

Laxman Dutt Degala | CS560 | February 28, 2014

Lab5 Assignment

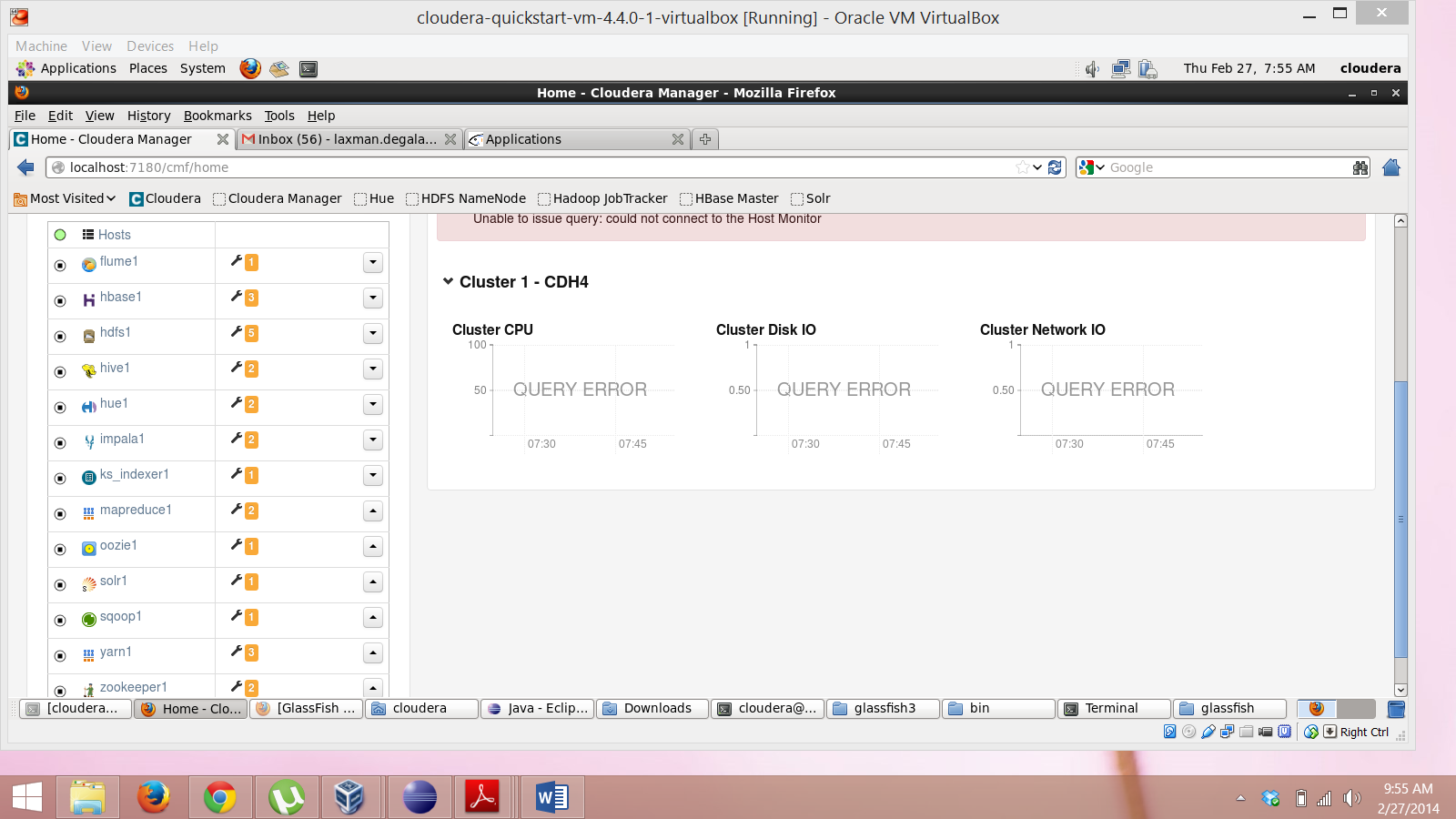
**Contents:**

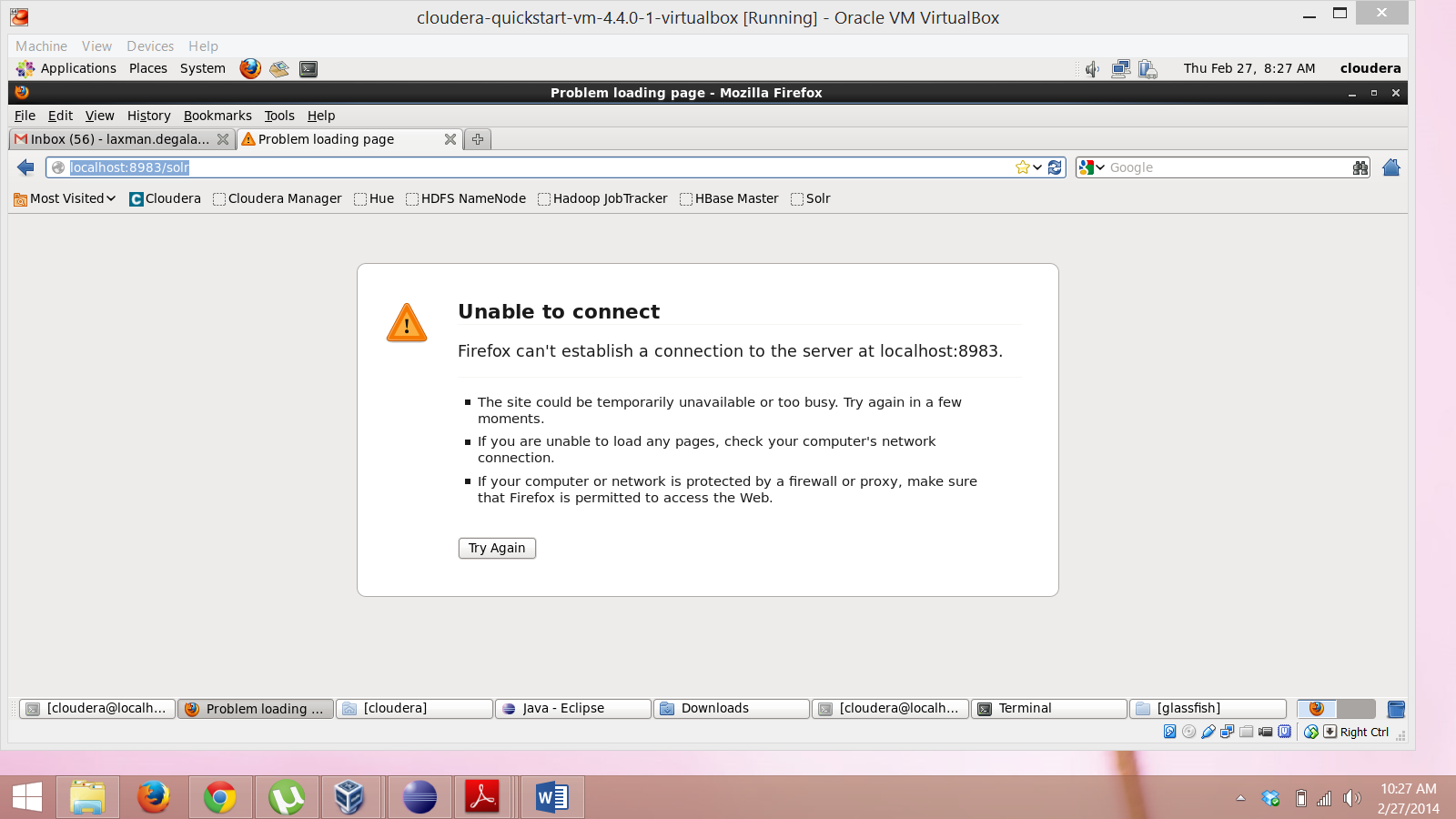
This document includes description about

1. SOLR installation on Image
2. Installation of Glassfish server
3. Preparation of data for Running Mahout Application using Classifier
4. Taking the output in .txt format and converting it to JSON
5. Pushing the JSON data to SOLR
6. Retrieving data from SOLR and popping on the application screen
7. Issues faced while working on the task
8. Step by Step process and commands for running the process
9. Lessons Learnt
10. **SOLR Installation on Cloudera Image**:

There are instances where we have SOLR server comes as built-in feature along with Cloudera image. If it comes as inbuilt, we need not install SOLR server again by going to the *Parcels*. If it’s downloaded on top of built in SOLR application, it would stop working and also all the other applications including HDFS and Zookeeper would also stop running.

Resolution: Go back to the Parcels again and deactivate the SOLR and restart the Cloudera Manager, Everything would get back to normal.





This issue has been resolved after deactivating the SOLR in *Parcels.*

1. **Installation of Glassfish server**

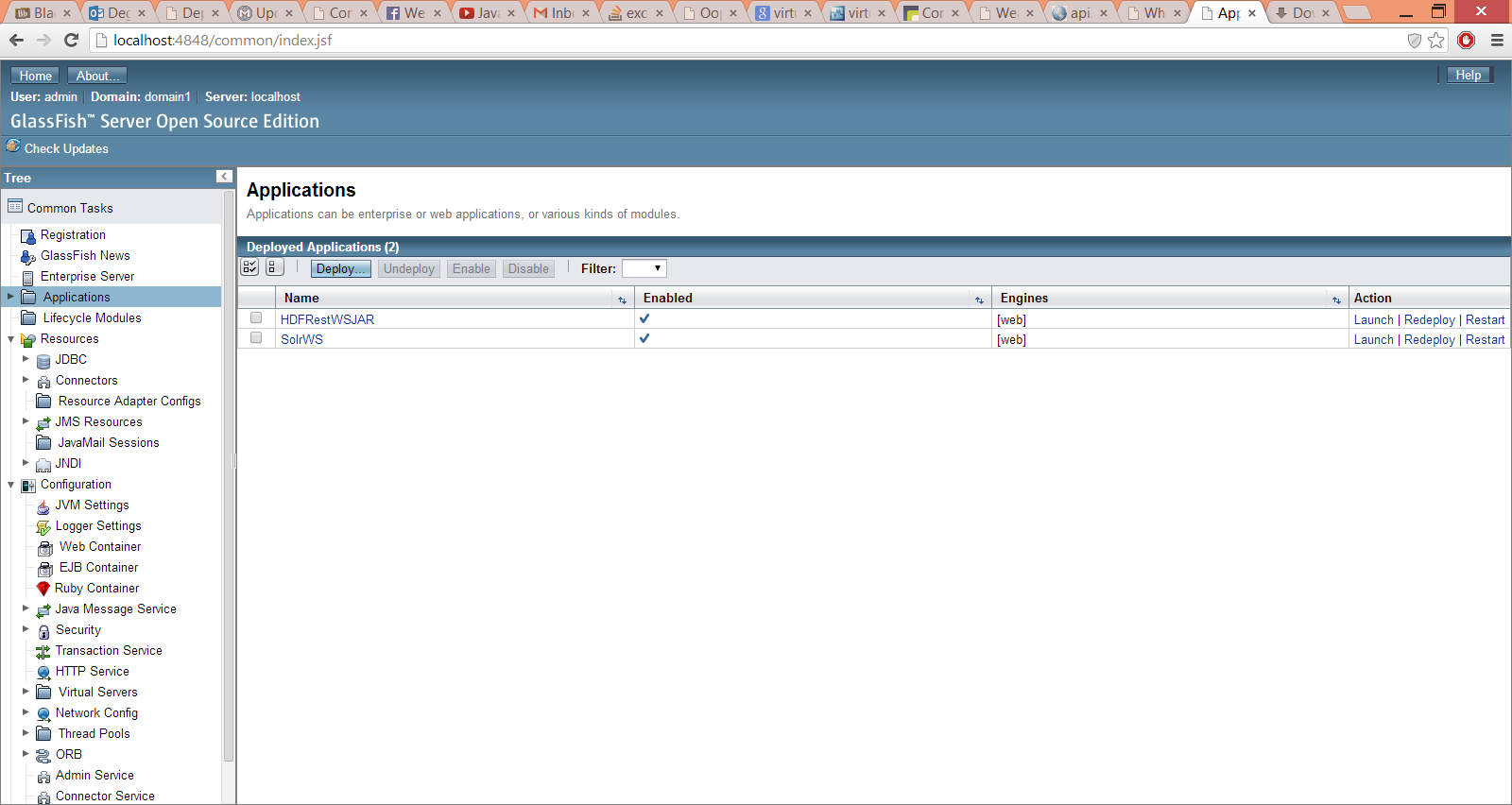
Have faced serious problems when installing Glassfish server. Installation wouldn’t run successfully or would abort in the middle.

Resolution: Issue has been resolved when uninstalled JDK 1.7 and installed JDK 1.6. Got to know that JDK 1.7 is not compatible for few features.

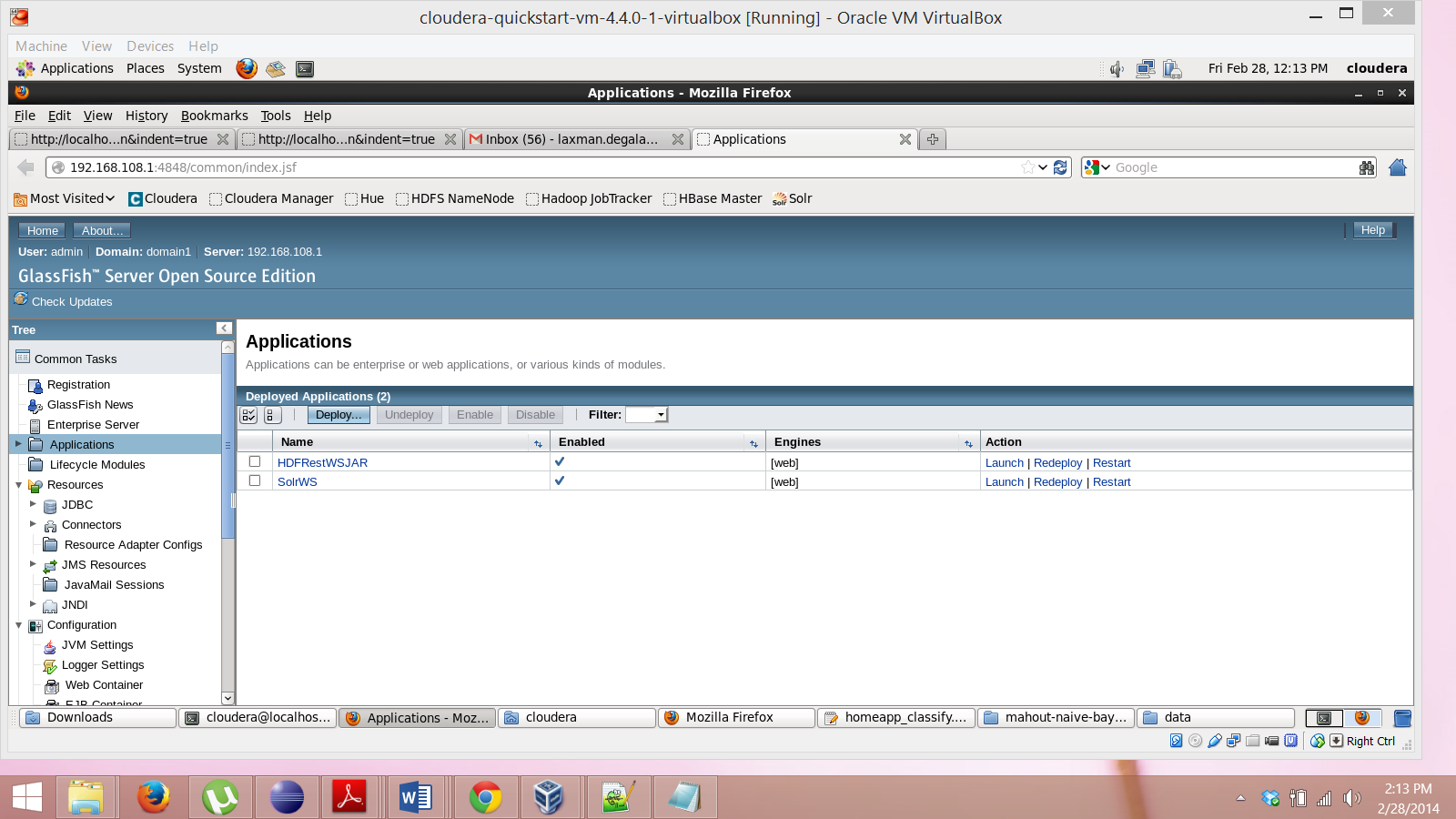
Please find below the attached snapshots of Glassfish server in Local and Remote machines:

You can see both the projects installed in Glassfish server in Local machine are deployed in the remote.

Glassfish server in Windows (Local Machine):

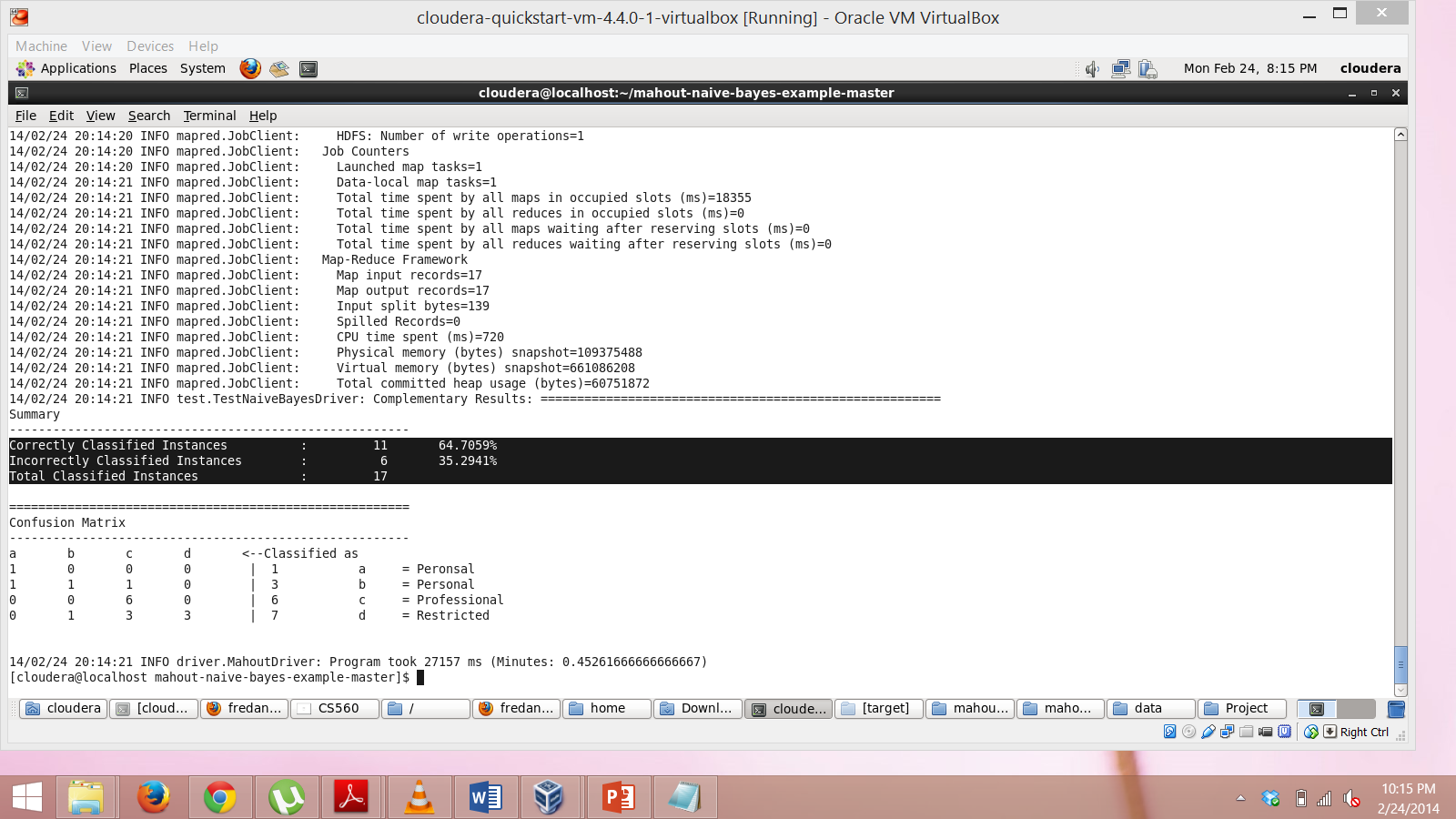


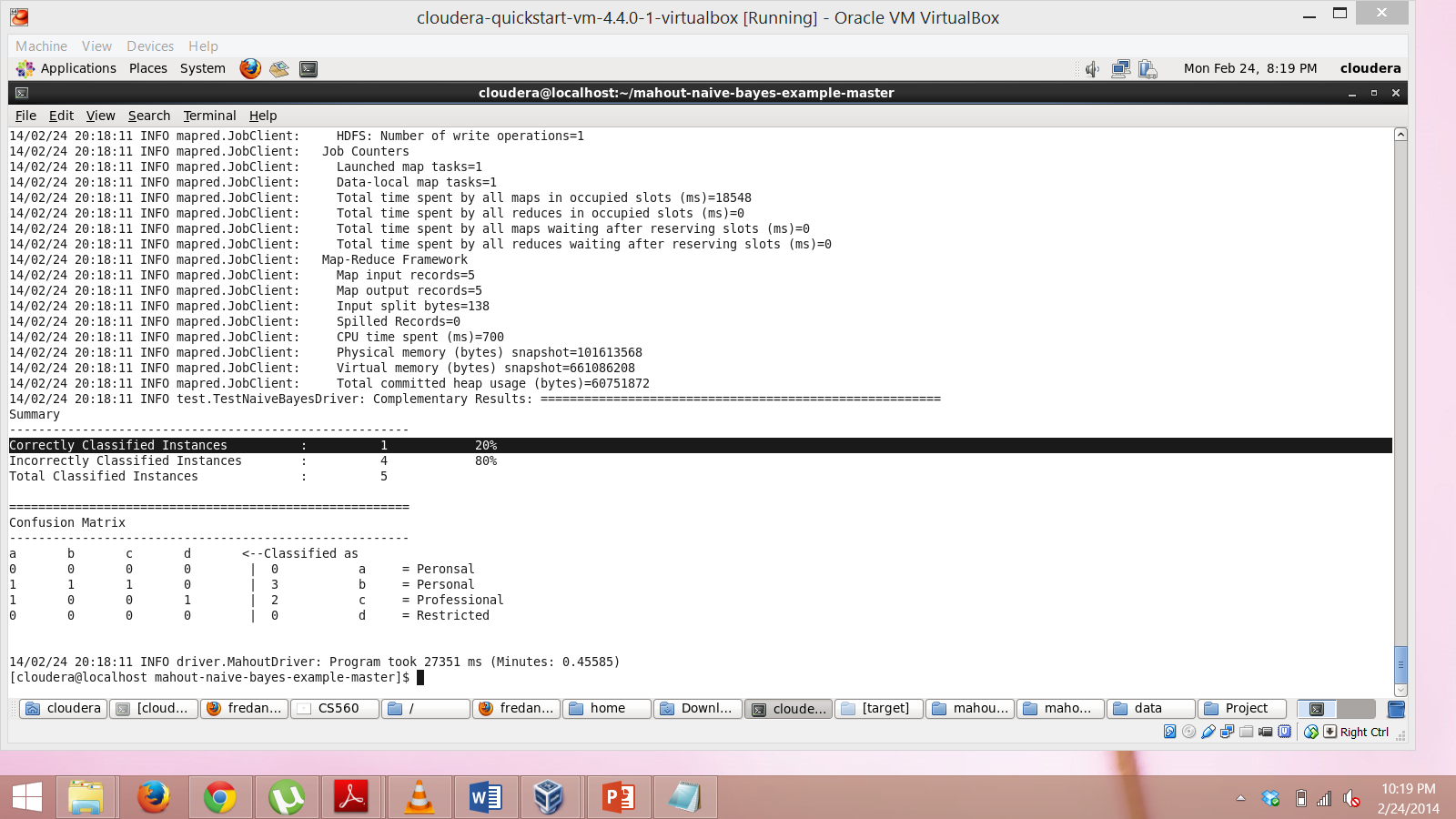
Glassfish server in Remote machine, when run using the IPAddressOfLocalMachine:4848.



**3) Train and Test data for running Mahout application:**

It took 6 to 8 iterations for creating the data. Initial versions of data have given an accuracy of 65% with Train data and 20% with Test data. Please find the snapshots below.

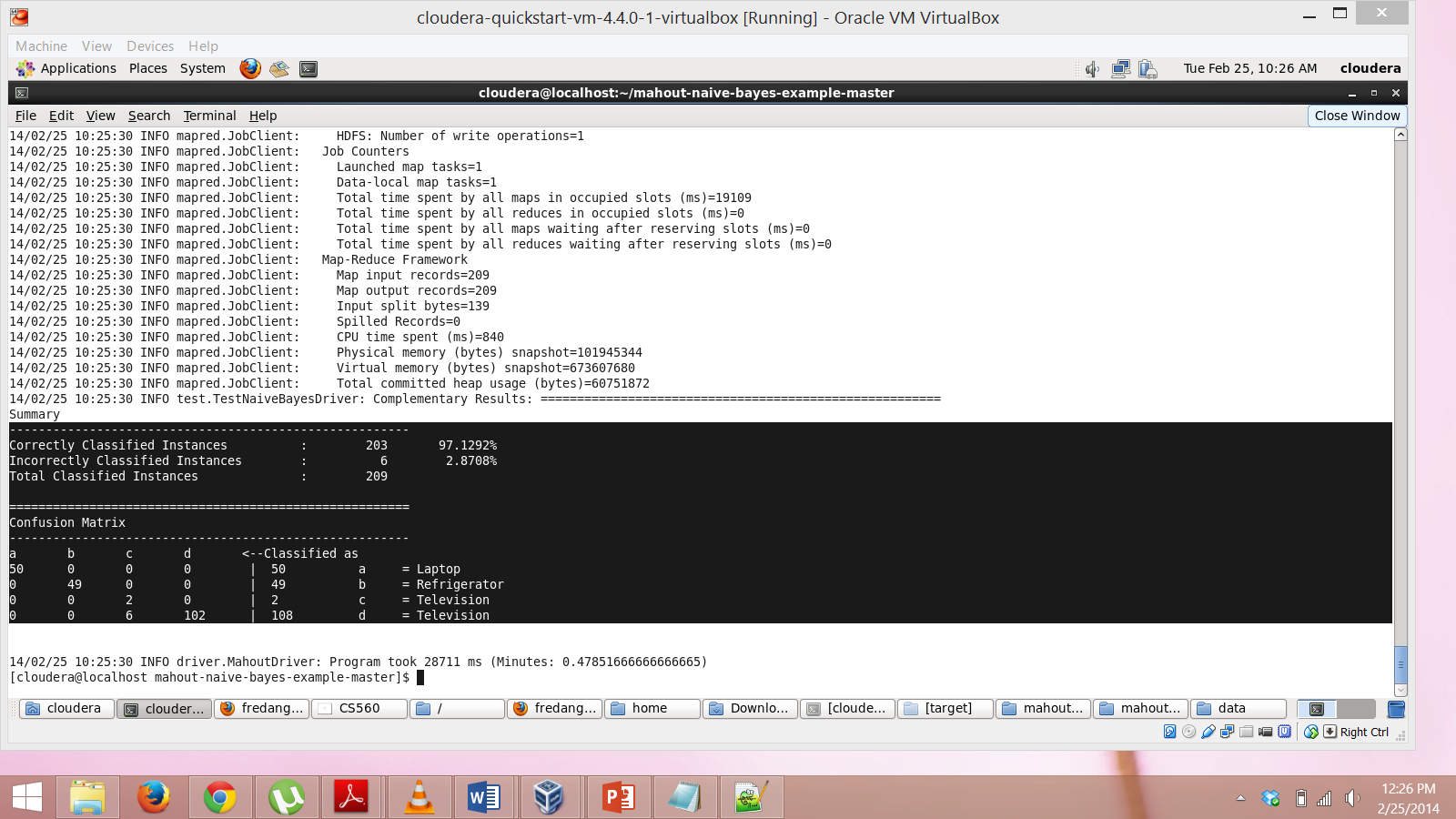
Train data: 65%

Test data: 20% accuracy 

Final version – after 6 iterations:

Checking if the classifier is working fine for Train data:

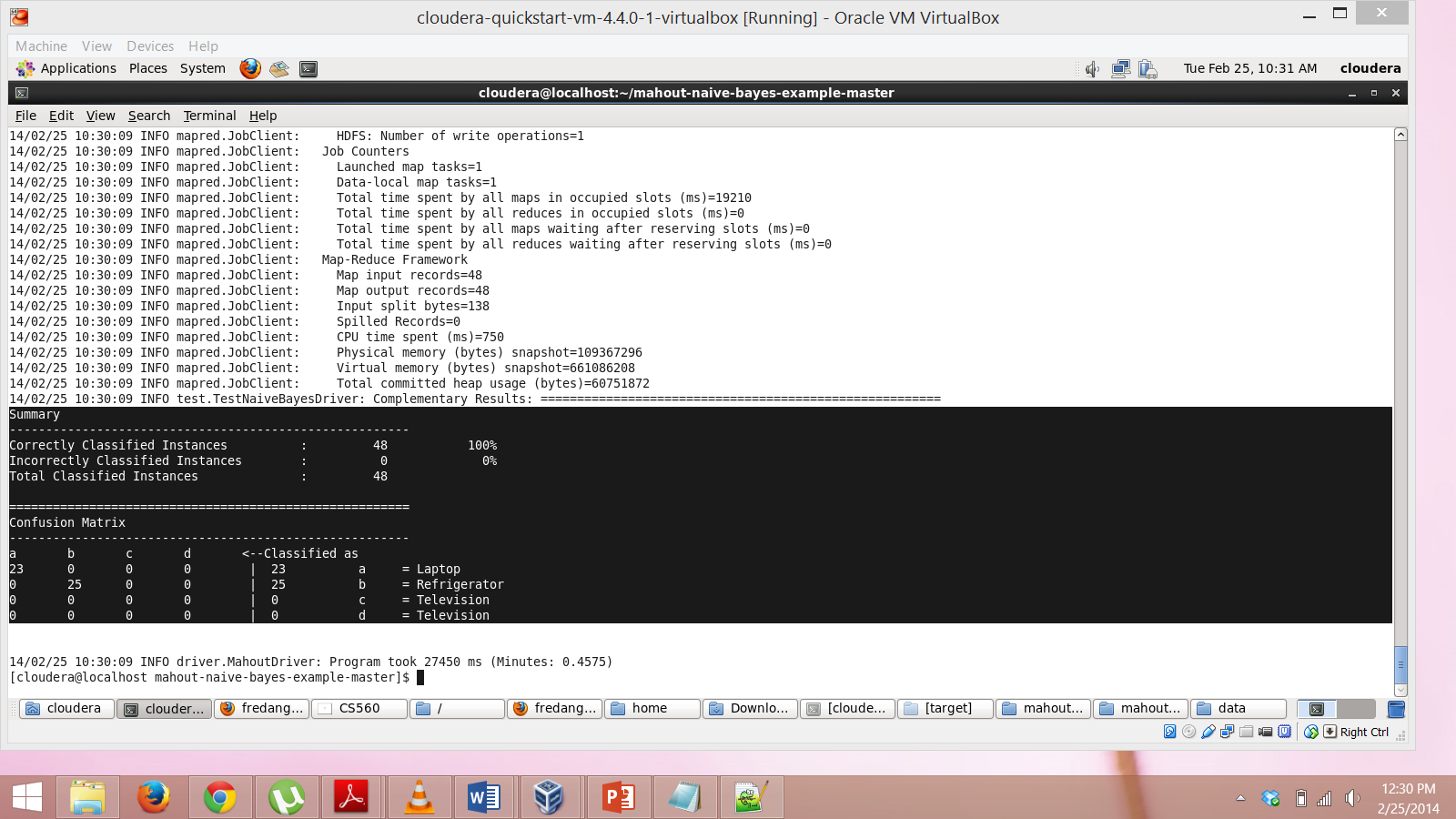
Correctly classified instances: 97.12



**Check if classifier works for Test data:**

It worked fine, Accuracy is 100% but the test data doesn’t have a classifier, Television. This is the reason why accuracy is 100% otherwise it would have got down to 80%.

This has also raised some questions on who the data is split in Mahout when we ask machine to split the data. Television classifier was not at all picked for the test data.



**To classify the data based on training set:**

Now after training and testing, data which is to be classified is fed to Mahout and the results are good. Attached is the output data in .txt format. All the commands and reason for executing the commands is written in a separate document.



**4. Converting the above attached classified data – output after running Mahout Application is converted to .JSON format and is pushed to SOLR.**

Java source codes and .txt and .JSON format files can be found in *Work files* folder. It was not yielding the expected results, so made cosmetic changes to the data in JSON format.

Please also see the attached file – *Data in SOLR* for detailed explanation with snapshots.

Corresponding JSON files after retrieving the data are:

Classifier 1: Refrigerator <http://localhost:8983/solr/collection2_shard1_replica1/select?q=id%3A%22Refrigerator%22&wt=json&indent=true>

Classifier 2: Television <http://localhost:8983/solr/collection2_shard1_replica1/select?q=id%3A%22Television%22&wt=json&indent=true>

Classifier 3: Laptop <http://localhost:8983/solr/collection2_shard1_replica1/select?q=id%3A%22Laptop%22&wt=json&indent=true>

**6) Using the above stated data, web service is built and the data is retrieved and popped up on the screen.**

Have tried using HTML5 to retrieve the data from SOLR using AJAX.

Please find the application at <http://laxmanduttdegala.github.io/> .

Couldn’t retrieve the data because of following reason: “Cross Origin XMLHTTPRequest”. This could be due to the reason that we are using Intranet IP address at the beginning of the JSon formatted URL. Will update you once I fix this.

We would give a classifier as input and the output will be the items which is retrieved from SOLR.

**7) Issues faced while performing the task**:

a) Activating SOLR – Explained above in point 1.

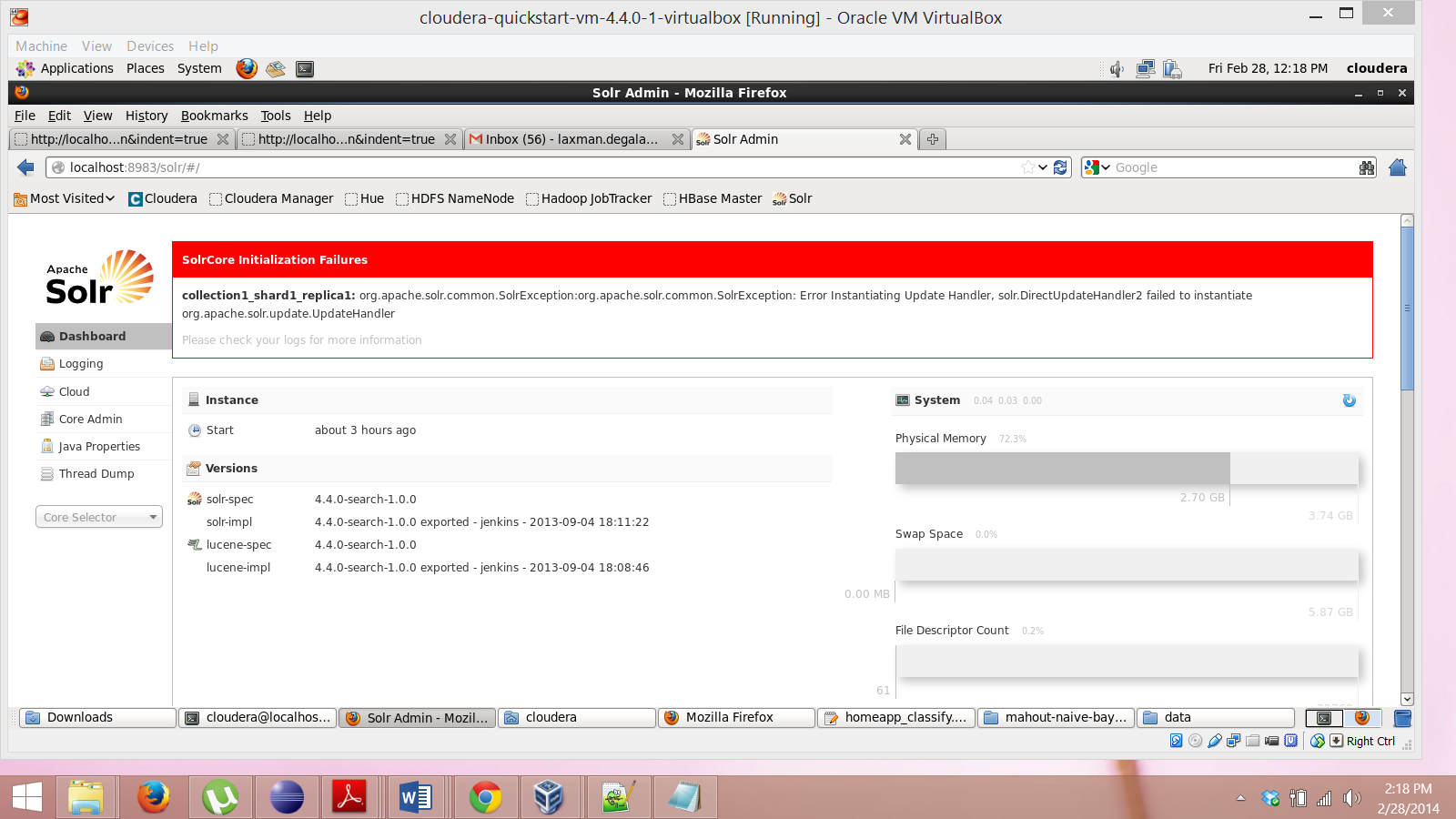
b) Installing Glassfish – Explained in point 2.

c) Iterations of data creation to increase the accuracy.

d) Putty didn’t work. Has few permissions issues so I have worked on the image that I installed on my computer.

e) Conversion from .txt to .json was tough and time taking.

f) When restarted Cloudera machine, Colection1/SHARD1 in SOLR where the data is stored cannot be seen. Please find below the attached screen shots:



Resolution: Due to insufficient time, I couldn’t find the correct resolution for this. I have recreated the data by creating another instance in Core Admin – Collection2.

**8) Lessons Learnt from Lab5:**

1. How to run an application in Mahout. What exactly Machine Learning is.
2. How to Train and Test the data, though didn’t work in detail on Algorithms, Used the existing ones.
3. What exactly is a Mapper and a Reducer, How they work.
4. Vector Weightage
5. How to work on SOLR. How to Push the data to SOLR (using shell scripting commands) and retrieve the data from SOLR (CSV, JSON, XML etc)
6. Creation of web service and using the same in our application.

This assignment has given me clear picture of what to be done in the current project.