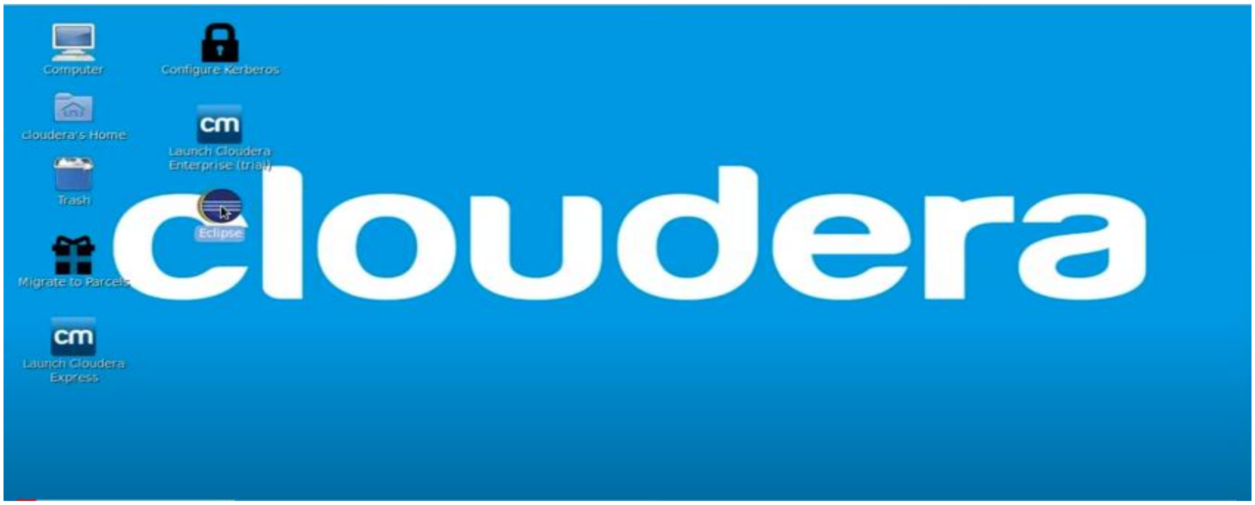
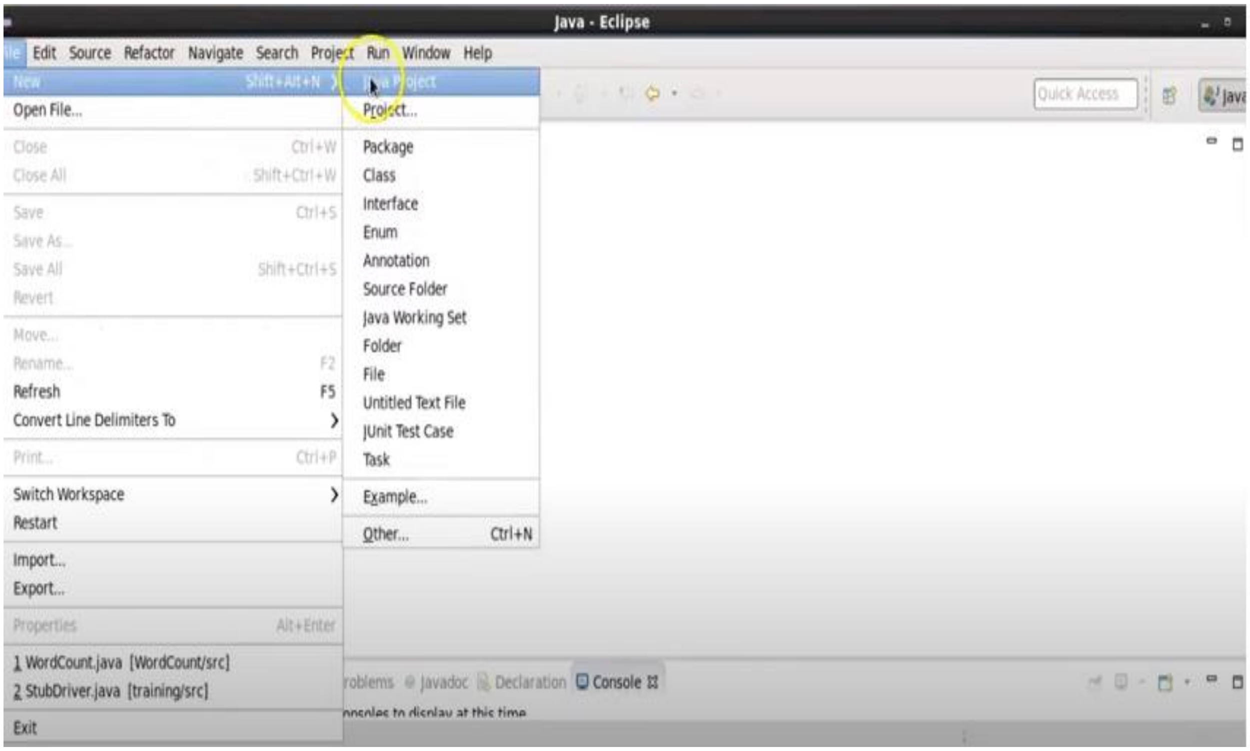
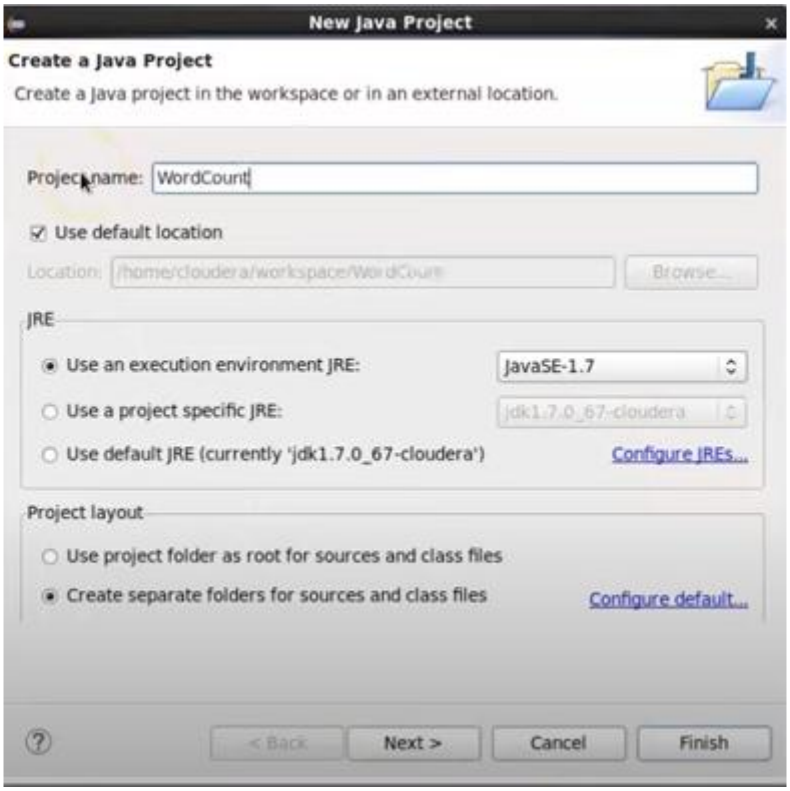
1. Cloudera desktop → Eclipse Icon



2. In Eclipse window, File → New → Java Project



3. Give the Project name as **WordCount**



New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: WordCount

☑ Use default location

Location: /home/cloudera/workspace/WordCount

JRE

● Use an execution environment JRE: JavaSE-1.7

○ Use a project specific JRE: jdk1.7.0_67-cloudera

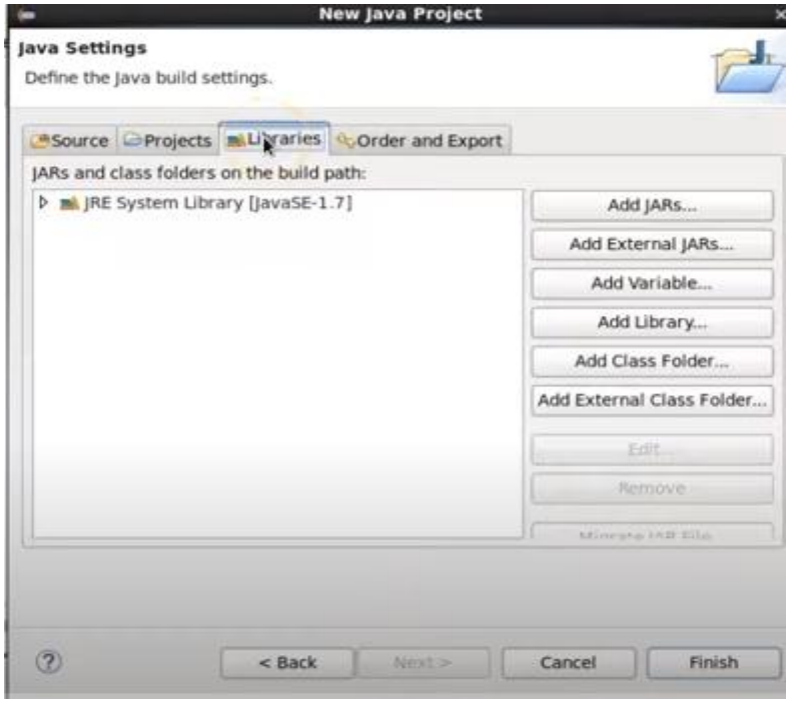
○ Use default JRE (currently 'jdk1.7.0_67-cloudera')

Project layout

○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

4. Click **next** button. Don't click finish button
5. Click **Libraries**



New Java Project

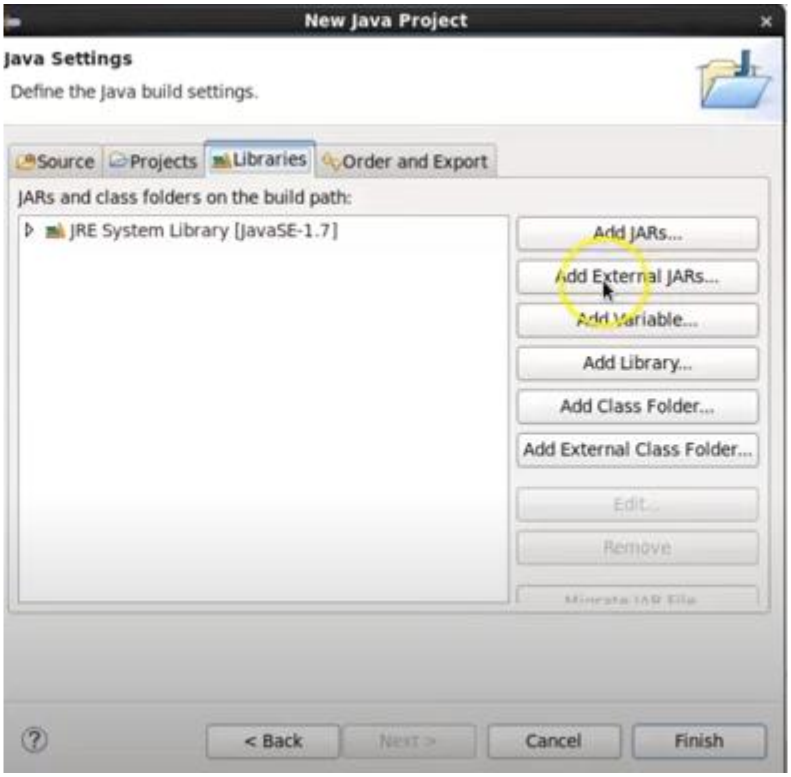
Java Settings

Define the Java build settings.

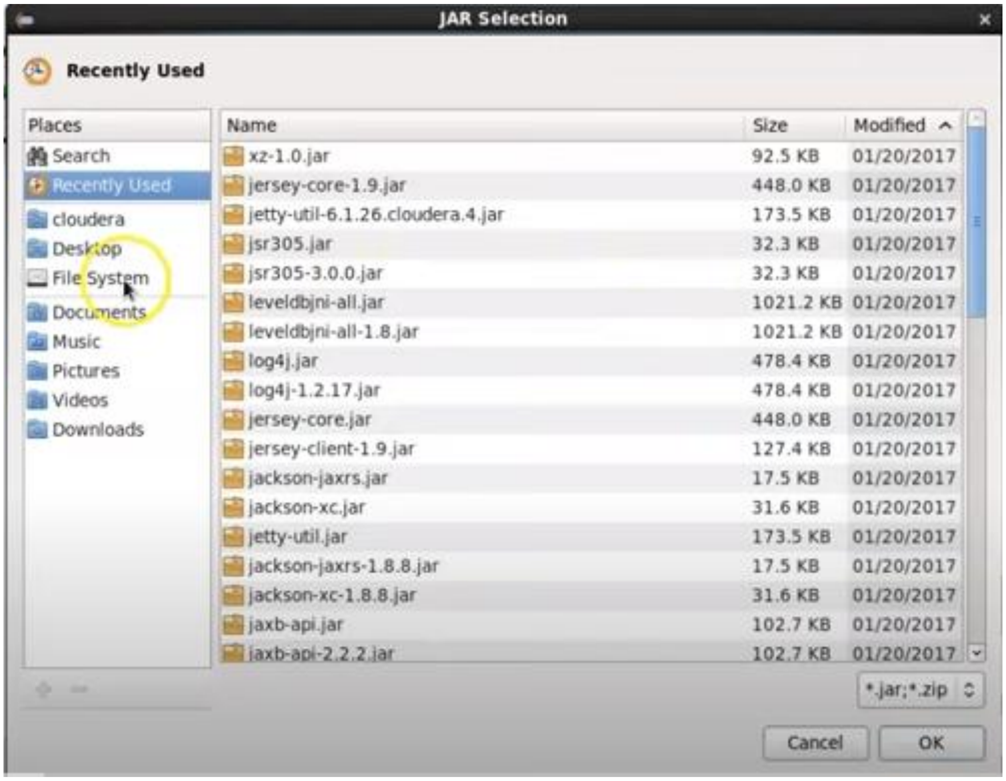
JARs and class folders on the build path:

▷ JRE System Library [JavaSE-1.7]

6. To import the libraries click on **Add External JARs**

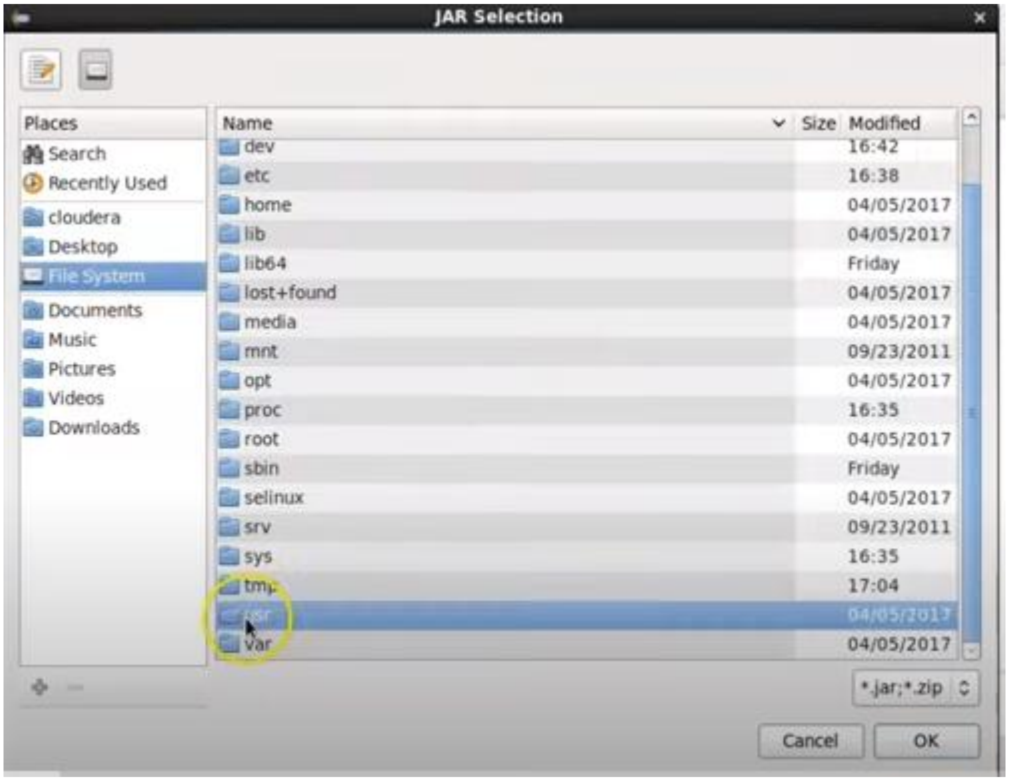


7. Select **File system**



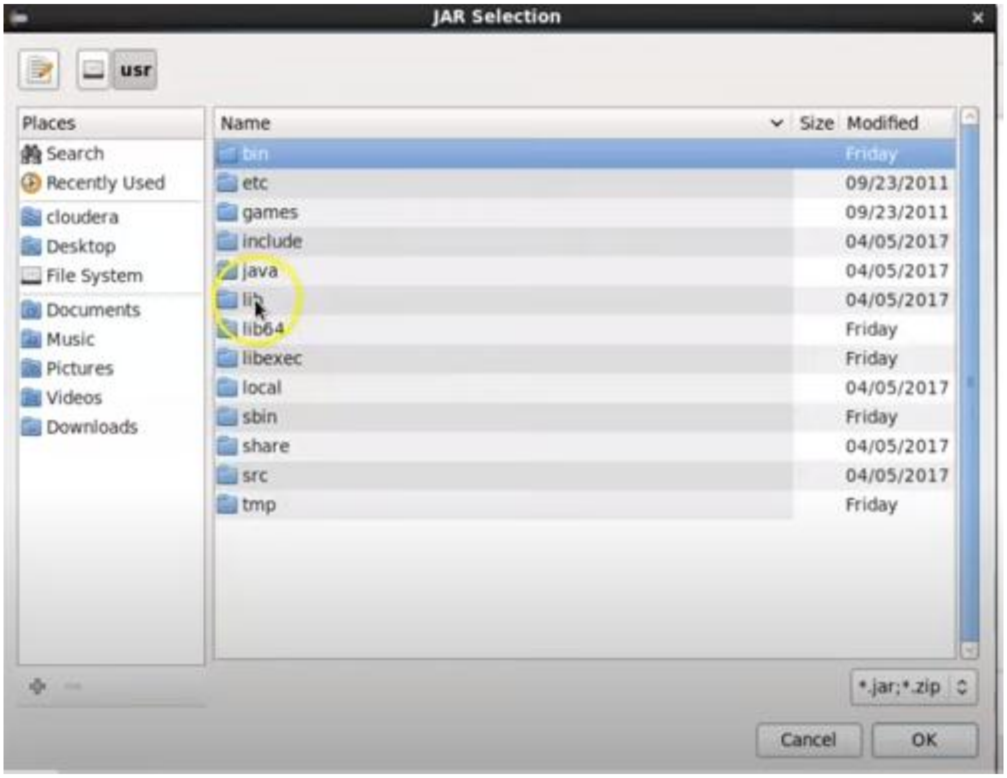
Name	Size	Modified
xz-1.0.jar	92.5 KB	01/20/2017
jersey-core-1.9.jar	448.0 KB	01/20/2017
jetty-util-6.1.26.cloudera.4.jar	173.5 KB	01/20/2017
jsr305.jar	32.3 KB	01/20/2017
jsr305-3.0.0.jar	32.3 KB	01/20/2017
leveldbjni-all.jar	1021.2 KB	01/20/2017
leveldbjni-all-1.8.jar	1021.2 KB	01/20/2017
log4j.jar	478.4 KB	01/20/2017
log4j-1.2.17.jar	478.4 KB	01/20/2017
jersey-core.jar	448.0 KB	01/20/2017
jersey-client-1.9.jar	127.4 KB	01/20/2017
jackson-jaxrs.jar	17.5 KB	01/20/2017
jackson-xc.jar	31.6 KB	01/20/2017
jetty-util.jar	173.5 KB	01/20/2017
jackson-jaxrs-1.8.8.jar	17.5 KB	01/20/2017
jackson-xc-1.8.8.jar	31.6 KB	01/20/2017
jaxb-api.jar	102.7 KB	01/20/2017
jaxb-api-2.2.2.jar	102.7 KB	01/20/2017

8. Select **usr**



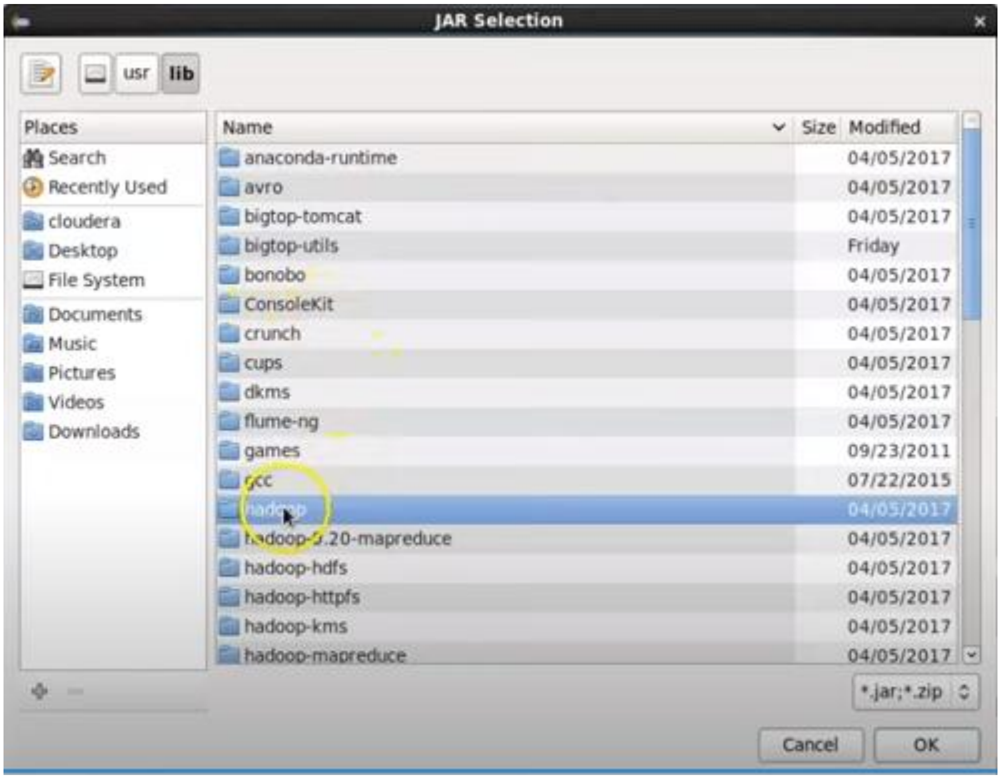
Name	Size	Modified
dev		16:42
etc		16:38
home		04/05/2017
lib		04/05/2017
lib64		Friday
lost+found		04/05/2017
media		04/05/2017
mnt		09/23/2011
opt		04/05/2017
proc		16:35
root		04/05/2017
sbin		Friday
selinux		04/05/2017
srv		09/23/2011
sys		16:35
tmp		17:04
usr		04/05/2017
var		04/05/2017

9. usr → Lib



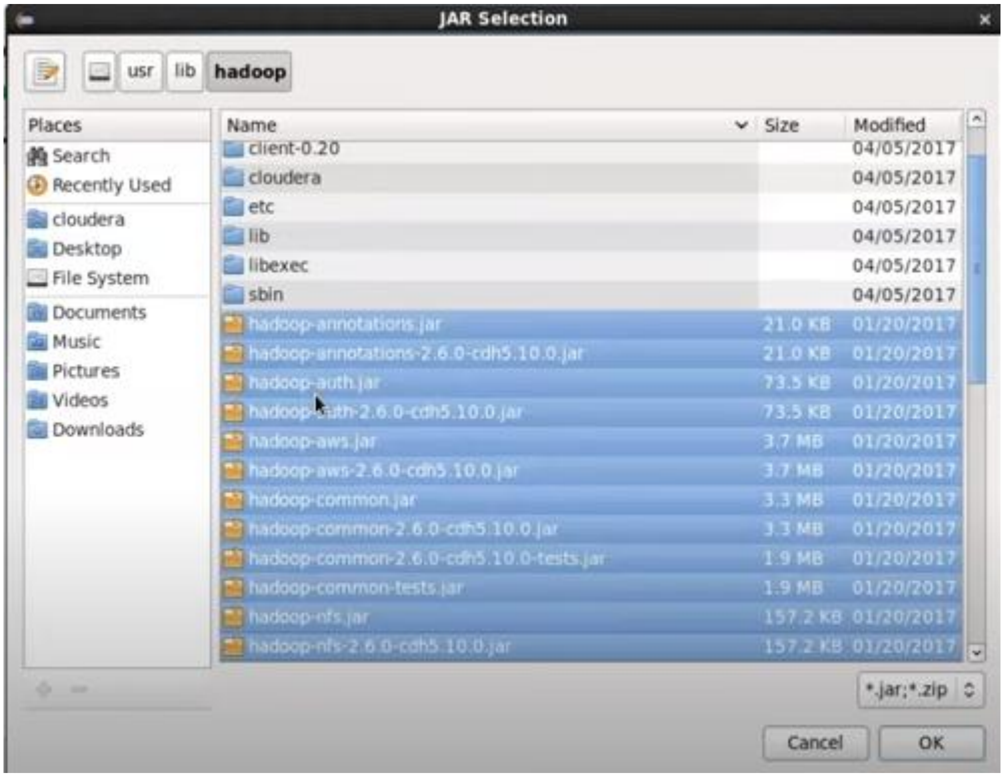
Name	Size	Modified
bin		Friday
etc		09/23/2011
games		09/23/2011
include		04/05/2017
java		04/05/2017
lib		04/05/2017
lib64		Friday
libexec		Friday
local		04/05/2017
sbin		Friday
share		04/05/2017
src		04/05/2017
tmp		Friday

10. Lib → Hadoop



Name	Size	Modified
anaconda-runtime		04/05/2017
avro		04/05/2017
bigtop-tomcat		04/05/2017
bigtop-utils		Friday
bonobo		04/05/2017
ConsoleKit		04/05/2017
crunch		04/05/2017
cups		04/05/2017
dkms		04/05/2017
flume-ng		04/05/2017
games		09/23/2011
gcc		07/22/2015
hadoop		04/05/2017
hadoop-0.20-mapreduce		04/05/2017
hadoop-hdfs		04/05/2017
hadoop-httpfs		04/05/2017
hadoop-kms		04/05/2017
hadoop-mapreduce		04/05/2017

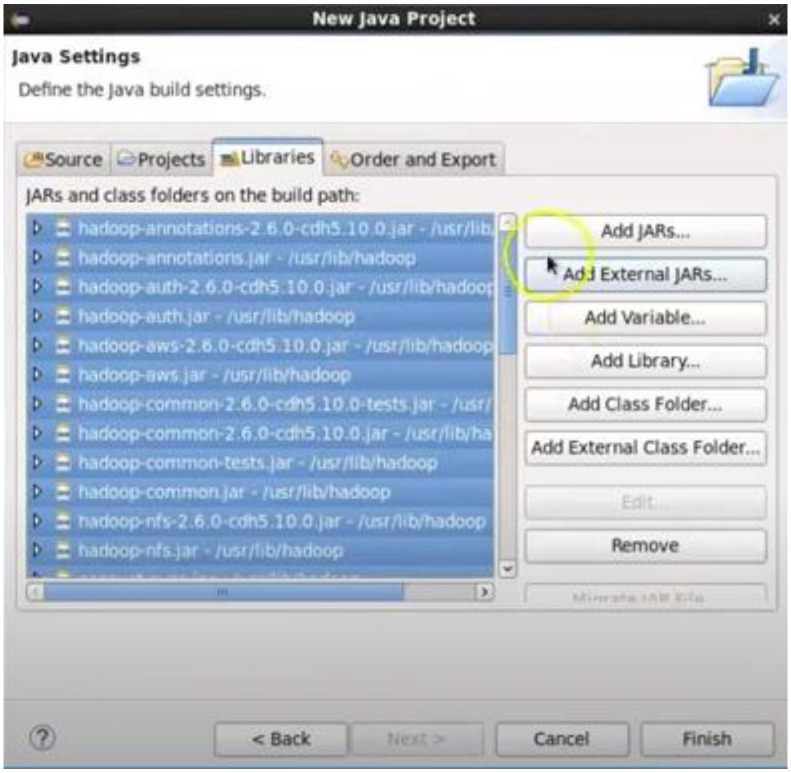
11. Select all the jar files



Name	Size	Modified
client-0.20		04/05/2017
cloudera		04/05/2017
etc		04/05/2017
lib		04/05/2017
libexec		04/05/2017
sbin		04/05/2017
hadoop-annotations.jar	21.0 KB	01/20/2017
hadoop-annotations-2.6.0-cdh5.10.0.jar	21.0 KB	01/20/2017
hadoop-auth.jar	73.5 KB	01/20/2017
hadoop-auth-2.6.0-cdh5.10.0.jar	73.5 KB	01/20/2017
hadoop-aws.jar	3.7 MB	01/20/2017
hadoop-aws-2.6.0-cdh5.10.0.jar	3.7 MB	01/20/2017
hadoop-common.jar	3.3 MB	01/20/2017
hadoop-common-2.6.0-cdh5.10.0.jar	3.3 MB	01/20/2017
hadoop-common-2.6.0-cdh5.10.0-tests.jar	1.9 MB	01/20/2017
hadoop-common-tests.jar	1.9 MB	01/20/2017
hadoop-nfs.jar	157.2 KB	01/20/2017
hadoop-nfs-2.6.0-cdh5.10.0.jar	157.2 KB	01/20/2017

12. Click on OK BUTTON

13. All the **jar files** would be added to the **libraries**



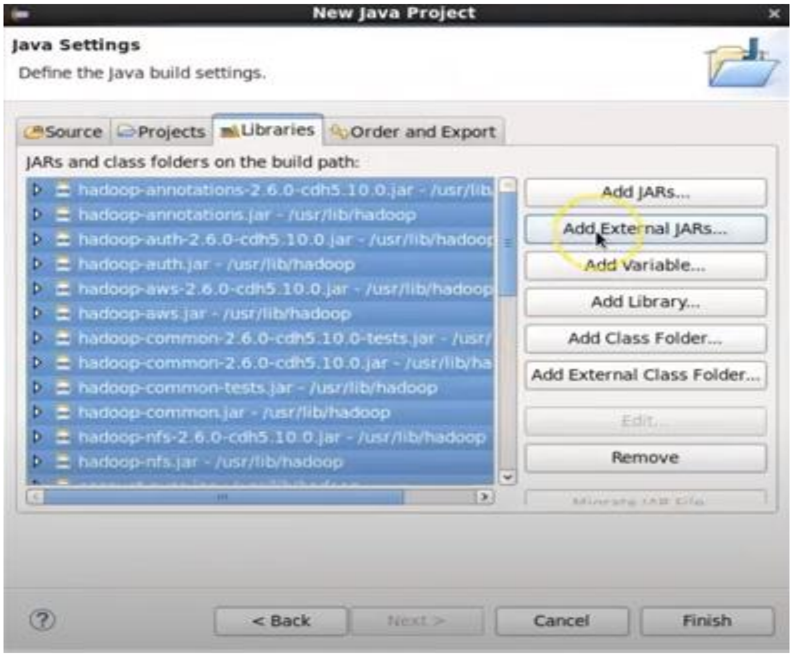
Java Settings

Define the Java build settings.

JARs and class folders on the build path:

```
hadoop-annotations-2.6.0-cdh5.10.0.jar - /usr/lib
hadoop-annotations.jar - /usr/lib/hadoop
hadoop-auth-2.6.0-cdh5.10.0.jar - /usr/lib/hadoop
hadoop-auth.jar - /usr/lib/hadoop
hadoop-aws-2.6.0-cdh5.10.0.jar - /usr/lib/hadoop
hadoop-aws.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.10.0-tests.jar - /usr/
hadoop-common-2.6.0-cdh5.10.0.jar - /usr/lib/ha
hadoop-common-tests.jar - /usr/lib/hadoop
hadoop-common.jar - /usr/lib/hadoop
hadoop-nfs-2.6.0-cdh5.10.0.jar - /usr/lib/hadoop
hadoop-nfs.jar - /usr/lib/hadoop
```

14. Click again **Add External Jars**



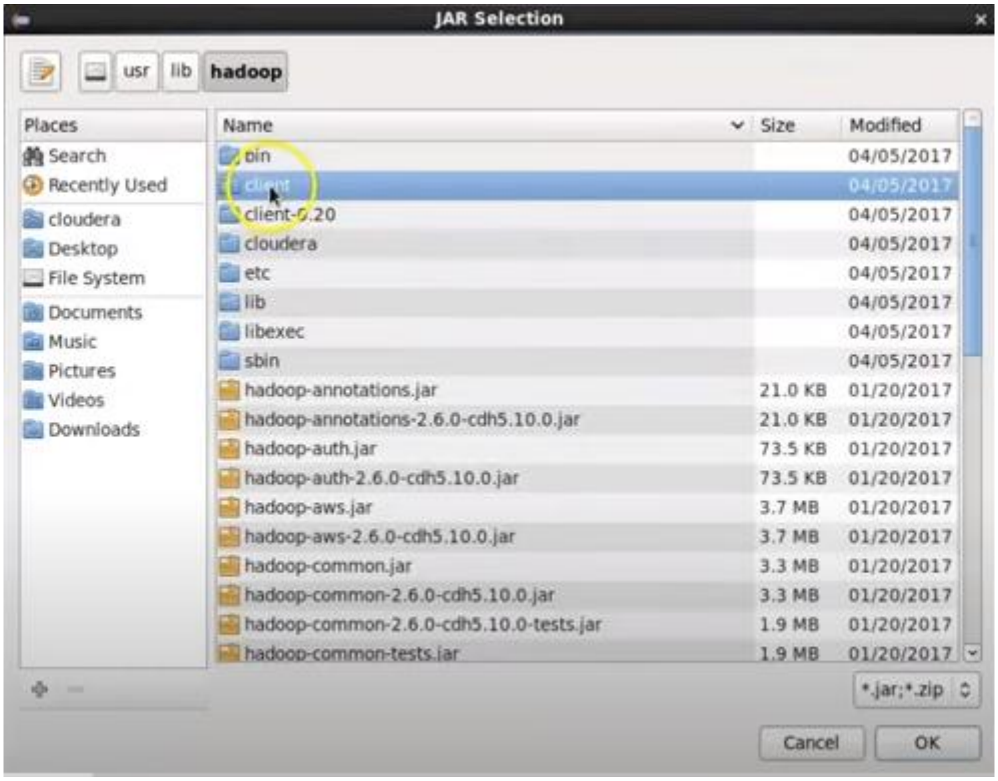
Java Settings

Define the Java build settings.

JARs and class folders on the build path:

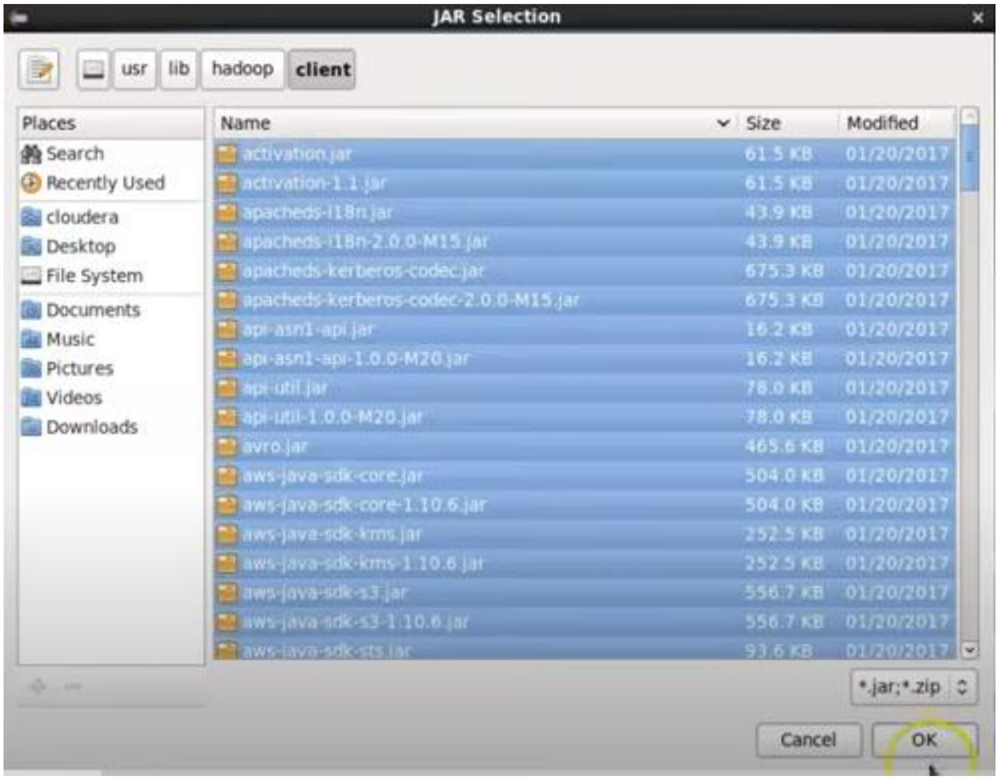
```
hadoop-annotations-2.6.0-cdh5.10.0.jar - /usr/lib
hadoop-annotations.jar - /usr/lib/hadoop
hadoop-auth-2.6.0-cdh5.10.0.jar - /usr/lib/hadoop
hadoop-auth.jar - /usr/lib/hadoop
hadoop-aws-2.6.0-cdh5.10.0.jar - /usr/lib/hadoop
hadoop-aws.jar - /usr/lib/hadoop
hadoop-common-2.6.0-cdh5.10.0-tests.jar - /usr/
hadoop-common-2.6.0-cdh5.10.0.jar - /usr/lib/ha
hadoop-common-tests.jar - /usr/lib/hadoop
hadoop-common.jar - /usr/lib/hadoop
hadoop-nfs-2.6.0-cdh5.10.0.jar - /usr/lib/hadoop
hadoop-nfs.jar - /usr/lib/hadoop
```

15. Click **client**



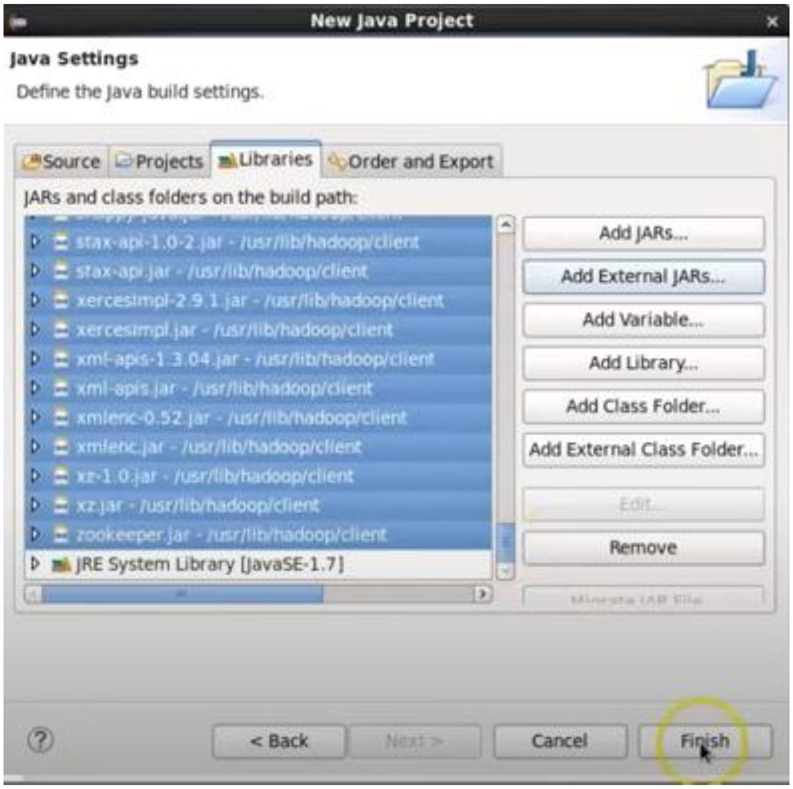
Name	Size	Modified
bin		04/05/2017
client		04/05/2017
client-0.20		04/05/2017
cloudera		04/05/2017
etc		04/05/2017
lib		04/05/2017
libexec		04/05/2017
sbin		04/05/2017
hadoop-annotations.jar	21.0 KB	01/20/2017
hadoop-annotations-2.6.0-cdh5.10.0.jar	21.0 KB	01/20/2017
hadoop-auth.jar	73.5 KB	01/20/2017
hadoop-auth-2.6.0-cdh5.10.0.jar	73.5 KB	01/20/2017
hadoop-aws.jar	3.7 MB	01/20/2017
hadoop-aws-2.6.0-cdh5.10.0.jar	3.7 MB	01/20/2017
hadoop-common.jar	3.3 MB	01/20/2017
hadoop-common-2.6.0-cdh5.10.0.jar	3.3 MB	01/20/2017
hadoop-common-2.6.0-cdh5.10.0-tests.jar	1.9 MB	01/20/2017
hadoop-common-tests.jar	1.9 MB	01/20/2017

16. Select all the **jar files** and click on **OK**



Name	Size	Modified
activation.jar	61.5 KB	01/20/2017
activation-1.1.jar	61.5 KB	01/20/2017
apacheds-i18n.jar	43.9 KB	01/20/2017
apacheds-i18n-2.0.0-M15.jar	43.9 KB	01/20/2017
apacheds-kerberos-codec.jar	675.3 KB	01/20/2017
apacheds-kerberos-codec-2.0.0-M15.jar	675.3 KB	01/20/2017
api-asn1-api.jar	16.2 KB	01/20/2017
api-asn1-api-1.0.0-M20.jar	16.2 KB	01/20/2017
api-util.jar	78.0 KB	01/20/2017
api-util-1.0.0-M20.jar	78.0 KB	01/20/2017
avro.jar	465.6 KB	01/20/2017
aws-java-sdk-core.jar	504.0 KB	01/20/2017
aws-java-sdk-core-1.10.6.jar	504.0 KB	01/20/2017
aws-java-sdk-kms.jar	252.5 KB	01/20/2017
aws-java-sdk-kms-1.10.6.jar	252.5 KB	01/20/2017
aws-java-sdk-s3.jar	556.7 KB	01/20/2017
aws-java-sdk-s3-1.10.6.jar	556.7 KB	01/20/2017
aws-java-sdk-sts.jar	93.6 KB	01/20/2017

17. **Once all the JAR files are added to the libraries click on Finish**



New Java Project

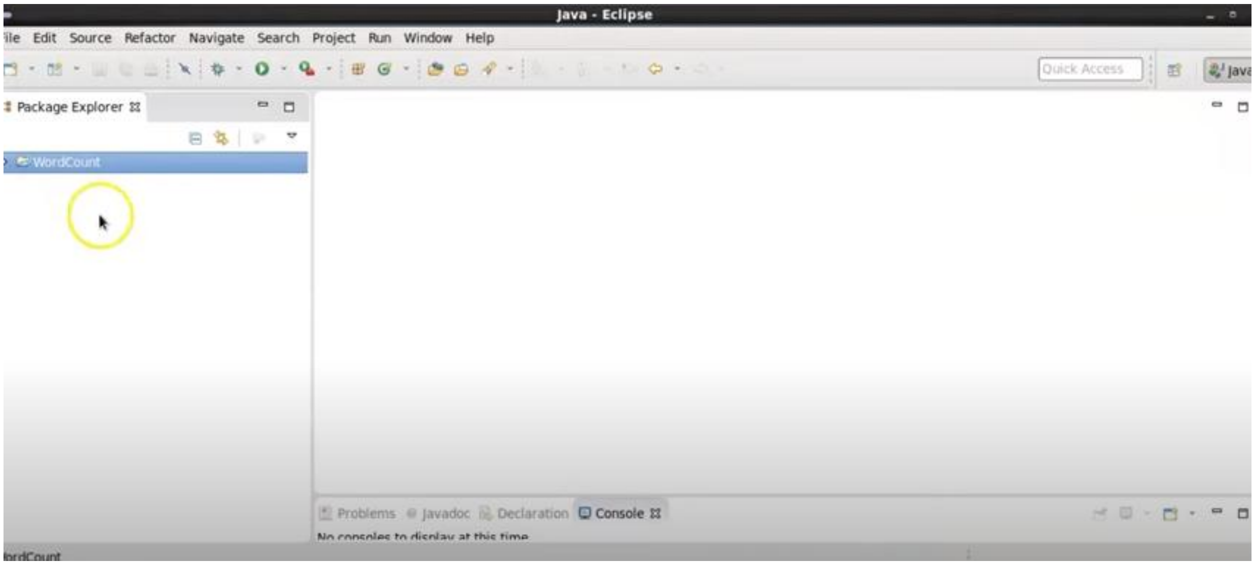
Java Settings

Define the Java build settings.

JARs and class folders on the build path:

- stax-api-1.0-2.jar - /usr/lib/hadoop/client
- stax-api.jar - /usr/lib/hadoop/client
- xercesImpl-2.9.1.jar - /usr/lib/hadoop/client
- xercesImpl.jar - /usr/lib/hadoop/client
- xml-apis-1.3.04.jar - /usr/lib/hadoop/client
- xml-apis.jar - /usr/lib/hadoop/client
- xmlenc-0.52.jar - /usr/lib/hadoop/client
- xmlenc.jar - /usr/lib/hadoop/client
- xz-1.0.jar - /usr/lib/hadoop/client
- xz.jar - /usr/lib/hadoop/client
- zookeeper.jar - /usr/lib/hadoop/client
- JRE System Library [JavaSE-1.7]

18. **You can see file WordCount here**



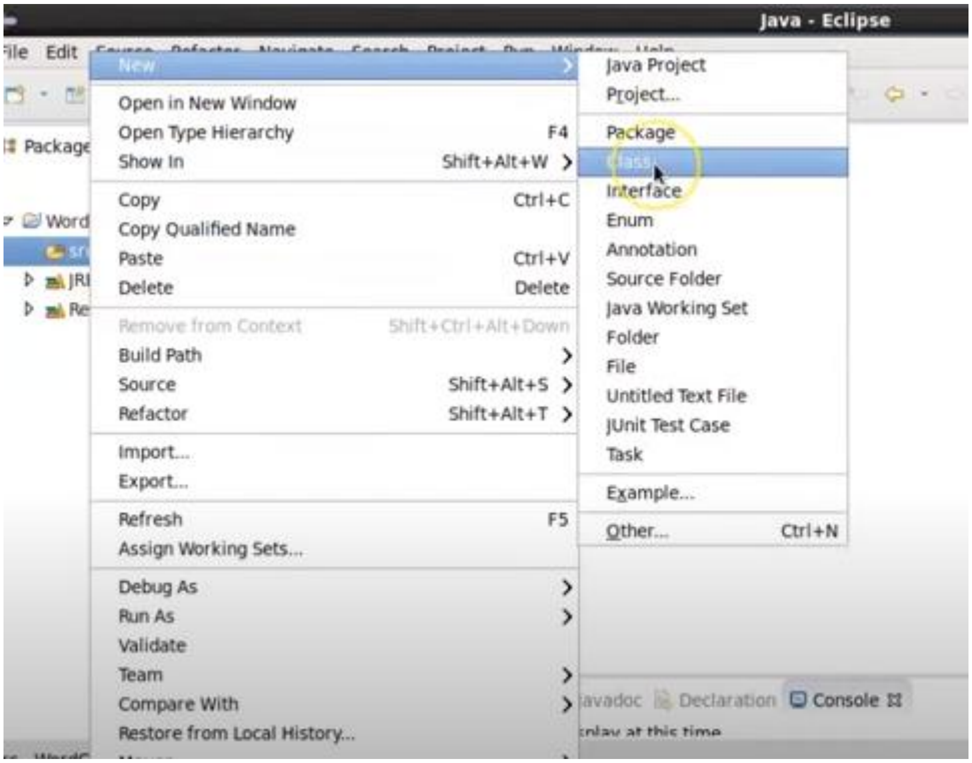
Java - Eclipse

Package Explorer

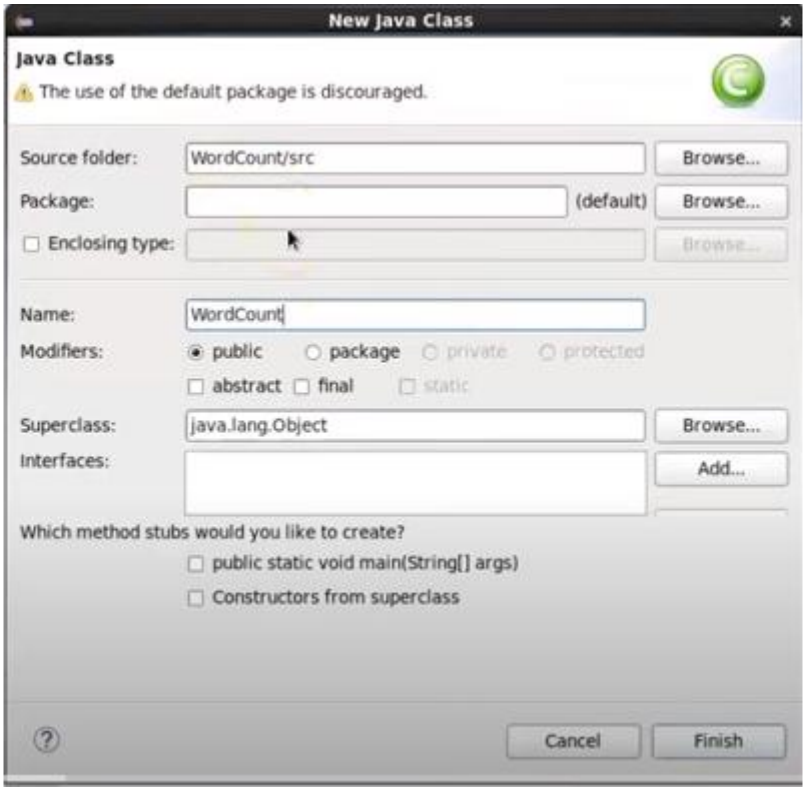
WordCount

No consoles to display at this time.

19. Click WordCount → src → new → class



20. In the name field type **WordCount** and click **Finish**



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

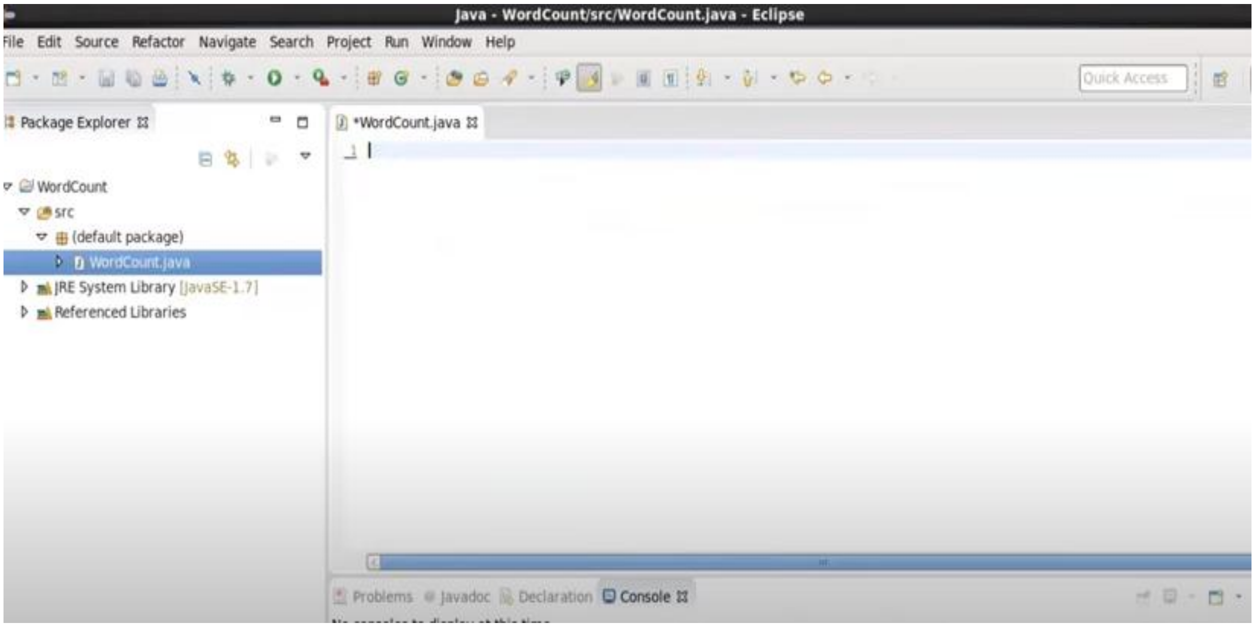
Interfaces:

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

21. **Enter the java program in the File WordCount.java**

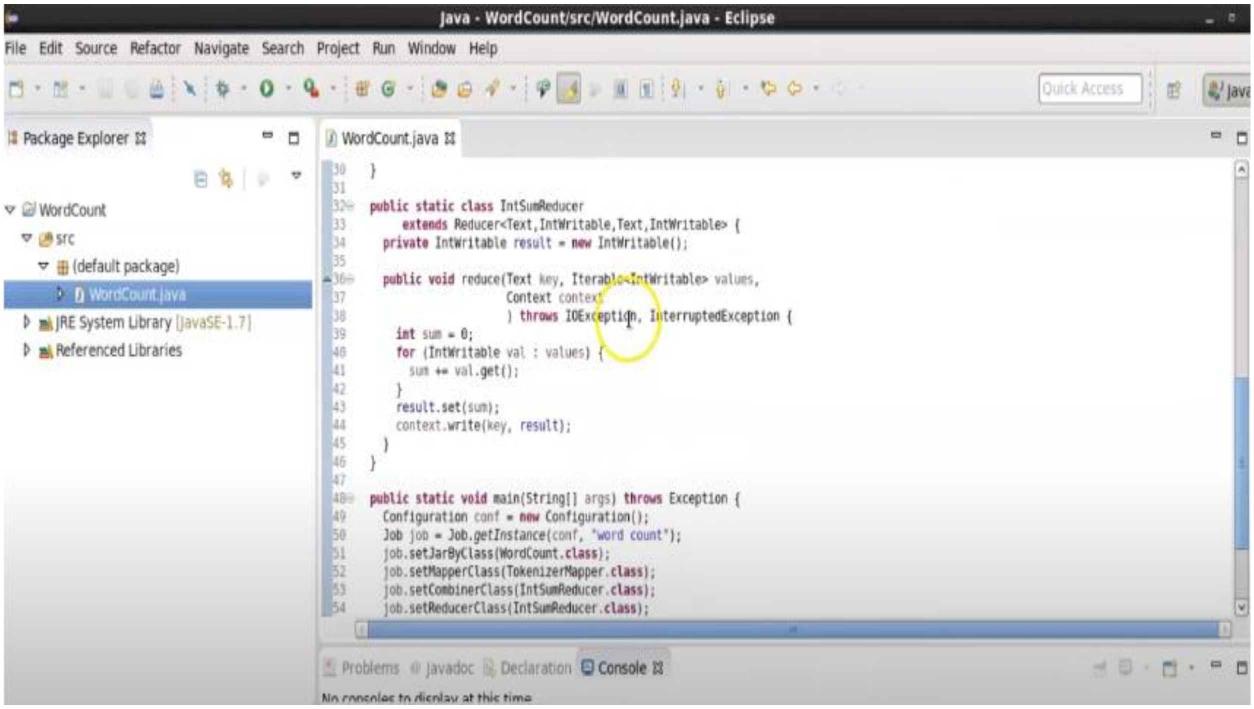

Java - WordCount/src/WordCount.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
WordCount
src
(default package)
WordCount.java
JRE System Library [JavaSE-1.7]
Referenced Libraries
*WordCount.java
Problems
Javadoc
Declaration
Console

No consoles to display at this time.

22. **Source code for mapreduce client**

https://hadoop.apache.org/docs/r3.2.1/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

23. **paste the code in eclipse console and save (Ctrl+S)**



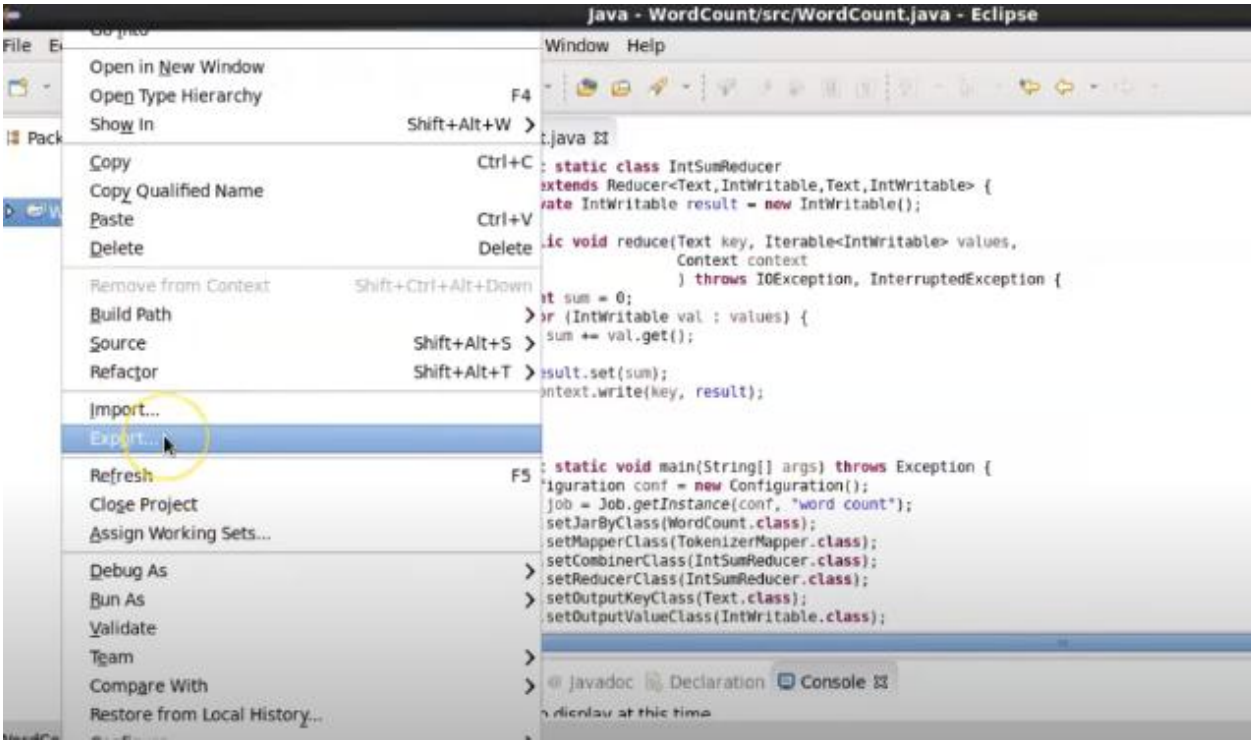
```java
  }

  public static class IntSumReducer
       extends Reducer<Text,IntWritable,Text,IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                       ) throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }

  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
```

No consoles to display at this time

24. In the console window check any errors (Errors will be indicated with red mark)

25. Next step export the JAR files.

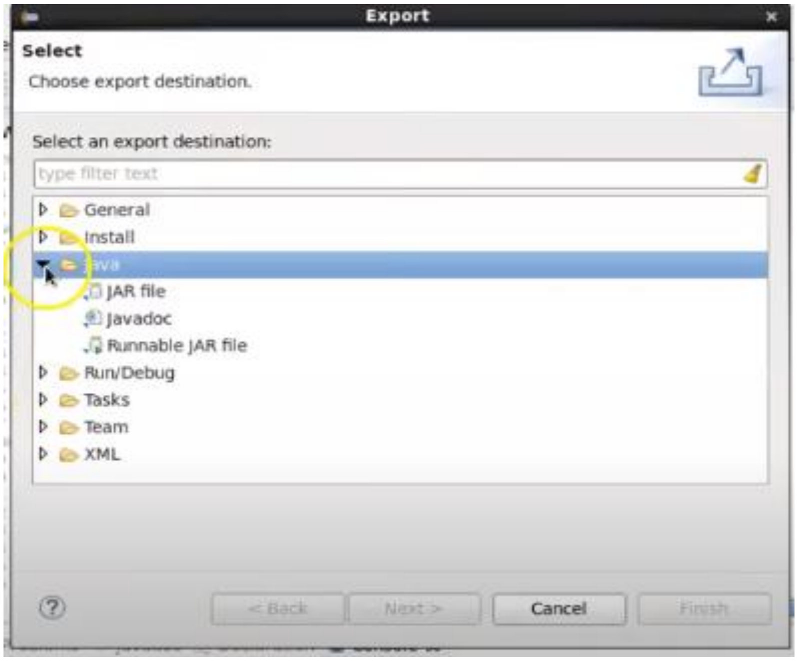
Rightclick on wordcount ⟶ Export



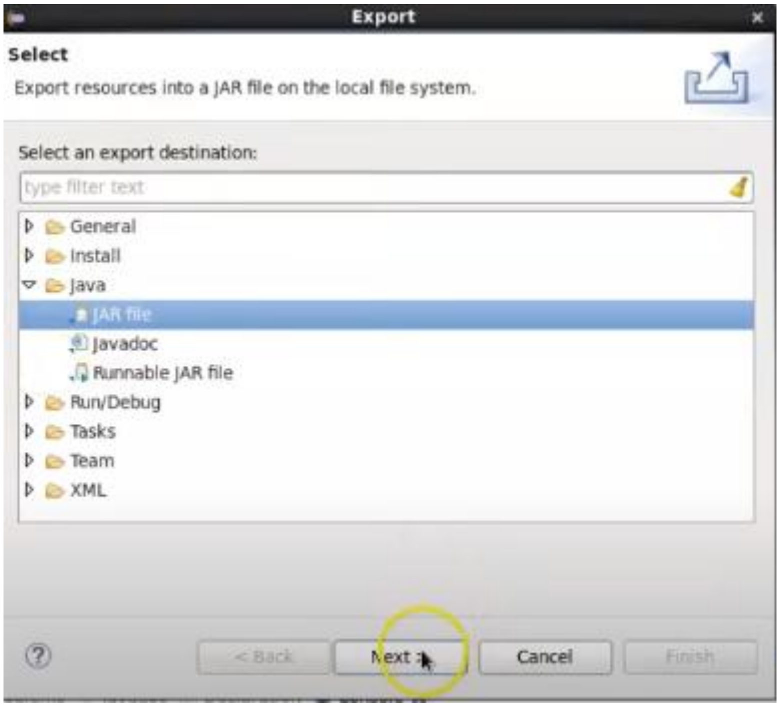
```
: static class IntSumReducer
xtends Reducer<Text,IntWritable,Text,IntWritable> {
vate IntWritable result = new IntWritable();

ic void reduce(Text key, Iterable<IntWritable> values,
                Context context
                ) throws IOException, InterruptedException {
nt sum = 0;
or (IntWritable val : values) {
 sum += val.get();

esult.set(sum);
ontext.write(key, result);


: static void main(String[] args) throws Exception {
iguration conf = new Configuration();
 job = Job.getInstance(conf, "word count");
 setJarByClass(WordCount.class);
 setMapperClass(TokenizerMapper.class);
 setCombinerClass(IntSumReducer.class);
 setReducerClass(IntSumReducer.class);
 setOutputKeyClass(Text.class);
 setOutputValueClass(IntWritable.class);
```

```
n display at this time
```

26. Select java



27. Select JAR file and click next

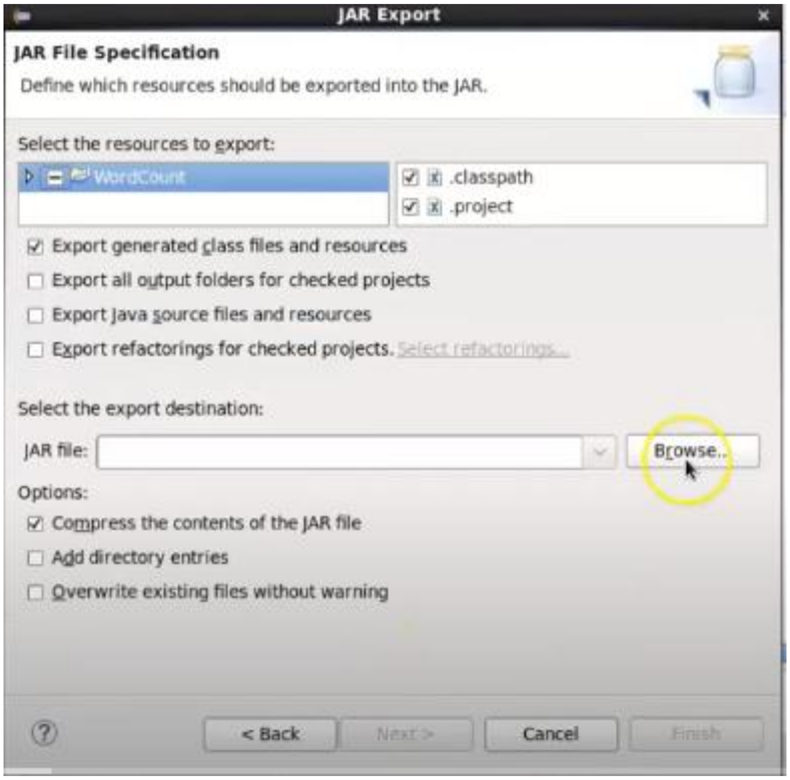


Select

Export resources into a JAR file on the local file system.

Select an export destination:

28. Browse the JAR File



JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

☑ .classpath

☑ .project

☑ Export generated class files and resources

□ Export all output folders for checked projects

□ Export Java source files and resources

□ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file:

Options:

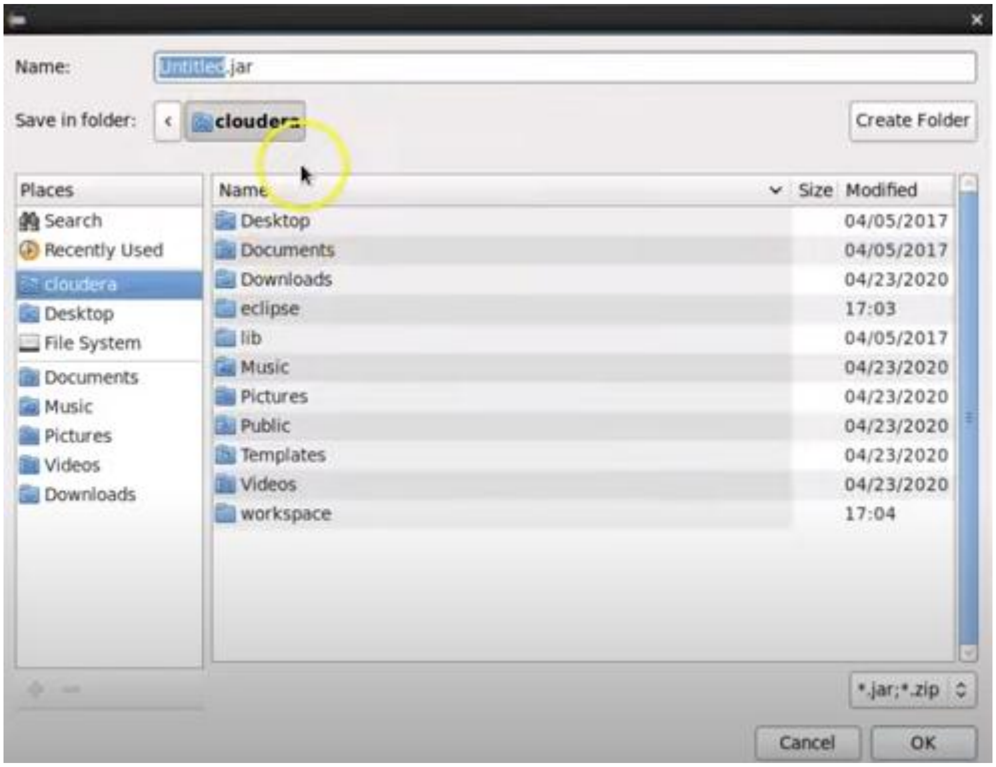
☑ Compress the contents of the JAR file

□ Add directory entries

□ Overwrite existing files without warning

29. **Select Cloudera**

You can find untitled.jar at the top

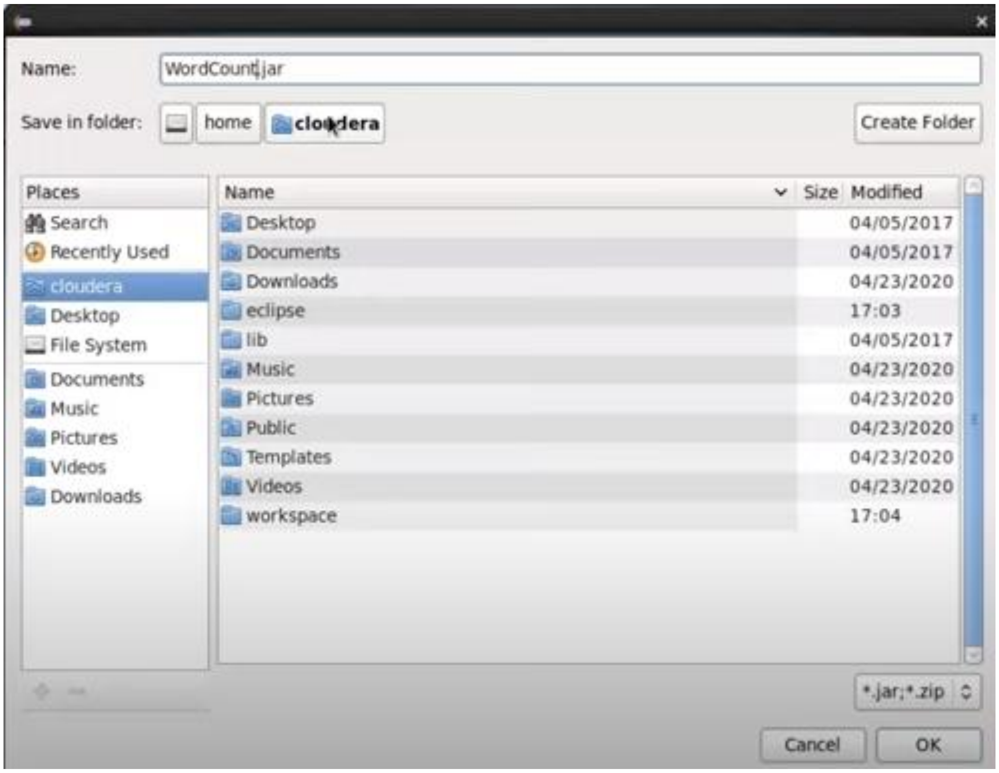


Name: Untitled.jar

Save in folder: cloudera

Name	Size	Modified
Desktop		04/05/2017
Documents		04/05/2017
Downloads		04/23/2020
eclipse		17:03
lib		04/05/2017
Music		04/23/2020
Pictures		04/23/2020
Public		04/23/2020
Templates		04/23/2020
Videos		04/23/2020
workspace		17:04

30. **Rename it as WordCount.**

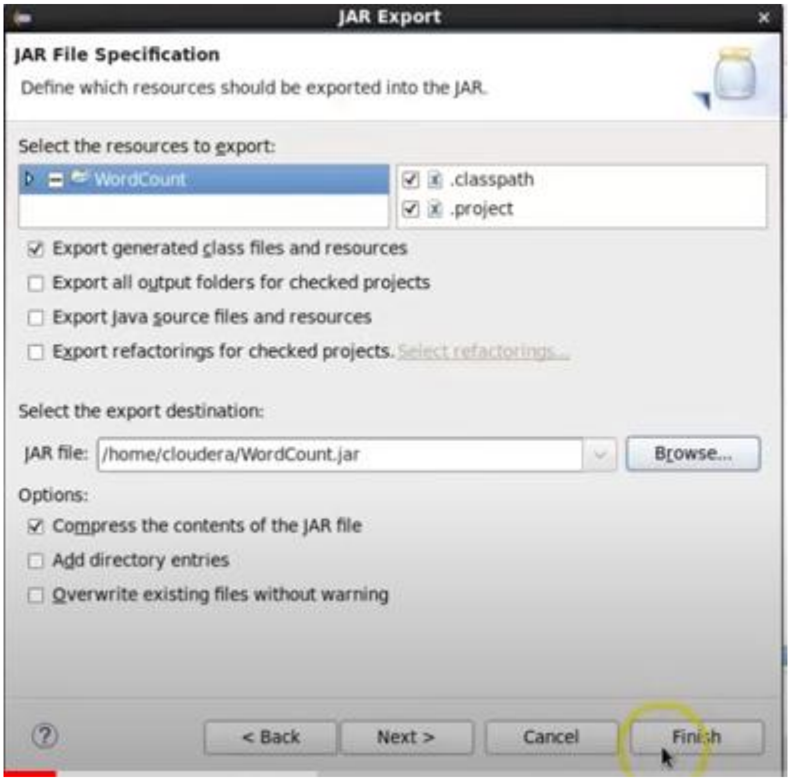


Name: WordCount.jar

Save in folder: home cloudera

Name	Size	Modified
Desktop		04/05/2017
Documents		04/05/2017
Downloads		04/23/2020
eclipse		17:03
lib		04/05/2017
Music		04/23/2020
Pictures		04/23/2020
Public		04/23/2020
Templates		04/23/2020
Videos		04/23/2020
workspace		17:04

31. Click ok and then click finish



JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- WordCount
- ☑ .classpath
- ☑ .project

☑ Export generated class files and resources

☐ Export all output folders for checked projects

☐ Export Java source files and resources

☐ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: /home/cloudera/WordCount.jar

Options:

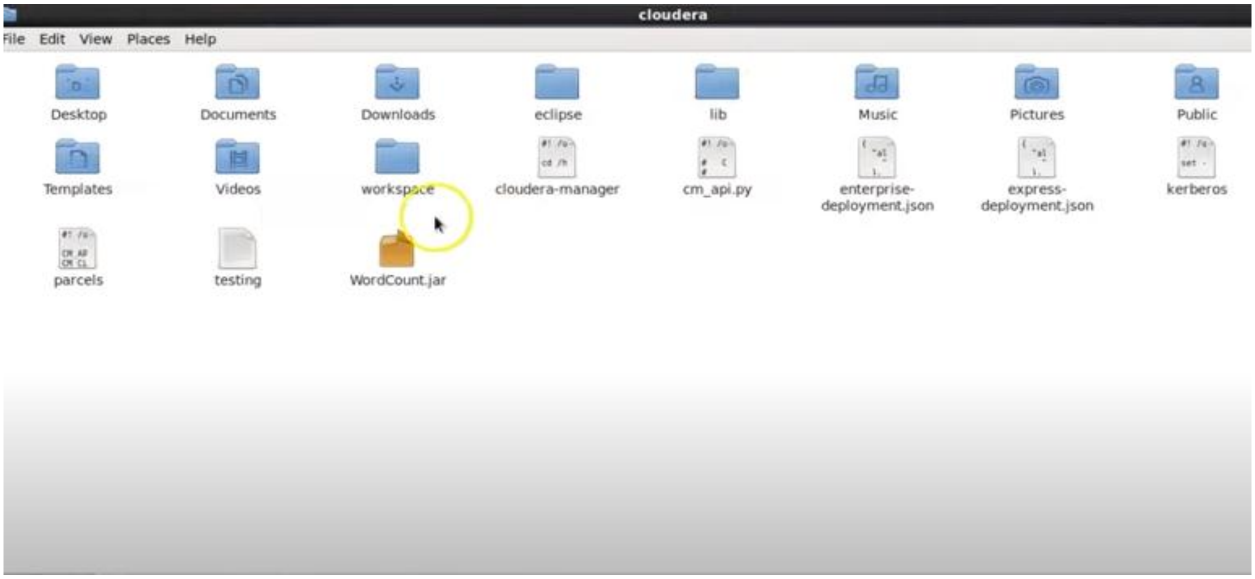
☑ Compress the contents of the JAR file

☐ Add directory entries

☐ Overwrite existing files without warning

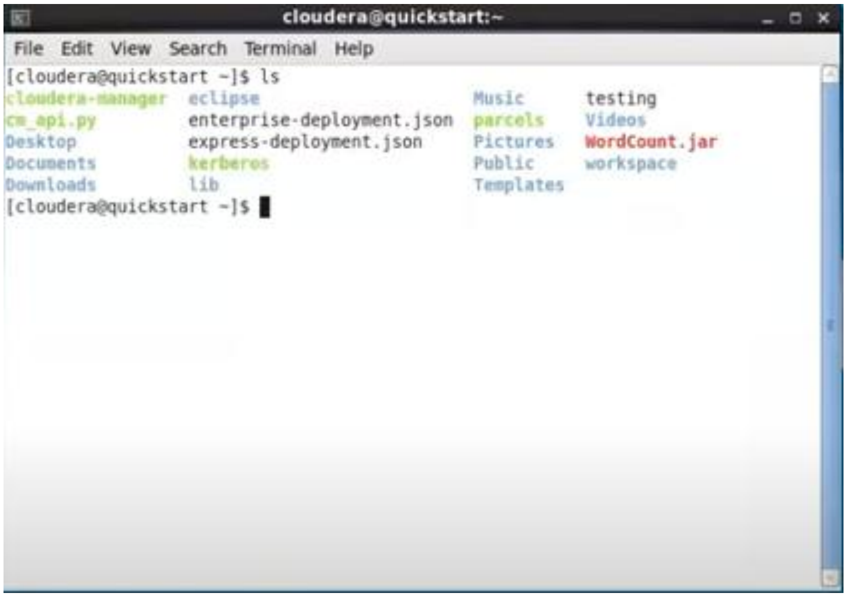
32. we will check in the filesystem whether WordCount.jar is exported successfully or not

In desktop click Computer → File system → home → cloudera → wordcount.jar



33. Once the jar file is verified open terminal

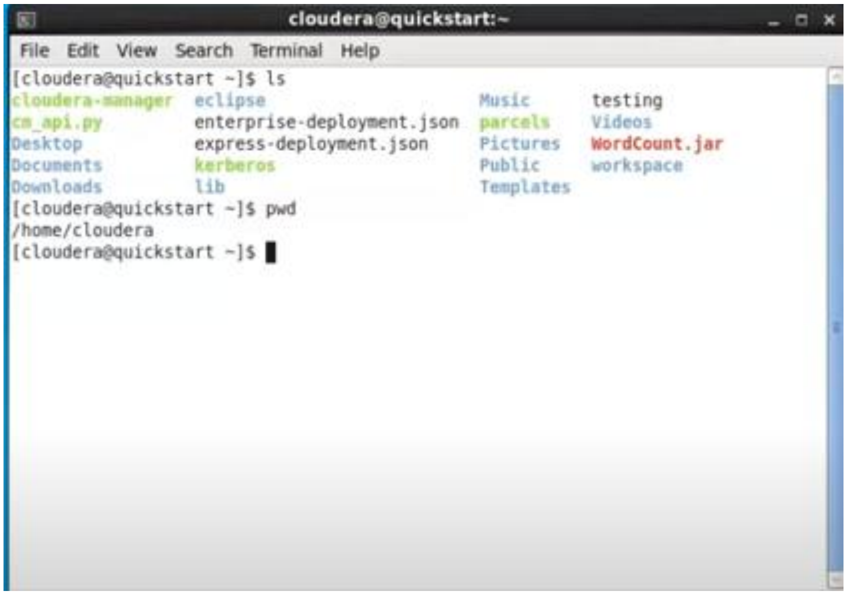
Type ls



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

34. Check the working directory

Type pwd



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

35. Create a file

```
cat > / home/cloudera/Processfile1.txt
```

“

-

Enter the words

-

Press Ctrl+Z to stop entering

36. To verify the contents present in Processfile1.txt

```
cat  /home/cloudera/Processfile1.txt
```

37. Move the file Processfile1.txt to hdfs

38. To check whether hdfs is working

```
Hdfs dfs –ls
```

39. To see the list of directories in hdfs

```
Hdfs dfs –ls  /
```

40. Create a directory

```
hdfs dfs –mkdir /inputfolder1
```

41. move the file (Processfile1.txt) into hadoop system

```
hdfs dfs –put /home/cloudera/Processfile1.txt /inputfolder1/
```

42. check whether the file is copied into the hdfs and to display the contents of the file

```
hdfs dfs –cat /inputfolder1/Processfile1.txt
```

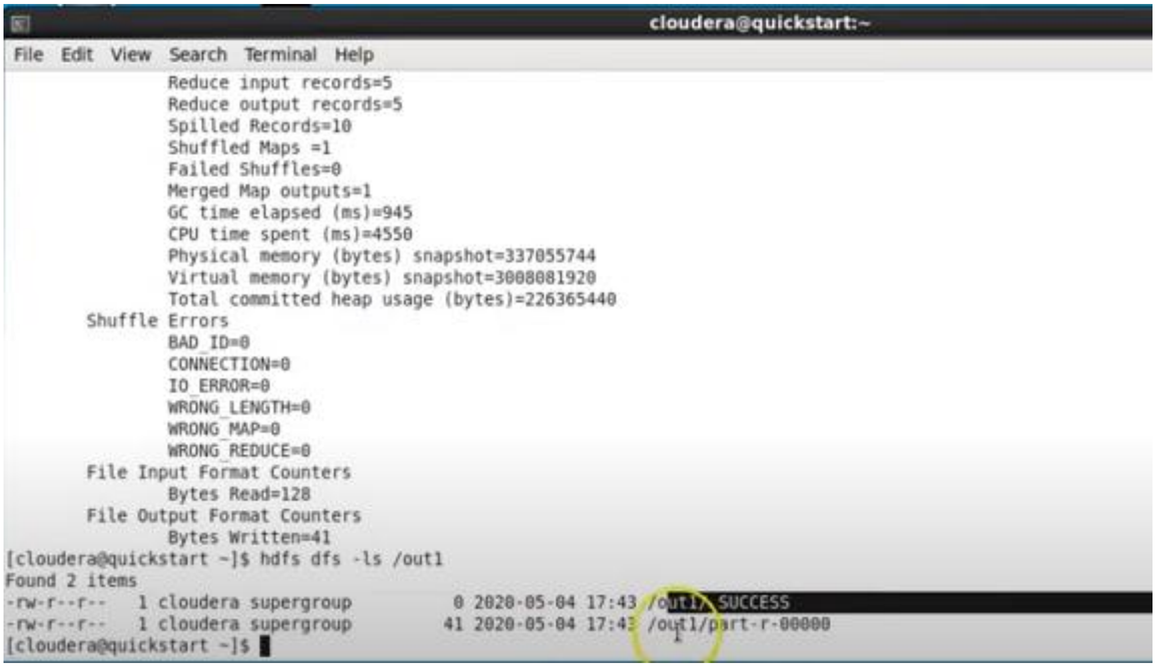
43. Open new terminal

```
Hadoop jar /home/cloudera/WordCount.jar WordCount /inputfolder1/Processfile1.txt  /out1
```

In the output u can see the no of splits, map tasks, reduce tasks, map output records

44. To see the output file of the mapreduce in out1 directory

hdfs dfs –ls /out1



```
                Reduce input records=5
                Reduce output records=5
                Spilled Records=10
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=945
                CPU time spent (ms)=4550
                Physical memory (bytes) snapshot=337055744
                Virtual memory (bytes) snapshot=3008081920
                Total committed heap usage (bytes)=226365440
        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=128
        File Output Format Counters
                Bytes Written=41
[cloudera@quickstart ~]$ hdfs dfs -ls /out1
Found 2 items
-rw-r--r--   1 cloudera supergroup          0 2020-05-04 17:43 /out1/_SUCCESS
-rw-r--r--   1 cloudera supergroup         41 2020-05-04 17:43 /out1/part-r-00000
[cloudera@quickstart ~]$
```

45. To see the output

hdfs dfs –cat /out1/part-r-00000