**Maharishi University of Management**

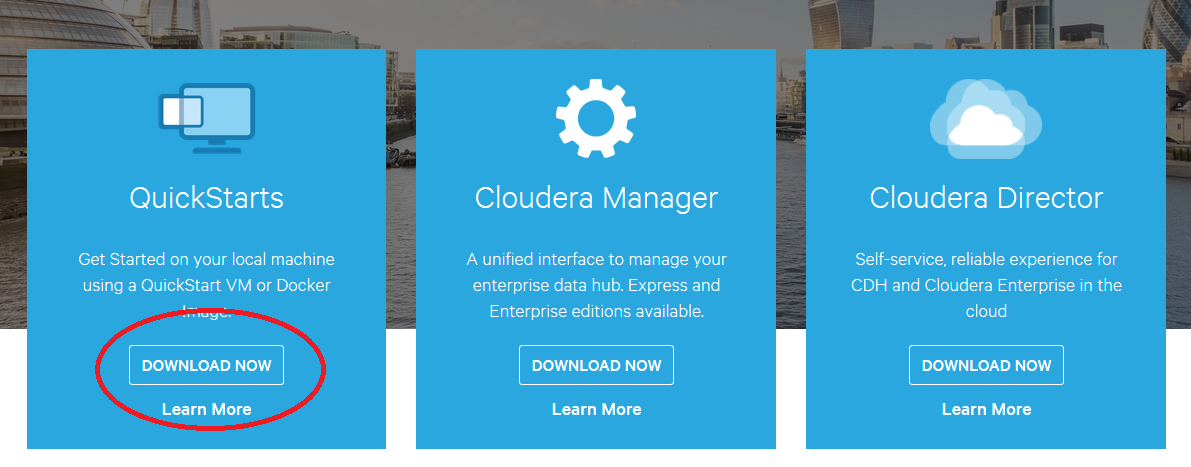
**CS522 – Big Data**

**Dr. Premchand Nair**

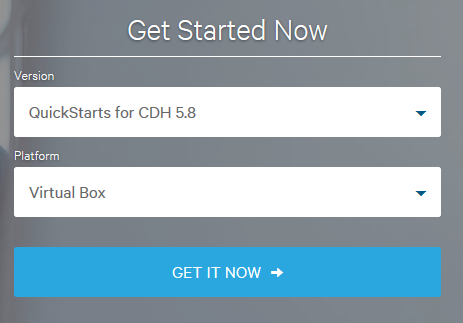
**Set Up a Single Node Cluster**

1. **Download Cloudera**

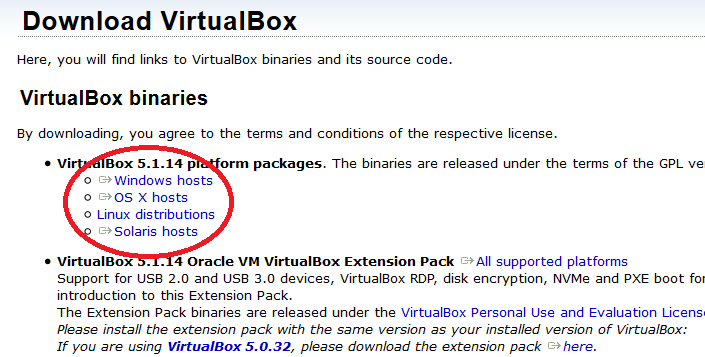
* Go to <http://www.cloudera.com/downloads.html>

****

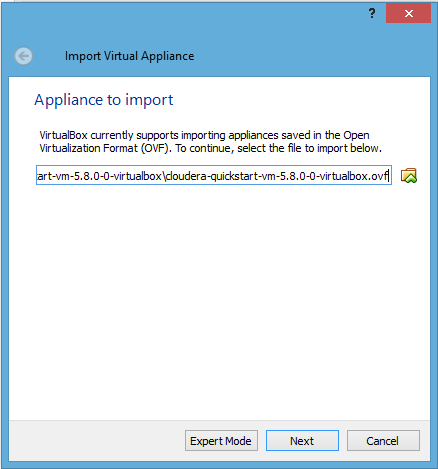
* Choose version and platform as image below

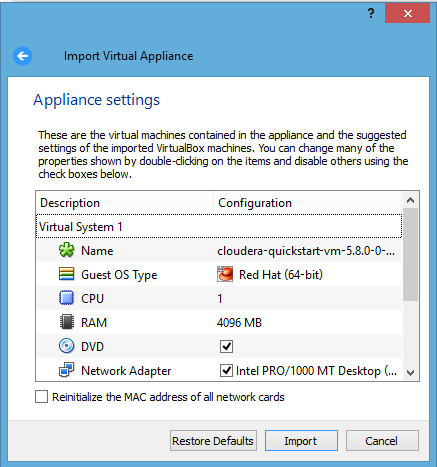
****

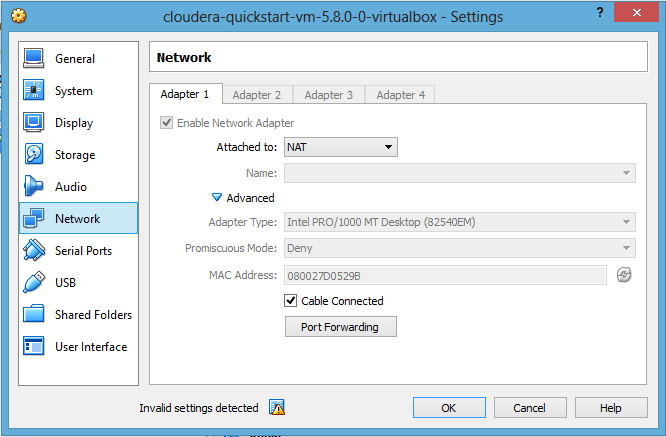
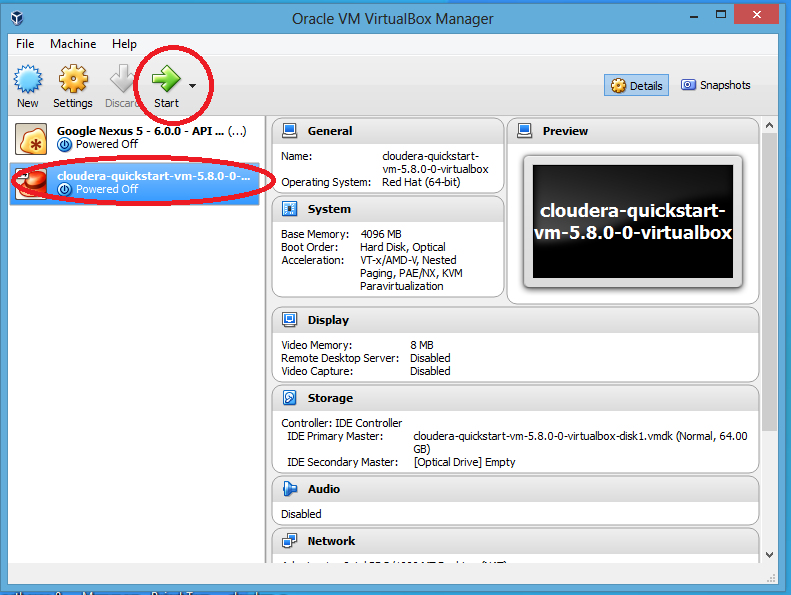
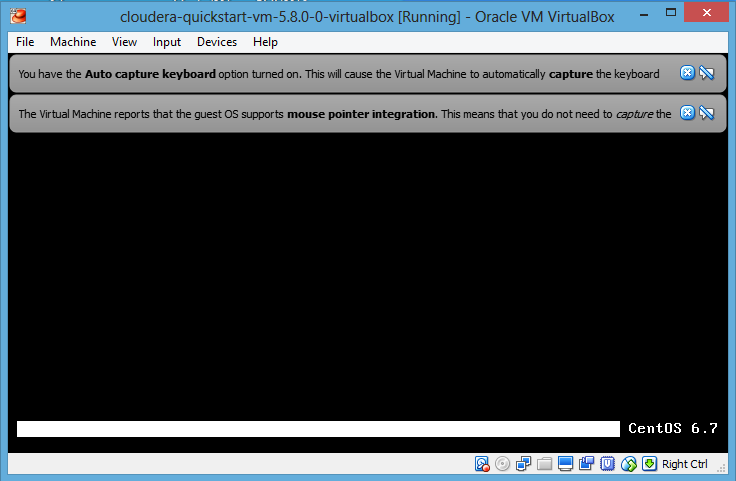
* **Download Virtual Box from** [**https://www.virtualbox.org/wiki/Downloads**](https://www.virtualbox.org/wiki/Downloads)

****

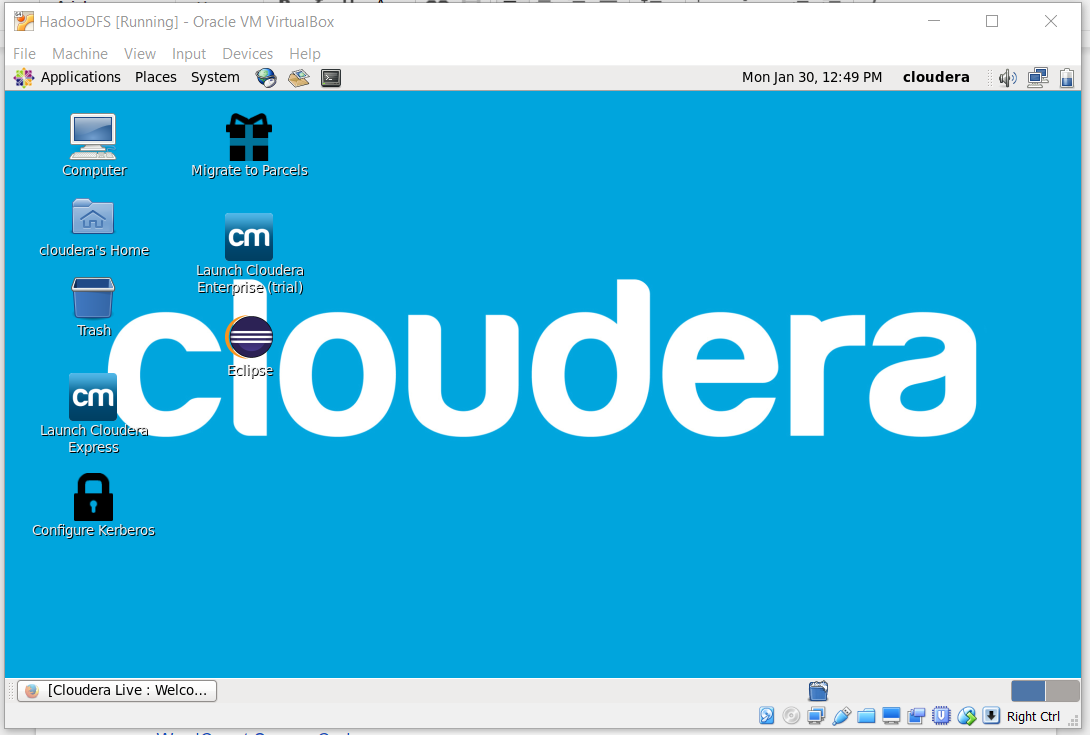
* **Click on File > Import Appliance**

****

****

* **To enable network access go to Machine > Settings > Network and choose NAT in Attached to option as below**
* **Now start cloudera by clicking start button**
* 

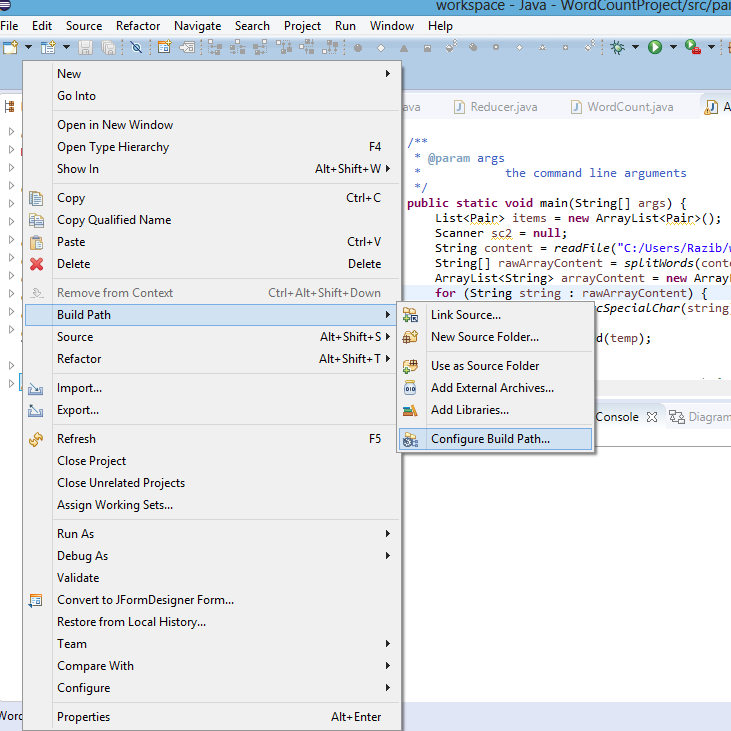
1. **Desktop will look like below**

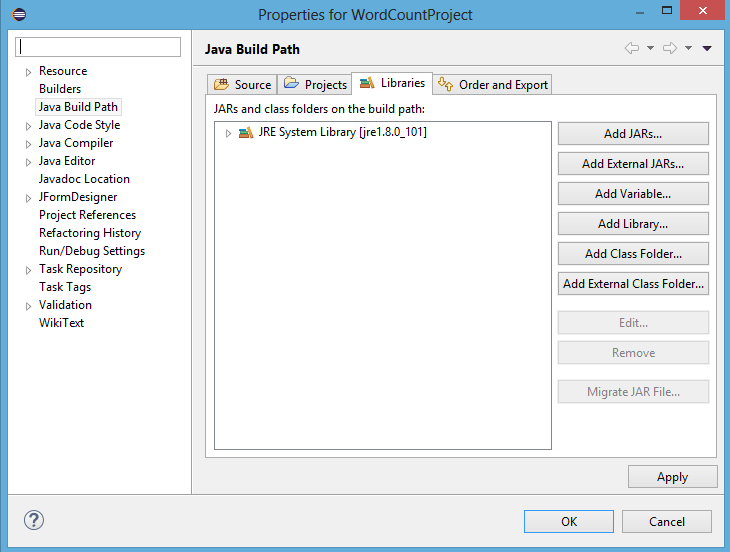


* **Open eclipse and start new project**
* **Download hadoop common and hadoop mapreduce client core from below link**

<http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-common/2.7.3>  
<http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapreduce-client-core/2.7.3>

* **Add jar file to project**

****

****

You must add all the libraries inside the following paths:

**File system/usr/lib/hadoop/client-0.20**

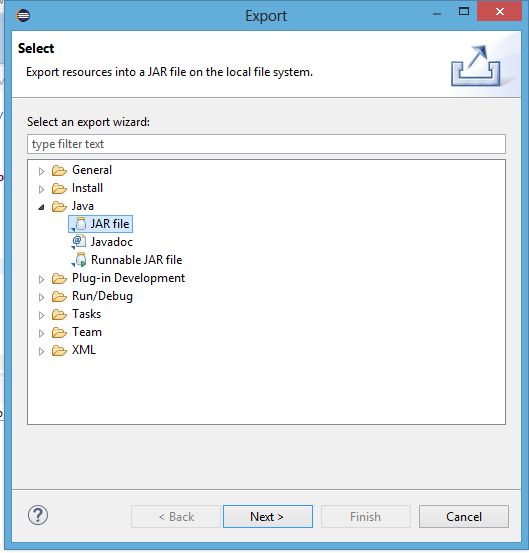
**File system/usr/lib/hadoop**

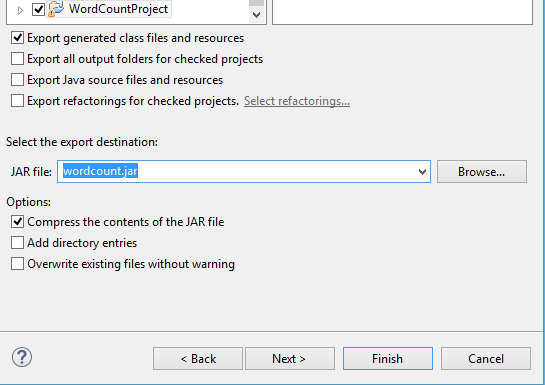
**File system/usr/lib/hadoop/lib**

* **Get WordCount example from** **<https://wiki.apache.org/hadoop/WordCount>**
* **AFTER THIS LINE IN THE CODE** Job job = new Job(conf, "wordcount"); **ADD**

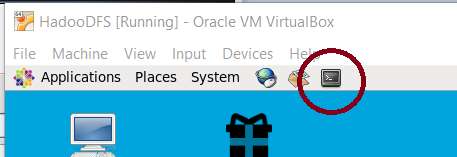
job.setJarByClass(WordCount.class);

**Export to WordCount.jar to test the WordCount** example**. Right click on Project, choose “Export”. The Export window will appear, choose Java -> JAR file. Click Next.**

****

****

* **Open terminal program**



* **Open Terminal and first create input and output locations in HDFS. Using the following commands:**

$ sudo su hdfs

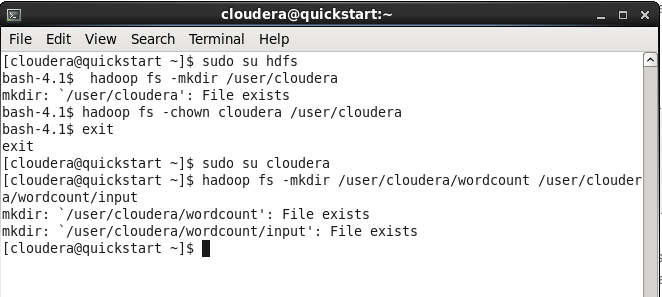
$ hadoop fs -mkdir /user/cloudera

$ hadoop fs -chown cloudera /user/cloudera

$ exit

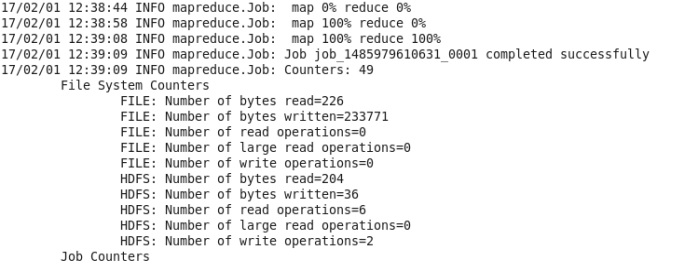
$ sudo su cloudera

$ hadoop fs -mkdir /user/cloudera/wordcount /user/cloudera/wordcount/input

****

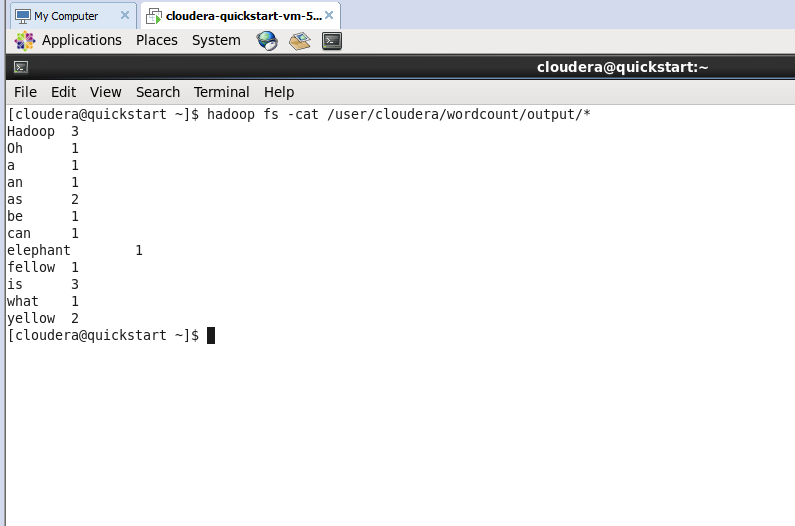
Run the WordCount application from the JAR file, given the paths to the input and output directories in HDFS.

hadoop jar wordcount.jar org.myorg.WordCount /user/cloudera/wordcount/input /user/cloudera/wordcount/output



**In order to see the output use the following command which will be as follows or you can see from the browser by downloading the output file.**

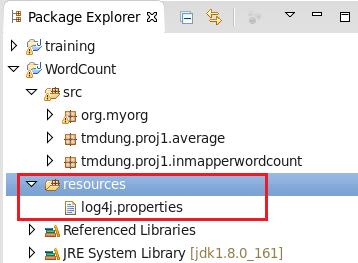
**hadoop fs -cat /user/cloudera/wordcount/output/\***

****

**Hadoop setup is complete. Continue only if you want to debug Hadoop.**

**Add logging for Hadoop**

1. Hadoop uses log4j by default. Firsts step is to configure its settings:
   1. Add **resources** folder to your project. Then, add **log4j.properties** file to that folder



* 1. Open log4j.properties file and add the following settings:

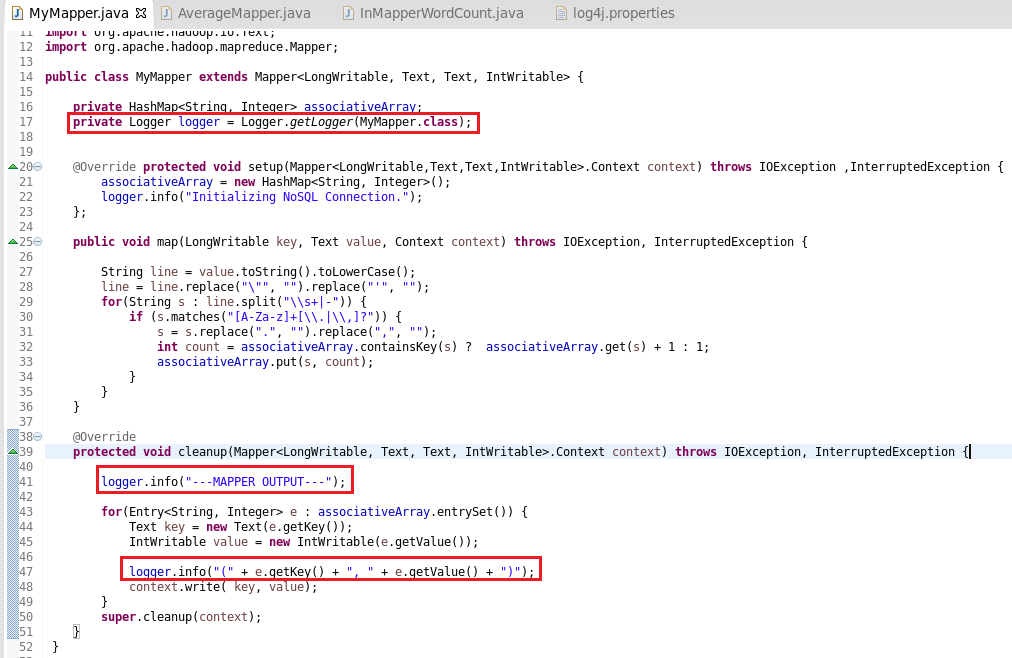
log4j.rootLogger=DEBUG, CA

log4j.appender.CA=org.apache.log4j.ConsoleAppender

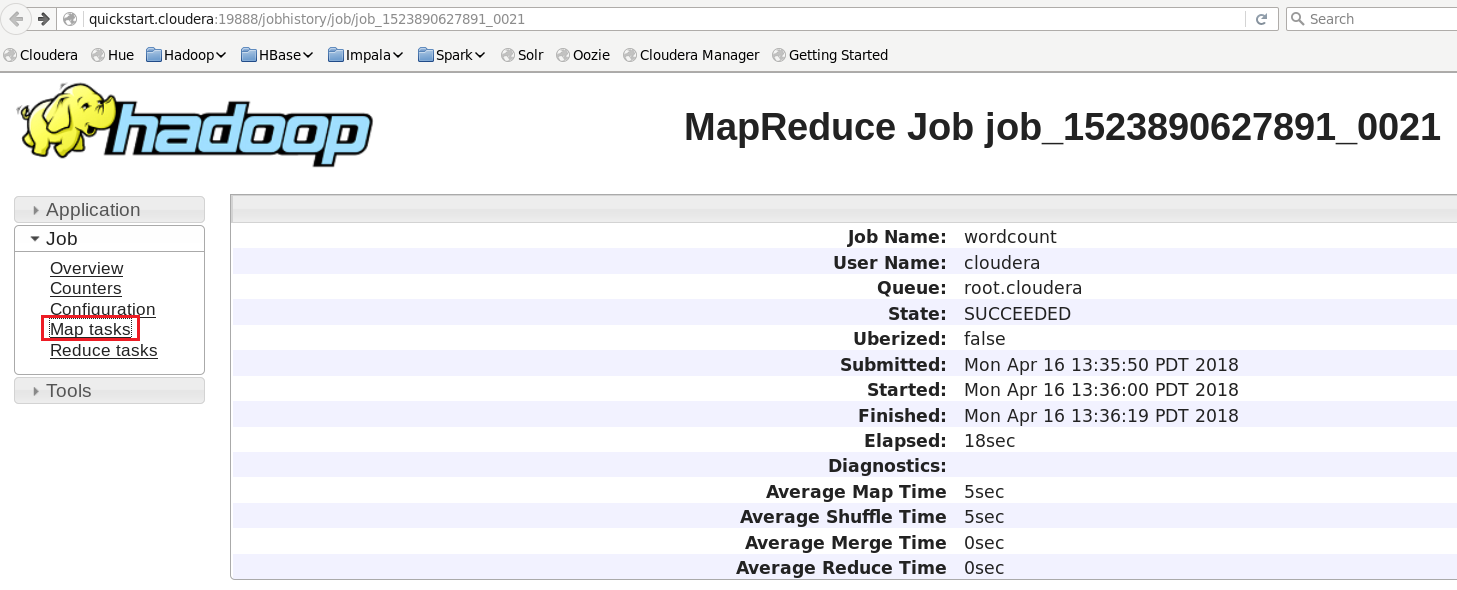
log4j.appender.CA.layout=org.apache.log4j.PatternLayout

log4j.appender.CA.layout.ConversionPattern=%-4r [%t] %-5p %c %x - %m%n

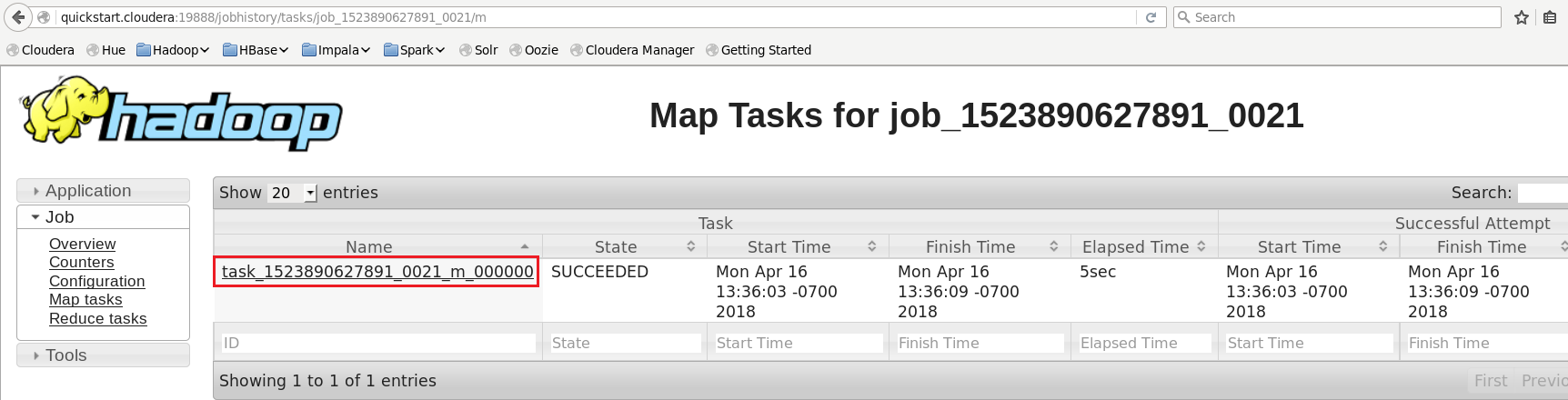
1. Add code to log the info we need:
   1. Create Logger instance:
      1. private Logger logger = Logger.getLogger(MyMapper.class);
   2. Use respective log methods for your purpose:
      1. logger.info("Log information");
      2. logger.error("Log some error");
      3. logger.debug("Debug information");
2. Sample:



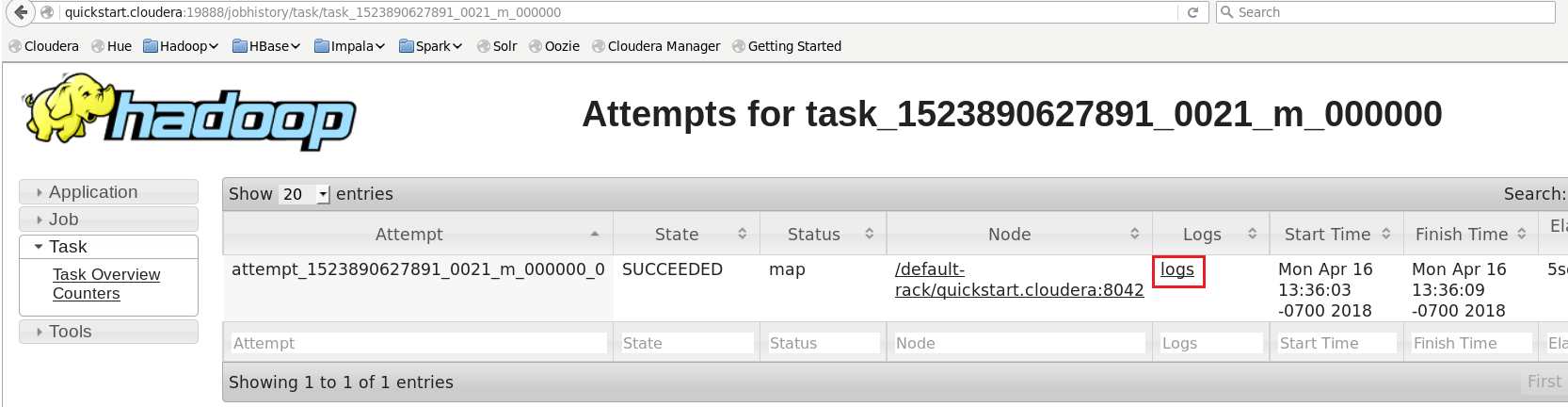
1. View logs:
   1. The logger will by default write log to Hadoop tracker system’s log file.
   2. One way to see it is open [http://quickstart.cloudera:8088/proxy/<application\_ID>/](http://quickstart.cloudera:8088/proxy/%3capplication_ID%3e/) in browser
   3. Select Map Tasks on left menu



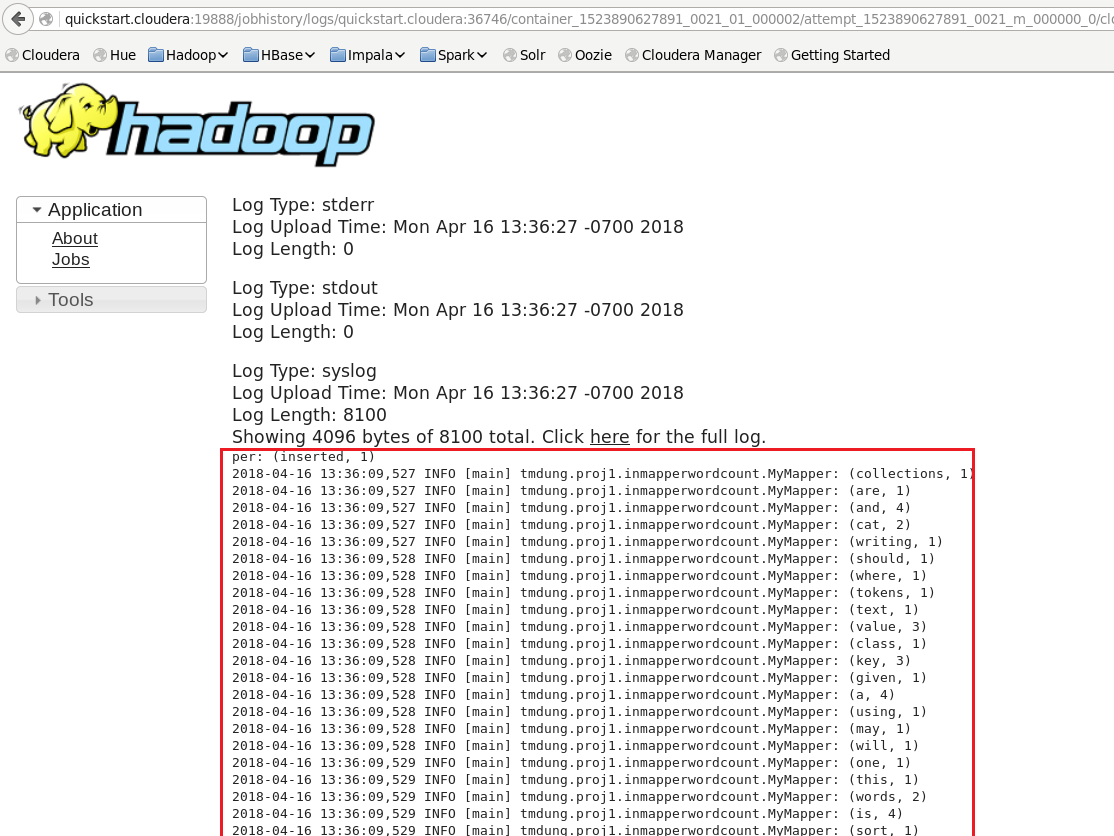
* 1. Select on the task link:



* 1. Select logs:



* 1. You can see the content of logging in syslog section:



End

**Running multiple MapReduce tasks**

Option 1: <https://coe4bd.github.io/HadoopHowTo/multipleJobsSingle/multipleJobsSingle.html>

Option 2:

Use Oozie http://oozie.apache.org/