

PRACTICAL NO 3

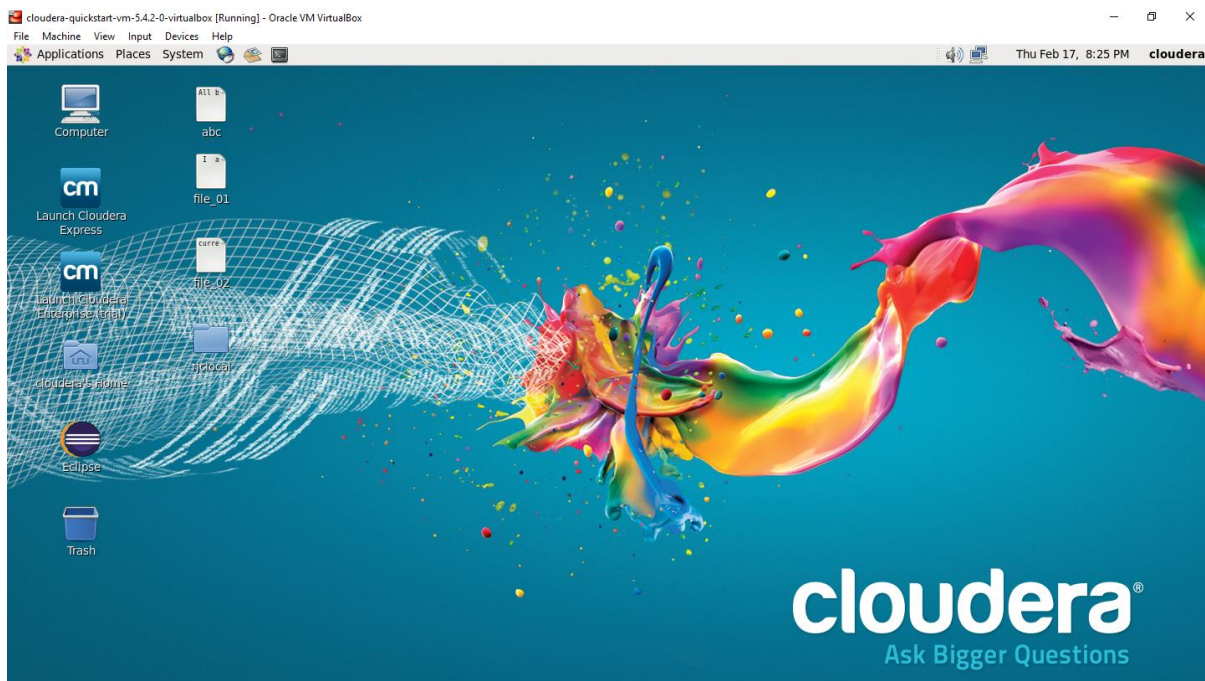
To Implement Wordcount problem using Hadoop

MapReduce in Eclipse: (With Combiner & Without Combiner)

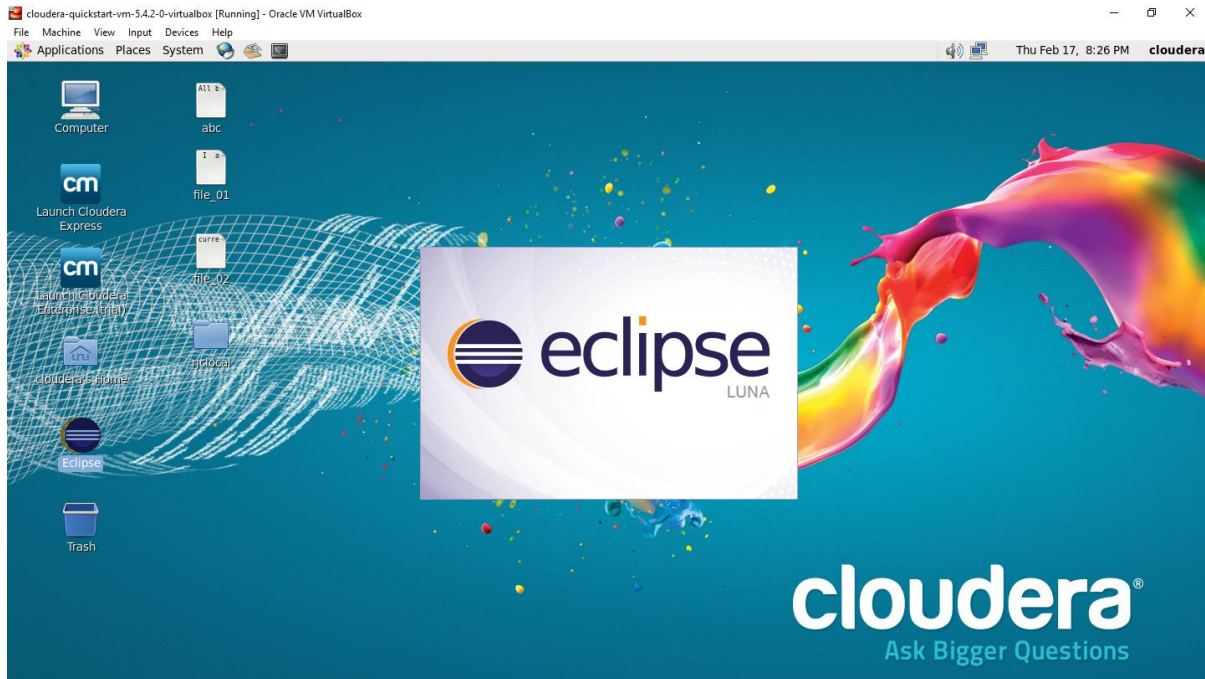
- **Steps for Word Count in Cloudera**

- **With Combiner**

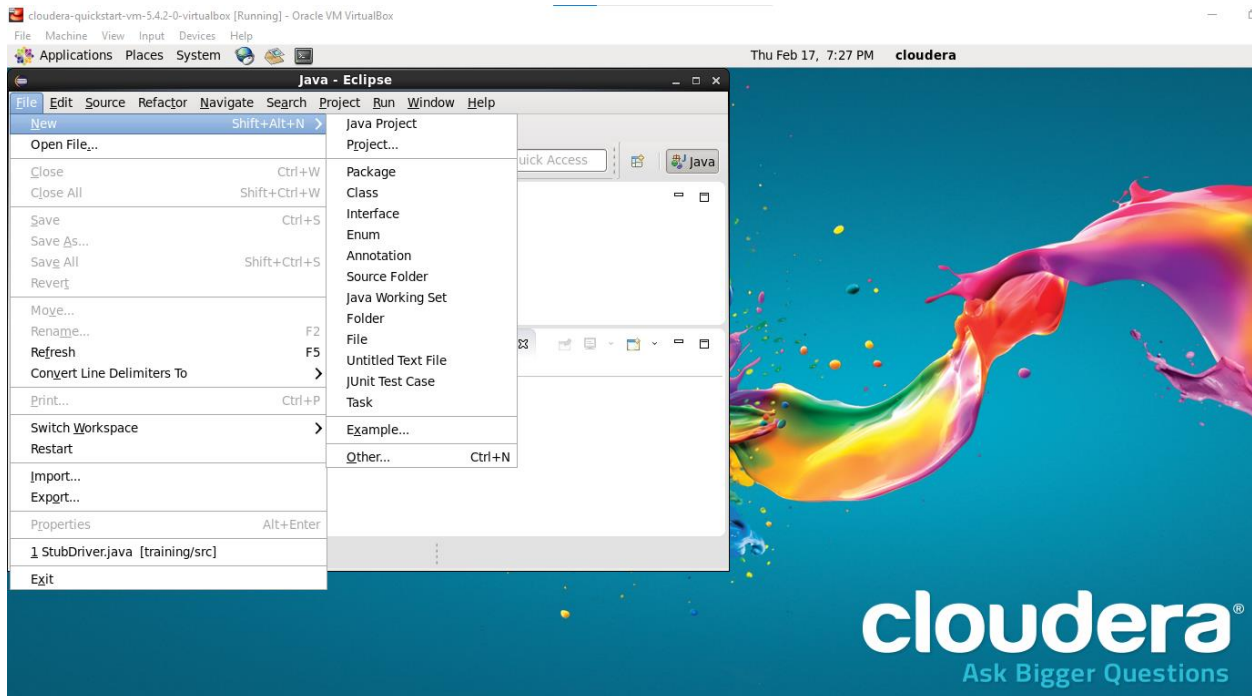
1) Open virtual box and then start cloudera quickstart

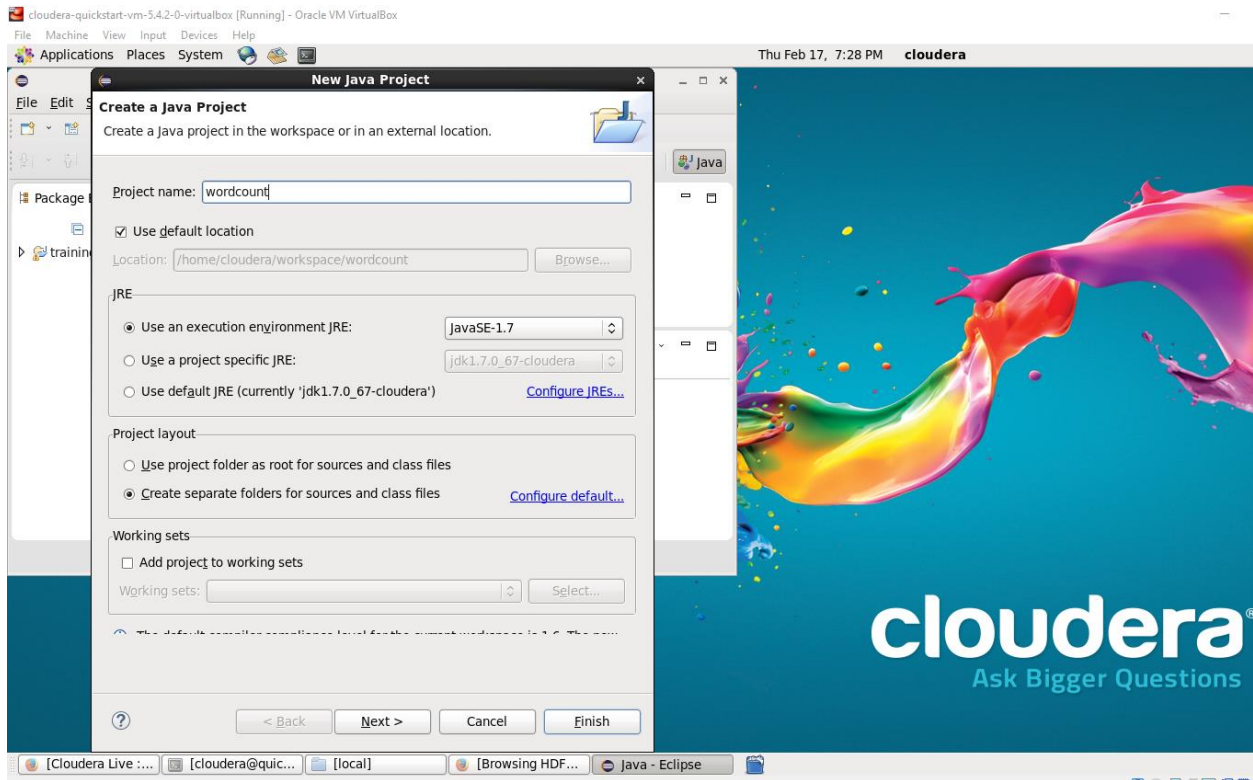


2) Open Eclipse present on the cloudera desktop



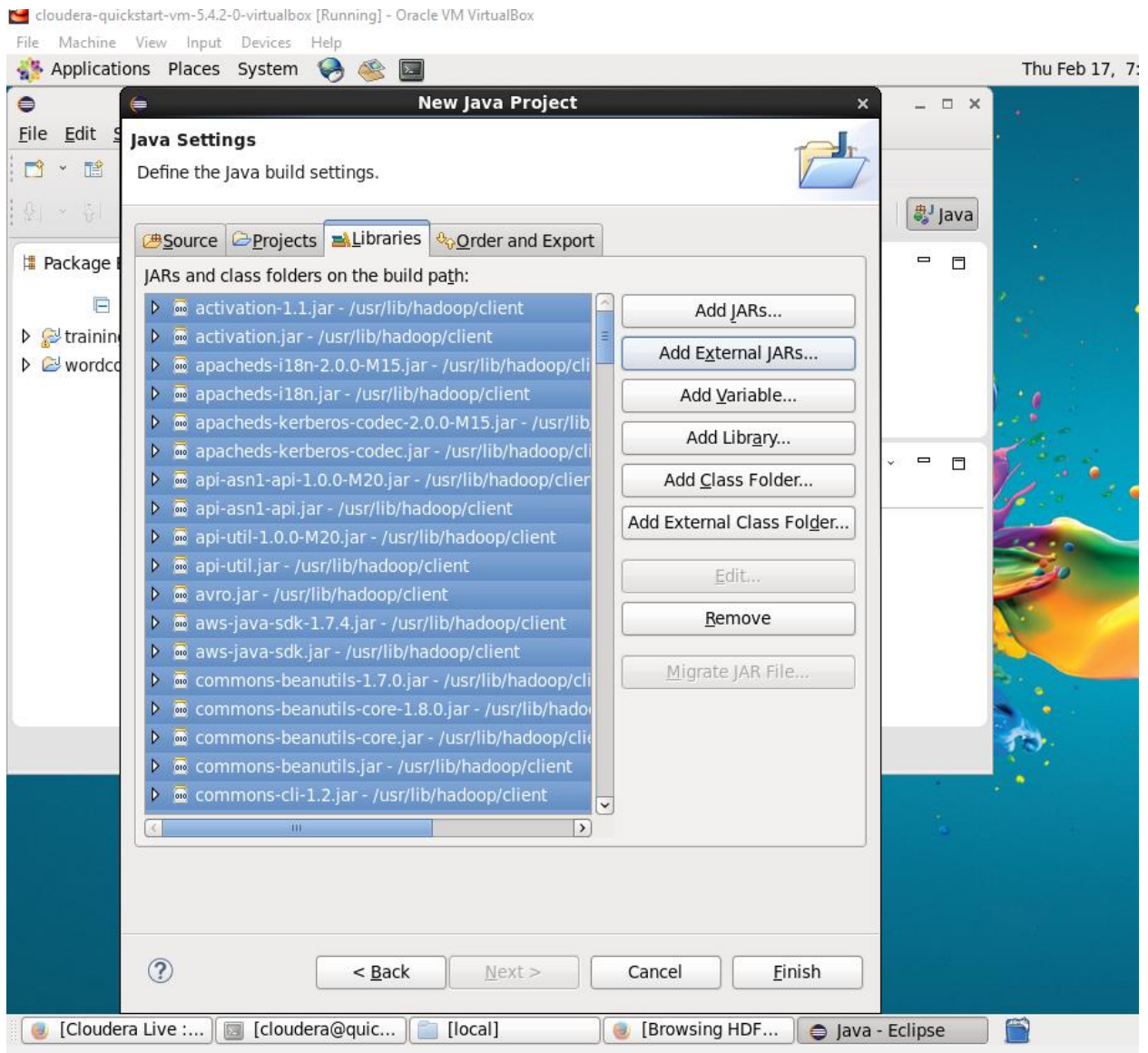
- 3) Create a new Java project clicking: File -> New -> Project -> Java Project -> Next ("WordCount" is the project name).



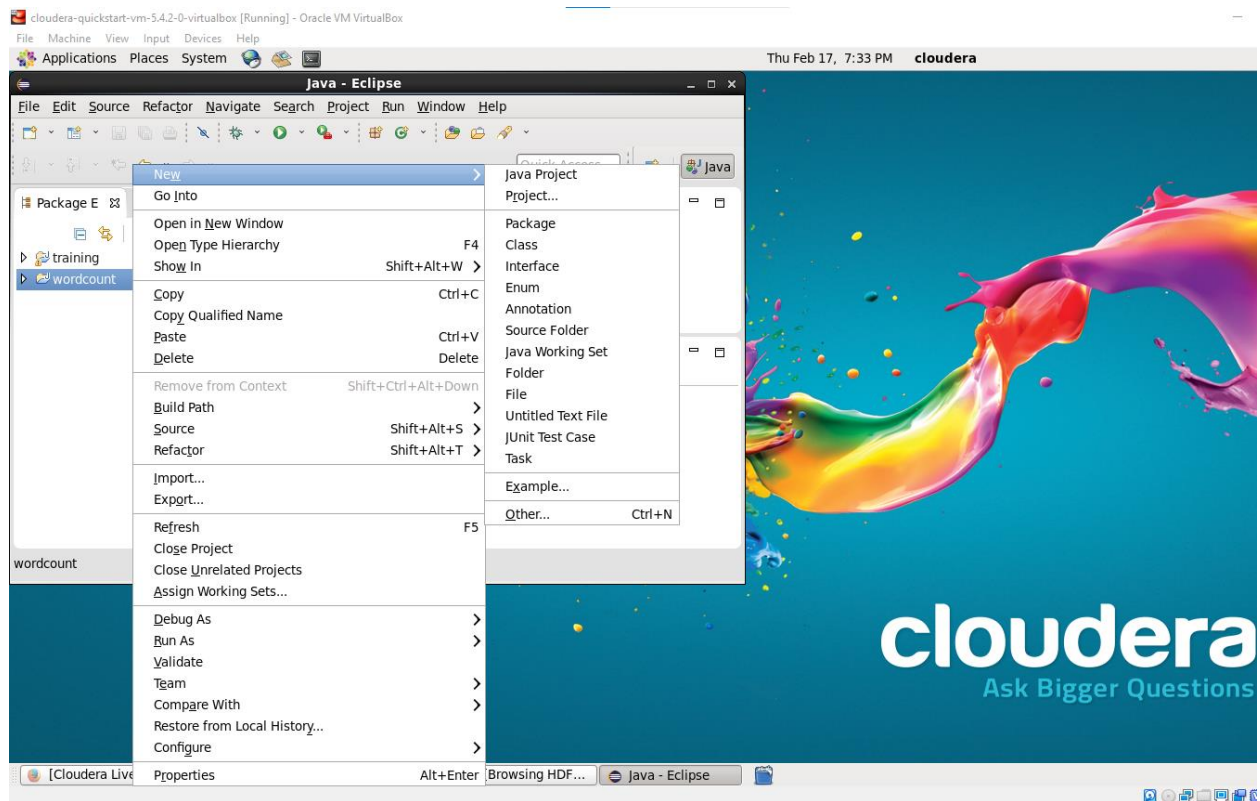


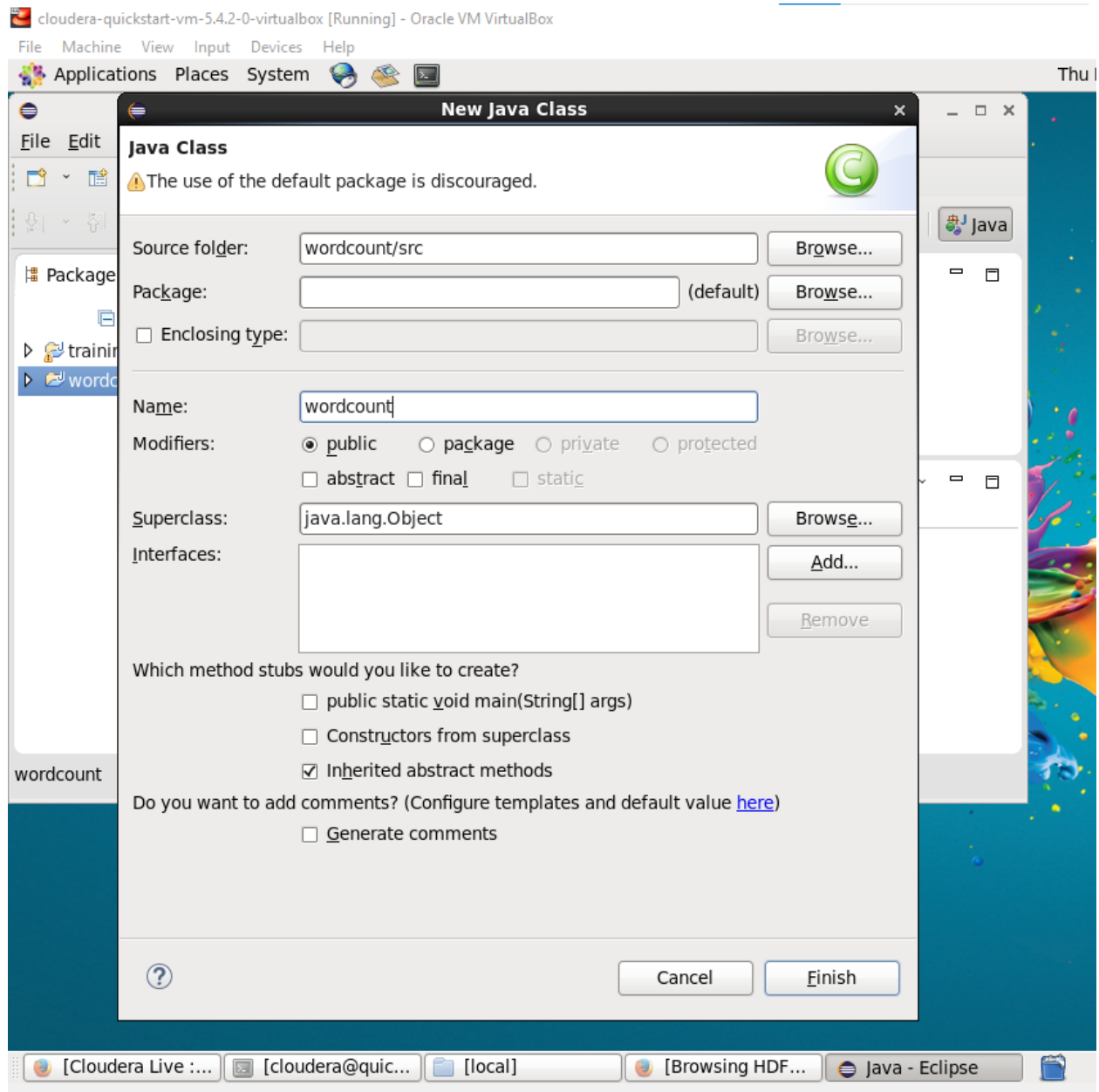
4) Adding the Hadoop libraries to the project Click on Libraries -> Add External JARs
Click on

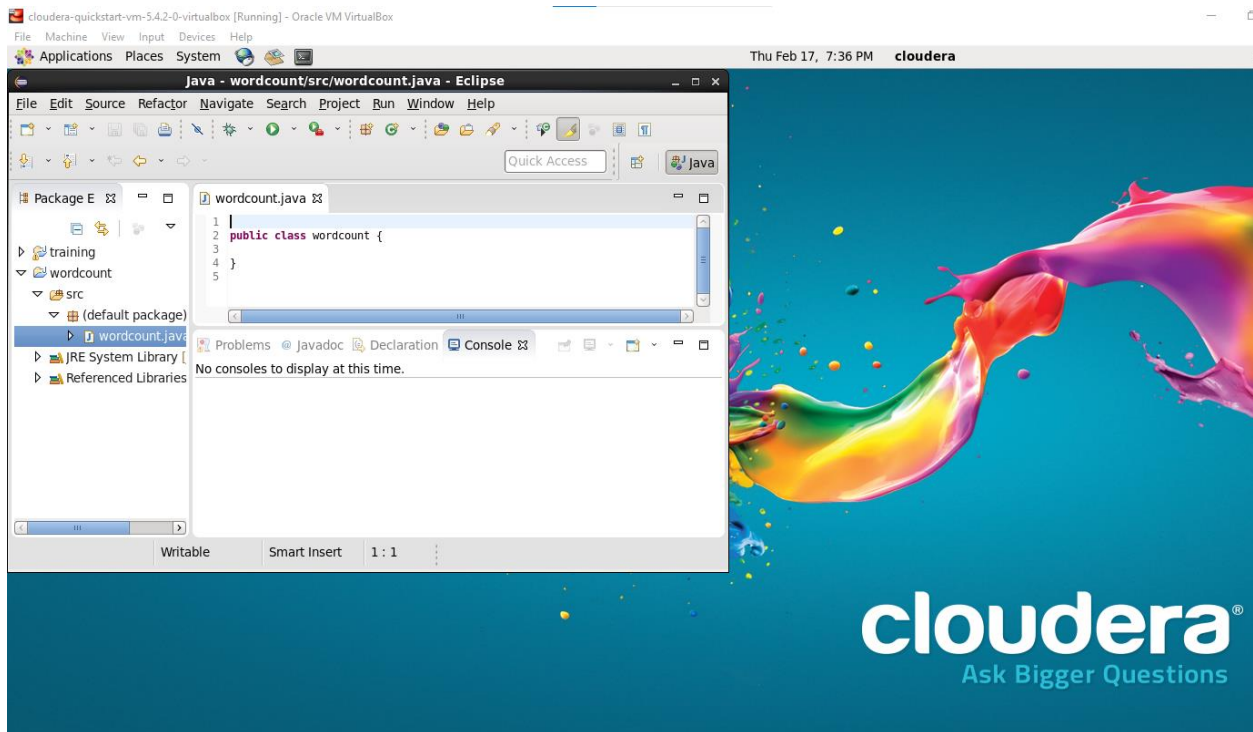
File System -> usr -> lib -> hadoop Select all the libraries (JAR Files) -> click OK Click on
Add External jars, -> client -> select all jar files -> ok -> Finish



- 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

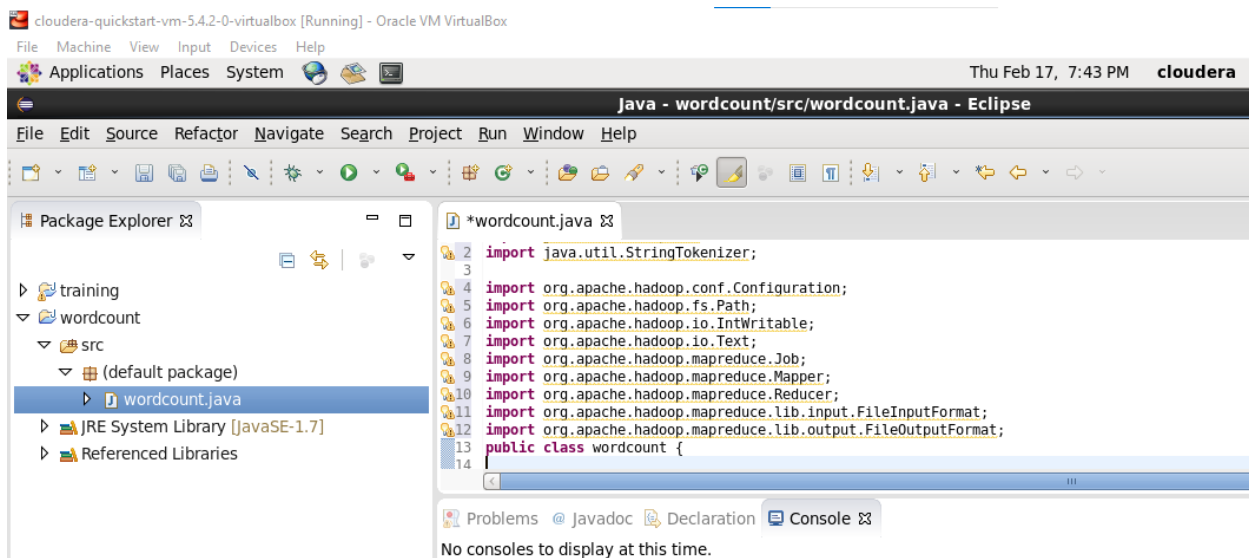




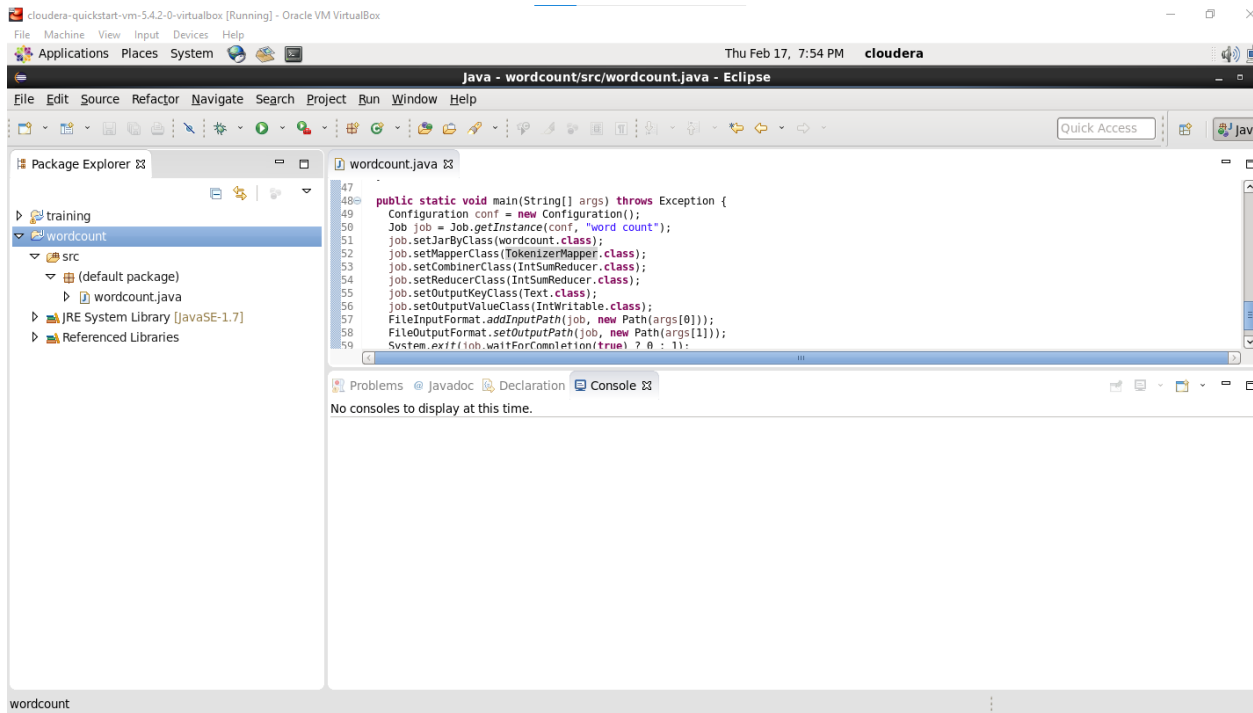


Source code:

Packages

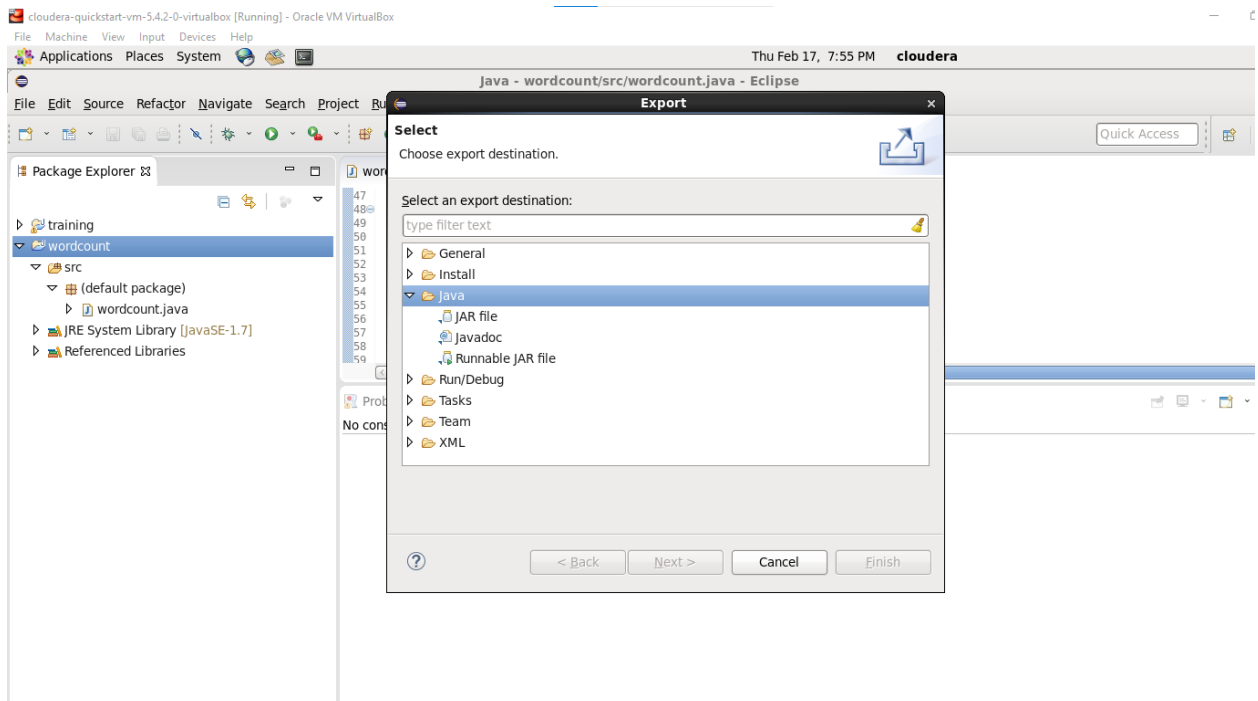
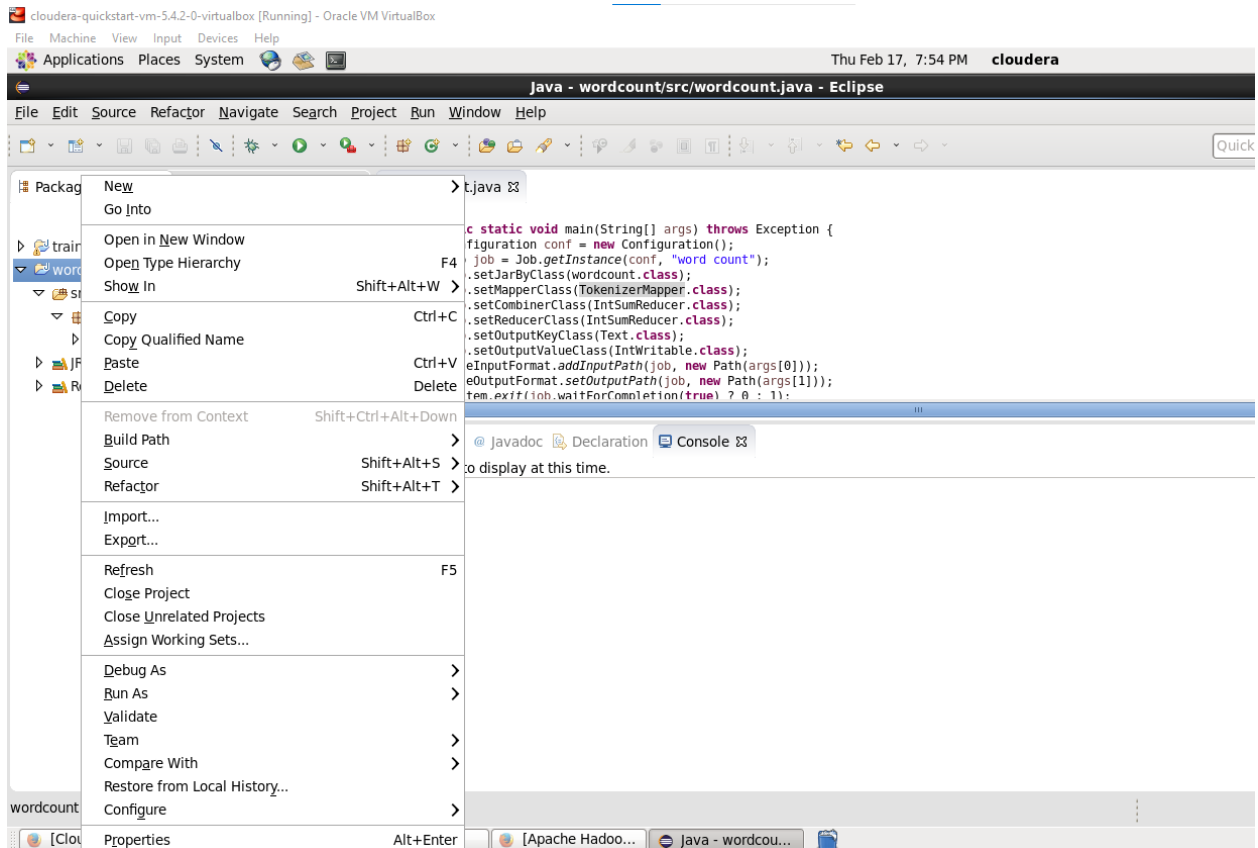


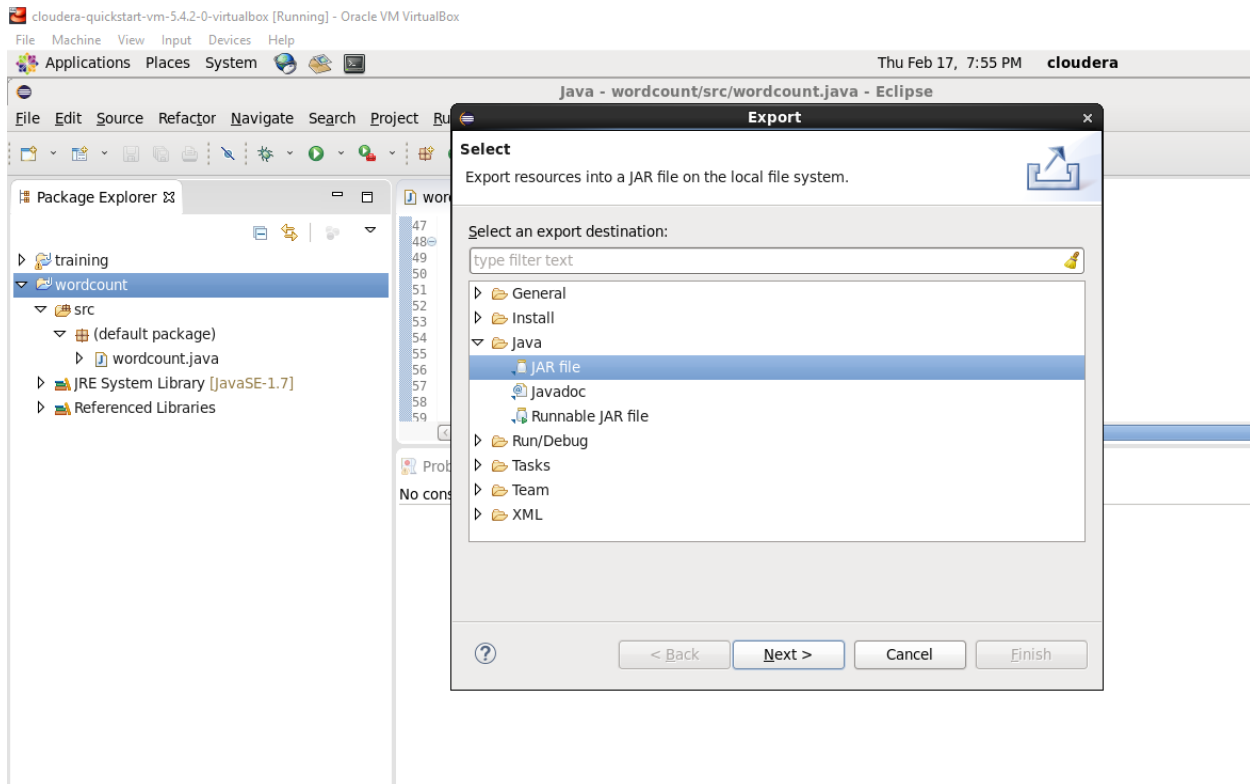
Mapper Logic



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





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

Check current working directory

->pwd

->ls

8) We need to create an input file in local file system

Creating an input file named as “abc”.

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

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

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

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

```

cloudera@quickstart:~$ ls
akpo          Documents      kerberos      Videos
AP            Downloads      lib           WordCount.jar
cloudera-manager eclipse        Music         workspace
cm_api.py     enterprise-deployment.json Pictures
demo.txt      express-deployment.json Public
Desktop       file1          Templates

[cloudera@quickstart ~]$ pwd
/home/cloudera
[cloudera@quickstart ~]$ hdfs dfs -ls/
-ls/: Unknown command
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 9 items
drwxr-xr-x  - hbase      supergroup      0 2022-02-14 20:12 /hbase
-rw-r--r--  3 cloudera  supergroup      30 2022-02-14 19:32 /output
drwxr-xr-x  - cloudera  supergroup      0 2022-02-14 20:55 /rjc2122
drwxr-xr-x  - cloudera  supergroup      0 2022-02-14 20:25 /rjcNew
drwxr-xr-x  - cloudera  supergroup      0 2022-02-14 20:39 /rjcnew1
drwxr-xr-x  - solr      solr            0 2015-06-09 03:38 /solr
drwxrwxrwx  - hdfs      supergroup      0 2022-02-07 21:44 /tmp
drwxr-xr-x  - hdfs      supergroup      0 2015-06-09 03:38 /user
drwxr-xr-x  - hdfs      supergroup      0 2015-06-09 03:36 /var
[cloudera@quickstart ~]$ hdfs dfs -mkdir /inputdir
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 10 items
drwxr-xr-x  - hbase      supergroup      0 2022-02-14 20:12 /hbase
drwxr-xr-x  - cloudera  supergroup      0 2022-02-17 20:04 /inputdir
-rw-r--r--  3 cloudera  supergroup      30 2022-02-14 19:32 /output
drwxr-xr-x  - cloudera  supergroup      0 2022-02-14 20:55 /rjc2122
drwxr-xr-x  - cloudera  supergroup      0 2022-02-14 20:25 /rjcNew
drwxr-xr-x  - cloudera  supergroup      0 2022-02-14 20:39 /rjcnew1
drwxr-xr-x  - solr      solr            0 2015-06-09 03:38 /solr
drwxrwxrwx  - hdfs      supergroup      0 2022-02-07 21:44 /tmp
drwxr-xr-x  - hdfs      supergroup      0 2015-06-09 03:38 /user
drwxr-xr-x  - hdfs      supergroup      0 2015-06-09 03:36 /var
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Desktop/abc /inputdir
[cloudera@quickstart ~]$ hdfs dfs -ls /inputdir
Found 1 items
-rw-r--r--  1 cloudera  supergroup      805 2022-02-17 20:07 /inputdir/abc
[cloudera@quickstart ~]$ hdfs dfs -cat /inputdir/abc

```

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
All but the simplest human behaviour is ascribed to intelligence, while even the
most complicated insect behaviour is never taken as an indication of intelligen
ce. What is the difference? Consider the behaviour of the digger wasp, Sphex ich
neumonous. When the female wasp returns to her burrow with food, she first depos
its it on the threshold, checks for intruders inside her burrow, and only then,
if the coast is clear, carries her food inside. The real nature of the wasp's in
stinctual behaviour is revealed if the food is moved a few inches away from the
entrance to her burrow while she is inside: on emerging, she will repeat the who
le procedure as often as the food is displaced. Intelligence-conspicuously absen
t in the case of Sphex-must include the ability to adapt to new circumstances.

```

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`

```
[cloudera@quickstart ~]$ hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /outputdir
22/02/17 20:09:44 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
22/02/17 20:09:45 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
22/02/17 20:09:45 INFO input.FileInputFormat: Total input paths to process : 1
22/02/17 20:09:45 INFO mapreduce.JobSubmitter: number of splits:1
22/02/17 20:09:46 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1644894610889_0001
22/02/17 20:09:47 INFO impl.YarnClientImpl: Submitted application application_1644894610889_0001
22/02/17 20:09:47 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1644894610889_0001/
22/02/17 20:09:47 INFO mapreduce.Job: Running job: job_1644894610889_0001
22/02/17 20:10:02 INFO mapreduce.Job: Job job_1644894610889_0001 running in uber mode : false
22/02/17 20:10:02 INFO mapreduce.Job: map 0% reduce 0%
22/02/17 20:10:17 INFO mapreduce.Job: map 100% reduce 0%
22/02/17 20:10:28 INFO mapreduce.Job: map 100% reduce 100%
22/02/17 20:10:28 INFO mapreduce.Job: Job job_1644894610889_0001 completed successfully
22/02/17 20:10:28 INFO mapreduce.Job: Counters: 49
    File System Counters
      FILE: Number of bytes read=1156
      FILE: Number of bytes written=222899
      FILE: Number of read operations=0
      FILE: Number of large read operations=0
```

Map-Reduce Framework

```
cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
22/02/17 20:10:28 INFO mapreduce.Job: Job job_1644894610889_0001 completed successfully
22/02/17 20:10:28 INFO mapreduce.Job: Counters: 49
    File System Counters
      FILE: Number of bytes read=1156
      FILE: Number of bytes written=222899
      FILE: Number of read operations=0
      FILE: Number of large read operations=0
      FILE: Number of write operations=0
      HDFS: Number of bytes read=914
      HDFS: Number of bytes written=799
      HDFS: Number of read operations=6
      HDFS: Number of large read operations=0
      HDFS: Number of write operations=2
    Job Counters
      Launched map tasks=1
      Launched reduce tasks=1
      Data-local map tasks=1
      Total time spent by all maps in occupied slots (ms)=11842
      Total time spent by all reduces in occupied slots (ms)=8821
      Total time spent by all map tasks (ms)=11842
      Total time spent by all reduce tasks (ms)=8821
      Total vcore-seconds taken by all map tasks=11842
      Total vcore-seconds taken by all reduce tasks=8821
      Total megabyte-seconds taken by all map tasks=12126208
      Total megabyte-seconds taken by all reduce tasks=9032704
    Map-Reduce Framework
      Map input records=1
      Map output records=132
      Map output bytes=1333
      Map output materialized bytes=1156
      Input split bytes=109
      Combine input records=132
      Combine output records=88
      Reduce input groups=88
      Reduce shuffle bytes=1156
      Reduce input records=88
      Reduce output records=88
      Spilled Records=176
      Shuffled Maps =1
```

As we can see in the above output,

Combine input records=132

Combine output records=88

And Reduce shuffle bytes coming as,

Reduce shuffle bytes=1876

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 /



```

cloudera@quickstart:~$ hdfs dfs -ls /
Found 11 items
drwxr-xr-x - hbase supergroup 0 2022-02-14 20:12 /hbase
drwxr-xr-x - cloudera supergroup 0 2022-02-17 20:07 /inputdir
drwxr-xr-x - cloudera supergroup 0 2022-02-17 20:10 /outputdir
-rw-r--r-- 3 cloudera supergroup 30 2022-02-14 19:32 /output
drwxr-xr-x - cloudera supergroup 0 2022-02-14 20:55 /rjc2122
drwxr-xr-x - cloudera supergroup 0 2022-02-14 20:25 /rjcNew
drwxr-xr-x - cloudera supergroup 0 2022-02-14 20:39 /rjcnew1
drwxr-xr-x - solr solr 0 2015-06-09 03:38 /solr
drwxrwxrwx - hdfs supergroup 0 2022-02-07 21:44 /tmp
drwxr-xr-x - hdfs supergroup 0 2015-06-09 03:38 /user
drwxr-xr-x - hdfs supergroup 0 2015-06-09 03:36 /var
cloudera@quickstart:~$ hdfs dfs -ls /outputdir
ls: '/outputdir': No such file or directory
cloudera@quickstart:~$ hdfs dfs -ls /outputdir
Found 2 items
-rw-r--r-- 1 cloudera supergroup 0 2022-02-17 20:10 /outputdir/_SUCCESS
-rw-r--r-- 1 cloudera supergroup 799 2022-02-17 20:10 /outputdir/part-r-00000
cloudera@quickstart:~$ hdfs dfs -cat /outputdir/part-r-00000

```

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

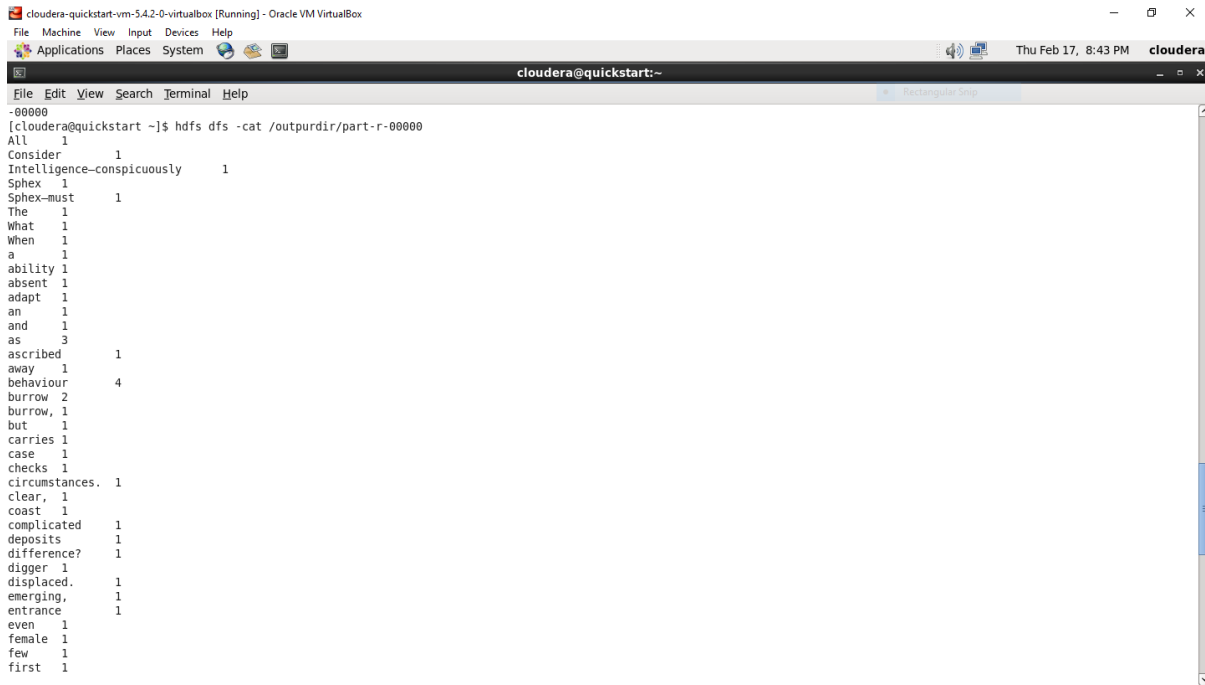
/outputdir

```

cloudera@quickstart:~$ hdfs dfs -ls /outputdir
Found 2 items
-rw-r--r-- 1 cloudera supergroup 0 2022-02-17 20:10 /outputdir/_SUCCESS
-rw-r--r-- 1 cloudera supergroup 799 2022-02-17 20:10 /outputdir/part-r-00000
cloudera@quickstart:~$ hdfs dfs -cat /outputdir/part-r-00000

```

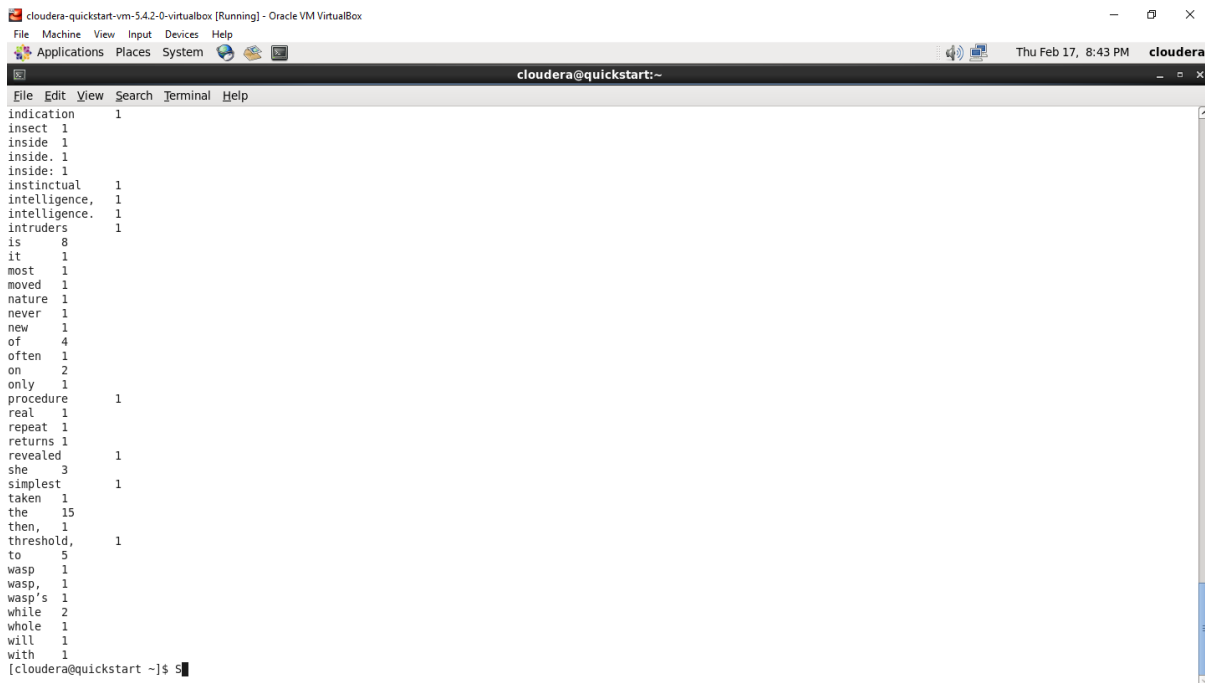
Now we want to read the content of the **part-r-00000** file which present inside the **outputdir** using command **hdfs dfs -cat /outputdir/part-r-00000**



```

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
-00000
[cloudera@quickstart ~]$ hdfs dfs -cat /outpurdir/part-r-00000
All 1
Consider 1
Intelligence-conspicuously 1
Sphex 1
Sphex-must 1
The 1
What 1
When 1
a 1
ability 1
absent 1
adapt 1
an 1
and 1
as 3
ascribed 1
away 1
behaviour 4
burrow 2
burrow, 1
but 1
carries 1
case 1
checks 1
circumstances. 1
clear, 1
coast 1
complicated 1
deposits 1
difference? 1
digger 1
displaced. 1
emerging, 1
entrance 1
even 1
female 1
few 1
first 1

```



```

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
indication 1
insect 1
inside 1
inside, 1
inside: 1
instinctual 1
intelligence, 1
intelligence. 1
intruders 1
is 8
it 1
most 1
moved 1
nature 1
never 1
new 1
of 4
often 1
on 2
only 1
procedure 1
real 1
repeat 1
returns 1
revealed 1
she 3
simplest 1
taken 1
the 15
then, 1
threshold, 1
to 5
wasp 1
wasp, 1
wasp's 1
while 2
whole 1
will 1
with 1
[cloudera@quickstart ~]$

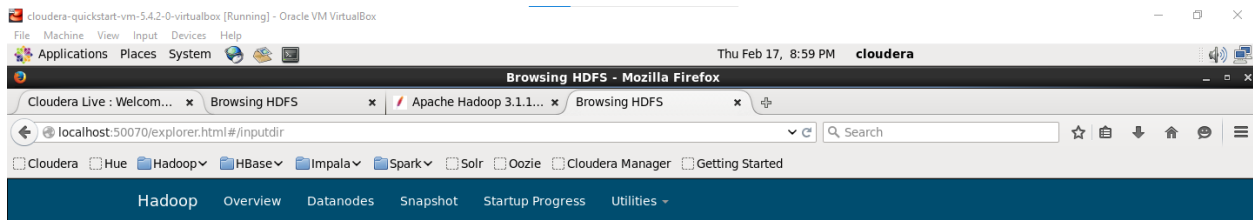
```

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->Utilities ->Browse the file system



Browse Directory

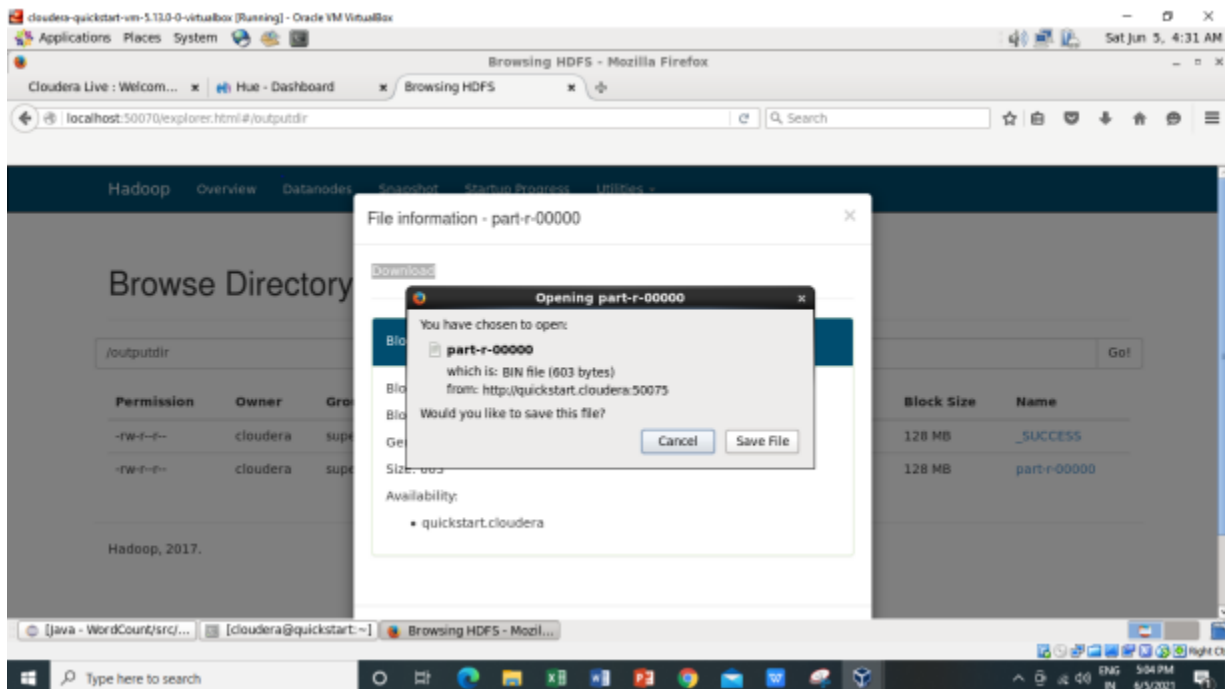
/inputdir

Go

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	cloudera	supergroup	35 B	1	128 MB	abc

Hadoop, 2014.

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

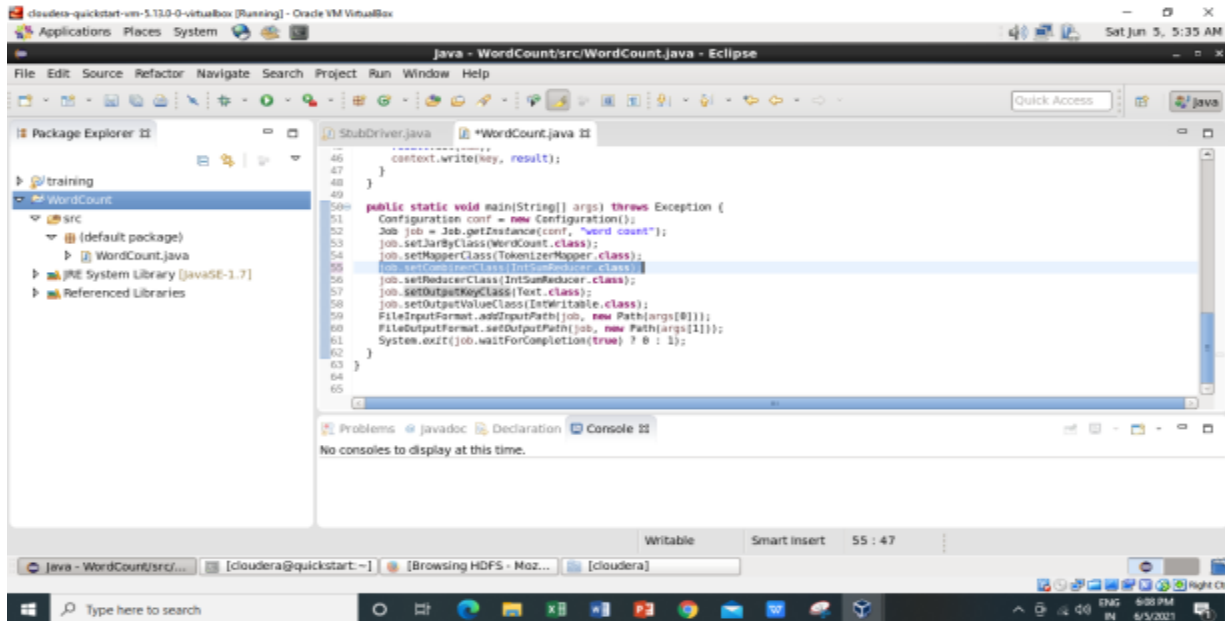


Inside the **part-r-00000** file it will have the same output as we are getting after executing using command **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**

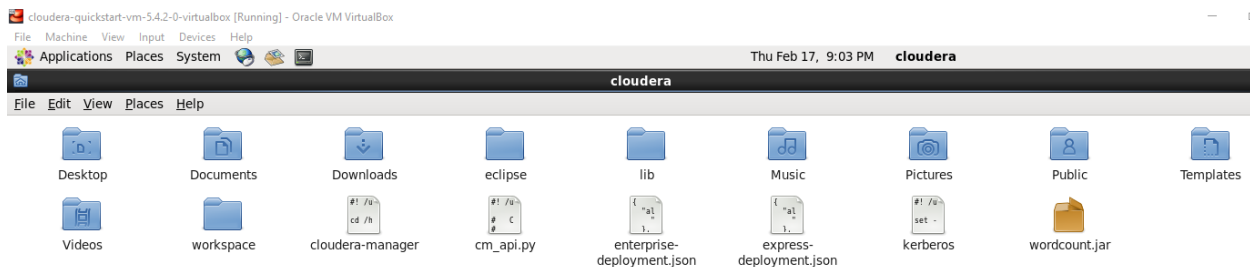
For every execution of this program we need to delete the output directory or give a new name to the output directory every time.

➤ Implementation of WordCount problem using Hadoop MapReduce (Without Combiner) in Eclipse:

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

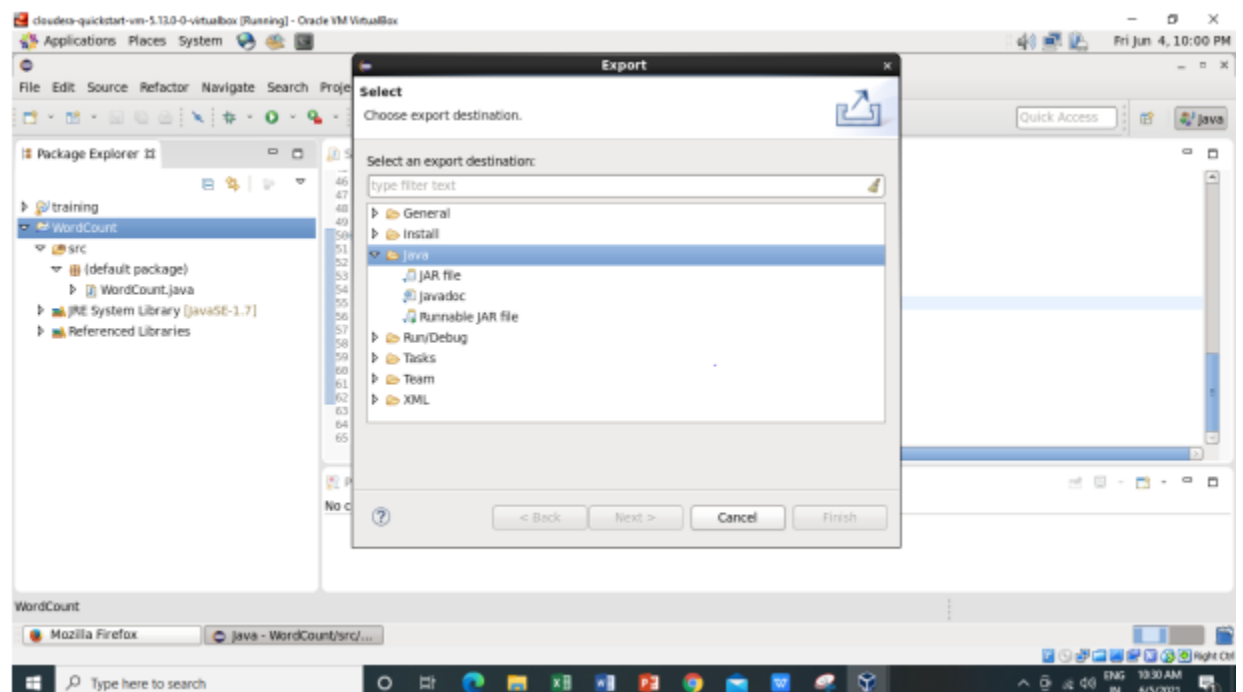
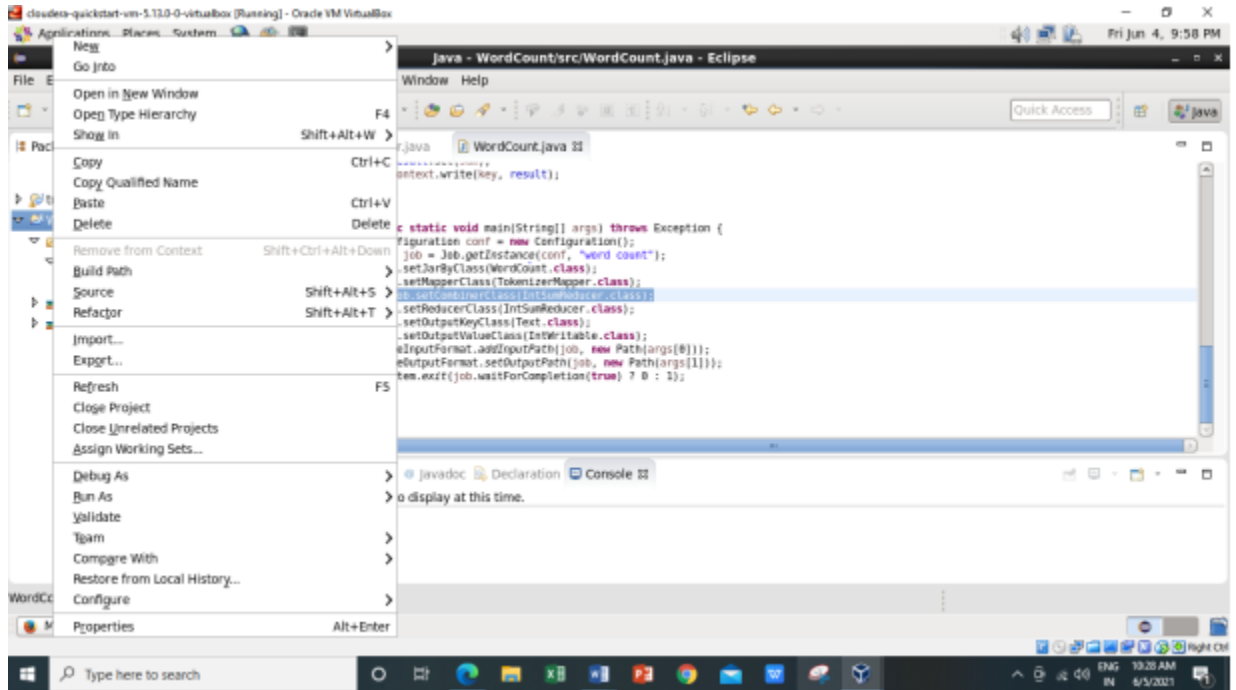


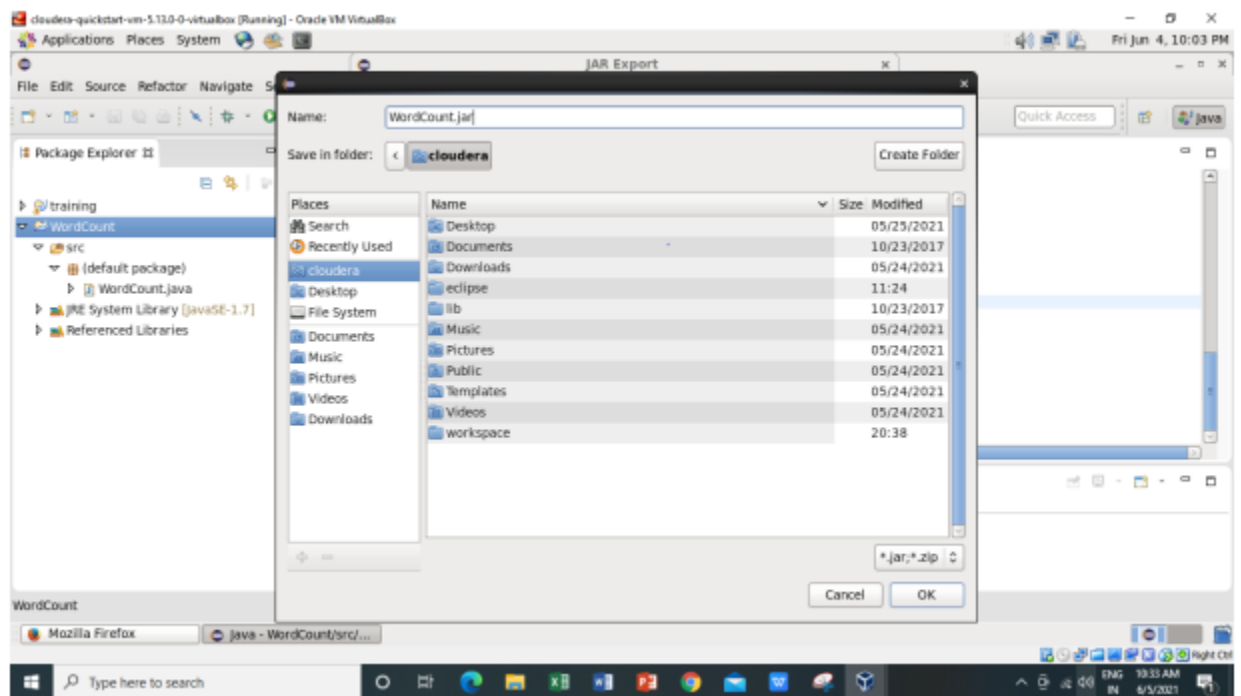
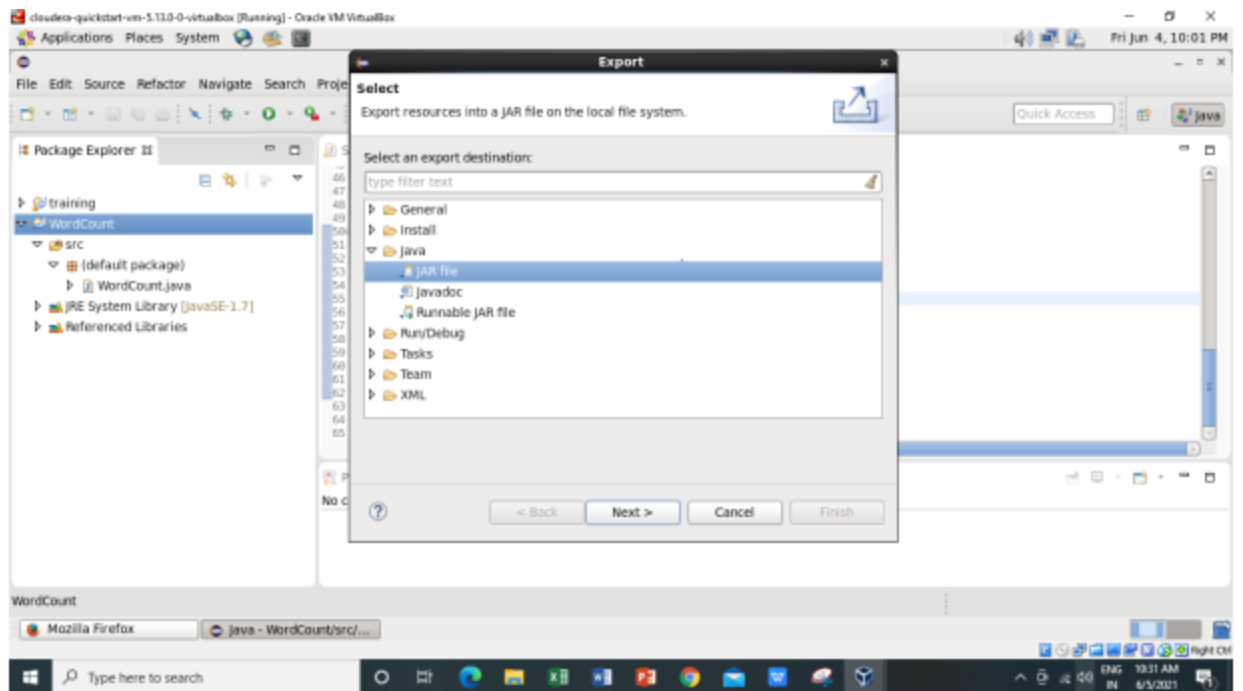
2) And will delete the WordCount.jar file in which all jar files are present from /home/cloudera.

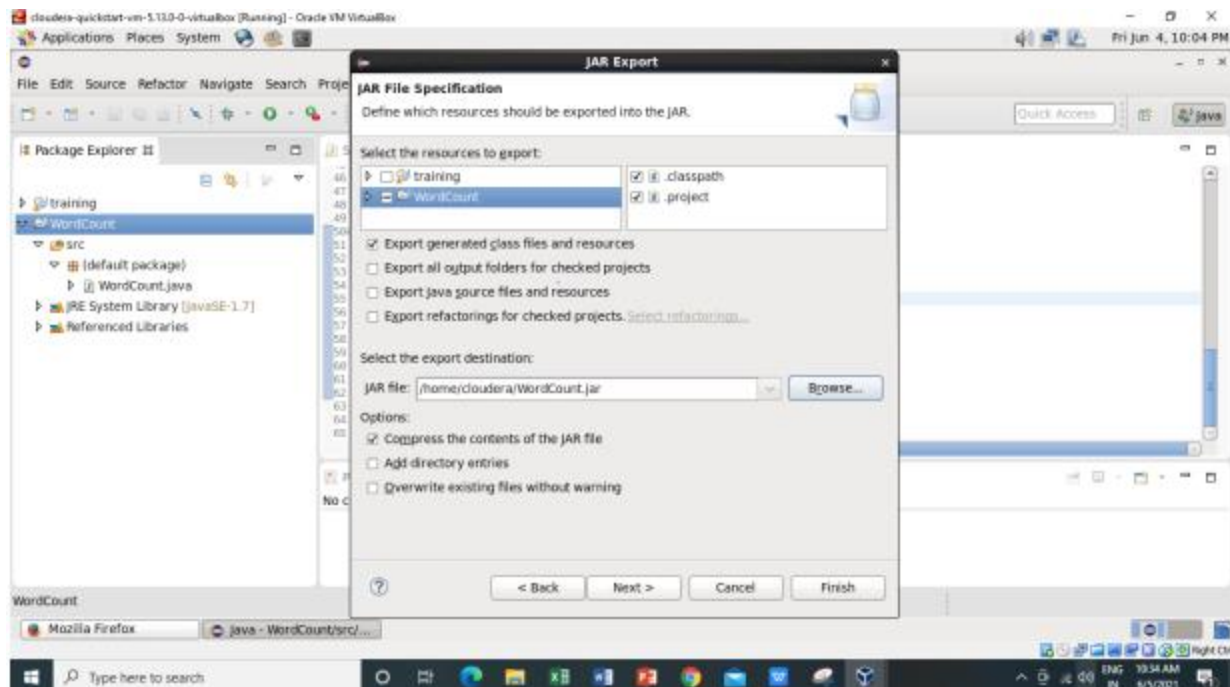


We have successfully deleted the WordCount.jar file.

- 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







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'.

```

cloudera@quickstart:~$ hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /outputdir
21/06/05 00:28:35 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/06/05 00:28:43 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
21/06/05 00:28:50 INFO input.FileInputFormat: Total input paths to process : 1
21/06/05 00:28:55 INFO mapreduce.JobSubmitter: number of splits:1
21/06/05 00:28:57 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1622818056286_0001
21/06/05 00:29:20 INFO impl.YarnClientImpl: Submitted application application_1622818056286_0001
21/06/05 00:29:28 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1622818056286_0001/
21/06/05 00:29:28 INFO mapreduce.Job: Running job: job_1622818056286_0001
21/06/05 00:34:57 INFO mapreduce.Job: Job job_1622818056286_0001 running in uber mode : false
21/06/05 00:34:57 INFO mapreduce.Job: map 0% reduce 0%
21/06/05 00:37:10 INFO mapreduce.Job: map 100% reduce 0%
21/06/05 00:37:30 INFO mapreduce.Job: map 100% reduce 100%
21/06/05 00:37:31 INFO mapreduce.Job: Job job_1622818056286_0001 completed successfully
21/06/05 00:37:32 INFO mapreduce.Job: Counters: 49
  File System Counters
    FILE: Number of bytes read=1876
    FILE: Number of bytes written=290079
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=922
    HDFS: Number of bytes written=603
    HDFS: Number of read operations=6
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=129994
    Total time spent by all reduces in occupied slots (ms)=16903
    Total time spent by all map tasks (ms)=129994

```

```

cloudera@quickstart:~$ hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /outputdir
21/06/05 00:28:35 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
21/06/05 00:28:43 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
21/06/05 00:28:50 INFO input.FileInputFormat: Total input paths to process : 1
21/06/05 00:28:55 INFO mapreduce.JobSubmitter: number of splits:1
21/06/05 00:28:57 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1622818056286_0001
21/06/05 00:29:20 INFO impl.YarnClientImpl: Submitted application application_1622818056286_0001
21/06/05 00:29:28 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1622818056286_0001/
21/06/05 00:29:28 INFO mapreduce.Job: Running job: job_1622818056286_0001
21/06/05 00:34:57 INFO mapreduce.Job: Job job_1622818056286_0001 running in uber mode : false
21/06/05 00:34:57 INFO mapreduce.Job: map 0% reduce 0%
21/06/05 00:37:10 INFO mapreduce.Job: map 100% reduce 0%
21/06/05 00:37:30 INFO mapreduce.Job: map 100% reduce 100%
21/06/05 00:37:31 INFO mapreduce.Job: Job job_1622818056286_0001 completed successfully
21/06/05 00:37:32 INFO mapreduce.Job: Counters: 49
  File System Counters
    FILE: Number of bytes read=1876
    FILE: Number of bytes written=290079
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=922
    HDFS: Number of bytes written=603
    HDFS: Number of read operations=6
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Launched map tasks=1
    Launched reduce tasks=1
    Data-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=129994
    Total time spent by all reduces in occupied slots (ms)=16903
    Total time spent by all map tasks (ms)=129994
  Map-Reduce Framework
    Map input records=8
    Map output records=177
    Map output bytes=1516
    Map output materialized bytes=1876
    Input split bytes=109
    Combine input records=0
    Combine output records=0
    Reduce input groups=84
    Reduce shuffle bytes=1876
    Reduce input records=177
    Reduce output records=84
    Spilled Records=354
    Shuffled Maps =1
    Failed Shuffles=0
    Merged Map outputs=1
    GC time elapsed (ms)=572
    CPU time spent (ms)=3570
    Physical memory (bytes) snapshot=309050112
    Virtual memory (bytes) snapshot=3015163904
    Total committed heap usage (bytes)=152905120
  Shuffle Errors
    BAD ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
  File Input Format Counters
    Bytes Read=813
  File Output Format Counters
    Bytes Written=603

```

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

Combine input records=0

Combine output records=0

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

Combine input records=132

Combine output records=88

- 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,

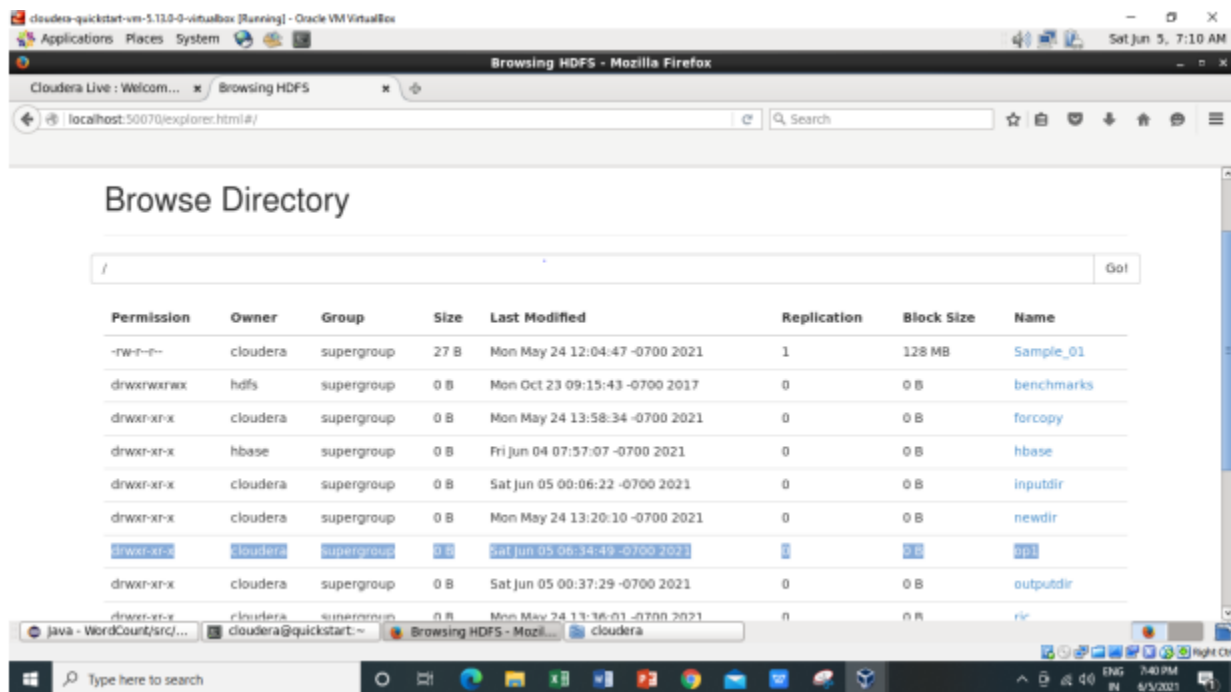
Reduce shuffle bytes=1876

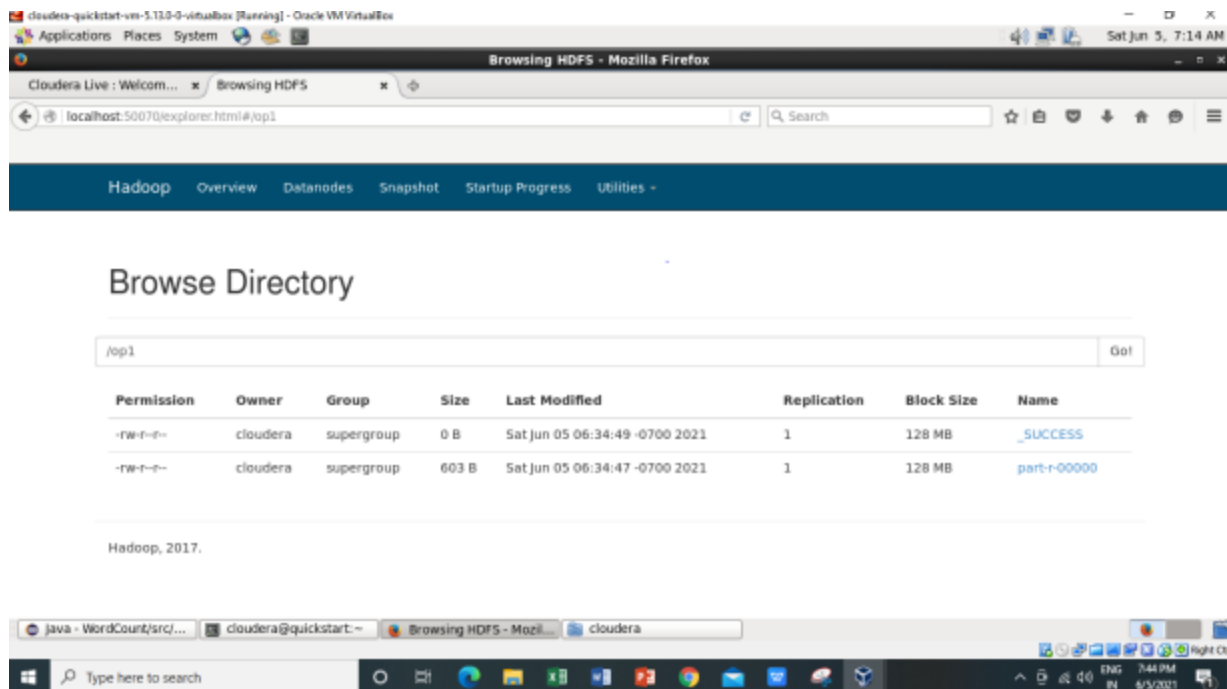
- So Combiner is used to save the Network Bandwidth. So for saving the Network bandwidth we make use of combiner. So instead of sending every word over the network what we do is we incorporate the logic of the reducer at the combiner side so that the less amount of information can be transmitted over 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.

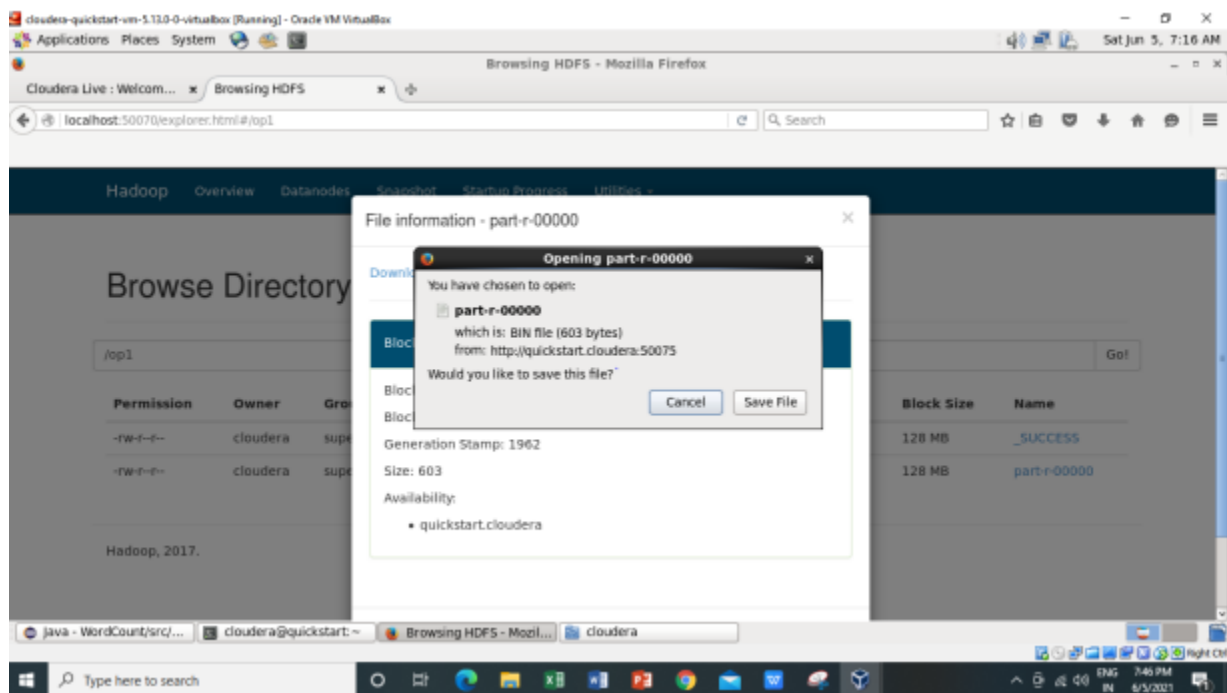
Browse the Directory by

Hadoop->HDFS Namenode->Utilities ->Browse the file system

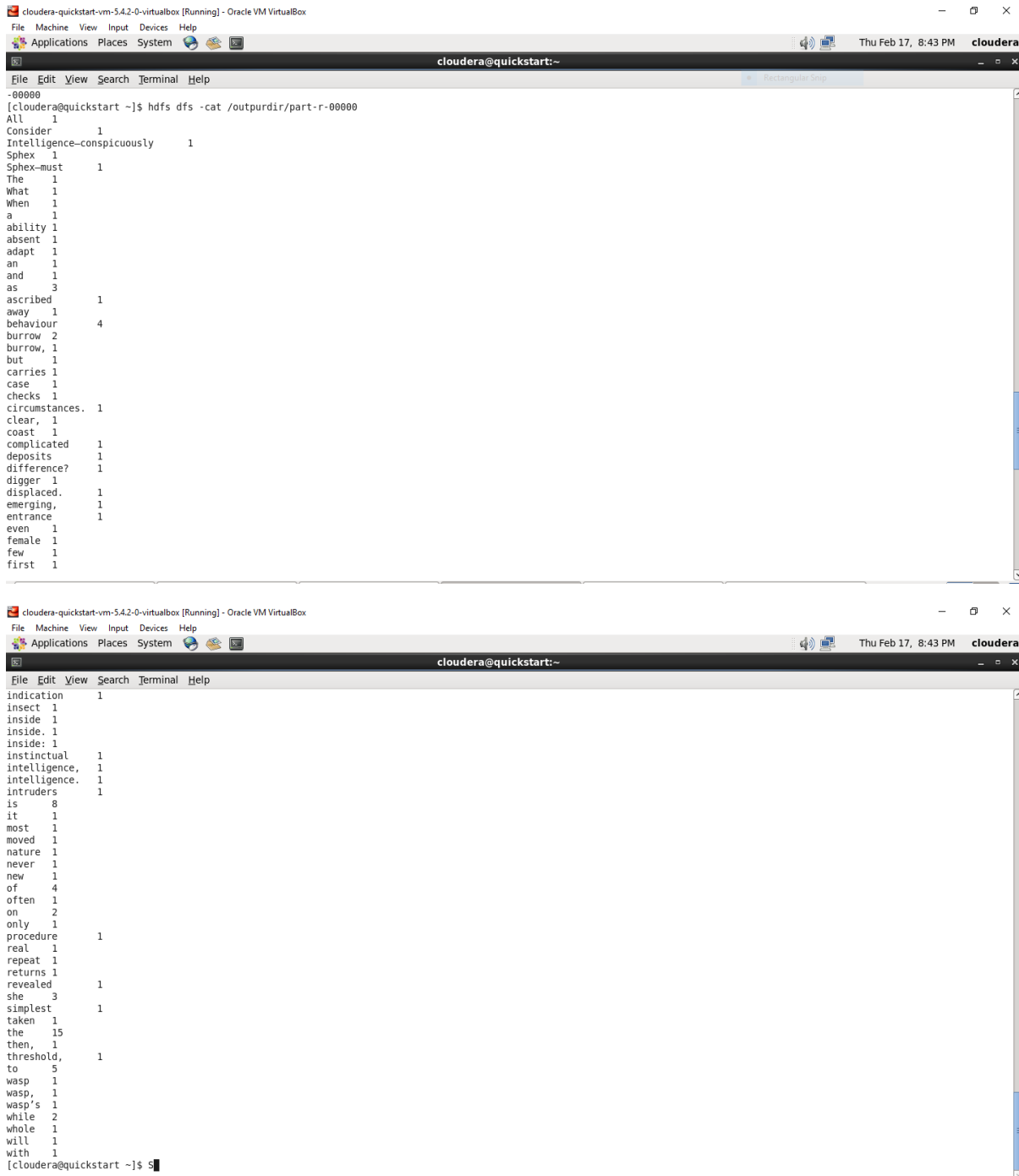




Now downloading the **part-r-00000** file.



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**



The image shows two screenshots of a Cloudera Quickstart VM terminal window. The window title is "cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox". The terminal shows the command `hdfs dfs -cat /outpurdir/part-r-00000` being executed. The output is a list of words and their frequencies, sorted by frequency in descending order.

First Screenshot Output:

```

-00000
[cloudera@quickstart ~]$ hdfs dfs -cat /outpurdir/part-r-00000
All 1
Consider 1
Intelligence-conspicuously 1
Sphex 1
Sphex-must 1
The 1
What 1
When 1
a 1
ability 1
absent 1
adapt 1
an 1
and 1
as 3
ascribed 1
away 1
behaviour 4
burrow 2
burrow, 1
but 1
carries 1
case 1
checks 1
circumstances. 1
clear, 1
coast 1
complicated 1
deposits 1
difference? 1
digger 1
displaced. 1
emerging, 1
entrance 1
even 1
female 1
few 1
first 1

```

Second Screenshot Output:

```

indication 1
insect 1
inside 1
inside. 1
inside: 1
instinctual 1
intelligence, 1
intelligence. 1
intruders 1
is 8
it 1
most 1
moved 1
nature 1
never 1
new 1
of 4
often 1
on 2
only 1
procedure 1
real 1
repeat 1
returns 1
revealed 1
she 3
simplest 1
taken 1
the 15
then, 1
threshold, 1
to 5
wasp 1
wasp, 1
wasp's 1
while 2
whole 1
will 1
with 1
[cloudera@quickstart ~]$

```