

Maharishi University of

Management

CS522 – Big Data

Dr. Premchand Nair

Tutorial

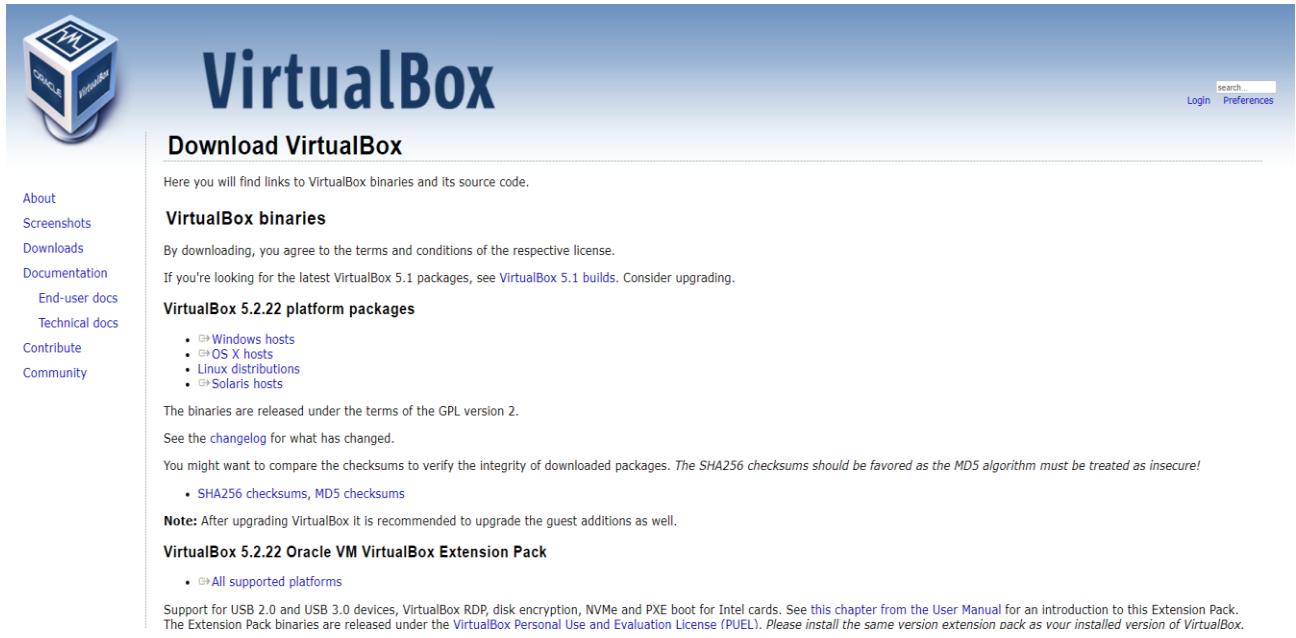
Set Up a Single Node Cluster

Autor	ID	Date
Anne Guimaraes	986742	11/10/2018
Edwin Fonseca	986553	11/10/2018

This document describes the steps how to set up and configure a single-node Hadoop installation and optionally an eclipse development environment to create and test your programs on Microsoft Windows.

I. Install Oracle Virtual Box

Go to the virtual box website to download the installer:
<https://www.virtualbox.org/wiki/Downloads>



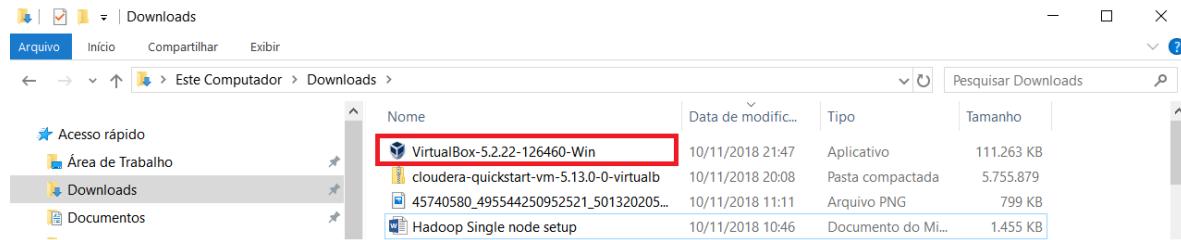
The screenshot shows the VirtualBox download page. On the left is a sidebar with links: About, Screenshots, Downloads, Documentation, End-user docs, Technical docs, Contribute, and Community. The main content area has a large blue header "VirtualBox". Below it is a section titled "Download VirtualBox" with a sub-section "VirtualBox binaries". It contains text about license terms and links to "VirtualBox 5.1 builds". A list titled "VirtualBox 5.2.22 platform packages" includes "Windows hosts", "OS X hosts", "Linux distributions", and "Solaris hosts". The "Windows hosts" link is highlighted with a red box. Below this is a note about GPL version 2 and a changelog link. A note about SHA256 checksums is present. A "Note" says to upgrade guest additions after upgrading VirtualBox. A "VirtualBox Extension Pack" section is also shown with a note about USB support and PUEL.

- Select the option: “Windows Hosts” to download setup file for Windows.



The screenshot shows the VirtualBox download page. The sidebar and main content area are identical to the previous screenshot, but the "Windows hosts" link in the "VirtualBox 5.2.22 platform packages" list is now highlighted with a red box, indicating it has been selected.

After download, double click on setup file to perform the VirtualBox installation



After finished installation, you will able to run the program.



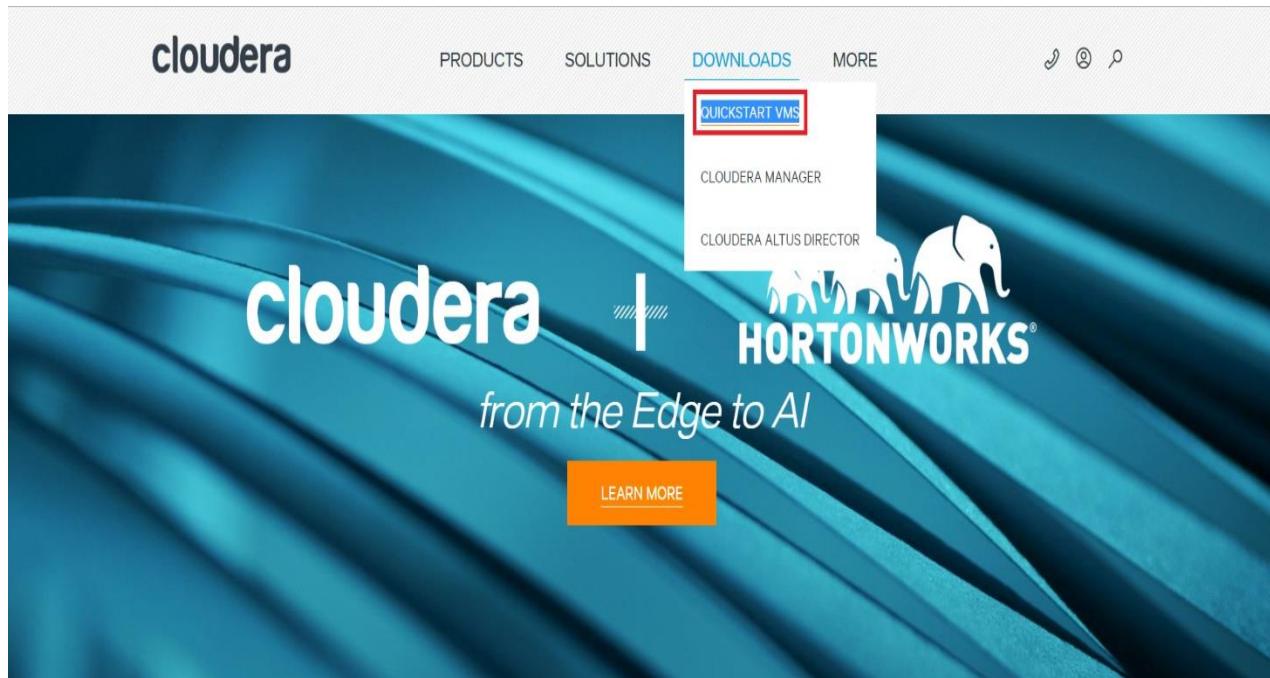
II. Download & Install Cloudera

a. Get Cloudera

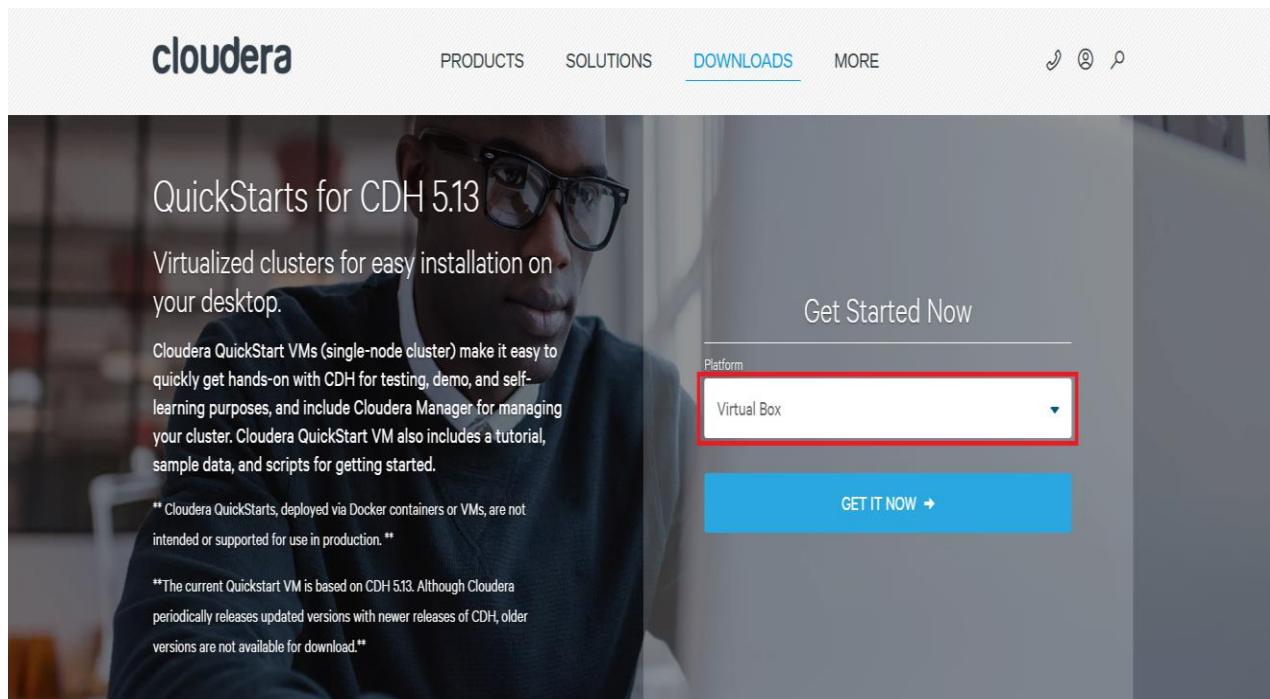
Go to the Cloudera website to download the installer:
<https://www.cloudera.com/>

The screenshot shows the official website for Cloudera. The header includes the 'cloudera' logo, navigation links for 'PRODUCTS', 'SOLUTIONS', 'DOWNLOADS', and 'MORE', and social media icons. The main visual is a large blue background with white text. It features the 'cloudera' logo in a large, bold font next to a plus sign, followed by the 'HORTONWORKS' logo with three white elephants. Below this, the tagline 'from the Edge to AI' is displayed. At the bottom left, there is an orange 'LEARN MORE' button.

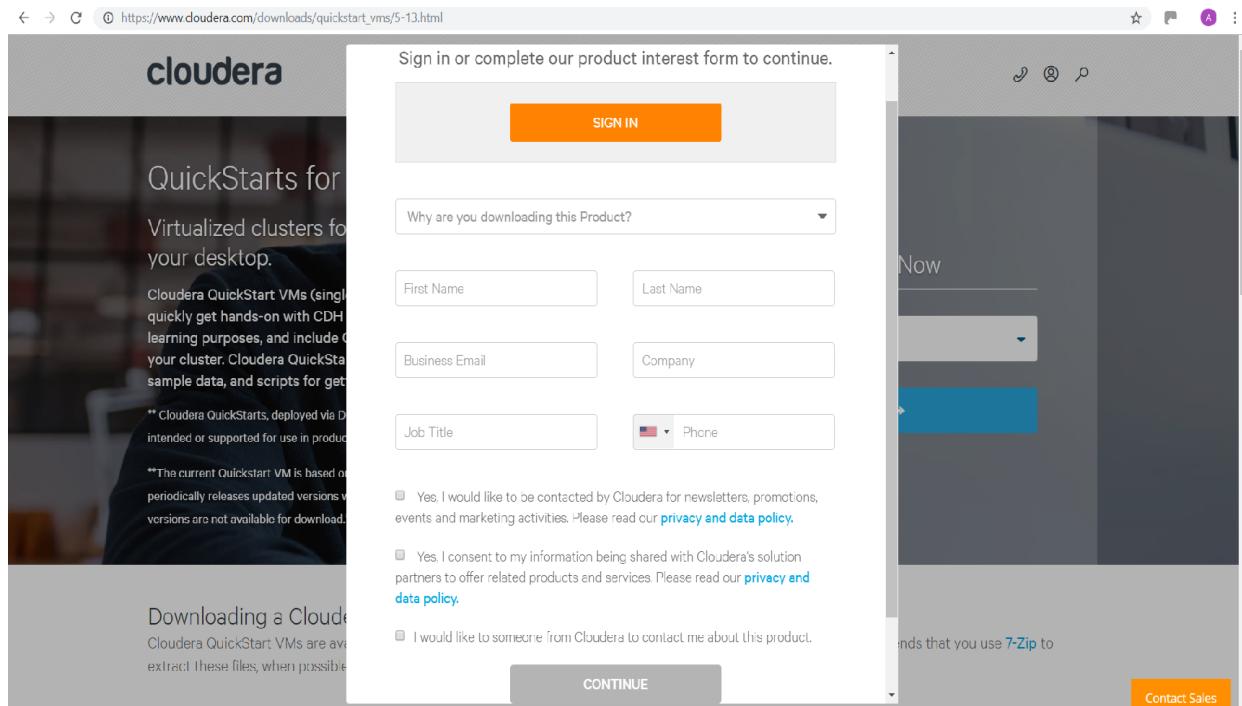
- Click in the option: “Downloads” and then, the option: “QUICKSTARTVMS”.



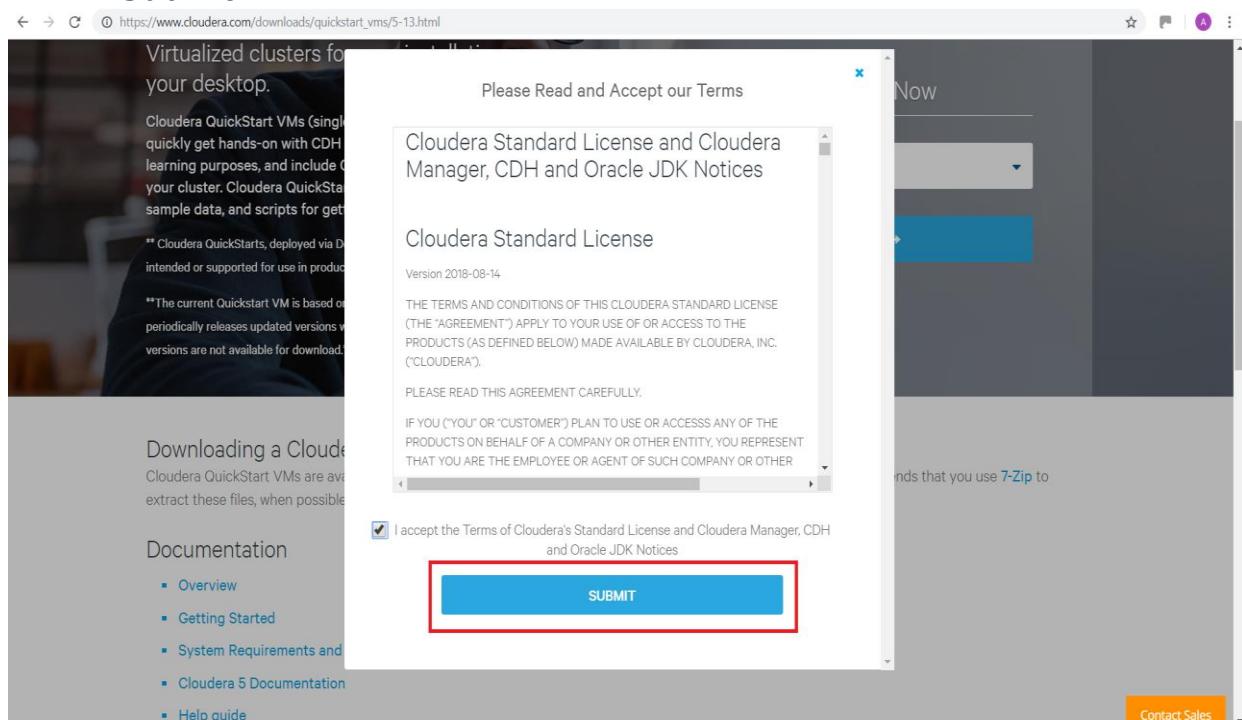
- After, it is necessary to select the option: “Virtual box” in the field: Platform and click in the bottom: “Get it now”.



- To be able to start the Download, you will need to Sign In, you will need an account to download Cloudera, if you don't have an account yet, you need to fill all the fields and click in "Continue".



- After, you will need to accept the Terms, clicking in the bottom: "Submit".



- Download will start, you will get the zip file after finished

Downloading a Cloudera QuickStart VM

Cloudera QuickStart VMs are available as Zip archives in Docker, VMware, KVM, and VirtualBox formats. Cloudera recommends that you use [7-Zip](#) to extract these files, when possible. (7-Zip performs well with large files.)

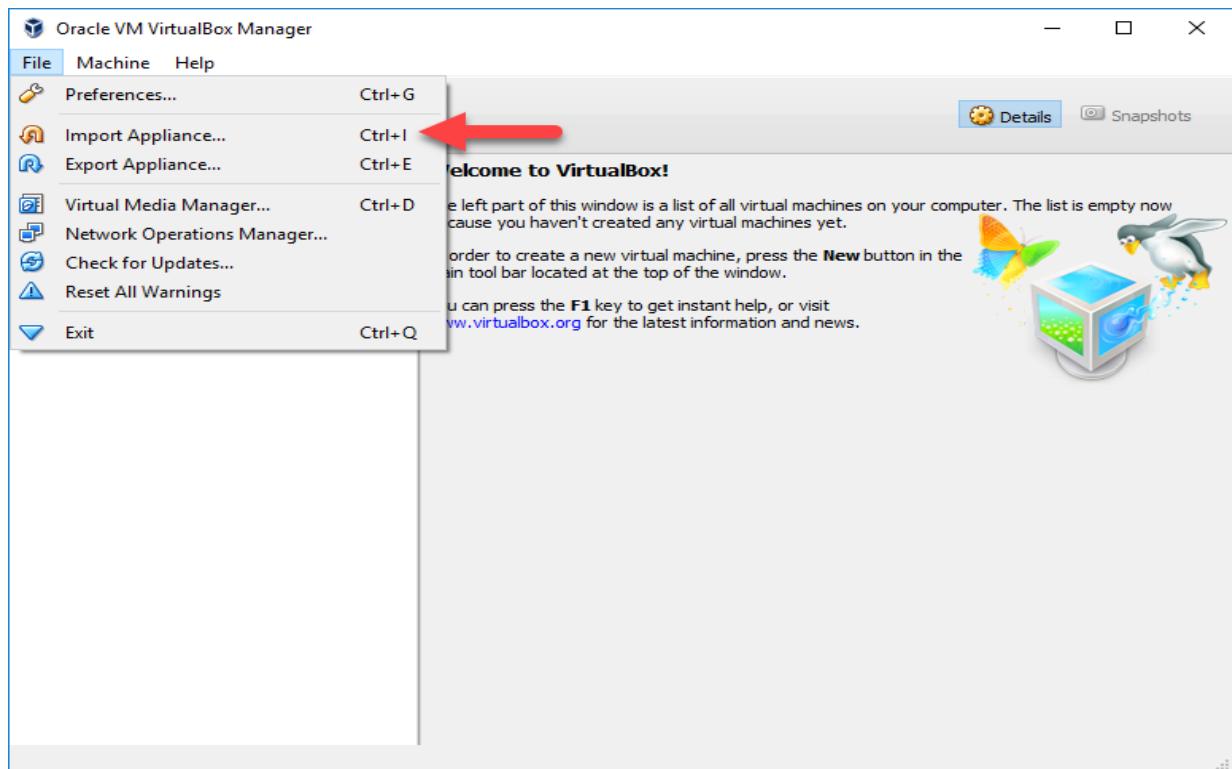
Documentation

- [Overview](#)
- [Getting Started](#)
- [System Requirements and Prerequisites](#)
- [Cloudera 5 Documentation](#)
- [Help guide](#)

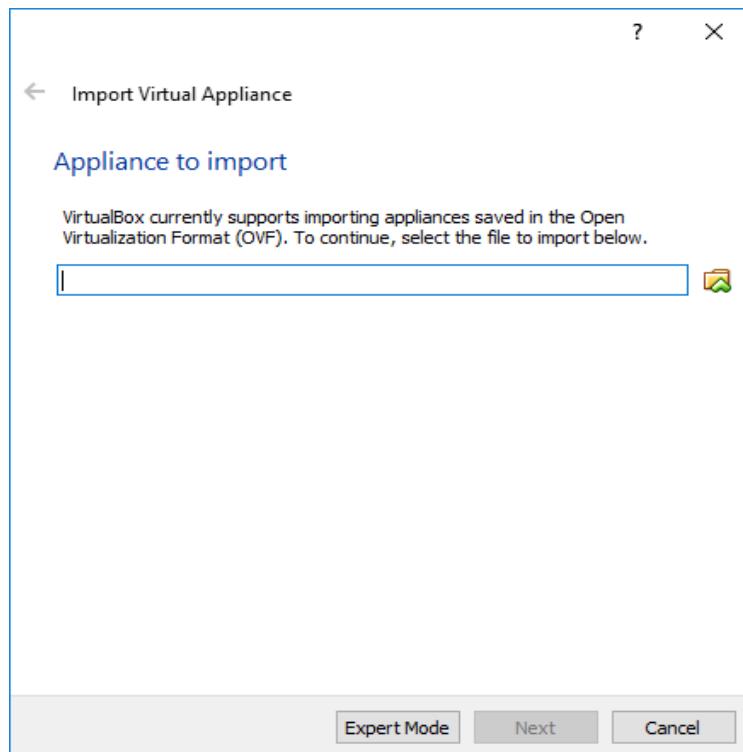


b. Install Cloudera

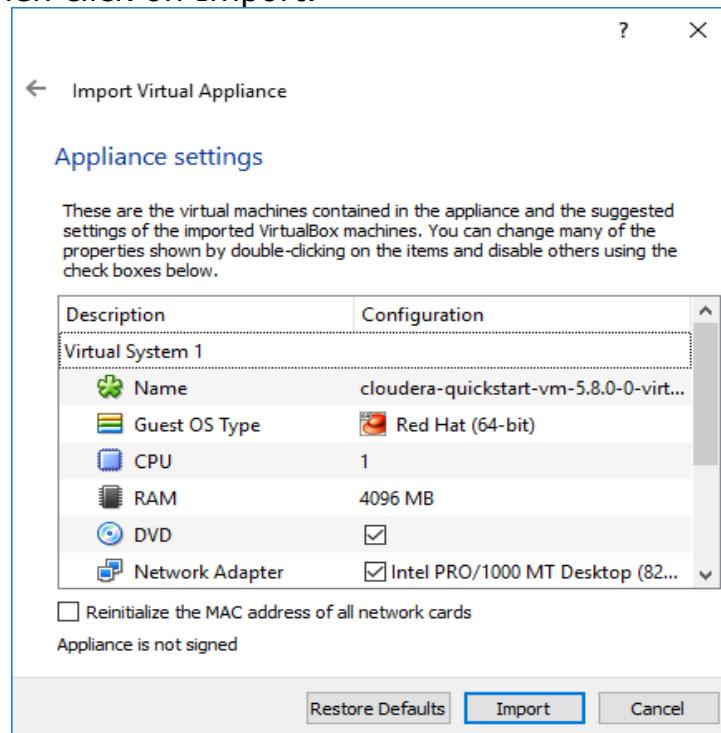
After you extract and unzip the file, you can import it to VirtualBox. Run VirtualBox, then click on "Import Appliance".



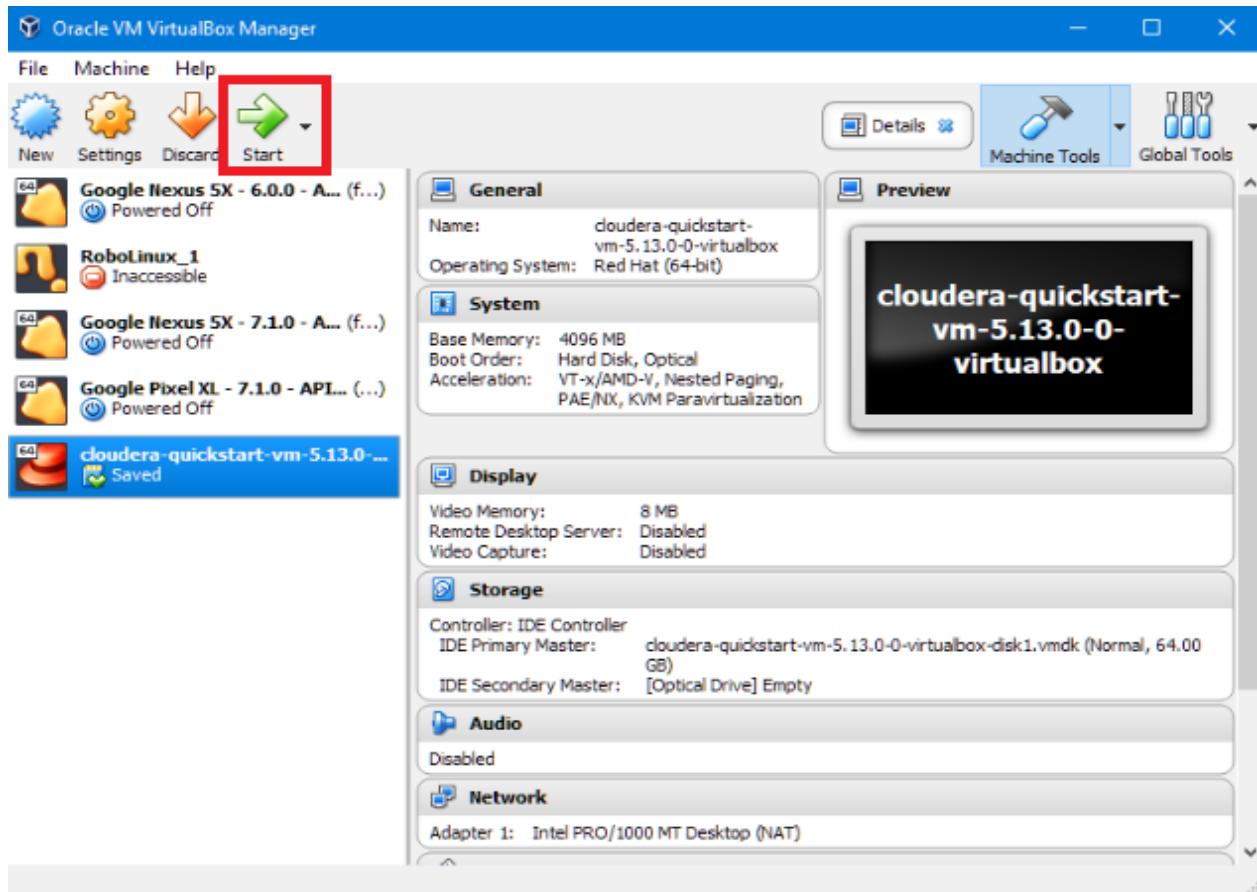
- Select your VirtualBox image file location, after that, click in the bottom: "Next".



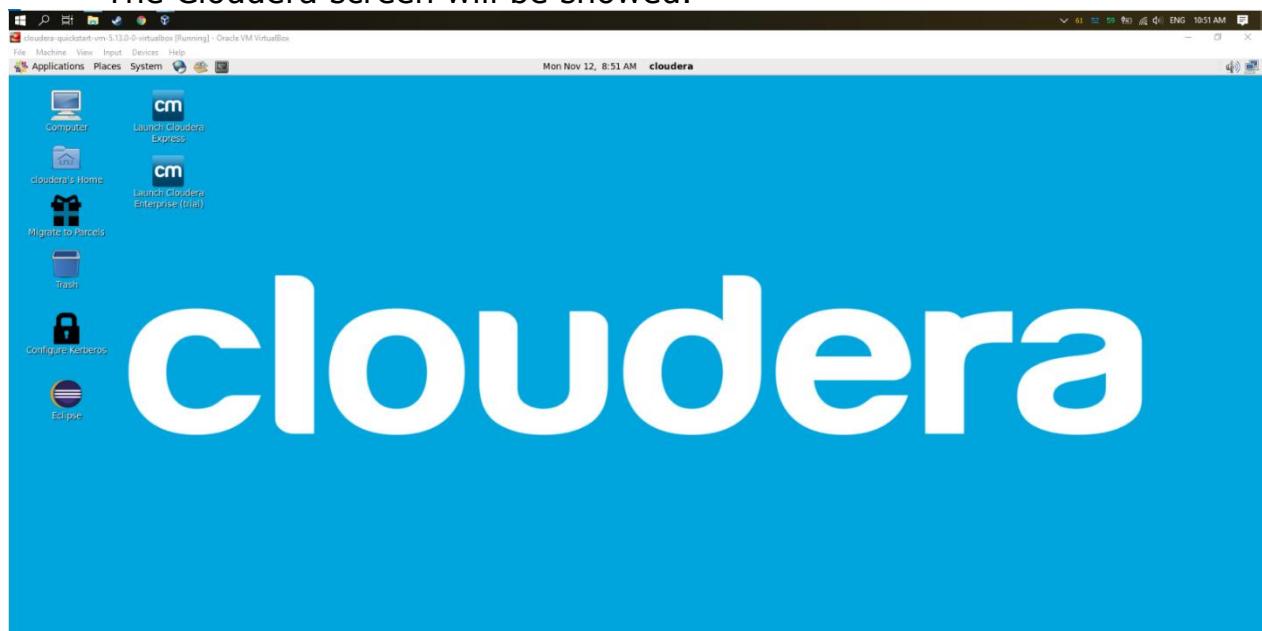
- Configure your setting. Minimum RAM requirement for this VM is 4GB. Then click on Import.



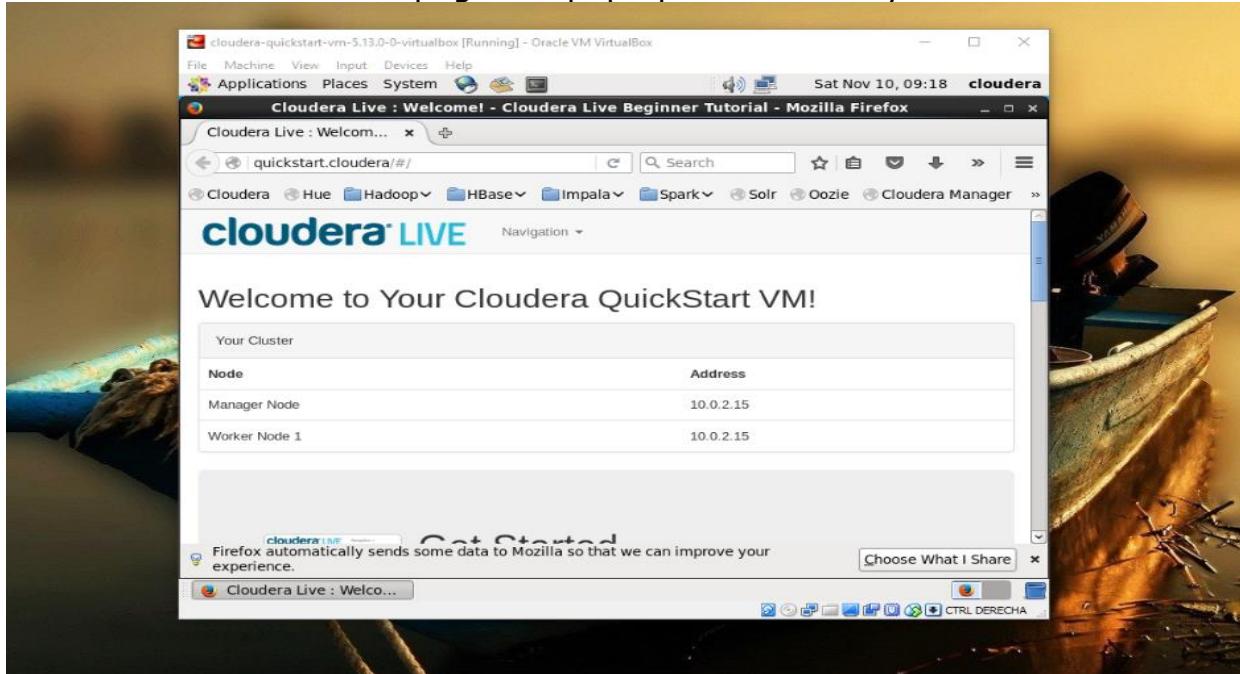
- After finished import, you will able to see your Virtual Machine. Click Start to start the VM.



- The Cloudera screen will be showed.



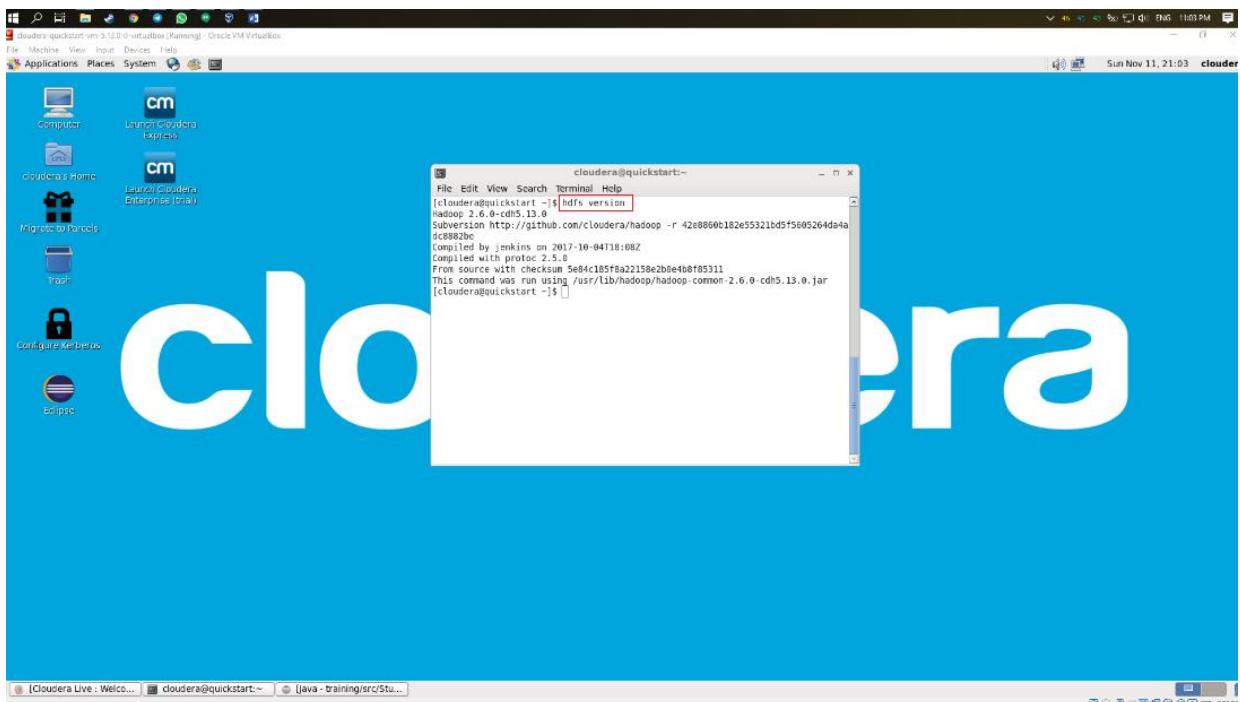
- Cloudera Welcome page will pop up automatically.



III. Get WordCount (Test run)

To start to work with WordCount, you will need to import Hadoop libraries to run WordCount.

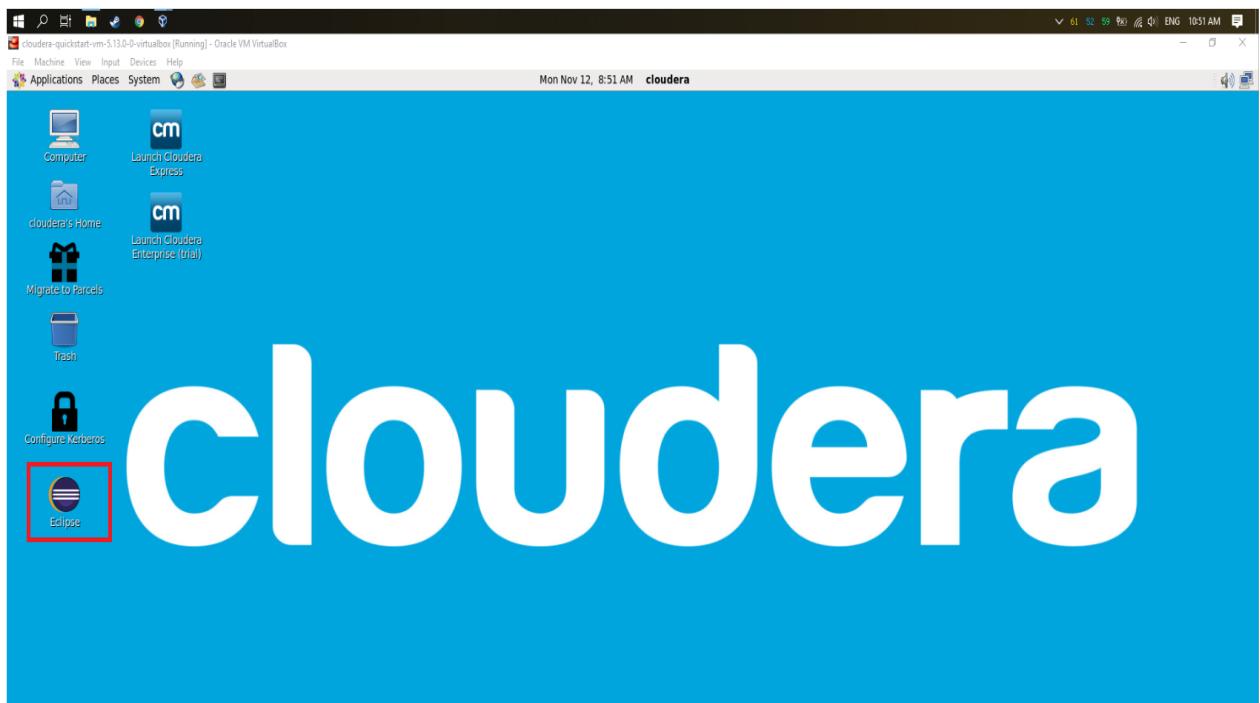
- You need to check your Hadoop version. Go to Terminal -> run command and type: "hdfs version".



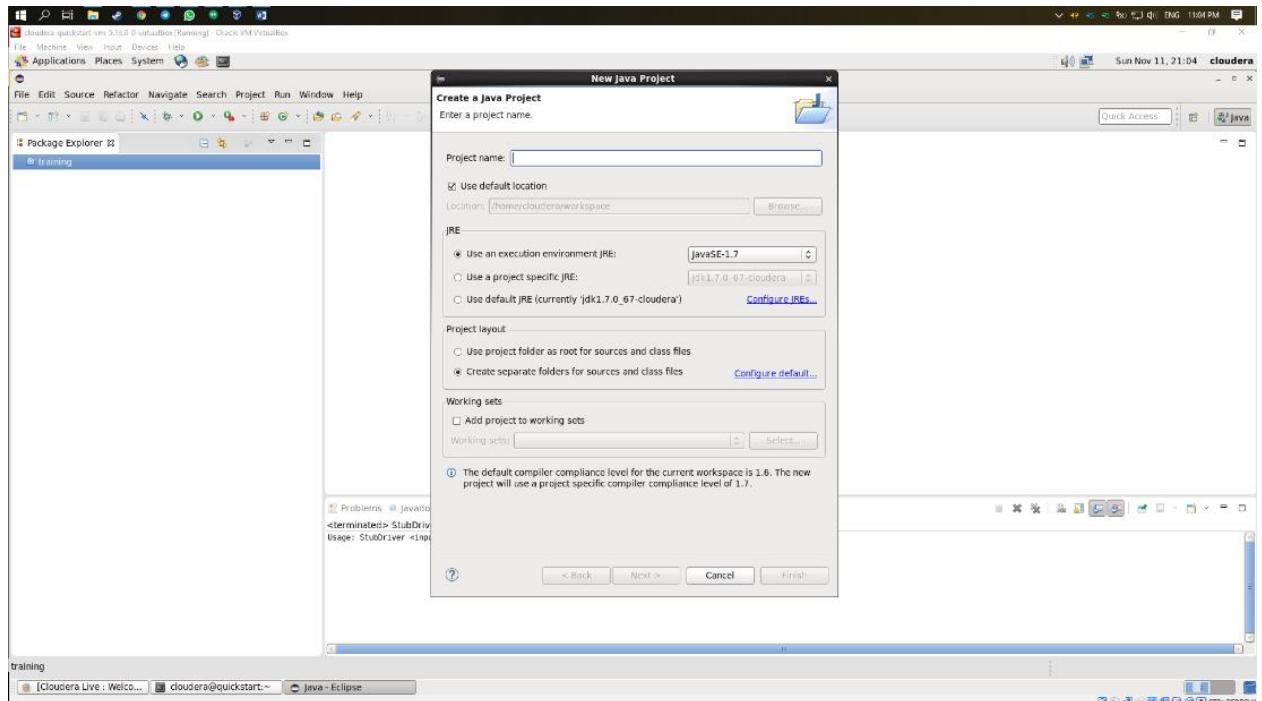
- Hadoop version is 2.6.0, so we will need to download libraries exactly as this version. It is necessary to do the download of Hadoop library from this website:

<http://mynrepository.com/artifact/org.apache.hadoop/hadoop-common/>

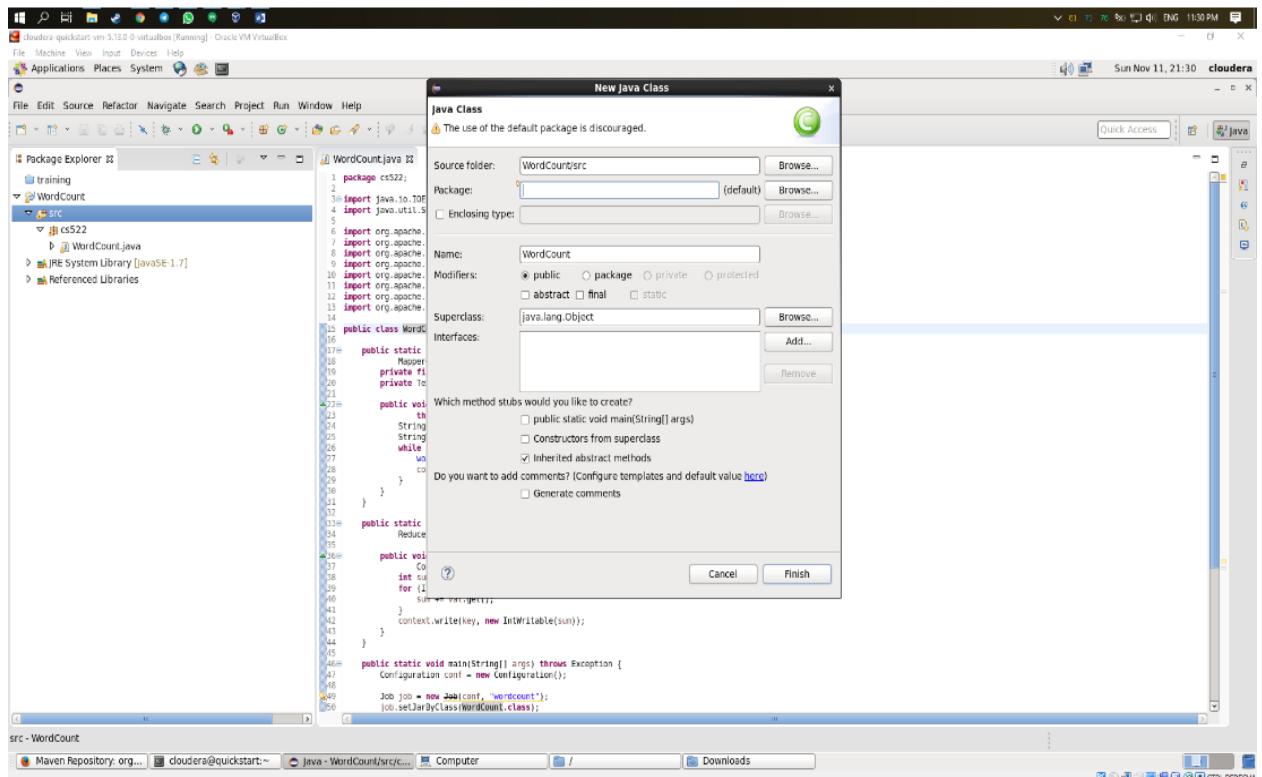
- You need to download the file and you will need it to import to Eclipse later. Eclipse is available on your VM machine.



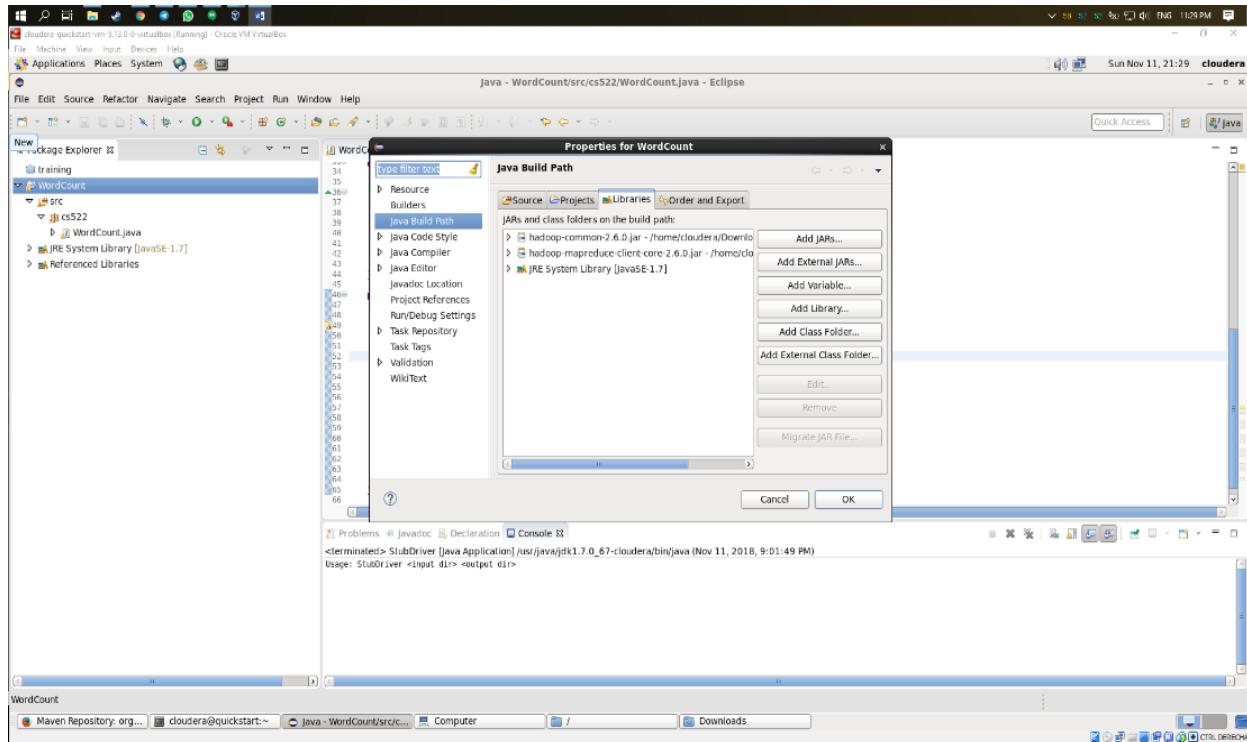
- Run the Eclipse IDE and then create Java Project



- Setup project name and JRE



- You must to add all required libraries



a) Running WordCount

We will show the execution of the algorithm WordCount, that count the words from an input source and return a count of each word.

- Algorithm(WordCount):

```
package cs522.pointC;

import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.fs.FileSystem;
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 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();

    FileSystem fs = FileSystem.get(conf);

    if(fs.exists(new Path(args[1]))){
        fs.delete(new Path(args[1]),true);
    }

    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);

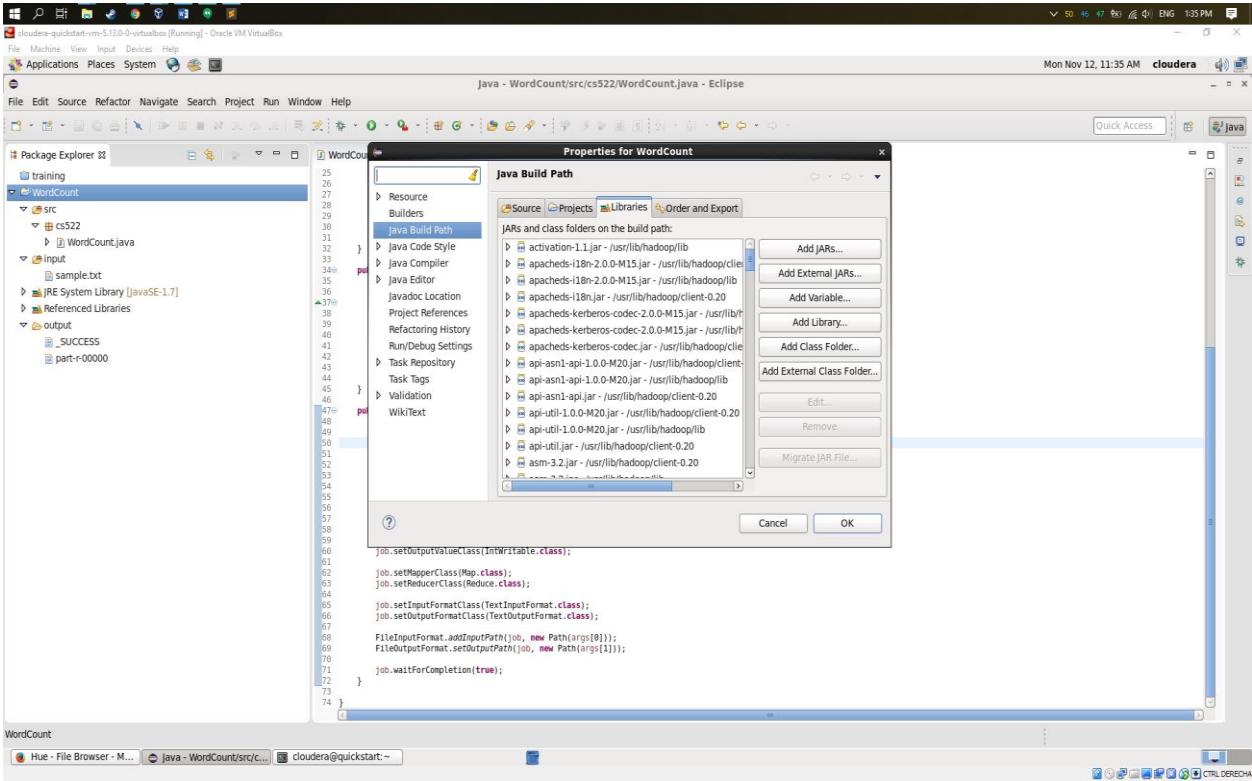
    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);

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

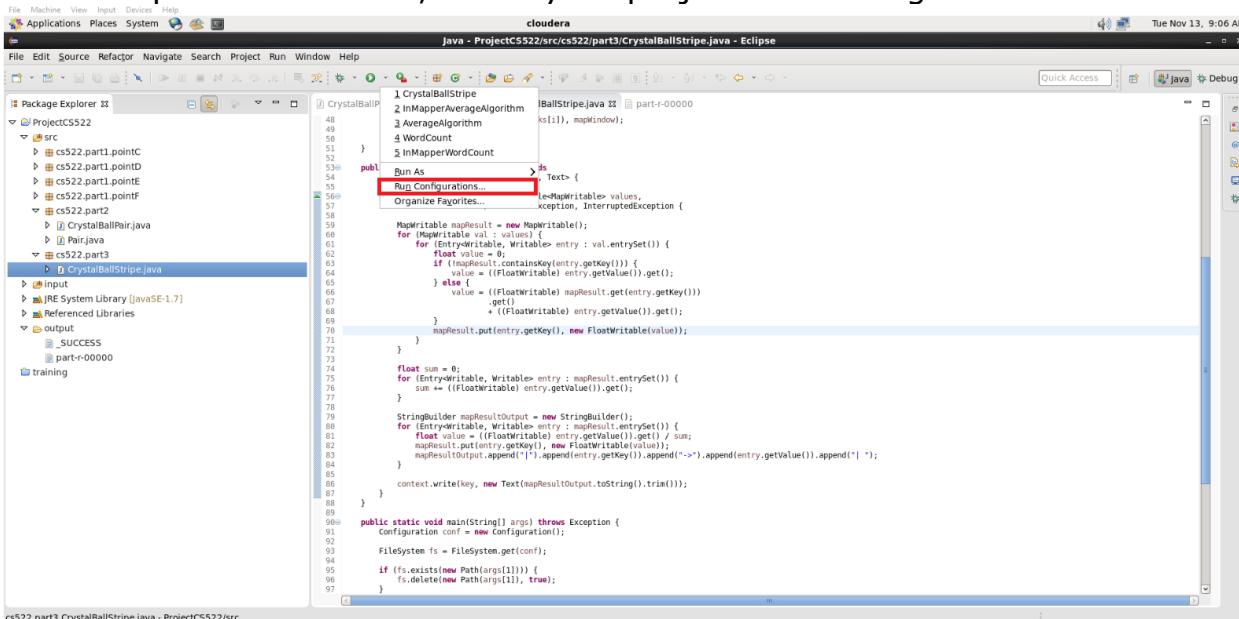
    job.waitForCompletion(true);
}
}

```

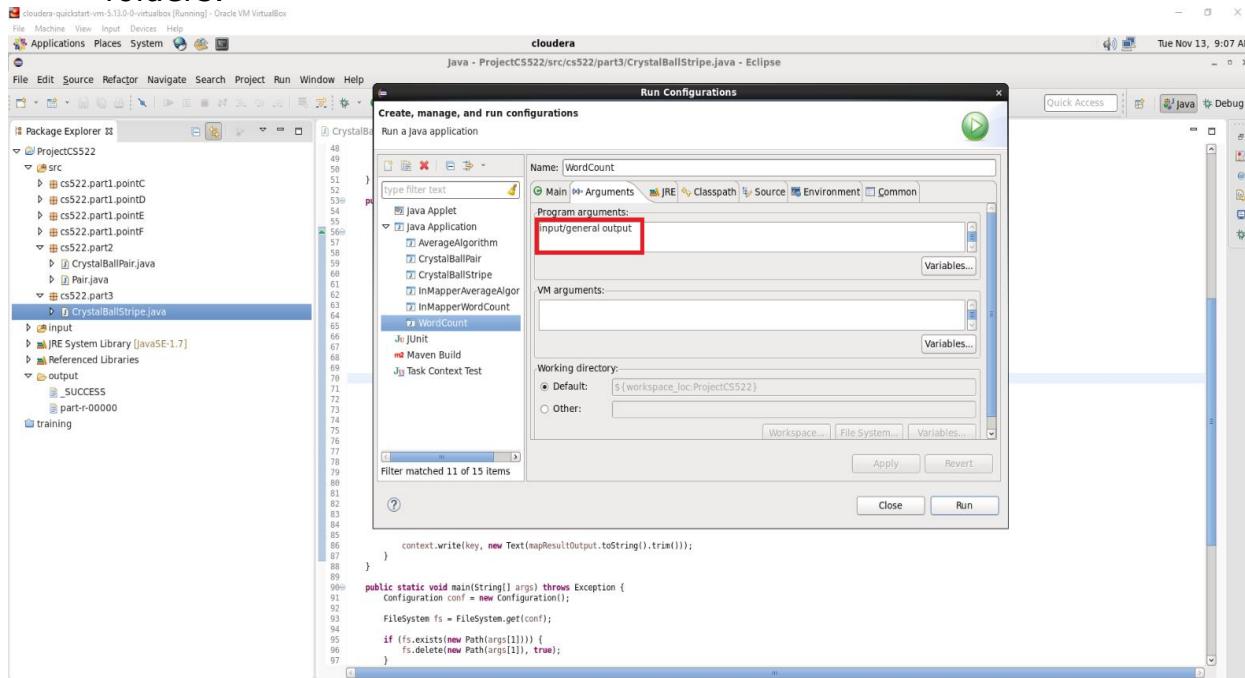
- We need to Add all required libraries as below
- File system/usr/lib/hadoop/client-0.20
 File system/usr/lib/hadoop
 File system/usr/lib/hadoop/lib



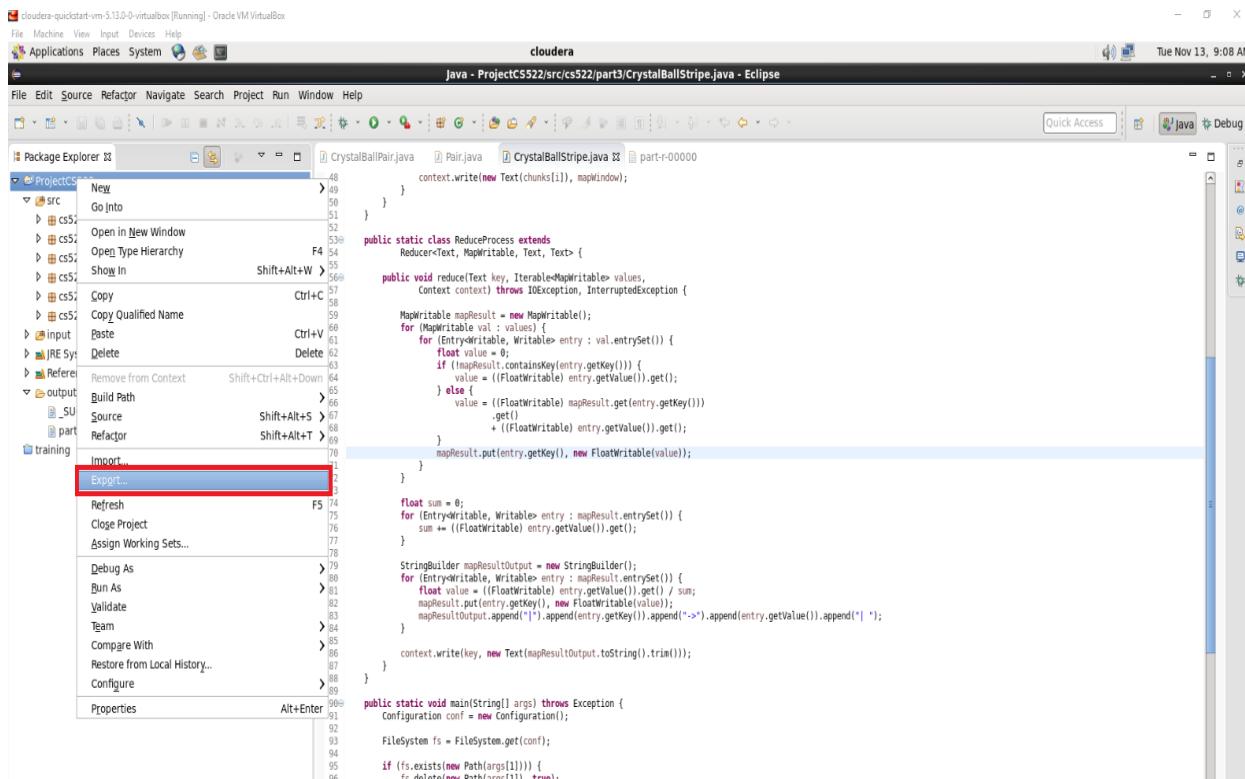
- To run the class "WordCount" you will need to setup the input and output folders. First, check your project Run Configuration:



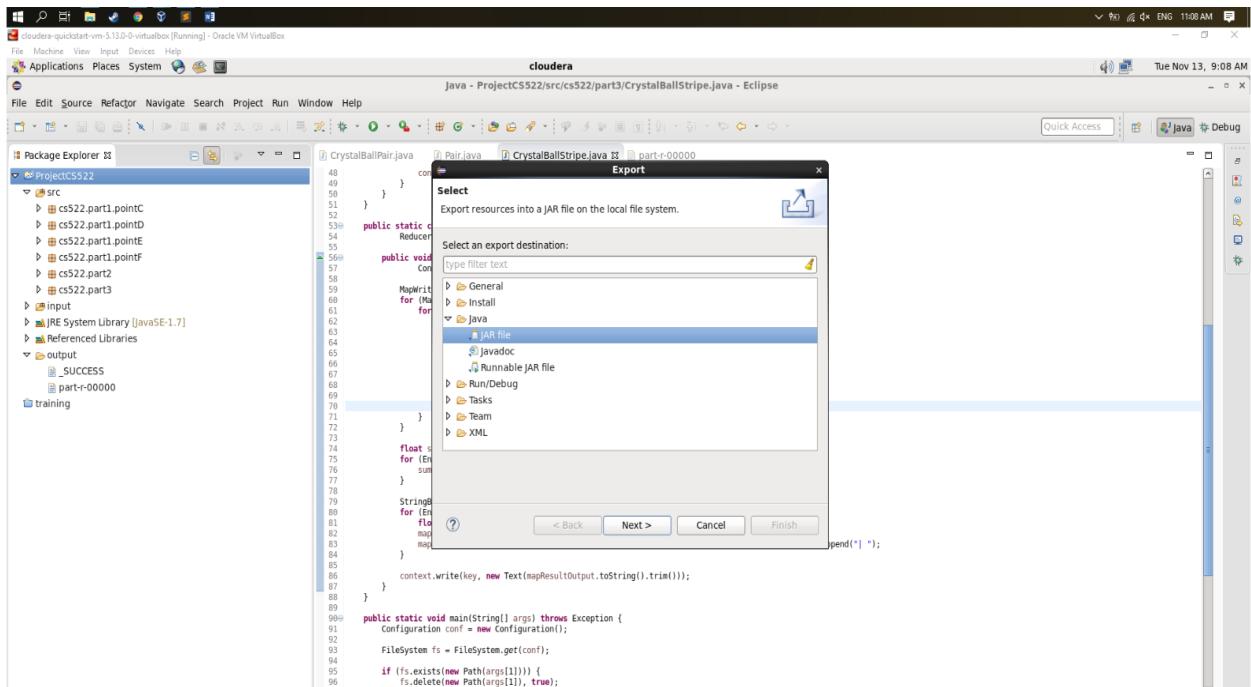
- You will need to configure the name/location to input and output folders.



- In Eclipse, right click on your class file -> Choose Export

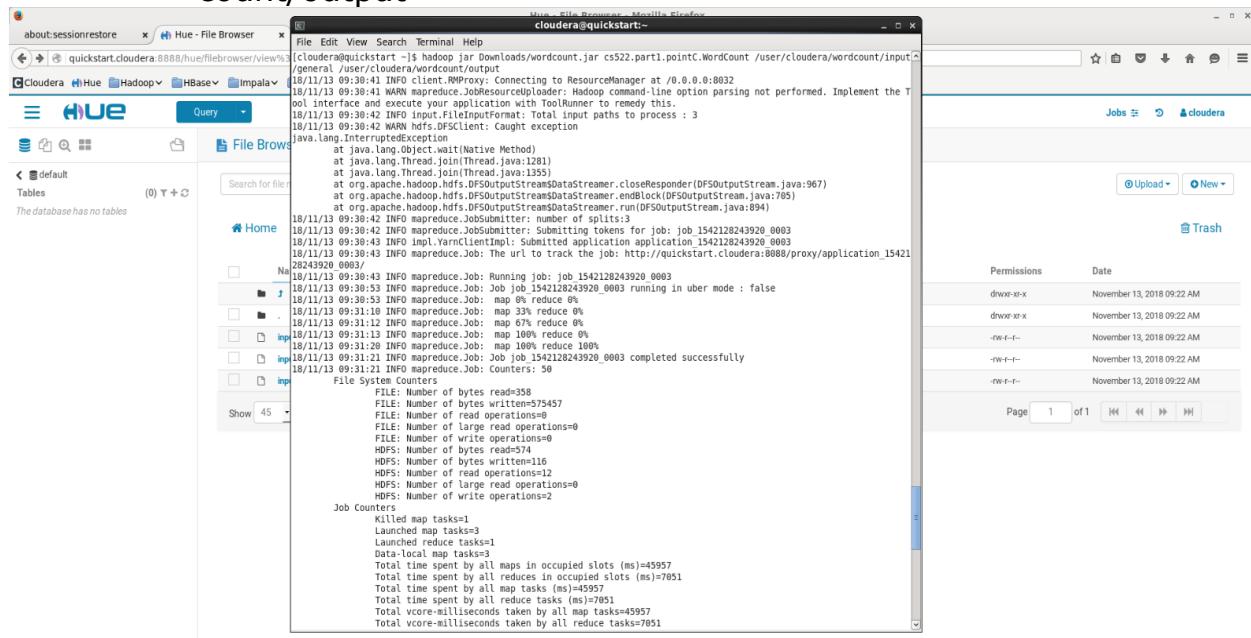


- Select the option: "Java" -> JAR file



- To run the WordCount application from JAR file, passing the paths to the input and output directories.

- Run command: `$ hadoop jar Downloads/wordcount.jar cs522.part1.pointC.WordCount /user/cloudera/wordcount/input/general/user/cloudera/wordcount/output`



- You can see the output results:

The image shows three separate terminal windows, each displaying the output of a Hadoop wordcount job. The output lists words and their counts, such as bat (2), cat (3), cs (1), eat (2), edu (1), fat (1), hat (3), jat (1), kat (1), lat (1), mat (3), mun (2), and mun.cs (1). The terminals are part of a Hue interface, which includes a file browser and a jobs page.

```

Total vcore-milliseconds taken by all reduce tasks=7051
Total megabyte-milliseconds taken by all map tasks=47059968
Total megabyte-milliseconds taken by all reduce tasks=7220224
Map-Reduce Framework
  Map input records=6
  Map output records=35
  Map output bytes=282
  Map output materialized bytes=370
  Input split bytes=435
  Combine input records=0
  Combine output records=0
  Reduce input groups=19
  Reduce shuffle bytes=370
  Reduce input records=35
  Reduce output records=19
  Spilled Records=70
  Shuffled Maps =3
  Failed Shuffles=0
  Merged Map outputs=3
  GC time elapsed (ms)=386
  CPU time spent (ms)=2440
  Physical memory (bytes) snapshot=940625920
  Virtual memory (bytes) snapshot=6236090368
  Total committed heap usage (bytes)=791674880
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=139
File Output Format Counters
  Bytes Written=116
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/wordcount/output/*
bat 2
cat 3
cs 1
eat 2
edu 1
fat 1
hat 3
jat 1
kat 1
lat 1
mat 3
mun 2
mun.cs 1
oat 2

```

- You also can check the output file on the web browser

