Experiment 6 - To implement Word Count problem using Pig

What is Apache Pig

Apache Pig is a high-level data flow platform for executing MapReduce programs of Hadoop. The language used for Pig is Pig Latin.

The Pig scripts get internally converted to Map Reduce jobs and get executed on data stored in HDFS. Apart from that, Pig can also execute its job in Apache Tez or Apache Spark.

Pig can handle any type of data, i.e., structured, semi-structured or unstructured and stores the corresponding results into Hadoop Data File System. Every task which can be achieved using PIG can also be achieved using java used in MapReduce.

Features of Apache Pig

The various uses of Pig technology.

1) Ease of programming

Writing complex java programs for map reduce is quite tough for non-programmers. Pig makes this process easy. In the Pig, the queries are converted to MapReduce internally.

2) Optimization opportunities

It is how tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.

3) Extensibility

A user-defined function is written in which the user can write their logic to execute over the data set.

4) Flexible

It can easily handle structured as well as unstructured data.

5) In-built operators

It contains various type of operators such as sort, filter and joins.

Advantages of Apache Pig

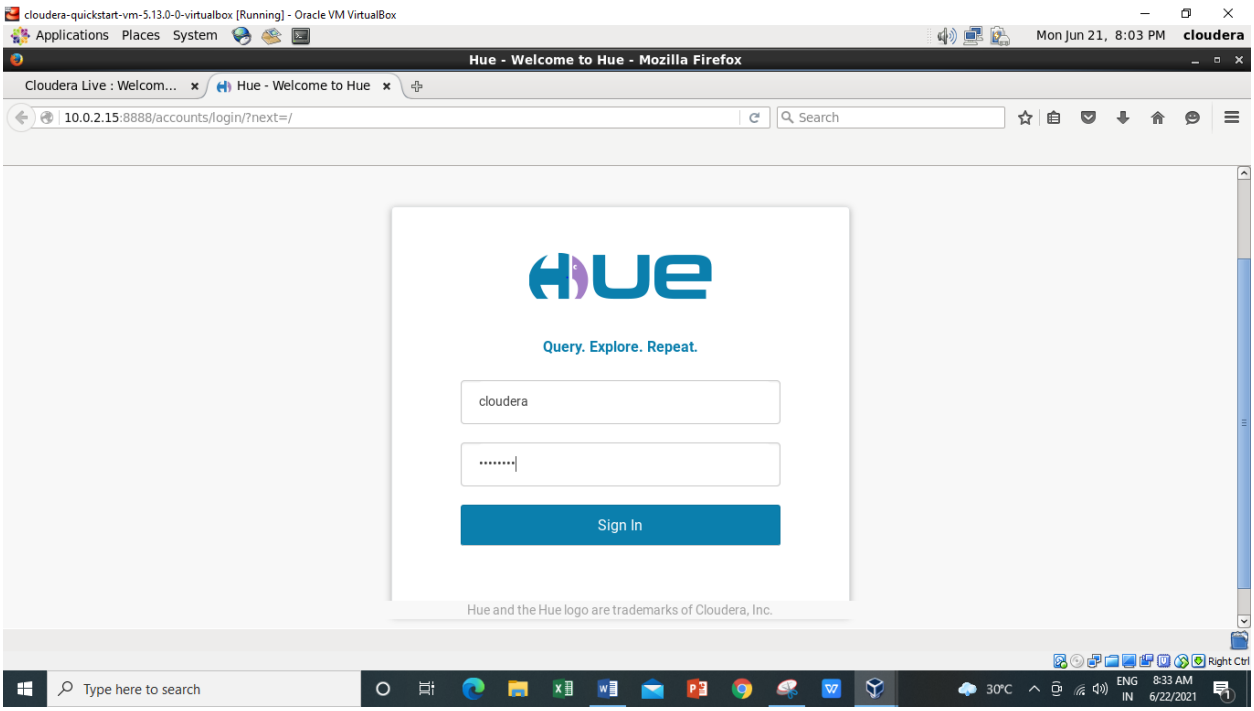
- Less code - The Pig consumes less line of code to perform any operation.
- Reusability - The Pig code is flexible enough to reuse again.
- Nested data types - The Pig provides a useful concept of nested data types like tuple, bag, and map.

To implement Word Count problem using Pig

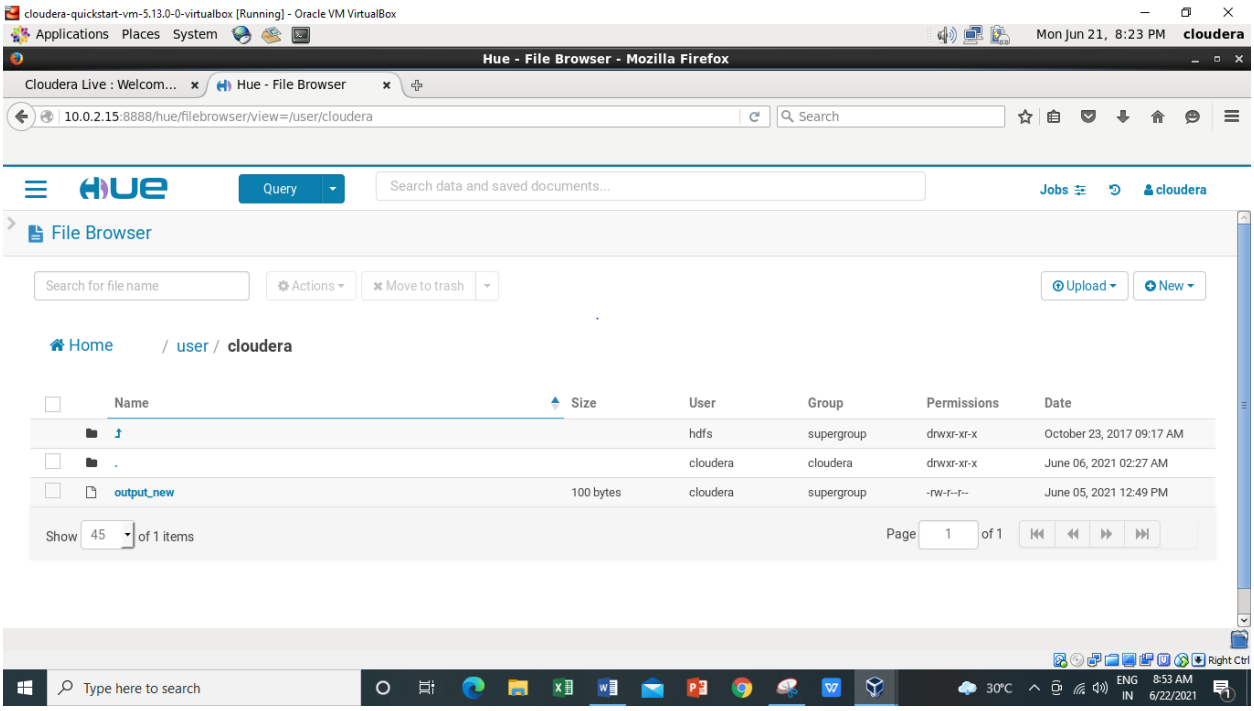
Steps:

1) Start the cloudera.

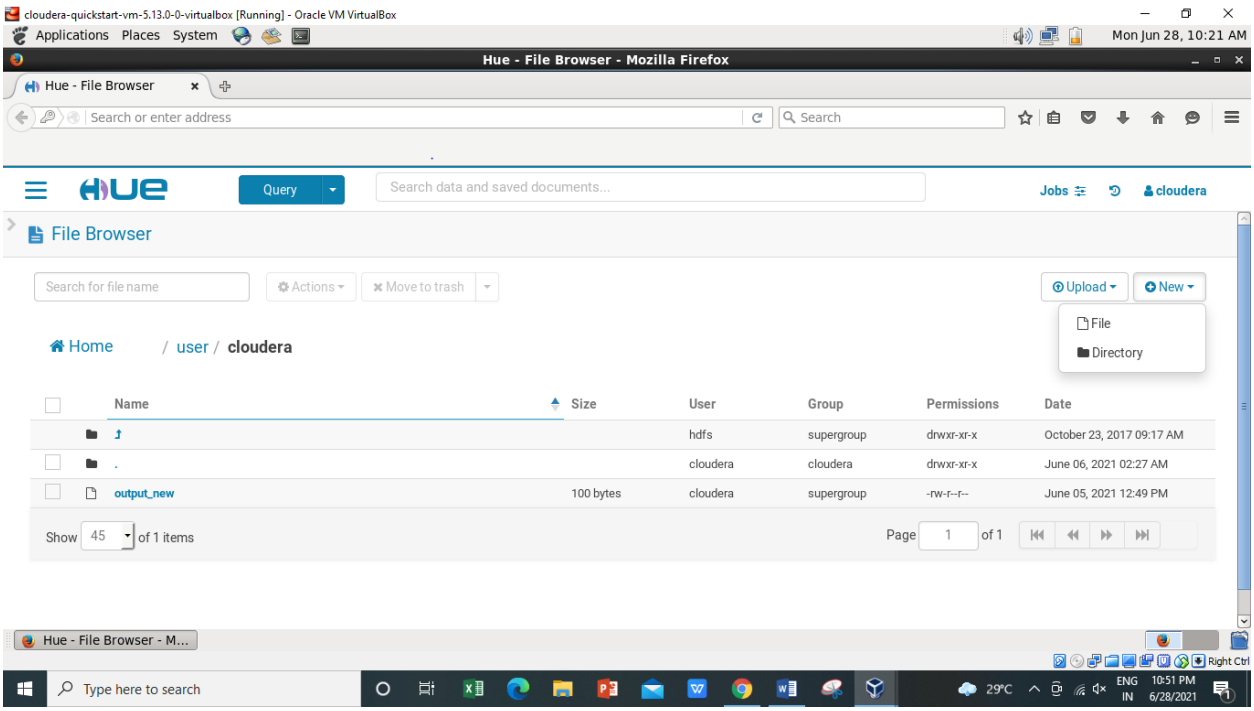
2) Open the browser. And then open Hue and login.

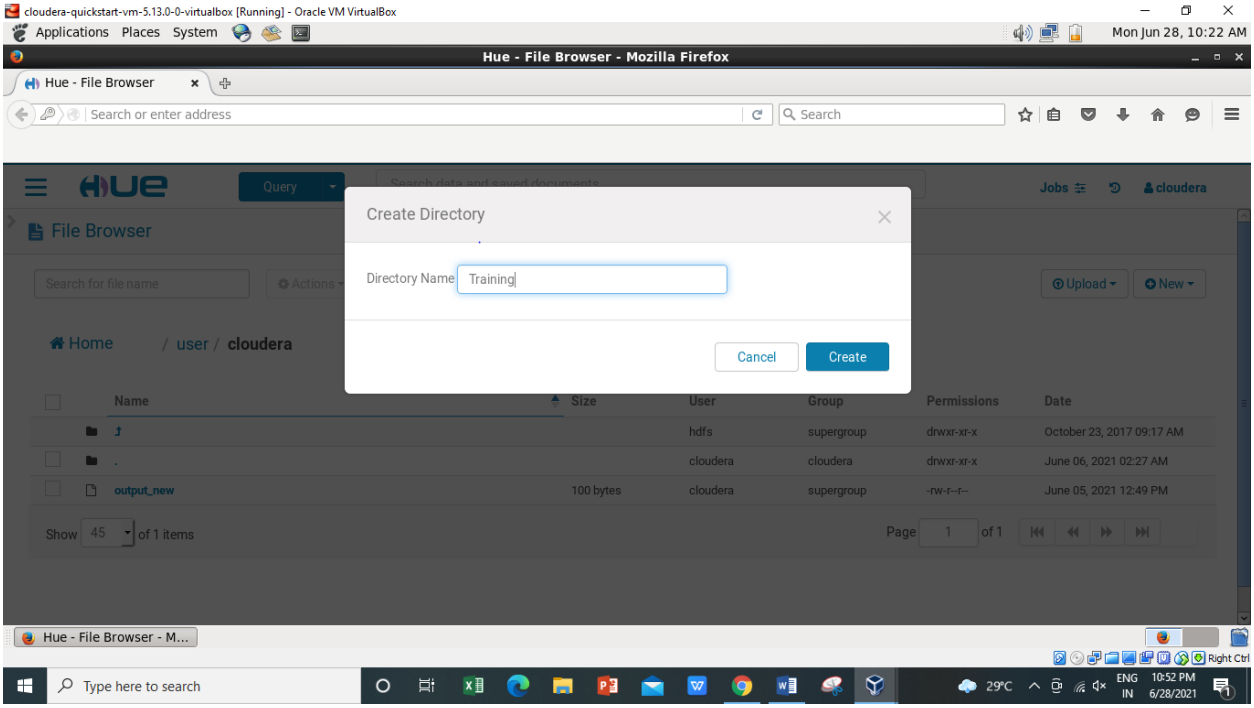


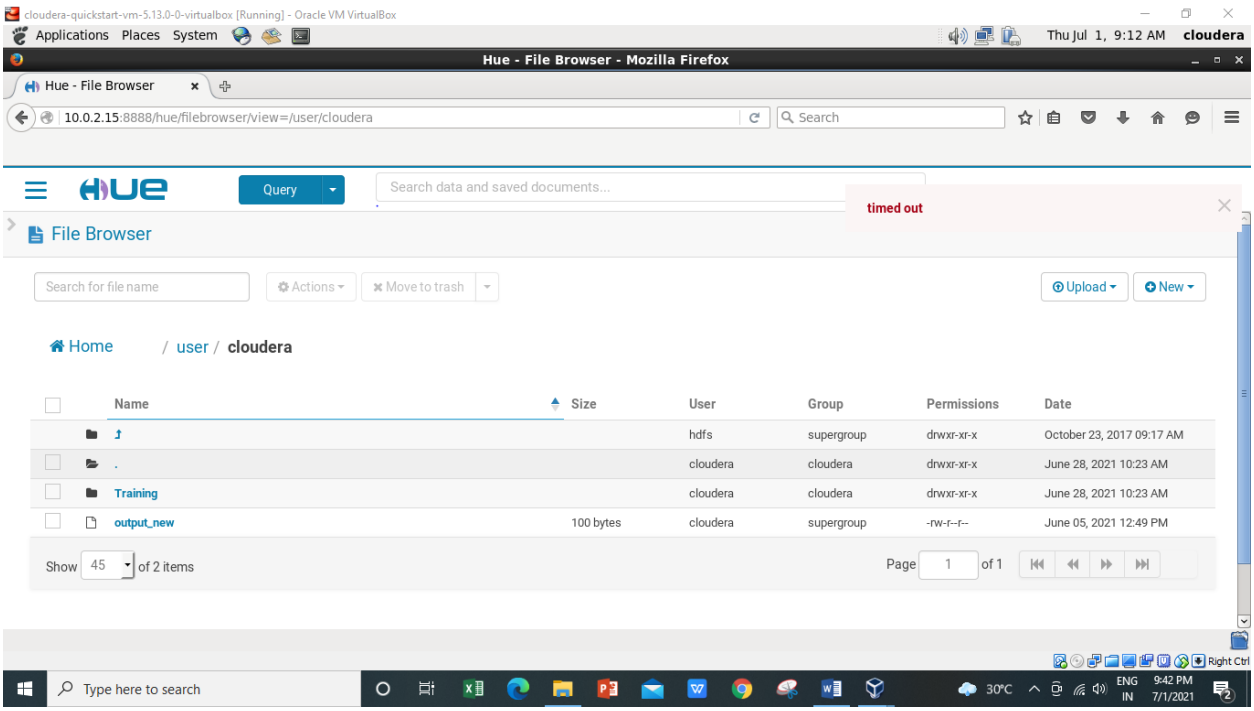
3) Now open the directory /user/cloudera



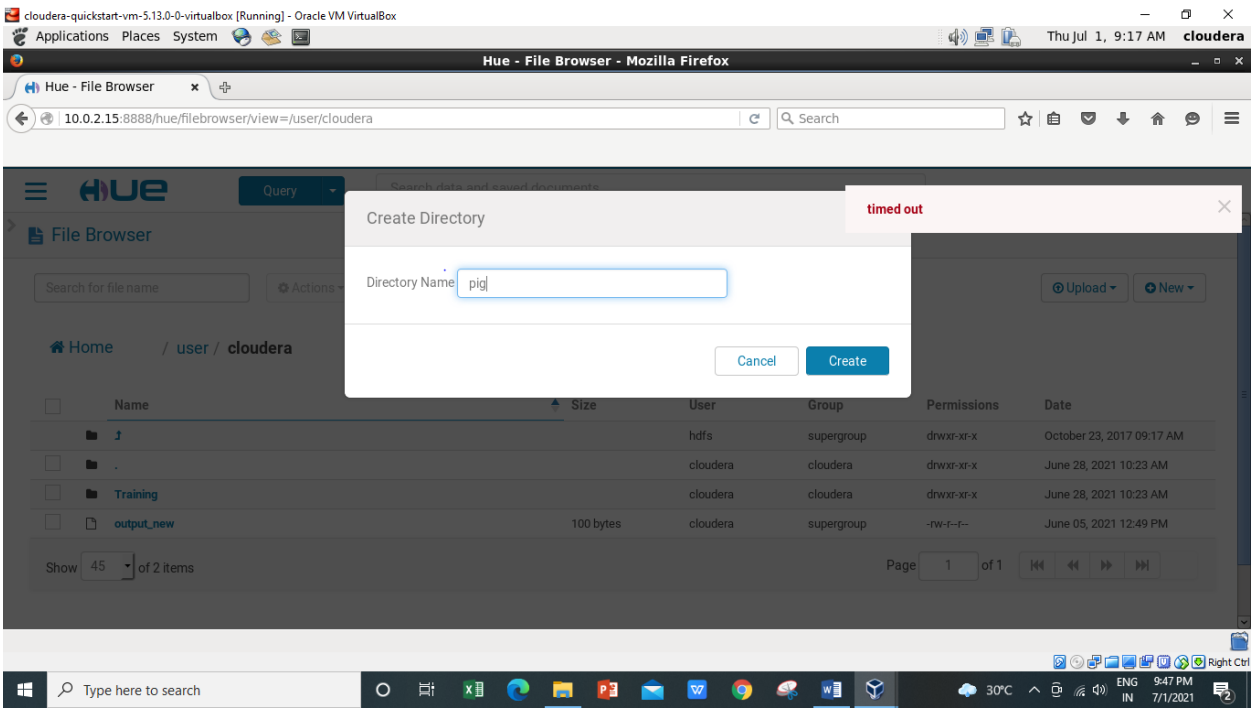
4) Now we are creating the directory as **Training** inside /user/cloudera

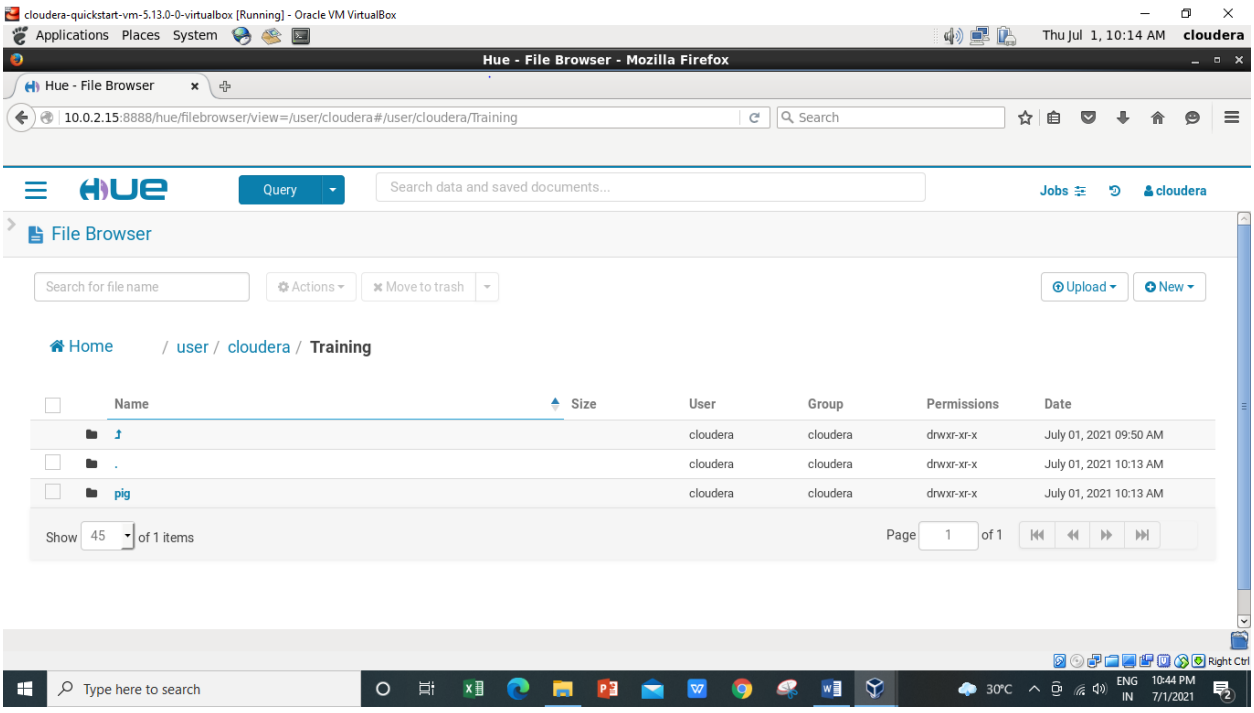

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Mon Jun 28, 10:21 AM
Hue - File Browser - Mozilla Firefox
Hue - File Browser
Search or enter address
Search
Query
Search data and saved documents...
Jobs
cloudera
File Browser
Search for file name
Actions
Move to trash
Upload
New
File
Directory
Home / user / cloudera
Name
Size
User
Group
Permissions
Date
hdfs supergroup drwxr-xr-x October 23, 2017 09:17 AM
cloudera cloudera drwxr-xr-x June 06, 2021 02:27 AM
output_new 100 bytes cloudera supergroup -rw-r--r-- June 05, 2021 12:49 PM
Show 45 of 1 items
Page 1 of 1
Hue - File Browser - M...
Right Ctrl
Type here to search
29°C
ENG IN
10:51 PM 6/28/2021


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Mon Jun 28, 10:22 AM
Hue - File Browser - Mozilla Firefox
Hue - File Browser
Search or enter address
Search
Query
Jobs
cloudera
File Browser
Create Directory
Directory Name
Training
Cancel
Create
Upload
New
Home / user / cloudera
Name
Size
User
Group
Permissions
Date
hdfs supergroup drwxr-xr-x October 23, 2017 09:17 AM
cloudera cloudera drwxr-xr-x June 06, 2021 02:27 AM
output_new 100 bytes cloudera supergroup -rw-r--r-- June 05, 2021 12:49 PM
Show 45 of 1 items
Page 1 of 1
Hue - File Browser - M...
Right Ctrl
Type here to search
29°C
ENG IN
10:52 PM 6/28/2021

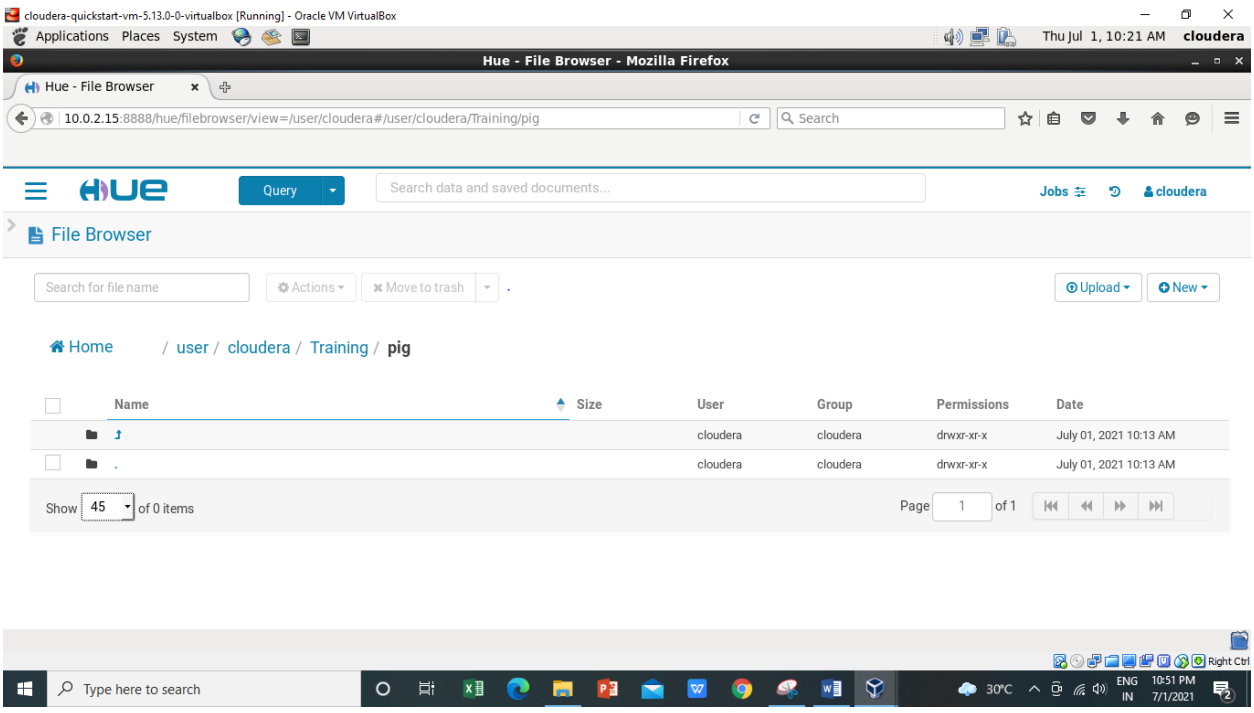


5) After creating Training directory now creating the **pig** directory inside Training.

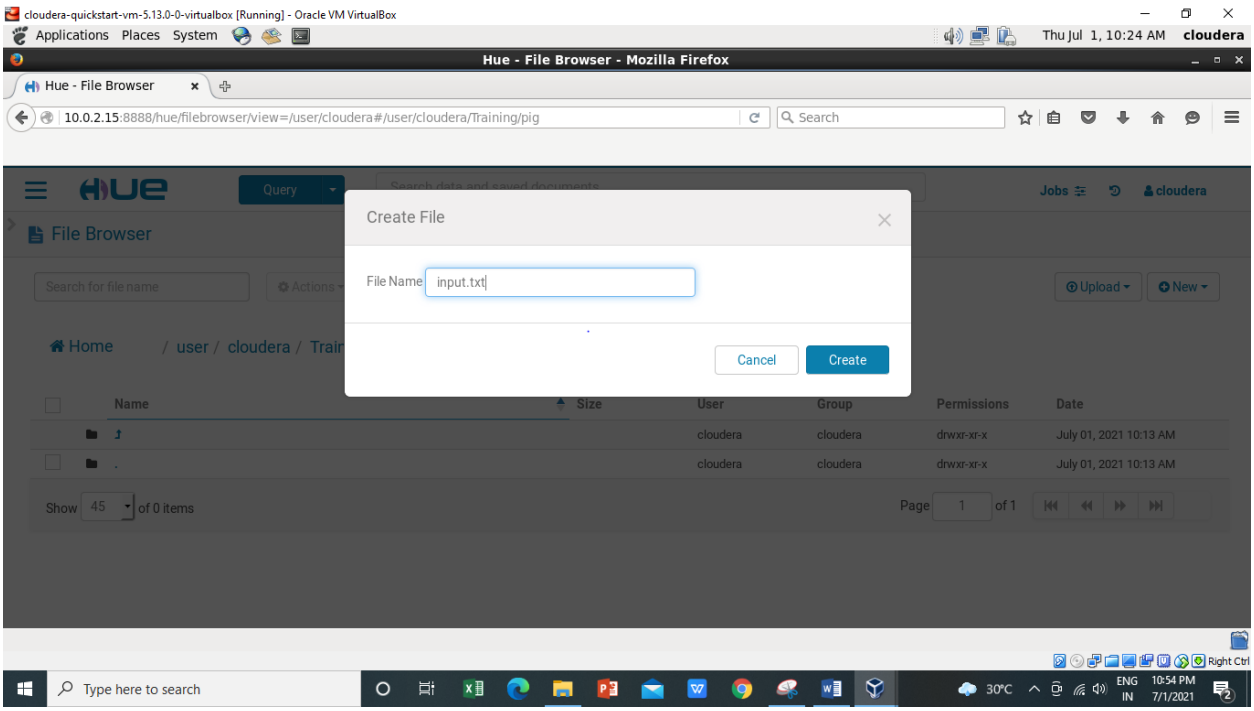


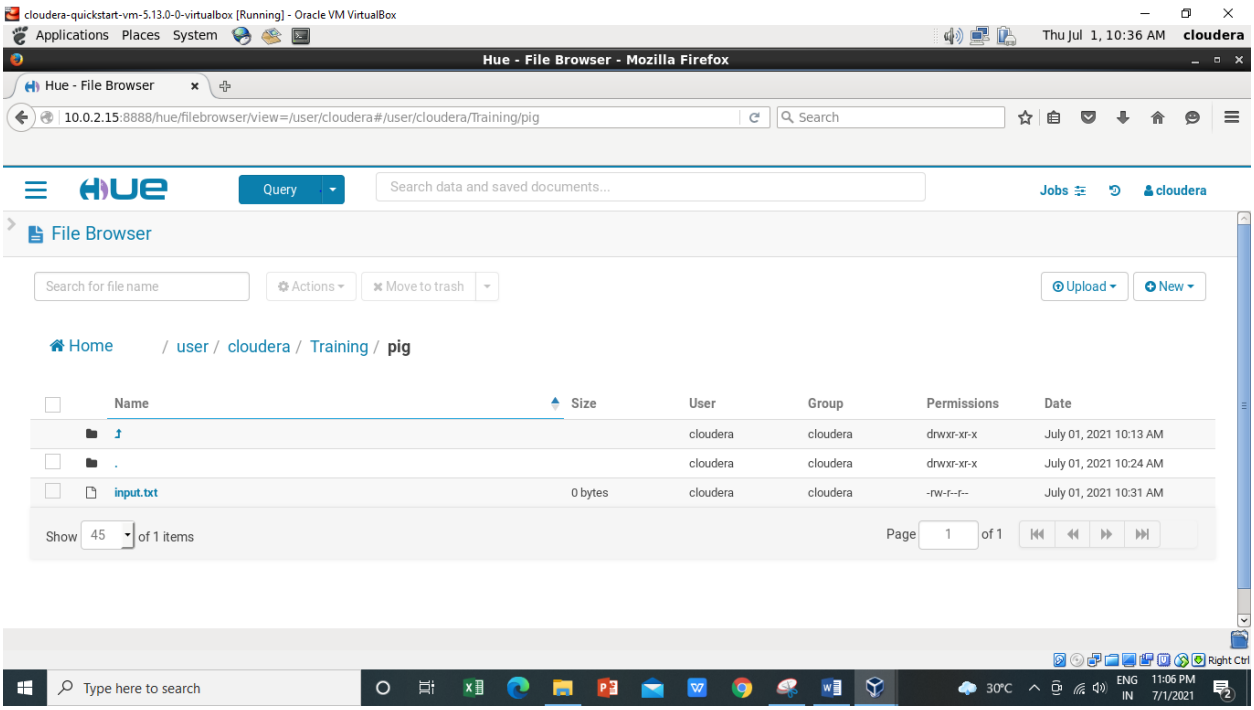


6) pig directory has been created inside /user/cloudera/Training

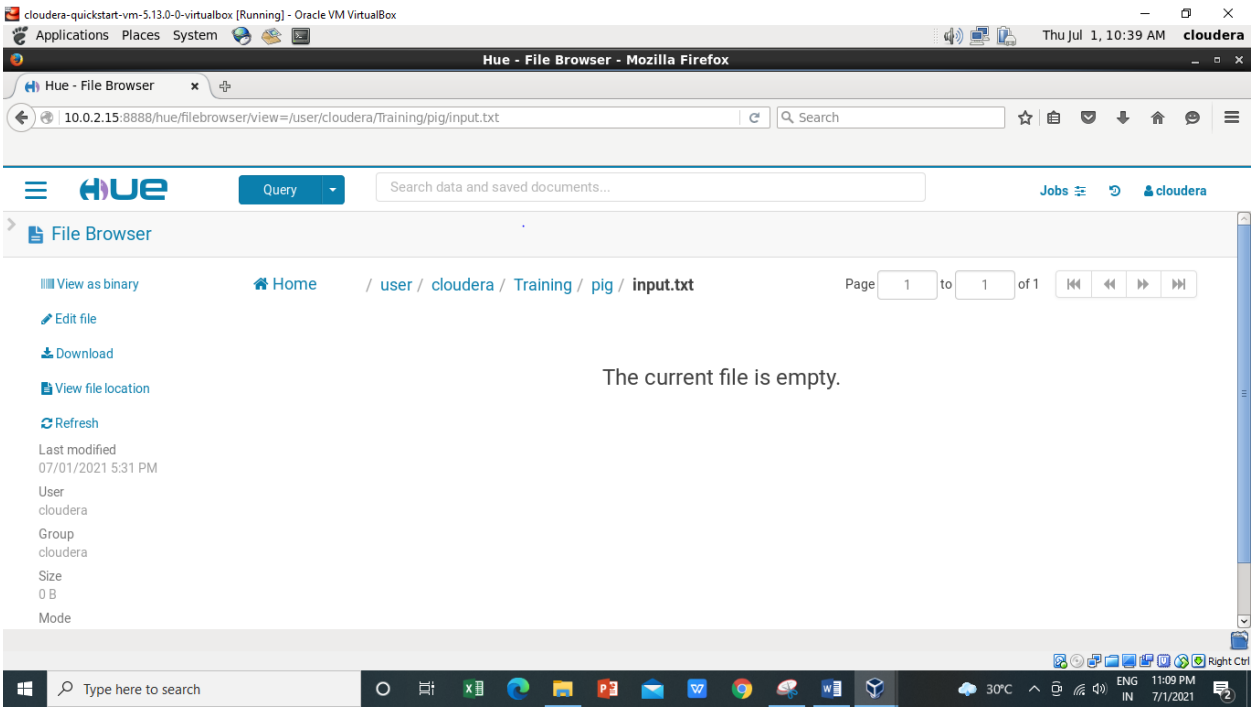


7) Creating **input.txt file** inside **/usr/cloudera/training/pig** directory

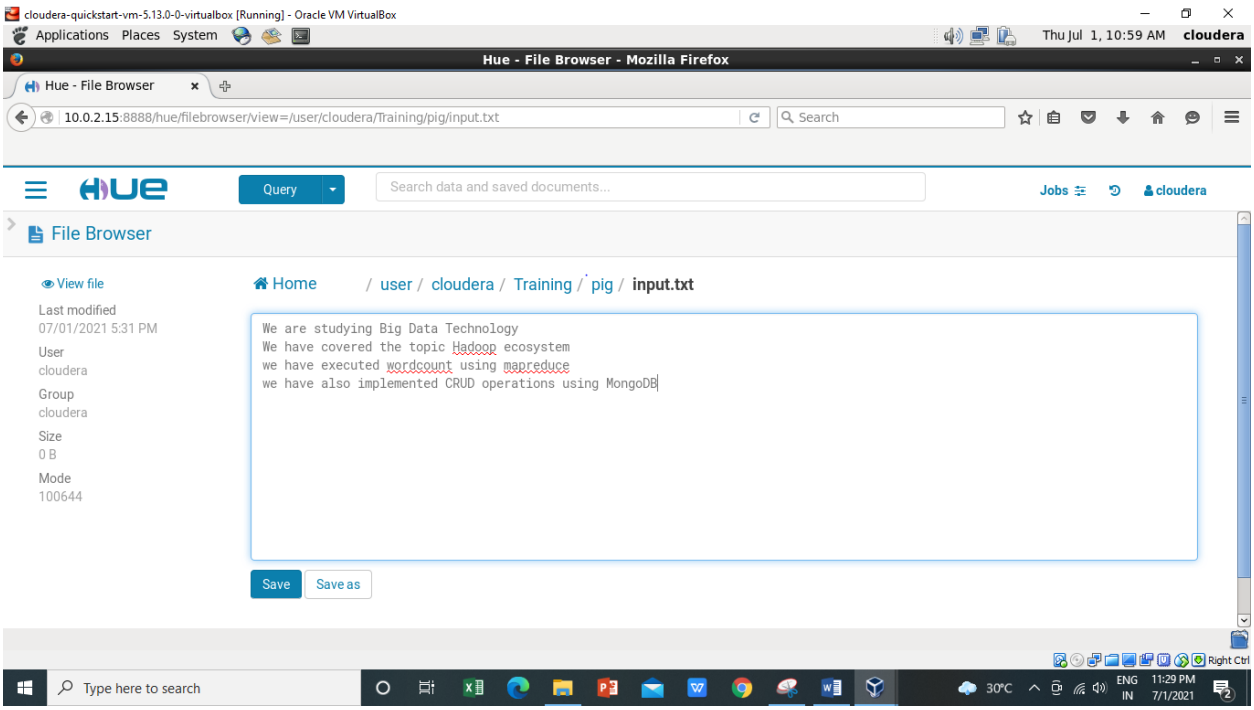




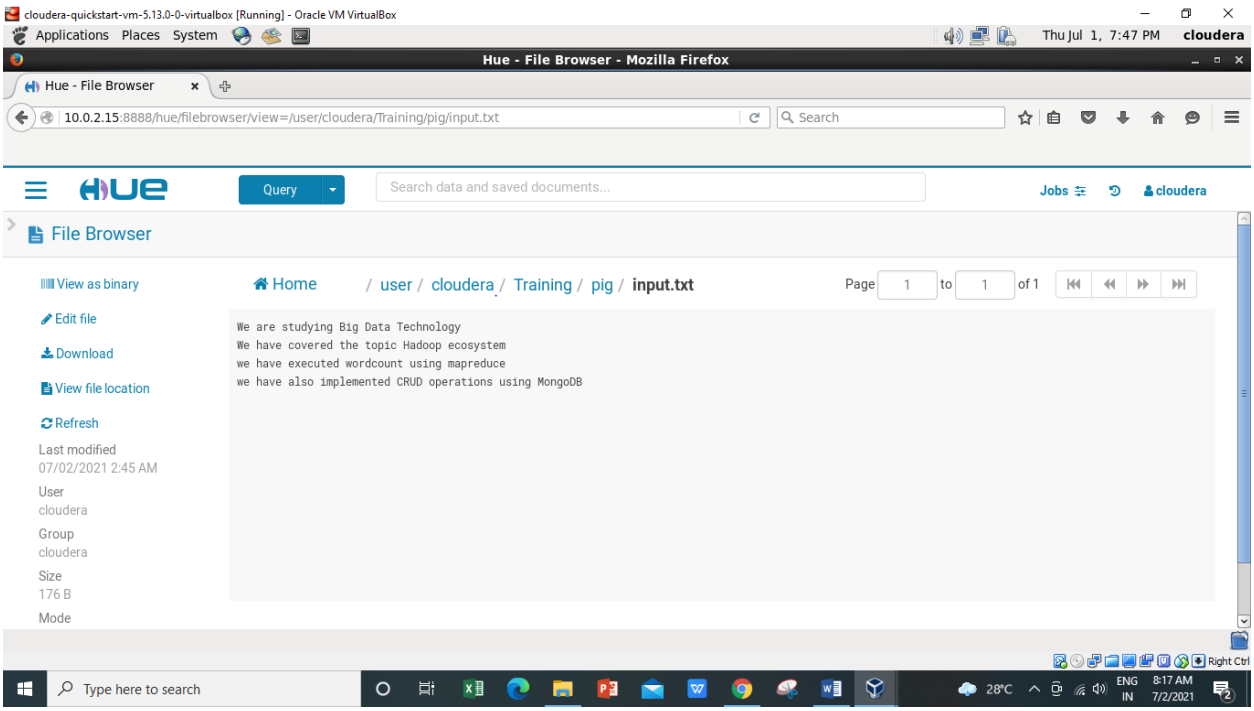
8) Adding some contents to this **input.txt** file.



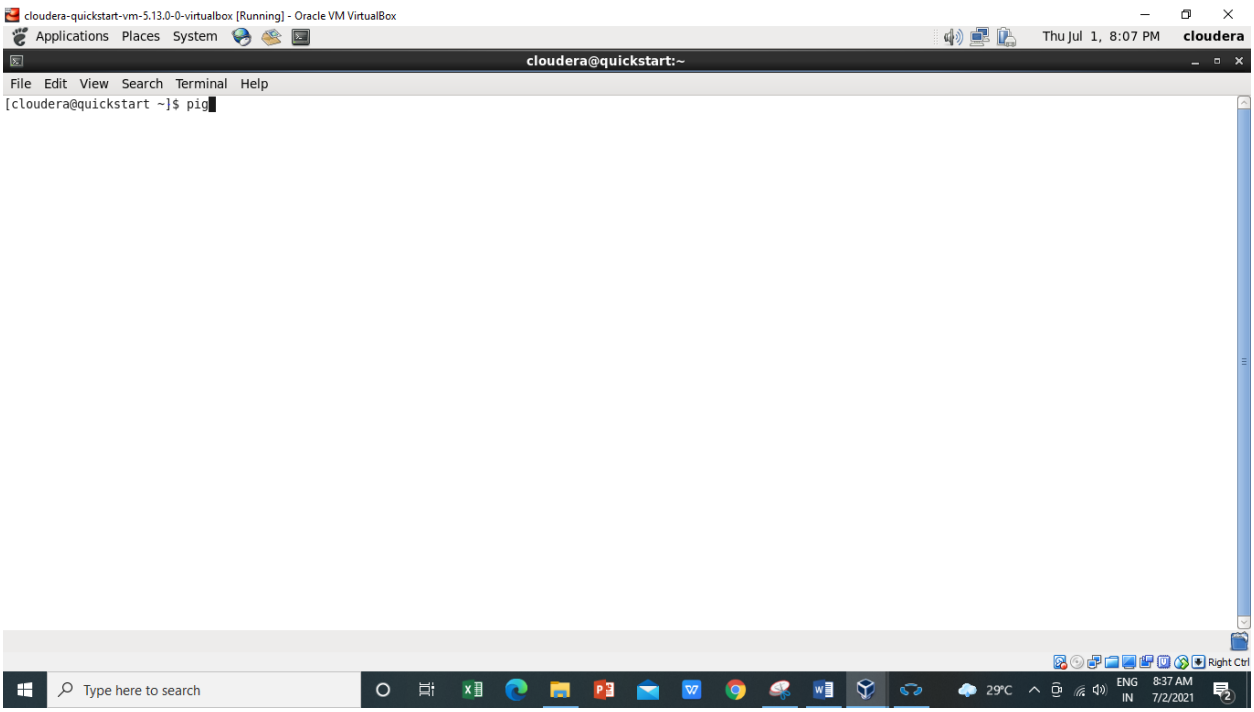
Click on **Edit file** option



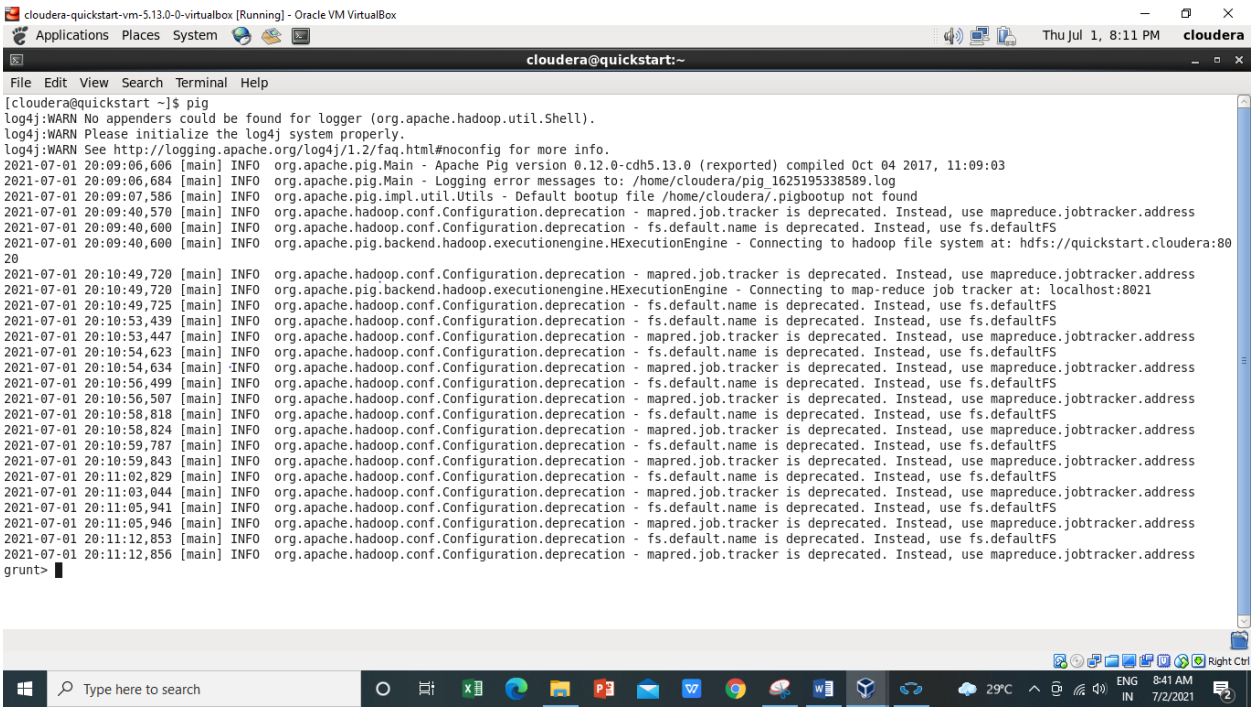
Save the **input.txt** file



9) Now Open the terminal. And start **Pig** by typing **pig** on terminal.



Now the pig started



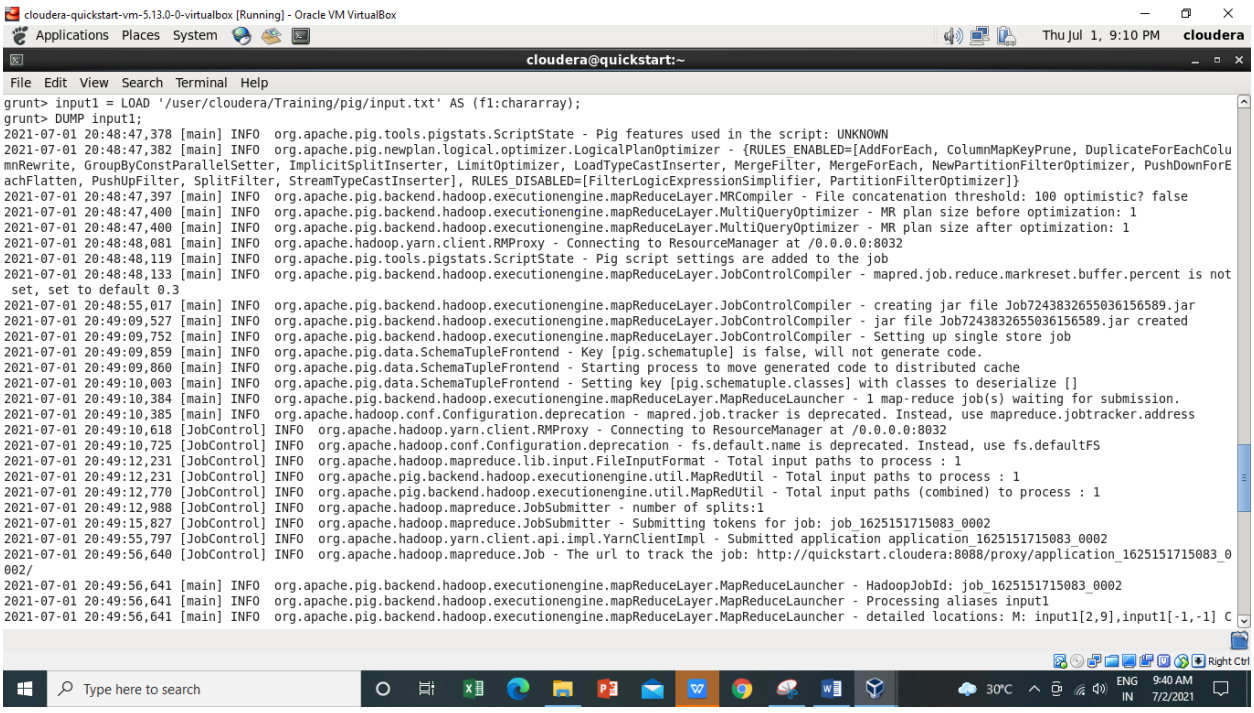
10) Now we have to load that input file where ever it is stored. By typing the command

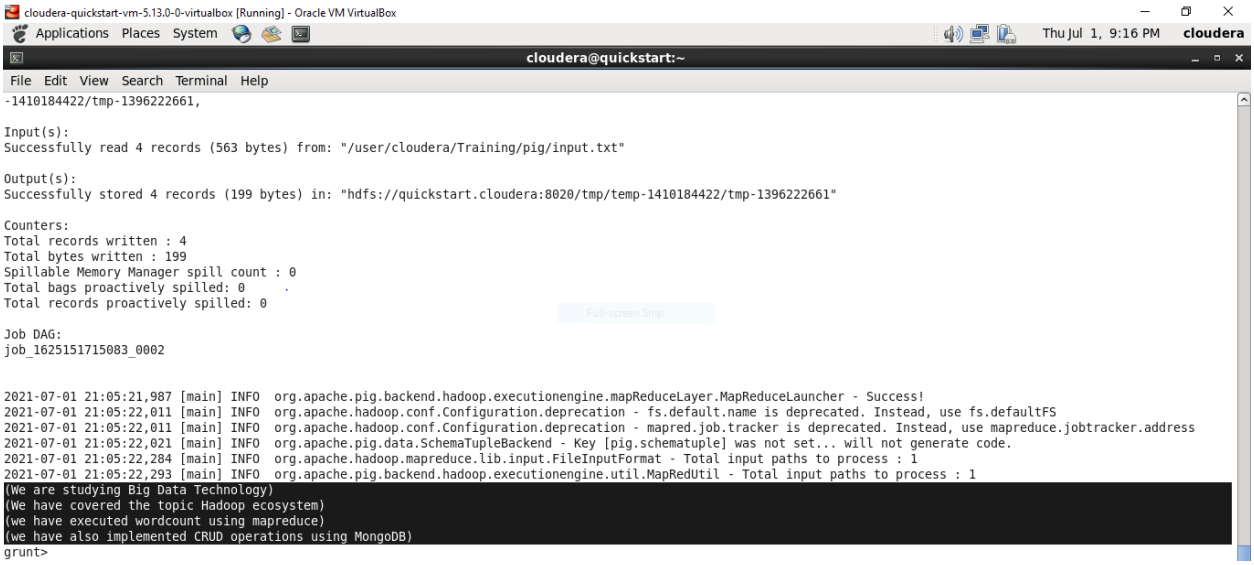
Input1 = LOAD '/usr/cloudera/Training/pig/input.txt' AS (f1:chararray);

```
grunt> input1 = LOAD '/user/cloudera/Training/pig/input.txt' AS (f1:chararray);
grunt>
```

11) Now we are dumping the data. **It will done the mapreduce task**.

DUMP input1;

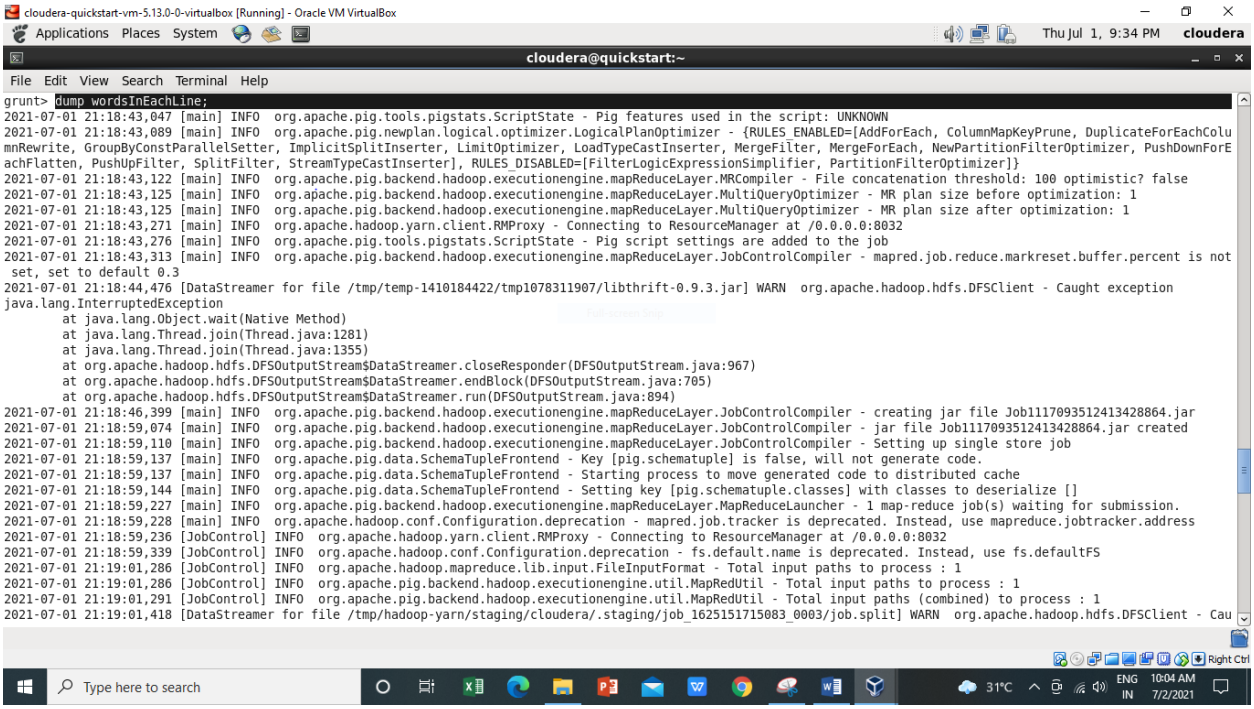


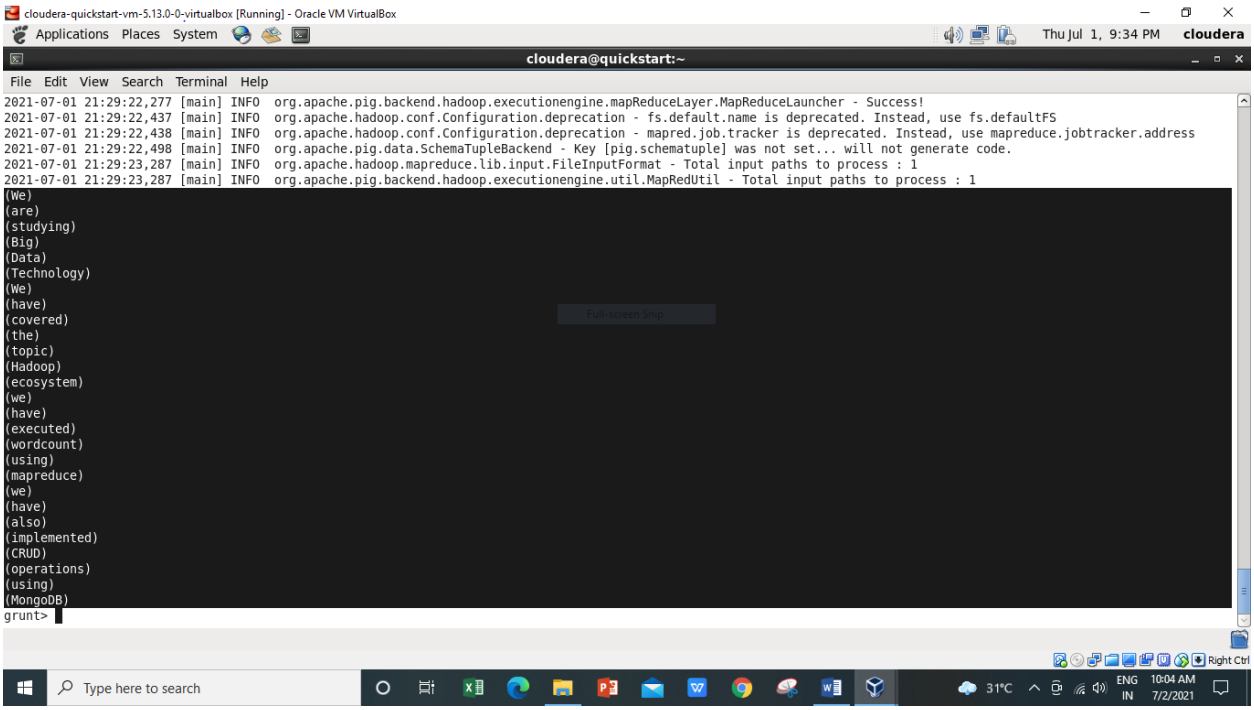


12) wordsInEachLine = FOREACH input1 GENERATE flatten(TOKENIZE(f1)) as word;

```
grunt> wordsInEachLine = FOREACH input1 GENERATE flatten(TOKENIZE(f1)) as word;
2021-07-01 21:14:45,902 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2021-07-01 21:14:45,902 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
grunt>
```

13) dump wordsInEachLine;



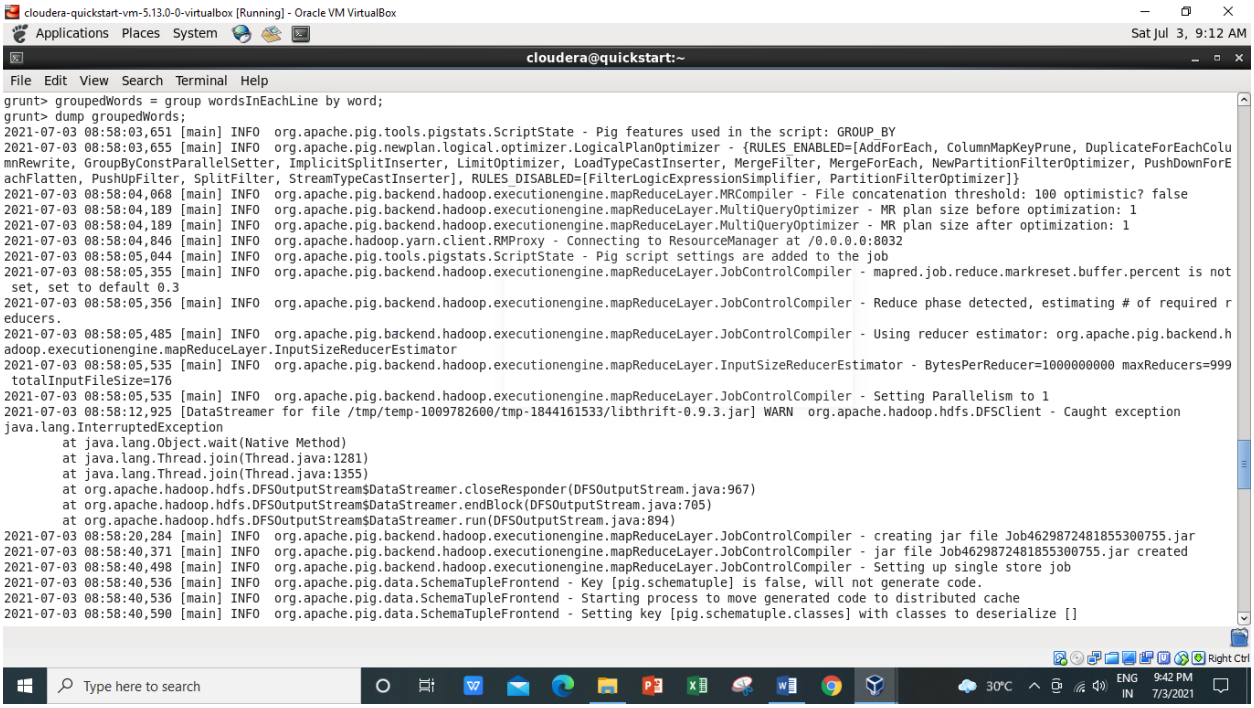


14) Now grouping the words present in each line.

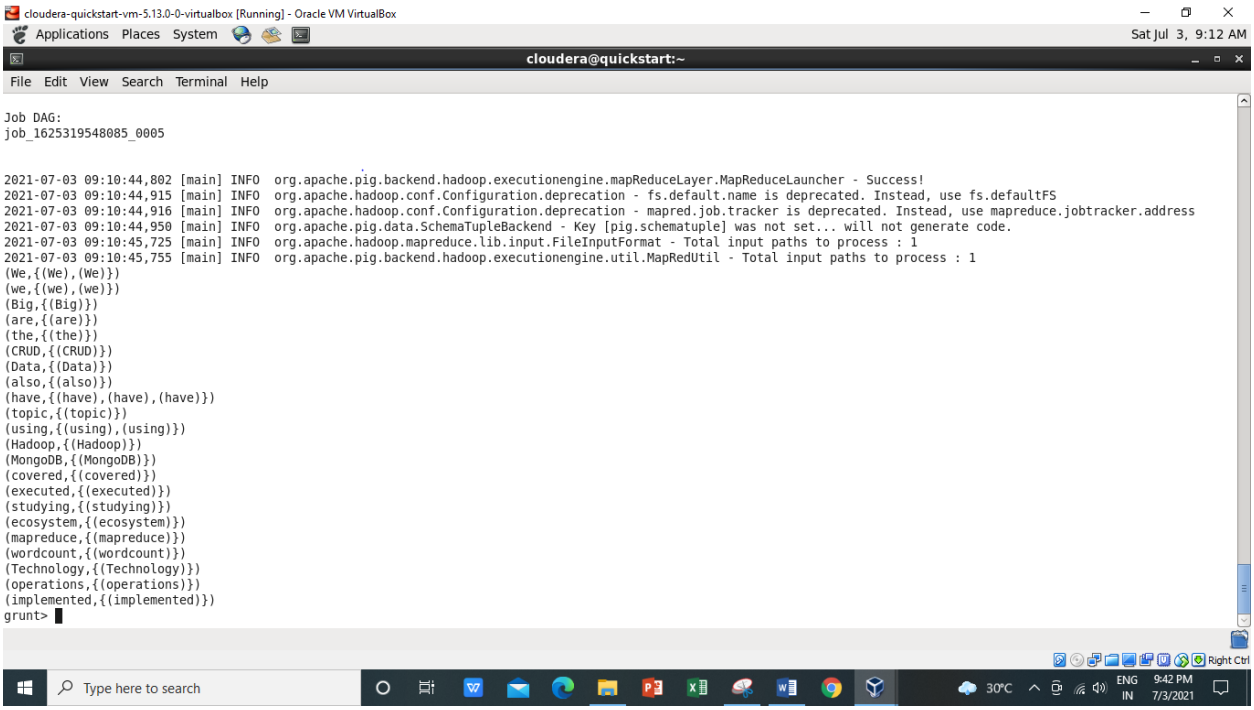
groupedWords = group wordsInEachLine by word;

```
grunt> groupedWords = group wordsInEachLine by word;
grunt>
```

dump groupedWords;



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Sat Jul 3, 9:12 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
grunt> groupedWords = group wordsInEachLine by word;
grunt> dump groupedWords;
2021-07-03 08:58:03,651 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GROUP_BY
2021-07-03 08:58:03,655 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColu
mnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForE
achFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
2021-07-03 08:58:04,068 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MRCompiler - File concatenation threshold: 100 optimistic? false
2021-07-03 08:58:04,189 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size before optimization: 1
2021-07-03 08:58:04,189 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MultiQueryOptimizer - MR plan size after optimization: 1
2021-07-03 08:58:04,846 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at /0.0.0.0:8032
2021-07-03 08:58:05,044 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig script settings are added to the job
2021-07-03 08:58:05,355 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - mapred.job.reduce.markreset.buffer.percent is not
 set, set to default 0.3
2021-07-03 08:58:05,356 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Reduce phase detected, estimating # of required r
educers.
2021-07-03 08:58:05,485 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Using reducer estimator: org.apache.pig.backend.h
adoop.executionengine.mapReduceLayer.InputSizeReducerEstimator
2021-07-03 08:58:05,535 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.InputSizeReducerEstimator - BytesPerReducer=1000000000 maxReducers=999
 totalInputFileSize=176
2021-07-03 08:58:05,535 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting Parallelism to 1
2021-07-03 08:58:12,925 [DataStreamer for file /tmp/temp-1009782600/tmp-1844161533/libthrift-0.9.3.jar] WARN  org.apache.hadoop.hdfs.DFSClient - Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1281)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
2021-07-03 08:58:20,284 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - creating jar file Job4629872481855300755.jar
2021-07-03 08:58:40,371 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - jar file Job4629872481855300755.jar created
2021-07-03 08:58:40,498 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.JobControlCompiler - Setting up single store job
2021-07-03 08:58:40,536 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Key [pig.schematuple] is false, will not generate code.
2021-07-03 08:58:40,536 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Starting process to move generated code to distributed cache
2021-07-03 08:58:40,590 [main] INFO  org.apache.pig.data.SchemaTupleFrontend - Setting key [pig.schematuple.classes] with classes to deserialize []
Right Ctrl
Type here to search
30°C
ENG IN
9:42 PM 7/3/2021
```



```
cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Sat Jul 3, 9:12 AM
cloudera@quickstart:~
File Edit View Search Terminal Help

Job DAG:
job_1625319548085_0005


2021-07-03 09:10:44,802 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2021-07-03 09:10:44,915 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2021-07-03 09:10:44,916 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2021-07-03 09:10:44,950 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2021-07-03 09:10:45,725 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2021-07-03 09:10:45,755 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(We,{(We),(We)})
(we,{(we),(we)})
(Big,{(Big)})
(are,{(are)})
(the,{(the)})
(CRUD,{(CRUD)})
(Data,{(Data)})
(also,{(also)})
(have,{(have),(have),(have)})
(topic,{(topic)})
(using,{(using),(using)})
(Hadoop,{(Hadoop)})
(MongoDB,{(MongoDB)})
(covered,{(covered)})
(executed,{(executed)})
(studying,{(studying)})
(ecosystem,{(ecosystem)})
(mapreduce,{(mapreduce)})
(wordcount,{(wordcount)})
(Technology,{(Technology)})
(operations,{(operations)})
(implemented,{(implemented)})
grunt>
Right Ctrl
Type here to search
30°C
ENG IN
9:42 PM 7/3/2021
```

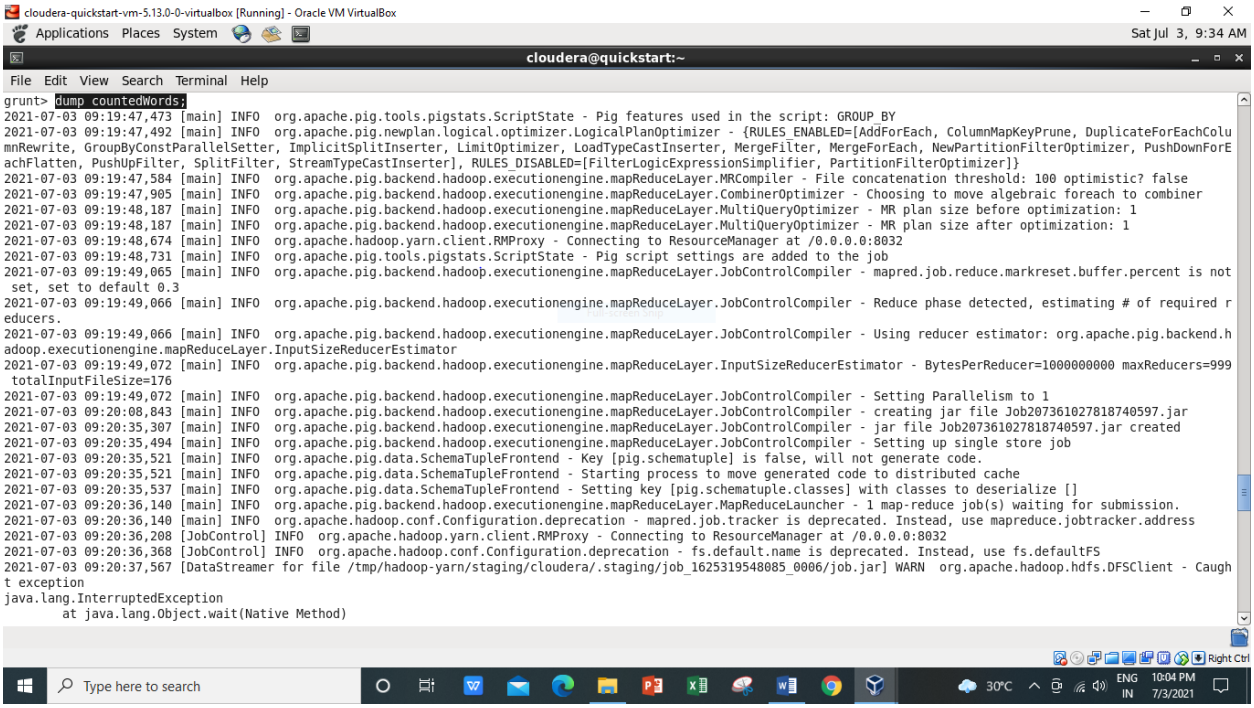
15) Now we count those words. For each group we count words in each line.

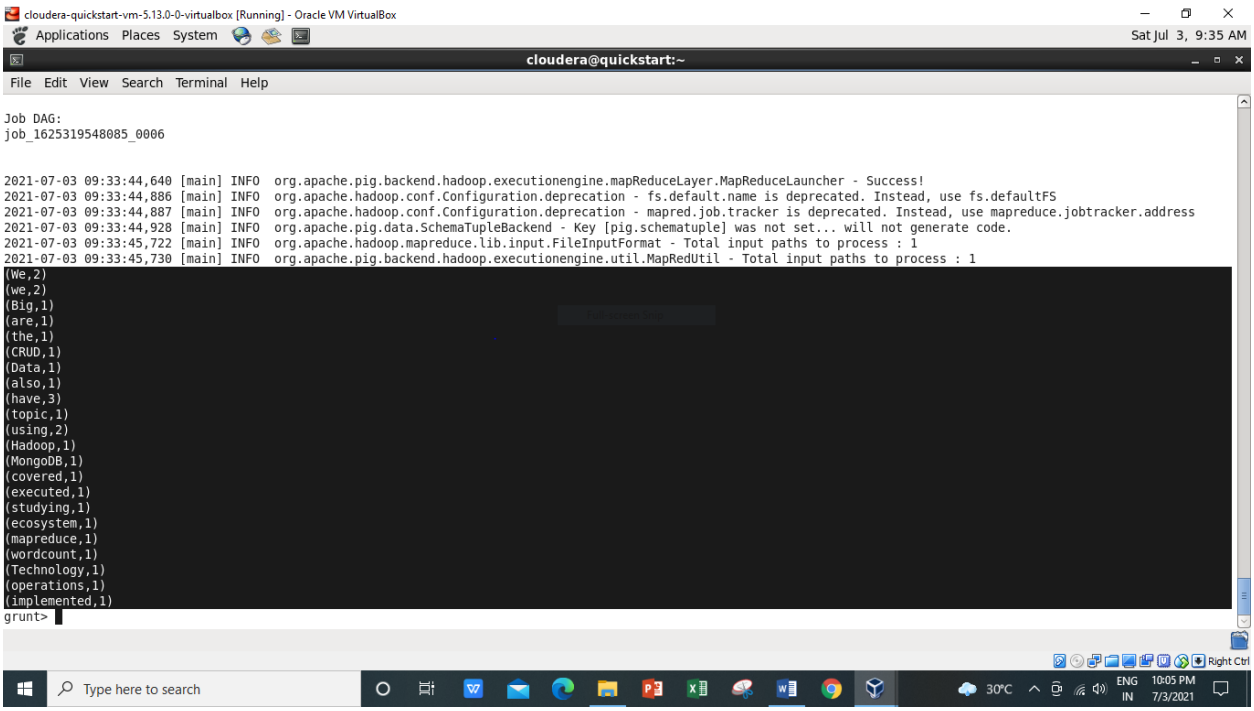
countedWords = foreach groupedWords generate group, COUNT(wordsInEachLine);

```
grunt> countedWords = foreach groupedWords generate group, COUNT(wordsInEachLine);
grunt>
```

16) dump countedWords;

Now the Final Output we are getting as word count for every word.





As we can see from above image the Word “We” start with capital W occurred twice, word “we” start with small w occurred twice, word “Big” occurred once, and so on.

17) Now Exit from the grunt shell using quit command.

```
grunt> quit
[cloudera@quickstart ~]$
```