SPARK DEVELOPER TRAINING – LAB GUIDE

Version 1.0

Abstract

This is the lab guide for the participants to learn and complete all lab exercises as part of the Apache Spark Developer training.

Manaranjan Pradhan

manaranjan@enablecloud.com

https://www.linkedin.com/in/manaranjanpradhan

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

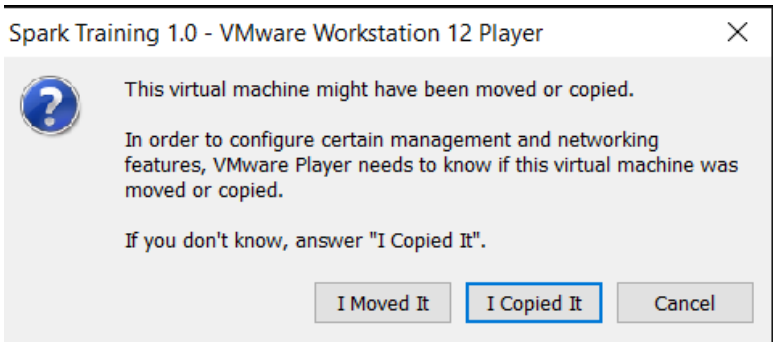
1) Launching the VM

Go to the directory on your windows or mac machine, where you have copied the lab related files

I. Install VMWare Player 6.0+ or VMWare fusion for MAC
II. Unrar the following files in your lab\software directory
    - Spark Training - Base 1.0

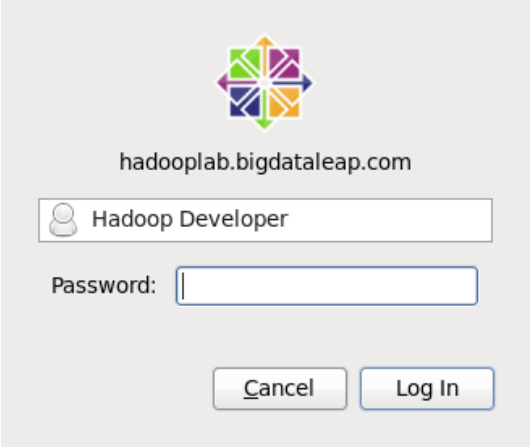
Note: For the following steps all installable are available in the lab\software directory.

III. Install **winscp439setup.exe** (This is an ftp tool) . For MAC, you can use scp tool.
IV. Copy **putty.exe** on to your desktop
V. Start VMWare player or VMWare fusion
VI. Open the VM using
    - **Choose Player -> File -> Open** option
    - Browse to the directory where the ***Spark Training - Base 1.0*** is un-tared and select the .vmx file to open the VM.
    - Click on "**I have copied**" option
    - If you encounter ***VT-x not enabled error***, you need to restart your desktop/laptop and go to BIOS and enable VT-x. The following link will help. http://amiduos.com/support/knowledge-base/article/enabling-virtualization-in-lenovo-systems
    - Select “I Copied It”



    - Select “Hadoop Developer” user

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan



- Enter Password
  - Get the password from the instructor

2) Configuring the VM

a. Get your VM's IP address

Right click on your VM console and select 'open in terminal'

Run ***ifconfig*** and make note of you ip address

```
[hadoop@hadooplab ~]$ ifconfig
eth1      Link encap:Ethernet  HWaddr 00:0C:29:D5:09:72
          inet addr:192.168.217.131  Bcast:192.168.217.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fed5:972/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3272 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4267 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:251040 (245.1 KiB)  TX bytes:957729 (935.2 KiB)
```

b. Configure your VM's hosts file

sudo vi /etc/hosts

Change the following ip address to the one obtained in the previous step

192.168.217.131 hadooplab.bigdataleap.com hadooplab

Save and exit the file. And verify if the settings are working file by running the following command.

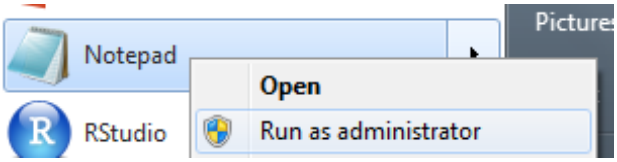
ping hadooplab.bigdataleap.com

If you are getting reply from the VM, then it is configured properly.

c. Configure your laptop/desktop's windows' hosts file

Run notepad in administrator mode. Note: righ click on notepad icon and run it as administrator.

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan



Then do File -> Open from notepad and go to the directory C:\Windows\System32\drivers\etc

And open the hosts file in the notepad

Add the following line as the last line in the file

Note: MAC users should update their /etc/hosts file to add VM's hostname and IP address

192.168.217.131 hadooplab.bigdataleap.com (**Note: change IP to your VM's IP Address)**

Save and exit the file. And verify if the settings are working file by running the following command in the windows command shell.

ping hadooplab.bigdataleap.com

If you are getting reply from the VM, then it is configured properly.

d. Know the directories available in the VM for hands on exercises

Go to lab directory is available in /home/hadoop and list the directories available inside it.

cd /home/hadoop/lab

/home/hadoop/lab contains the following directories and will be used for the following purposes.

lab/data – data required for lab exercises

lab/downloads – all software installable are downloaded and kept here.

lab/software – hadoop, spark and hive is installed in this software. This is the home directory for all software.

lab/programs – all code should be developed and stored here.

lab/cluster – configured for hadoop internal directories to store blocks, fsimages and temporary working file for the framework.

lab/results – all exercise output should be stored here.

3) Starting Spark

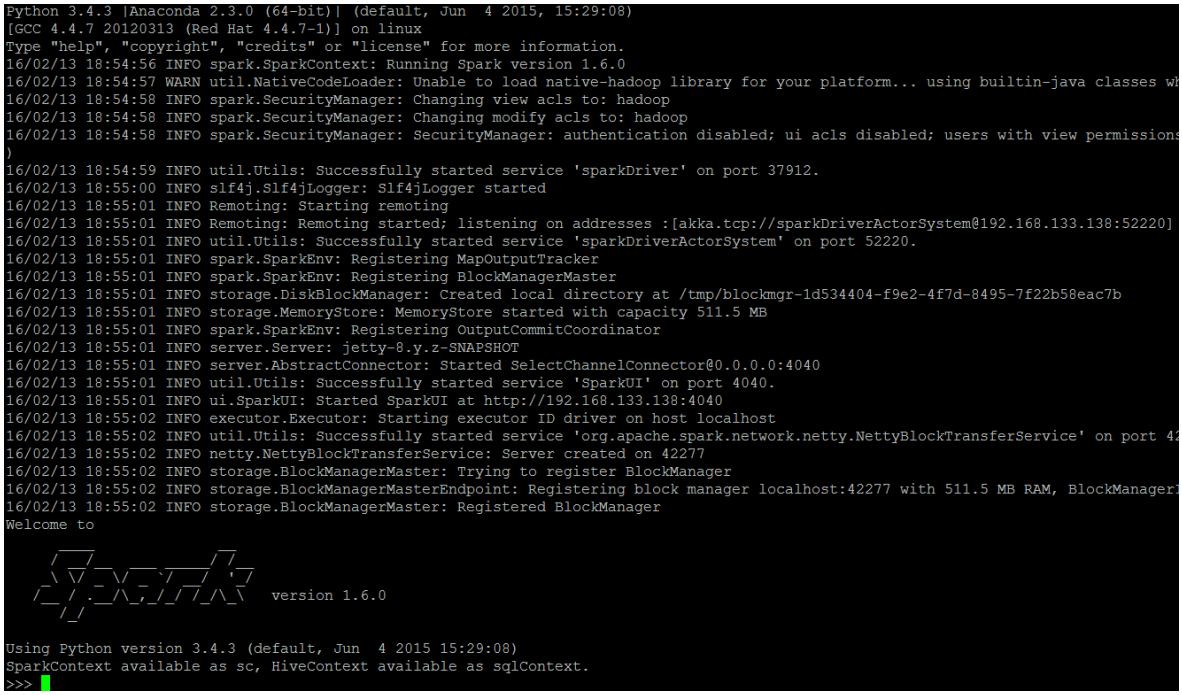
A. Running spark in local mode

Enter the command at linux prompt

pyspark --master local[2]

The spark console should start as shown in the figure below along with Spark Version.

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan



Type the following commands at the spark prompt to verify some more information.

>>> sc.version

'1.6.0'

>>> sc.master

'local[2]'

>>> sc.startTime

1455386096092

>>> sc.pythonVer

'3.4'

>>> sc.pythonExec

'python3'

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

4) Running first spark program on command line

The first program will be a word count problem.

Enter the following lines at ***pyspark*** prompt

```
file = sc.textFile("file:///home/hadoop/lab/data/words")
file.first()
word_tokens = file.flatMap( lambda line: line.split() )
word_tokens.first()

word_ones = word_tokens.map( lambda word: (word, 1 ))
word_ones.first()

word_counts = word_ones.reduceByKey(lambda a, b: a+b)
word_counts.first()
word_counts.foreach(print)
```

5) Using IPython Notebook

- Change directory to /home/hadoop/lab

  cd /home/hadoop

- Start the ipython notbook server

  nohup ipython notebook --profile=pyspark &

- Check the port number of the ipython notebook server port

  tail -f nohup.out

```
[hadoop@sparklab ~]$ tail -f nohup.out
[I 21:16:12.180 NotebookApp] Using MathJax from CDN: https://cdn.mathjax.org/mathjax/latest/MathJax.js
[W 21:16:12.321 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 21:16:12.465 NotebookApp] Serving notebooks from local directory: /home/hadoop
[I 21:16:12.466 NotebookApp] 0 active kernels
[I 21:16:12.466 NotebookApp] The IPython Notebook is running at: http://[all ip addresses on your system]:9998/
[I 21:16:12.466 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Make a note of the port number on which the notebook server is started.

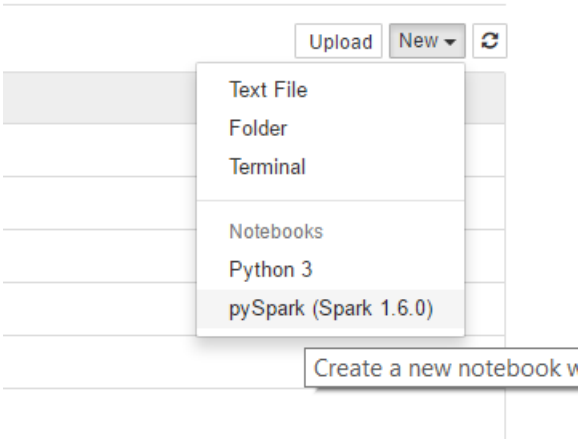
- Open a browser and enter the following URL (Change the port number as shown in the output above)

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

http:// <VM IP Address>:9998/

This should list the directories under /home/hadoop under the VM.

- Then traverse to lab/programs. Drop down on the “new” button available on the right side of the page. And click on pyspark (Spark 1.6.0)



- It should open a new notebook. Click on “Untitled” on top of the page and rename it to “WordCount”.

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

```
In [1]: sc
Out[1]: <pyspark.context.SparkContext at 0x7f668c8ce240>

In [2]: file = sc.textFile("file:///home/hadoop/lab/data/words")

In [3]: file.first()
Out[3]: 'Big data[1][2] is the term for a collection of data sets so large and complex that it beco
        d database management tools or traditional data processing applications. The challenges inc
        earch, sharing, transfer, analysis[4] and visualization. The trend to larger data sets is d
        vable from analysis of a single large set of related data, as compared to separate smaller
        ta, allowing correlations to be found to "spot business trends, determine quality of resear
        tions, combat crime, and determine real-time roadway traffic conditions."[5][6][7]A visuali
        its. At multiple terabytes in size, the text and images of Wikipedia are a classic example

In [4]: word_tokens = file.flatMap( lambda line: line.split() ).map( lambda word: (word, 1 ))

In [5]: word_tokens.first()
Out[5]: ('Big', 1)

In [6]: word_counts = word_tokens.reduceByKey(lambda a, b: a+b)

In [7]: word_counts.first()
Out[7]: ('analysis[4]', 1)

In [9]: tokens = word_counts.collect()

In [10]: tokens[0:10]
Out[10]: [('analysis[4]', 1),
          ('mobile', 1),
          ('Digitization', 1),
          ('before', 1),
          ('trends,', 1),
          ('At', 1),
          ('limitations', 2),
          ('http://www.martinhilbert.net/WorldInfoCapacity.htm', 1),
          ('business', 2),
          ('information-sensing', 1)]
```

- To shut down a notebook, go to ***Running*** tab and click ***shutdown*** against the notebook link.

6) Working with Spark APIs – using Pyspark (Interactive)

The guide for this will be shared before the workshop.

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

7) Spark Programming (Batch): Spark-submit

The **pyspark** program to find the top captains is available as topCaptains.py

- Go to programs directory

  cd /home/hadoop/lab/programs

- Submit the program for execution

  spark-submit --master local[1] --name topcaptains topCaptains.py

- Go to /home/hadoop/lab/results directory. topCaptains directory should have been created.

  cd /home/hadoop/lab/results

- Go to topCaptains directory and list the files

  [hadoop@sparklab topCaptains]$ ls -l

  total 4

  -rw-r--r--. 1 hadoop root 224 Feb 13 22:39 part-00000

  -rw-r--r--. 1 hadoop root 0 Feb 13 22:39 _SUCCESS

- Print the content of the file part-00000

  [hadoop@sparklab topCaptains]$ cat part-00000

  ('Smith G C', 0.61, 0.49)
  ('Fleming S P', 0.45, 0.35)
  ('Border A R', 0.6, 0.34)
  ('Dhoni M S*', 0.55, 0.45)
  ('Waugh S R', 0.63, 0.72)
  ('Cronje W J', 0.71, 0.51)
  ('Ranatunga A', 0.46, 0.21)
  ('Ponting R T', 0.72, 0.62)

8) Working with Spark DataFrames

The guide for this will be shared before the workshop.

9) Stop IPython Notebook

a. Shutdown all IPython notebooks

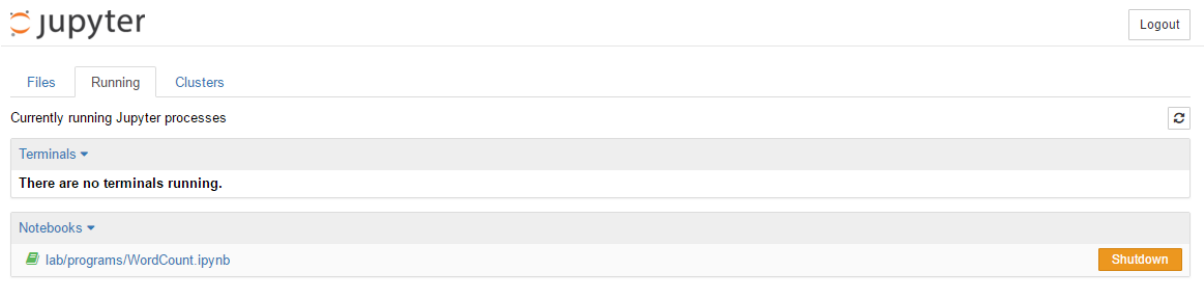
Save the notebooks opened in your browser. And close the notebook windows.

b. Shutdown all IPython notebook server

- **Stop IPython Notebook Process**

  Shut down all the notebooks running

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan



Shutdown all the notebooks running before proceeding to next step.

- Get the process id for the ipython notebook
  Run the following command at linux prompt
  ps -A | grep ipython

```
[hadoop@sparklab ~]$ ps -A | grep ipython
 2390 ?        00:00:09 ipython
[hadoop@sparklab ~]$ kill -9 2390
```

Note down the process id

- Kill the ipython notebook server

  kill -9 <process id>

10) Start Hadoop Services

a. Start HDFS and YARN services

- Go to ***/home/hadoop/lab/software/hadoop-2.7.1/sbin*** directory and type the following command

  cd ***/home/hadoop/lab/software/hadoop-2.7.1/sbin***

  ./start-dfs.sh

  ## Note: verify if all the following three processes have started by typing ***jps*** command

```
2750 NameNode
2964 SecondaryNameNode
2840 DataNode
```

- And then type the following command
  ./start-yarn.sh
- Run jps and verify if all the following processes are running
- Run the history server, which will provide information about completed jobs
  Go to ***/home/hadoop/lab/software/hadoop-2.7.1/sbin*** directory and type the following command

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

./mr-jobhistory-daemon.sh start historyserver

And run jps to confirm if the history server is started or not.

```
[hadoop@hadooplab sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/hadoop/lab/software/hadoop-2.3.0
bigdataleap.com.out
[hadoop@hadooplab sbin]$ jps
3165 DataNode
3286 SecondaryNameNode
5546 Jps
5513 JobHistoryServer
3076 NameNode
3560 ResourceManager
3655 NodeManager
```

➢ If all six processes are running, then hadoop is up and running

11) Working with HDFS

- Listing Directories

  hdfs dfs -ls /

- Creating directory

  hdfs dfs -mkdir /sparklab

- Copying files

  Copy the file ***txnjsonsmall*** from VM's directory to HDFS directory /sparklab

  hdfs dfs -copyFromLocal /home/hadoop/lab/data/txnjsonsmall /sparklab

- Useful File system commands
  hdfs fsck /sparklab/txnjsonsmall -files -blocks -locations

```
Connecting to namenode via http://sparklab.awesomestats.in:50070/fsck?ugi=hadoop&files=1&blocks=1&locations=1&path=%2Fsparklab%2Ftxnjsonsmall
FSCK started by hadoop (auth:SIMPLE) from /192.168.133.129 for path /sparklab/txnjsonsmall at Sun Feb 14 00:42:41 CET 2016
/sparklab/txnjsonsmall 588495 bytes, 1 block(s):  OK
0. BP-1598173478-192.168.229.144-1441355465832:blk_1073742758_1934 len=588495 repl=1 [DatanodeInfoWithStorage[192.168.133.129:50010,DS-9824f1

Status: HEALTHY
 Total size:    588495 B
 Total dirs:    0
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      1 (avg. block size 588495 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    1
 Average block replication:     1.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          1
 Number of racks:               1
FSCK ended at Sun Feb 14 00:42:41 CET 2016 in 5 milliseconds


The filesystem under path '/sparklab/txnjsonsmall' is HEALTHY
```

The file to blocks mapping are shown as a result of the above command.

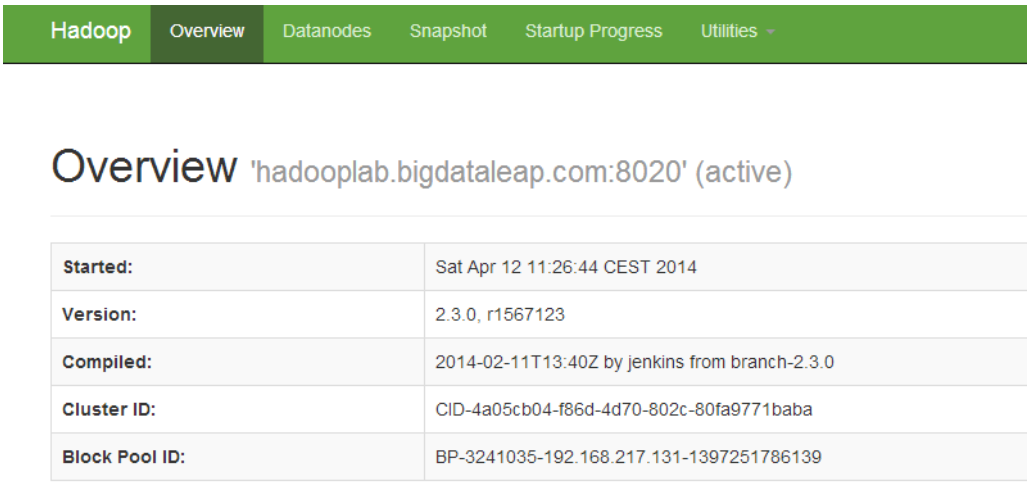
- HDFS Web UI

  Open your browser & enter the following url

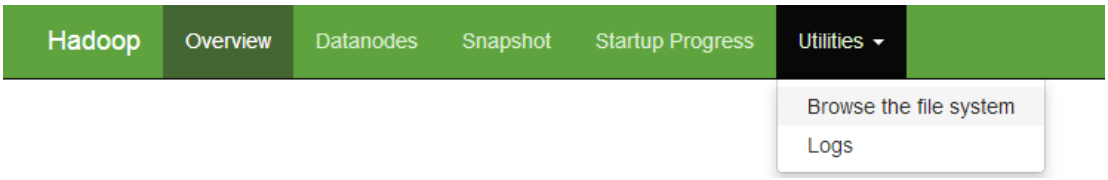
Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

http://hadooplab.bigdataleap.com:50070/

(you can replace the domain with IP address of you vm, if it is not working)



- File system explorer and log explorer is available under utilities menu



12) Configuring Spark to run on YARN mode

- **Configure .bash_profile**

  Go to home directory of your vm

  cd /home/hadoop

  Open the .bash_profile file from ***WinSCP*** or using ***vi*** editor

  Copy & paste the whole "***export PYSPARK_SUBMIT_ARGS***" line to a new line

  comment the current "***export PYSPARK_SUBMIT_ARGS***" settings

  Change the deployment mode in the other "***export PYSPARK_SUBMIT_ARGS***" line from ***local[2]*** to ***yarn-client*** *(as shown below)*

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

```
PATH=$PATH:$HOME/bin:/home/hadoop/lab/software/hadoop-2.7.1/sbin
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64
export HADOOP_INSTALL=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_COMMON_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_HDFS_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_MAPRED_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_YARN_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_CONF_DIR=/home/hadoop/lab/software/hadoop-2.7.1/etc/hadoop
export YARN_CONF_DIR=$HADOOP_CONF_DIR
export PATH=$PATH:$HADOOP_INSTALL/bin
export SQOOP_HOME=/home/hadoop/lab/software/sqoop-1.4.4.bin__hadoop-2.0.4-alpha
export PATH=$PATH:$SQOOP_HOME/bin
export HIVE_HOME=/home/hadoop/lab/software/apache-hive-1.2.1-bin
export PATH=$PATH:$HIVE_HOME/bin
export PIG_INSTALL=/home/hadoop/lab/software/pig-0.12.0
export OOZIE_HOME=/home/hadoop/lab/software/oozie-4.0.0
export PATH=$PATH:$PIG_INSTALL/bin:$OOZIE_HOME/bin
export PATH=$PATH:/home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6/bin
#export IPYTHON=1
#export IPYTHON_OPTS="notebook"
export SPARK_HOME=/home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6
#export PYSPARK_SUBMIT_ARGS="--master local[2] pyspark-shell --packages com.databricks:sp
export PYSPARK_SUBMIT_ARGS="--master yarn-client pyspark-shell --packages com.databricks:
export PYSPARK_PYTHON=python3
export HADOOP_CMD="/home/hadoop/lab/software/hadoop-2.7.1/bin"
export HADOOP_STREAMING="/home/hadoop/lab/software/hadoop-2.7.1/share/hadoop/tools/lib/ha
```

save and close.

- **Run the bash_profile**

  Go the putty terminal and change directory to /home/hadoop

  cd /home/hadoop

  Then run .bash_profile

  . .bash_profile

  Run the following command at the linux prompt

  export

  It should show the new settings for PYSPARK_SUBMIT_ARGS

- **Re-Start the ipython notbook server**

  cd /home/hadoop

  rm nohup.out

  nohup ipython notebook --profile=pyspark &

- Check the port number of the ipython notebook server port

  tail -f nohup.out

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

```
[hadoop@sparklab ~]$ tail -f nohup.out
[I 21:16:12.180 NotebookApp] Using MathJax from CDN: https://cdn.mathjax.org/mathjax/latest/MathJax.js
[W 21:16:12.321 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 21:16:12.465 NotebookApp] Serving notebooks from local directory: /home/hadoop
[I 21:16:12.466 NotebookApp] 0 active kernels
[I 21:16:12.466 NotebookApp] The IPython Notebook is running at: http://[all ip addresses on your system]:9998/
[I 21:16:12.466 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Make a note of the port number on which the notebook server is started.

- Open a browser and enter the following URL (Change the port number as shown in the output above)

  http:// <VM IP Address>:9998/

  This should list the directories under ***/home/hadoop*** under the VM.

  Traverse to the ***lab/programs*** folder

13) Working with Hadoop: HDFS, YARN & Spark SQL

The guide for this will be shared before the workshop.

14) Monitoring & Debugging

The guide for this will be shared before the workshop.

15) Shutdown Hadoop cluster

a. Stop HDFS and YARN services

- Go to ***/home/hadoop/lab/software/hadoop-2.7.1/sbin*** directory and type the following command

  cd ***/home/hadoop/lab/software/hadoop-2.7.1/sbin***

  ./stop-dfs.sh

- And then type the following command
  ./stop-yarn.sh
- Stop the history server

  ./mr-jobhistory-daemon.sh stop historyserver

- Run jps and make sure that all six processes are stopped.

b. Configure .bash_profile

Open the .bash_profile file from WinSCP

Commen the line “export PYSPARK_SUBMIT_ARGS” which contains yarn-client and uncomment the line which contain local[2]

save and close.

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

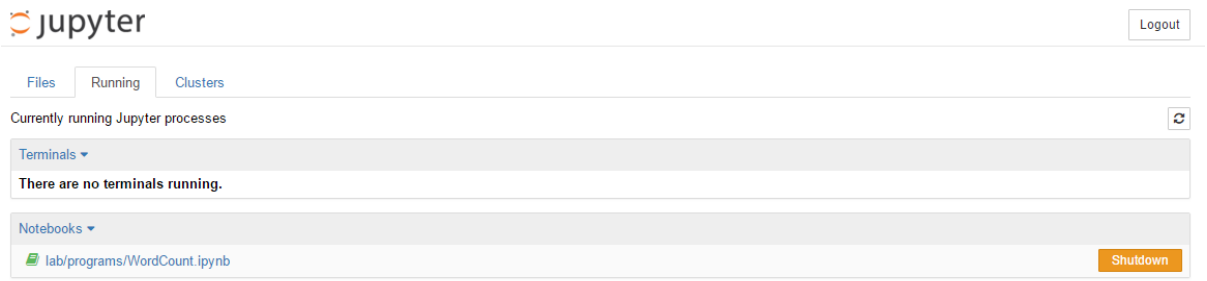
- **Run the bash_profile**

  Run the following command at the command prompt.

  . .bash_profile

c. Stop and restart IPython Notebook Process

Shut down all the notebooks running



Shutdown all the notebooks running before proceeding to next step.

- Get the process id for the ipython notebook
  Run the following command at linux prompt
  ps -A | grep ipython

```
[hadoop@sparklab ~]$ ps -A | grep ipython
 2390 ?        00:00:09 ipython
[hadoop@sparklab ~]$ kill -9 2390
```

  Note down the process id

- Kill the ipython notebook server

  kill -9 <process id>

- **Re-Start the ipython notbook server**

  cd /home/hadoop

  rm nohup.out

  nohup ipython notebook --profile=pyspark &

- Check the port number of the ipython notebook server port

  tail -f nohup.out

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

```
[hadoop@sparklab ~]$ tail -f nohup.out
[I 21:16:12.180 NotebookApp] Using MathJax from CDN: https://cdn.mathjax.org/mathjax/latest/MathJax.js
[W 21:16:12.321 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 21:16:12.465 NotebookApp] Serving notebooks from local directory: /home/hadoop
[I 21:16:12.466 NotebookApp] 0 active kernels
[I 21:16:12.466 NotebookApp] The IPython Notebook is running at: http://[all ip addresses on your system]:9998/
[I 21:16:12.466 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

Make a note of the port number on which the notebook server is started.

- Open a browser and enter the following URL (Change the port number as shown in the output above)

  http:// <VM IP Address>:9998/

  This should list the directories under ***/home/hadoop*** under the VM.

  Traverse to the ***lab/programs*** folder

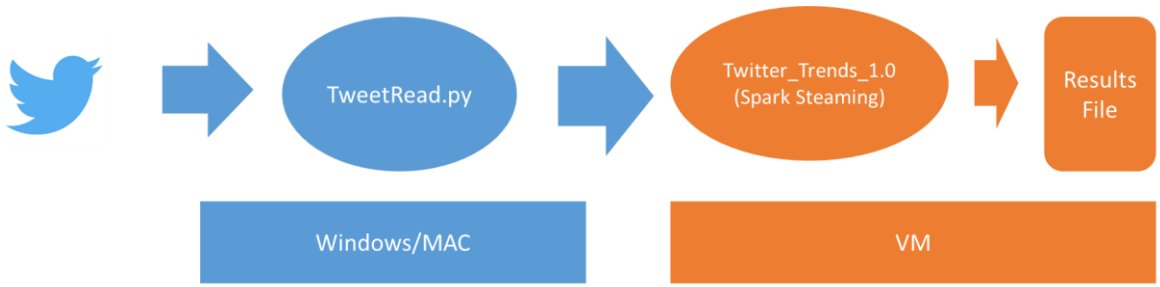
16) Working with log files and spark sql functions

The lab guide will be provided by the instructor.

17) Using Spark Streaming

a. Configure twitter account

- Get the ip address of your local machine (Windows or MAC)
- Ping the ip address from the spark VM. It should get replies.
- Open the command terminal on your host machine (Windows or MAC)
- Go to the directory spark/code (Under your lab distribution folder)
- Follow the diagram below to understand how to accomplish this task.



- Acts as a broker.
- Reads the tweets
- Send to the clients, who has connected to this broker (on a tcp port).

- Spark Streaming Application pulling the messages from the broker and processing it.
- Extracts the tags from tweets and counts it over a period (defined as window).
- Writes the tags and their counts to a file.

Step 1:

Open a twitter account or sign in to your twitter account

Step 2:

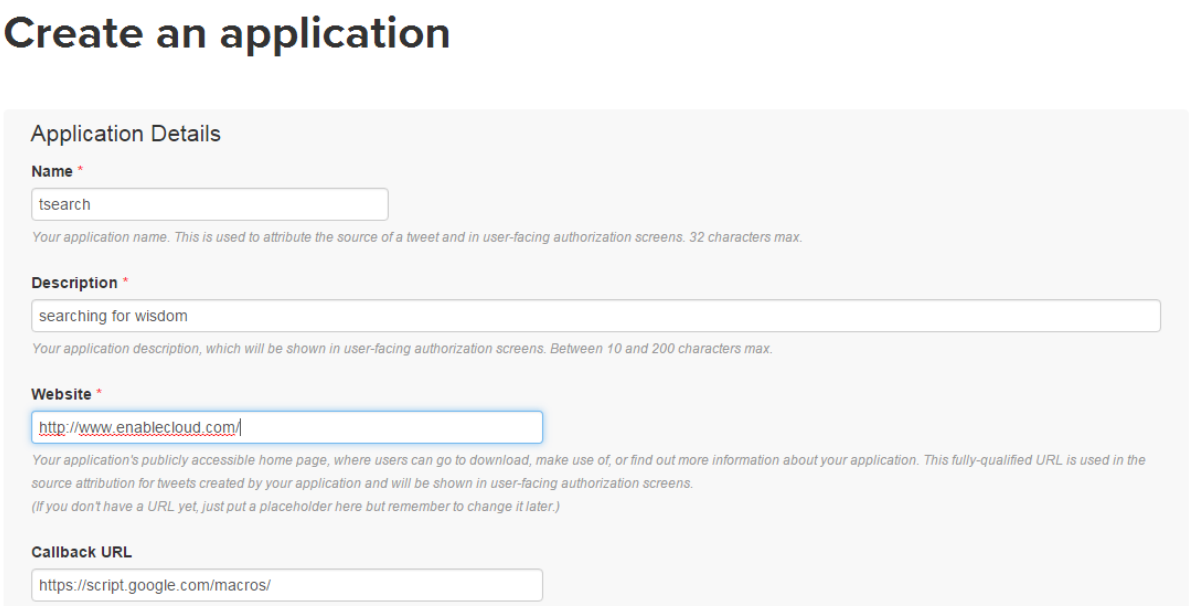
Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

Go to https://apps.twitter.com/

Click on ***"Create New App"***

Enter the details. (The name should be unique).

You can enter any url or skip it.



Once application is created successfully, click on the application and click on ***"Keys and Access Tokens"***

Copy all details to a notepad

- Consumer API key and secret
- Access token and secret

b. Start TweetRead Service

- Go to the directory spark/code (Under your lab distribution folder of your windows or mac machine)
- Open the program TweetRead.py in an editor and modify the following parameters
  - Change the host IP address to the IP address of your windows or MAC Machine
  - Update the variables with appropriate tokens. And save the program.
- Go the terminal and run the program TweetRead.py as follows

  python TweetRead.py

c. Write the twitter stream code

- Go to http://www.awesomestats.in/spark-twitter-stream/
- Start a new ipython notebook and start writing the code
- Change the IP address of the following line to the IP address of your windows or mac machine. (Where TweetRead.py is listening)
  - lines = ssc.socketTextStream("172.19.96.85", 5555)
- Change the search key word to filer twitter feeds

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

- Then execute the steps one by one
- The last step will run indefinitely.
- The last step will show the trend chart and keep updating it every 10 seconds

Step 8:

After the lab is completed, stop all processes.

- Stop TweetRead.py running on windows and mac machine. (***Ctrl + c*** will work)
- Stop the ipython notebook. Select kernel -> interrupt
- Then execute the last stop ***ssc.stop()***
- Go to the notebook tree page and go to the **"running"** tab and shutdown the application

- Go to the VM putty session and stop the tail -f results command. (***Ctrl + c*** will work)

18) Visualization, Statistics & Machine Learning Libraries

The guide for this will be shared before the workshop.

19) Assignment: Visualization, Statistics & Machine Learning Library

The guide for this will be shared before the workshop.

20) Appendix A: Configuring Spark

A. Install Spark

- Go to software installation directory

cd /home/hadoop/lab/software

- Untar the spark installable

tar -xvf /home/hadoop/lab/downloads/spark-1.6.0-bin-hadoop2.6.tgz

B. Configure spark

- Configure the paths in .bash_profile

This is already configured. So, skip this step. This is only given for your reference.

```
PATH=$PATH:$HOME/bin:/home/hadoop/lab/software/hadoop-2.7.1/sbin
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64
export HADOOP_INSTALL=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_COMMON_HOME=/home/hadoop/lab/software/hadoop-2.7.1
```


Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

```
export HADOOP_HDFS_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_MAPRED_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_YARN_HOME=/home/hadoop/lab/software/hadoop-2.7.1
export HADOOP_CONF_DIR=/home/hadoop/lab/software/hadoop-2.7.1/etc/hadoop
export YARN_CONF_DIR=$HADOOP_CONF_DIR
export PATH=$PATH:$HADOOP_INSTALL/bin
export SQOOP_HOME=/home/hadoop/lab/software/sqoop-1.4.4.bin__hadoop-2.0.4-alpha
export PATH=$PATH:$SQOOP_HOME/bin
export HIVE_HOME=/home/hadoop/lab/software/apache-hive-1.2.1-bin
export PATH=$PATH:$HIVE_HOME/bin
export PIG_INSTALL=/home/hadoop/lab/software/pig-0.12.0
export OOZIE_HOME=/home/hadoop/lab/software/oozie-4.0.0
export PATH=$PATH:$PIG_INSTALL/bin:$OOZIE_HOME/bin
export PATH=$PATH:/home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6/bin
#export IPYTHON=1
#export IPYTHON_OPTS="notebook"
export SPARK_HOME=/home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6
export PYSPARK_SUBMIT_ARGS="--master local[2] pyspark-shell --packages
com.databricks:spark-csv_2.10:1.3.0 --jars /home/hadoop/lab/software/apache-hive-1.2.1-
bin/lib/*,/home/hadoop/lab/software/apache-hive-1.2.1-bin/lib/mysql-connector-java-5.1.30-
bin.jar --file /home/hadoop/lab/software/apache-hive-1.2.1-bin/conf/hive-site.xml"
export PYSPARK_PYTHON=python3
export HADOOP_CMD="/home/hadoop/lab/software/hadoop-2.7.1/bin"
export HADOOP_STREAMING="/home/hadoop/lab/software/hadoop-
2.7.1/share/hadoop/tools/lib/hadoop-streaming-2.7.1.jar"
```

- Configure spark default configs
  A template config file is available in lab/template directory
  Copy the file into spark's **$SPARK_HOME/conf** directory.

  cp /home/hadoop/lab/templates/spark-defaults.conf /home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6/conf/

21) Appendix: Configuring Hadoop

All directory paths are under home directory ***/home/hadoop***

d. Untar Hadoop jar file

Note: Change your directory to lab/software and untar the hadoop tar file from lab/downloads directory into lab/software folder.
Follow the following steps

- Go to lab/software
  cd /home/hadoop/lab/software
- Untar Hadoop files into software folder
  tar -xvf /home/hadoop/lab/downloads/hadoop-2.3.0.tar.gz
- *Browse through the directories and check which subdirectory contains what files*

e. Set up .bash_profile

(Note: Skip this step. This is already configured. This is only given for your understanding.)

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

- Open .bash_profile file under home directory

  cd /home/hadoop

  vi .bash_profile

  Enter the following settings

```
PATH=$PATH:$HOME/bin
export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64
export HADOOP_INSTALL=/home/hadoop/lab/software/hadoop-2.3.0
export HADOOP_COMMON_HOME=/home/hadoop/lab/software/hadoop-2.3.0
export HADOOP_HDFS_HOME=/home/hadoop/lab/software/hadoop-2.3.0
export HADOOP_MAPRED_HOME=/home/hadoop/lab/software/hadoop-2.3.0
export HADOOP_YARN_HOME=/home/hadoop/lab/software/hadoop-2.3.0
export HADOOP_CONF_DIR=/home/hadoop/lab/software/hadoop-2.3.0/etc/hadoop
export YARN_CONF_DIR=$HADOOP_CONF_DIR
export PATH=$PATH:$HADOOP_INSTALL/bin
export SQOOP_HOME=/home/hadoop/lab/software/sqoop-1.4.4.bin__hadoop-2.0.4-alpha
export PATH=$PATH:$SQOOP_HOME/bin
export HIVE_HOME=/home/hadoop/lab/software/apache-hive-0.13.0-bin
export PATH=$PATH:$HIVE_HOME/bin
export PIG_INSTALL=/home/hadoop/lab/software/pig-0.12.0
export OOZIE_HOME=/home/hadoop/lab/software/oozie-4.0.0
export PATH=$PATH:$PIG_INSTALL/bin:$OOZIE_HOME/bin
export PATH
```

- Save and exit .bash_profile
- run following command

  . .bash_profile

- Verify whether variables are defined or not by typing ***export*** at command prompt

  export

- Check the following versions

  java –version

```
[hadoop@hadooplab ~]$ java -version
java version "1.7.0_51"
OpenJDK Runtime Environment (rhel-2.4.4.1.el6_5-x86_64 u51-b02)
OpenJDK 64-Bit Server VM (build 24.45-b08, mixed mode)
```

hadoop version

```
[hadoop@hadooplab ~]$ hadoop version
Hadoop 2.3.0
Subversion http://svn.apache.org/repos/asf/hadoop/common -r 1567123
Compiled by jenkins on 2014-02-11T13:40Z
Compiled with protoc 2.5.0
From source with checksum dfe46336fbc6a044bc124392ec06b85
This command was run using /home/hadoop/lab/software/hadoop-2.3.0/share/
```

f. Configuring pseudo-distributed mode

Go to conf directory of hadoop installation folder

cd /home/hadoop/lab/software/hadoop-2.3.0/etc/hadoop

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

Note: You need not type the following files. The following files are already available in the **lab/references** folder on your windows or mac machine, where you copied the contents of the USB drive. You can transfer these files from your windows machine to your VM using WinSCP or scp command in MAC.

Note for MAC Users: People using MAC machine can use scp command

- core-site.xml

```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://hadooplab.bigdataleap.com:8020/</value>
</property>
</configuration>
```

- hdfs-site.xml

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.blocksize</name>
<value>67108864</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///home/hadoop/lab/cluster/hdfs/nn</value>
</property>
<property>
<name>fs.checkpoint.dir</name>
<value>file:///home/hadoop/lab/cluster/hdfs/snn</value>
</property>
<property>
<name>dfs.namenode.checkpoint.period</name>
<value>3600</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:///home/hadoop/lab/cluster/hdfs/dn</value>
</property>
<property>
<name>dfs.namenode.secondary.http-address</name>
<value>hadooplab.bigdataleap.com:50090</value>
</property>
</configuration>
```

- yarn-site.xml

Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

```
<configuration>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>hadooplab.bigdataleap.com:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>hadooplab.bigdataleap.com:8088</value>
  </property>
  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>/home/hadoop/lab/cluster/yarn/local</value>
  </property>
  <property>
    <name>yarn.nodemanager.remote-app-log-dir</name>
    <value>/home/hadoop/lab/cluster/yarn/remote</value>
  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/home/hadoop/lab/cluster/yarn/logs</value>
  </property>
  <property>
    <name>yarn.nodemanager.resource.memory-mb</name>
    <value>3072</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
  <property>
    <name>yarn.scheduler.maximum-allocation-mb</name>
    <value>3072</value>
  </property>
  <property>
    <name>yarn.scheduler.minimum-allocation-mb</name>
    <value>300</value>
  </property>
  <property>
   <name>yarn.nodemanager.vmem-check-enabled</name>
   <value>false</value>
  </property>
  <property>
   <name>yarn.log.server.url</name>
   <value>http://hadooplab.bigdataleap.com:19888/jobhistory/logs</value>
  </property>
  <property>
    <name>yarn.nodemanager.vmem-pmem-ratio</name>
    <value> 4 </value>
  </property>
</configuration>
```


Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

- mapred-site.xml

```
<configuration>
<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>
<property>
    <name>mapreduce.cluster.local.dir</name>
    <value>/home/hadoop/lab/cluster/mr/local</value>
</property>
<property>
    <name>mapreduce.map.memory.mb</name>
    <value>300</value>
</property>
<property>
    <name>mapreduce.reduce.memory.mb</name>
    <value>300</value>
</property>
<property>
    <name>mapreduce.map.java.opts</name>
    <value>-Xmx300m</value>
</property>
<property>
    <name>mapreduce.reduce.java.opts</name>
    <value>-Xmx300m</value>
</property>
<property>
    <name>mapreduce.jobhistory.webapp.address</name>
    <value>hadooplab.bigdataleap.com:19888</value>
</property>
<property>
<property>
    <name>mapreduce.map.log.level</name>
    <value>INFO</value>
</property>
<property>
    <name>mapreduce.reduce.log.level</name>
    <value>INFO</value>
</property>
<property>
    <name>yarn.app.mapreduce.am.resource.mb</name>
    <value>300</value>
</property>
<property>
        <name>mapreduce.cluster.administrators</name>
        <value>hadoop</value>
</property>
<property>
        <name>mapreduce.reduce.log.level</name>
        <value>INFO</value>
</property>
<property>
        <name>mapreduce.map.log.level</name>
        <value>INFO</value>
</property>
</configuration>
```


Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan

g. Copy the 64 bit libraries

- Copy the 64 bit native libraries
  Go to the following directory
  cd /home/hadoop/lab/downloads/lib64bit/

cp libhadoop.so.1.0.0 $HADOOP_INSTALL/lib/native/

cp libhdfs.so.0.0.0 $HADOOP_INSTALL/lib/native/

h. Configure JAVA_HOME

- Go to ***/home/hadoop/lab/software/hadoop-2.3.0/etc/hadoop*** directory

  Enter the following line

  export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64

  at the beginning of all the following files:

  - hadoop-env.sh
  - mapred-env.sh
  - yarn-env.sh

  Note: Comment the existing JAVA_HOME line if already there.

i. Format the namenode

- Enter the following command at prompt
  (Note: Please type the command on your putty terminal, do not copy and paste)

  hdfs namenode -format

- Go to /home/hadoop/lab/cluster/hdfs/nn/current directory and verify whether all files have been created.
  - fsimage (file system image) and it's md5 file (fingerprint)
  - VERSION (contains unique cluster, layout version and other details…)

```
[hadoop@hadooplab hadoop]$ cd /home/hadoop/lab/cluster/hdfs/nn/current/
[hadoop@hadooplab current]$ ls -l
total 16
-rw-r--r--. 1 hadoop root 218 Apr 29 13:58 fsimage_0000000000000000000
-rw-r--r--. 1 hadoop root  62 Apr 29 13:58 fsimage_0000000000000000000.md5
-rw-r--r--. 1 hadoop root   2 Apr 29 13:58 seen_txid
-rw-r--r--. 1 hadoop root 207 Apr 29 13:58 VERSION
```


Manaranjan Pradhan
manaranjan@enablecloud.com
https://www.linkedin.com/in/manaranjanpradhan