

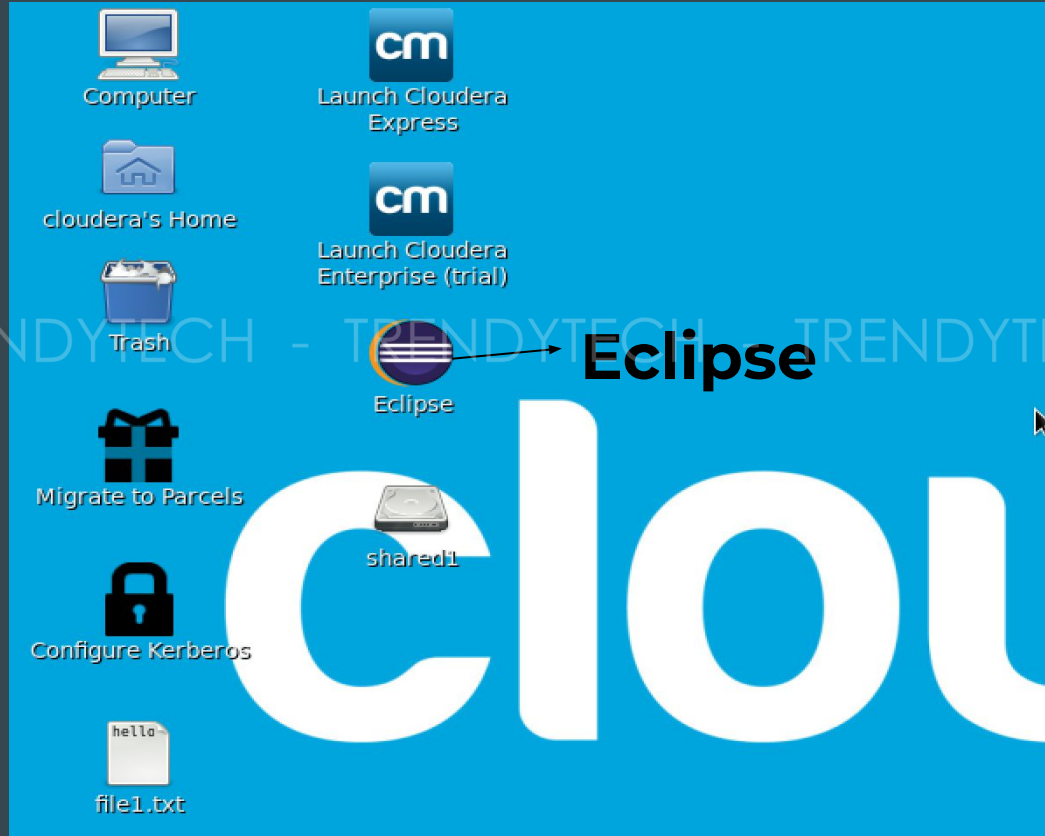


Running Mapreduce Program

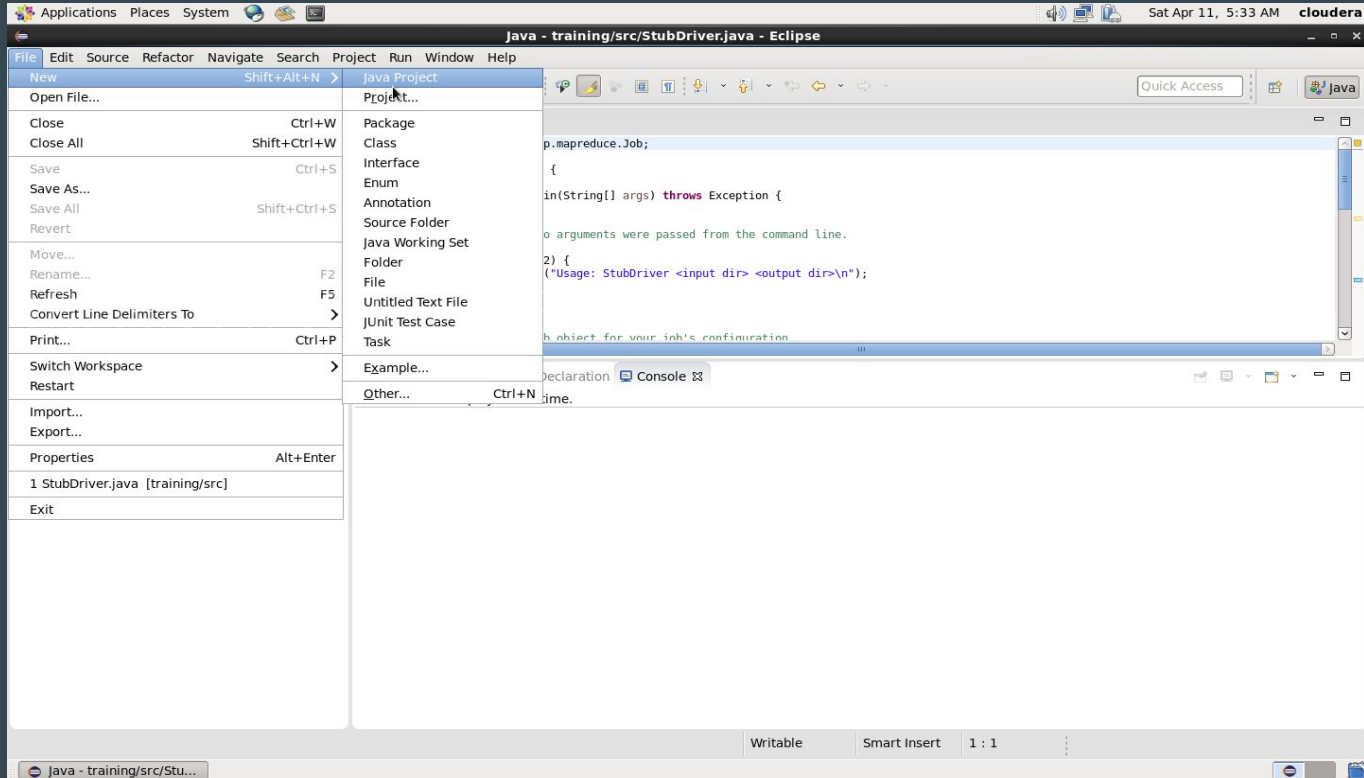
...

MR Practical

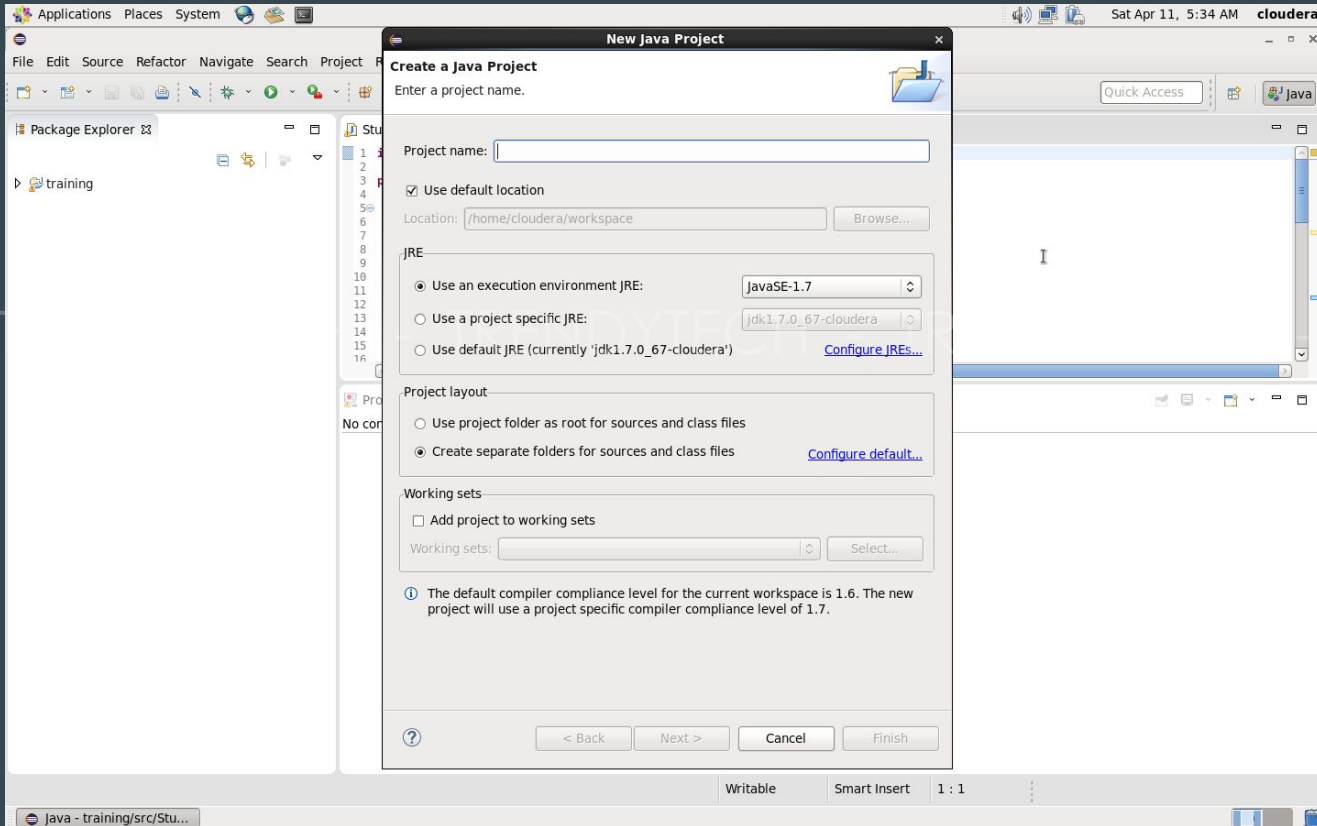
Eclipse is Present on Cloudera Desktop



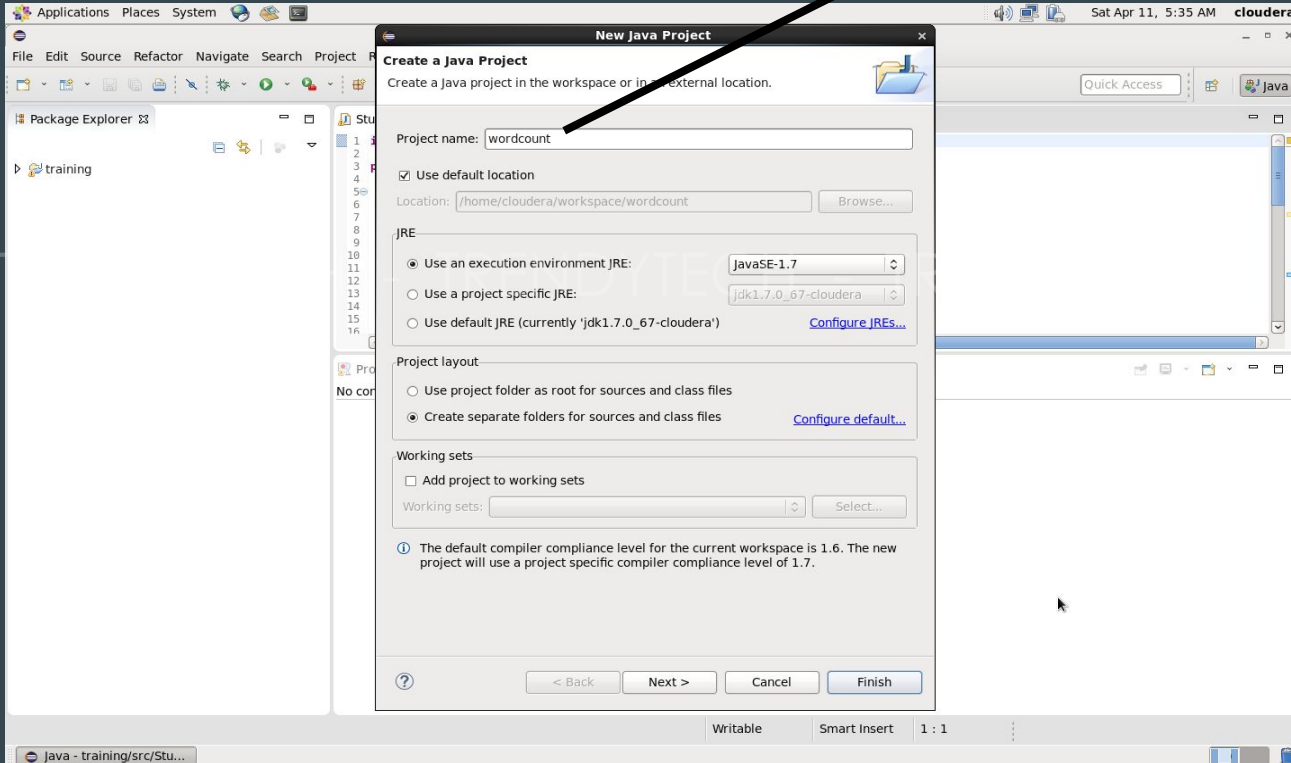
Create a new Java Project



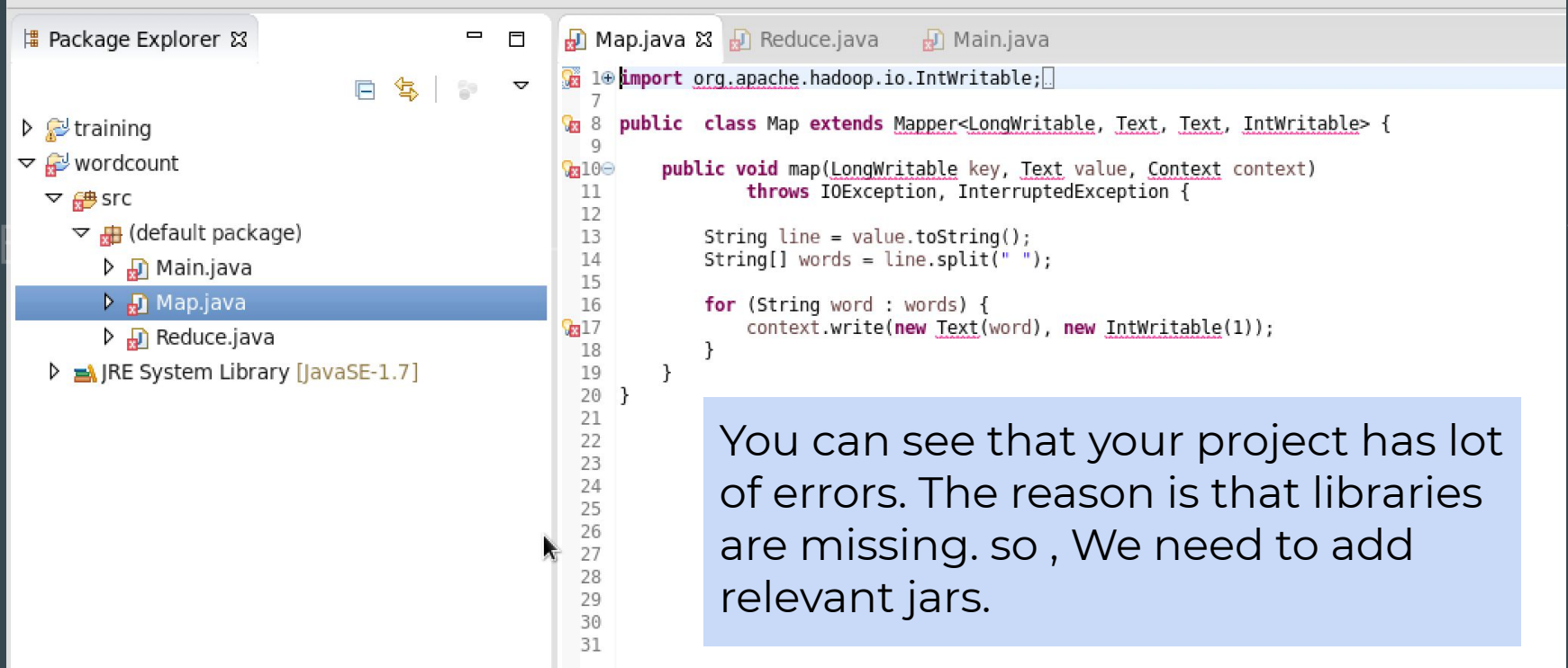
Give your project name



I gave wordcount as project name



Copy all the 3 files in src folder

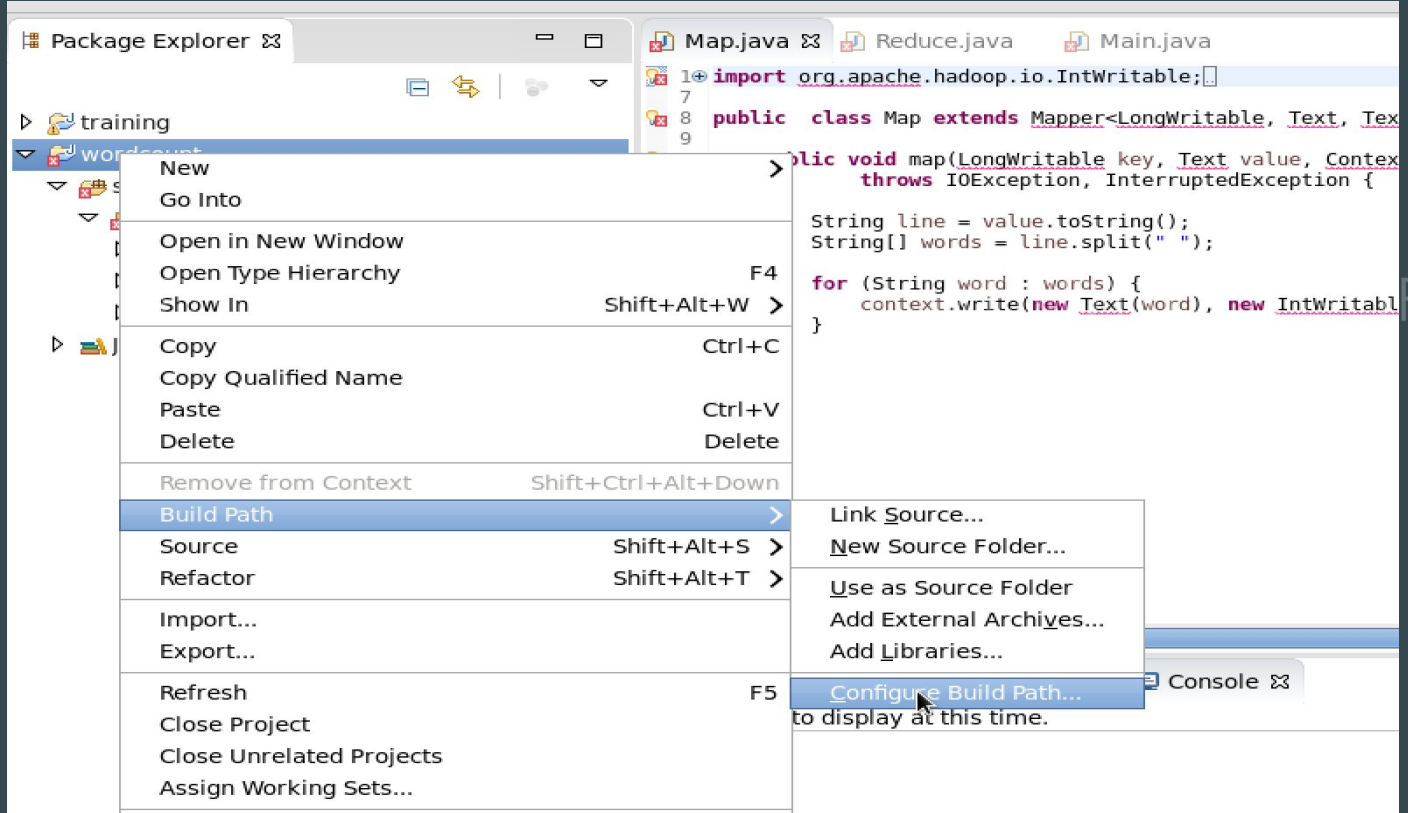


The screenshot shows an IDE with the Package Explorer on the left and the Editor on the right. The Package Explorer shows a project named 'training' with a sub-project 'wordcount'. Inside 'wordcount', there is a 'src' folder containing a '(default package)' with three files: 'Main.java', 'Map.java', and 'Reduce.java'. The 'Map.java' file is selected. The Editor shows the code for 'Map.java', which includes an import statement for 'org.apache.hadoop.io.IntWritable' and a class definition 'Map' extending 'Mapper<LongWritable, Text, Text, IntWritable>'. The code defines a 'map' method that takes a 'LongWritable' key, a 'Text' value, and a 'Context' context, and throws 'IOException' and 'InterruptedException'. The method body splits the input line into words and writes each word to the context as a 'Text' object with a value of 1. The code is as follows:

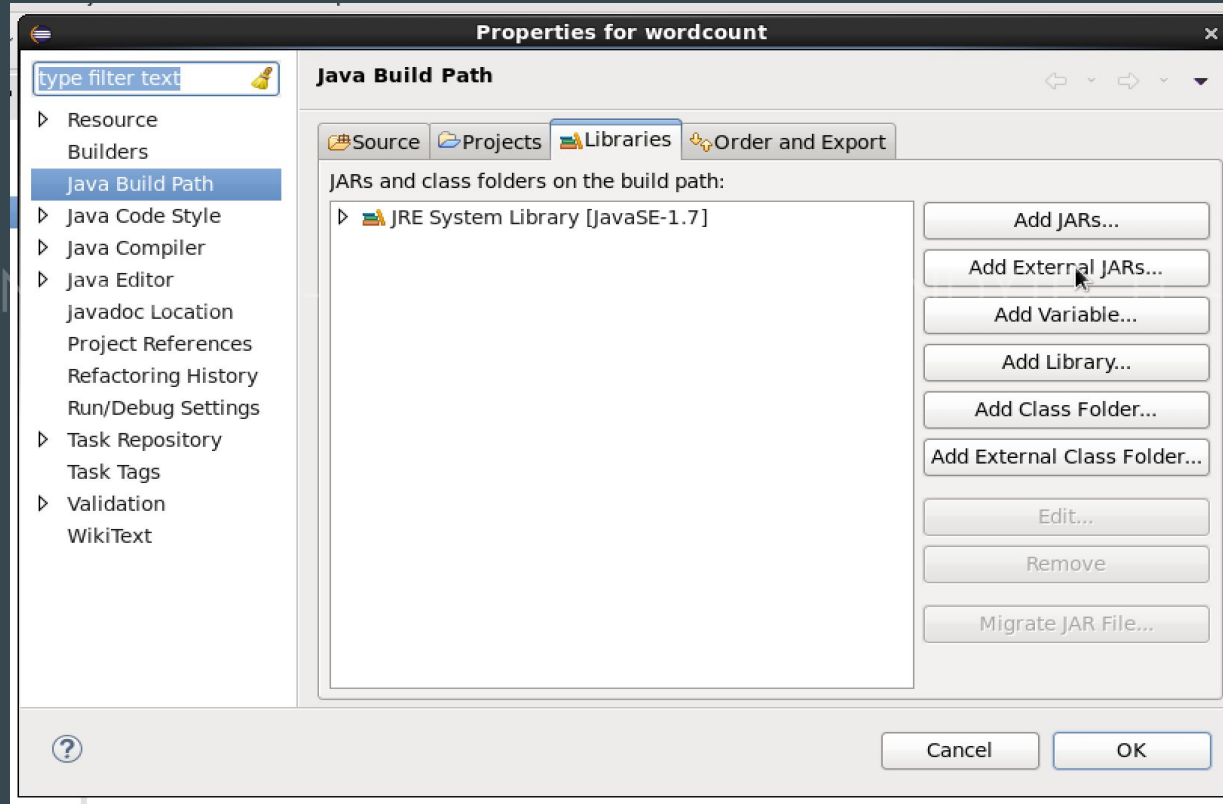
```
1 import org.apache.hadoop.io.IntWritable;
2
3
4
5
6
7
8 public class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
9
10 public void map(LongWritable key, Text value, Context context)
11     throws IOException, InterruptedException {
12
13     String line = value.toString();
14     String[] words = line.split(" ");
15
16     for (String word : words) {
17         context.write(new Text(word), new IntWritable(1));
18     }
19 }
20
21
22
23
24
25
26
27
28
29
30
31
```

You can see that your project has lot of errors. The reason is that libraries are missing. so , We need to add relevant jars.

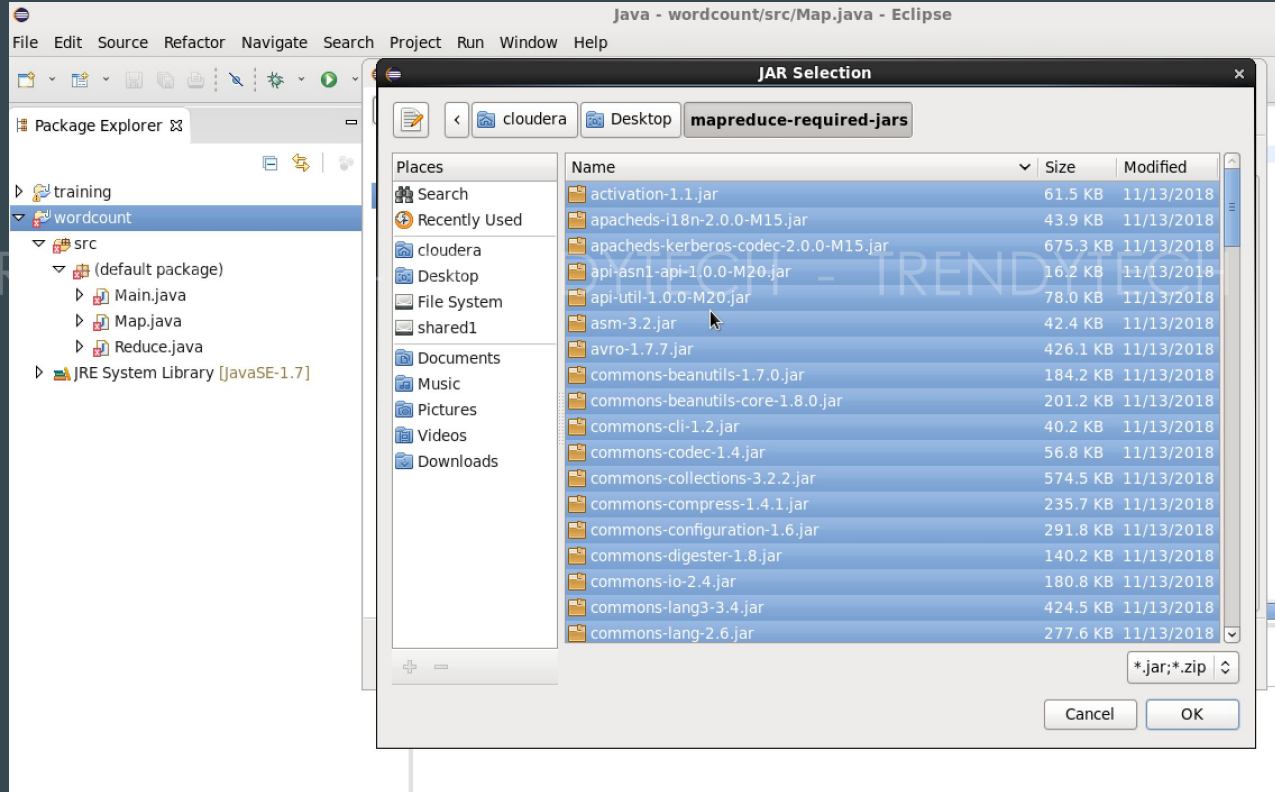
Adding the relevant Jars to resolve errors



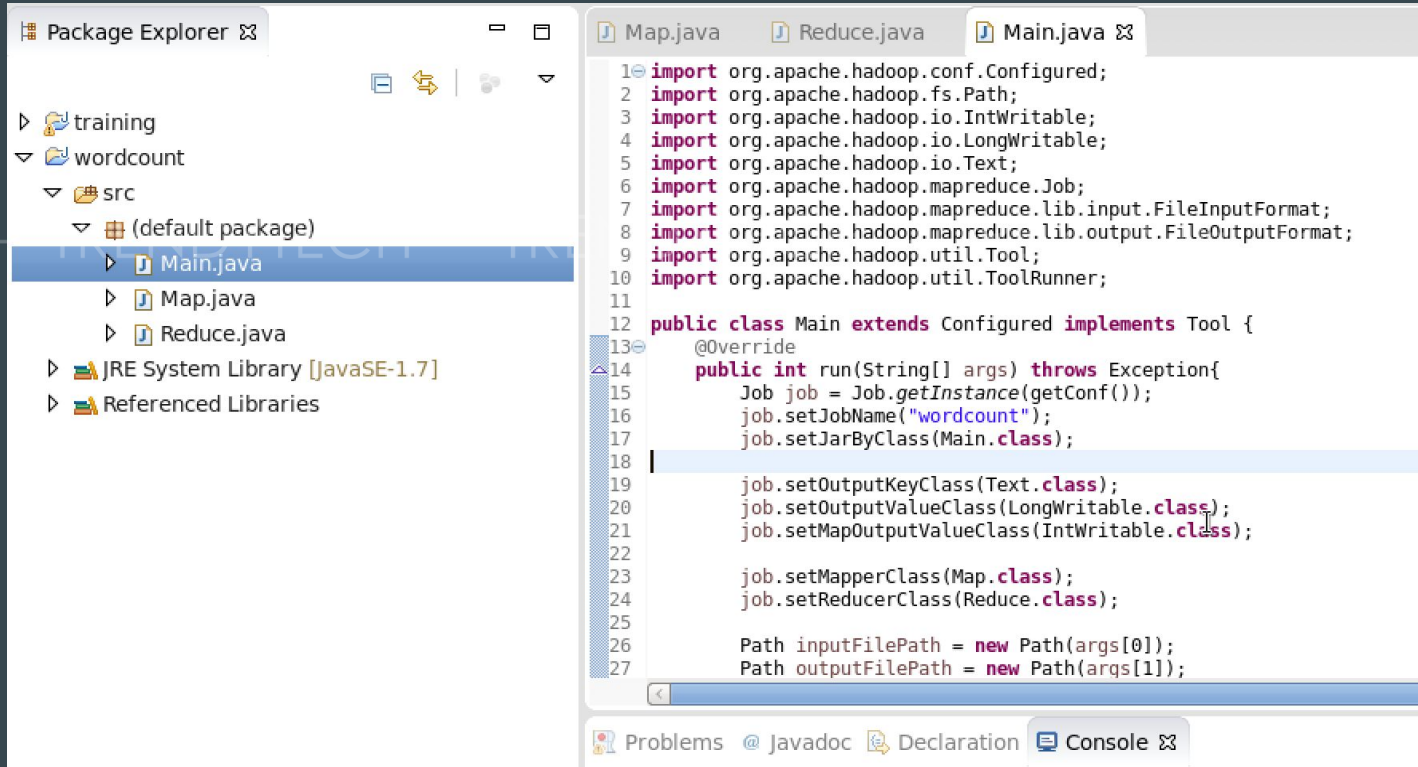
Click on Add External Jars under Libraries



Select all the jars from the jars that are given



Once all relevant jars are added then all errors will go away



The screenshot shows an IDE interface with the Package Explorer on the left and the Main.java file open in the editor on the right.

Package Explorer:

- training
 - wordcount
 - src
 - (default package)
 - Main.java (selected)
 - Map.java
 - Reduce.java
- JRE System Library [JavaSE-1.7]
- Referenced Libraries

Main.java Code:

```
1 import org.apache.hadoop.conf.Configured;
2 import org.apache.hadoop.fs.Path;
3 import org.apache.hadoop.io.IntWritable;
4 import org.apache.hadoop.io.LongWritable;
5 import org.apache.hadoop.io.Text;
6 import org.apache.hadoop.mapreduce.Job;
7 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
8 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
9 import org.apache.hadoop.util.Tool;
10 import org.apache.hadoop.util.ToolRunner;
11
12 public class Main extends Configured implements Tool {
13     @Override
14     public int run(String[] args) throws Exception{
15         Job job = Job.getInstance(getConf());
16         job.setJobName("wordcount");
17         job.setJarByClass(Main.class);
18
19         job.setOutputKeyClass(Text.class);
20         job.setOutputValueClass(LongWritable.class);
21         job.setMapOutputValueClass(IntWritable.class);
22
23         job.setMapperClass(Map.class);
24         job.setReducerClass(Reduce.class);
25
26         Path inputFilePath = new Path(args[0]);
27         Path outputFilePath = new Path(args[1]);
```

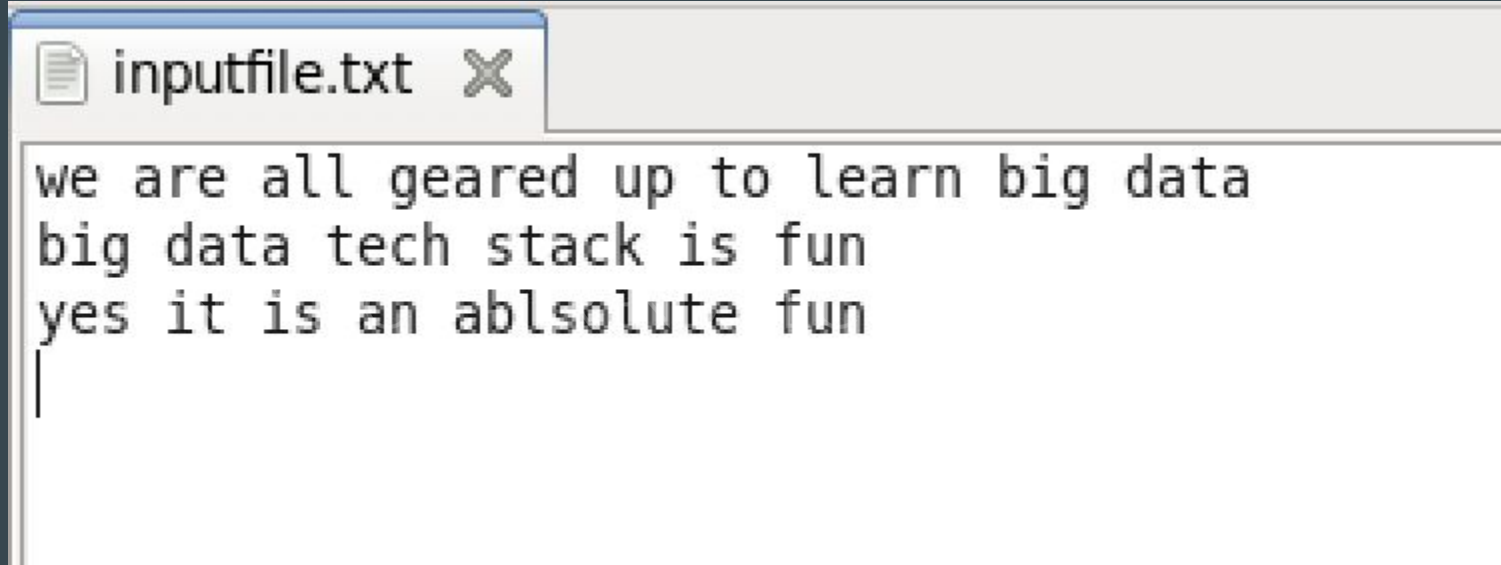


Now we need to create a input file with some content

```
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ cd Desktop/  
[cloudera@quickstart Desktop]$ mkdir inputfolder  
[cloudera@quickstart Desktop]$ cd inputfolder  
[cloudera@quickstart inputfolder]$ gedit inputfile.txt
```

Open terminal -> navigate to Desktop -> create a new directory -> inside new directory create a file using gedit.

Have some content in file



```
inputfile.txt X
we are all geared up to learn big data
big data tech stack is fun
yes it is an absolute fun
|
```

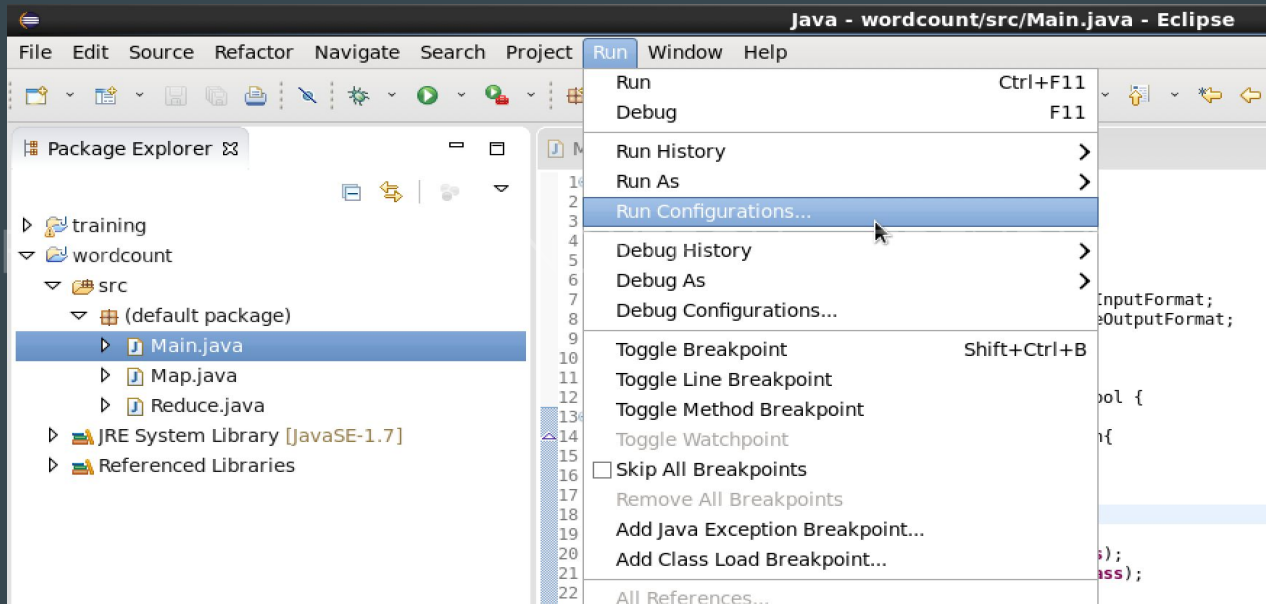
Make sure some words repeat so that you can visualize the results well.



Get the complete input file path using pwd command in terminal

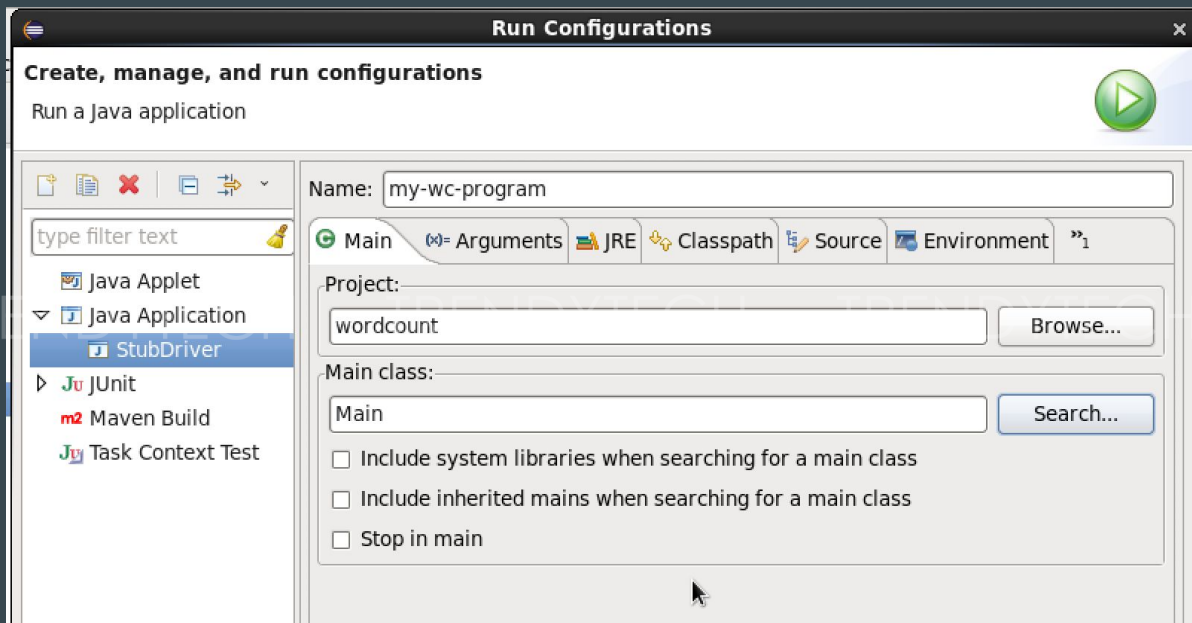
```
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ cd Desktop/  
[cloudera@quickstart Desktop]$ mkdir inputfolder  
[cloudera@quickstart Desktop]$ cd inputfolder  
[cloudera@quickstart inputfolder]$ gedit inputfile.txt  
[cloudera@quickstart inputfolder]$ pwd  
/home/cloudera/Desktop/inputfolder  
[cloudera@quickstart inputfolder]$
```

Now time to set the arguments and Run the project



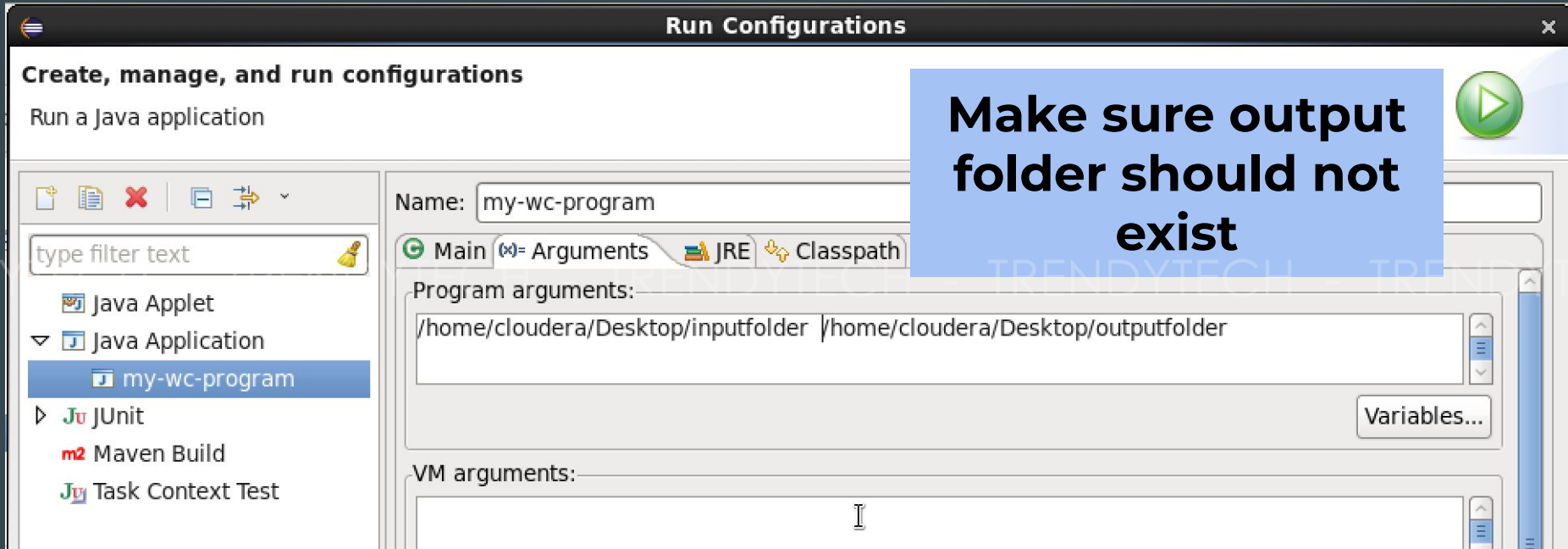
Click on Run -> Run Configurations

Setting up Run Configurations



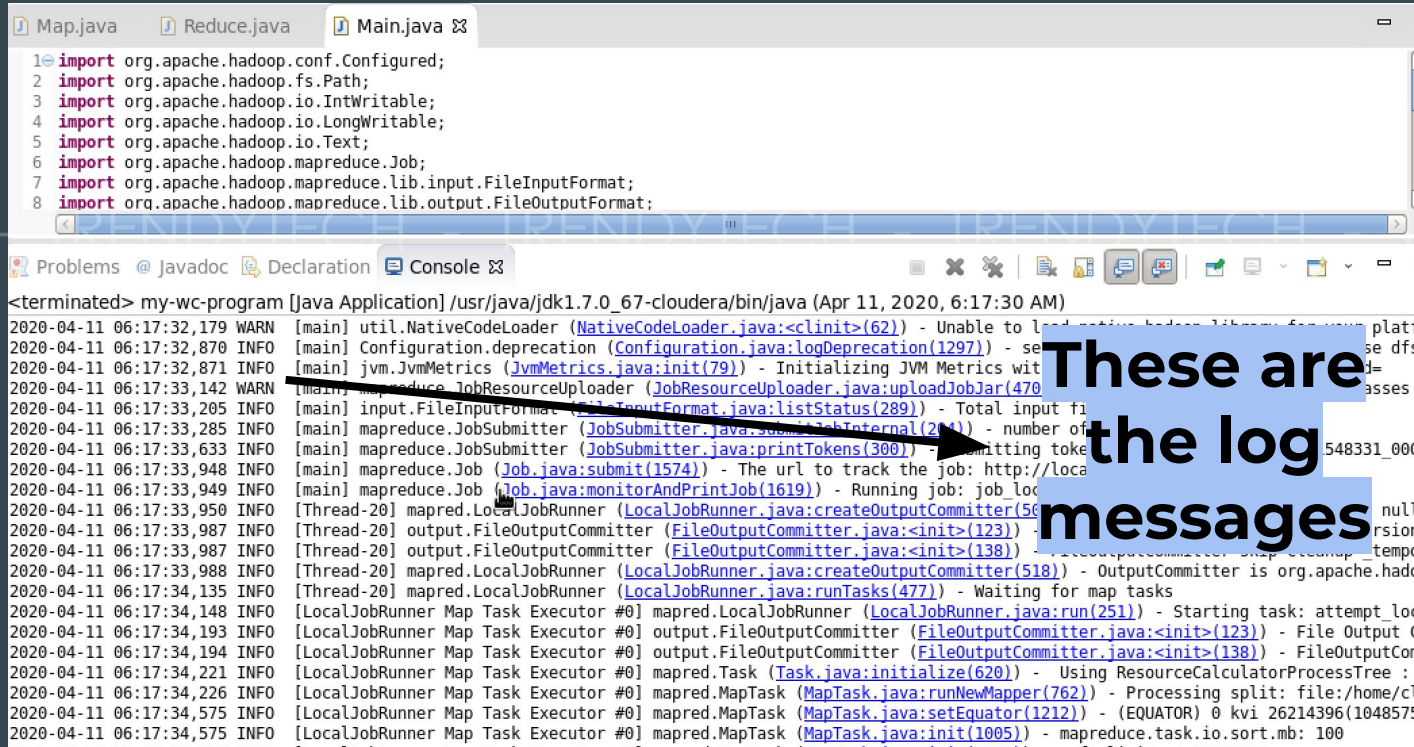
**Give your configuration a Name.
Also set your project name & Main class**

Setting up the runtime arguments and run it



We need to give 2 parameters separated by space. 1st one is input folder path & second is output folder path.

Just check that there should not be any error messages in the logs



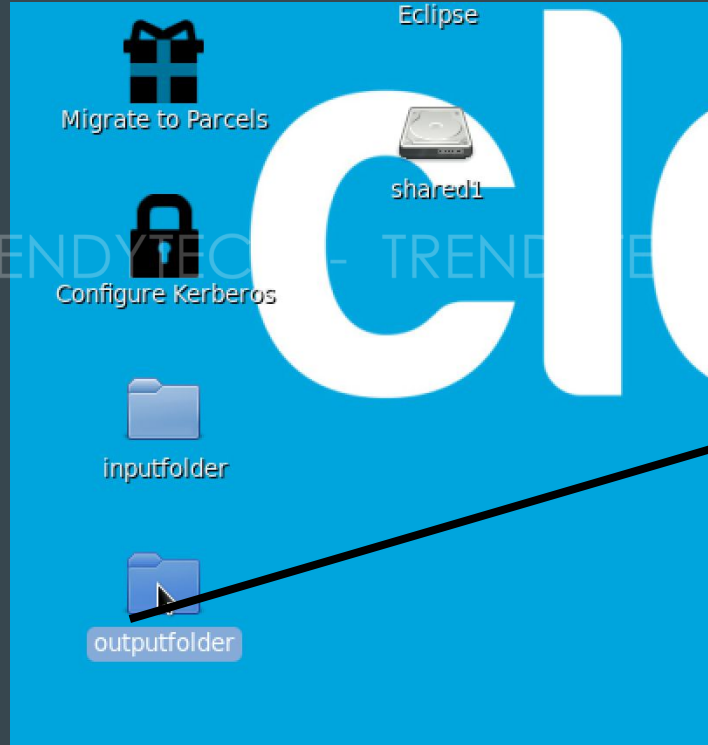
```
1 import org.apache.hadoop.conf.Configured;
2 import org.apache.hadoop.fs.Path;
3 import org.apache.hadoop.io.IntWritable;
4 import org.apache.hadoop.io.LongWritable;
5 import org.apache.hadoop.io.Text;
6 import org.apache.hadoop.mapreduce.Job;
7 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
8 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
```

Problems @ Javadoc Declaration Console

<terminated> my-wc-program [Java Application] /usr/java/jdk1.7.0_67-cloudera/bin/java (Apr 11, 2020, 6:17:30 AM)

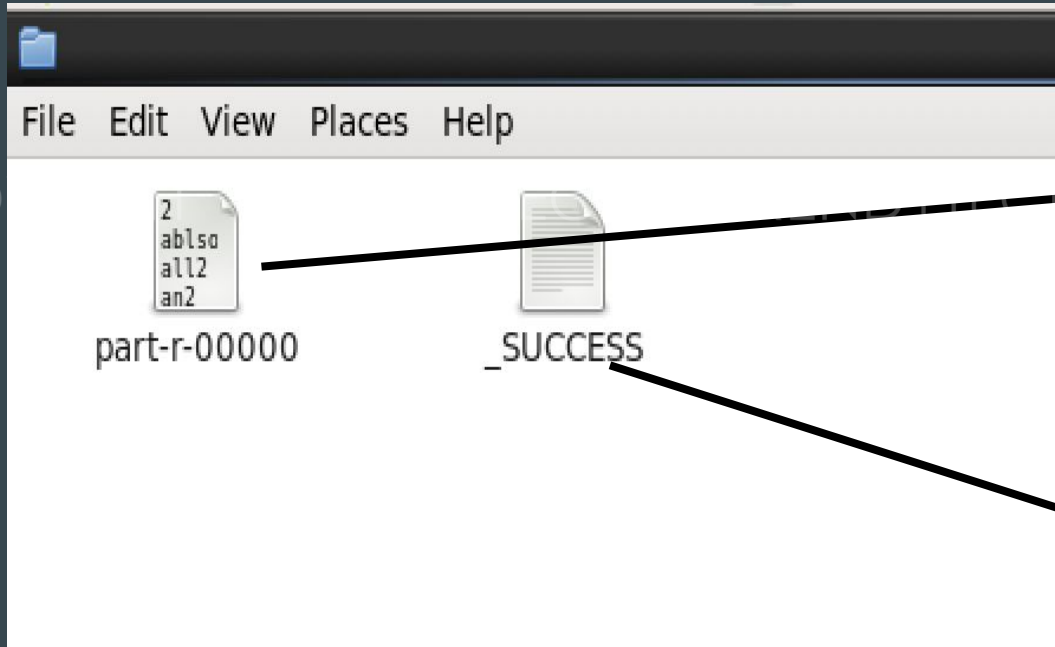
```
2020-04-11 06:17:32,179 WARN [main] util.NativeCodeLoader (NativeCodeLoader.java:<clinit>(62)) - Unable to load native hadoop library for your platform
2020-04-11 06:17:32,870 INFO [main] Configuration.deprecation (Configuration.java:logDeprecation(1297)) - see https://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-common/compatibility2.html for upgrade instructions
2020-04-11 06:17:32,871 INFO [main] jvm.JvmMetrics (JvmMetrics.java:init(79)) - Initializing JVM Metrics with unit: seconds
2020-04-11 06:17:33,142 WARN [main] mapreduce.JobResourceUploader (JobResourceUploader.java:uploadJobJar(470)) - Job jar is not uploaded
2020-04-11 06:17:33,205 INFO [main] input.FileInputFormat (FileInputFormat.java:listStatus(289)) - Total input files to process: 1
2020-04-11 06:17:33,285 INFO [main] mapreduce.JobSubmitter (JobSubmitter.java:submitJobInternal(244)) - number of maps: 1
2020-04-11 06:17:33,633 INFO [main] mapreduce.JobSubmitter (JobSubmitter.java:printTokens(300)) - Submitting tokens for job: job_1518331000000_000000
2020-04-11 06:17:33,948 INFO [main] mapreduce.Job (Job.java:submit(1574)) - The url to track the job: http://localhost:8020/job_1518331000000_000000
2020-04-11 06:17:33,949 INFO [main] mapreduce.Job (Job.java:monitorAndPrintJob(1619)) - Running job: job_1518331000000_000000
2020-04-11 06:17:33,950 INFO [Thread-20] mapred.LocalJobRunner (LocalJobRunner.java:createOutputCommitter(518)) - OutputCommitter is org.apache.hadoop.mapreduce.v2.lib.output.FileOutputCommitter
2020-04-11 06:17:33,987 INFO [Thread-20] output.FileOutputCommitter (FileOutputCommitter.java:<init>(123)) - Using temp directory: /tmp/hadoop-mapreduce/usercache/hadoop-mapreduce-mapreducejob_1518331000000_000000/tmp
2020-04-11 06:17:33,987 INFO [Thread-20] output.FileOutputCommitter (FileOutputCommitter.java:<init>(138)) - FileOutputCommitter cleanup temp directory: /tmp/hadoop-mapreduce/usercache/hadoop-mapreduce-mapreducejob_1518331000000_000000/tmp
2020-04-11 06:17:33,988 INFO [Thread-20] mapred.LocalJobRunner (LocalJobRunner.java:createOutputCommitter(518)) - OutputCommitter is org.apache.hadoop.mapreduce.v2.lib.output.FileOutputCommitter
2020-04-11 06:17:34,135 INFO [Thread-20] mapred.LocalJobRunner (LocalJobRunner.java:runTasks(477)) - Waiting for map tasks
2020-04-11 06:17:34,148 INFO [LocalJobRunner Map Task Executor #0] mapred.LocalJobRunner (LocalJobRunner.java:run(251)) - Starting task: attempt localJobRunner_1518331000000_000000_m0
2020-04-11 06:17:34,193 INFO [LocalJobRunner Map Task Executor #0] output.FileOutputCommitter (FileOutputCommitter.java:<init>(123)) - FileOutputCommitter cleanup temp directory: /tmp/hadoop-mapreduce/usercache/hadoop-mapreduce-mapreducejob_1518331000000_000000/tmp
2020-04-11 06:17:34,194 INFO [LocalJobRunner Map Task Executor #0] output.FileOutputCommitter (FileOutputCommitter.java:<init>(138)) - FileOutputCommitter cleanup temp directory: /tmp/hadoop-mapreduce/usercache/hadoop-mapreduce-mapreducejob_1518331000000_000000/tmp
2020-04-11 06:17:34,221 INFO [LocalJobRunner Map Task Executor #0] mapred.Task (Task.java:initialize(620)) - Using ResourceCalculatorProcessTree : org.apache.hadoop.mapreduce.v2.app.task.ResourceCalculatorProcessTree$1
2020-04-11 06:17:34,226 INFO [LocalJobRunner Map Task Executor #0] mapred.MapTask (MapTask.java:runNewMapper(762)) - Processing split: file:/home/cloudera/wordcount/words.txt:1
2020-04-11 06:17:34,575 INFO [LocalJobRunner Map Task Executor #0] mapred.MapTask (MapTask.java:setEquator(1212)) - (EQUATOR) 0 kvi 26214396(1048575)
2020-04-11 06:17:34,575 INFO [LocalJobRunner Map Task Executor #0] mapred.MapTask (MapTask.java:run(1005)) - mapreduce.task.io.sort.mb: 100
```

If the MR job is successful, a new output folder should be created



**You can see
the new
output
folder is
created.**

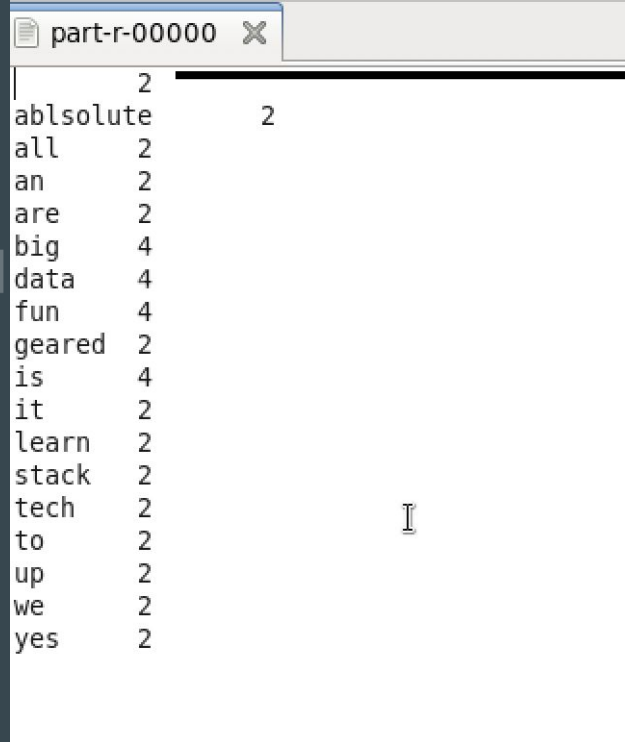
See the files in output folder



Since we used one reducer there should be one part file.

_SUCCESS indicated job is successful

See the content of output file



	2	
absolute	2	
all	2	
an	2	
are	2	
big	4	
data	4	
fun	4	
geared	2	
is	4	
it	2	
learn	2	
stack	2	
tech	2	
to	2	
up	2	
we	2	
yes	2	

You can see empty space is also treated as word. Because if we give 2 spaces consecutively. The second space is treated as word.

The output is in ascending order of the words (keys)



You can check the output through terminal also using linux commands

```
[cloudera@quickstart ~]$  
[cloudera@quickstart ~]$ cd Desktop/  
[cloudera@quickstart Desktop]$ mkdir inputfolder  
[cloudera@quickstart Desktop]$ cd inputfolder  
[cloudera@quickstart inputfolder]$ gedit inputfile.txt  
[cloudera@quickstart inputfolder]$ pwd  
/home/cloudera/Desktop/inputfolder  
[cloudera@quickstart inputfolder]$ cd ..  
[cloudera@quickstart Desktop]$ cd outputfolder/  
[cloudera@quickstart outputfolder]$ ls -ltr  
total 4  
-rw-r--r-- 1 cloudera cloudera 114 Apr 11 06:17 part-r-00000  
-rw-r--r-- 1 cloudera cloudera  0 Apr 11 06:17 _SUCCESS  
[cloudera@quickstart outputfolder]$
```

cat command to see the output content

```
[cloudera@quickstart outputfolder]$ cat part-r-00000
2
absolute      2
all           2
an            2
are           2
big           4
data          4
fun           4
geared        2
is            4
it            2
learn         2
stack         2
tech          2
to            2
up            2
we            2
yes           2
```



We have successfully executed a mapreduce program

Happy Learning!!!



5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT



9108179578

Call for more details



Follow US

Trainer Mr. Sumit Mittal

Phone 9108179578

Email trendytech.sumit@gmail.com

Website <https://trendytech.in/courses/big-data-online-training/>

LinkedIn <https://www.linkedin.com/in/bigdatabysumit/>

Twitter @BigdataBySumit

Instagram bigdatabysumit

Facebook <https://www.facebook.com/trendytech.in/>

Youtube TrendyTech