Assignment – 3

Experiment 3 - To Implement WordCount problem using Hadoop MapReduce in Eclipse: (Without Combiner & With Combiner)

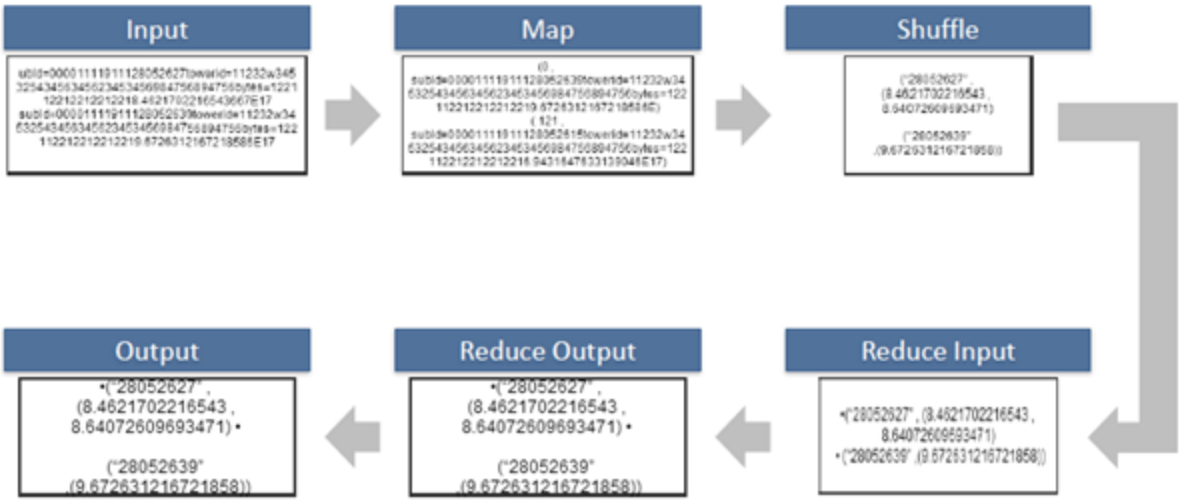
What is MapReduce ?

A MapReduce is a data processing tool which is used to process the data parallelly in a distributed form. It was developed in 2004, on the basis of paper titled as "MapReduce: Simplified Data Processing on Large Clusters," published by Google.

The MapReduce is a paradigm which has two phases, the mapper phase, and the reducer phase. In the Mapper, the input is given in the form of a key-value pair. The output of the Mapper is fed to the reducer as input. The reducer runs only after the Mapper is over. The reducer too takes input in key-value format, and the output of reducer is the final output.

Steps in Map Reduce

- The map takes data in the form of pairs and returns a list of <key, value> pairs. The keys will not be unique in this case.
- Using the output of Map, sort and shuffle are applied by the Hadoop architecture. This sort and shuffle acts on these list of <key, value> pairs and sends out unique keys and a list of values associated with this unique key <key, list(values)>.
- An output of sort and shuffle sent to the reducer phase. The reducer performs a defined function on a list of values for unique keys, and Final output <key, value> will be stored/displayed.

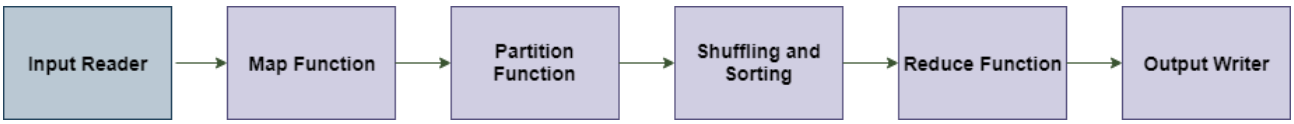


Sort and Shuffle

The sort and shuffle occur on the output of Mapper and before the reducer. When the Mapper task is complete, the results are sorted by key, partitioned if there are multiple reducers, and then written to disk. Using the input from each Mapper <k2,v2>, we collect all the values for each unique key k2. This output from the shuffle phase in the form of <k2, list(v2)> is sent as input to reducer phase.

Data Flow In MapReduce

MapReduce is used to compute the huge amount of data . To handle the upcoming data in a parallel and distributed form, the data has to flow from various phases.



Phases of MapReduce data flow

Input reader

The input reader reads the upcoming data and splits it into the data blocks of the appropriate size (64 MB to 128 MB). Each data block is associated with a Map function.

Once input reads the data, it generates the corresponding key-value pairs. The input files reside in HDFS.

Map function

The map function process the upcoming key-value pairs and generated the corresponding output key-value pairs. The map input and output type may be different from each other.

Partition function

The partition function assigns the output of each Map function to the appropriate reducer. The available key and value provide this function. It returns the index of reducers.

Shuffling and Sorting

The data are shuffled between/within nodes so that it moves out from the map and get ready to process for reduce function. Sometimes, the shuffling of data can take much computation time.

The sorting operation is performed on input data for Reduce function. Here, the data is compared using comparison function and arranged in a sorted form.

Reduce function

The Reduce function is assigned to each unique key. These keys are already arranged in sorted order. The values associated with the keys can iterate the Reduce and generates the corresponding output.

Output writer

Once the data flow from all the above phases, Output writer executes. The role of Output writer is to write the Reduce output to the stable storage.

To Implement WordCount problem using Hadoop MapReduce in Eclipse:

Hadoop WordCount operation occurs in 3 stages –

- Mapper Phase
- Shuffle Phase
- Reducer Phase

Hadoop WordCount - Mapper Phase Execution

- The text from the input text file is tokenized into words to form a key value pair with all the words present in the input text file. The key is the word from the input file and value is ‘1’.

- For instance if you consider the sentence “An elephant is an animal”.
- The mapper phase in the WordCount example will split the string into individual tokens i.e. words. In this case, the entire sentence will be split into 5 tokens (one for each word) with a value 1 as shown below –

  Key-Value pairs from Hadoop Map Phase Execution-

  (an,1)

  (elephant,1)

  (is,1)

  (an,1)

  (animal,1)

Hadoop WordCount Example- Shuffle Phase Execution

- After the map phase execution is completed successfully, shuffle phase is executed automatically wherein the key-value pairs generated in the map phase are taken as input and then sorted in alphabetical order.
- After the shuffle phase is executed from the WordCount example code, the output will look like this -

  (an,1)

  (an,1)

  (animal,1)

  (elephant,1)

  (is,1)

Hadoop WordCount Example- Reducer Phase Execution

- In the reduce phase, all the keys are grouped together and the values for similar keys are added up to find the occurrences for a particular word.

- It is like an aggregation phase for the keys generated by the map phase. The reducer phase takes the output of shuffle phase as input and then reduces the key-value pairs to unique keys with values added up.
- In our example “An elephant is an animal.” is the only word that appears twice in the sentence.
- After the execution of the reduce phase of MapReduce WordCount example program, appears as a key only once but with a count of 2 as shown below -

  (an,2)

  (animal,1)

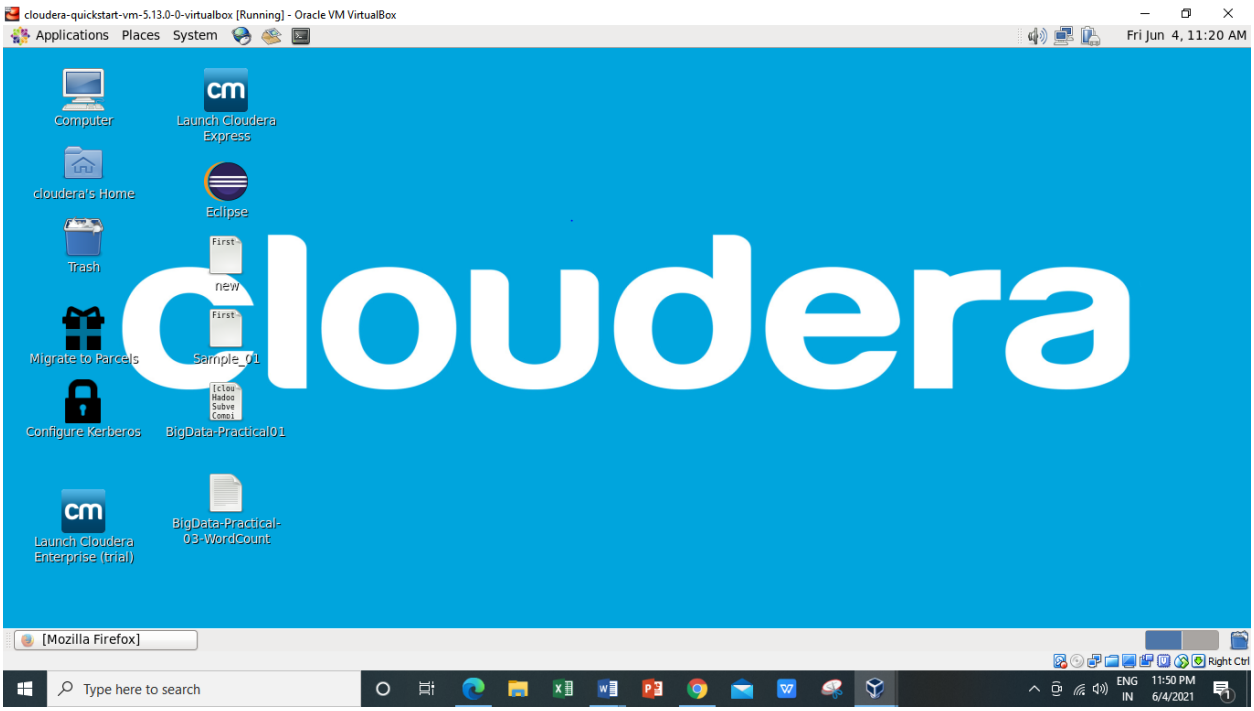
  (elephant,1)

  (is,1)

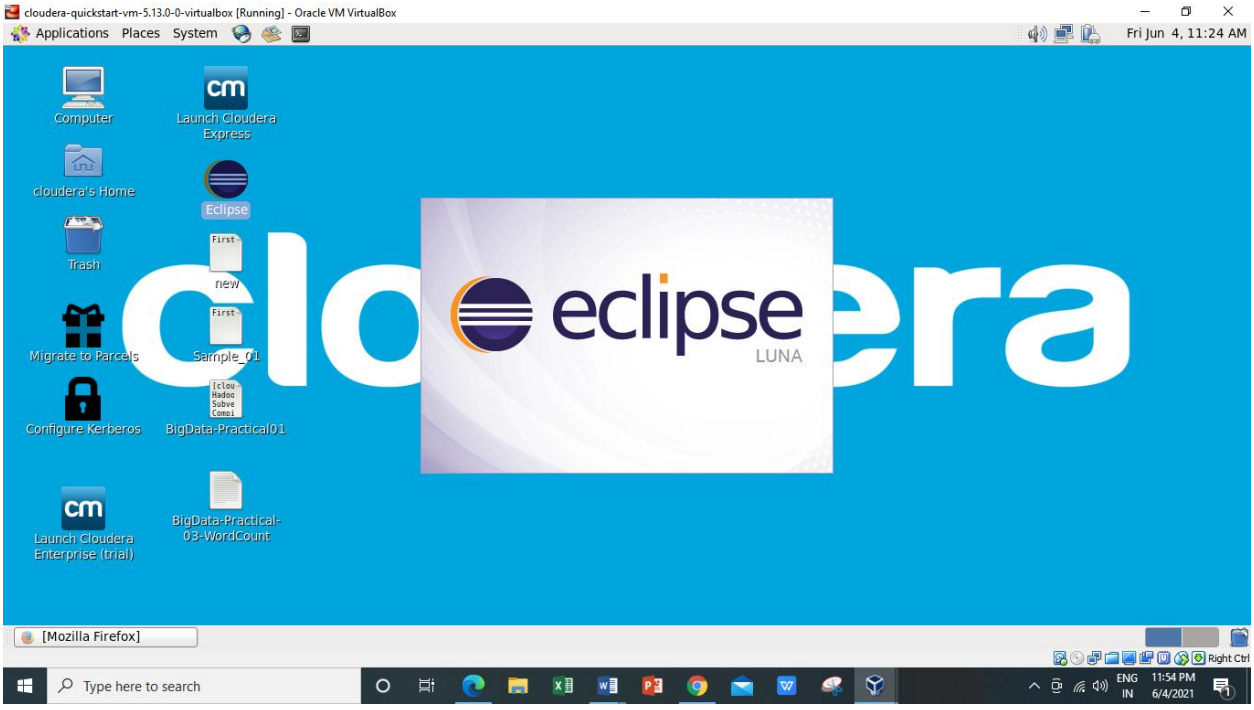
- This is how the MapReduce word count program executes and outputs the number of occurrences of a word in any given input file.
- An important point to note during the execution of the WordCount example is that the mapper class in the WordCount program will execute completely on the entire input file and not just a single sentence.
- Suppose if the input file has 15 lines then the mapper class will split the words of all the 15 lines and form initial key value pairs for the entire dataset.
- The reducer execution will begin only after the mapper phase is executed successfully.

Steps for Word Count in Cloudera: (Without Combiner)

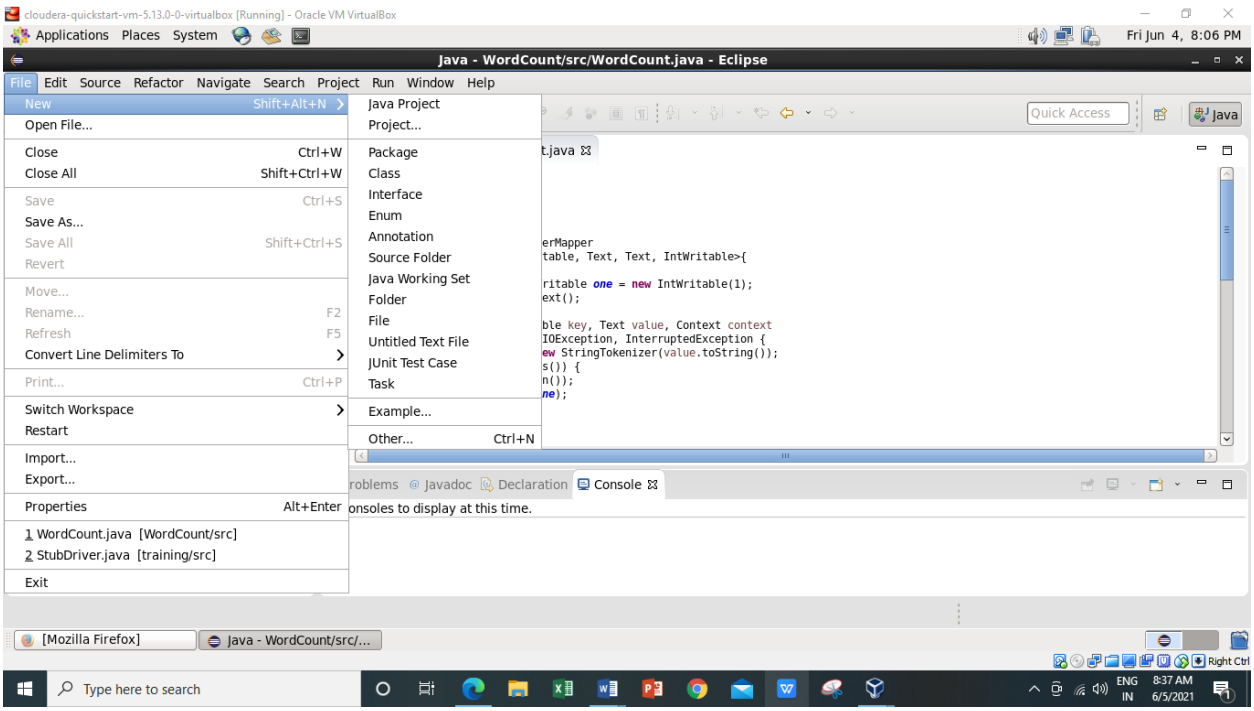
1) Open virtual box and then start cloudera quickstart

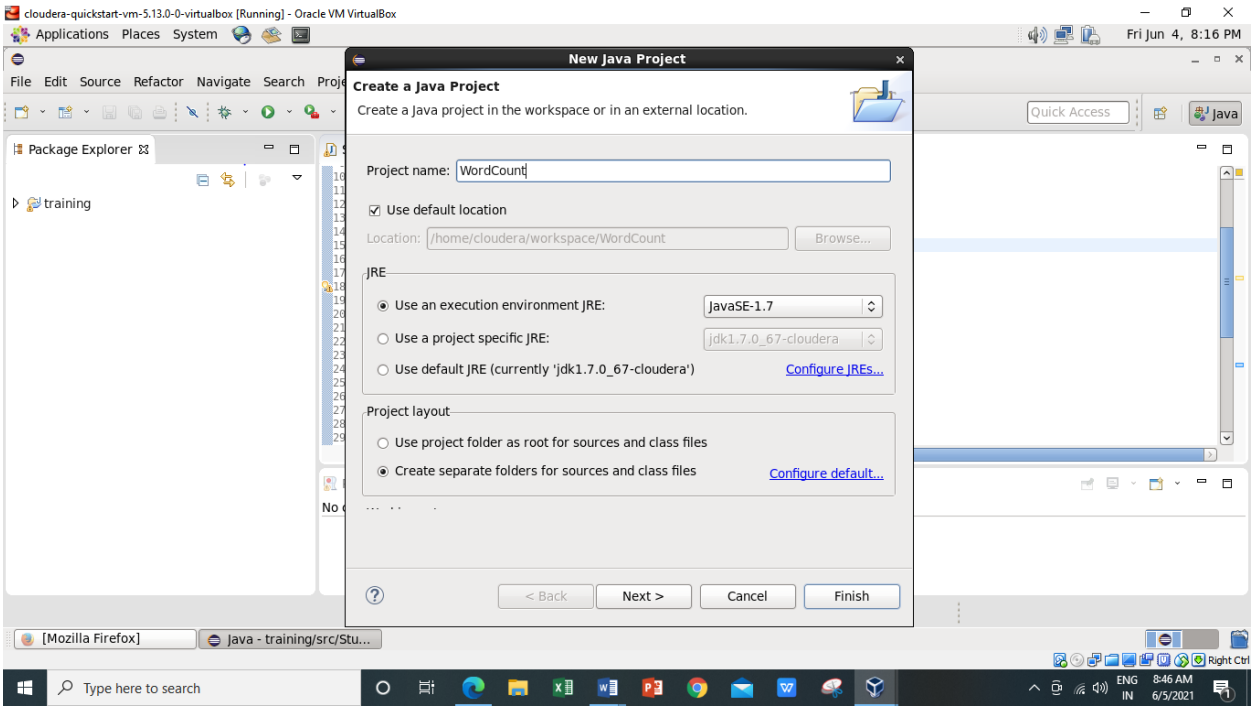


2) Open Eclipse present on the cloudera desktop

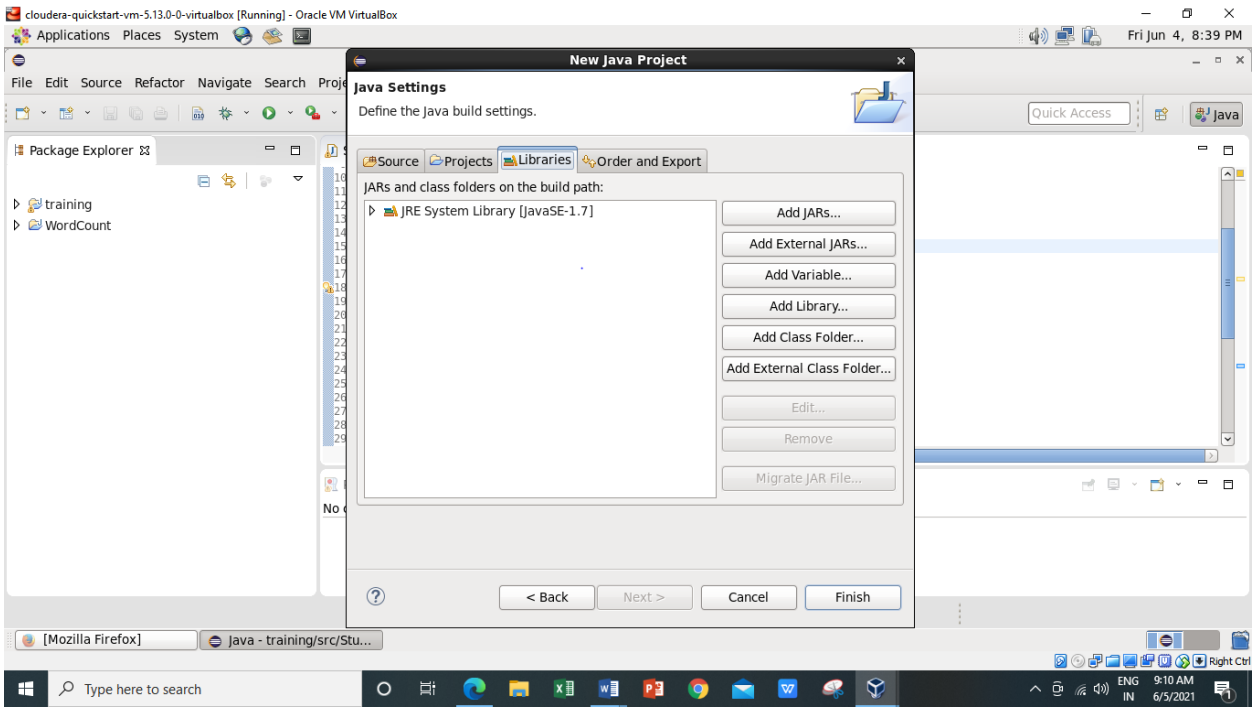


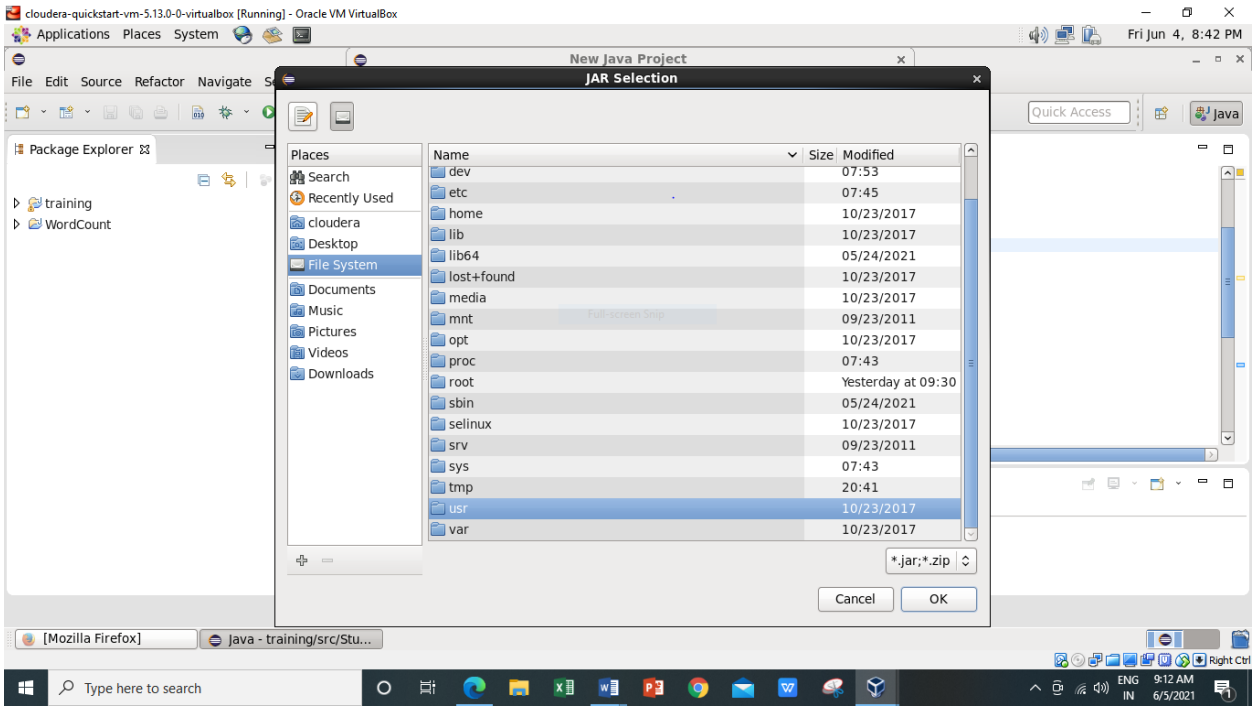
3) Create a new Java project clicking: File -> New -> Project -> Java Project -> Next ("WordCount" is the project name).

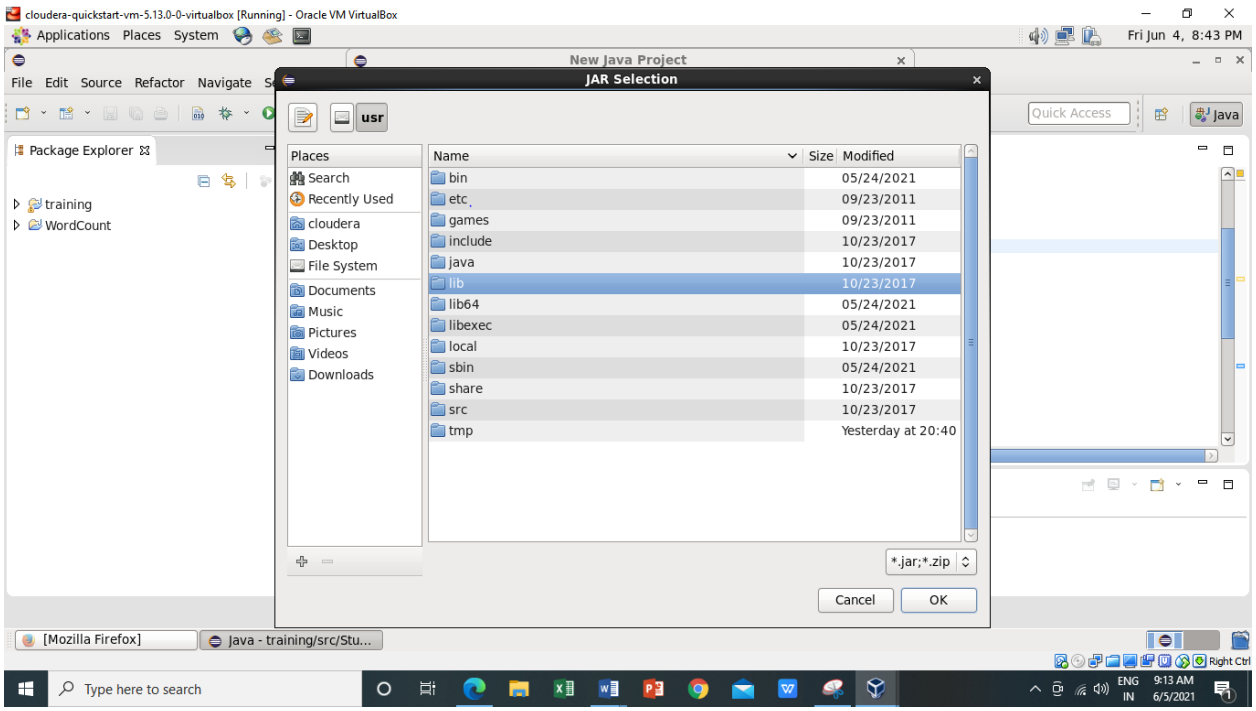


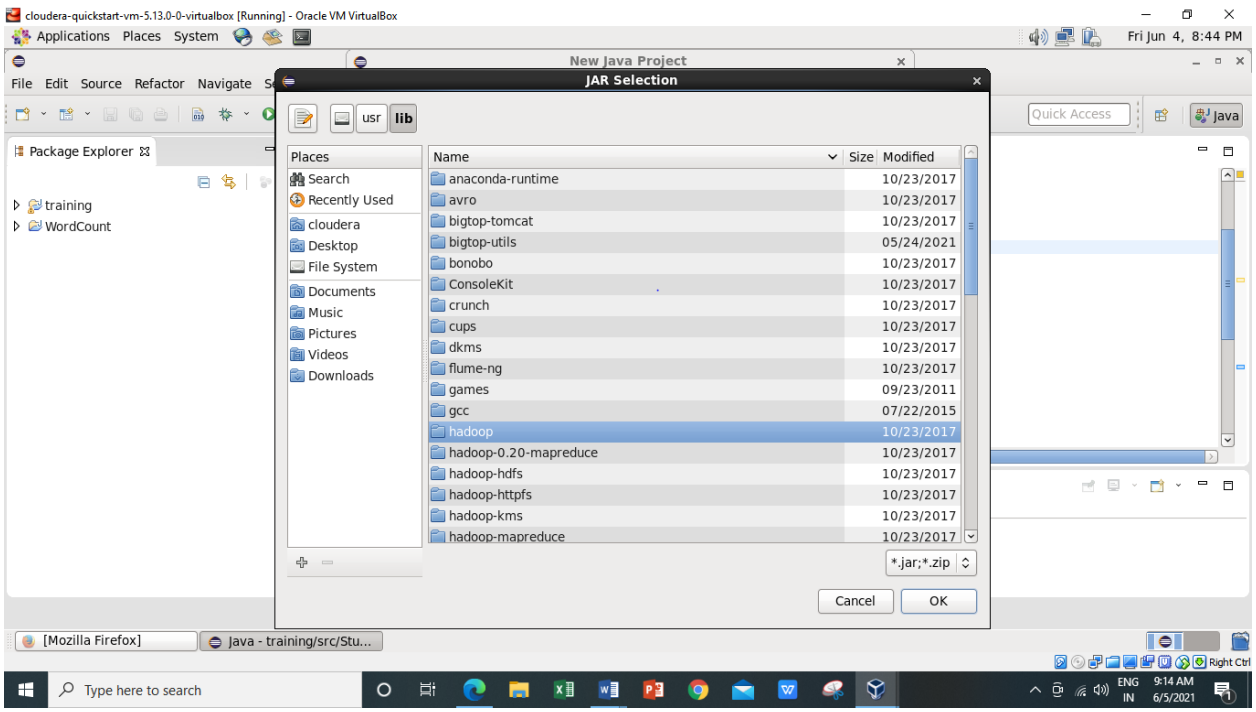


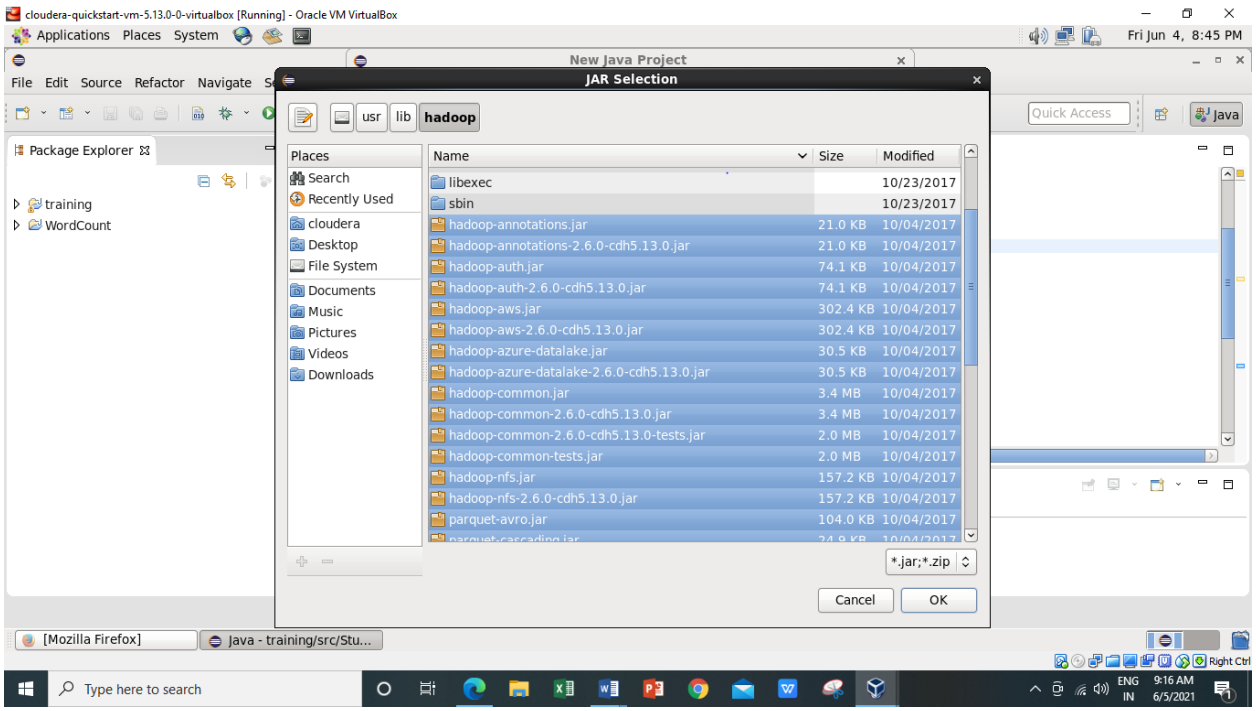
4) Adding the Hadoop libraries to the project Click on Libraries -> Add External JARs Click on File System -> usr -> lib -> hadoop Select all the libraries (JAR Files) -> click OK Click on Add External jars, -> client -> select all jar files -> ok -> Finish

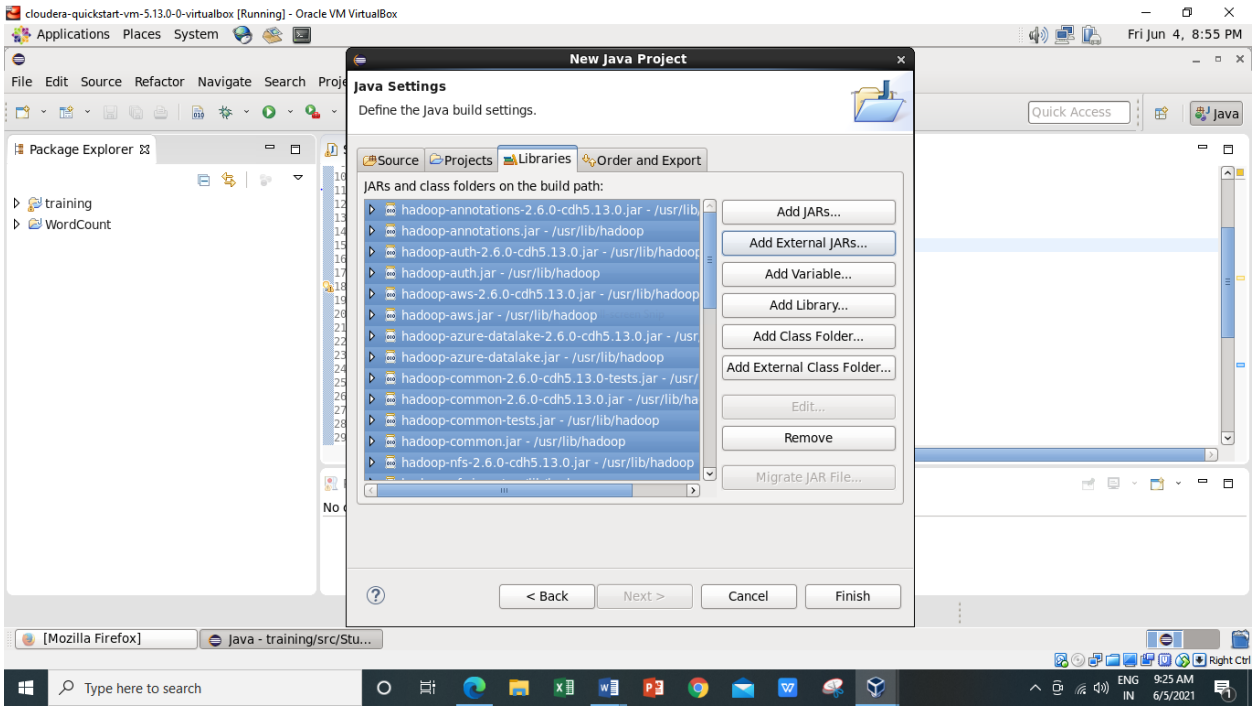

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:39 PM
New Java Project
Java Settings
Define the Java build settings.
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
JRE System Library [JavaSE-1.7]
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
< Back
Next >
Cancel
Finish
Package Explorer
training
WordCount
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search

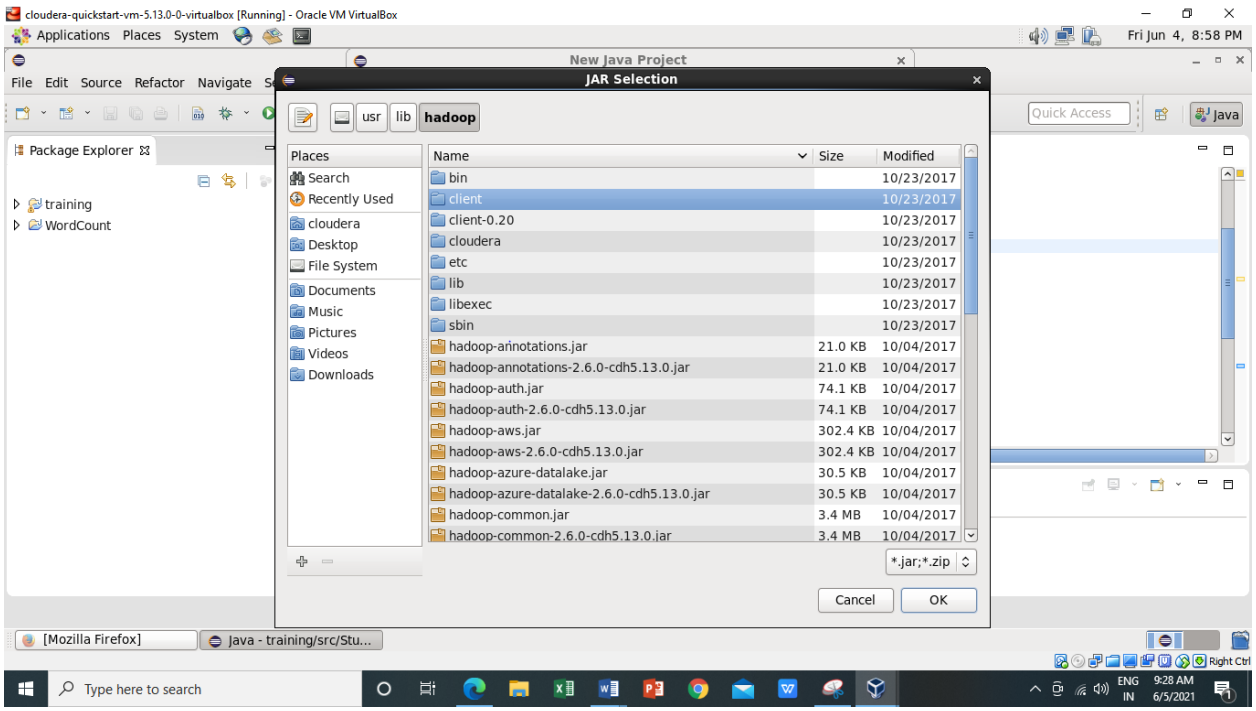

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:42 PM
New Java Project
JAR Selection
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
dev 07:53
etc 07:45
home 10/23/2017
lib 10/23/2017
lib64 05/24/2021
lost+found 10/23/2017
media 10/23/2017
mnt 09/23/2011
opt 10/23/2017
proc 07:43
root Yesterday at 09:30
sbin 05/24/2021
selinux 10/23/2017
srv 09/23/2011
sys 07:43
tmp 20:41
usr 10/23/2017
var 10/23/2017
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search

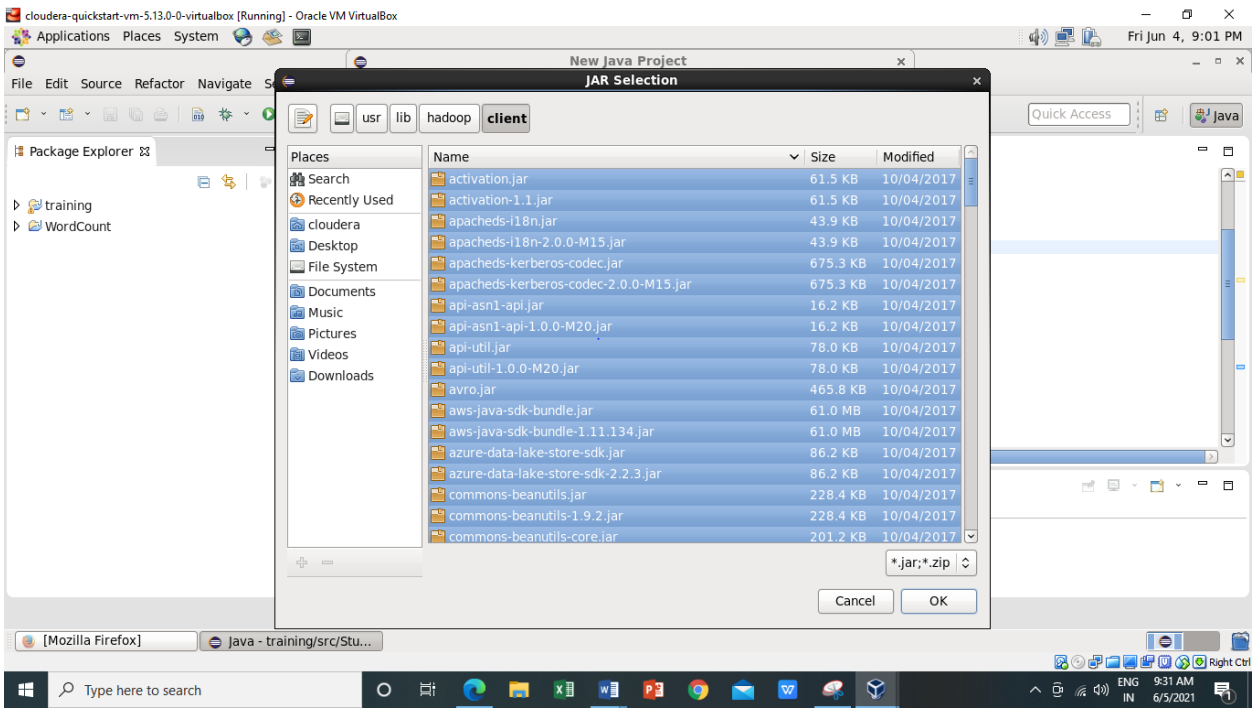

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:43 PM
New Java Project
File Edit Source Refactor Navigate
JAR Selection
usr
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
bin 05/24/2021
etc 09/23/2011
games 09/23/2011
include 10/23/2017
java 10/23/2017
lib 10/23/2017
lib64 05/24/2021
libexec 05/24/2021
local 10/23/2017
sbin 05/24/2021
share 10/23/2017
src 10/23/2017
tmp Yesterday at 20:40
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
Quick Access
Java
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
ENG IN 9:13 AM 6/5/2021

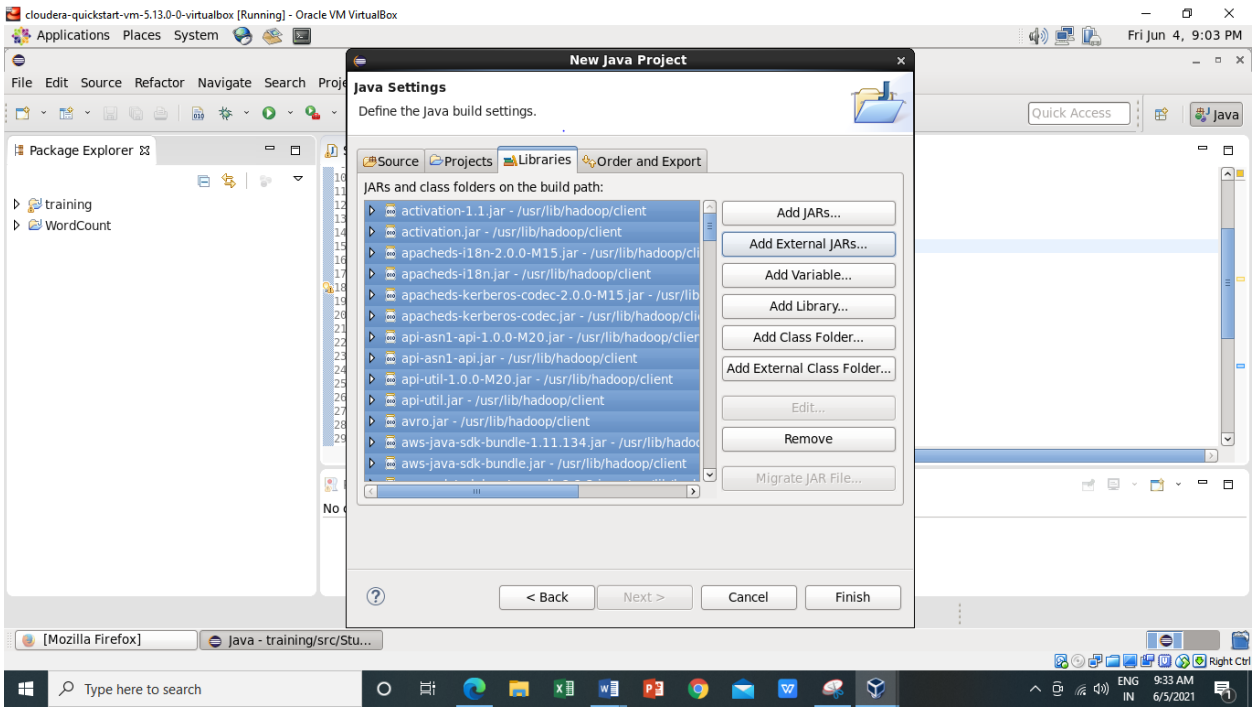

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:44 PM
New Java Project
File Edit Source Refactor Navigate
JAR Selection
usr
lib
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
anaconda-runtime 10/23/2017
avro 10/23/2017
bigtop-tomcat 10/23/2017
bigtop-utils 05/24/2021
bonobo 10/23/2017
ConsoleKit 10/23/2017
crunch 10/23/2017
cups 10/23/2017
dkms 10/23/2017
flume-ng 10/23/2017
games 09/23/2011
gcc 07/22/2015
hadoop 10/23/2017
hadoop-0.20-mapreduce 10/23/2017
hadoop-hdfs 10/23/2017
hadoop-httpfs 10/23/2017
hadoop-kms 10/23/2017
hadoop-mapreduce 10/23/2017
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
Quick Access
Java
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
ENG IN 9:14 AM 6/5/2021


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:45 PM
New Java Project
JAR Selection
usr lib hadoop
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
libexec 10/23/2017
sbin 10/23/2017
hadoop-annotations.jar 21.0 KB 10/04/2017
hadoop-annotations-2.6.0-cdh5.13.0.jar 21.0 KB 10/04/2017
hadoop-auth.jar 74.1 KB 10/04/2017
hadoop-auth-2.6.0-cdh5.13.0.jar 74.1 KB 10/04/2017
hadoop-aws.jar 302.4 KB 10/04/2017
hadoop-aws-2.6.0-cdh5.13.0.jar 302.4 KB 10/04/2017
hadoop-azure-datalake.jar 30.5 KB 10/04/2017
hadoop-azure-datalake-2.6.0-cdh5.13.0.jar 30.5 KB 10/04/2017
hadoop-common.jar 3.4 MB 10/04/2017
hadoop-common-2.6.0-cdh5.13.0.jar 3.4 MB 10/04/2017
hadoop-common-2.6.0-cdh5.13.0-tests.jar 2.0 MB 10/04/2017
hadoop-common-tests.jar 2.0 MB 10/04/2017
hadoop-nfs.jar 157.2 KB 10/04/2017
hadoop-nfs-2.6.0-cdh5.13.0.jar 157.2 KB 10/04/2017
parquet-avro.jar 104.0 KB 10/04/2017
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
ENG IN 9:16 AM 6/5/2021

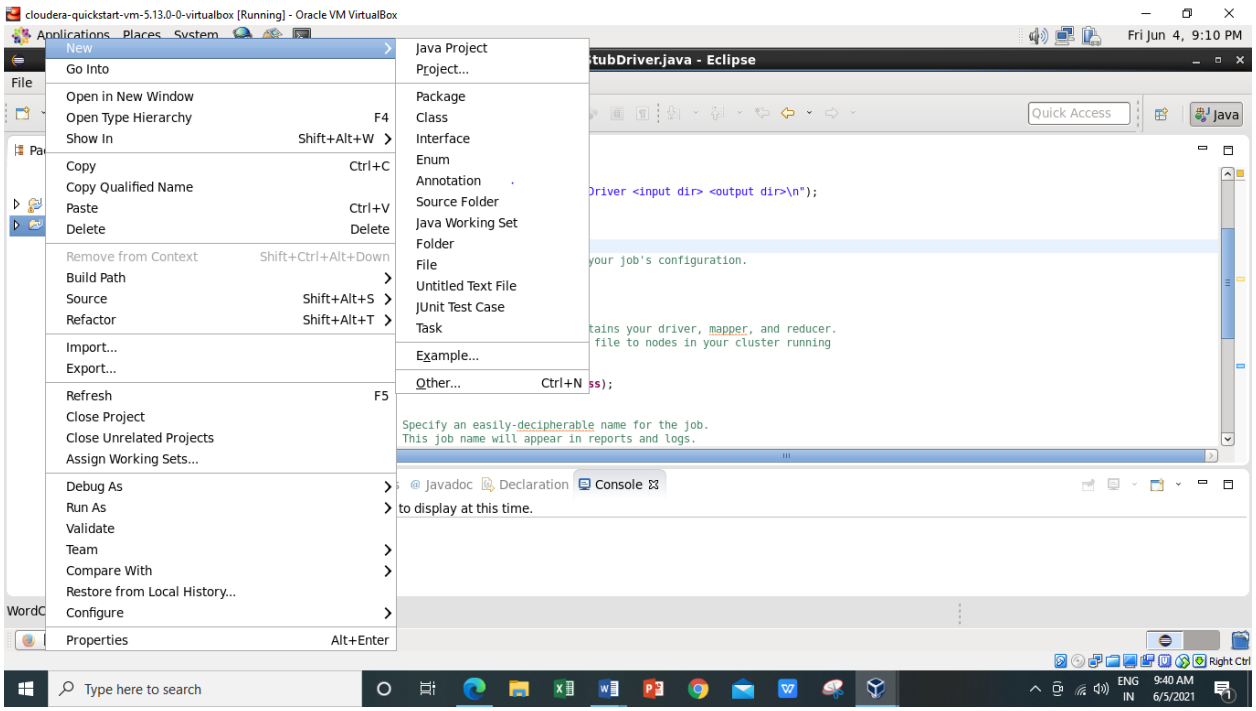

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:55 PM
New Java Project
Java Settings
Define the Java build settings.
Source Projects Libraries Order and Export
JARs and class folders on the build path:
hadoop-annotations-2.6.0-cdh5.13.0.jar - /usr/lib
hadoop-annotations.jar - /usr/lib/hadoop
hadoop-auth-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-auth.jar - /usr/lib/hadoop
hadoop-aws-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
hadoop-aws.jar - /usr/lib/hadoop
hadoop-azure-datalake-2.6.0-cdh5.13.0.jar - /usr
hadoop-azure-datalake.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.13.0-tests.jar - /usr/
hadoop-common-2.6.0-cdh5.13.0.jar - /usr/lib/ha
hadoop-common-tests.jar - /usr/lib/hadoop
hadoop-common.jar - /usr/lib/hadoop
hadoop-nfs-2.6.0-cdh5.13.0.jar - /usr/lib/hadoop
Add JARs...
Add External JARs...
Add Variable...
Add Library...
Add Class Folder...
Add External Class Folder...
Edit...
Remove
Migrate JAR File...
< Back
Next >
Cancel
Finish
Package Explorer
training
WordCount
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
ENG IN 9:25 AM 6/5/2021

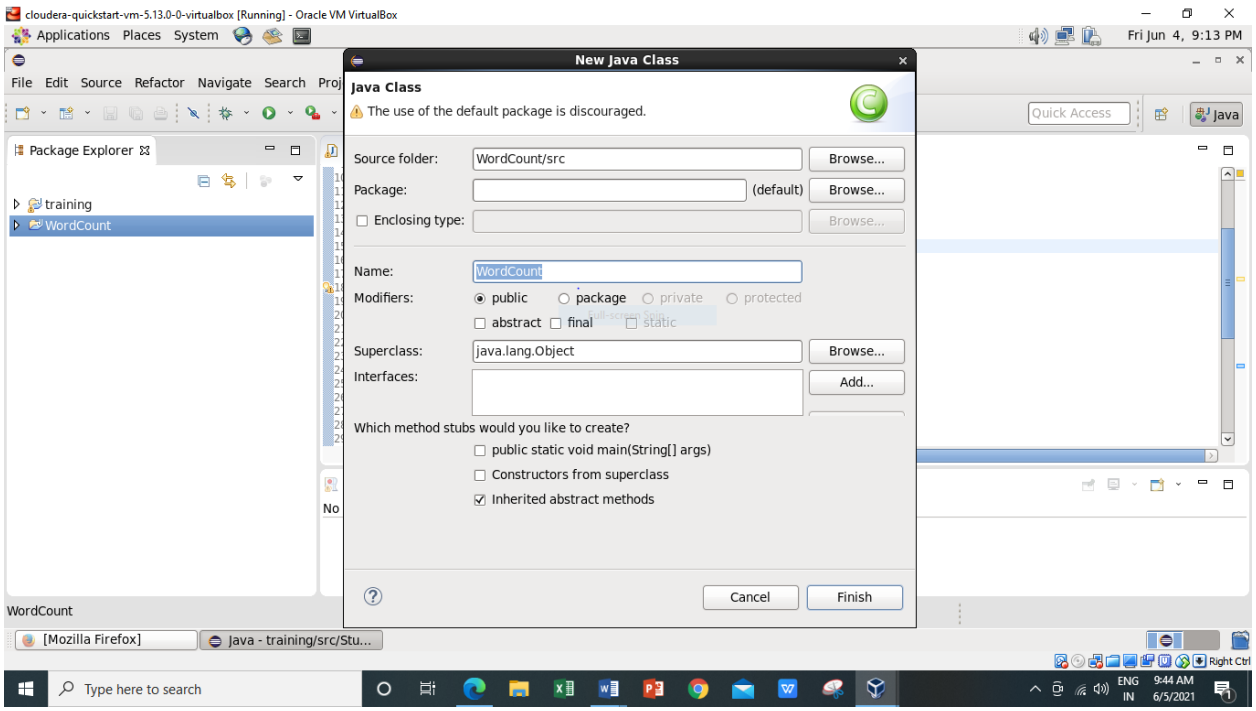

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 8:58 PM
New Java Project
JAR Selection
usr lib hadoop
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
bin 10/23/2017
client 10/23/2017
client-0.20 10/23/2017
cloudera 10/23/2017
etc 10/23/2017
lib 10/23/2017
libexec 10/23/2017
sbin 10/23/2017
hadoop-annotations.jar 21.0 KB 10/04/2017
hadoop-annotations-2.6.0-cdh5.13.0.jar 21.0 KB 10/04/2017
hadoop-auth.jar 74.1 KB 10/04/2017
hadoop-auth-2.6.0-cdh5.13.0.jar 74.1 KB 10/04/2017
hadoop-aws.jar 302.4 KB 10/04/2017
hadoop-aws-2.6.0-cdh5.13.0.jar 302.4 KB 10/04/2017
hadoop-azure-datalake.jar 30.5 KB 10/04/2017
hadoop-azure-datalake-2.6.0-cdh5.13.0.jar 30.5 KB 10/04/2017
hadoop-common.jar 3.4 MB 10/04/2017
hadoop-common-2.6.0-cdh5.13.0.jar 3.4 MB 10/04/2017
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
Quick Access
Java
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
ENG IN 9:28 AM 6/5/2021

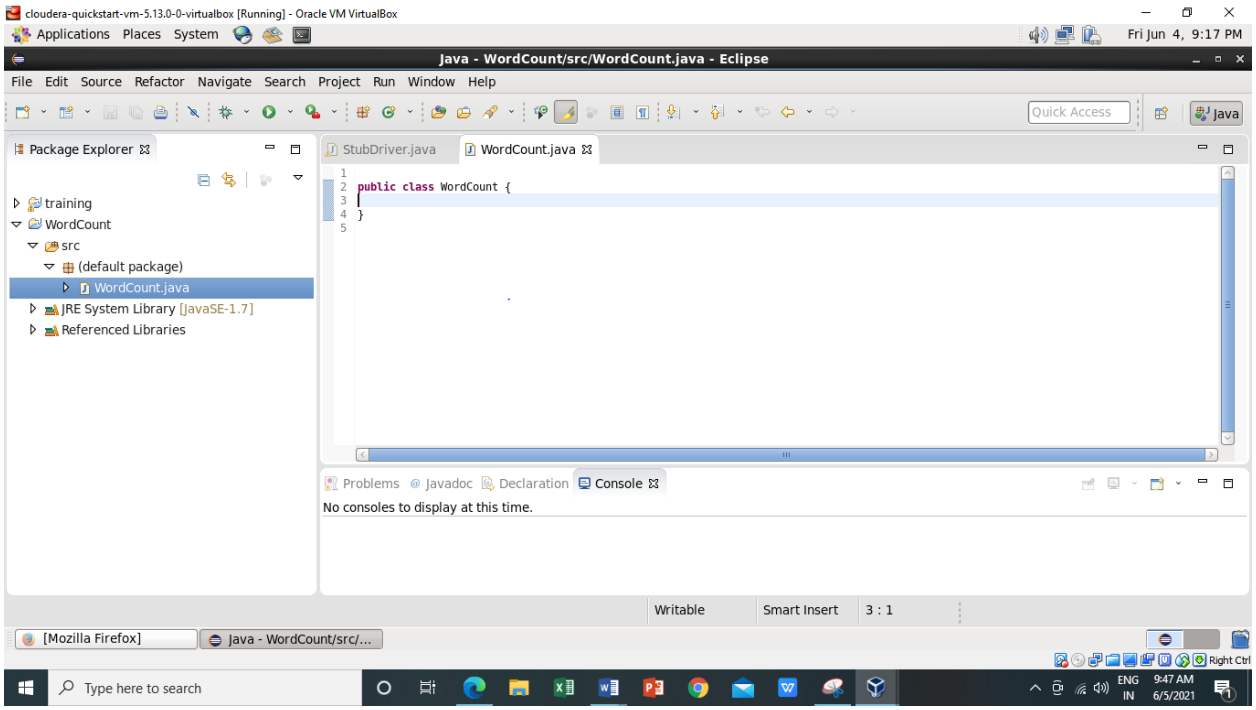

cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 9:01 PM
New Java Project
JAR Selection
usr lib hadoop client
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name Size Modified
activation.jar 61.5 KB 10/04/2017
activation-1.1.jar 61.5 KB 10/04/2017
apacheds-i18n.jar 43.9 KB 10/04/2017
apacheds-i18n-2.0.0-M15.jar 43.9 KB 10/04/2017
apacheds-kerberos-codec.jar 675.3 KB 10/04/2017
apacheds-kerberos-codec-2.0.0-M15.jar 675.3 KB 10/04/2017
api-asn1-api.jar 16.2 KB 10/04/2017
api-asn1-api-1.0.0-M20.jar 16.2 KB 10/04/2017
api-util.jar 78.0 KB 10/04/2017
api-util-1.0.0-M20.jar 78.0 KB 10/04/2017
avro.jar 465.8 KB 10/04/2017
aws-java-sdk-bundle.jar 61.0 MB 10/04/2017
aws-java-sdk-bundle-1.11.134.jar 61.0 MB 10/04/2017
azure-data-lake-store-sdk.jar 86.2 KB 10/04/2017
azure-data-lake-store-sdk-2.2.3.jar 86.2 KB 10/04/2017
commons-beanutils.jar 228.4 KB 10/04/2017
commons-beanutils-1.9.2.jar 228.4 KB 10/04/2017
commons-beanutils-core.jar 201.2 KB 10/04/2017
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
Quick Access
Java
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
ENG IN 9:31 AM 6/5/2021

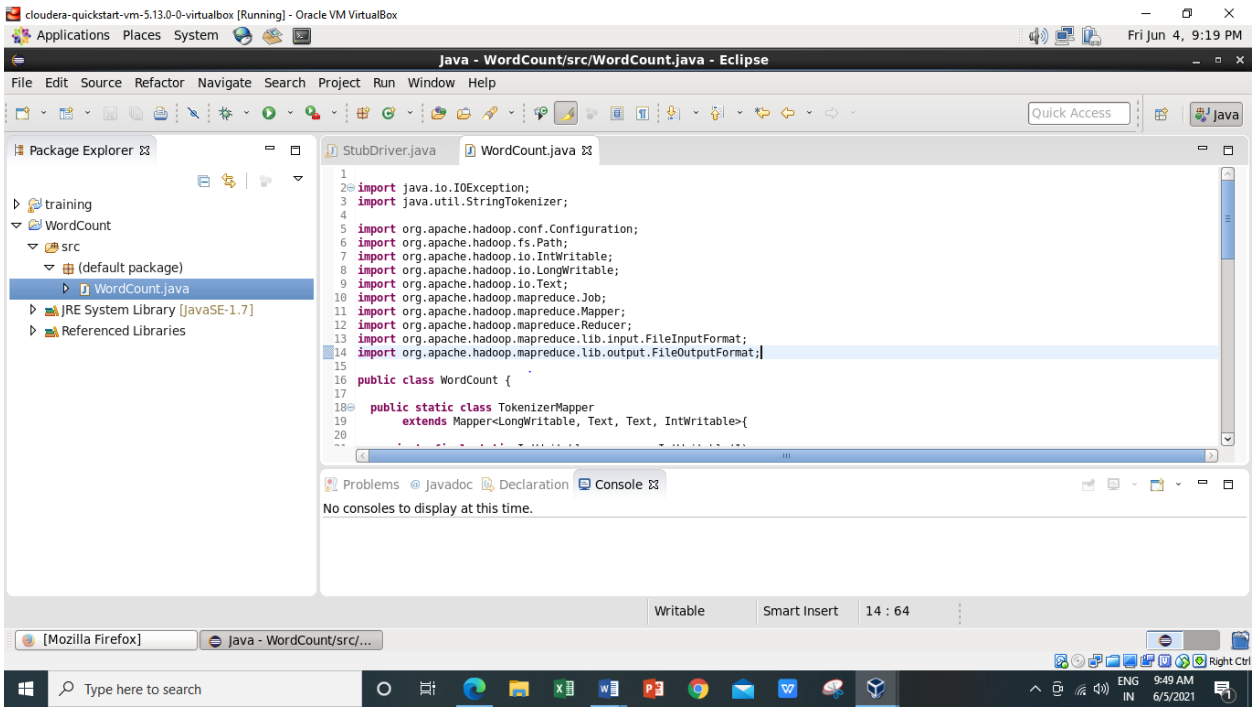


5) Right Click on the name of Project “WordCount” -> New -> class Don’t write anything for package Write Name Textbox write “WordCount” -> Finish Then WordCount.java window will pop up




cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 9:13 PM
New Java Class
Java Class
The use of the default package is discouraged.
Source folder: WordCount/src
Package:
(default)
Enclosing type:
Name: WordCount
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Cancel
Finish
Package Explorer
training
WordCount
[Mozilla Firefox]
Java - training/src/Stu...
Type here to search
9:44 AM
6/5/2021


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 9:17 PM
Java - WordCount/src/WordCount.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
StubDriver.java
WordCount.java
public class WordCount {
}
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Problems Javadoc Declaration Console
No consoles to display at this time.
Writable
Smart Insert
3 : 1
[Mozilla Firefox]
Java - WordCount/src/...
Type here to search
9:47 AM
6/5/2021



Source code:

Packages

```
1  import java.io.IOException;
2  import java.util.StringTokenizer;
3
4  import org.apache.hadoop.conf.Configuration;
5  import org.apache.hadoop.fs.Path;
6  import org.apache.hadoop.io.IntWritable;
7  import org.apache.hadoop.io.LongWritable;
8  import org.apache.hadoop.io.Text;
9  import org.apache.hadoop.mapreduce.Job;
0  import org.apache.hadoop.mapreduce.Mapper;
1  import org.apache.hadoop.mapreduce.Reducer;
2  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
3  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
4
```

Mapper Logic

```
15
16 public class WordCount {
17
18   public static class TokenizerMapper
19        extends Mapper<LongWritable, Text, Text, IntWritable>{
20
21     private final static IntWritable one = new IntWritable(1);
22     private Text word = new Text();
23
24     public void map(LongWritable key, Text value, Context context
25                     ) throws IOException, InterruptedException {
26       StringTokenizer itr = new StringTokenizer(value.toString());
27       while (itr.hasMoreTokens()) {
28         word.set(itr.nextToken());
29         context.write(word, one);
30       }
31     }
32   }
33
```

Reducer logic

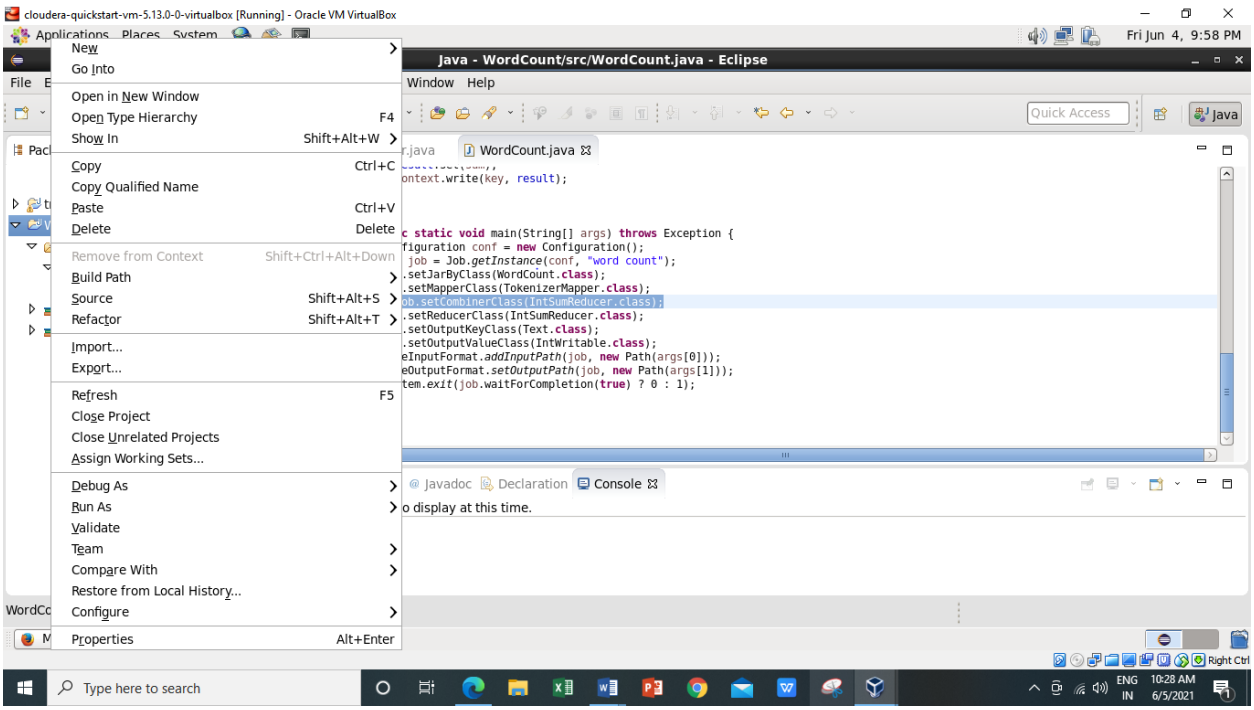
```
34   public static class IntSumReducer
35        extends Reducer<Text,IntWritable,Text,IntWritable> {
36     private IntWritable result = new IntWritable();
37
38     public void reduce(Text key, Iterable<IntWritable> values,
39                        Context context
40                        ) throws IOException, InterruptedException {
41       int sum = 0;
42       for (IntWritable val : values) {
43         sum += val.get();
44       }
45       result.set(sum);
46       context.write(key, result);
47     }
48   }
```

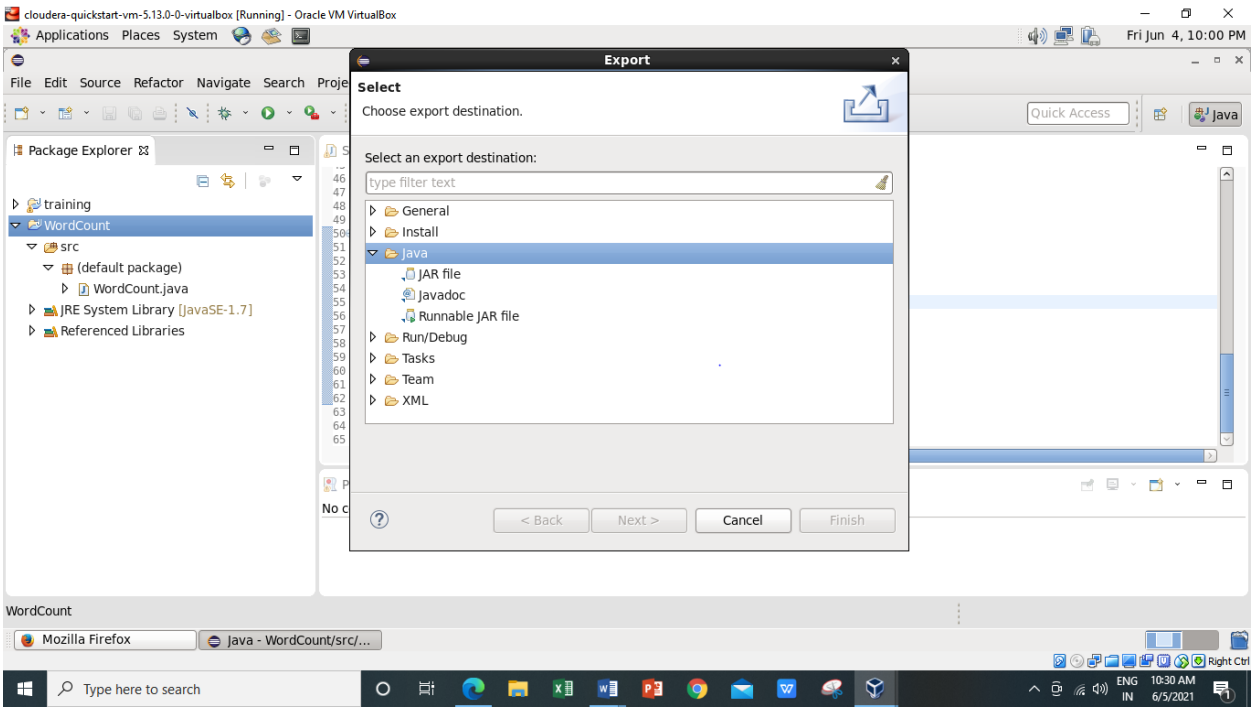
Main function

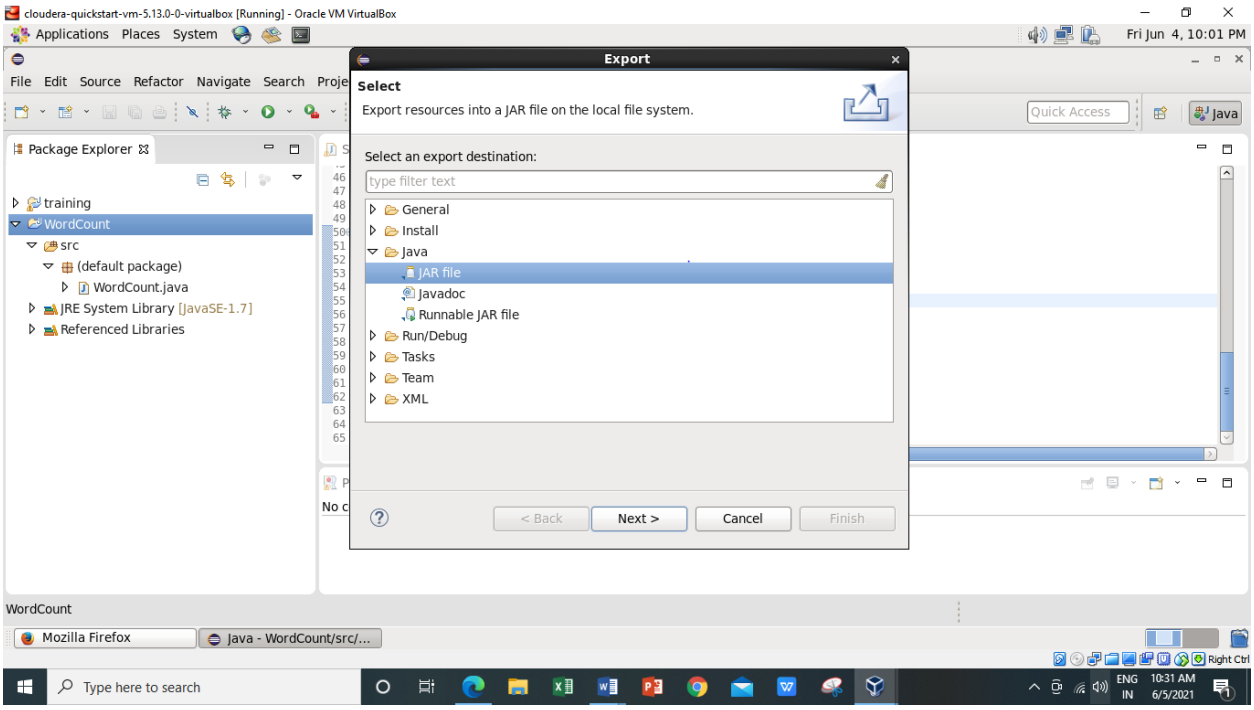
We are running the code without combiner. That is why we commented the combiner line in main function.

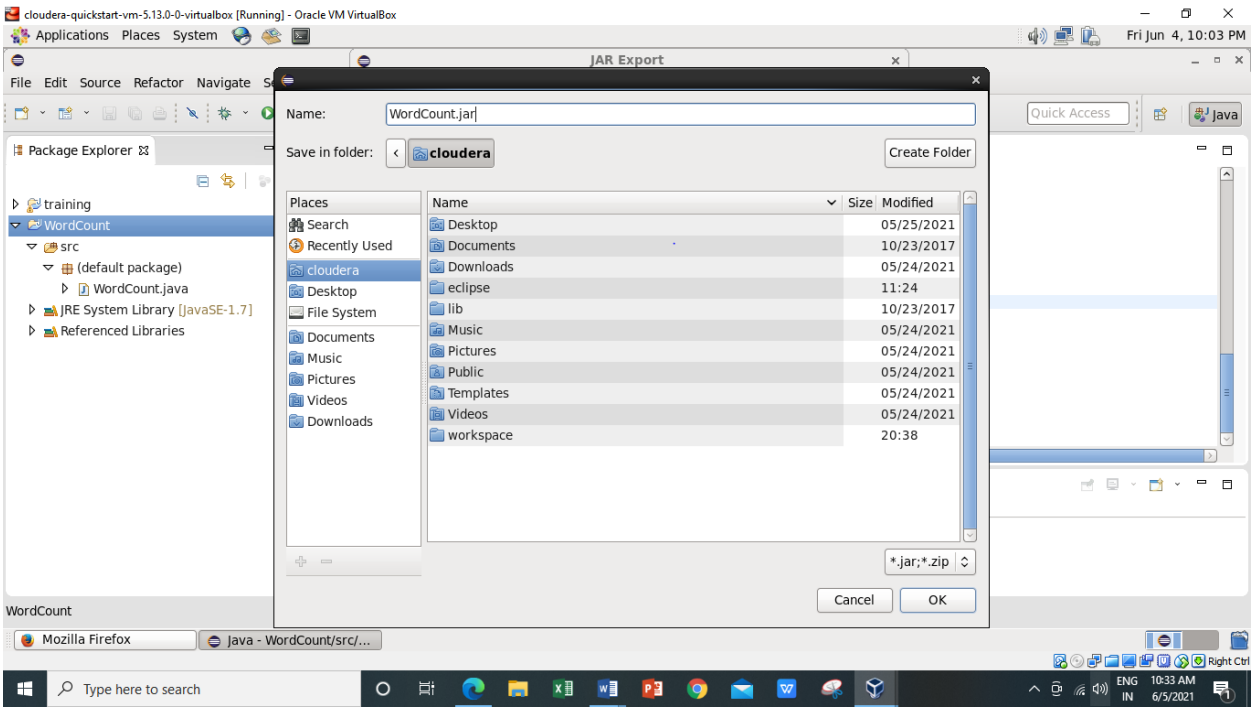
```
50  public static void main(String[] args) throws Exception {
51    Configuration conf = new Configuration();
52    Job job = Job.getInstance(conf, "word count");
53    job.setJarByClass(WordCount.class);
54    job.setMapperClass(TokenizerMapper.class);
55    //job.setCombinerClass(IntSumReducer.class);
56    job.setReducerClass(IntSumReducer.class);
57    job.setOutputKeyClass(Text.class);
58    job.setOutputValueClass(IntWritable.class);
59    FileInputFormat.addInputPath(job, new Path(args[0]));
60    FileOutputFormat.setOutputPath(job, new Path(args[1]));
61    System.exit(job.waitForCompletion(true) ? 0 : 1);
62  }
63 }
64
65
```

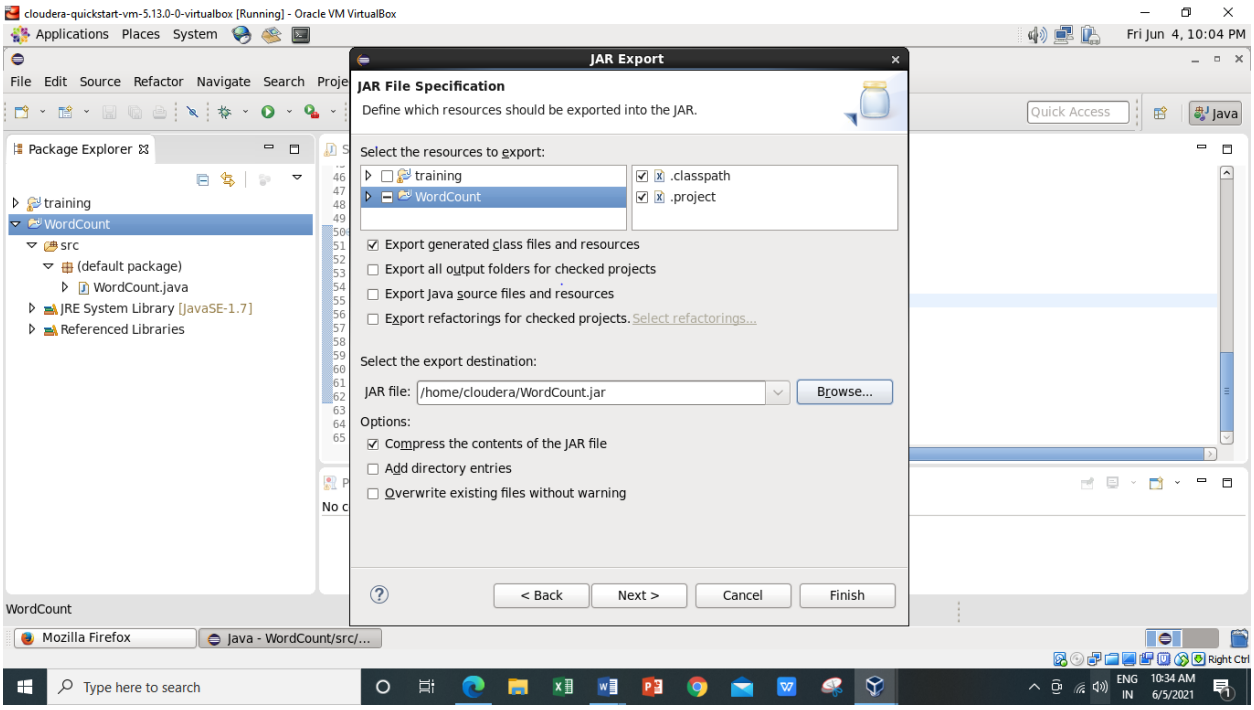
6) Right Click on the project name WordCount -> Export -> Java -> JAR File -> Next -> for select the export destination for JAR file: browse -> Name : WordCount.jar -> save in folder -> cloudera -> Finish -> OK




cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 10:00 PM
File Edit Source Refactor Navigate Search Proje
Quick Access
Java
Package Explorer
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Export
Select
Choose export destination.
Select an export destination:
type filter text
General
Install
Java
JAR file
Javadoc
Runnable JAR file
Run/Debug
Tasks
Team
XML
< Back
Next >
Cancel
Finish
WordCount
Mozilla Firefox
Java - WordCount/src/...
Type here to search
ENG
IN
10:30 AM
6/5/2021
Right Ctrl


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 10:01 PM
File Edit Source Refactor Navigate Search Proje
Quick Access
Java
Package Explorer
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Export
Select
Export resources into a JAR file on the local file system.
Select an export destination:
type filter text
General
Install
Java
JAR file
Javadoc
Runnable JAR file
Run/Debug
Tasks
Team
XML
< Back
Next >
Cancel
Finish
WordCount
Mozilla Firefox
Java - WordCount/src/...
Type here to search
ENG
IN
10:31 AM
6/5/2021
Right Ctrl



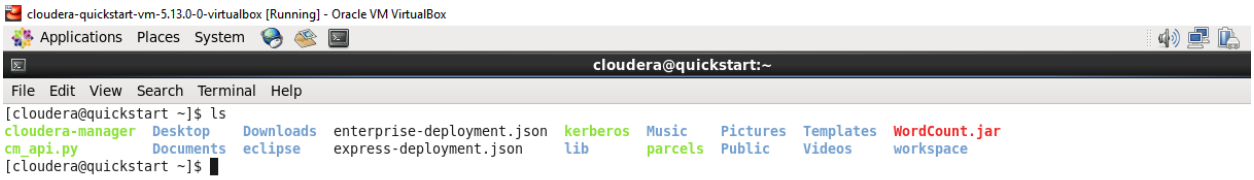


7) Verify jar file from terminal by using Open terminal & type " ls " There it will show WordCount.jar

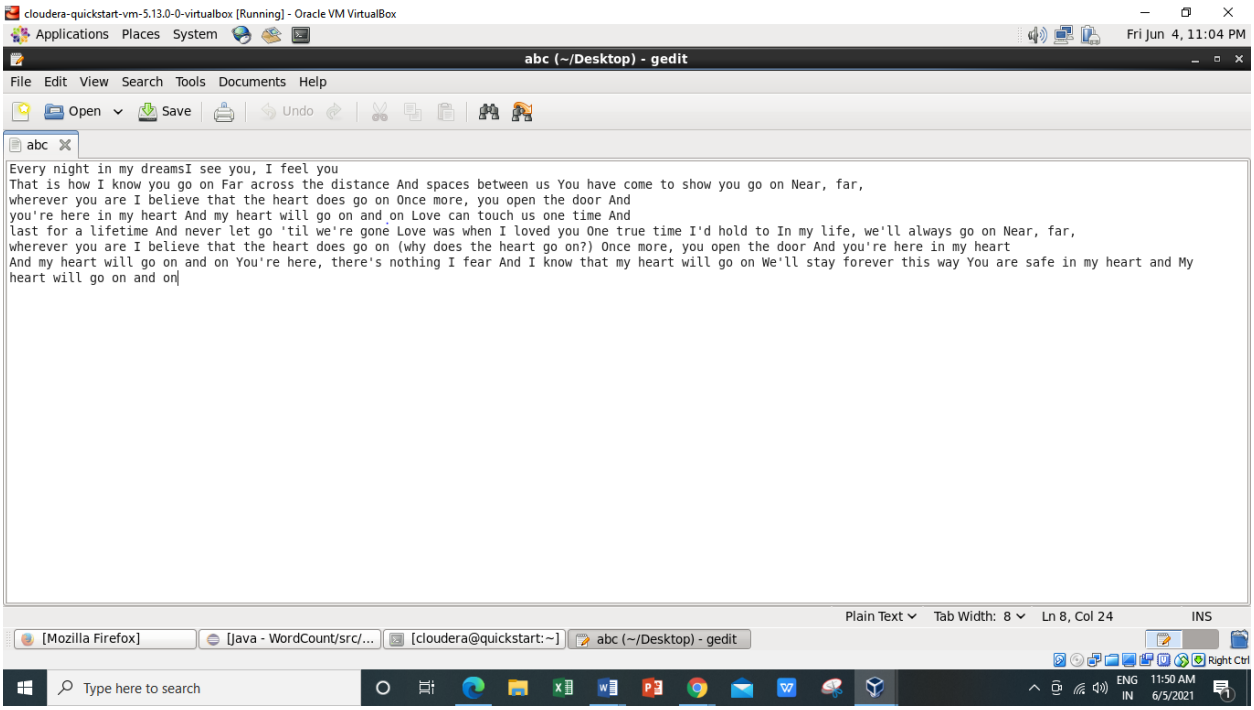
Check current working directory

->pwd

```
[cloudera@quickstart ~]$ pwd
/home/cloudera
[cloudera@quickstart ~]$
```



8) We need to create an input file in local file system
   Creating an input file named as **"abc".**



Here listing all the directory present in hdfs using **hdfs dfs -ls /** command

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 10 items
-rw-r--r--   1 cloudera supergroup         27 2021-05-24 12:04 /Sample_01
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:20 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:36 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

9) Now we have to move this input file to hdfs. For this we create a direcory on hdfs using command hdfs dfs -mkdir /inputnew.

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir /inputdir
[cloudera@quickstart ~]$
```

Then we can verify whether this directory is created or not using ls command **hdfs dfs -ls /**

```
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 11 items
-rw-r--r--   1 cloudera supergroup         27 2021-05-24 12:04 /Sample_01
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-04 23:34 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:20 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:36 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

Move the input file to this directory created in hdfs by using either put command or copyFromLocal command.

```
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Desktop/abc /inputdir/
[cloudera@quickstart ~]$
```

Now checking whether the **“abc”** present in /inputdir directory of hdfs or not using **hdfs dfs -ls /inputdir** command

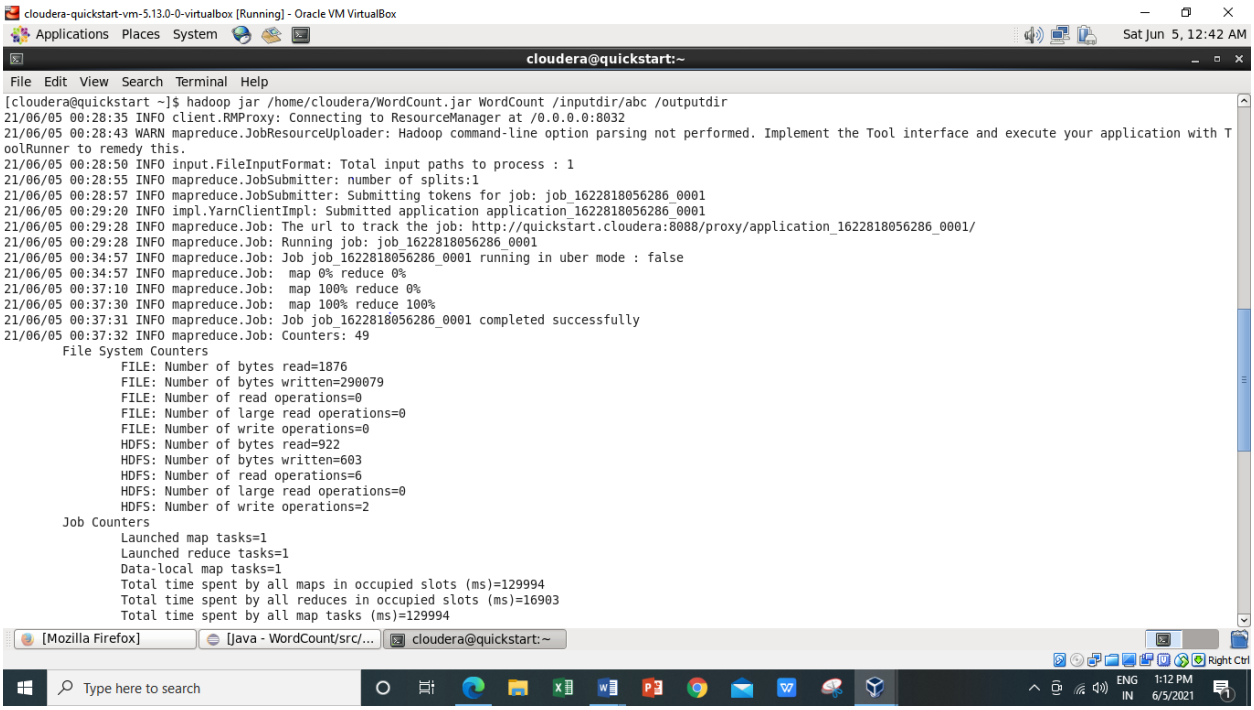
```
[cloudera@quickstart ~]$ hdfs dfs -ls /inputdir
Found 1 items
-rw-r--r--   1 cloudera supergroup        813 2021-06-05 00:06 /inputdir/abc
[cloudera@quickstart ~]$
```

As we can see **“abc”** file is present in /inputdir directory of hdfs. Now we will see the content of this file using **hdfs dfs –cat /inputdir/abc** command

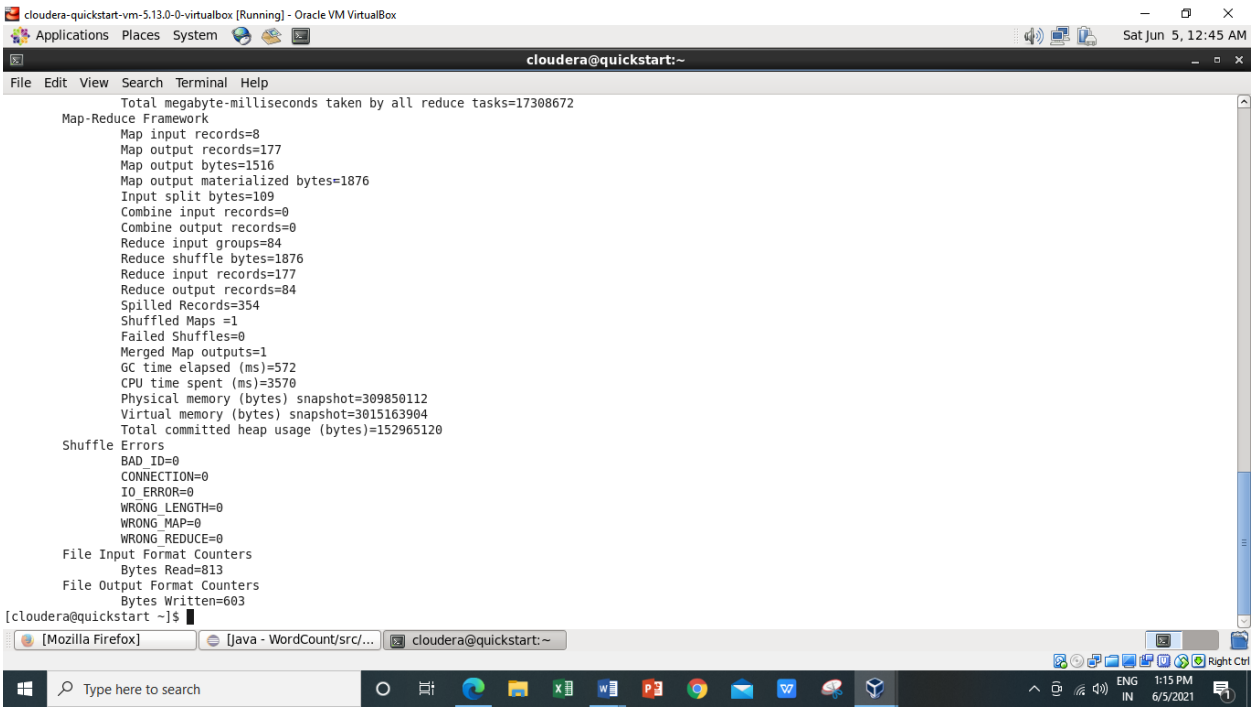
```
[cloudera@quickstart ~]$ hdfs dfs -cat /inputdir/abc
Every night in my dreamsI see you, I feel you
That is how I know you go on Far across the distance And spaces between us You have come to show you go on Near, far,
wherever you are I believe that the heart does go on Once more, you open the door And
you're here in my heart And my heart will go on and on Love can touch us one time And
last for a lifetime And never let go 'til we're gone Love was when I loved you One true time I'd hold to In my life, we'll always go on Near, far,
wherever you are I believe that the heart does go on (why does the heart go on?) Once more, you open the door And you're here in my heart
And my heart will go on and on You're here, there's nothing I fear And I know that my heart will go on We'll stay forever this way You are safe in my heart and My
heart will go on and on
[cloudera@quickstart ~]$
```

10) Running Mapreduce Program on Hadoop, syntax is hadoop jar jarFileName.jar ClassName /InputFileAddress /outputdir

i.e. **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /outputdir**



Map-Reduce Framework



As we can see in the above output,
Combine input records=0
Combine output records=0
We are getting this because we have commented the Combiner line in main function.

And Reduce shuffle bytes coming as,
Reduce shuffle bytes=1876

So when we are not using combiner 1876 bytes acting as an input for the reducer.

11) Then we can verify the content of outputdir directory and in that part-r file has the actual output by using the command Hdfs dfs -cat /outputdir/part-r-00000 This will give us final output. The same file can also be accessed using a browser. For every execution of this program we need to delete the output directory or give a new name to the output directory every time.

1st we are checking whether the outputdir directory is created in hdfs or not using command **hdfs dfs -ls /**

```
                Bytes Written=603
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 12 items
-rw-r--r--   1 cloudera supergroup         27 2021-05-24 12:04 /Sample_01
drwxrwxrwx   - hdfs     supergroup          0 2017-10-23 09:15 /benchmarks
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:58 /forcopy
drwxr-xr-x   - hbase    supergroup          0 2021-06-04 07:57 /hbase
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:06 /inputdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:20 /newdir
drwxr-xr-x   - cloudera supergroup          0 2021-06-05 00:37 /outputdir
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:36 /rjc
drwxr-xr-x   - cloudera supergroup          0 2021-05-24 13:55 /solr
drwxrwxrwt   - hdfs     supergroup          0 2021-05-24 10:39 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-10-23 09:17 /var
[cloudera@quickstart ~]$
```

Now let's check what we have inside this **outputdir** directory using command as **hdfs dfs -ls /outputdir**

```
[cloudera@quickstart ~]$ hdfs dfs -ls /outputdir
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2021-06-05 00:37 /outputdir/_SUCCESS
-rw-r--r--   1 cloudera supergroup        603 2021-06-05 00:37 /outputdir/part-r-00000
[cloudera@quickstart ~]$
```

Now we want to read the content of the **part-r-00000** file which present inside the **outputdir** using command **hdfs dfs -cat /outputdir/part-r-00000**

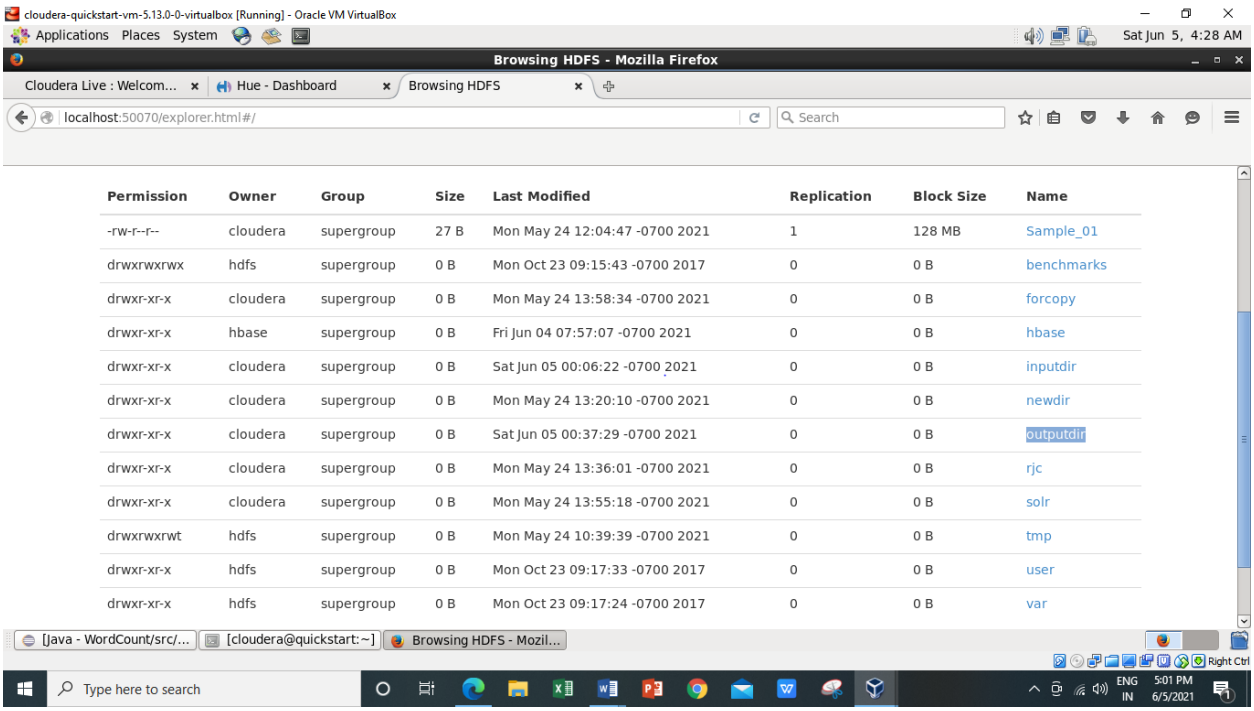
```
[cloudera@quickstart ~]$ hdfs dfs -cat /outputdir/part-r-00000
'til    1
(why    1
And     8
Every   1
Far     1
I       7
I'd     1
In      1
Love    2
My      1
Near,   2
Once    2
One     1
That    1
We'll   1
You     2
You're  1
a       1
across  1
always  1
and     4
are     3
believe 2
between 1
can     1
come    1
distance        1
does    3
door    2
dreamsI 1
```

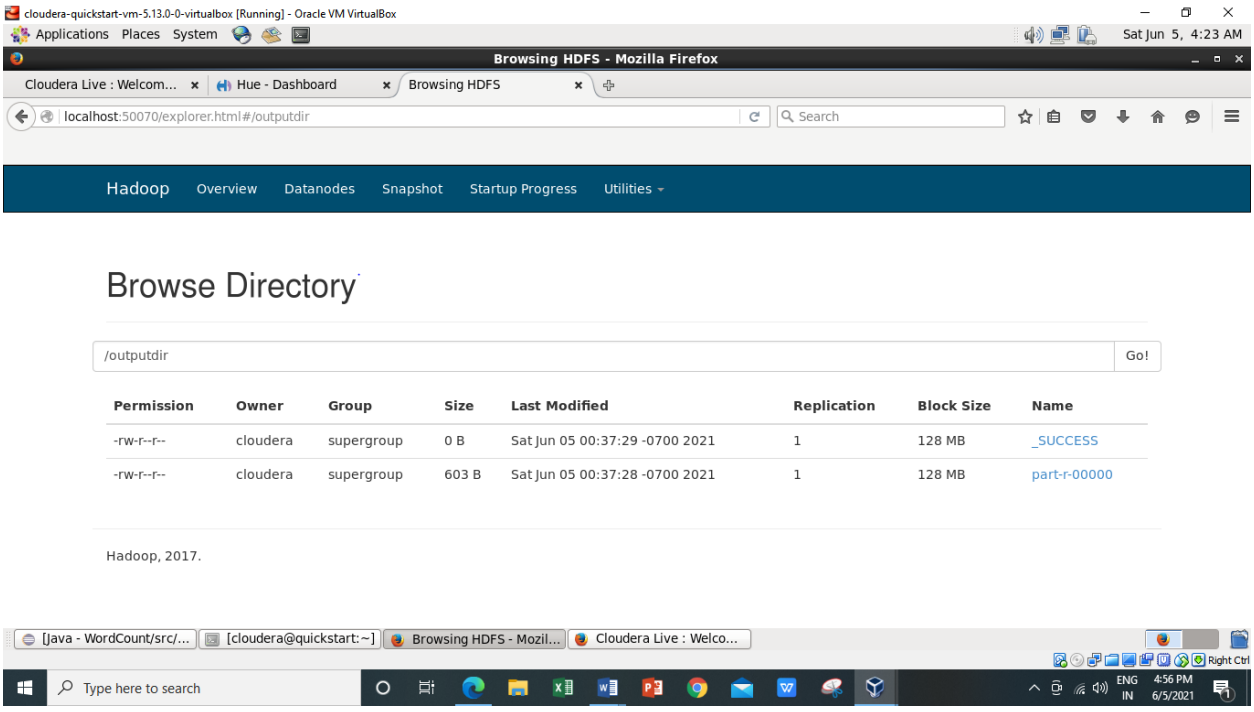
```
more,   2
my      8
never   1
night   1
nothing 1
on      12
on?)    1
one     1
open    2
safe    1
see     1
show    1
spaces  1
stay    1
that    3
the     6
there's 1
this    1
time    2
to      2
touch   1
true    1
us      2
was     1
way     1
we'll   1
we're   1
when    1
wherever        2
will    4
you     8
you're  2
you,    1
[cloudera@quickstart ~]$
```

It will give the count of number of times each word has occurred as output.

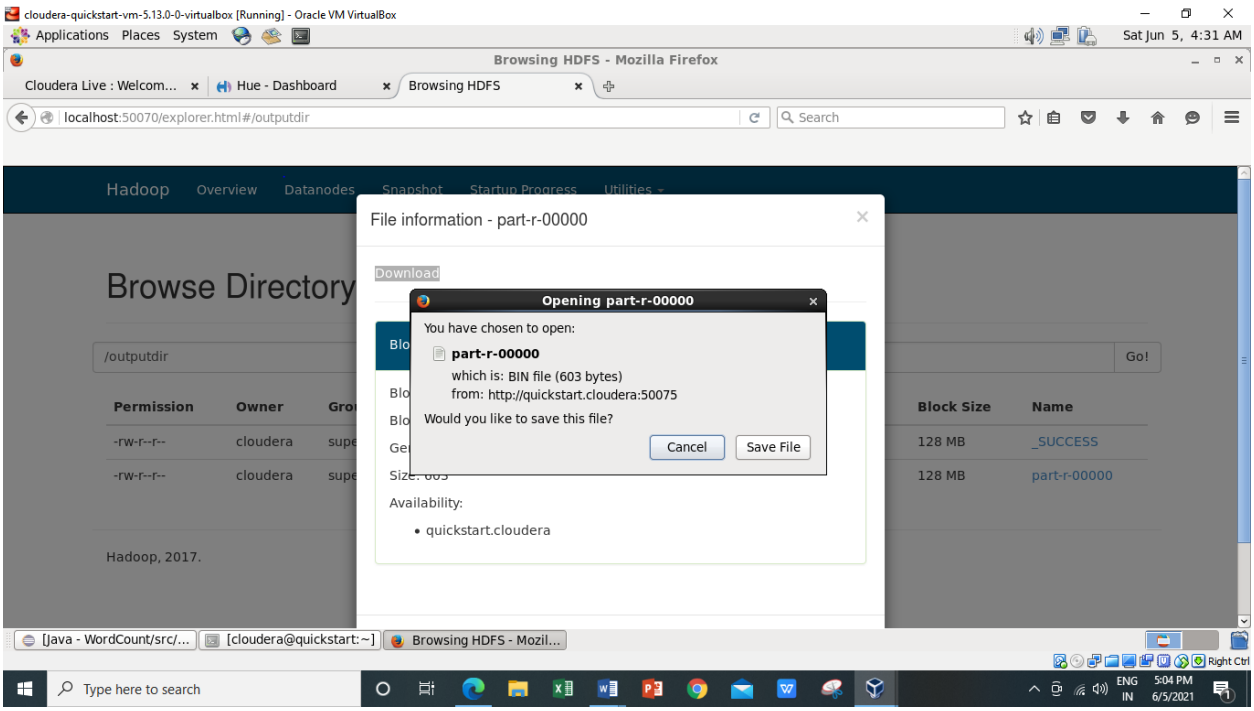
12) **The same file can also be accessed using a browser.**

Browse the Directory by
Hadoop->HDFS Namenode->Ultilities ->Browse the file system

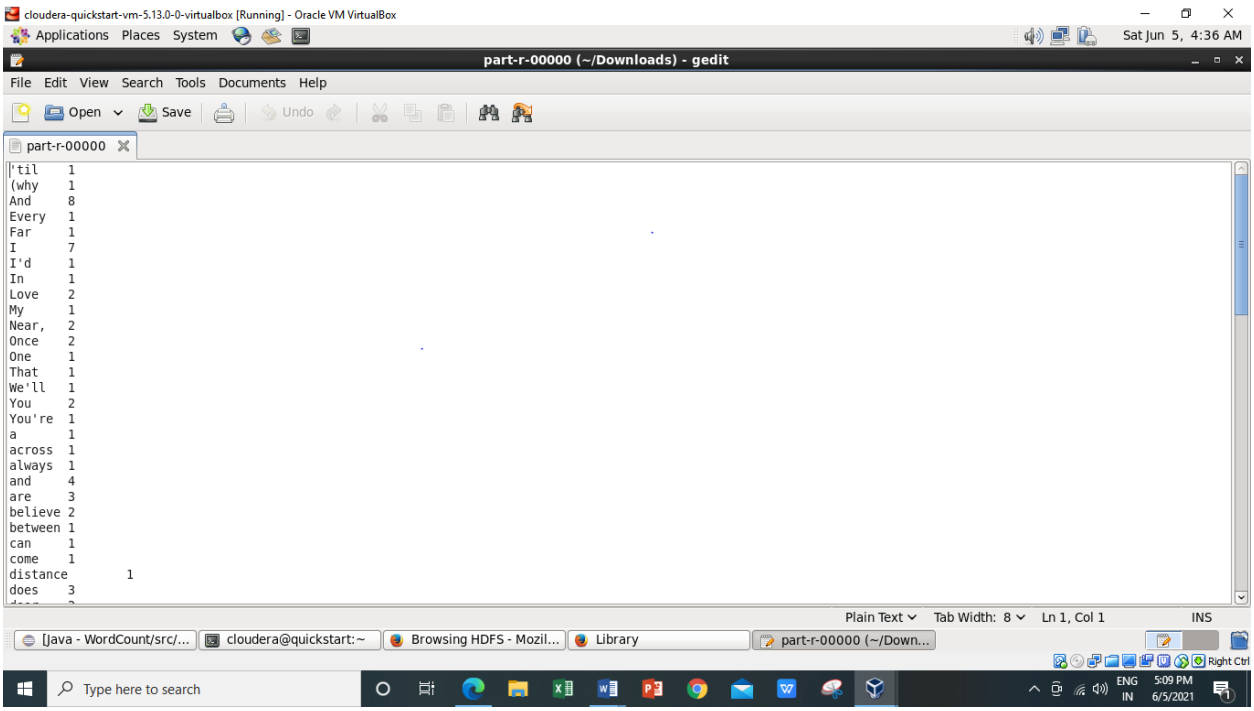


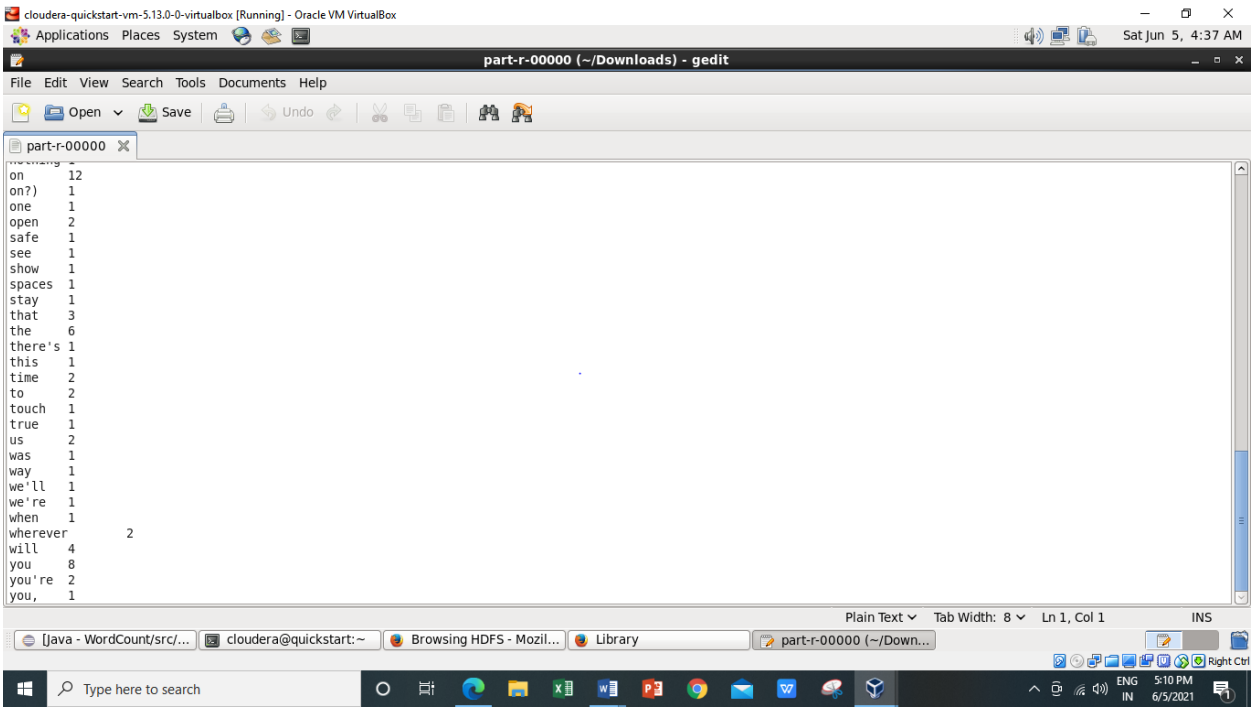


Now downloading the **part-r-00000** file**.**



Inside the **part-r-00000** file it will have the same output as we are getting after executing using command **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**

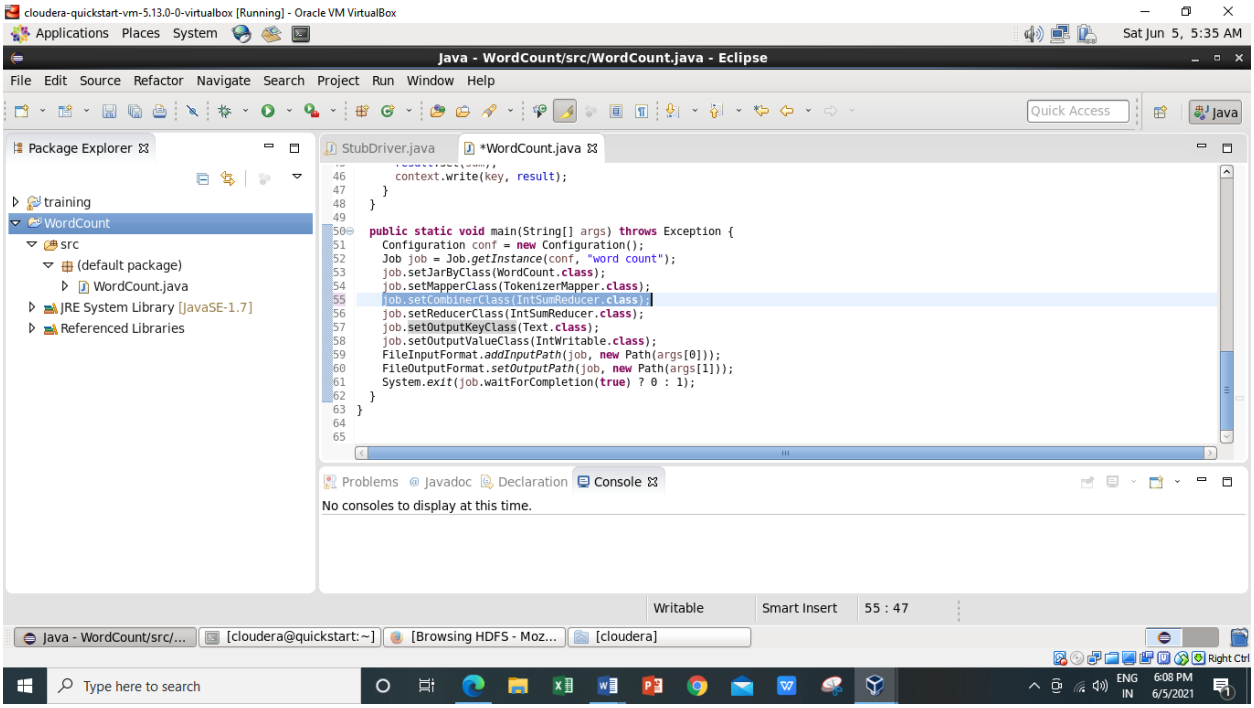




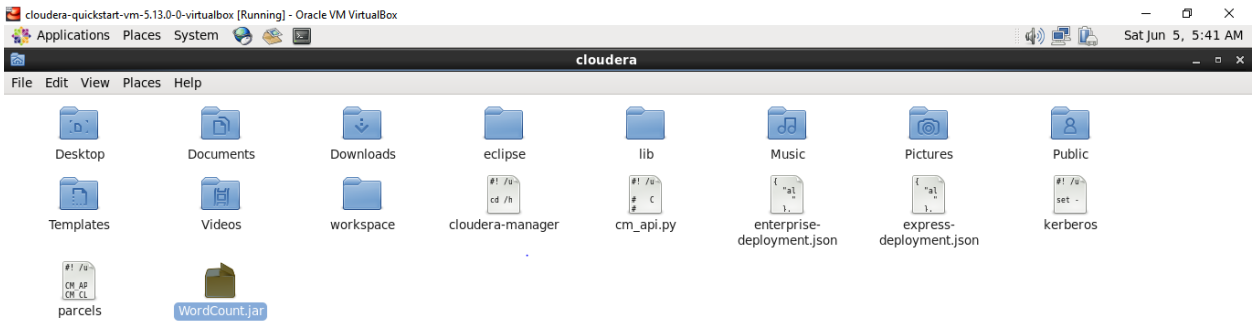
For every execution of this program we need to delete the output directory or give a new name to the output directory every time.

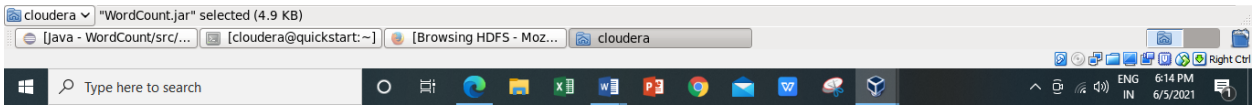
Implementation of WordCount problem using Hadoop MapReduce (With Combiner) in Eclipse:

1) We will perform the same steps as we have done above for WordCount (without using combiner) in that we just uncommenting the combiner line in main function.

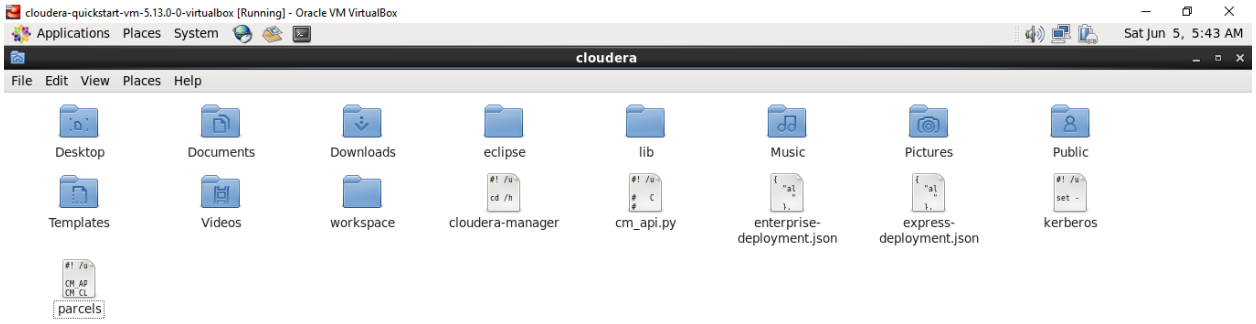


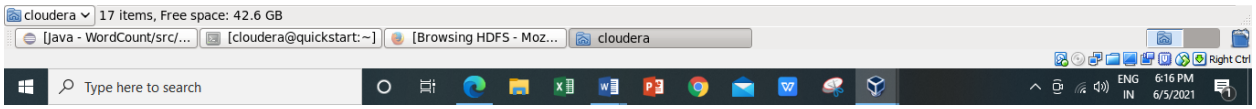
2) And will delete the WordCount.jar file in which all jar files are present from **/home/cloudera.**



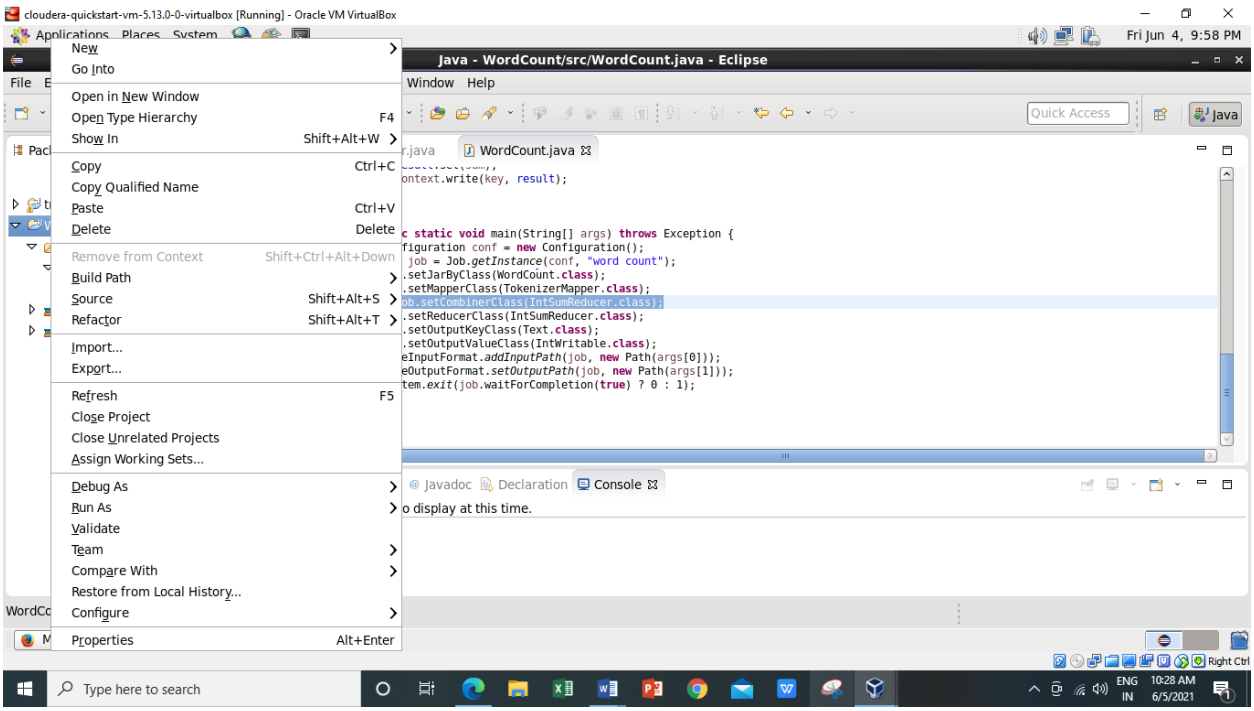


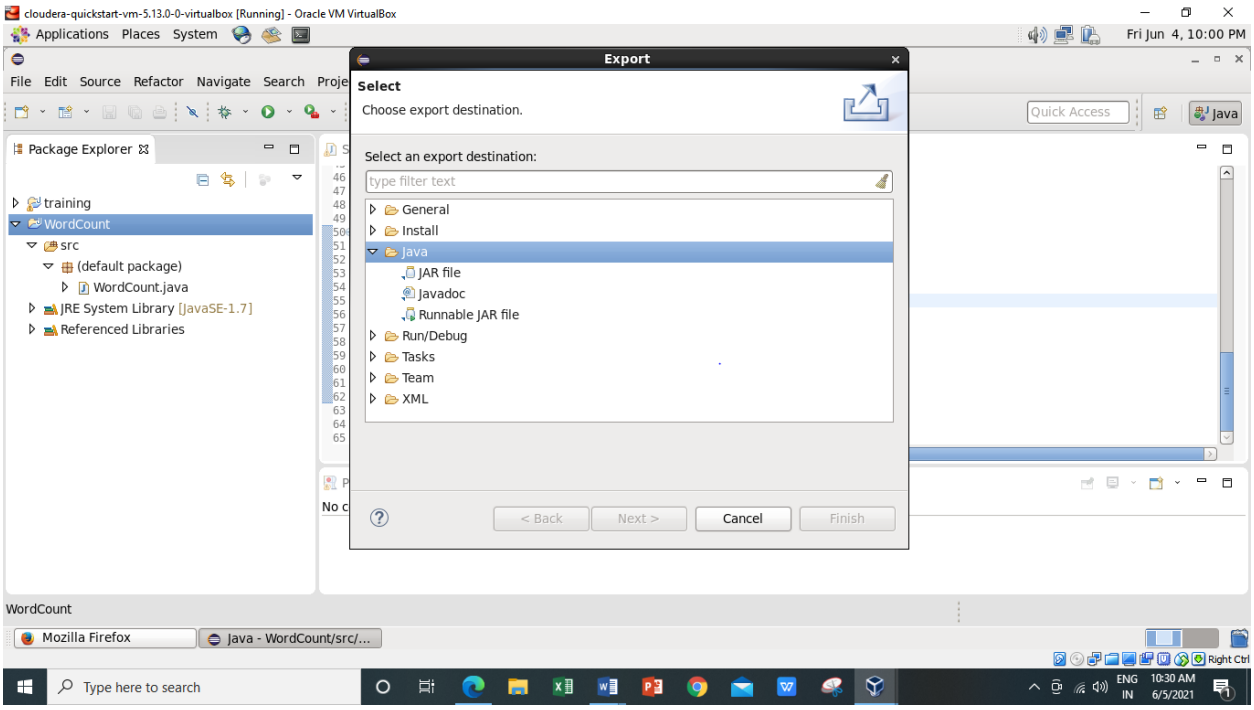
We have successfully deleted the WordCount.jar file

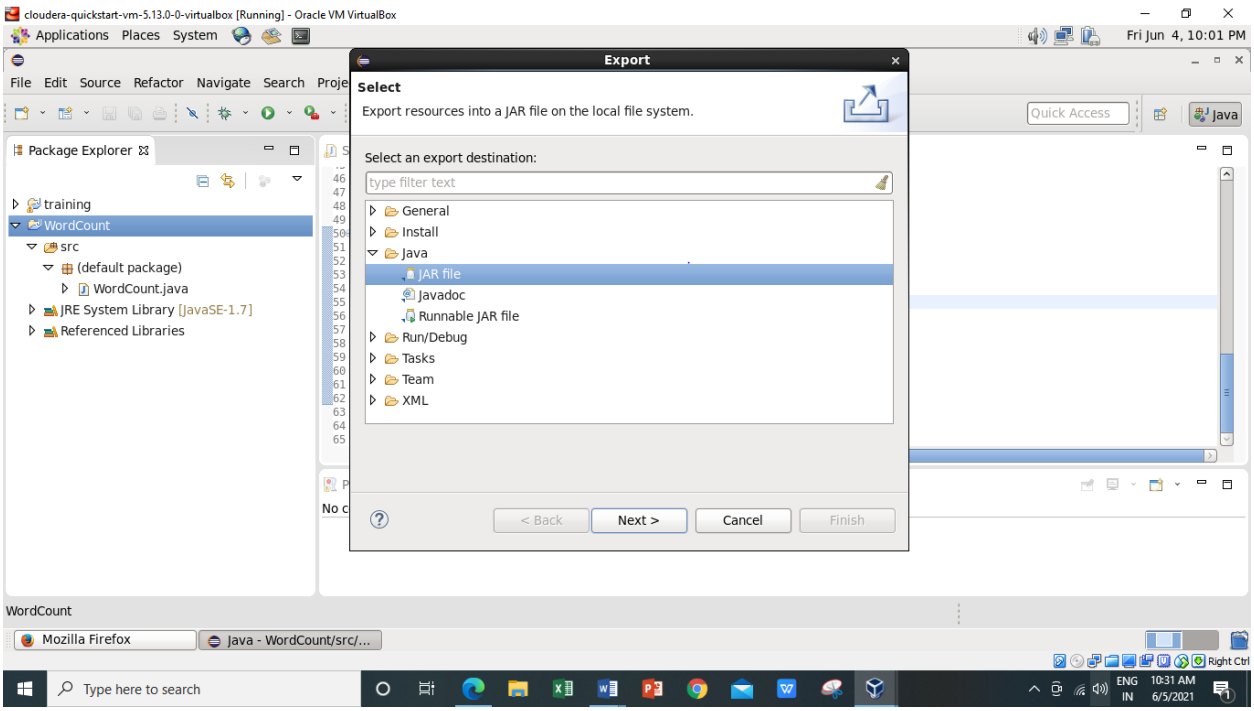


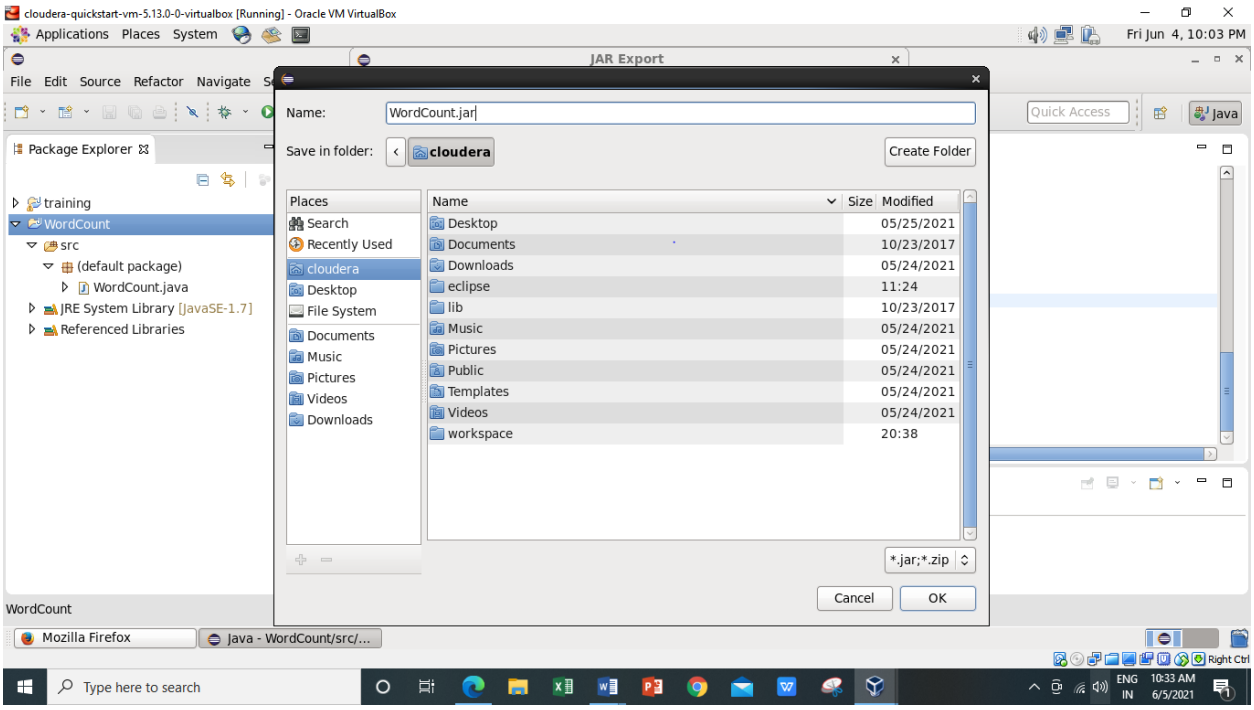


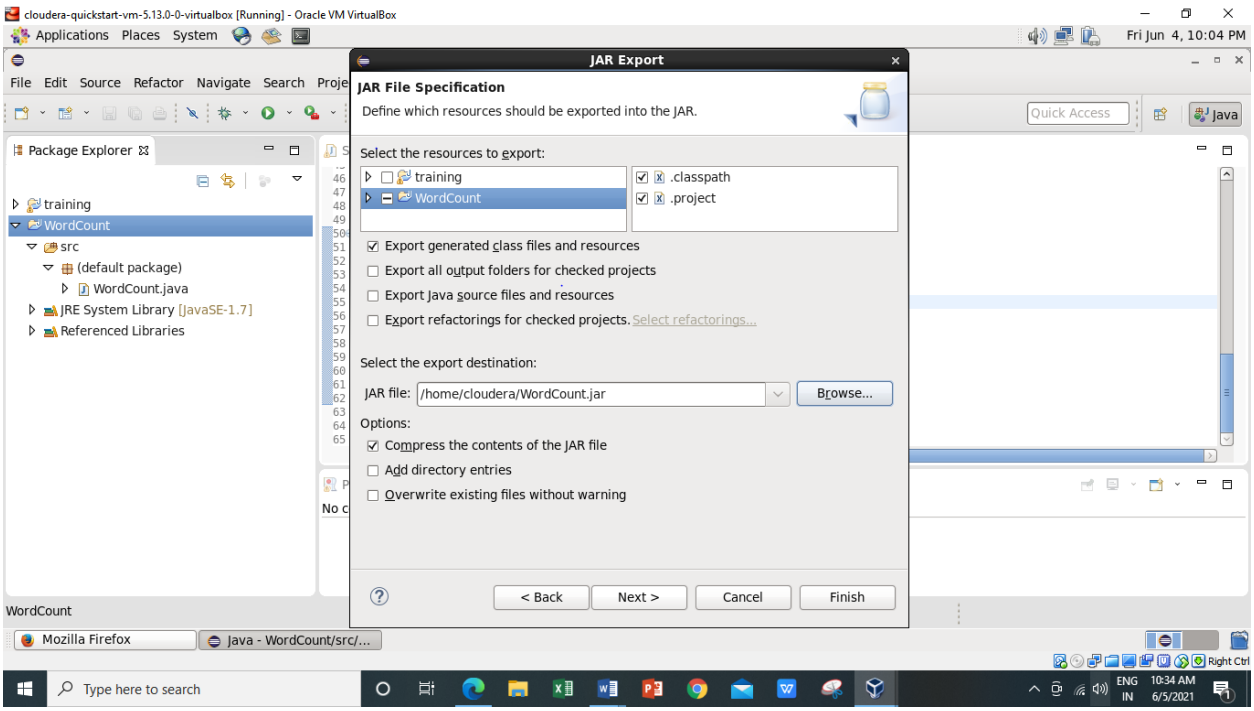
3) Now exporting the jar files Right Click on the project name WordCount -> Export -> Java -> JAR File -> Next -> for select the export destination for JAR file: browse -> Name : WordCount.jar -> save in folder -> cloudera -> Finish -> OK


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 9:58 PM
Java - WordCount/src/WordCount.java - Eclipse
File E
Window Help
Quick Access
Java
WordCount.java
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Shift+Alt+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Shift+Ctrl+Alt+Down
Build Path
Source Shift+Alt+S
Refactor Shift+Alt+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Debug As
Run As
Validate
Team
Compare With
Restore from Local History...
Configure
Properties Alt+Enter
Javadoc Declaration Console
o display at this time.
Type here to search
ENG IN
10:28 AM 6/5/2021


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 10:00 PM
File Edit Source Refactor Navigate Search
Quick Access
Java
Package Explorer
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Export
Select
Choose export destination.
Select an export destination:
type filter text
General
Install
Java
JAR file
Javadoc
Runnable JAR file
Run/Debug
Tasks
Team
XML
< Back
Next >
Cancel
Finish
WordCount
Mozilla Firefox
Java - WordCount/src/...
Type here to search
ENG IN
10:30 AM 6/5/2021


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 10:01 PM
Export
Select
Export resources into a JAR file on the local file system.
Select an export destination:
type filter text
General
Install
Java
JAR file
Javadoc
Runnable JAR file
Run/Debug
Tasks
Team
XML
< Back
Next >
Cancel
Finish
Package Explorer
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Quick Access
Java
WordCount
Mozilla Firefox
Java - WordCount/src/...
Type here to search
ENG IN
10:31 AM
6/5/2021


cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Fri Jun 4, 10:03 PM
JAR Export
Name:
WordCount.jar
Save in folder:
cloudera
Create Folder
Places
Search
Recently Used
cloudera
Desktop
File System
Documents
Music
Pictures
Videos
Downloads
Name
Size
Modified
Desktop 05/25/2021
Documents 10/23/2017
Downloads 05/24/2021
eclipse 11:24
lib 10/23/2017
Music 05/24/2021
Pictures 05/24/2021
Public 05/24/2021
Templates 05/24/2021
Videos 05/24/2021
workspace 20:38
.jar;.zip
Cancel
OK
Package Explorer
training
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
Quick Access
Java
WordCount
Mozilla Firefox
Java - WordCount/src/...
Type here to search
ENG IN
10:33 AM
6/5/2021



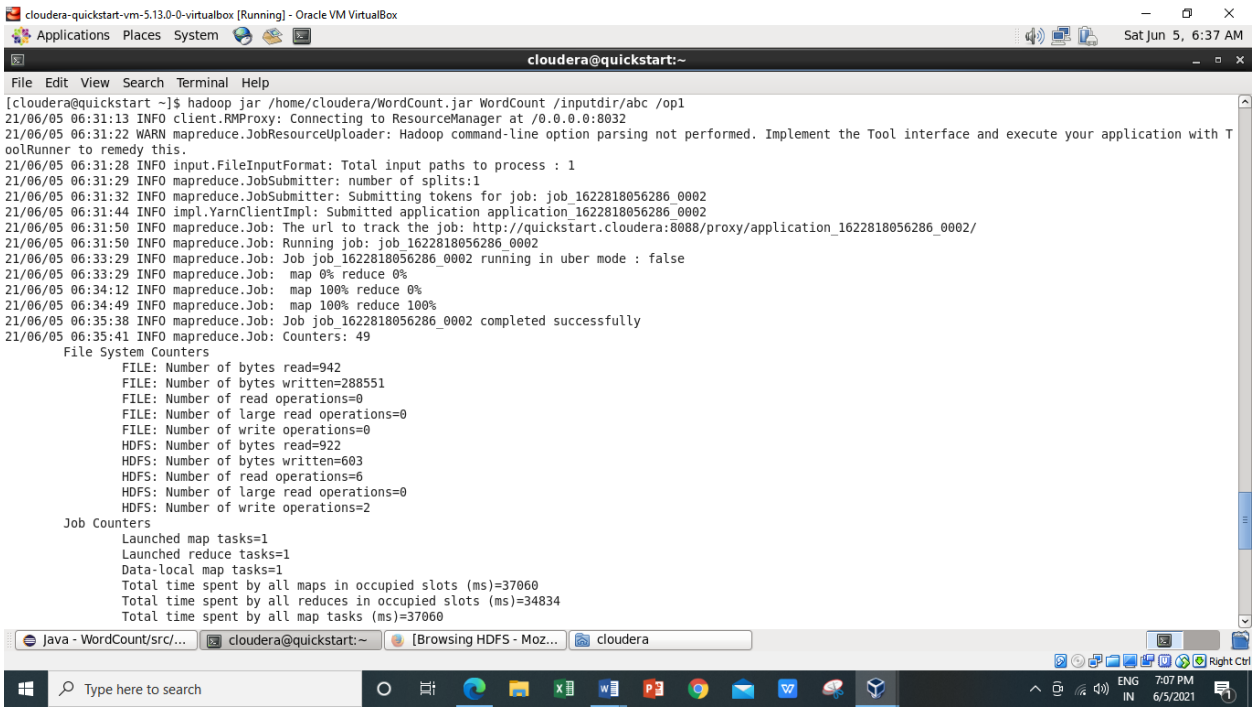
4) Now checking the WordCount.jar file is created or not using **–ls** command

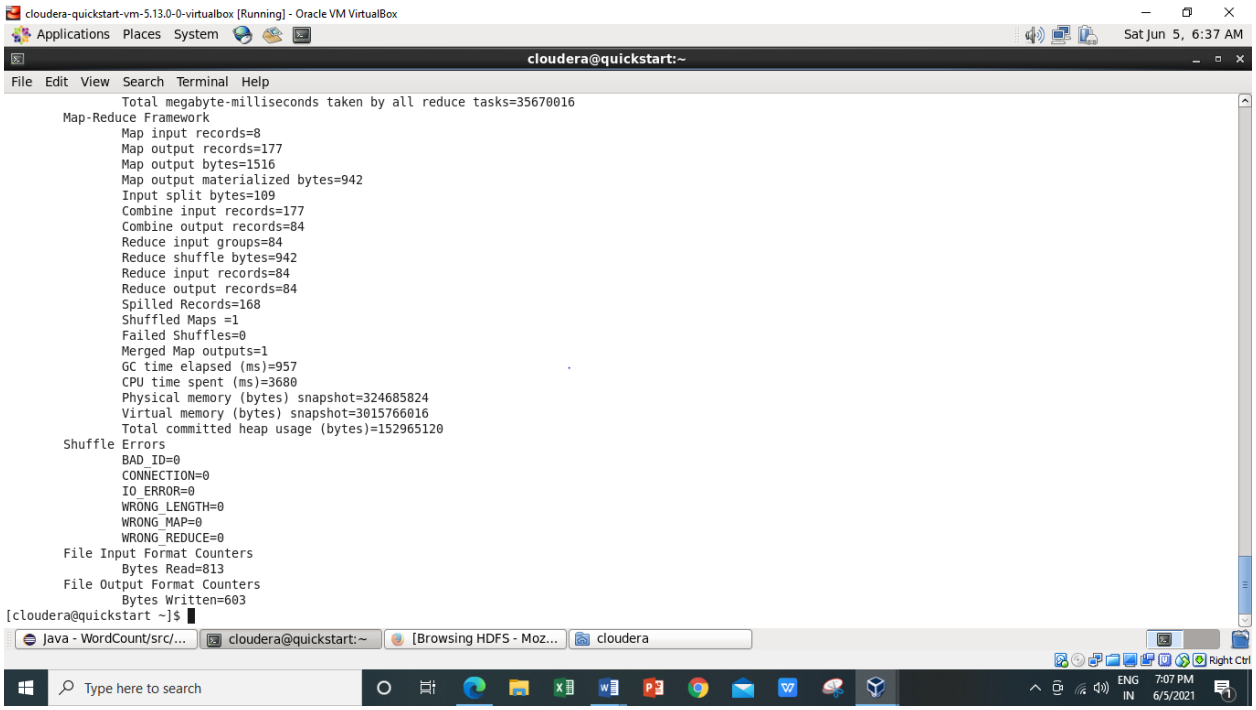
```
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    Downloads  enterprise-deployment.json  kerberos  Music    Pictures  Templates  WordCount.jar
cm_api.py         Documents  eclipse    express-deployment.json     lib       parcels  Public    Videos     workspace
[cloudera@quickstart ~]$
```

5) Running Mapreduce Program on Hadoop, syntax is hadoop jar jarFileName.jar ClassName /InputFileAddress /outputdir

   i.e. **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**

   here I am using the same input file ‘abc’ which I have created earlier for WordCount example (Without Combiner). **For every execution of this program we need to delete the output directory or give a new name to the output directory every time.** So here I am giving the new name to the output directory as **‘op1’**.





- As we can see from above image the the combiner input and output records coming out as,

  Combine input records=177

  Combine output records=84

  Earlier it was coming out as “zero” while executing WordCount (without combiner).

Combine input records=0

Combine output records=0

- And also here we are getting the Reduce Shuffle bytes as,

  Reduce shuffle bytes=942

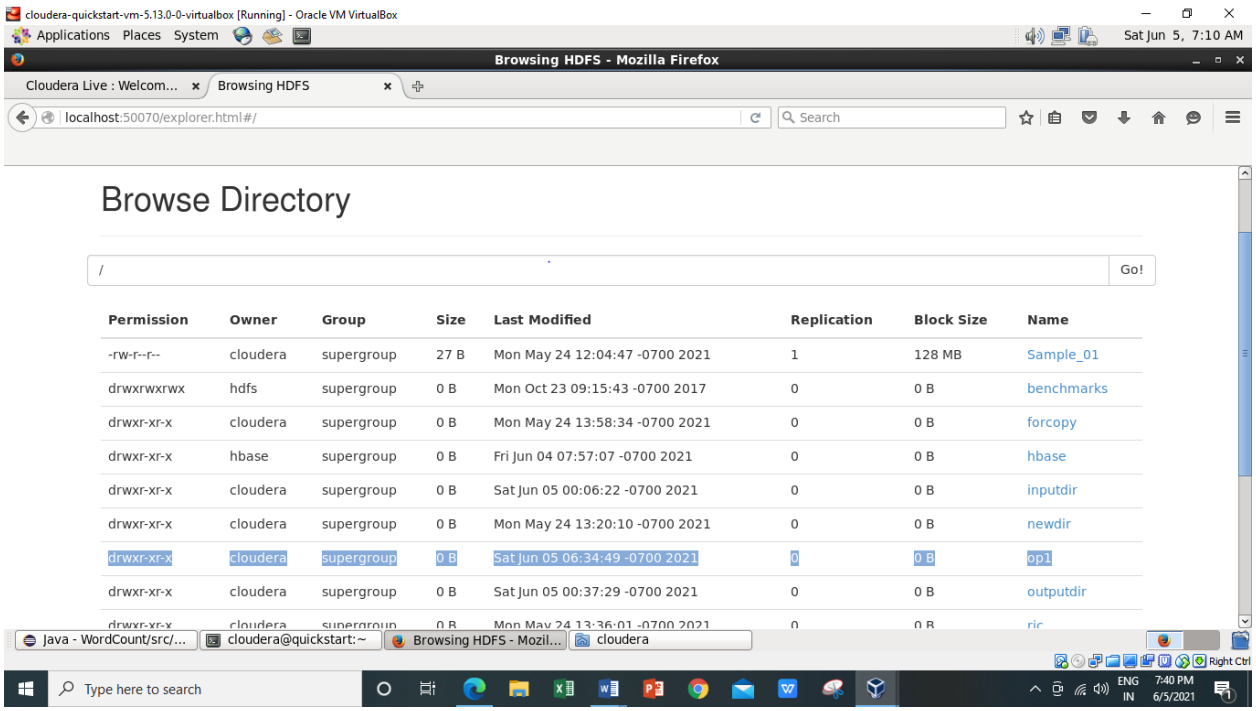
  Earlier while executing WordCount (without combiner) it is coming out as,

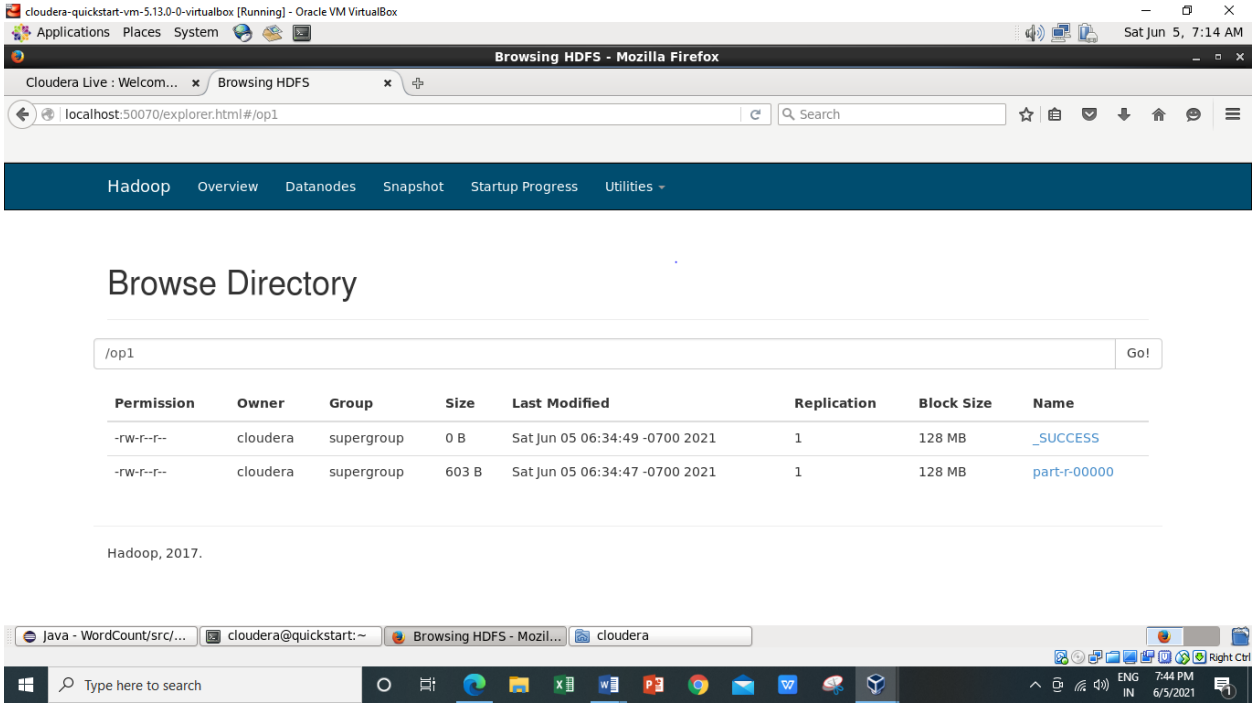
  Reduce shuffle bytes=1876

- **So Combiner is used to save the Network Bandwidth. So for saving the Network bandwidth we make use of combiner. So instead of sending every word over the network what we do is we incorporate the logic of the reducer at the combiner side so that the less amount of information can be transmitted over the network.**
- **So when we are not using combiner 1876 bytes acting as an input for the reducer. And when we are making use of the combiner so 942 bytes acting as input for the reducer.**

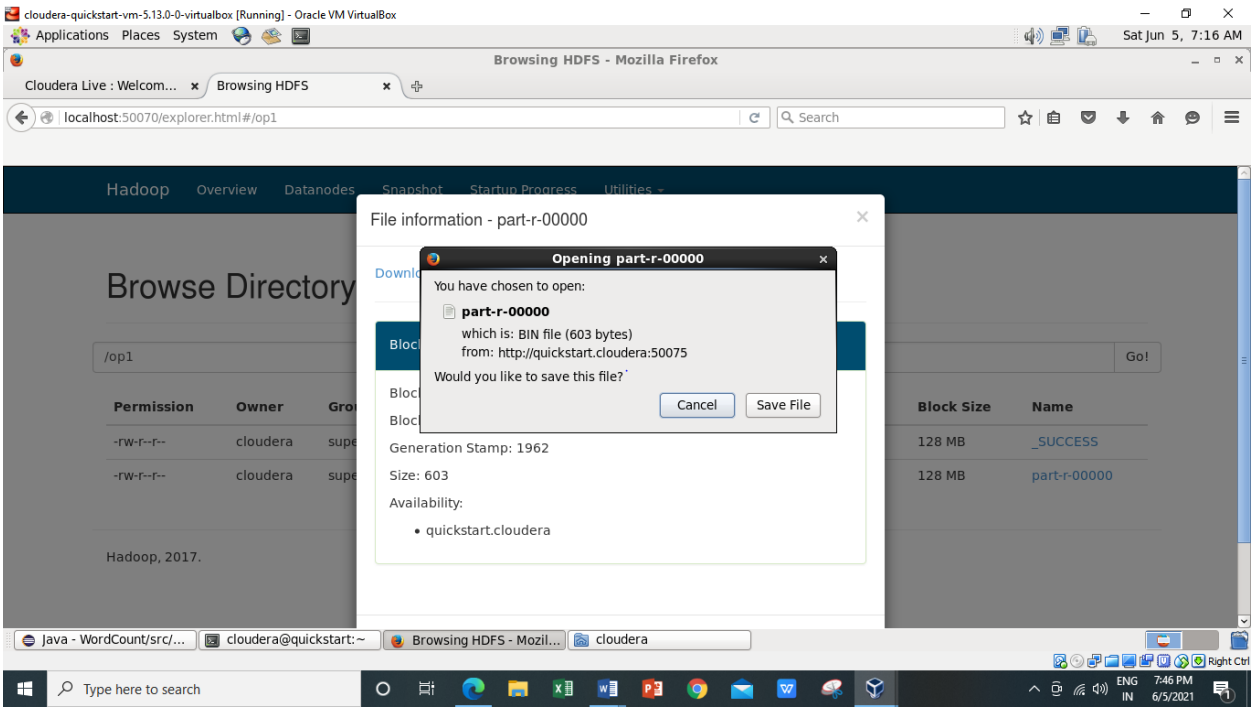
6) The same file can also be accessed using a browser.

   Browse the Directory by
   Hadoop->HDFS Namenode->Ultilities ->Browse the file system

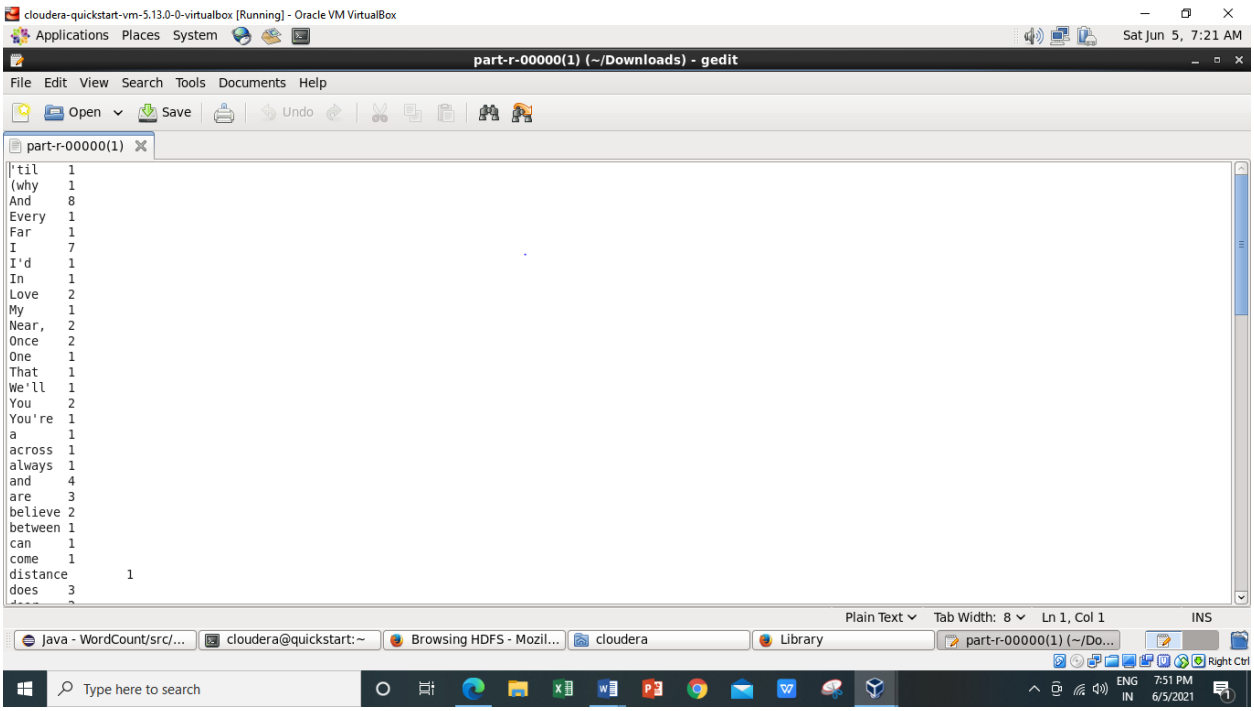


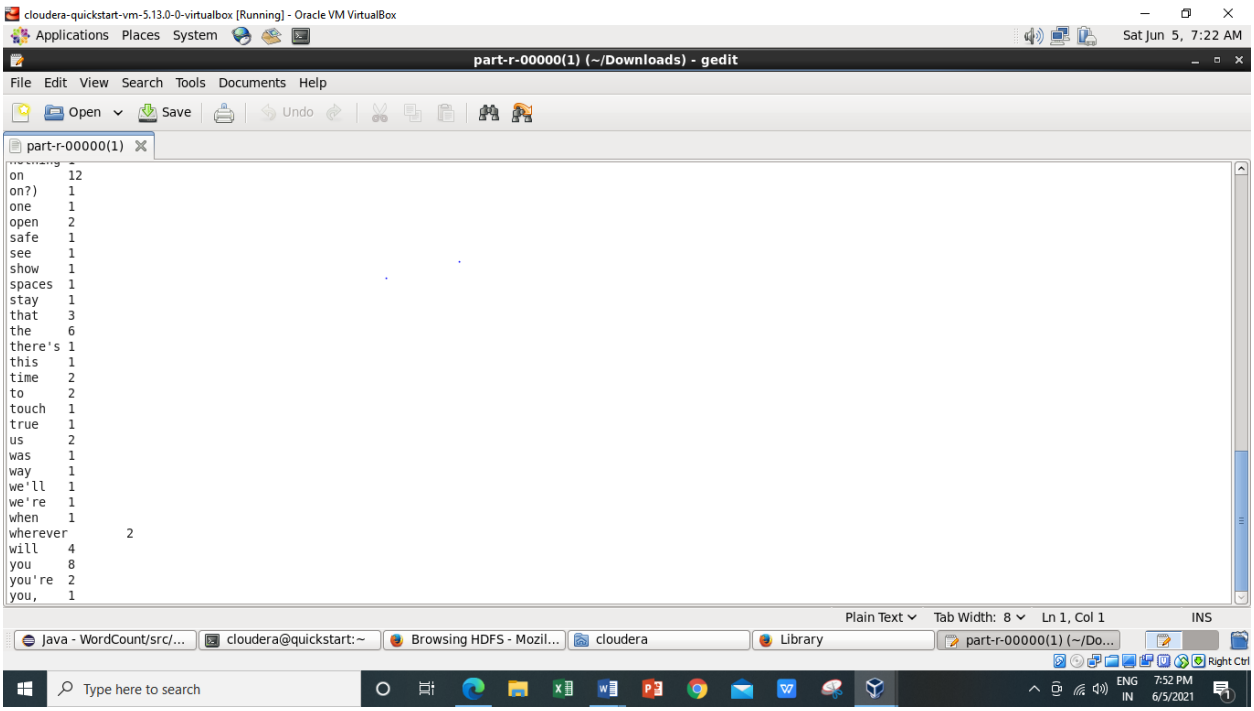


Now downloading the **part-r-00000** file**.**



Inside the **part-r-00000** file it will have the same output as we are getting after executing using command **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**




cloudera-quickstart-vm-5.13.0-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Sat Jun 5, 7:22 AM
part-r-00000(1) (~/Downloads) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
part-r-00000(1)
on 12
on?) 1
one 1
open 2
safe 1
see 1
show 1
spaces 1
stay 1
that 3
the 6
there's 1
this 1
time 2
to 2
touch 1
true 1
us 2
was 1
way 1
we'll 1
we're 1
when 1
wherever 2
will 4
you 8
you're 2
you, 1
Plain Text Tab Width: 8 Ln 1, Col 1 INS
Java - WordCount/src/... cloudera@quickstart:~ Browsing HDFS - Mozil... cloudera Library part-r-00000(1) (~/Do...
Right Ctrl
Type here to search
ENG IN 7:52 PM 6/5/2021