

Steps for Getting Hadoop Up and Running on VirtualBox

First we need to download the following applications and jar files.

1- Download VirtualBox from Oracle (<https://www.virtualbox.org/wiki/Downloads>)

2- Download Appliances(Sandbox) for the VirtualBox from Horton Works

(<http://hortonworks.com/hdp/downloads/>)

Download the one that says for VirtualBox.

3- Download Hadoop Version 2.6.0 from Apache using this link

(<http://mirror.symnds.com/software/Apache/hadoop/common/hadoop-2.6.0/>)

The version needs to be the same with the one available inside horton.

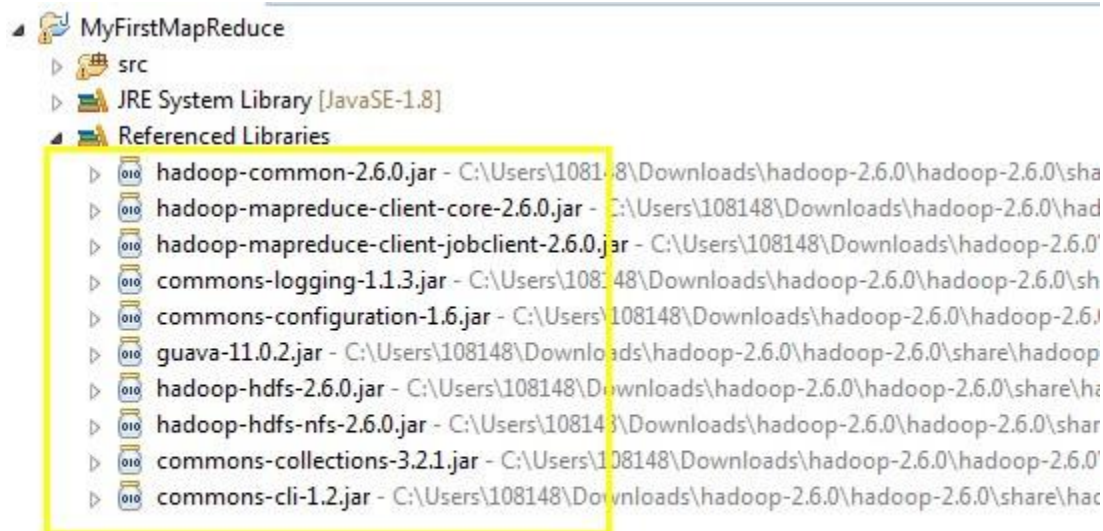
The version of the Hadoop inside the Horton can be checked by running this command on the linux inside the virtualBox.

> hadoop version

Or from

4- Make sure these jar files are available inside the hadoop folder you downloaded.

- a. hadoop-common-*.jar
- b. hadoop-mapreduce-client-core-*.jar
- c. hadoop-mapreduce-client-jobclient-*.jar
- d. commons-logging-1.1.3.jar
- e. commons-configuration-1.6.jar
- f. guava-11.0.2.jar
- g. hadoop-hdfs-2.6.0.jar
- h. hadoop-hdfs-nfs-2.6.0.jar
- i. commons-collection-3.2.1.jar
- j. commons-cli-1.2.jar



Using Maven Project and POM

You should be connected to the Internet for the maven to download dependencies.

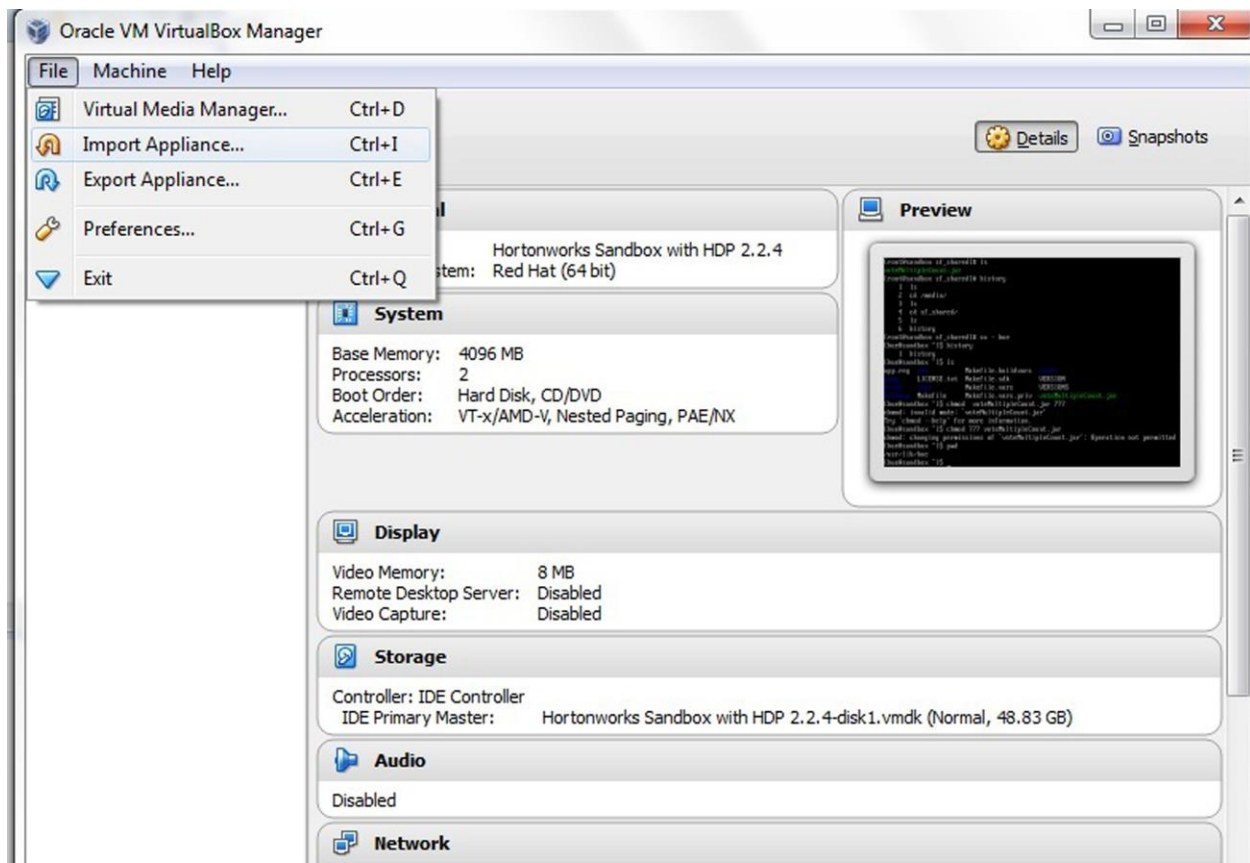
This is a second way to get all the necessary jar files imported automatically. Here you only need to define the artifacts in the pom.xml.

1. First create an ordinary java project. Add all the necessary MapReduce classes.
2. Right Click on the project and > configure > convert to maven project
3. Open the pom file with text editor, and you will see something like this. At the bottom add dependencies for the hadoop depending on the version you want.
4. Save it and it will fetch all the necessary jars

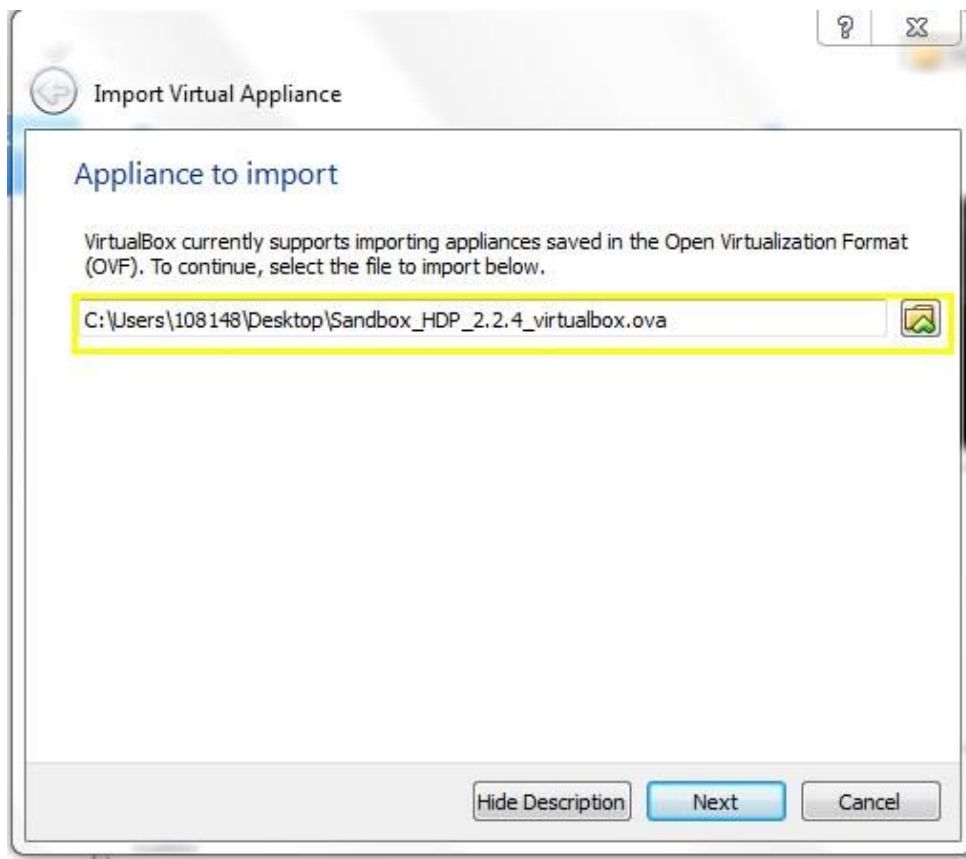
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>HybridApproachRelativeFrequencies</groupId>
  <artifactId>HybridApproachRelativeFrequencies</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.1</version>
        <configuration>
          <source>1.7</source>
          <target>1.7</target>
        </configuration>
      </plugin>
    </plugins>
  </build>
  <dependencies>
    <dependency>
      <groupId>org.apache.hadoop</groupId>
      <artifactId>hadoop-client</artifactId>
      <version>2.6.0</version>
    </dependency>
  </dependencies>
</project>
```

Installation Steps for VirtualBox and the Appliance

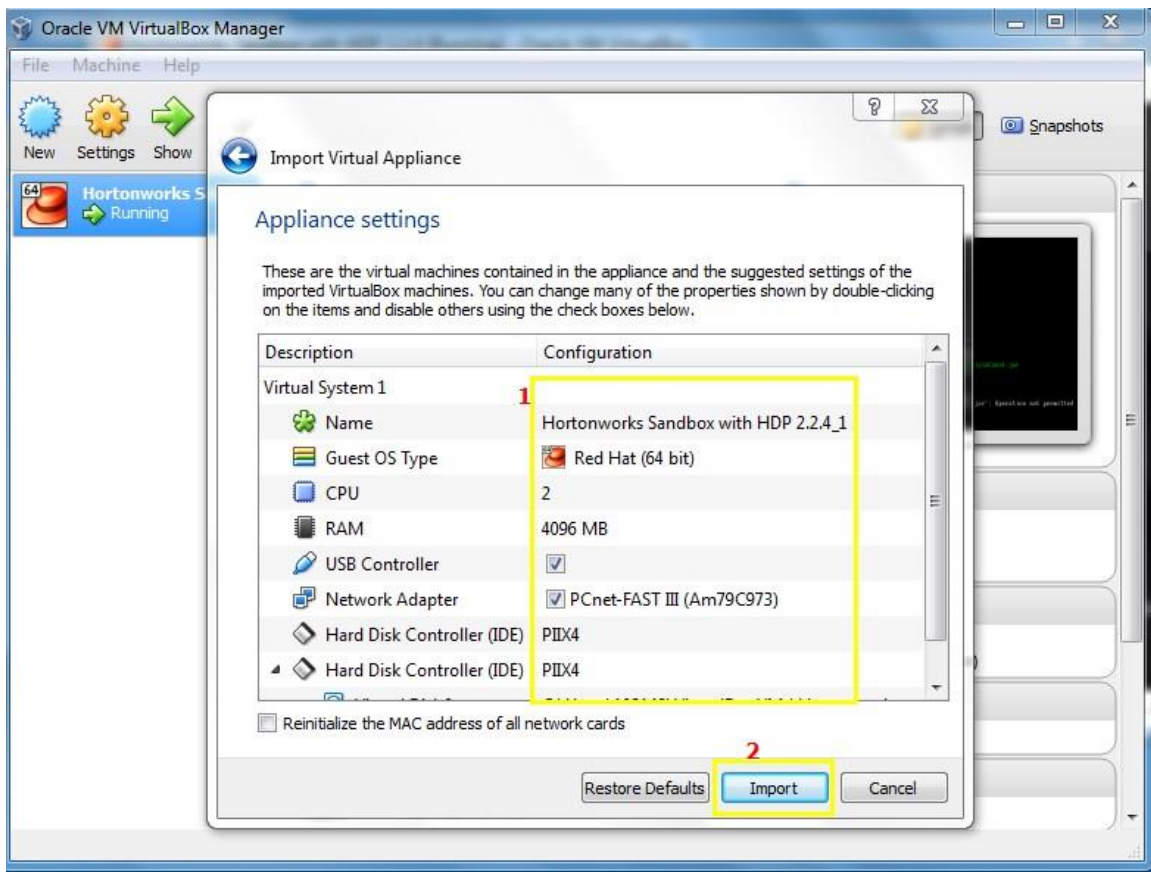
1- Install the VirtualBox you downloaded from oracle.



2- From the File menu of the VirtualBox select “import Appliance” and provide the path for the Appliance(Sandbox).



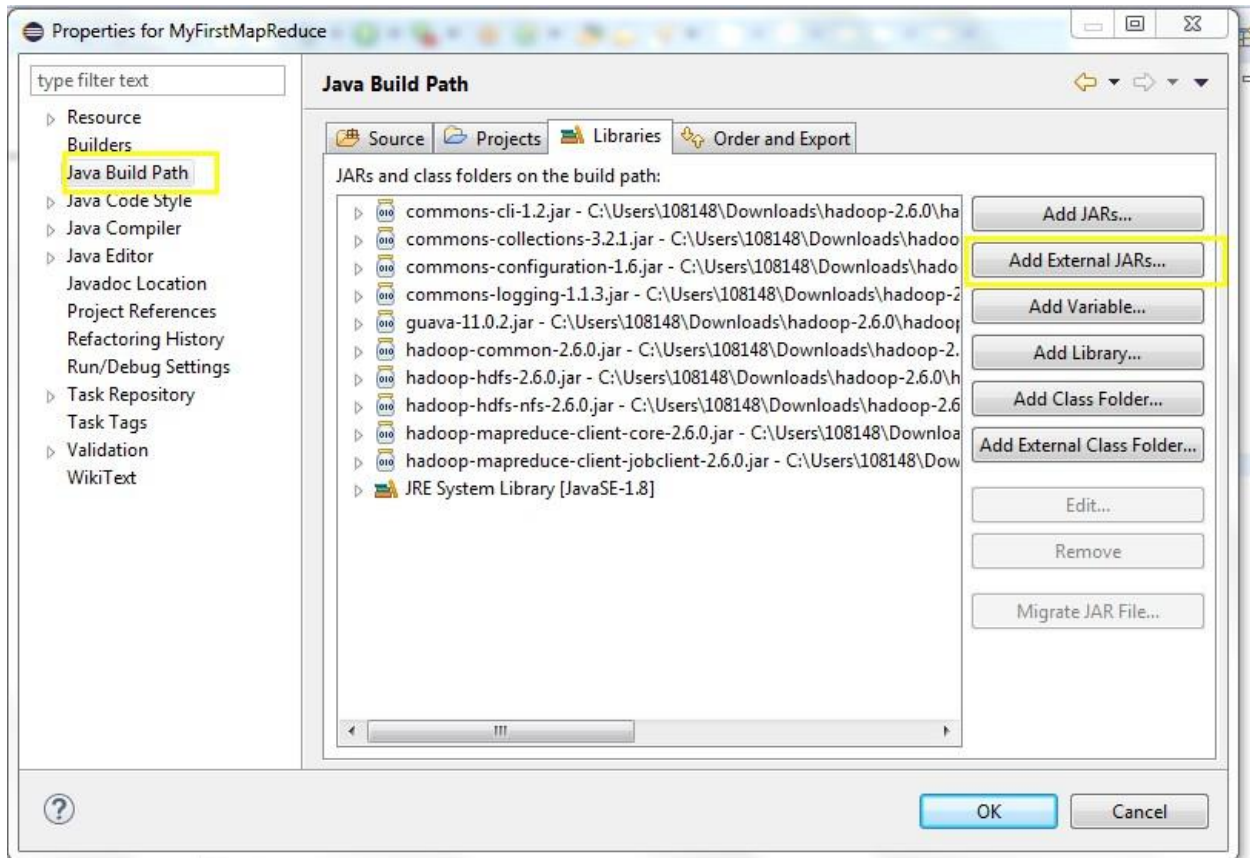
- 3- Once it is installed before powering it on, or before installing and start importing, we can change the parameters for memory, storage, shared folder, cd....etc



Eclipse Set up

Creating a Map Reduce project on Eclipse

- 1- Create a new project on Eclipse and name it “MyFirstMapReduce”.
- 2- Next right click on the project you just created and select properties. On the ‘Libraries’ tab select ‘Add External Jars’. Then add the 10 jar files you have already downloaded. The jar files may be less or more depending on the needs of your program).



- 3- Then create a Mapper Class let's say “CustomerHistoryMapper”
- 4- Now create a Reducer Class “CustomerHistoryReducer”
- 5- This is a driver class, “CustomerHistoryApplication”
- 6- And may be a “CustomerHistoryPartitioner” class
- 7- Now we need to check the java version inside the horton Sandbox.

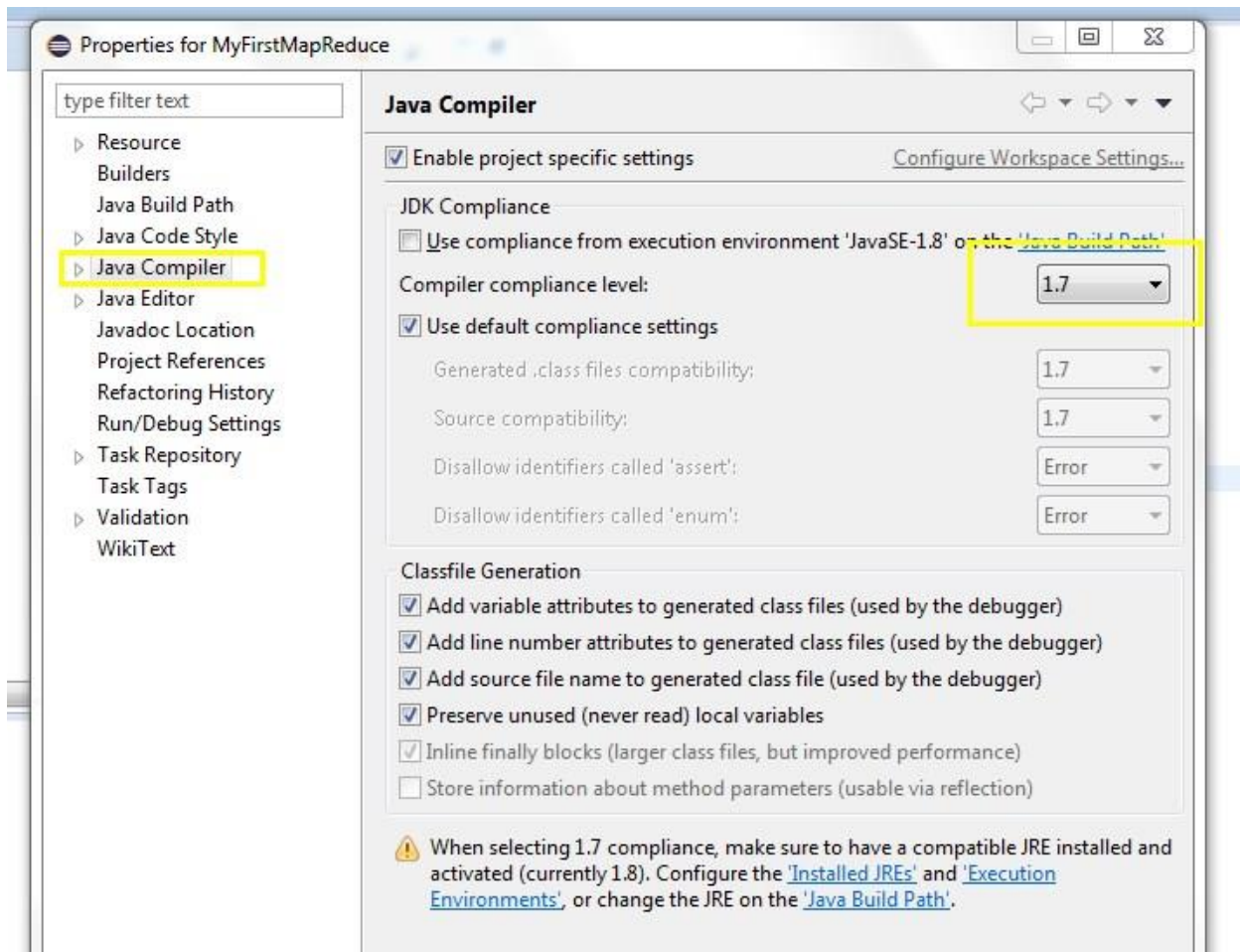
Run the following command from the linux in the Horton.

➤ Java -version

As I have java version “1.7” I will set up my compiler compliance level to 1.7.

This can be done by right-clicking on the project in eclipse > properties

On the left side select “java compiler”, then choose 1.7

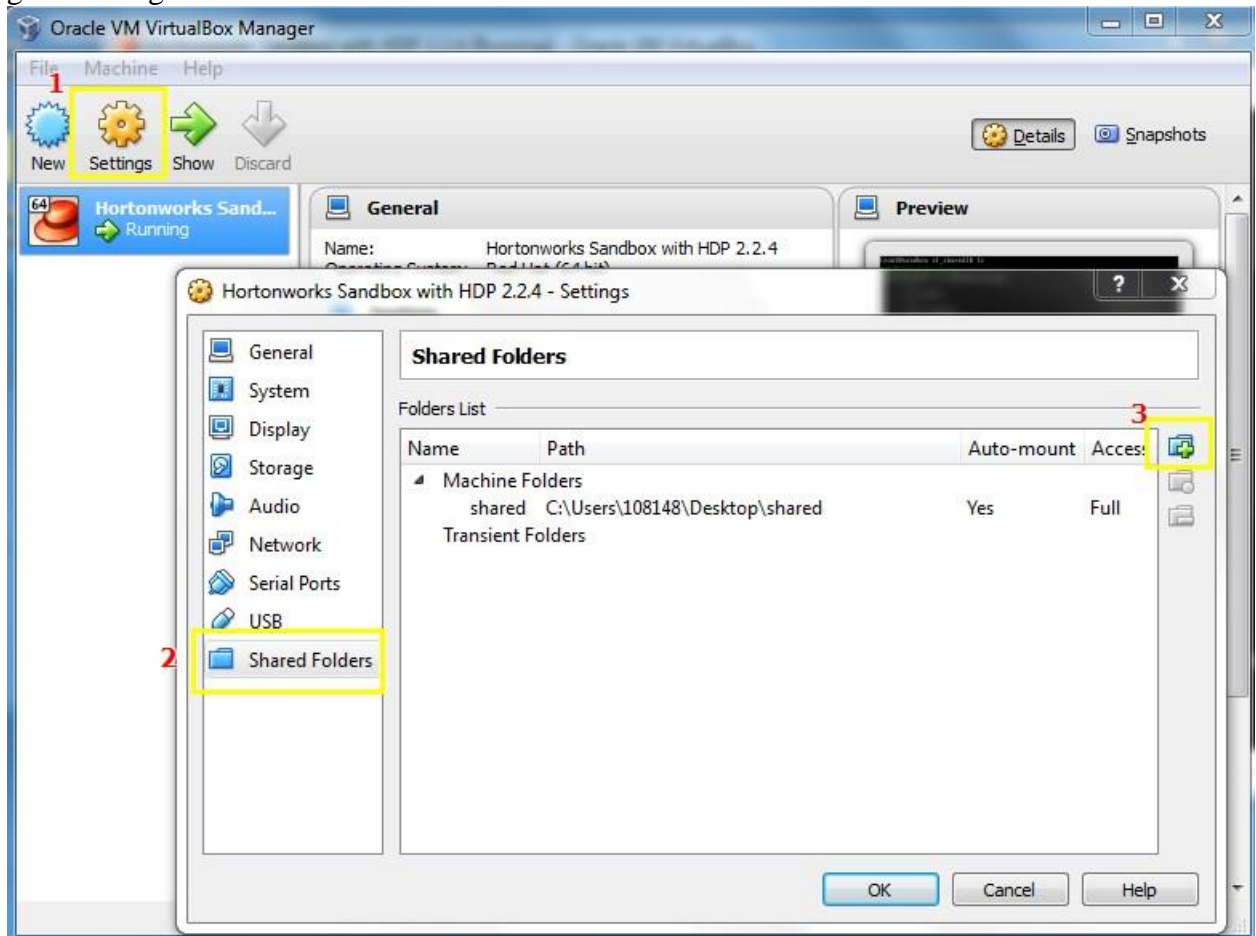


Shared Folder Configuration

In order to access files and jars available on our system from inside the virtual machine, we need to setup the shared folder in the virtualbox.

Put all the necessary jar files you might need from the inside of the virtual machine in this folder.

Next, while the virtual machine(“MyFirstHadoop”) is powered off, go to settings > shared folder

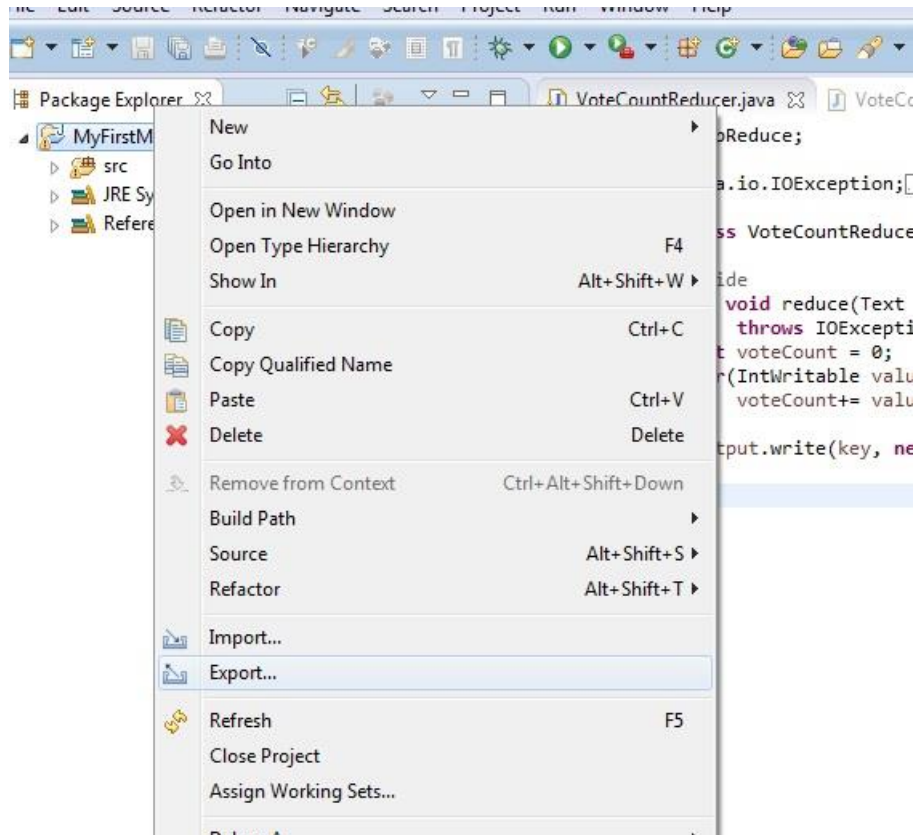


Using the folder icon on the right edge add any folder (let's say a folder named “sharedFolder” is available on your desktop). Select “auto mount ” from the check boxes.

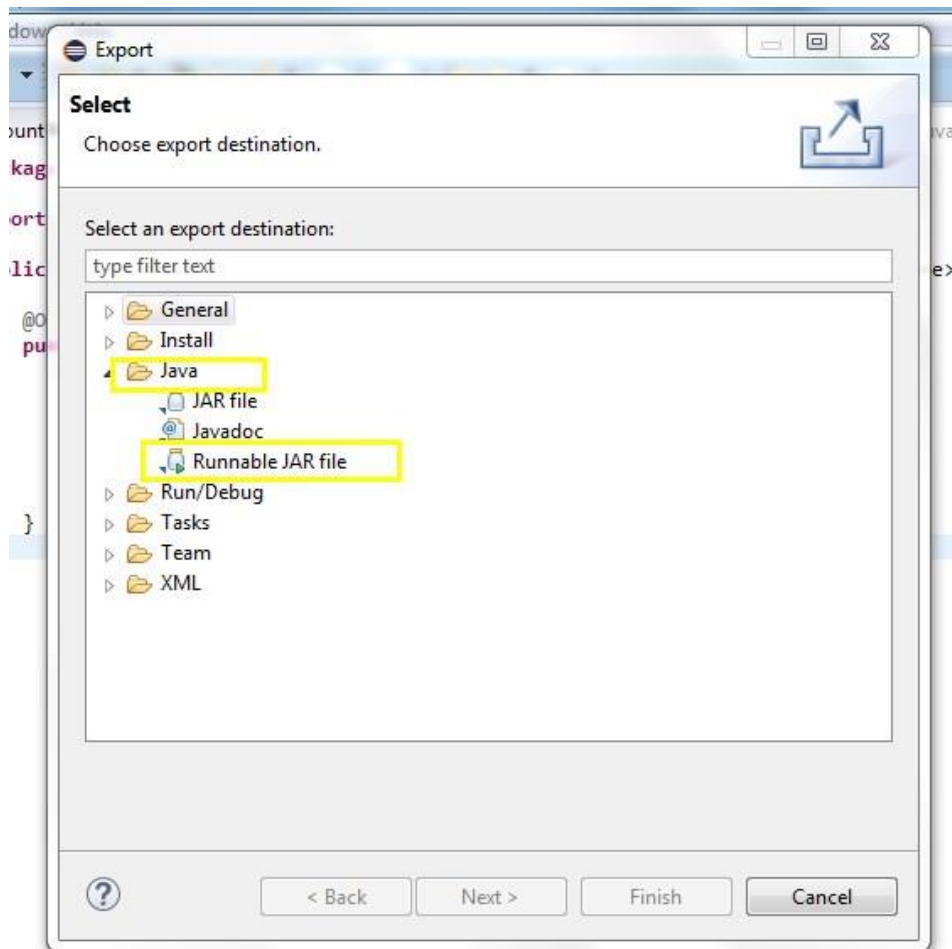
Creating a Runnable Jar File

Now the project is ready to be archived as jar. The steps for this are:

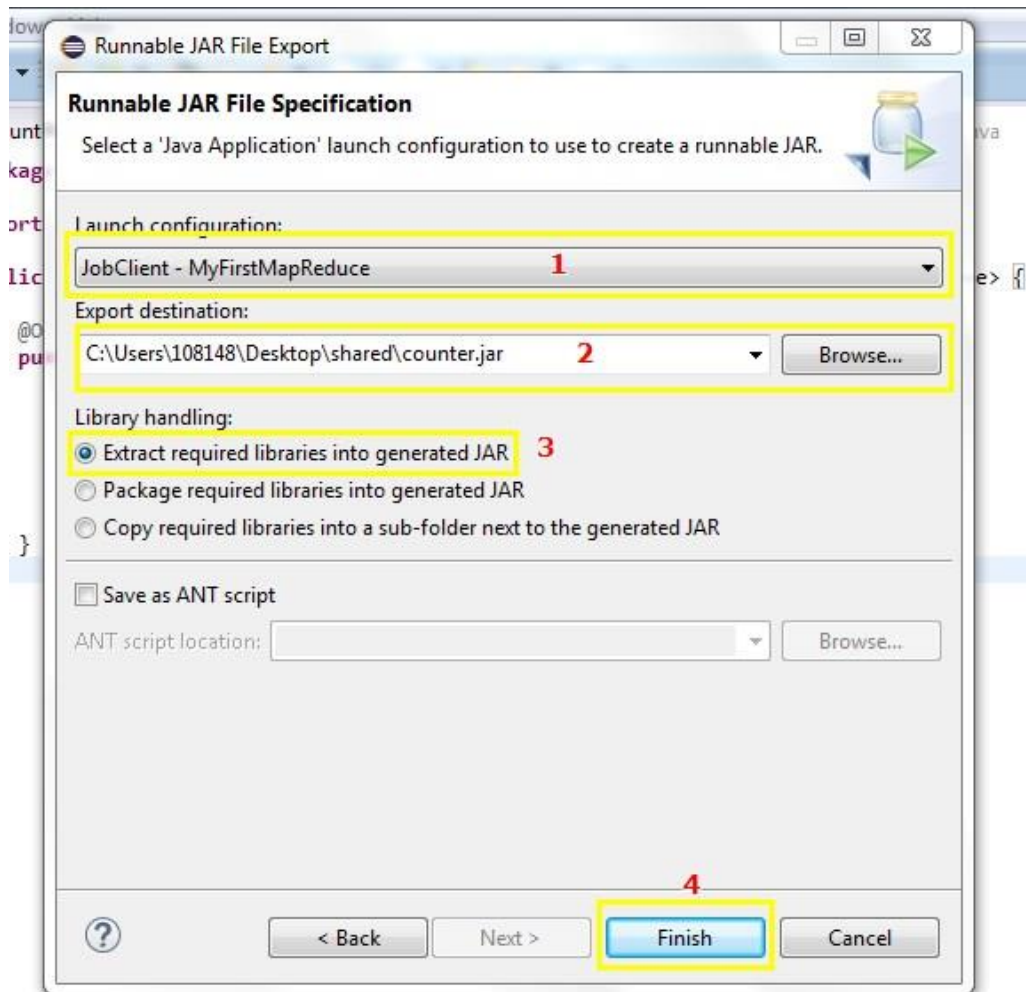
- 1- Right click on the project then “Export”



2- Select "Java", expand it now select "Runnable Jar"



- 3- In Launch Configurations select the project name “MyFirstMapReduce”
- 4- Provide a destination. Here make the destination to be on the shared folder you created earlier and give it some name let’s say “counter”.
- 5- Select “Extract required libraries into generated JAR”. Then Finish. (Ignore any warnings that may show up during the jar creation).



Now the jar named “counter.jar” is created inside the “sharedFolder” folder. Check to confirm!

Verify Shared Folder Access from Sandbox

- 1- Power on the virtual machine. Login using “root” and “hadoop” as user name and password respectively.
- 2- Run this command to verify if the shared folder and its contents are accessible.
 - `cd /media/`
Now list the files in this directory
 - `ls`
If you can see “sh_sharedFolder” from the list displayed you are good to go. Make sure the jar file is inside the folder by going in to the directory and listing the files.
 - `cd /sh_sharedFolder/`
 - `ls`
Now the “counter.jar” file should be listed.
- 3- Copy this jar file to the hue user’s directory.
 - `cp counter.jar /usr/lib/hue/`
- 4- Now change to “hue” user using this command

- su – hue
- 5- Changing the mode of the jar file we just copied to the hue user will be the next step. Currently we are “hue” user. Give rwx permissions using 777
So do:
 - sudo chmod 777 counter.jar

We can check all permission are granted using

- ls -l (all the 3 permissions should be -rwxrwxrwx)

Setting up the Hadoop and Input files from the Web

- 1- Use this link to access Horton from the browser (<http://127.0.0.1:8000/>)



Dive right in

Get started using Sandbox with your own datasets, and connect it to your existing tools and applications.

To use the web interface to explore HDP 2.2 Sandbox navigate to <http://127.0.0.1:8000/>. The username and password is **hue** and **1111** respectively.

To SSH into the VM and explore HDP 2.2 Sandbox from the command line

```
ssh root@127.0.0.1 -p 2222;
```

The password is **hadoop**

- 2- Login as **hue** using password **Hadoop**

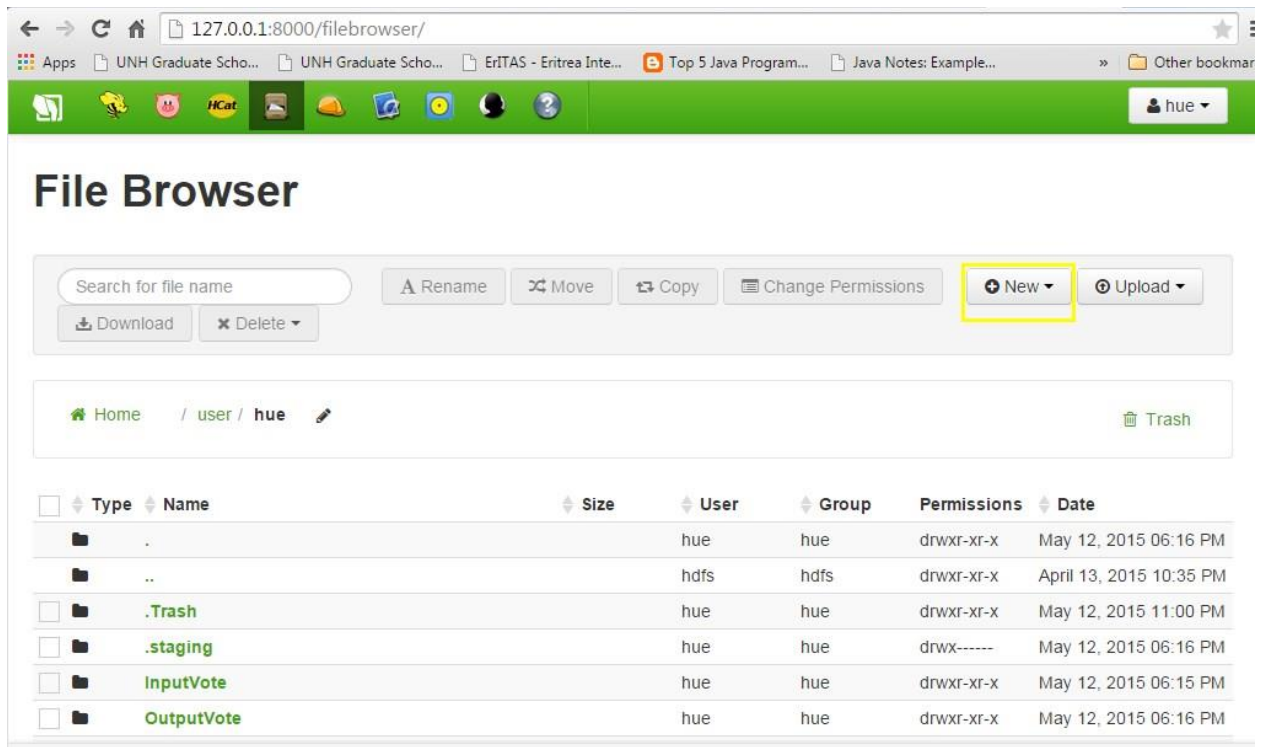


Hortonworks Sandbox with HDP 2.2

[Leave Feedback](#)

Component	Version
Hue	2.6.1-2
HDP	2.2.4
Hadoop	2.6.0
Pig	0.14.0
Hive-Hcatalog	0.14.0
Oozie	4.1.0
Ambari	2.0-1238
HBase	0.98.4

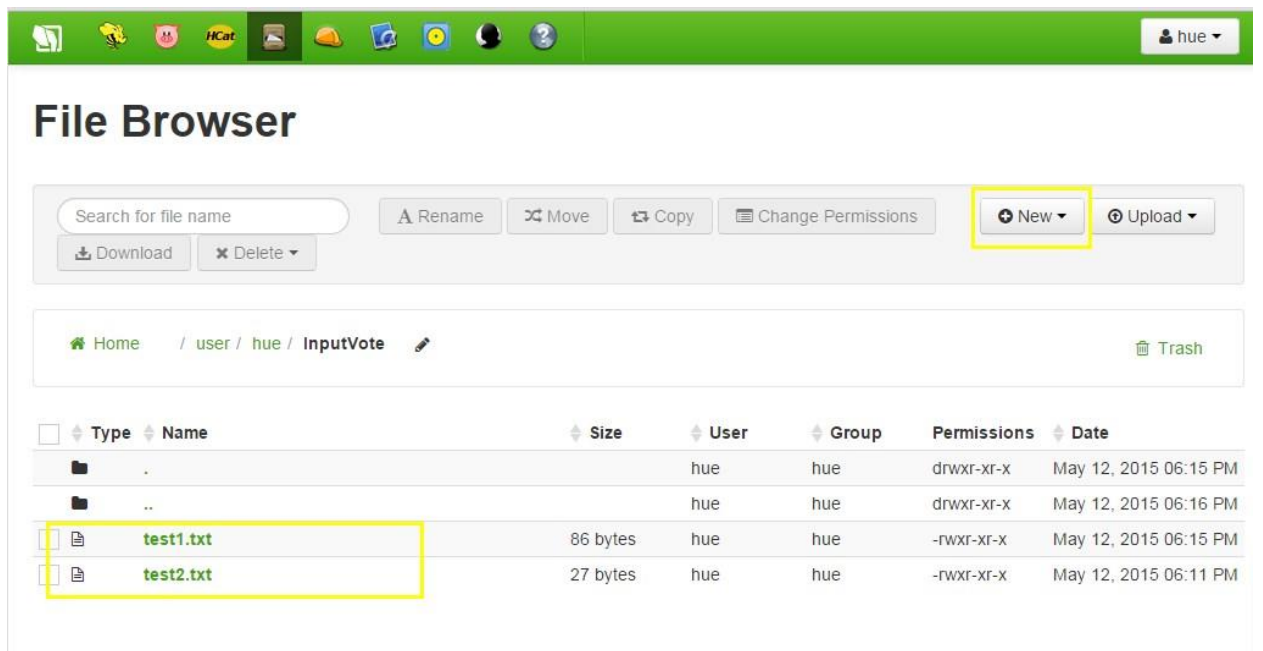
- 3- Go to **File Browser** and select the **New** button and then **Directory**. Create a directory with a name **InputFiles**.



The screenshot shows the File Browser interface at the URL 127.0.0.1:8000/filebrowser/. The top navigation bar includes a search bar and buttons for 'Rename', 'Move', 'Copy', 'Change Permissions', 'New', and 'Upload'. The 'New' button is highlighted with a yellow box. Below the navigation bar, the breadcrumb path is 'Home / user / hue'. The main content area displays a table of files and directories.

Type	Name	Size	User	Group	Permissions	Date
Folder	.		hue	hue	drwxr-xr-x	May 12, 2015 06:16 PM
Folder	..		hdfs	hdfs	drwxr-xr-x	April 13, 2015 10:35 PM
Folder	.Trash		hue	hue	drwxr-xr-x	May 12, 2015 11:00 PM
Folder	.staging		hue	hue	drwx-----	May 12, 2015 06:16 PM
Folder	InputVote		hue	hue	drwxr-xr-x	May 12, 2015 06:15 PM
Folder	OutputVote		hue	hue	drwxr-xr-x	May 12, 2015 06:16 PM

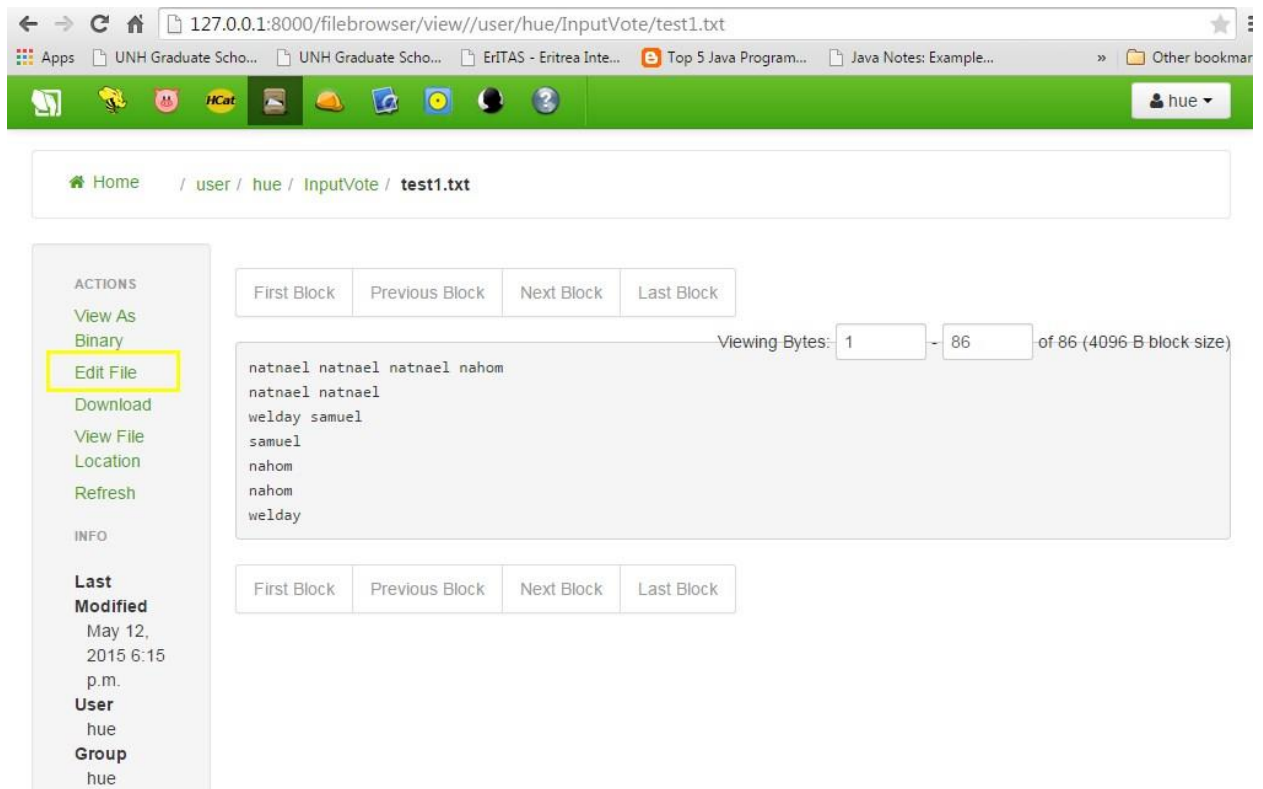
- 4- Open this newly created directory and create new files like test1.txt and test2.txt



The screenshot shows the File Browser interface with the 'New' button highlighted. The breadcrumb path is 'Home / user / hue / InputVote'. The main content area displays a table of files and directories, with the 'test1.txt' and 'test2.txt' files highlighted by a yellow box.

Type	Name	Size	User	Group	Permissions	Date
Folder	.		hue	hue	drwxr-xr-x	May 12, 2015 06:15 PM
Folder	..		hue	hue	drwxr-xr-x	May 12, 2015 06:16 PM
File	test1.txt	86 bytes	hue	hue	-rwxr-xr-x	May 12, 2015 06:15 PM
File	test2.txt	27 bytes	hue	hue	-rwxr-xr-x	May 12, 2015 06:11 PM

- 5- Add some text to these files using the **Edit File** menu on the right hand side and save the changes.



We don't need to create an output file and directory; they will be automatically created when we run our program later from the Horton sandbox.

Running the JAR from the virtual machine

Now let's go back to the virtual machine. We left it after granting all rwx permissions for the counter.jar file inside the "hue" user.

Now the final command for running the MapReduce program is the following:

We are on this directory `/usr/lib/hue`

We can check this using "pwd" (print working directory) command.

➤ `hadoop jar counter.jar /user/hue/InputFiles /user/hue/OutputFiles`

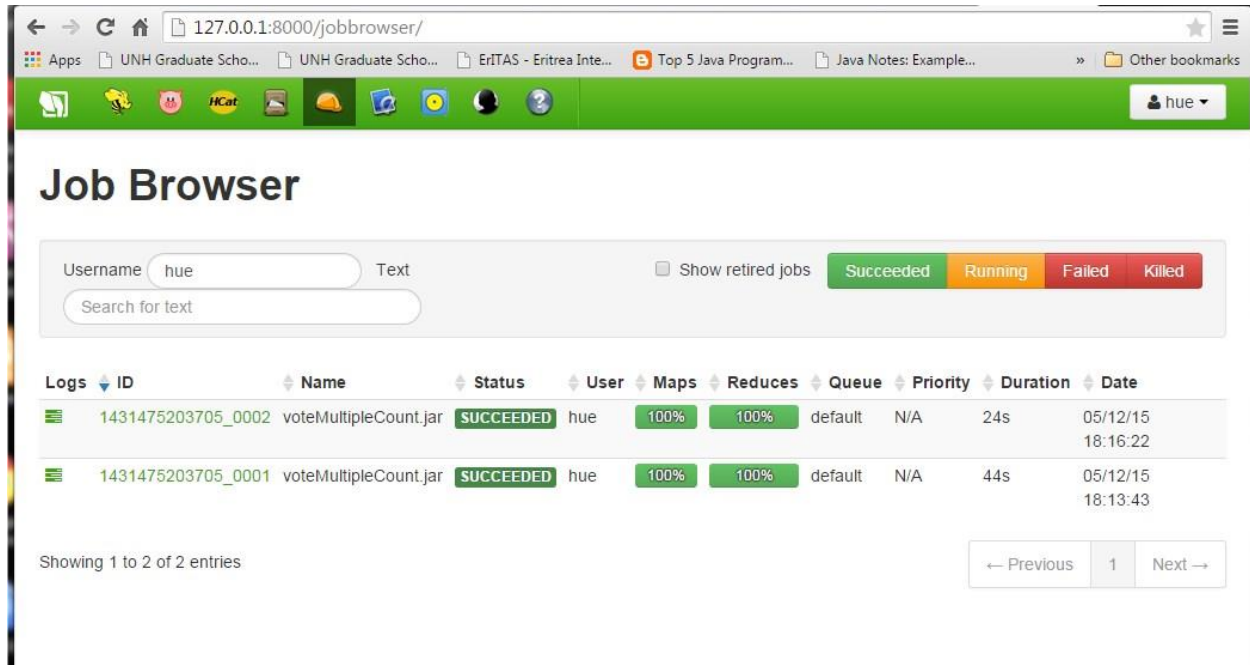
Remember, the InputFiles directory you created earlier is given as an input to this command and the output will be created automatically. These two paths are different: `/usr/lib/hue` and `/user/hue/InputFiles`. The first one is on the Linux in the virtual box, the second one while accessing Horton from the browser.

Monitoring Progress and Output from the Web

We can see the status of the running jobs from the browser.

Go to **Job Browser** and see the progress.

Once the Jobs are completed the output can be checked from the output folder we provided while running the “hadoop jar” command; Go to **File Browser** and open this path /user/hue/OutputFiles

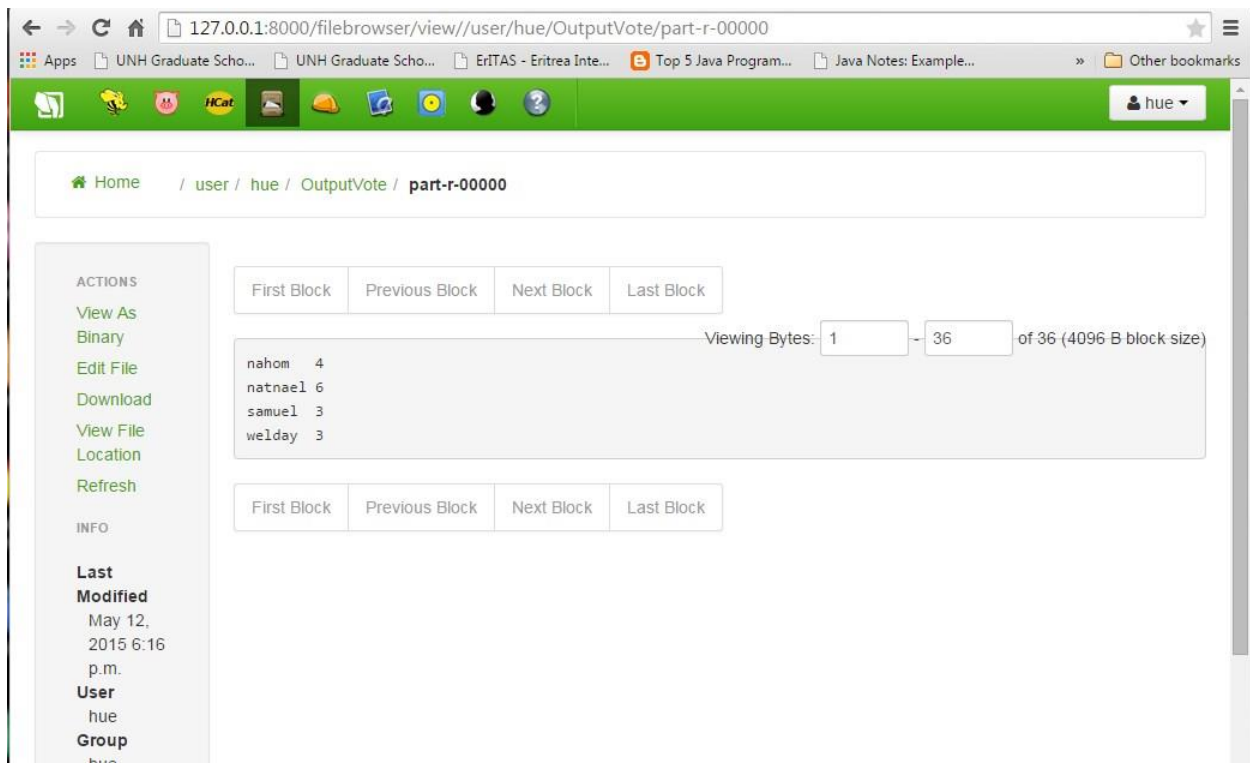


The screenshot shows the Hue Job Browser interface. At the top, there's a navigation bar with the URL 127.0.0.1:8000/jobbrowser/. Below the navigation bar, there's a search section with a username field set to 'hue' and a 'Show retired jobs' checkbox. A table of jobs is displayed with columns: Logs, ID, Name, Status, User, Maps, Reduces, Queue, Priority, Duration, and Date. Two jobs are listed, both with a status of 'SUCCEEDED'.

Logs	ID	Name	Status	User	Maps	Reduces	Queue	Priority	Duration	Date
	1431475203705_0002	voteMultipleCount.jar	SUCCEEDED	hue	100%	100%	default	N/A	24s	05/12/15 18:16:22
	1431475203705_0001	voteMultipleCount.jar	SUCCEEDED	hue	100%	100%	default	N/A	44s	05/12/15 18:13:43

Showing 1 to 2 of 2 entries

Checking Output of the MapReduce Jobs



The screenshot shows the Hue File Browser interface. The URL is 127.0.0.1:8000/filebrowser/view//user/hue/OutputVote/part-r-00000. The breadcrumb path is Home / user / hue / OutputVote / part-r-00000. On the left, there's a sidebar with 'ACTIONS' (View As, Binary, Edit File, Download, View File, Location, Refresh) and 'INFO' (Last Modified: May 12, 2015 6:16 p.m., User: hue, Group: hue). The main area shows a list of output blocks with columns: First Block, Previous Block, Next Block, Last Block. The current block is 'part-r-00000' and it contains the following text:

```
nahom 4
natnael 6
samuel 3
wellday 3
```

Viewing Bytes: 1 - 36 of 36 (4096 B block size)

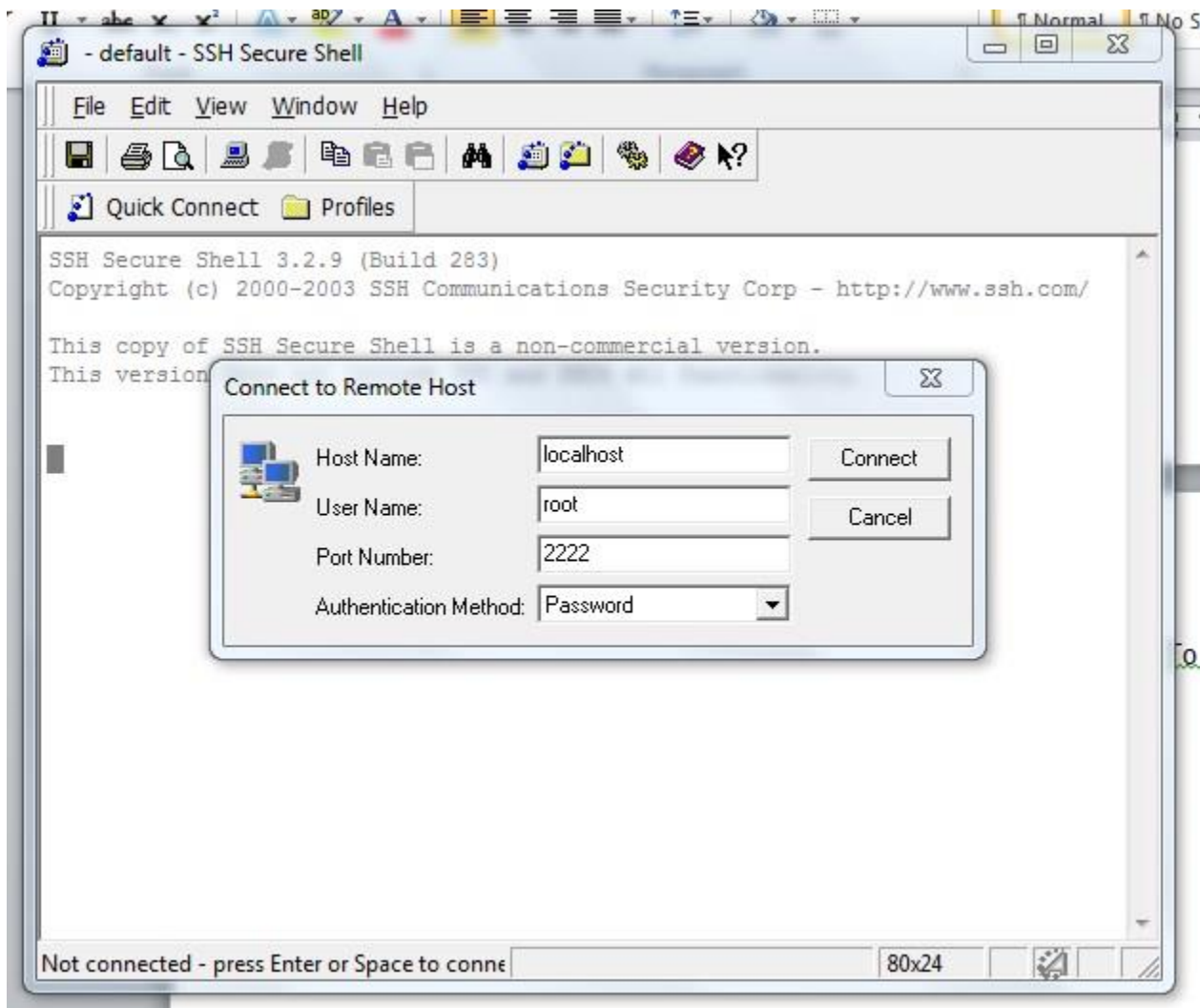
We can also access the Horton Sandbox remotely using SSH. To do this download Putty or Secure Shell Client.

Use Host Name – localhost

User Name – root

Port – 2222

password – hadoop



Writing a Batch file or Script

Also you can write a batch or a script for keeping all the command in one file and executing them together instead of writing them every time.

The steps are as follows

1 – Go to the Horton Sandbox command line

2 – Change to a root user using

➤ `su – root`

3 – Create a file lets mybatch

➤ `vi mybatch`

4 – Add a list of commands that you have used earlier. For example

```
#!/bin/bash
```

```
echo "-----"
```

```
echo "Now Executing MapReduce Job with Pairs Approach...."
```

```
echo "-----"
```

```
cp /media/sf_shared/pairs.jar /usr/lib/hue/
```

```
chmod 777 /usr/lib/hue/pairs.jar
```

```
hadoop jar pairs.jar /user/hue/inputData /user/hue/outputPairs
```

```
echo "-----"
```

```
echo "Completed Paris Execution!!!"
```

```
echo "-----"
```

5- Now save this file and change it's mode to executable.

➤ `chmod 777 mybatch`

6- Now run it as follows

➤ `./mybatch`

NB. The paths in the batch file should match with your actual paths and all the jar files should be available in the shared folder we created earlier.

