# Lab Session 1: MapReduce and Hadoop

## Objectives

- Understand the MapReduce programming model.
- Setting up Hadoop in a Pseudo Distributed Mode.

## Overview

You are required to install Hadoop on a single node cluster. You will then practice running few HDFS commands, creating analytics applications using MapReduce, and the executing them using Hadoop.

## Method 1

### Downloading Hadoop

- Follow the following command to download Hadoop on your machine: https://downloads.apache.org/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz
- Extract the downloaded file using the command:
  tar -xzvf hadoop-2.10.1.tar.gz

### Setting up Hadoop

- You will need to download the latest stable version of Hadoop (2.10.1) from this link: http://hadoop.apache.org/releases.html as described above.
- Setup the downloaded Hadoop version on your machine in a **Pseudo Distributed** mode. These are the steps that you will need to follow: https://goo.gl/8KVyGJ

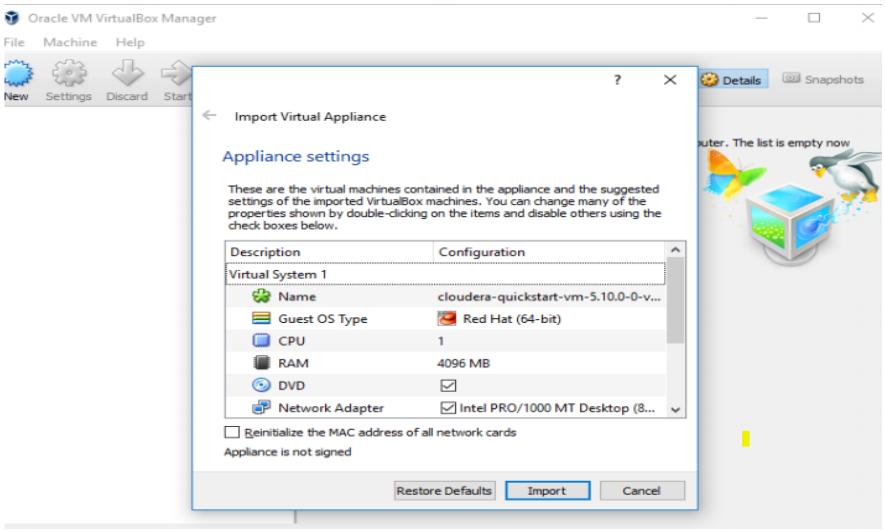
### HDFS

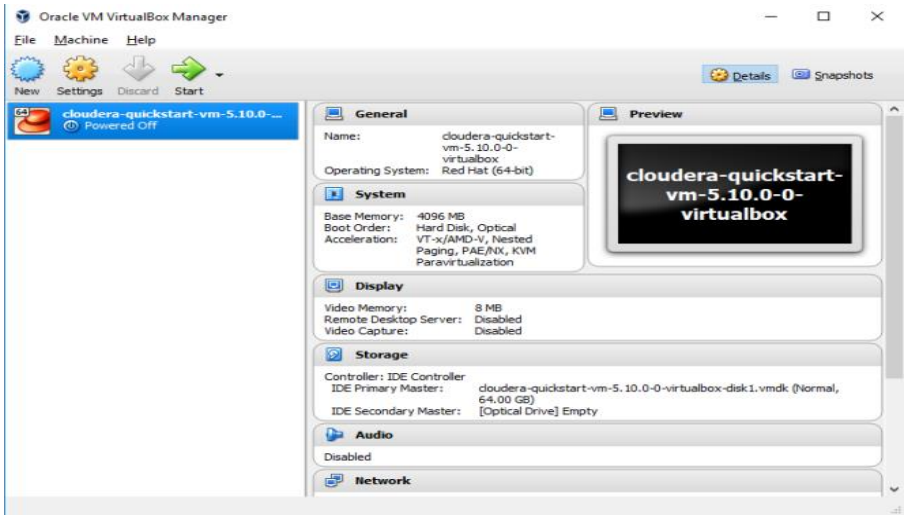
- Create a directory called input in your home directory.
- Download the following text files from the Gutenberg project, in Plain Text UTF-8 format. Download the zip file from here gutenbergprojectfiles.zip .
- Download the zip file, extract it, and store the files to the input directory on your machine.

- Create directory on HDFS with your userid using the commands: bin/hdfs dfs -mkdir /user and bin/hdfs dfs -mkdir /user/userid (userid is your user name)
- Copy the input directory from your local disk to HDFS using the Hadoop command: bin/hdfs dfs -copyFromLocal /home/userid/input /home/userid/input. The first path is the source, which is on your local disk. The second path is the destination, which is on HDFS.
- Now check that the files were already copied using this command: bin/hdfs dfs -ls /home/userid/input

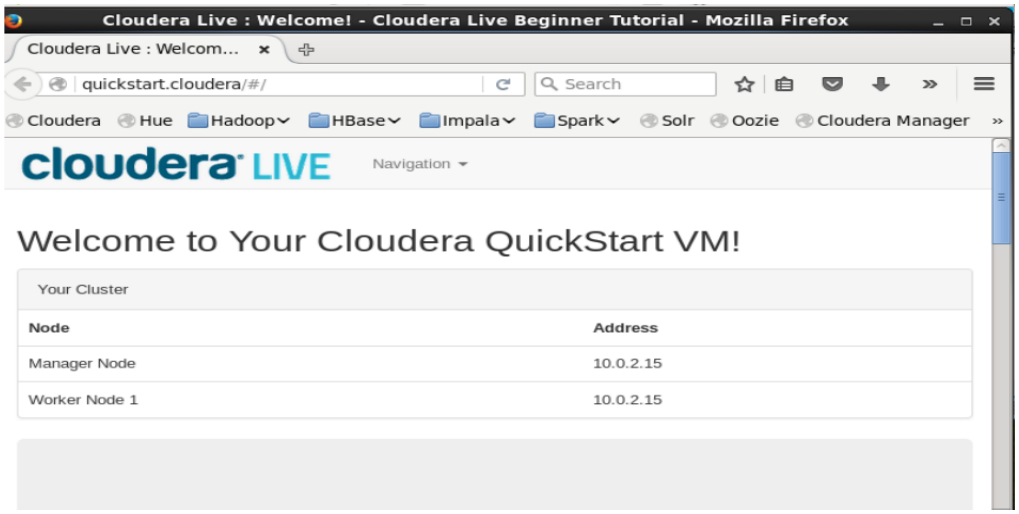
## Method 2

- you can install cloudera , it started as a hybrid open-source Apache Hadoop distribution. it's virtual machine include everything you need like OS & Hadoop that it's ready to use.
- Through opening the file .ovf , the virtualBox willl open .
- you need to import to start creating Virtual machine with this below Specifications.






- After Turning on the VM , it will appear the components of Hadoop.




- So Now you have single node of hadoop cluster. this cluster is enough to test the components of hadoop and know how to deal with it.

**HDFS**

- By opening terminal you can run few commands for example : to see all commands exist in the hadoop system just type $hadoop fs
- To see the contents in HDFS we write $hadoop fs -ls /
- To create folder called mydata in root directory we write $hadoop fs -mkdir /mydata
- To ensure the folder is created we write again $hadoop fs -ls /

```
[cloudera@quickstart ~]$ hadoop fs -ls /
Found 7 items
drwxrwxrwx   - hdfs     supergroup          0 2017-04-05 04:27 /benchmarks
drwxr-xr-x   - hbase    supergroup          0 2017-08-05 09:26 /hbase
drwxr-xr-x   - cloudera supergroup          0 2017-08-05 22:52 /mydata
drwxr-xr-x   - solr     solr                0 2017-04-05 04:29 /solr
drwxrwxrwt   - hdfs     supergroup          0 2017-08-05 10:57 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-08-05 10:57 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-04-05 04:29 /var
[cloudera@quickstart ~]$
```

- Note : You need to make a difference between linux and HDFS

  Ex : To view the stored files in linux : we wrote " ls" while in HDFS "hadoop fs -ls"

- To create file in linux system then copy it to HDFS

  $getit testfile.txt &  (& mean not hold other processes)

  Then save the file after writing in it.

- To ensure that the file exits we write $ ls

```
[cloudera@quickstart ~]$ ls
cloudera-manager  Downloads                 kerberos  Pictures      Videos
cm_api.py         eclipse                   lib       Public        workspace
Desktop           enterprise-deployment.json Music    Templates
Documents         express-deployment.json   parcels   testfile.txt
[1]+  Done                    gedit testfile.txt
[cloudera@quickstart ~]$
```

- To copy the file to HDFS we write $hadoop fs -put testfile.txt /mydata/test/
- To ensure the file exits we write $hadoop fs -ls /mydata/test/

```
[cloudera@quickstart ~]$ hadoop fs -ls /mydata/test/
Found 1 items
-rw-r--r--   1 cloudera supergroup         15 2017-08-05 23:17 /mydata/test/testfile.txt
[cloudera@quickstart ~]$
```

- Also you can copy from HDFS to local system by writing

  $hadoop fs -get /mydata/test/testfile.txt localtestfile.txt

- **Note : no command "cd" exist in hadoop as hadoop is stateless and doesn't remember the current folder**
- To delete the file we already created it (testfile.txt) we write

  $hadoop fs -rm /mydata/test/testfile.txt

### Word Count application

Now, you will create the Word Count application and run it as a Hadoop job on the data loaded on HDFS.

- You can create project through Eclpise application then you need to add libraries through " ADD EXTERNAL JARS" of hadoop and clients .
- You need to build the WordCount example described in this tutorial. Name the created jar file wc.jar.
- You are now ready to run the jar file using:
  bin/hadoop jar wc.jar WordCount /home/userid/input /home/userid/output

  $Hadoop jar jarFileName.jar ClassName InputFile.txt OutputFolderName

- Check the output files created in the /home/userid/output.
- To view the content of the result in output file

  $ hdfs dfs -cat /output/part-r-00000

- Copy the output directory to your local disk using:
  bin/hdfs dfs -get /home/userid/output /home/userid/output. You can also use copyToLocal or getmerge
- The output will be in the file /home/userid/output. Your implementation is correct if the following command produces the output shown here:

```
$ sort -n -k2 part-r-00000 | tail -10
he: 31787 was: 35732 that: 34186 I: 36348
in: 44649 a: 59576 to: 72663 of: 79176
and: 91134 the: 153053
```

**Resources**

- HDFS shell commands
- MapReduce Tutorial

**Notes**

- You may work in groups of 2 or 3.

**Good Luck**