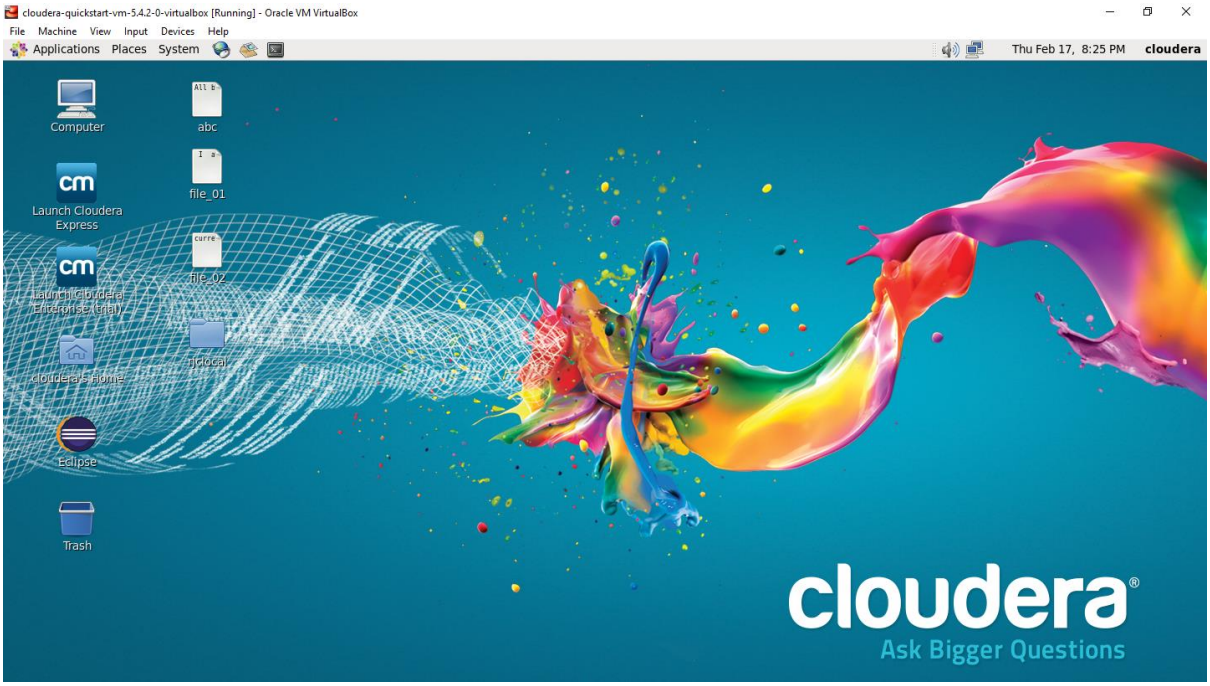
PRACTICAL NO 3

To Implement Wordcount problem using Hadoop

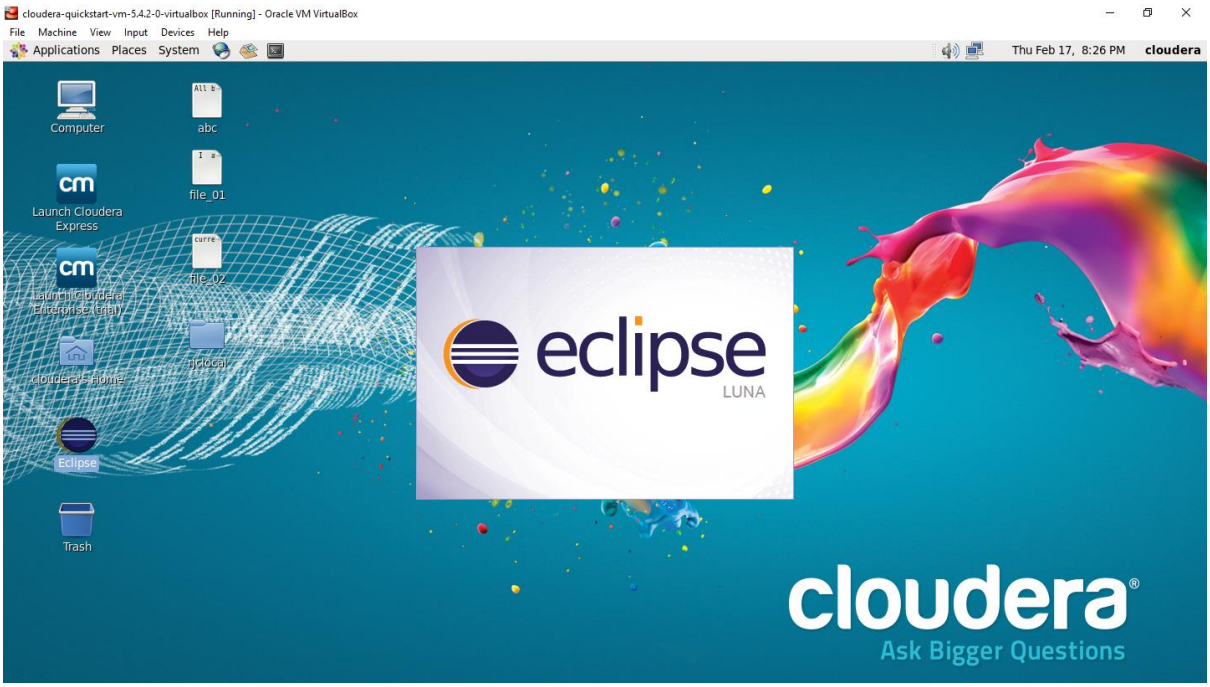
MapReduce in Eclipse: (With Combiner & Without Combiner)

- **Steps for Word Count in Cloudera**
  - ➢ **With Combiner**

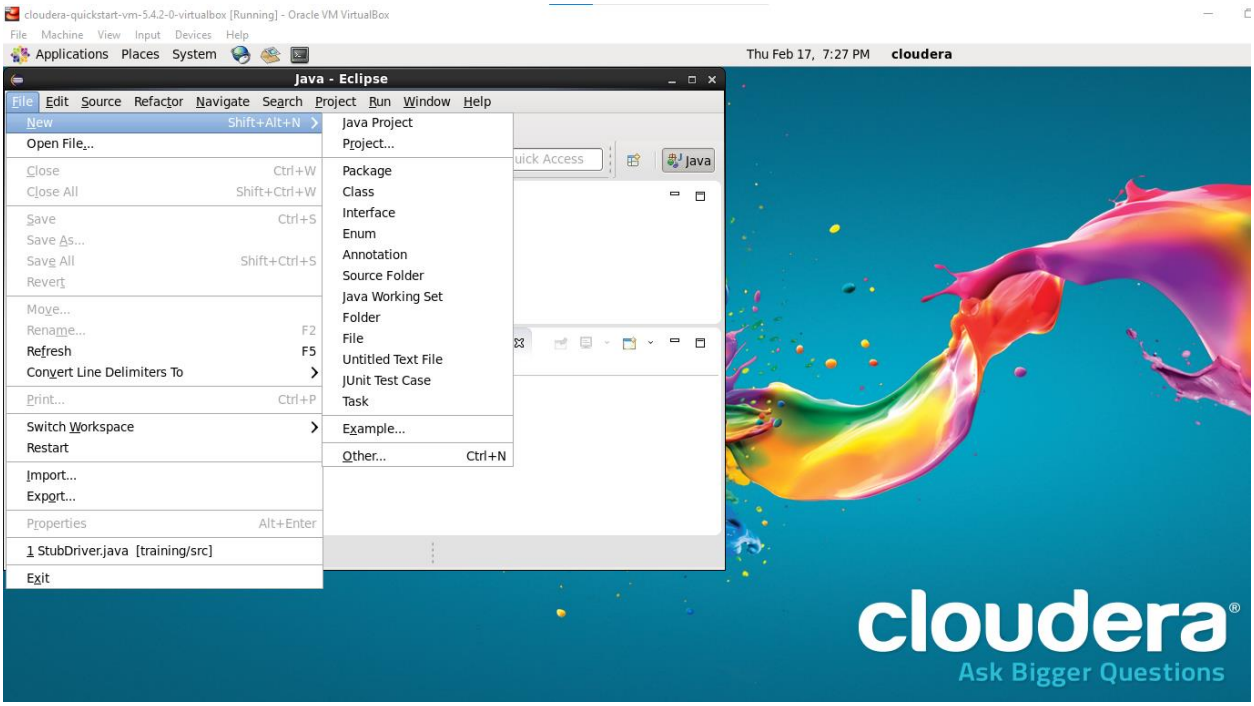
1) Open virtual box and then start cloudera quickstart

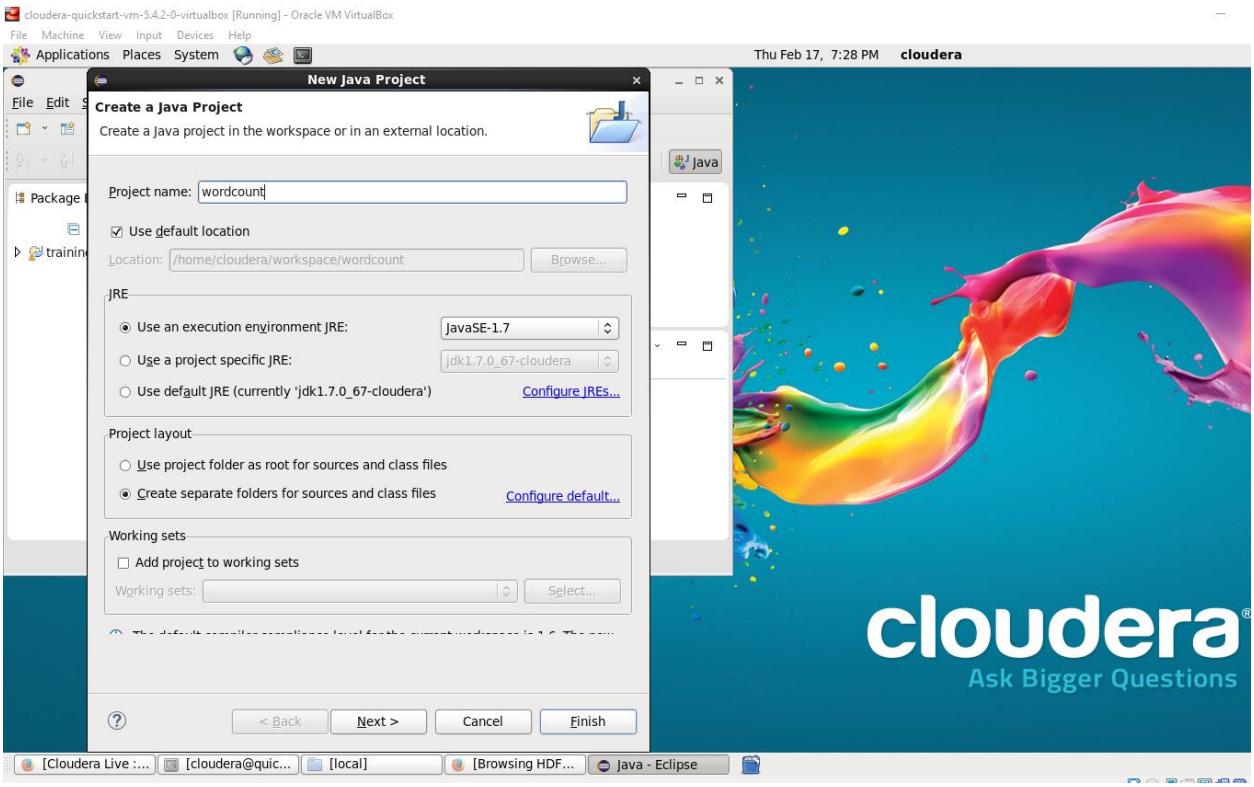


2) Open Eclipse present on the cloudera desktop



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

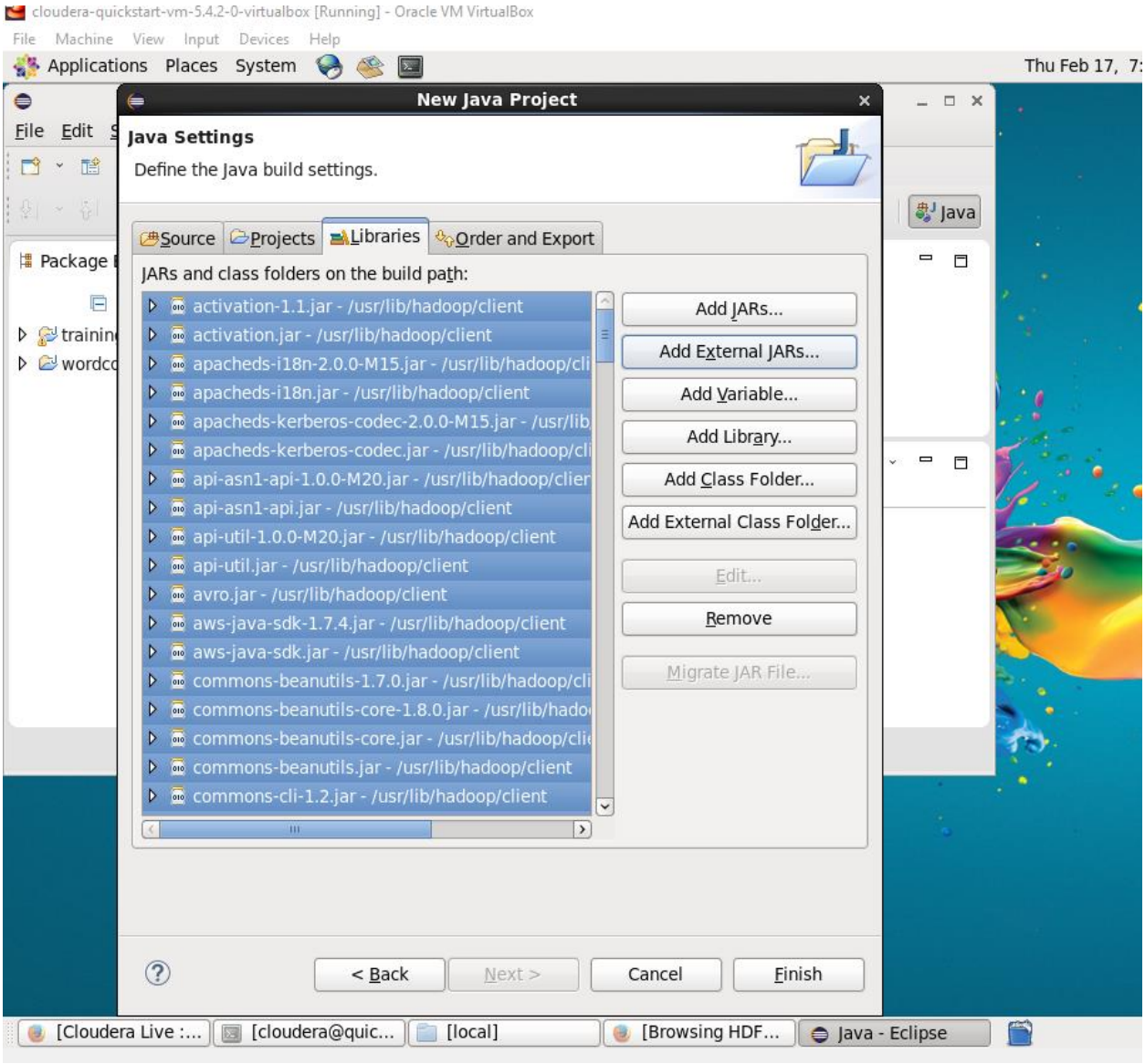




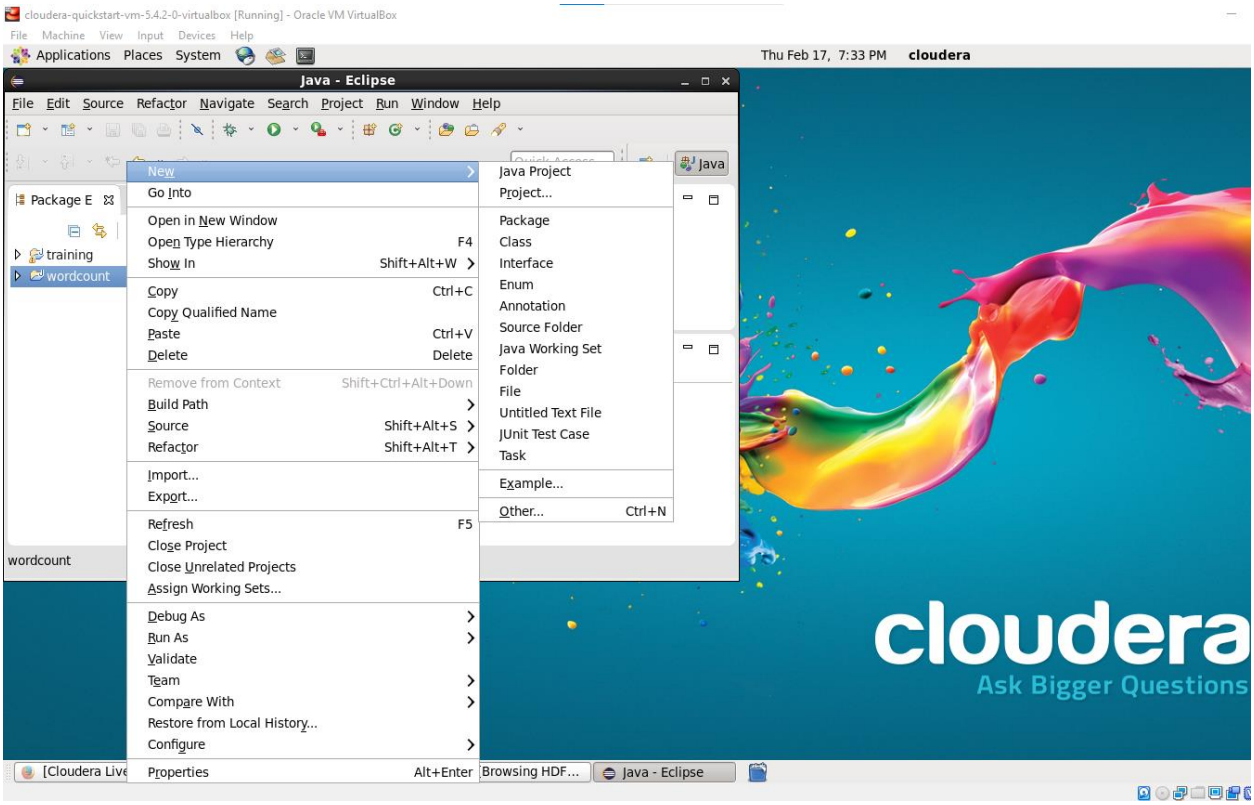
4) Adding the Hadoop libraries to the project Click on Libraries -> Add External JARs Click on

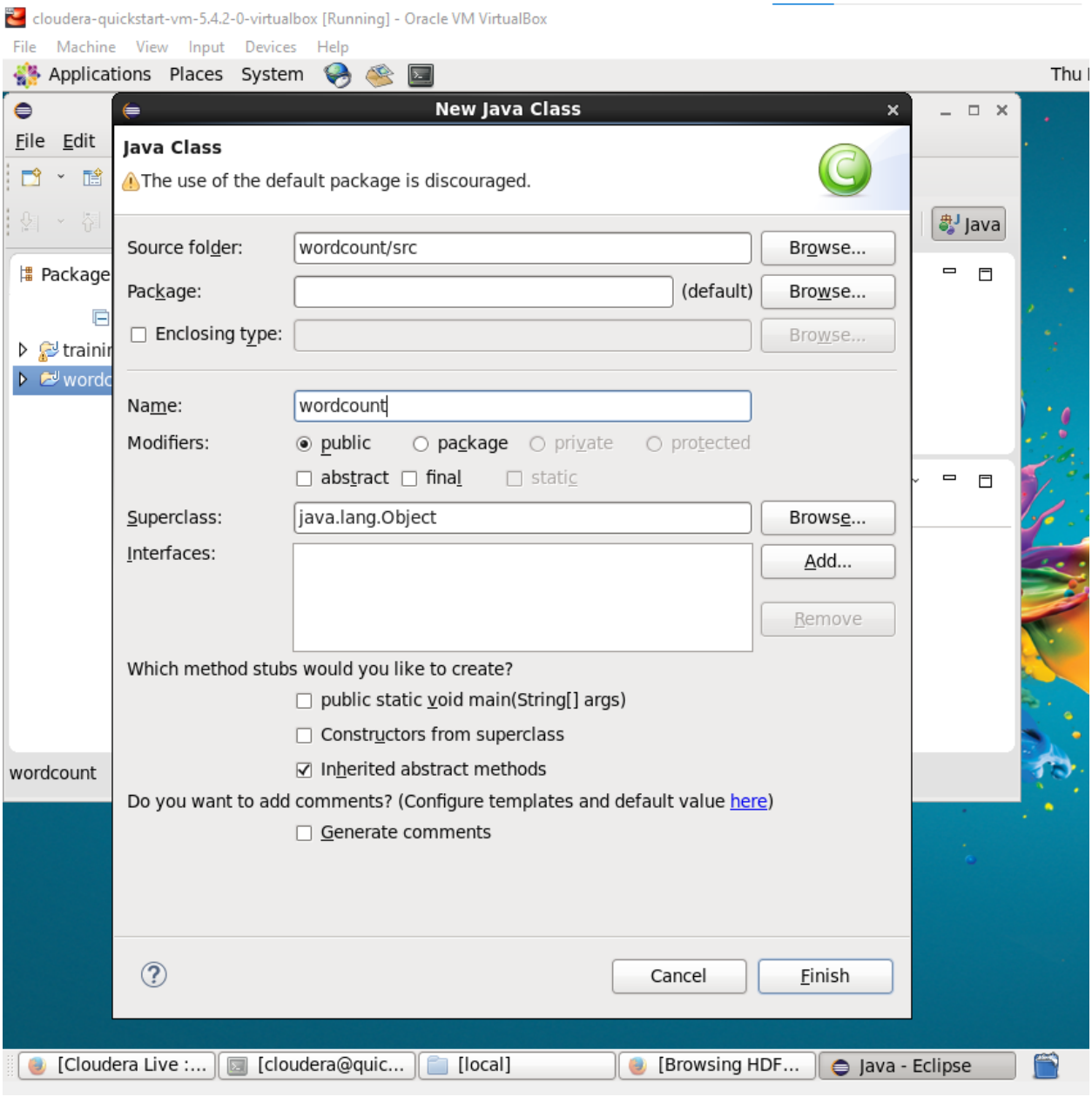
File System -> usr -> lib -> hadoop Select all the libraries (JAR Files) -> click OK Click on

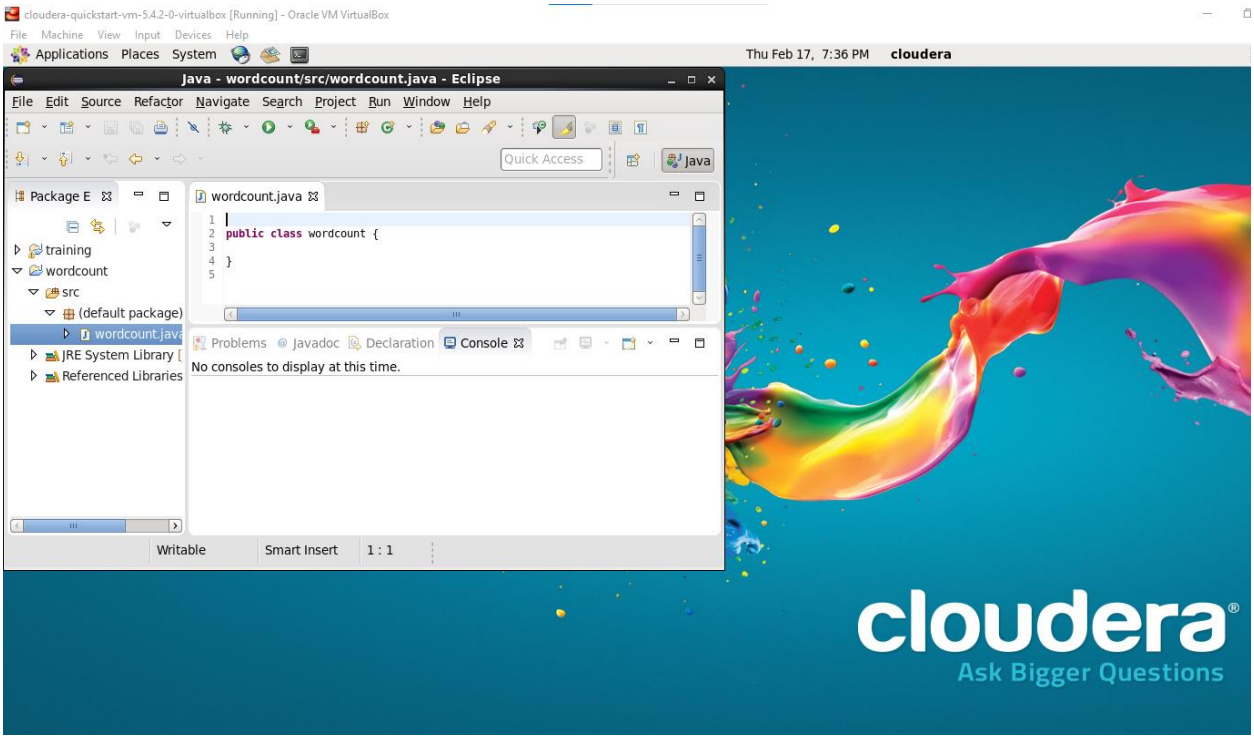
Add External jars, -> client -> select all jar files -> ok -> Finish



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

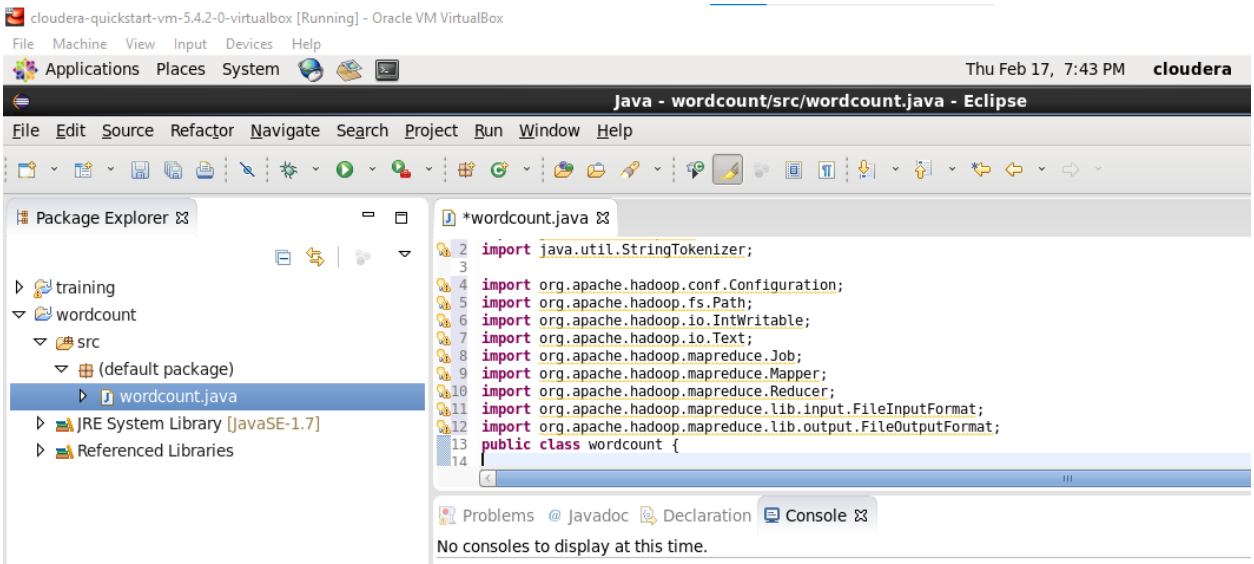

cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Thu Feb 17, 7:33 PM cloudera
Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package E
training
wordcount
New
Go Into
Open in New 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
Java Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
JUnit Test Case
Task
Example...
Other... Ctrl+N
wordcount
[Cloudera Live
Browsing HDF...
Java - Eclipse
cloudera
Ask Bigger Questions


cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
New Java Class
Java Class
The use of the default package is discouraged.
Source folder: wordcount/src
Browse...
Package:
(default)
Enclosing type:
Name: wordcount
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Cancel
Finish
wordcount
[Cloudera Live :...
[cloudera@quic...
[local]
[Browsing HDF...
Java - Eclipse

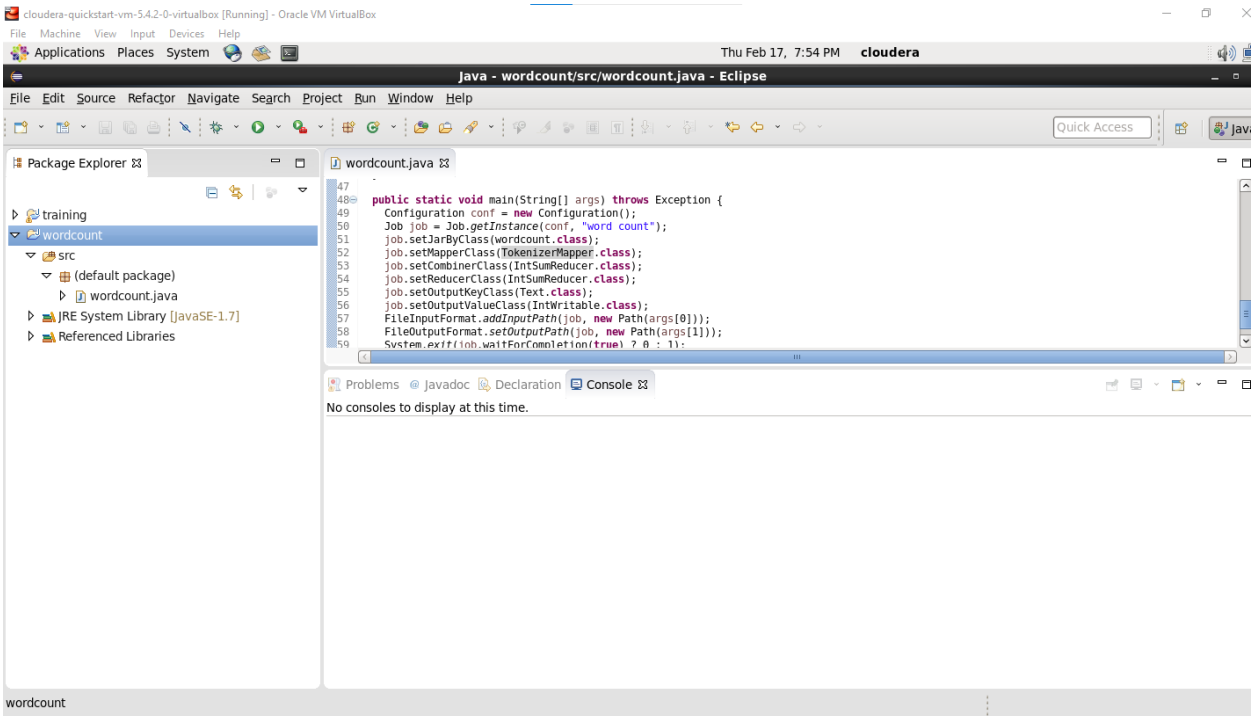


Source code:

Packages

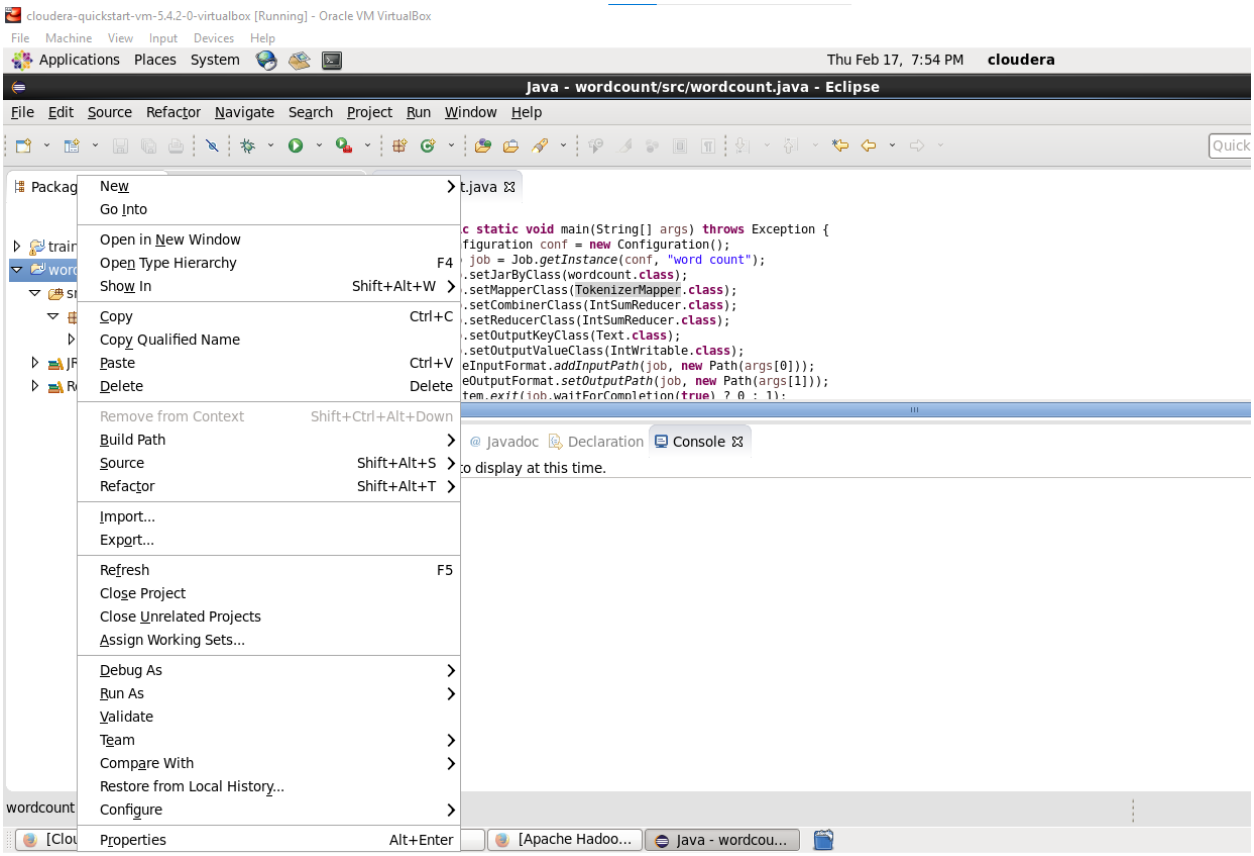


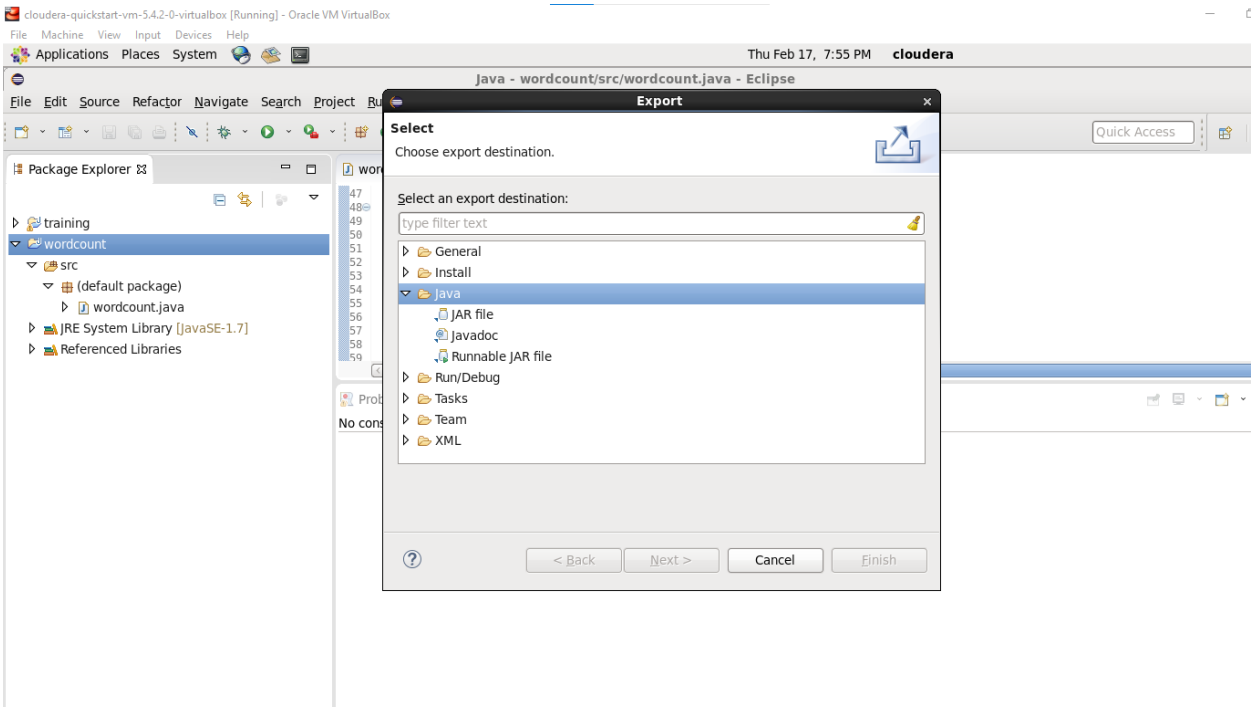
Mapper Logic

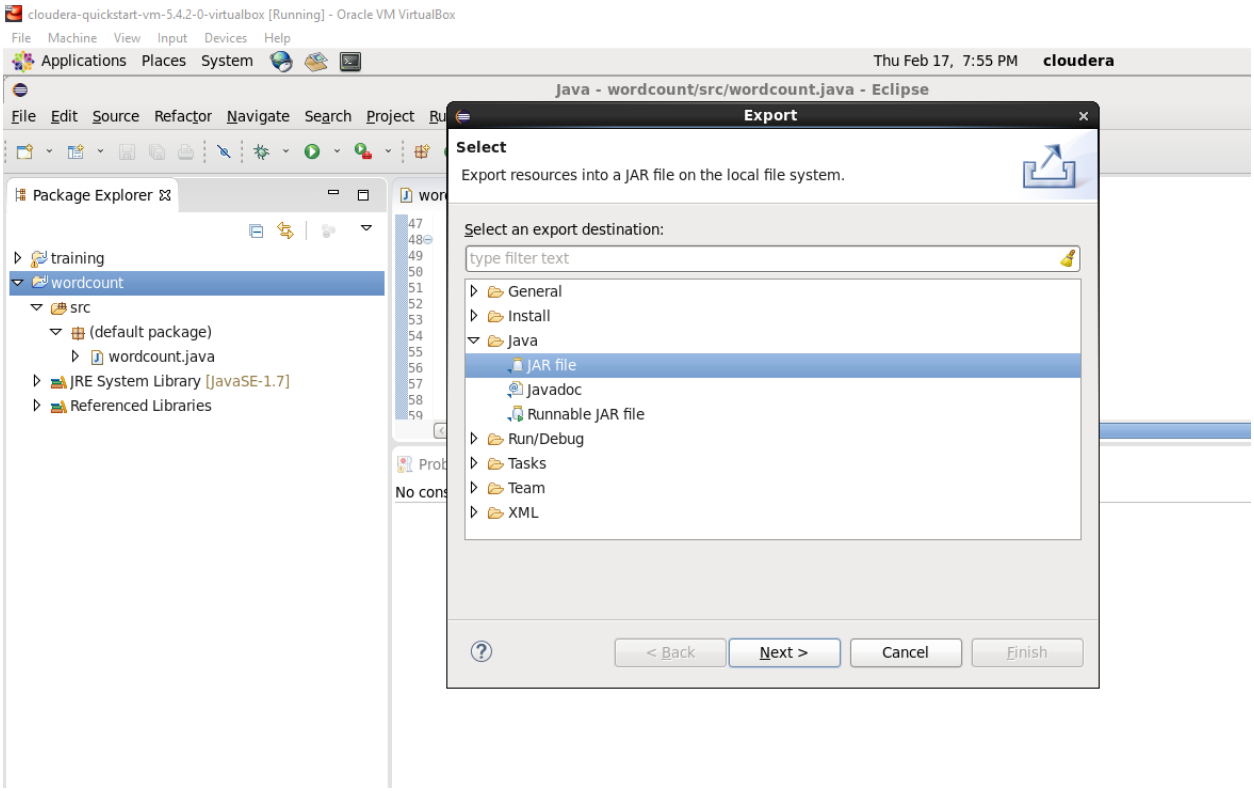


6) Right Click on the project name WordCount -> Export -> Java -> JAR File -> Next -> for select the export destination for JAR file: browse -> Name : WordCount.jar -> save in folder

-> cloudera -> Finish -> OK


cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Thu Feb 17, 7:54 PM cloudera
Java - wordcount/src/wordcount.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
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
wordcount
[Apache Hadoo...
Java - wordcou...


cloudera-quickstart-vm-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Thu Feb 17, 7:55 PM cloudera
Java - wordcount/src/wordcount.java - Eclipse
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
Package Explorer
training
wordcount
src
(default package)
wordcount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries



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

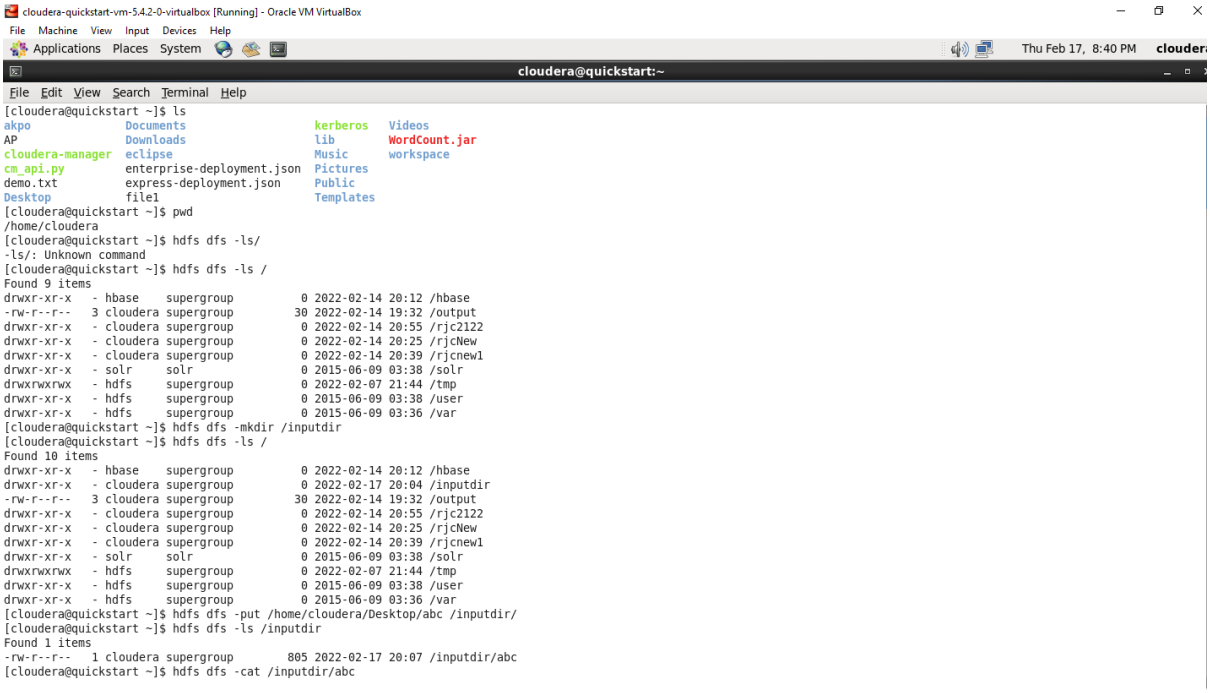
Creating an input file named as "abc".

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

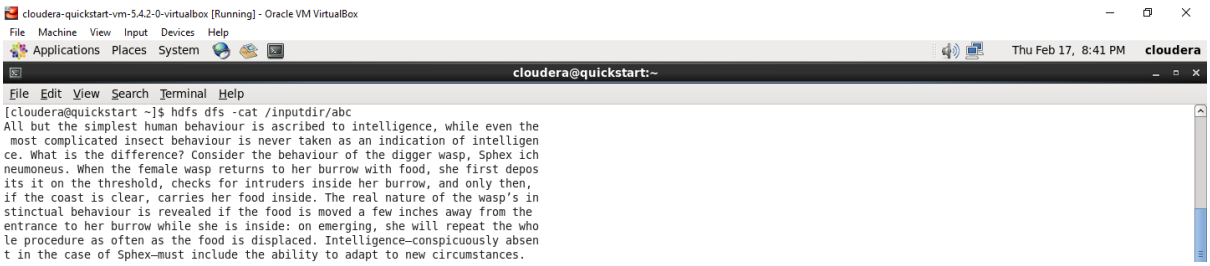
Then we can verify whether this directory is created or not using ls command hdfs dfs -ls /

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

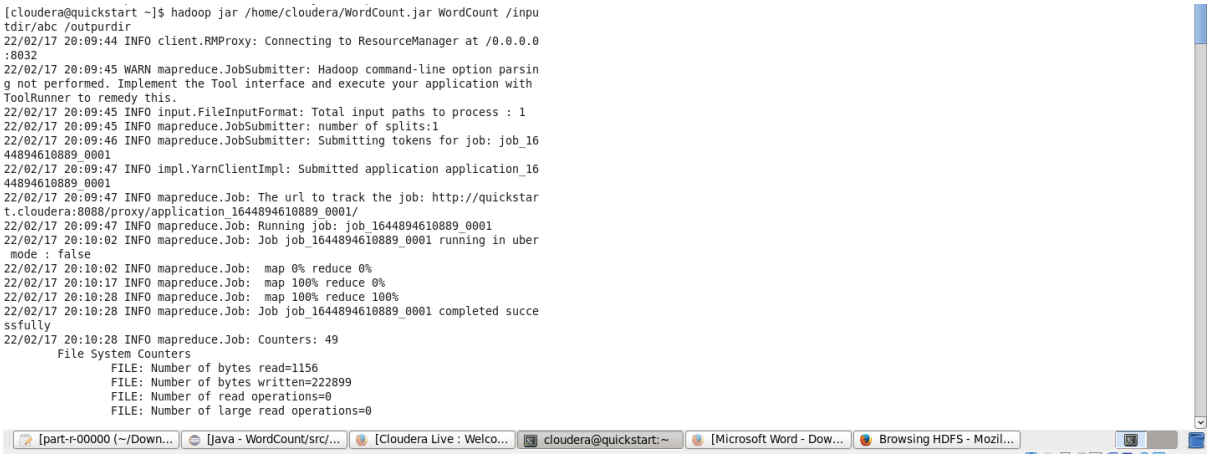


As we can see “abc” file is present in /inputdir directory of hdfs. Now we will see the content of this file using hdfs dfs –cat /inputdir/abc command

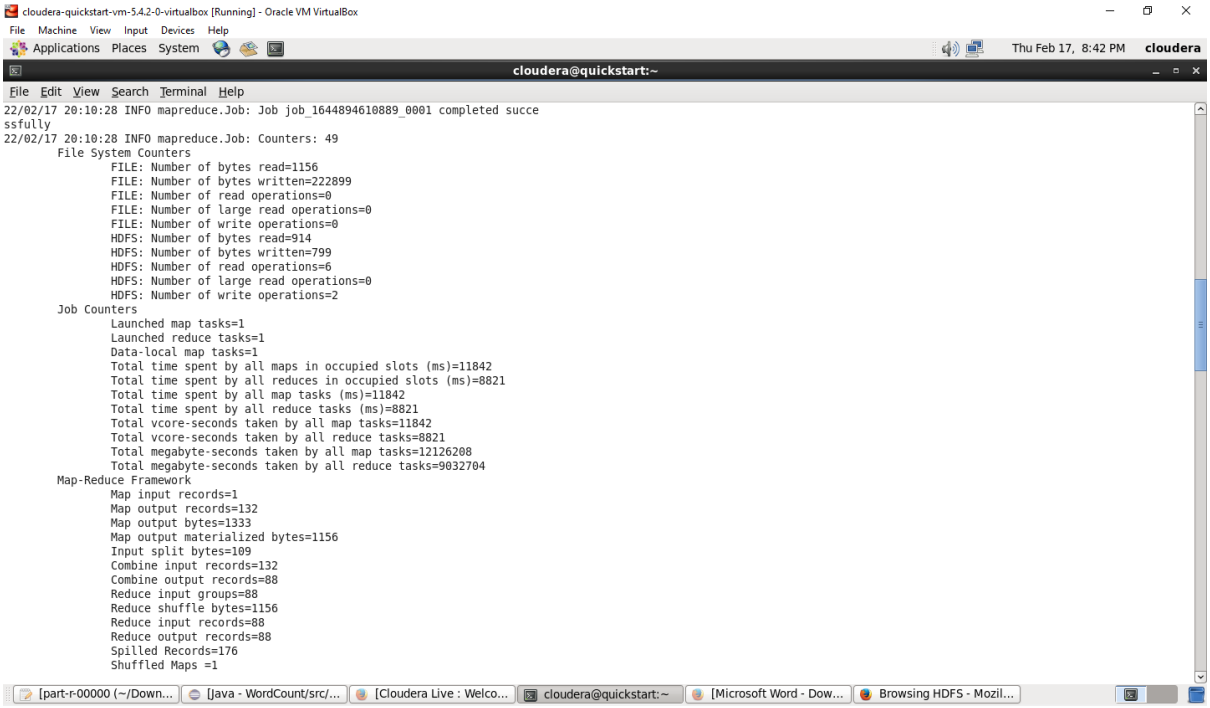


10) Running Mapreduce Program on Hadoop, syntax is hadoop jar jarFileName.jar ClassName /InputFileAddress /outputdir

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=132

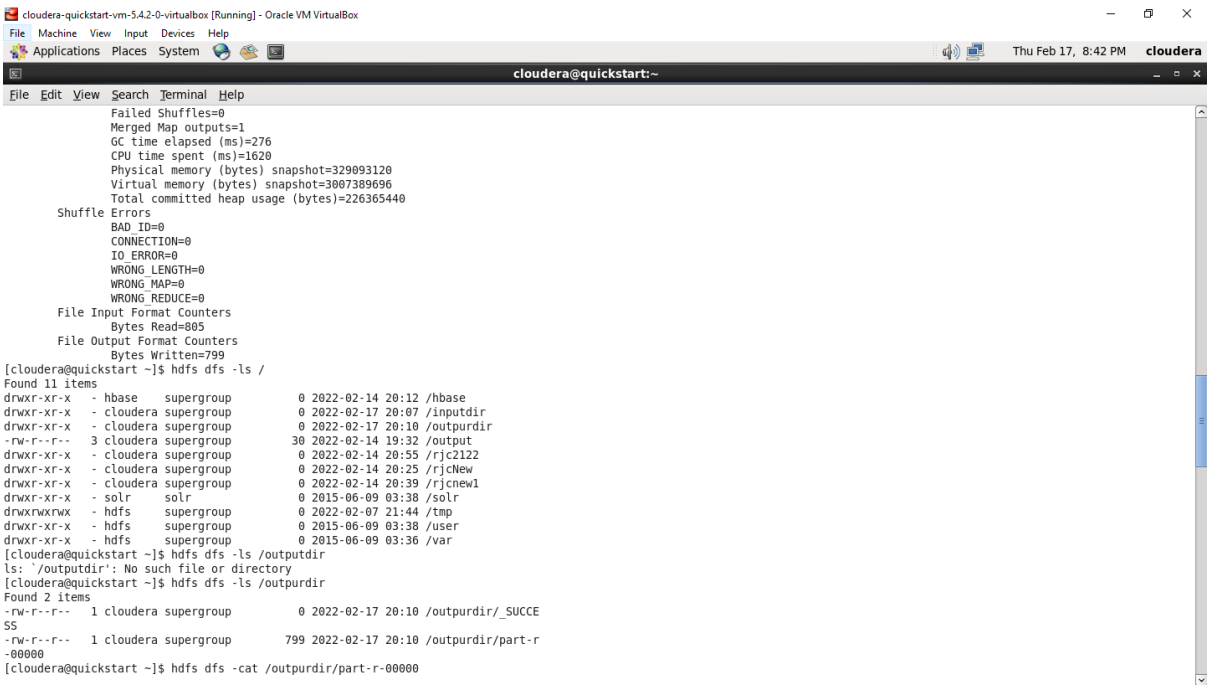
Combine output records=88

And Reduce shuffle bytes coming as,

Reduce shuffle bytes=1876

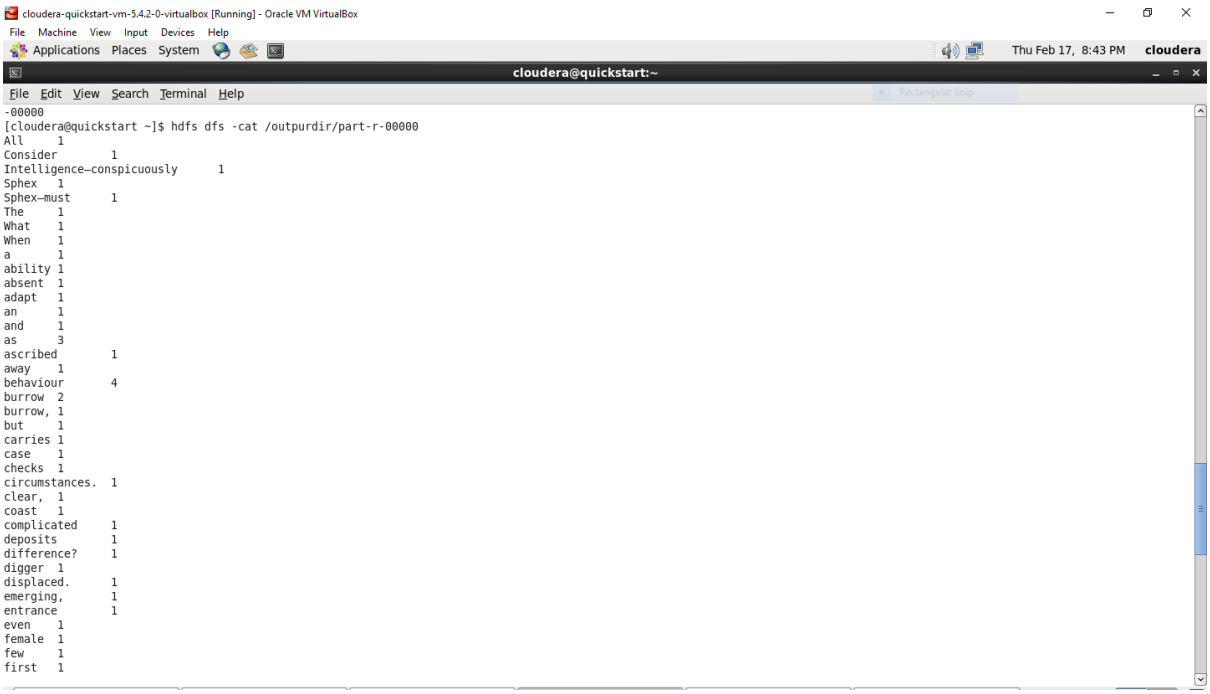
11) Then we can verify the content of outputdir directory and in that part-r file has the actual output by using the command Hdfs dfs -cat /outputdir/part-r-00000 This will give us final output. The same file can also be accessed using a browser. For every execution of this program we need to delete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

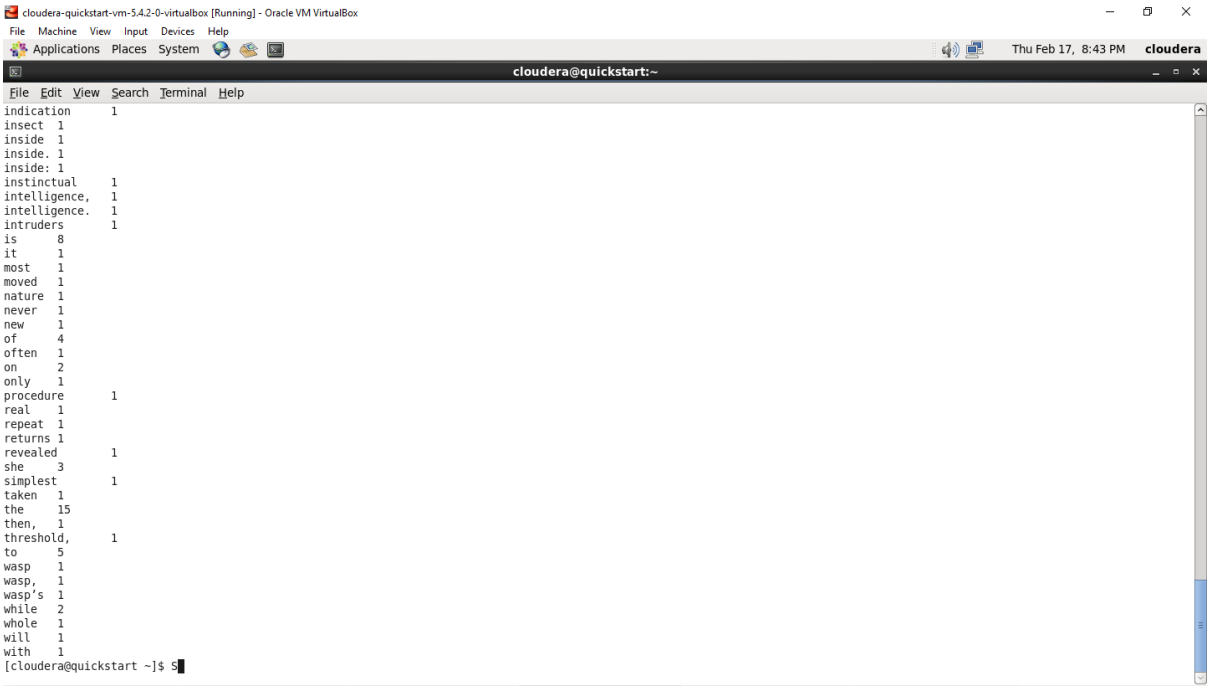
1st we are checking whether the outputdir directory is created in hdfs or not using command **hdfs dfs -ls /**



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

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**



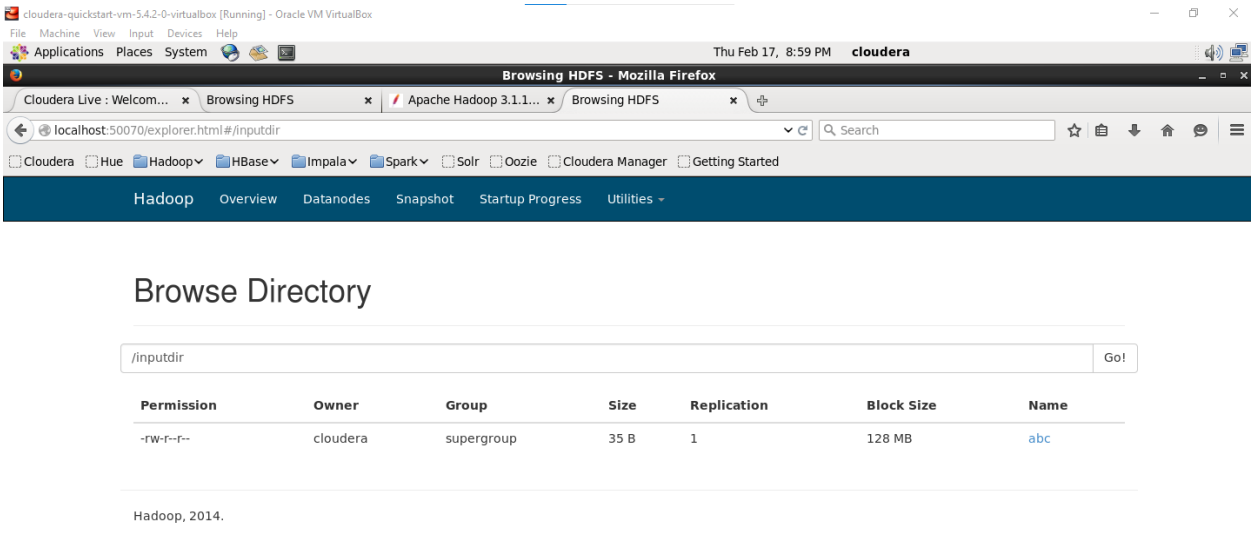


It will give the count of number of times each word has occurred as output.

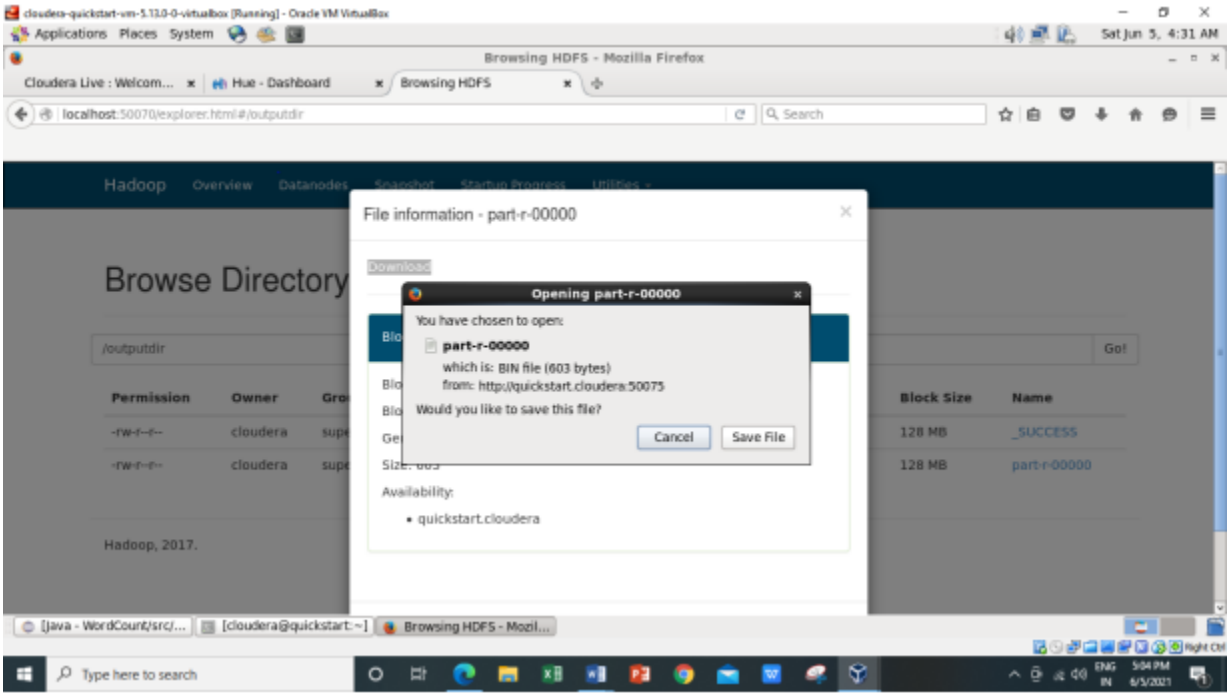
12) The same file can also be accessed using a browser.

Browse the Directory by

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



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

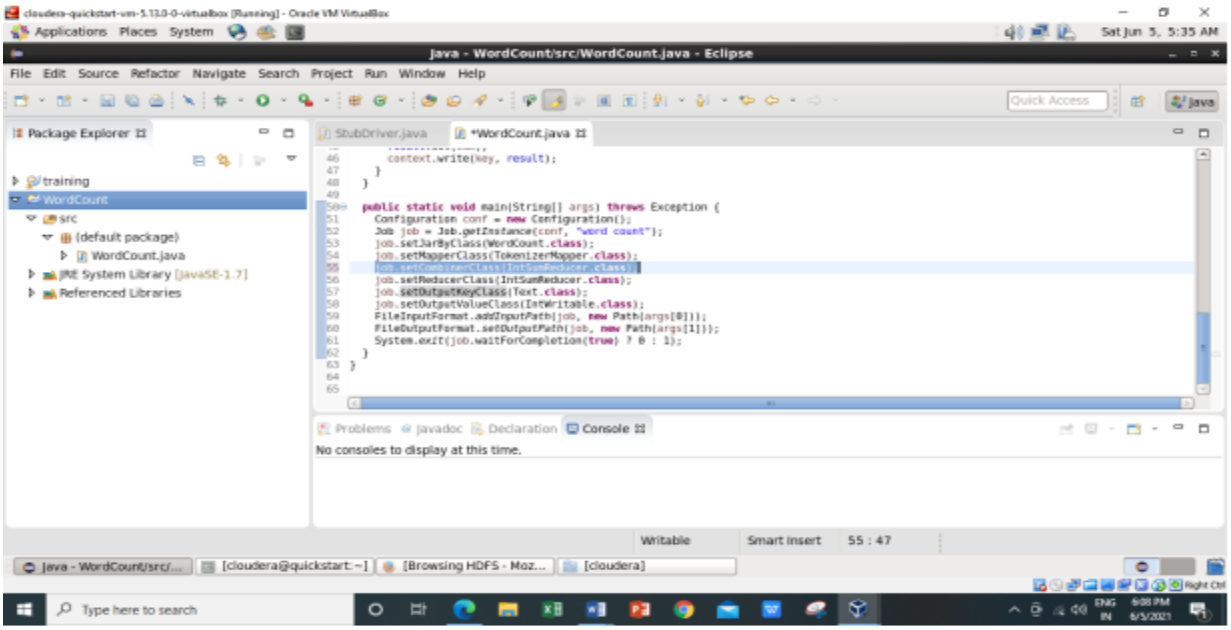


Inside the **part-r-00000** file it will have the same output as we are getting after executing using command **hadoop jar /home/cloudera/WordCount.jar WordCount /inputdir/abc /op1**

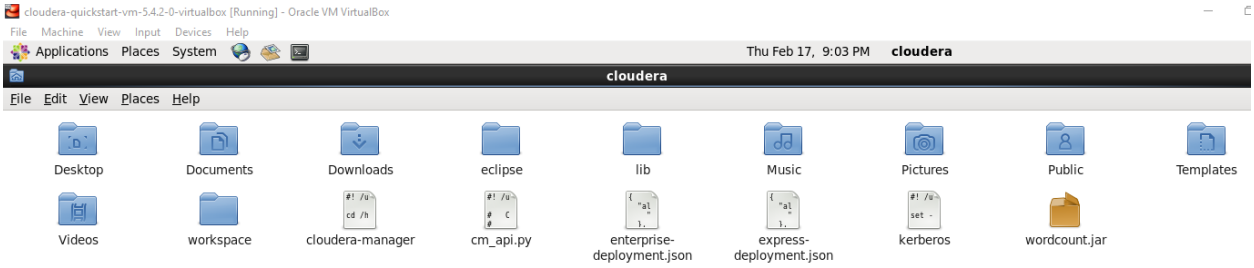
For every execution of this program we need to delete the output directory or give a new name to the output directory every time.

➢ Implementation of WordCount problem using Hadoop MapReduce (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**.**

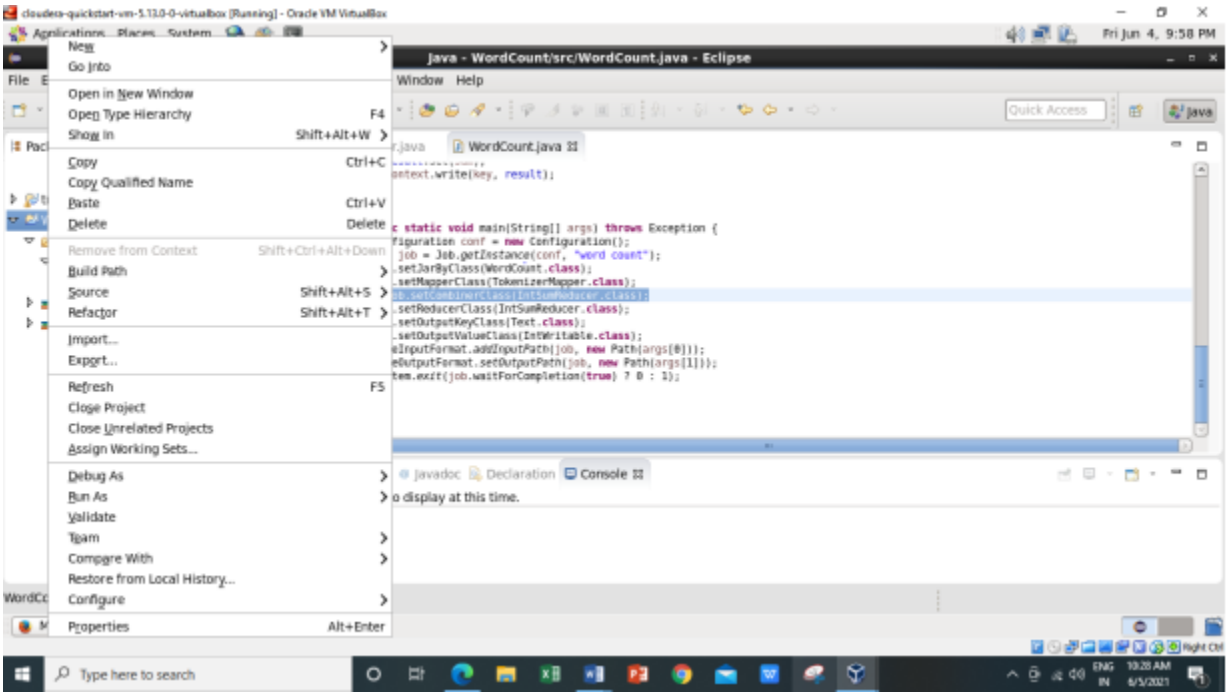


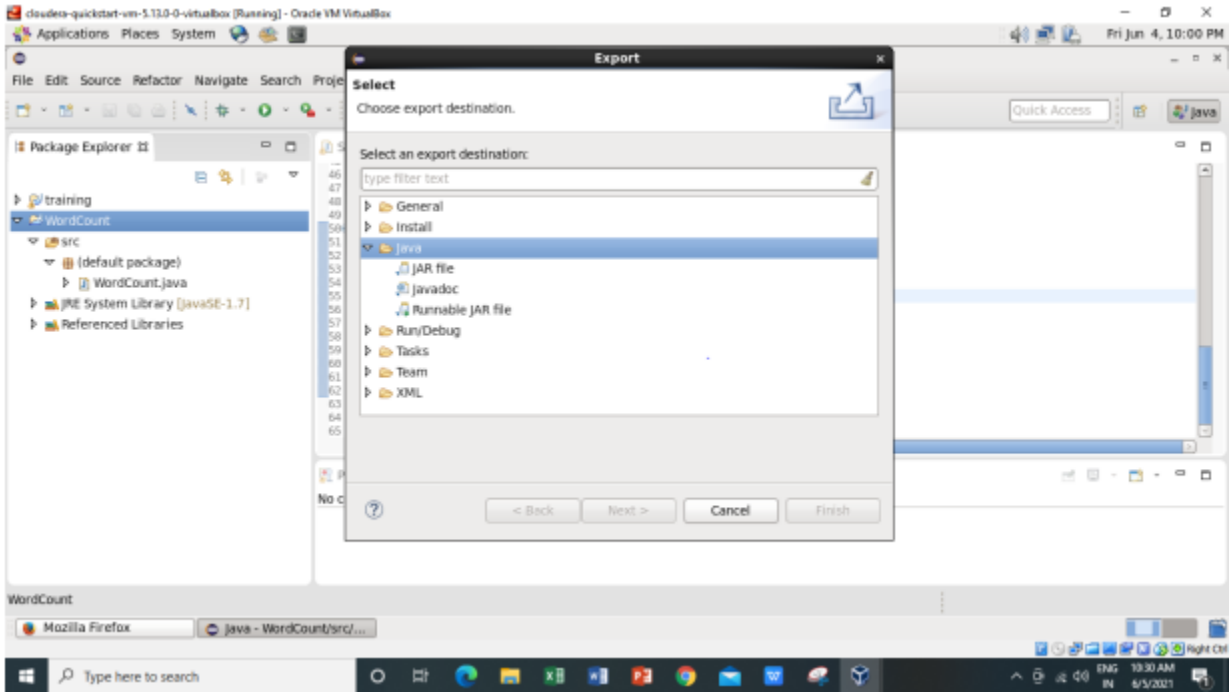
2) And will delete the WordCount.jar file in which all jar files are present from **/home/cloudera.**

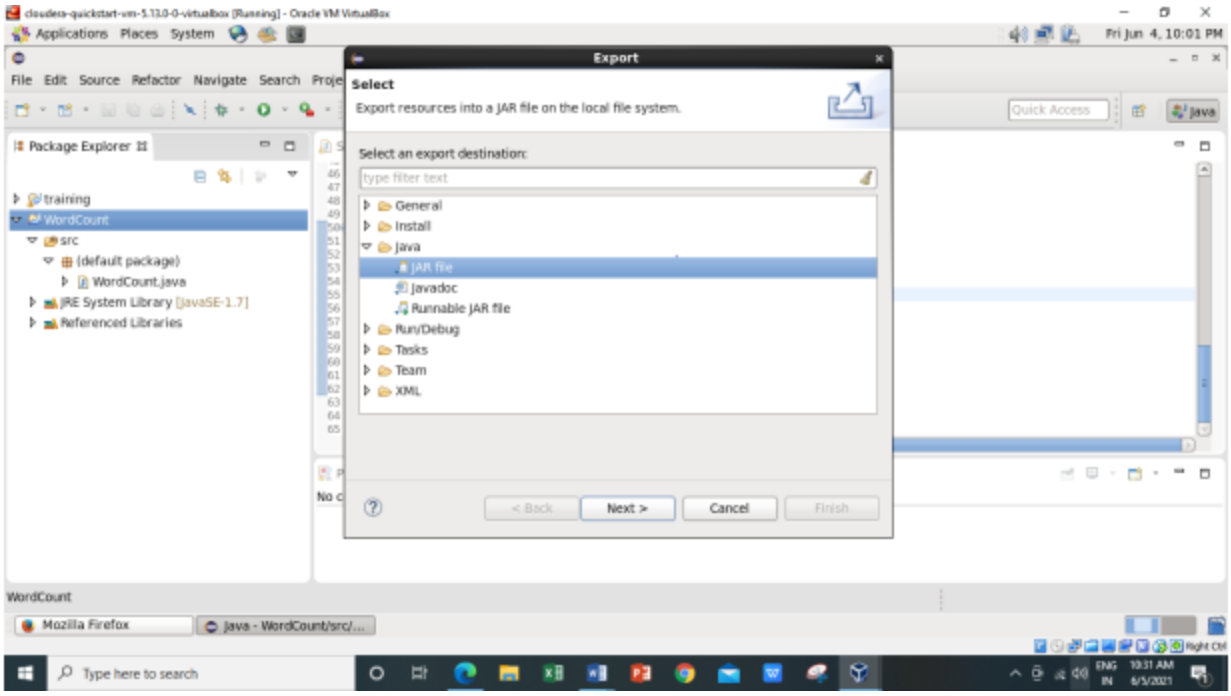


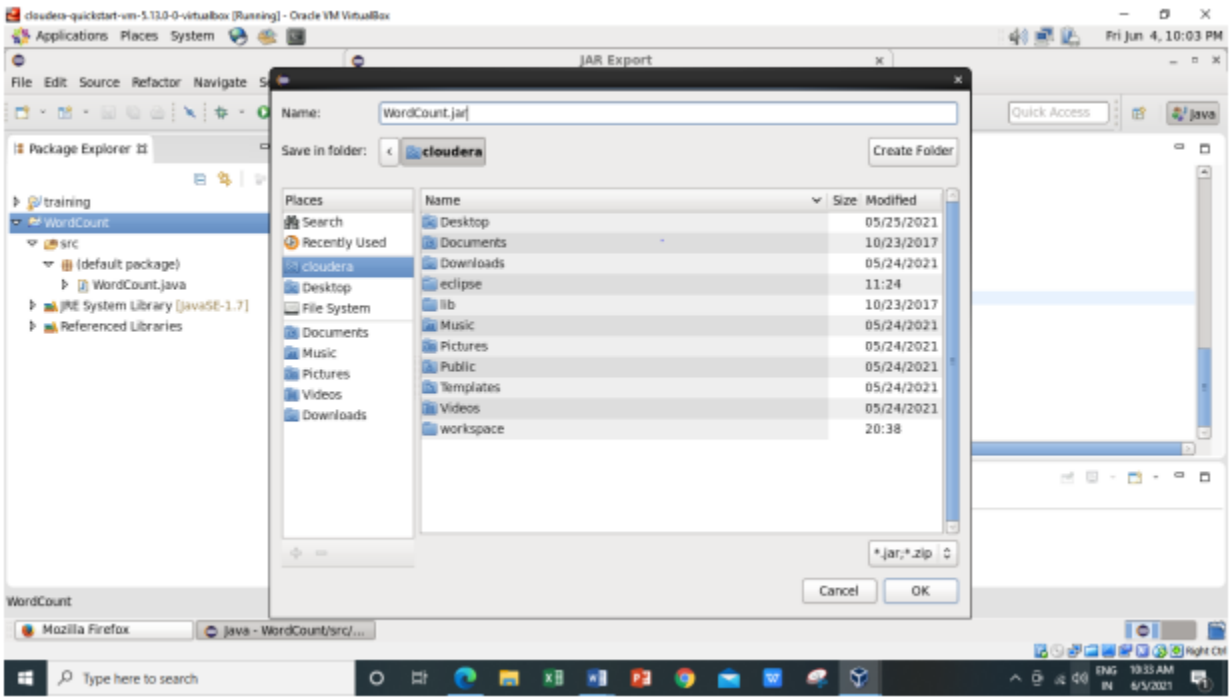
We have successfully deleted the WordCount.jar file.

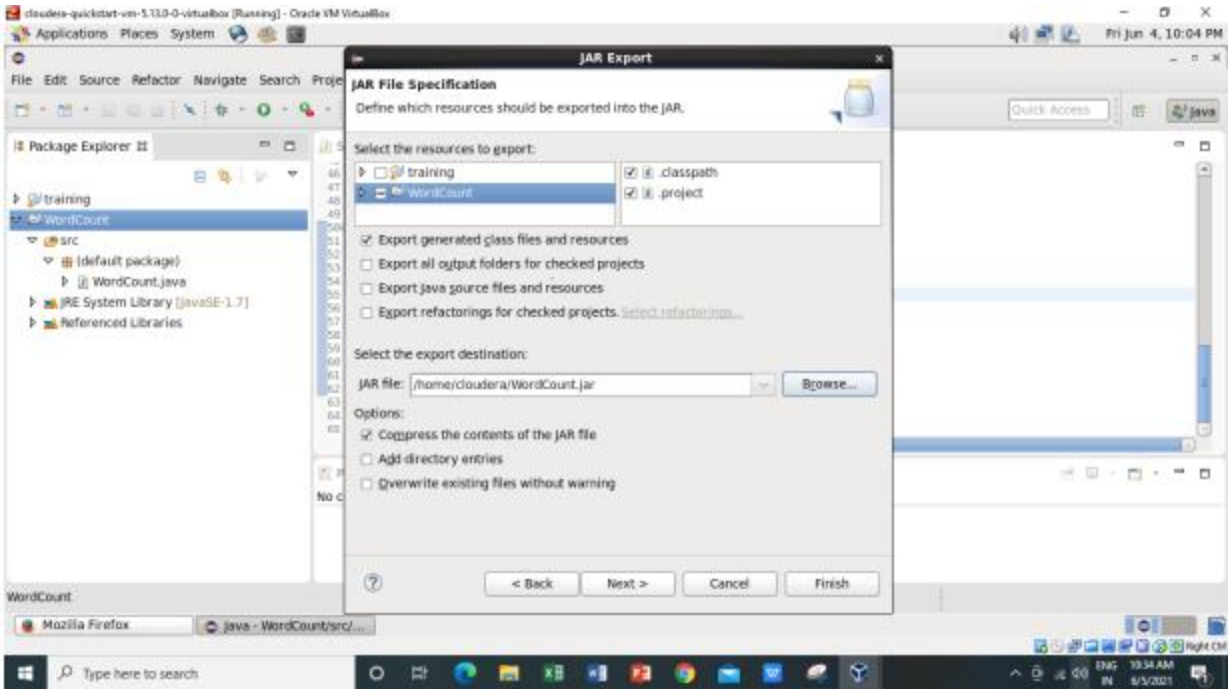
3) Now exporting the jar files Right Click on the project name WordCount -> Export -> Java ->JAR File -> Next -> for select the export destination for JAR file: browse -> Name : WordCount.jar -> save in folder -> cloudera -> Finish -> OK






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
WordCount
Mozilla Firefox
java - WordCount/src/...
Type here to search


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
WordCount
Mozilla Firefox
java - WordCount/src/...
Type here to search



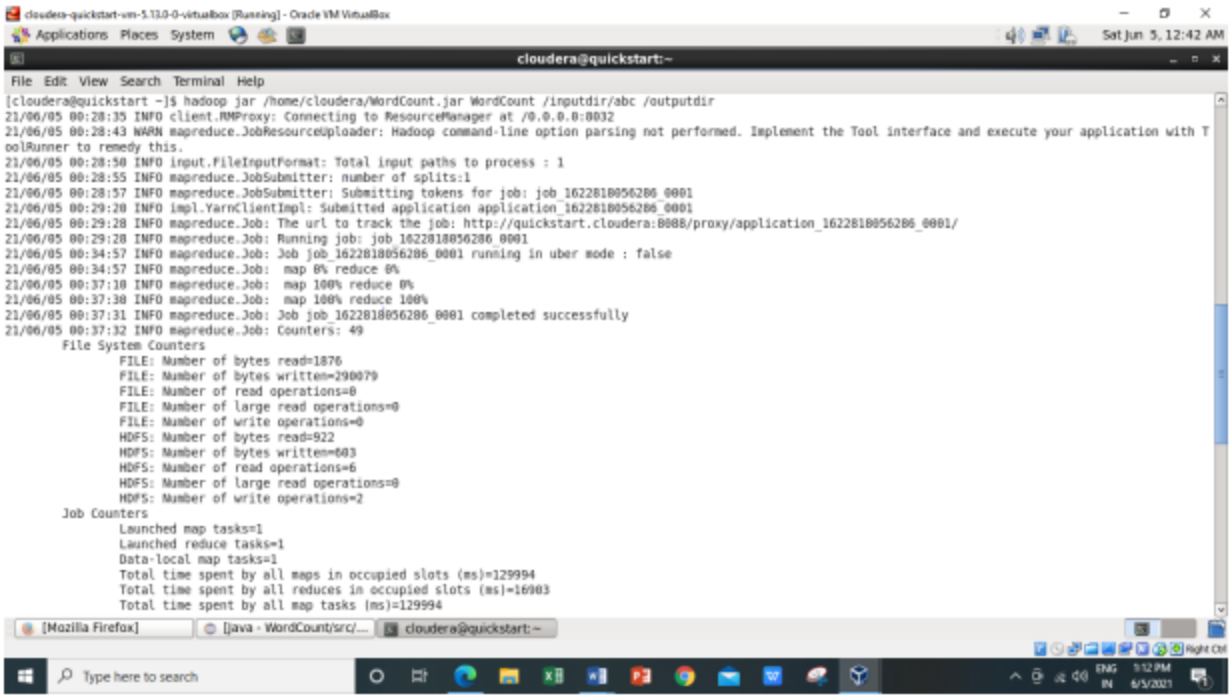
4) Now checking the WordCount.jar file is created or not using –ls command

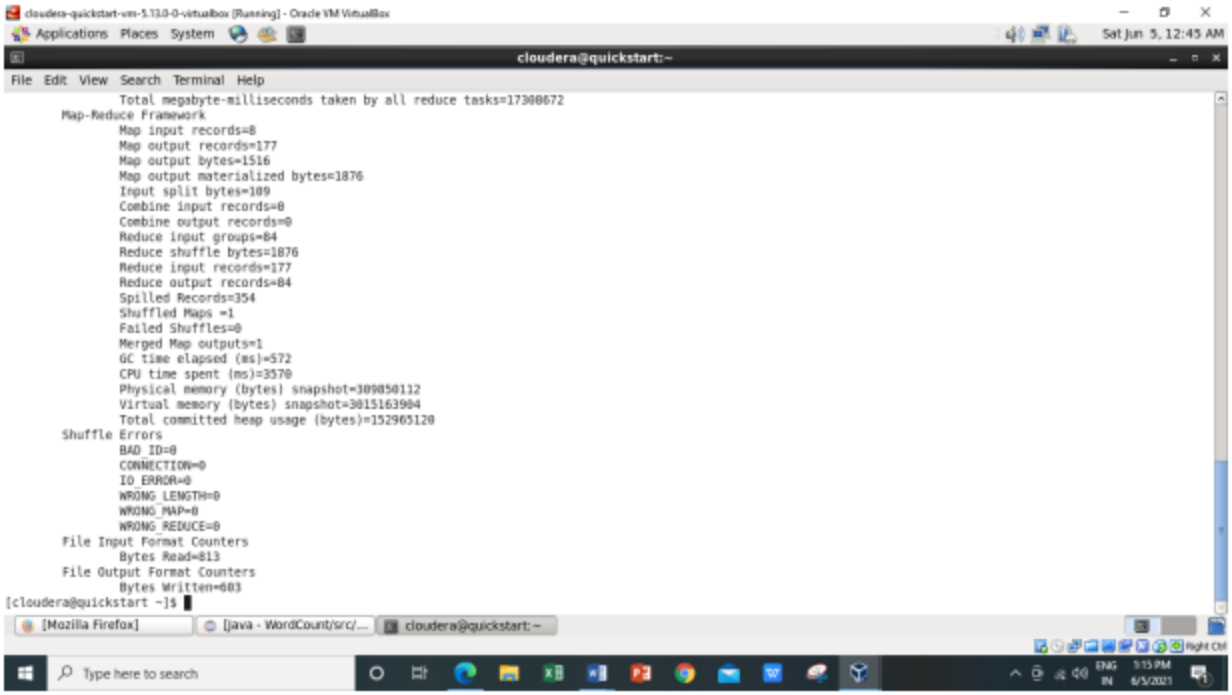
```
[cloudera@quickstart ~]$ ls
cloudera-manager  Desktop    Downloads  enterprise-deployment.json  kerberos  Music    Pictures  Templates  WordCount.jar
cm_api.py         Documents  eclipse    express-deployment.json     lib       parcels  Public    Videos     workspace
[cloudera@quickstart ~]$
```

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=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,

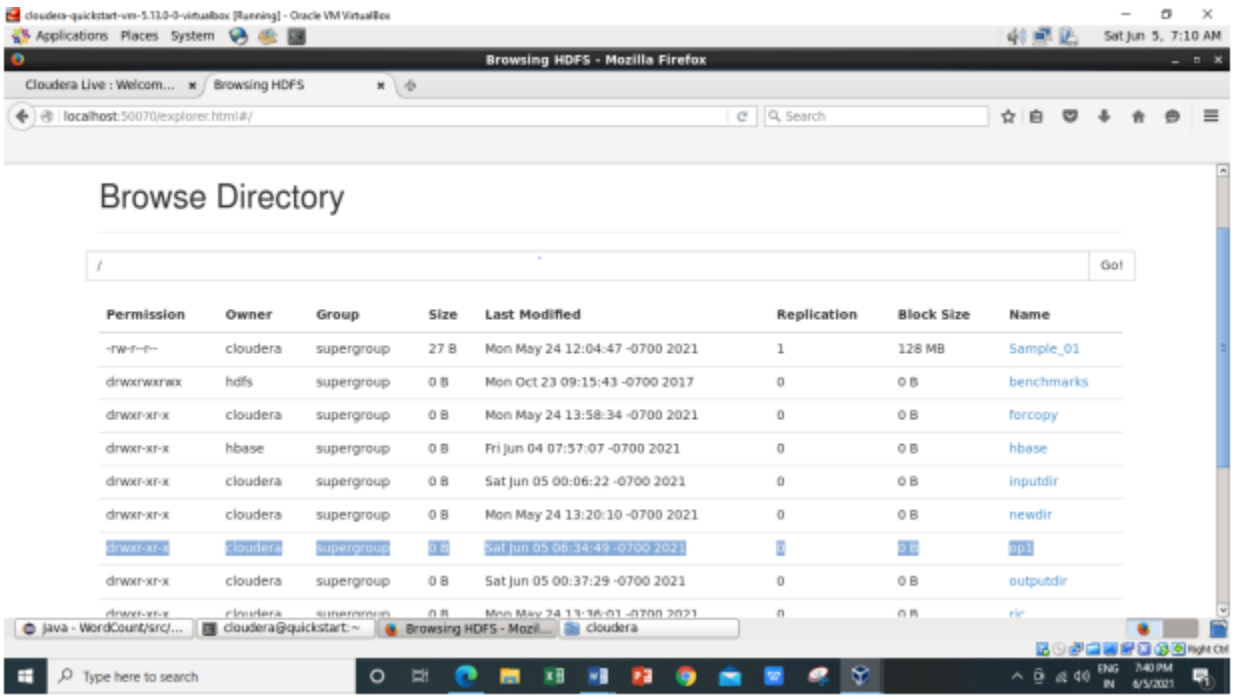
  Reduce shuffle bytes=1876

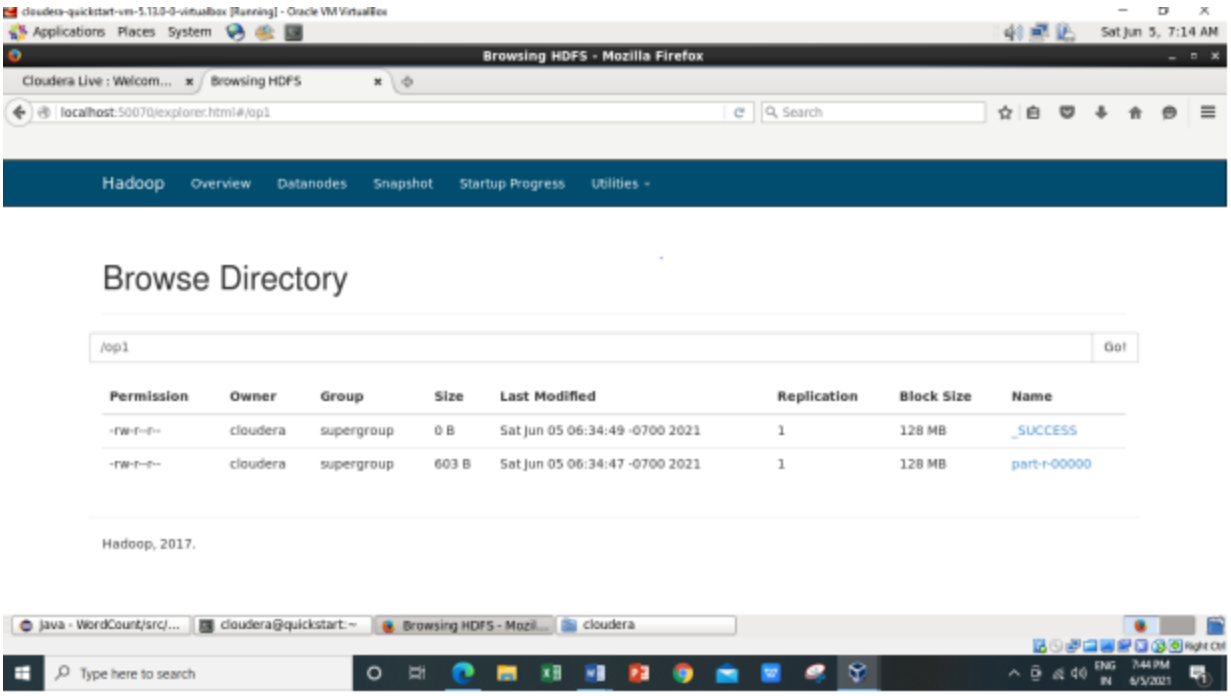
- So Combiner is used to save the Network Bandwidth. So for saving the Network bandwidth we make use of combiner. So instead of sending every word over the network what we do is we incorporate the logic of the reducer at the combiner side so that the less amount of information can be transmitted over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

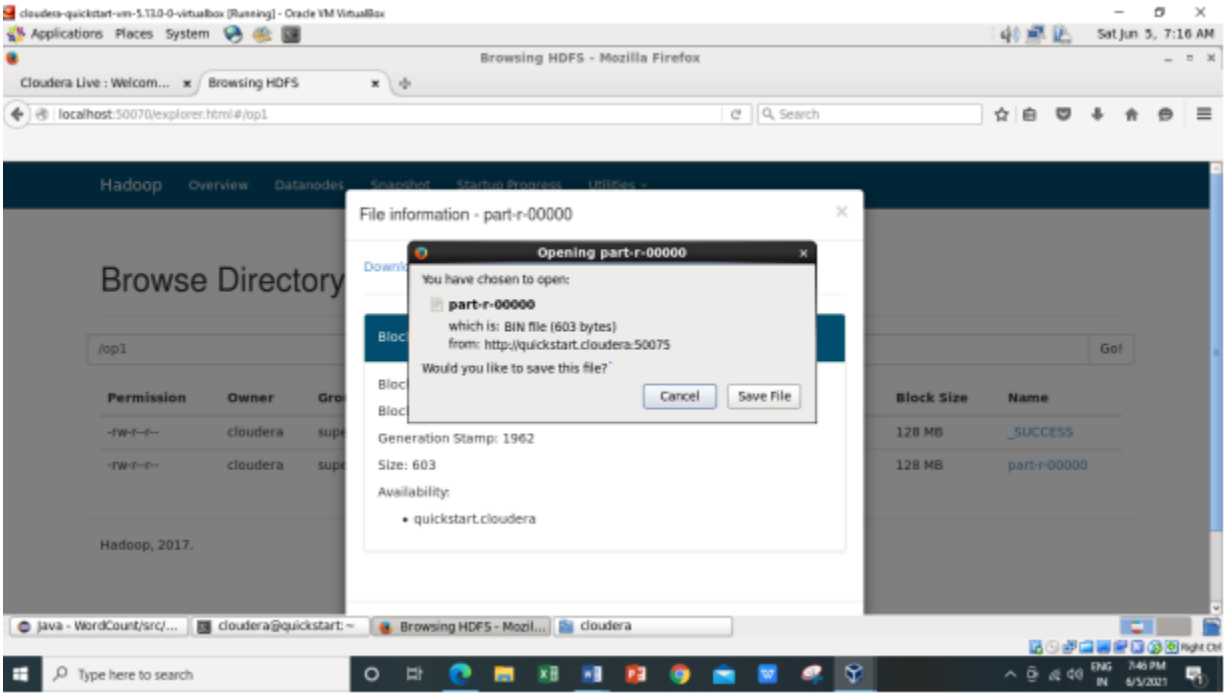
Browse the Directory by

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

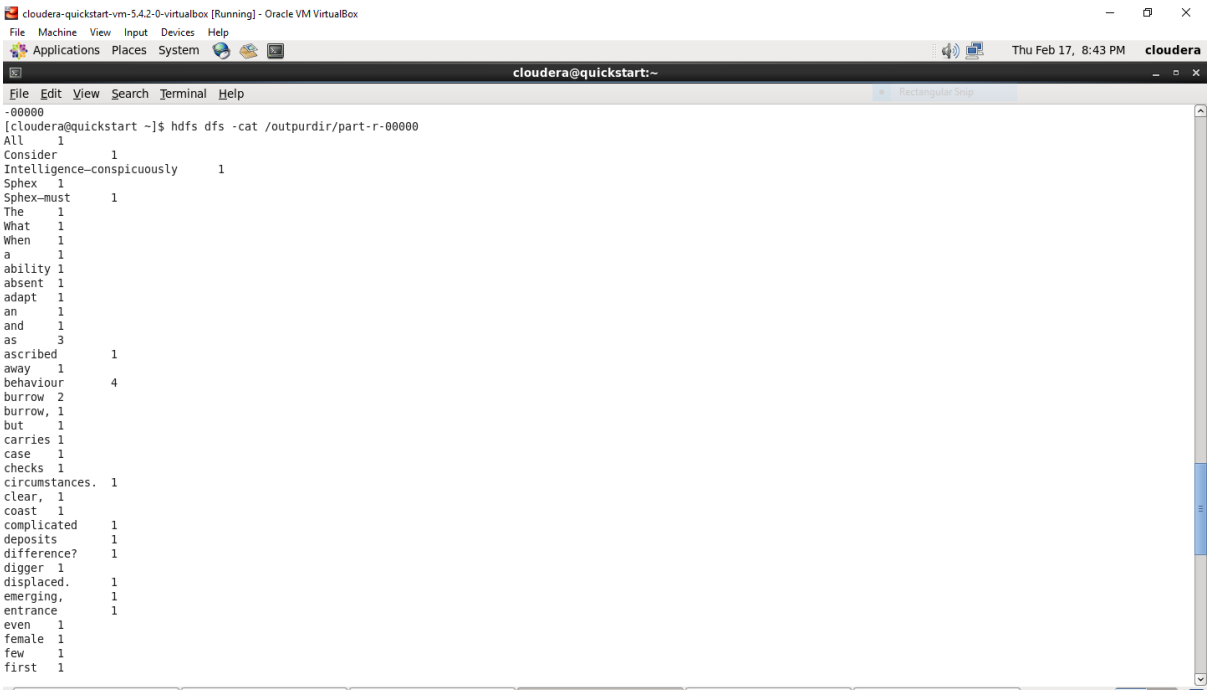


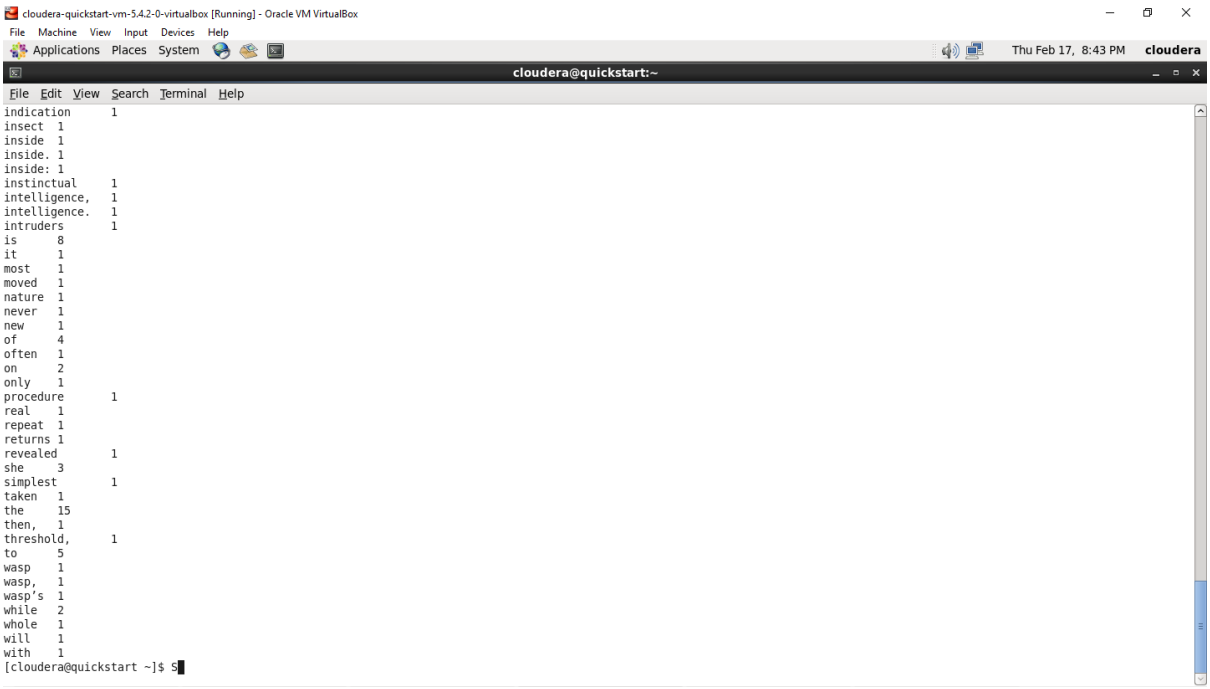


Now downloading the **part-r-00000** file.



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-5.4.2-0-virtualbox [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
Thu Feb 17, 8:43 PM cloudera
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
Thu Feb 17, 8:43 PM cloudera
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 ~]$ s