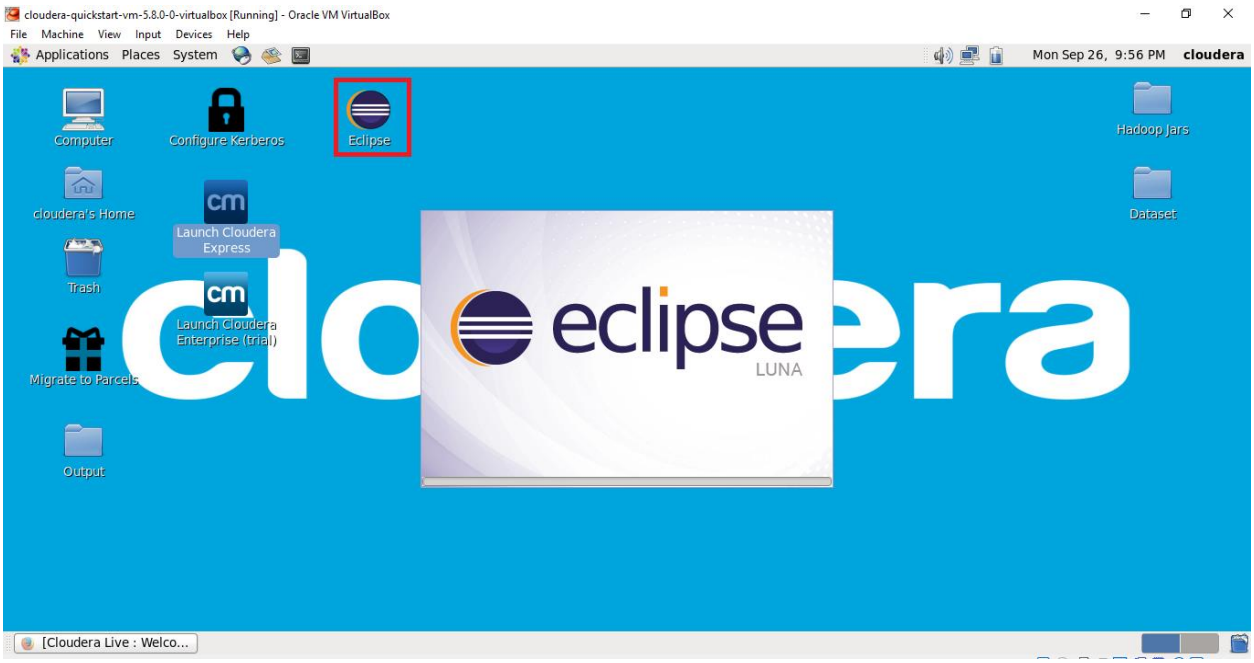
Run MapReduce program on Cloudera CDH 5.8

Prerequisites:

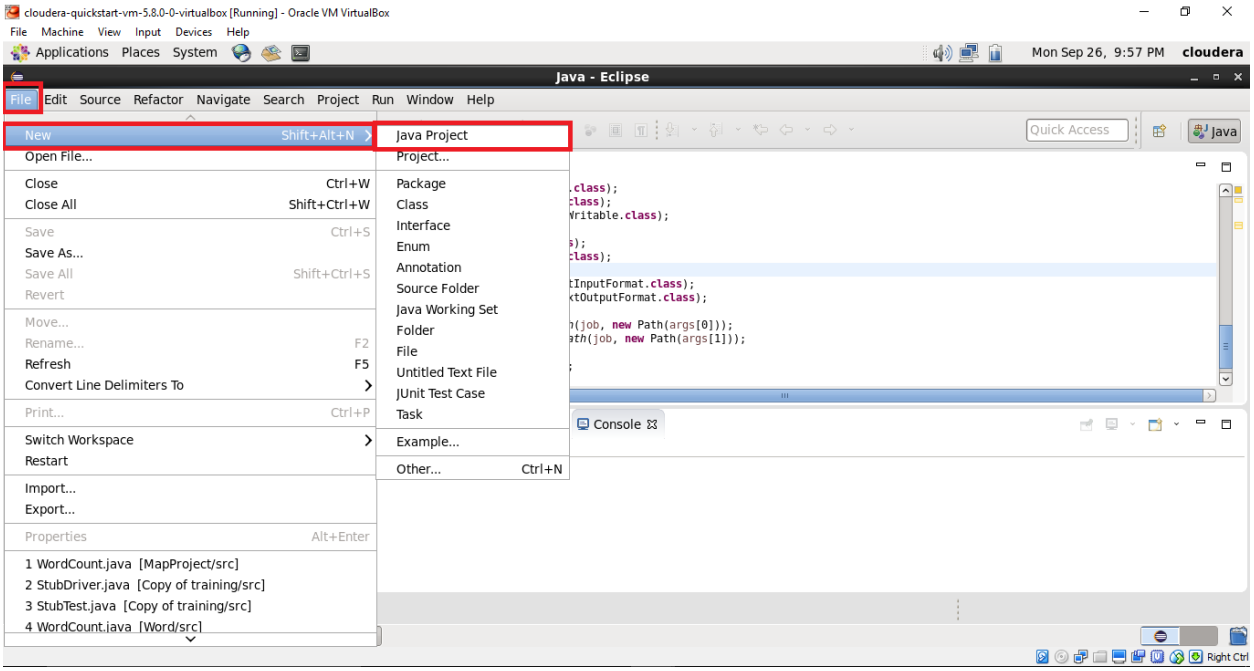
1) Cloudera CDH 5.8 installed on your oracle virtual box.
2) Working knowledge of basic java programming language

How to Create Map Reduce Program in Eclipse

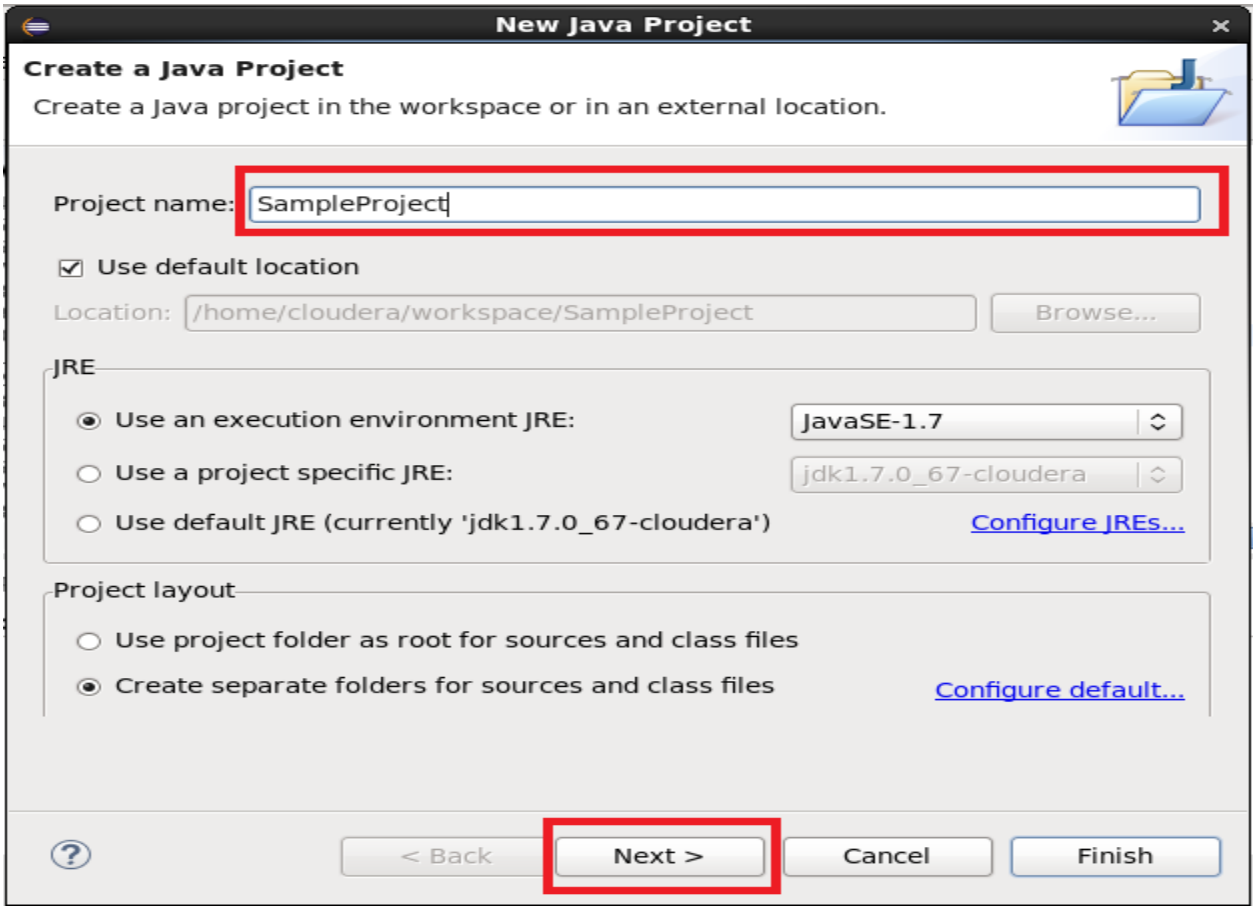
Step 1: Start Cloudera virtual machine Go to Desktop and open Eclipse



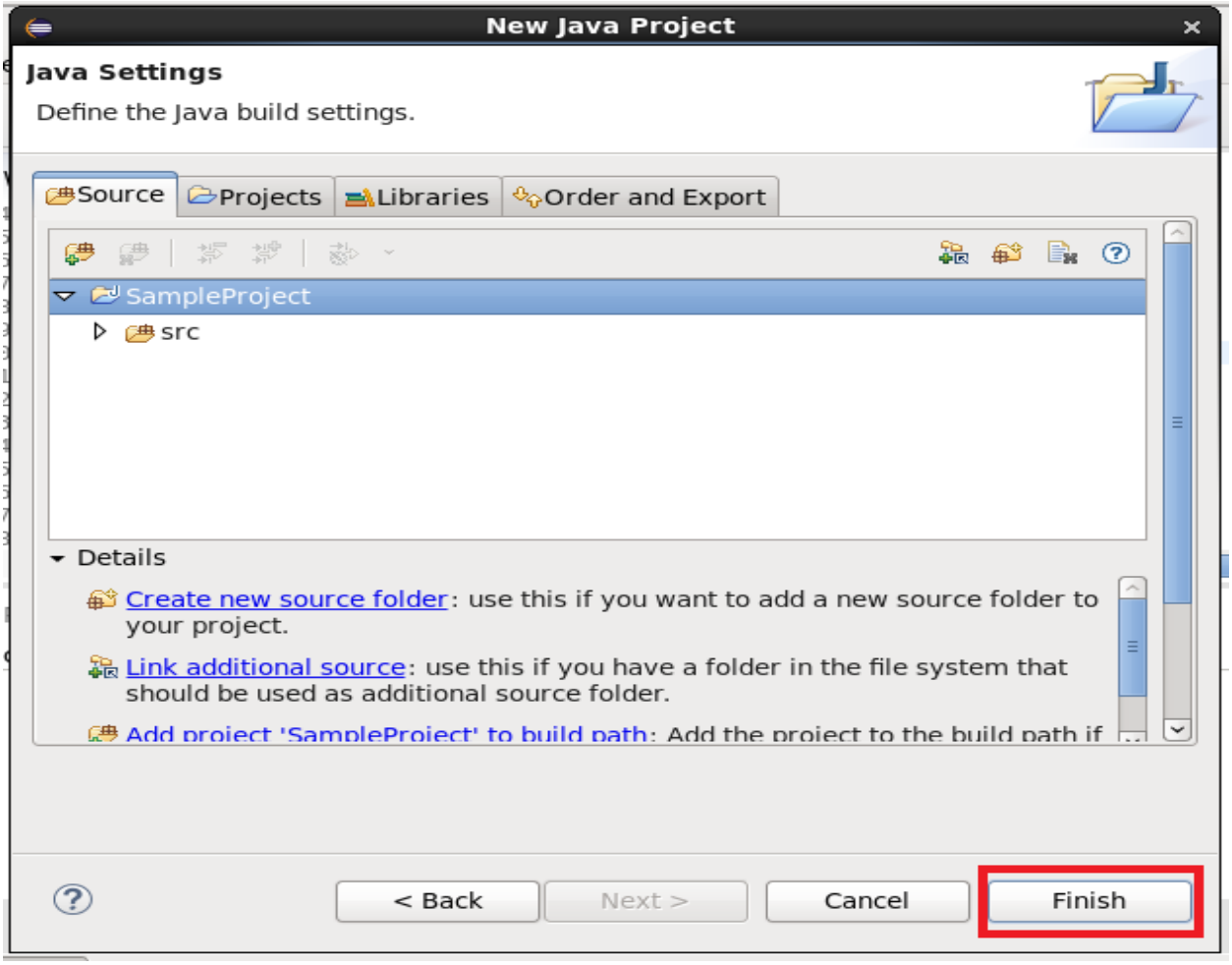
Step 2: Go to menu bar and click on File→New →Java Project.

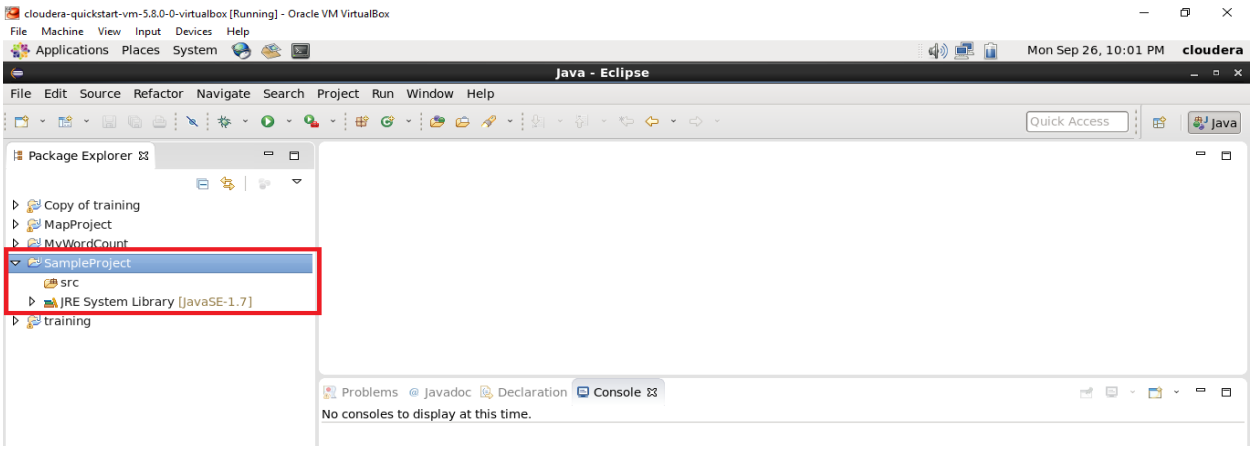


Step 3: Give the name for the project and click next.

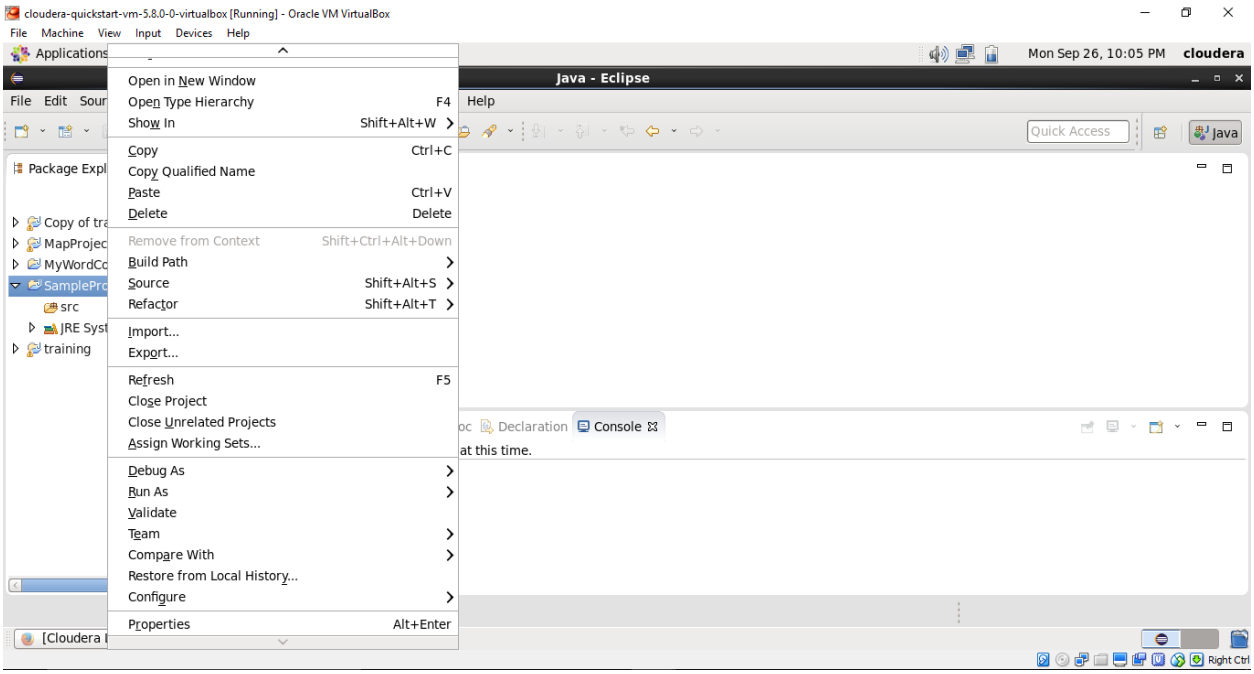


Step 4: Click Finish. After that you will find your project in Package Explorer.

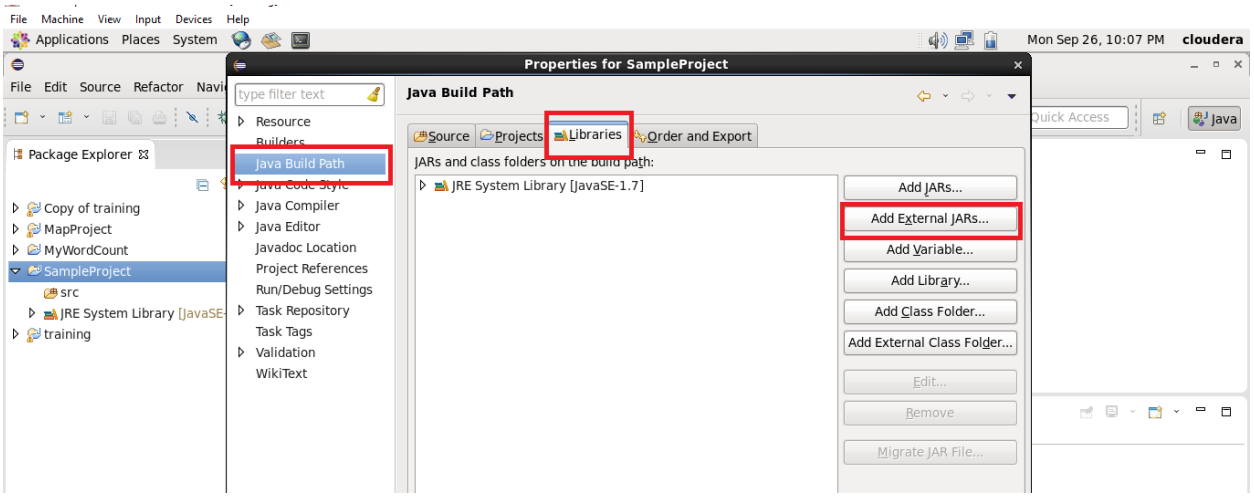




Step 5: Now It's Time to add Hadoop jars, that are required to run MapReduce project. Right Click on the project and click properties



Step 6: Go to Java Built Path, Click on Libraries and then Click Add External Jar



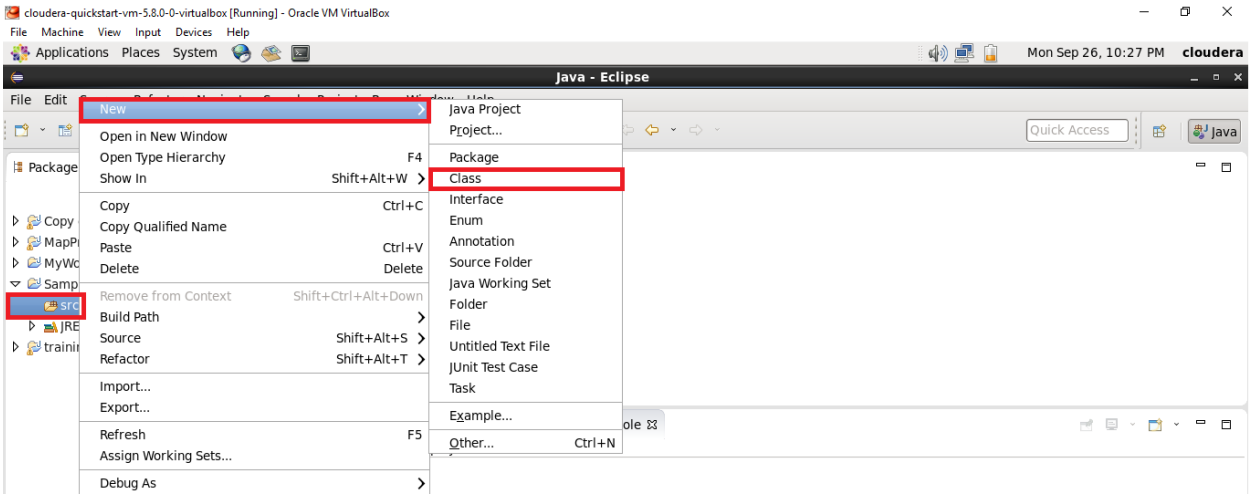
Step 7: Add the following Jars by going to these locations:

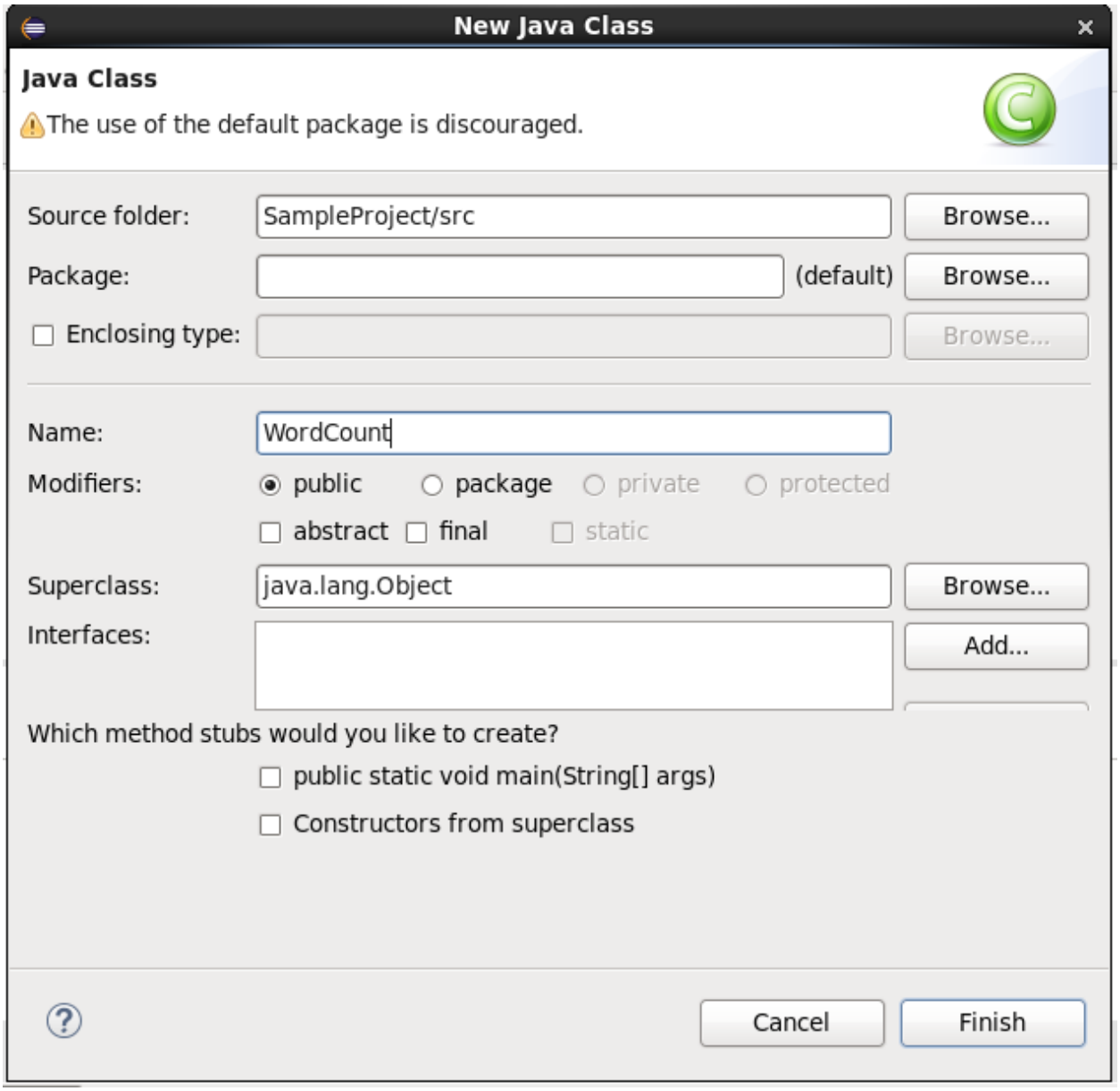
Usr/lib/Hadoop/Client-0.20 - All Jars from this folder

Usr/lib/Hadoop/ 3 jars from this folder 1) Hadoop-Annotations.jar 2) Hadoop-Auth.jar 3) Hadoop-common.jar

Usr/lib/Hadoop/lib -only one jar - commons-httpclient-3.1.jar

Step 8: Create a java class in SRC folder and give the mane of the class as WordCount and click finish.



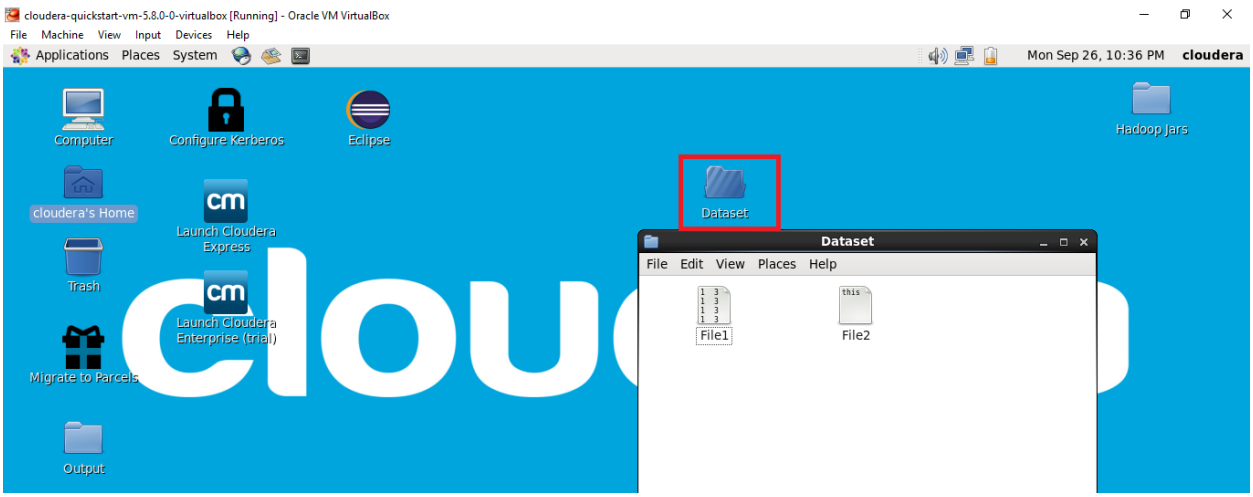


Step 9: Copy and paste the below code in the class.

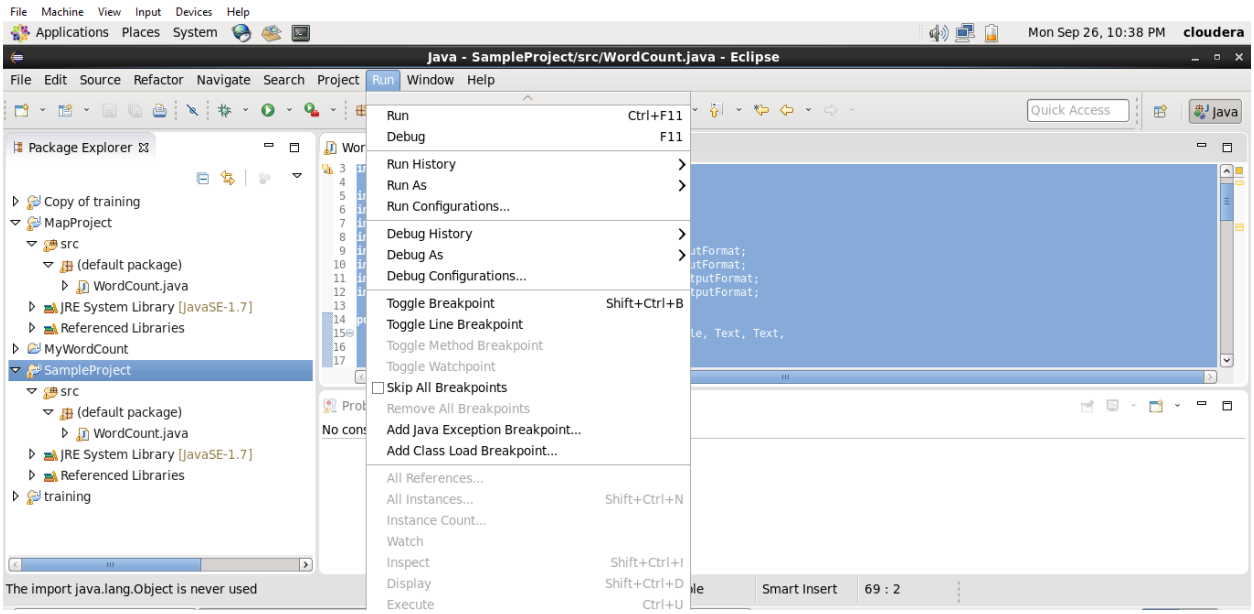
```
import java.io.IOException; import java.util.*;
import java.lang.Object;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {
   public static class Map extends Mapper<LongWritable, Text, Text,
```

```
                    IntWritable>
    {
       private final static IntWritable one = new IntWritable(1);
       private static final int UPPER_CASE = 0;
       private static final int LOWER_CASE = 0;
       private Text word = new Text();

       public void map(LongWritable key, Text value, Context context)
          throws IOException, InterruptedException
       {
          String line = value.toString();
          StringTokenizer tokenizer = new StringTokenizer(line);
          while (tokenizer.hasMoreTokens()) {
             word.set(tokenizer.nextToken());
             context.write(word, one);
          }
       }
    }

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

    public static void main(String[] args) throws Exception
    {
       Configuration conf = new Configuration();

       Job job = new Job(conf, "WordCount");
       job.setJarByClass(WordCount.class);
       job.setOutputKeyClass(Text.class);
       job.setOutputValueClass(IntWritable.class);

       job.setMapperClass(Map.class);
       job.setReducerClass(Reduce.class);

       job.setInputFormatClass(TextInputFormat.class);
       job.setOutputFormatClass(TextOutputFormat.class);

       FileInputFormat.addInputPath(job, new Path(args[0]));
       FileOutputFormat.setOutputPath(job, new Path(args[1]));

       job.waitForCompletion(true);
    }
}
```

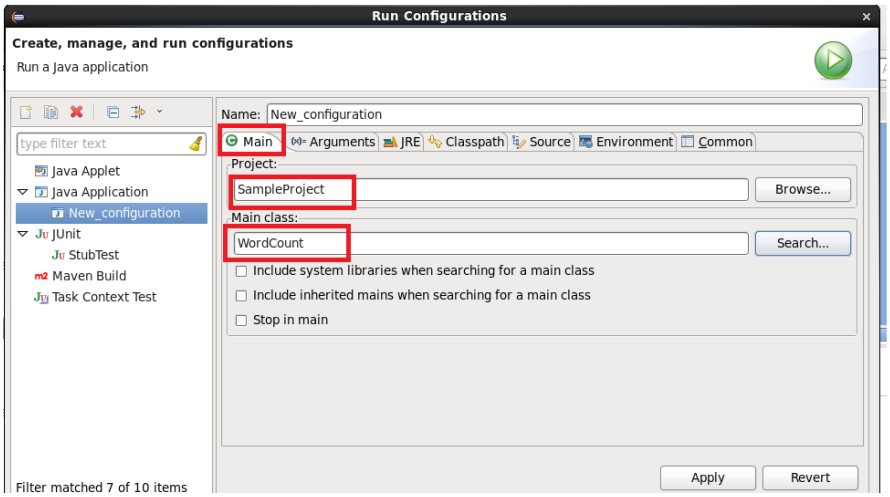
Step 10: Create a folder on Desktop and create multiple input file in the folder which contains different text.



Step 11: Go back to Eclipse and click on run from menu bar and select Run Configuration

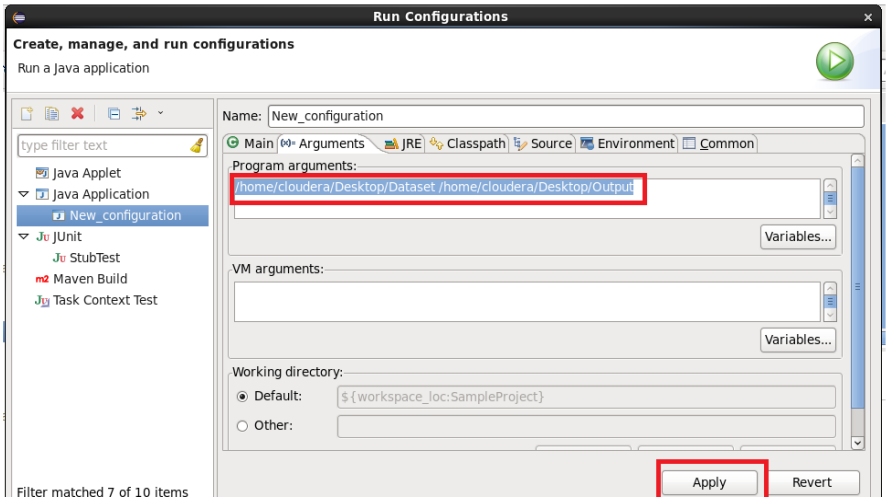


Step 12: In Main section select the project that we have created and also select the main class from that project.

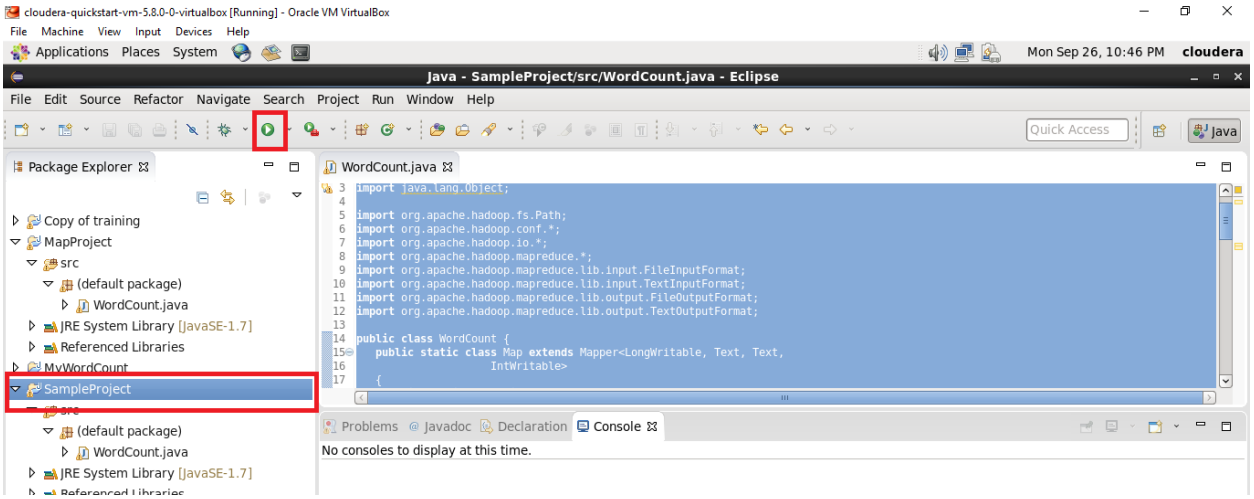


Step 13: Go to argument tab and provide the path of input and output folder separated by space (Do not create output folder It will be automatically created in the provided path) and then click apply

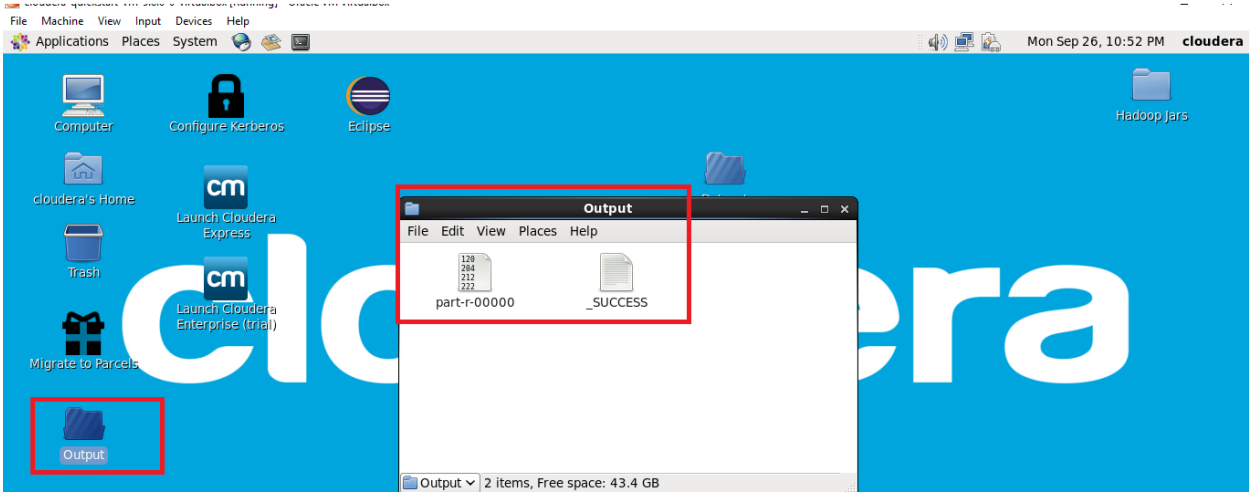
home/cloudera/Desktop/Dataset home/cloudera/Desktop/Output



Step 14: Run the program by selecting the project and clicking the run button.



Step 15: If there are no errors in the program you will be able to see the output folder in the provided location with the following output files.



For more detail visit

http://blog.cloudera.com/blog/2013/08/how-to-use-eclipse-with-mapreduce-in-clouderas-quickstart-vm/