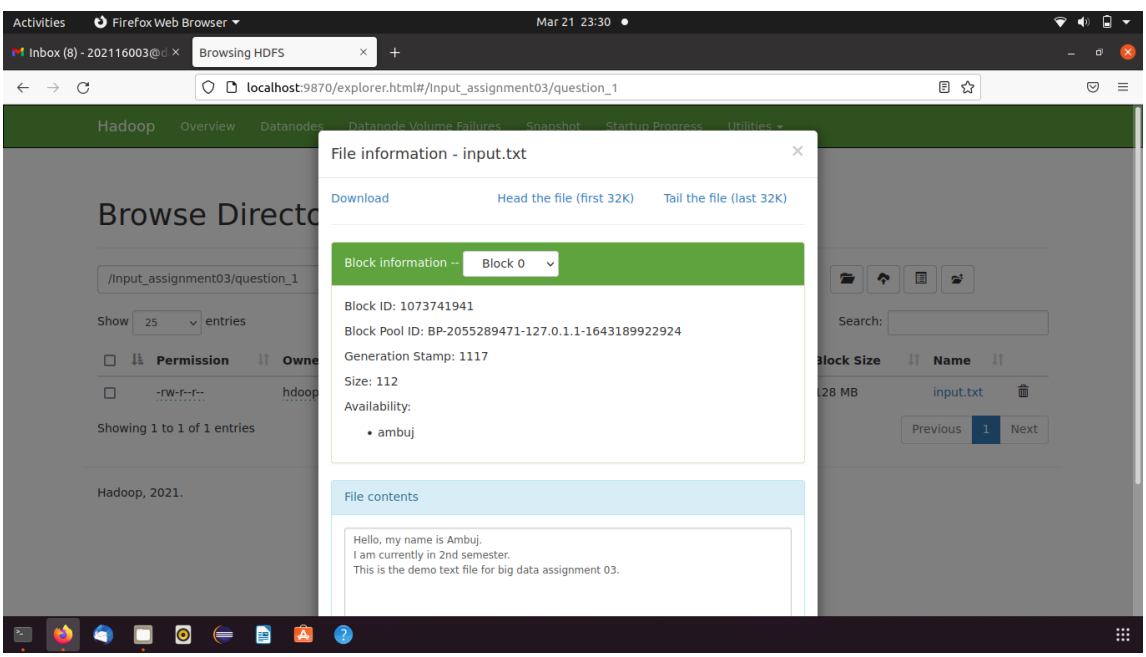


Fig: Checking the newly created file in HDFS

c.) copyToLocal -

We have used ‘copyToLocal’ command to put the file from HDFS from our local file system.

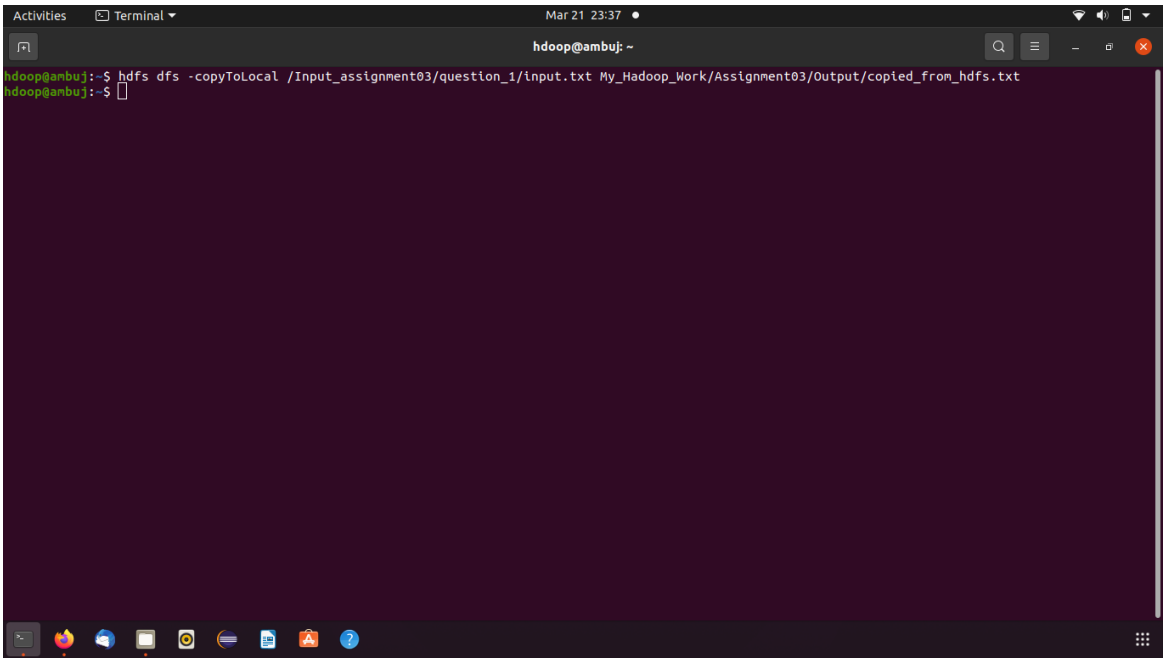


Fig: Using copyToLocal to put the file in local from HDFS

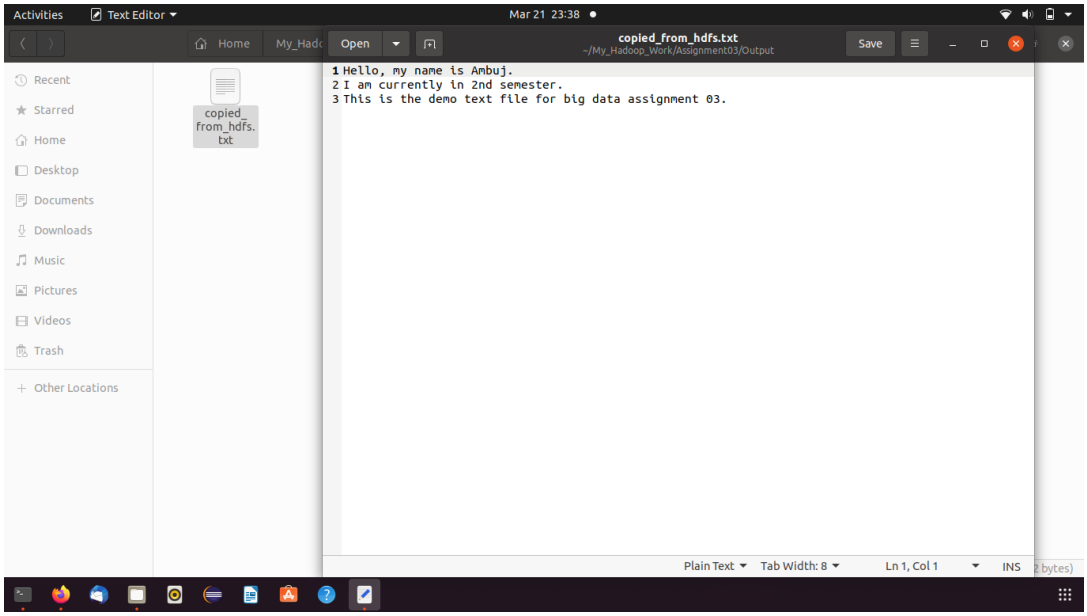


Fig: Checking the newly created file in local

d.) rm -

We have finally used ‘rm’ command to remove the ‘input.txt’ file from the question_1 folder.

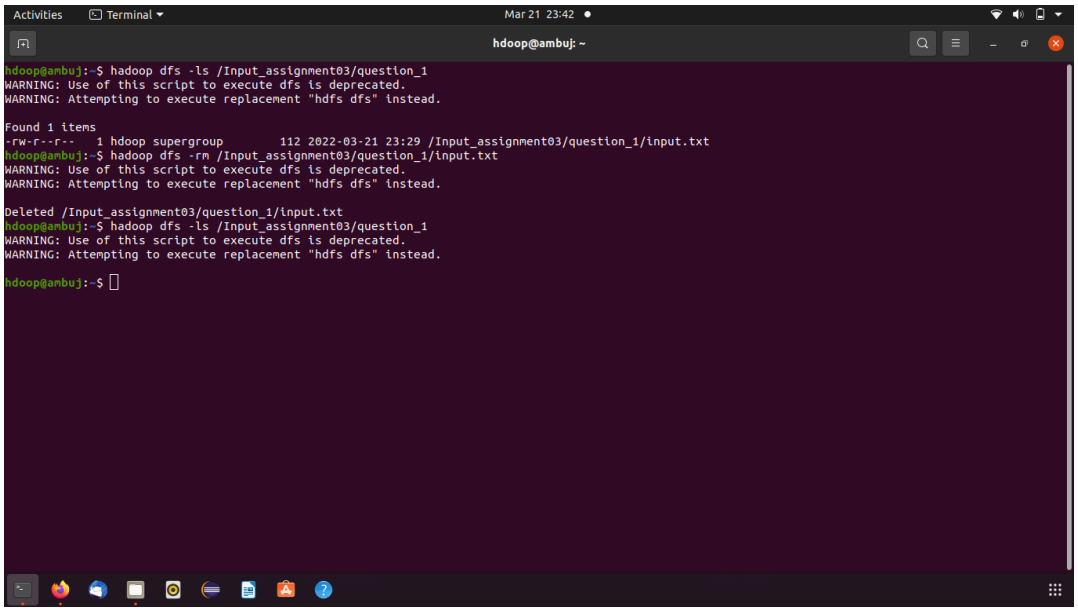


Fig: Deleting the file from HDFS using rm command

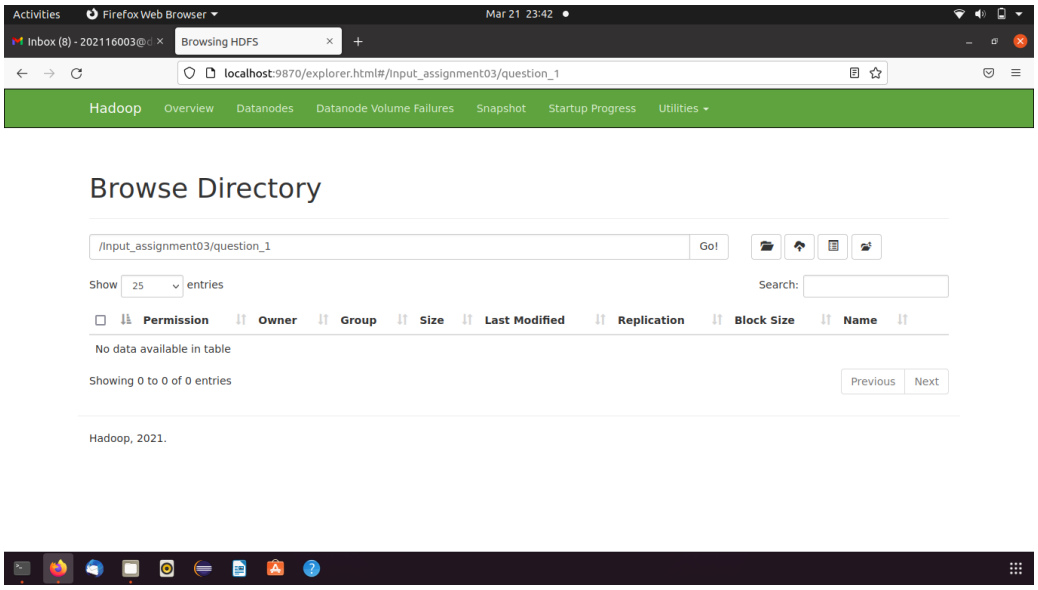


Fig: Confirming the results from interface

Part-(b)

After reading “Managing Files with the Hadoop File System Commands” in chapter 5 from Tom White’s book, we’ll be working on following 5 commands -

a.) ‘ls’

b.) ‘lsr’

c.) ‘put’

d.) ‘get’

e.) ‘mv’

a.) ‘ls’ -

‘ls’ command is used to list all the files and folders present in the path given.

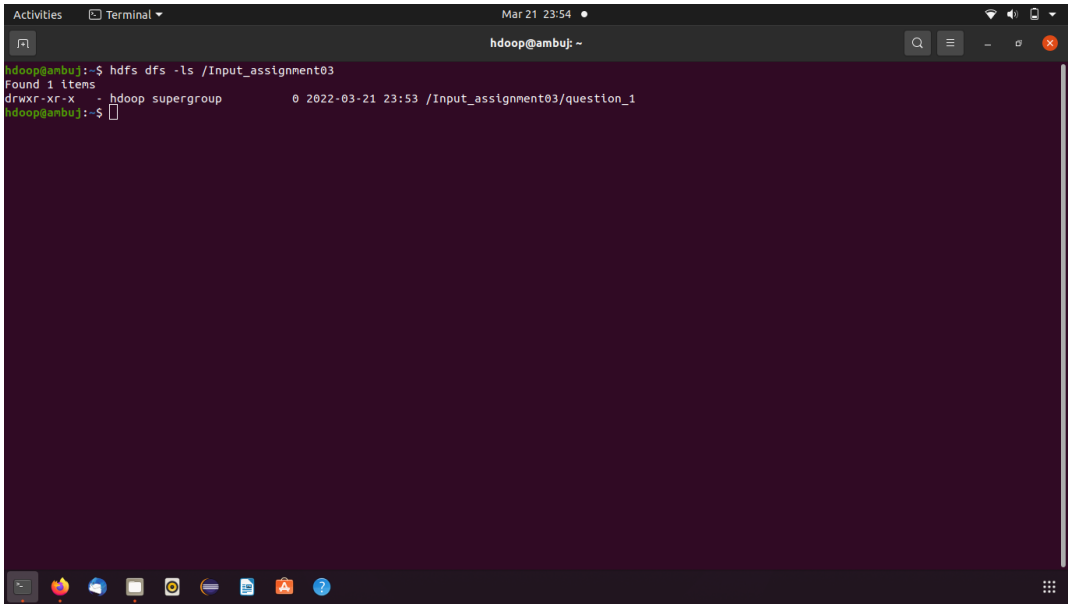


Fig: listing all the files and folders

b.) 'lsr' -

'lsr' command is used to ***recursively*** list all the files and folders present in the path given.

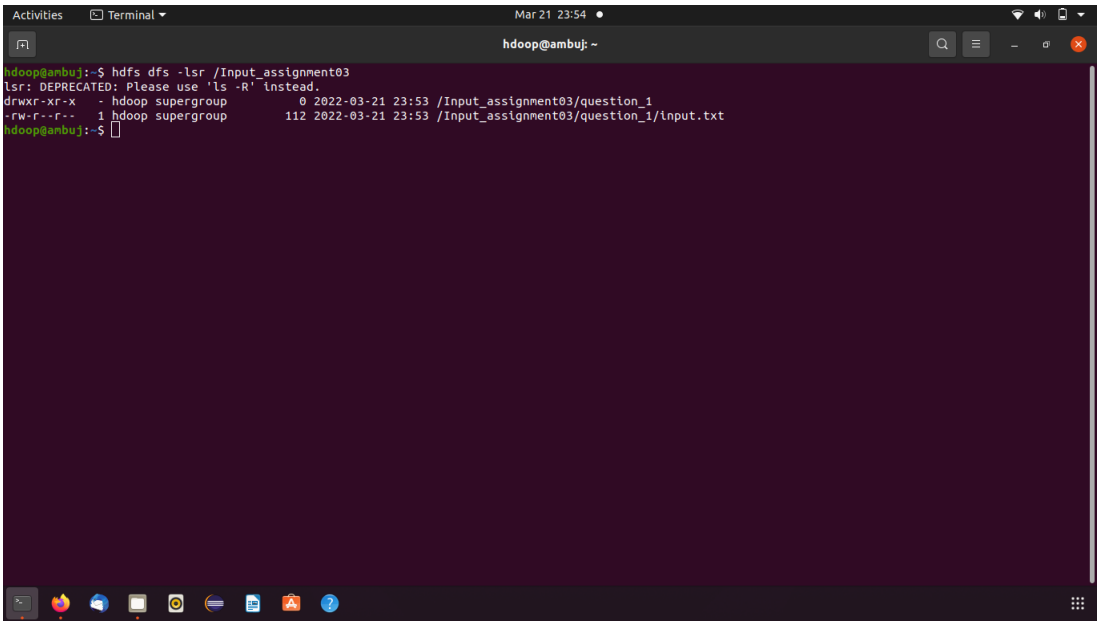


Fig: Recursively listing all the files and folders

c.) 'put' -

Similar to 'copyFromLocal' command, 'put' command is also used to transfer the file from local to HDFS.

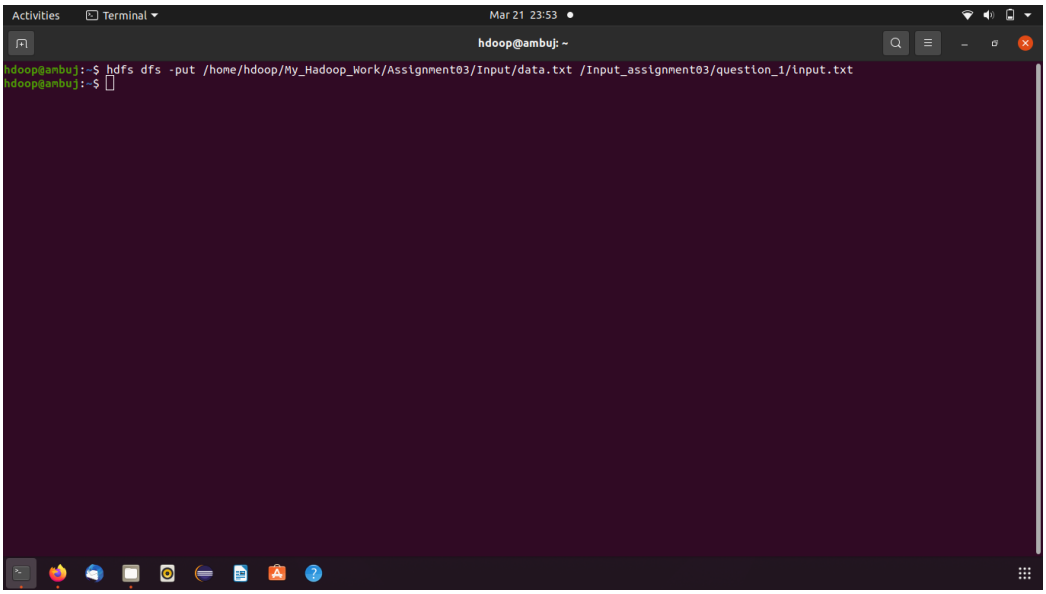


Fig: using 'put' command to transfer files from local to HDFS

d.) ‘get’ -

Similar to ‘copyToLocal’ command, ‘get’ command is also used to transfer the file from HDFS to local.

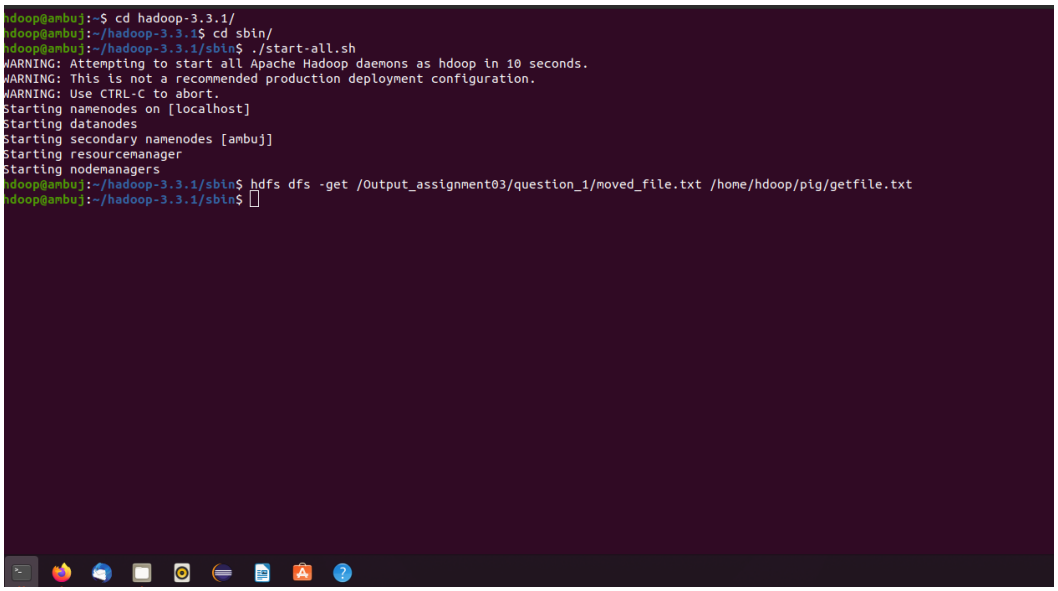


Fig: using ‘get’ command to transfer files from HDFS to local

e.) ‘mv’ -

‘mv’ command is used to move any given file from source path to destination path. In this case, the file gets deleted from the source path as well.

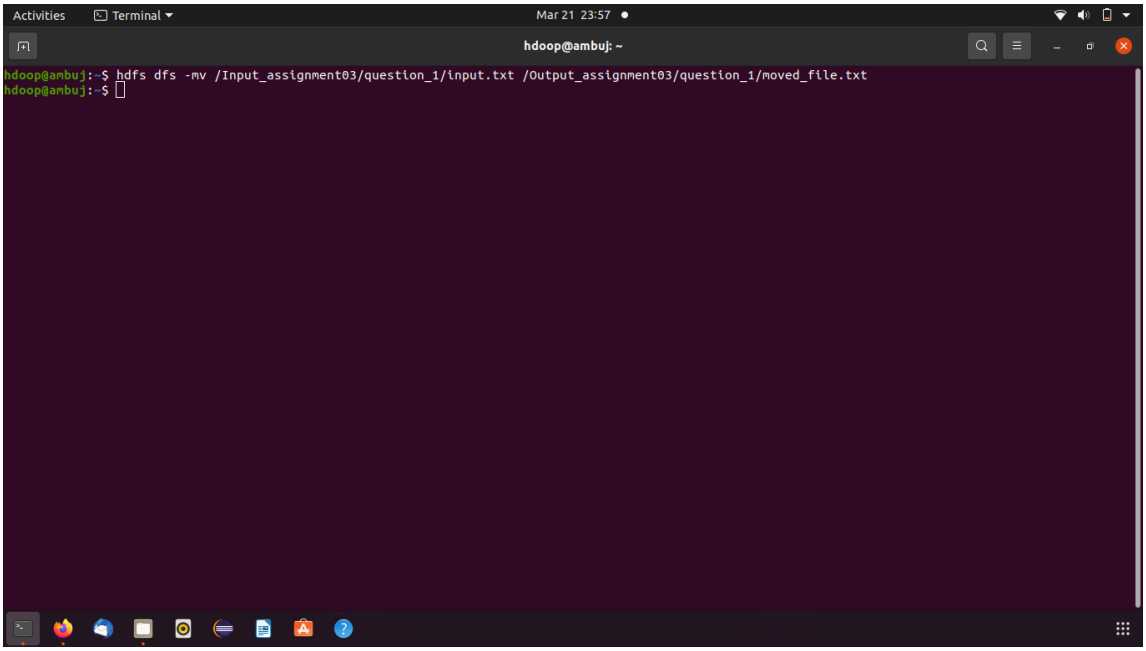


Fig: using ‘mv’ command to move file from one folder to another

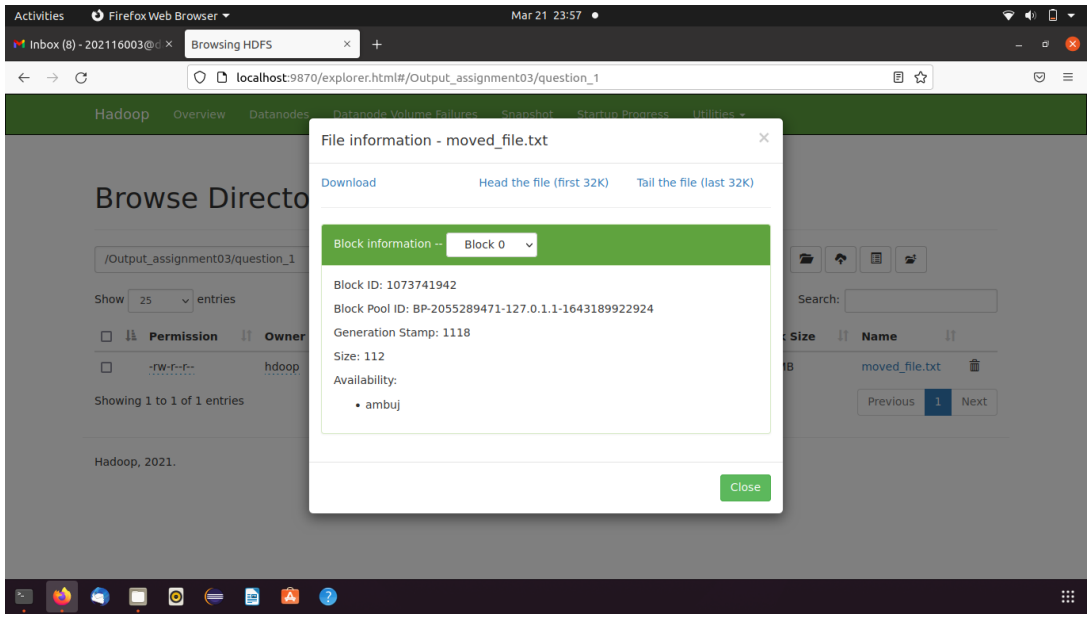


Fig: Cross-checking the newly created file at destination path

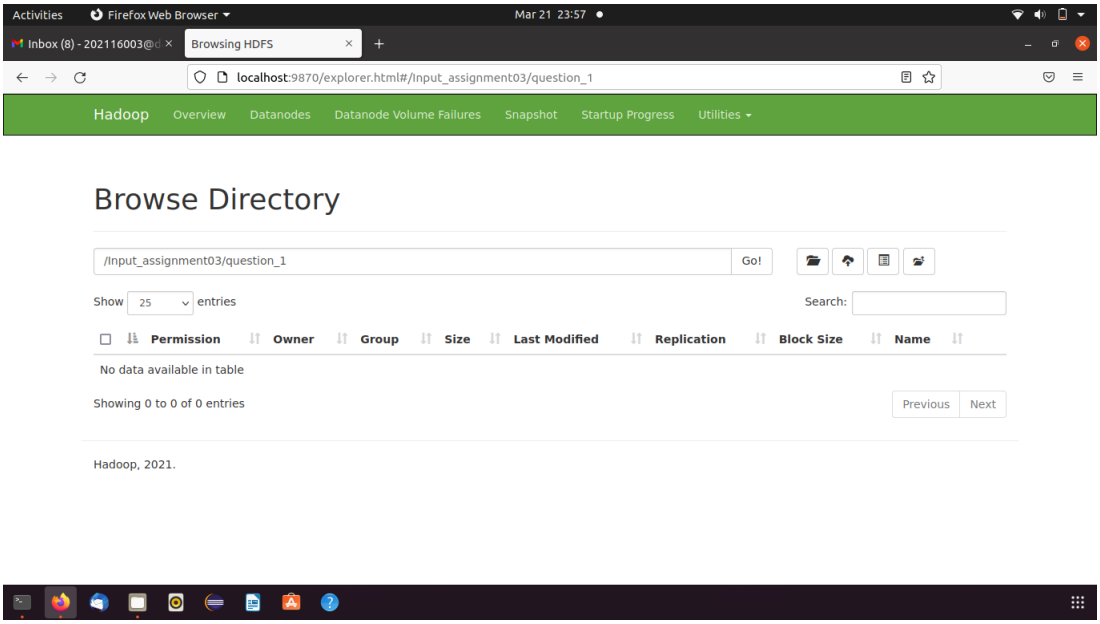
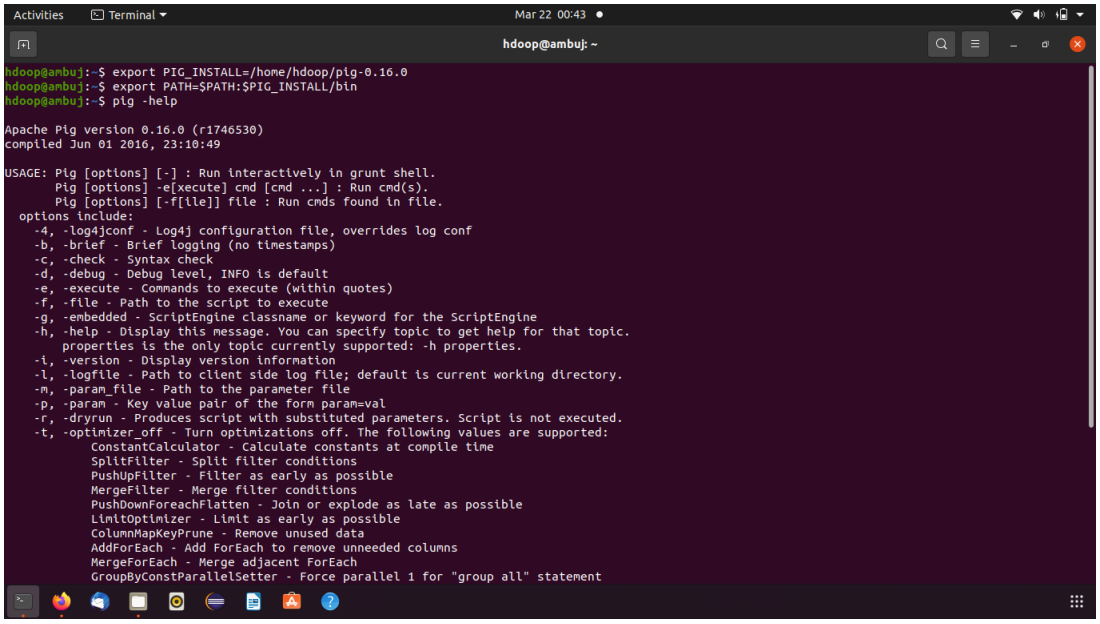


Fig: Cross-checking file deletion at source path

2nd Question:

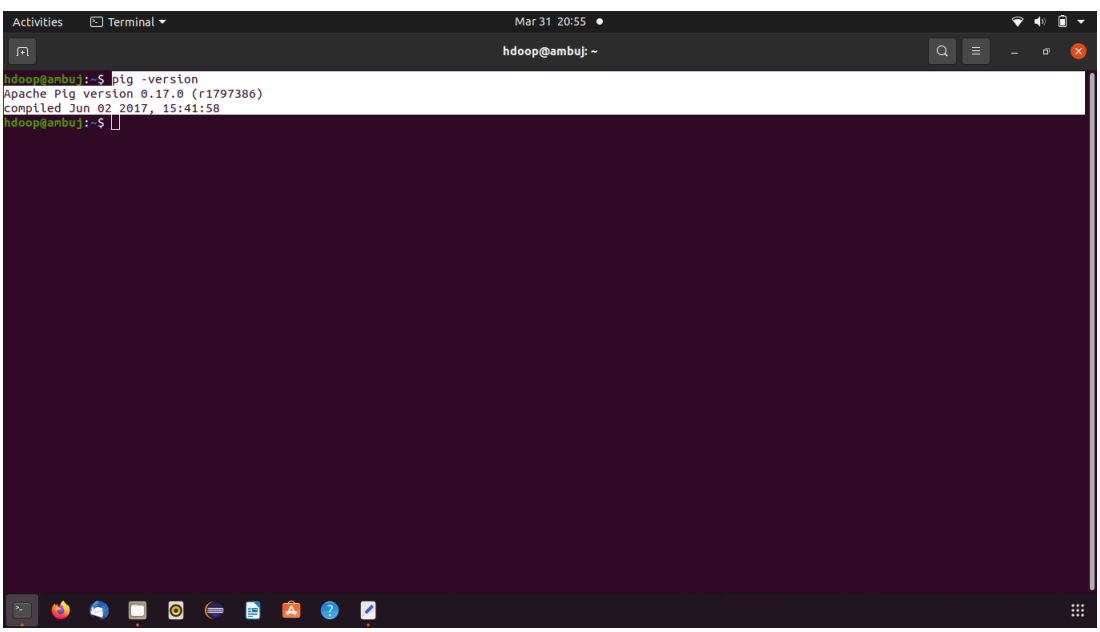
Pig setup:

After setting up the pig, we need to pass the paths to the setup-



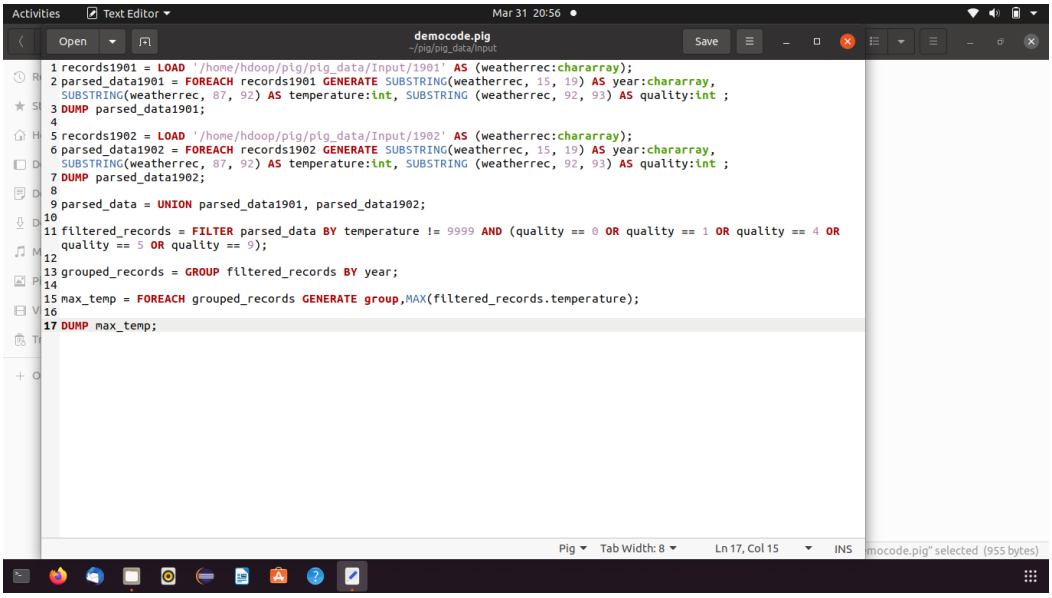
```
hdoop@ambuj:~$ export PIG_INSTALL=/home/hdoop/pig-0.16.0
hdoop@ambuj:~$ export PATH=$PATH:$PIG_INSTALL/bin
hdoop@ambuj:~$ pig -help

Apache Pig version 0.16.0 (r1746530)
compiled Jun 01 2016, 23:10:49

USAGE: Pig [options] [-] : Run interactively in grunt shell.
       Pig [options] -e[xecute] cmd [cmd ...] : Run cmd(s).
       Pig [options] [-f[ile]] file : Run cmds found in file.
  options include:
    -4, -log4jconf - Log4j configuration file, overrides log conf
    -b, -brief - Brief logging (no timestamps)
    -c, -check - Syntax check
    -d, -debug - Debug level, INFO is default
    -e, -execute - Commands to execute (within quotes)
    -f, -file - Path to the script to execute
    -g, -embedded - ScriptEngine classname or keyword for the ScriptEngine
    -h, -help - Display this message. You can specify topic to get help for that topic.
        properties is the only topic currently supported: -h properties.
    -i, -version - Display version information
    -l, -logfile - Path to client side log file; default is current working directory.
    -m, -param_file - Path to the parameter file
    -p, -param - Key value pair of the form param=val
    -r, -dryrun - Produces script with substituted parameters. Script is not executed.
    -t, -optimizer_off - Turn optimizations off. The following values are supported:
            ConstantCalculator - Calculate constants at compile time
            SplitFilter - Split filter conditions
            PushUpFilter - Filter as early as possible
            MergeFilter - Merge filter conditions
            PushDownForeachFlatten - Join or explode as late as possible
            LimitOptimizer - Limit as early as possible
            ColumnMapKeyPrune - Remove unused data
            AddForEach - Add ForEach to remove unneeded columns
            MergeForEach - Merge adjacent ForEach
            GroupByConstParallelSetter - Force parallel 1 for "group all" statement
```

Fig: setting up the paths for the pig



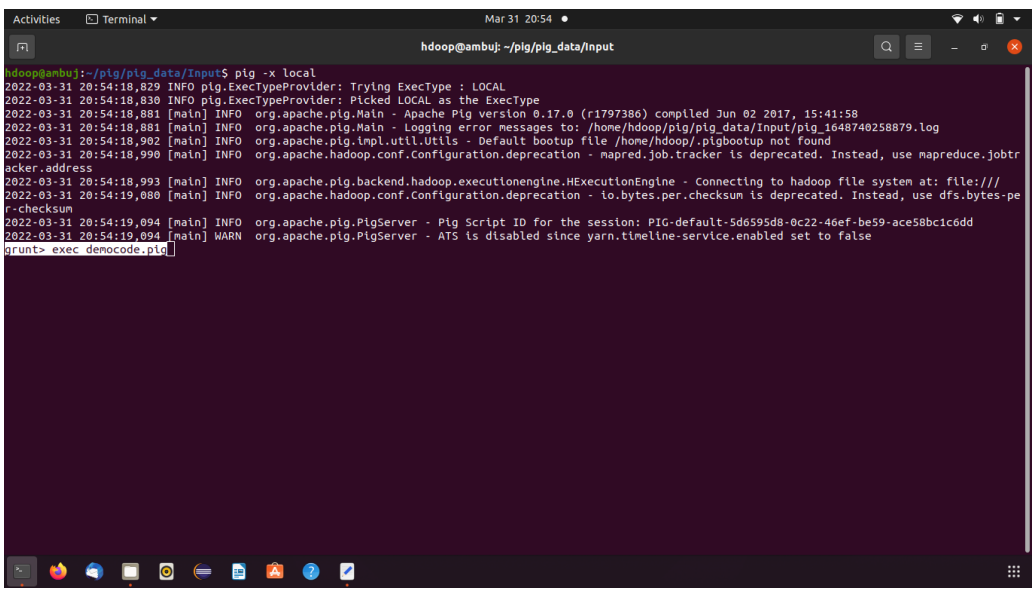
```
hdoop@ambuj:~$ pig -version
Apache Pig version 0.17.0 (r1797386)
compiled Jun 02 2017, 15:41:58
hdoop@ambuj:~$
```

Fig: Checking the pig setup



```
records1901 = LOAD '/home/hdoop/pig/pig_data/Input/1901' AS (weatherrec:chararray);
parsed_data1901 = FOREACH records1901 GENERATE SUBSTRING(weatherrec, 15, 19) AS year:chararray,
 SUBSTRING(weatherrec, 87, 92) AS temperature:int, SUBSTRING (weatherrec, 92, 93) AS quality:int ;
DUMP parsed_data1901;

records1902 = LOAD '/home/hdoop/pig/pig_data/Input/1902' AS (weatherrec:chararray);
parsed_data1902 = FOREACH records1902 GENERATE SUBSTRING(weatherrec, 15, 19) AS year:chararray,
 SUBSTRING(weatherrec, 87, 92) AS temperature:int, SUBSTRING (weatherrec, 92, 93) AS quality:int ;
DUMP parsed_data1902;

parsed_data = UNION parsed_data1901, parsed_data1902;

filtered_records = FILTER parsed_data BY temperature != 9999 AND (quality == 0 OR quality == 1 OR quality == 4 OR
 quality == 5 OR quality == 9);

grouped_records = GROUP filtered_records BY year;

max_temp = FOREACH grouped_records GENERATE group,MAX(filtered_records.temperature);

DUMP max_temp;
```

Fig: Code to work on the 1901 and 1902 files



```
hdoop@ambuj:~/pig/pig_data/Input$ pig -x local
2022-03-31 20:54:18,829 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2022-03-31 20:54:18,830 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2022-03-31 20:54:18,881 [main] INFO  org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2022-03-31 20:54:18,881 [main] INFO  org.apache.pig.Main - Logging error messages to: /home/hdoop/pig/pig_data/Input/pig_1648740258879.log
2022-03-31 20:54:18,902 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/hdoop/.pigbootup not found
2022-03-31 20:54:18,990 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtr
acker.address
2022-03-31 20:54:18,993 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: file:///
2022-03-31 20:54:19,080 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-pe
r-checksum
2022-03-31 20:54:19,094 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-5d6595d8-0c22-46ef-be59-ace58bc1c6dd
2022-03-31 20:54:19,094 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt> exec democode.pig
```

Fig: Starting the pig locally and executing the code file

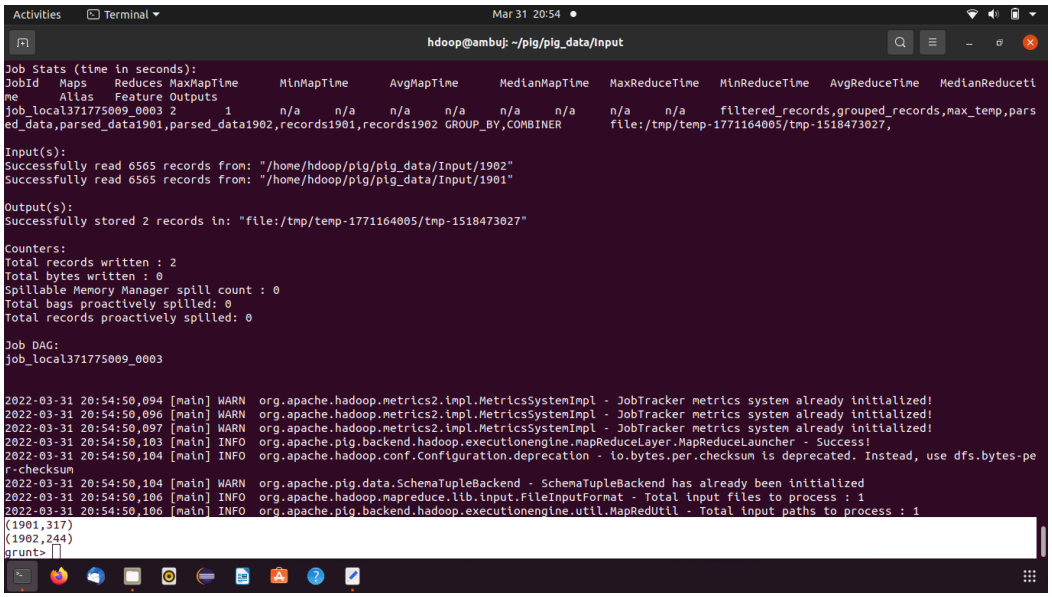


Fig: Final results

<u>Short Note on pig</u> -

Apache pig was performed to analyze larger sets of data representing them as data flows. Map–reduce algorithm takes a lot of time to perform map and reduce operations. Thus, we can say that apache Pig is an abstraction over MapReduce. Pig has 2 main components – pig latin and execution environment. Pig is built on top of hadoop. It has below advantages over map-reduce algorithm -

- Less development time
- Easy to learn
- Procedural language
- Dataflow
- Easy to control execution

Similarly, Apache Pig has the following disadvantages as well:

- Errors of Pig
- Not mature
- Support
- Minor one
- Implicit data schema
