1) STARTING SPARK 2

2) RUNNING FIRST SPARK PROGRAM ON COMMAND LINE 4

3) USING IPYTHON NOTEBOOK 4

4) WORKING WITH SPARK APIS – USING PYSPARK (INTERACTIVE) 6

5) SPARK PROGRAMMING (BATCH): SPARK-SUBMIT 7

6) PARTICIPANT’S ASSIGNMENT 7

7) WORKING WITH SPARK DATAFRAMES 7

8) START HADOOP SERVICES 8

9) WORKING WITH HDFS 9

10) CONFIGURING SPARK TO RUN ON YARN MODE 10

11) WORKING WITH HADOOP: HDFS, YARN & SPARK SQL 12

12) MONITORING & DEBUGGING 12

13) SHUTDOWN HADOOP CLUSTER 13

14) PARTICIPANT’S ASSIGNMENT 2: WORKING WITH SPARK DATAFRAMES 14

15) USING SPARK STREAMING 14

16) VISUALIZATION, STATISTICS & MACHINE LEARNING LIBRARIES 17

17) ASSIGNMENT: VISUALIZATION, STATISTICS & MACHINE LEARNING LIBRARY 17

18) APPENDIX A: CONFIGURING SPARK 17

19) APPENDIX: CONFIGURING HADOOP 18

1) Starting Spark

Note: Step A and B are already completed. So, directly go to step C.

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
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.

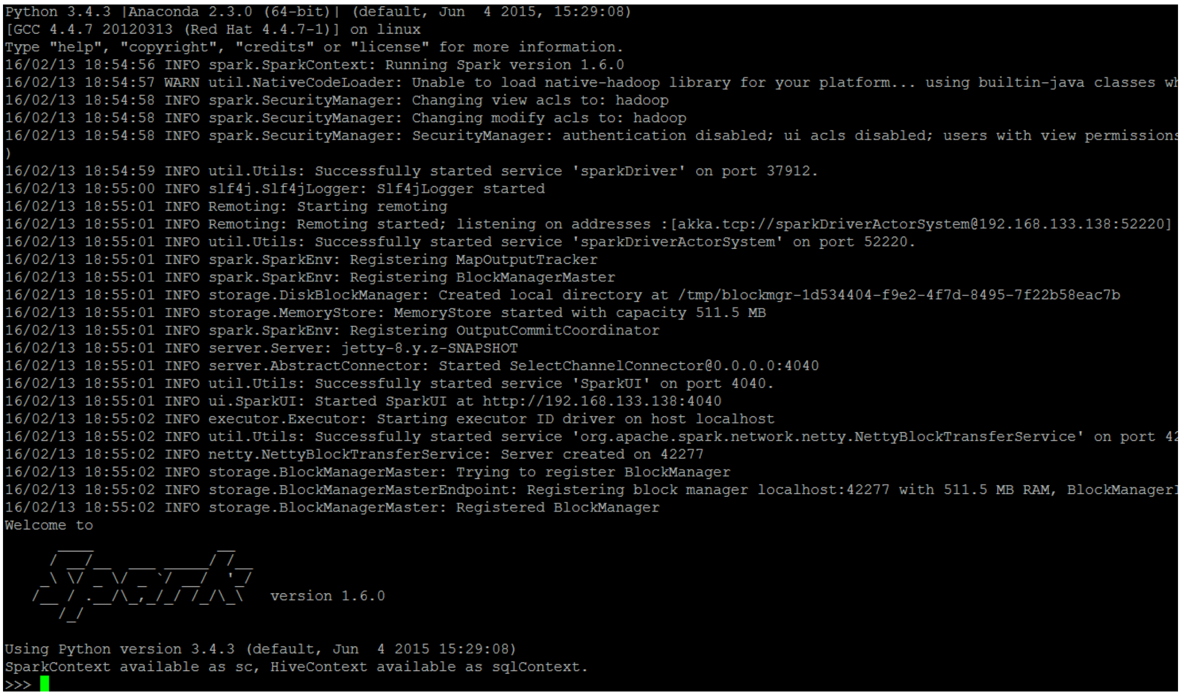
cp /home/hadoop/lab/templates/spark-defaults.conf /home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6/conf/

C. Check Spark Version

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.



```
Python 3.4.3 |Anaconda 2.3.0 (64-bit)| (default, Jun  4 2015, 15:29:08)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
16/02/13 18:54:56 INFO spark.SparkContext: Running Spark version 1.6.0
16/02/13 18:54:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes wh
16/02/13 18:54:58 INFO spark.SecurityManager: Changing view acls to: hadoop
16/02/13 18:54:58 INFO spark.SecurityManager: Changing modify acls to: hadoop
16/02/13 18:54:58 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions
)
16/02/13 18:54:59 INFO util.Utils: Successfully started service 'sparkDriver' on port 37912.
16/02/13 18:55:00 INFO slf4j.Slf4jLogger: Slf4jLogger started
16/02/13 18:55:01 INFO Remoting: Starting remoting
16/02/13 18:55:01 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkDriverActorSystem@192.168.133.138:52220]
16/02/13 18:55:01 INFO util.Utils: Successfully started service 'sparkDriverActorSystem' on port 52220.
16/02/13 18:55:01 INFO spark.SparkEnv: Registering MapOutputTracker
16/02/13 18:55:01 INFO spark.SparkEnv: Registering BlockManagerMaster
16/02/13 18:55:01 INFO storage.DiskBlockManager: Created local directory at /tmp/blockmgr-1d534404-f9e2-4f7d-8495-7f22b58eac7b
16/02/13 18:55:01 INFO storage.MemoryStore: MemoryStore started with capacity 511.5 MB
16/02/13 18:55:01 INFO spark.SparkEnv: Registering OutputCommitCoordinator
16/02/13 18:55:01 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/02/13 18:55:01 INFO server.AbstractConnector: Started SelectChannelConnector@0.0.0.0:4040
16/02/13 18:55:01 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
16/02/13 18:55:01 INFO ui.SparkUI: Started SparkUI at http://192.168.133.138:4040
16/02/13 18:55:02 INFO executor.Executor: Starting executor ID driver on host localhost
16/02/13 18:55:02 INFO util.Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 42
16/02/13 18:55:02 INFO netty.NettyBlockTransferService: Server created on 42277
16/02/13 18:55:02 INFO storage.BlockManagerMaster: Trying to register BlockManager
16/02/13 18:55:02 INFO storage.BlockManagerMasterEndpoint: Registering block manager localhost:42277 with 511.5 MB RAM, BlockManagerI
16/02/13 18:55:02 INFO storage.BlockManagerMaster: Registered BlockManager
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Python version 3.4.3 (default, Jun  4 2015 15:29:08)
SparkContext available as sc, HiveContext available as sqlContext.
>>>
```

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'
```

2) 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() ).map( lambda word: (word, 1 ))

word_tokens.first()

word_counts = word_tokens.reduceByKey(lambda a, b: a+b)

word_counts.first()

word_counts.foreach(print)
```

3) Using IPython Notebook

- Change directory to /home/hadoop/lab

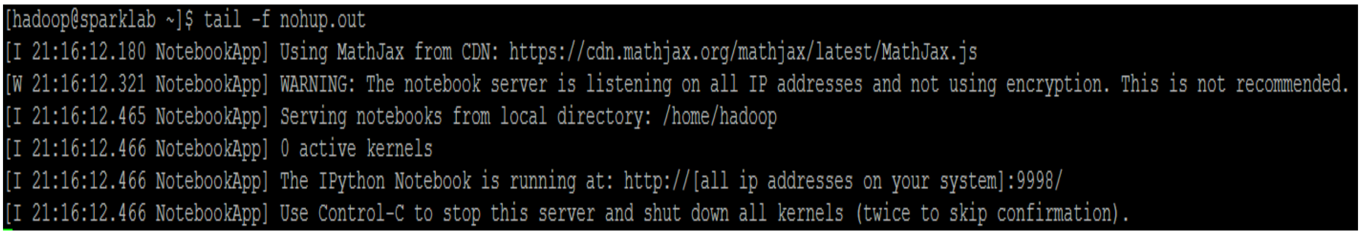
  ```
  cd /home/hadoop/lab
  ```

- Start the ipython notbook server

  ```
  nohup ipython notebook --profile=pyspark &
  ```

- Check the port number of the ipython notebook server port

  ```
  tail -f nohup.out
  ```



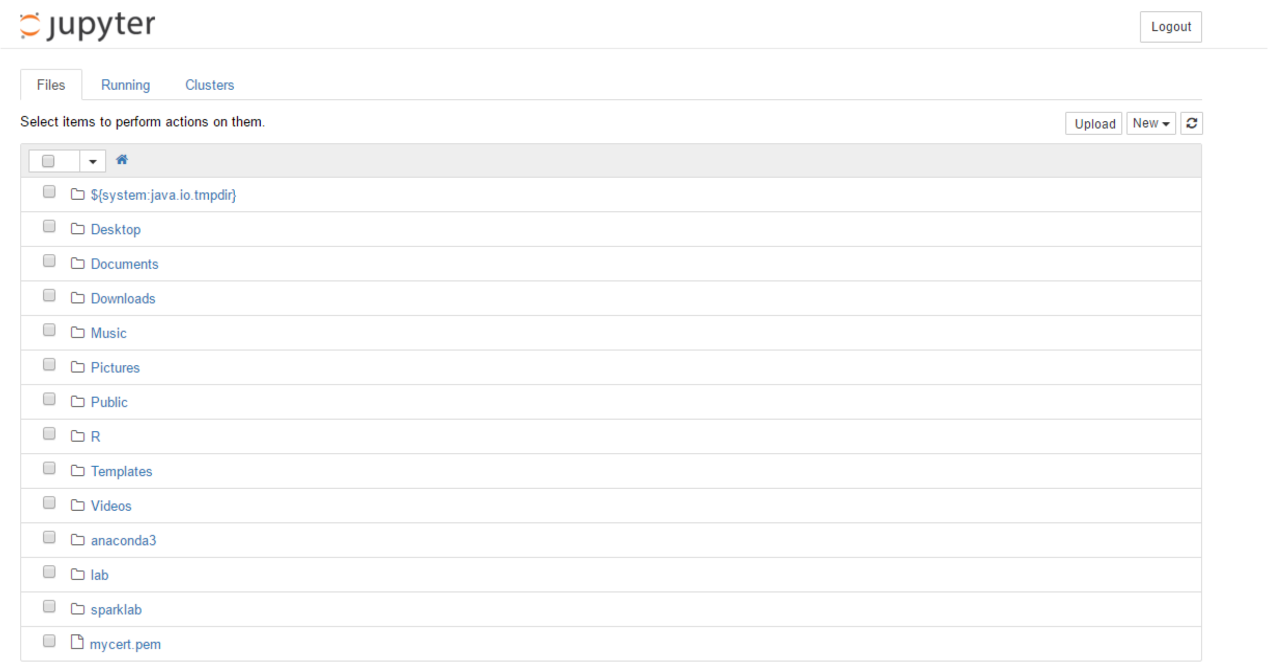
```
[hadoop@sparklab ~]$ tail -f nohup.out
[I 21:16:12.180 NotebookApp] Using MathJax from CDN: https://cdn.mathjax.org/mathjax/latest/MathJax.js
[W 21:16:12.321 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommended.
[I 21:16:12.465 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.

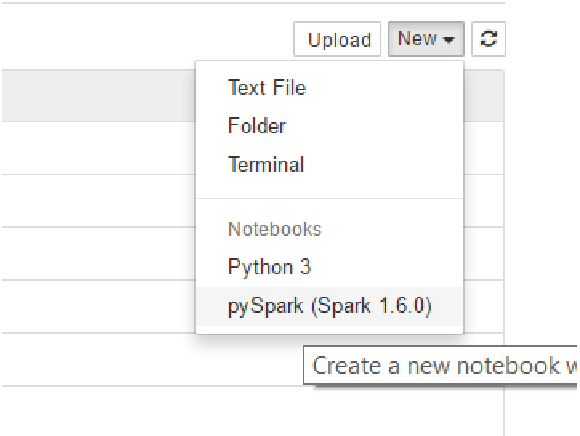
- Open a browser and enter the following URL (Change the port number as shown in the output above)

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

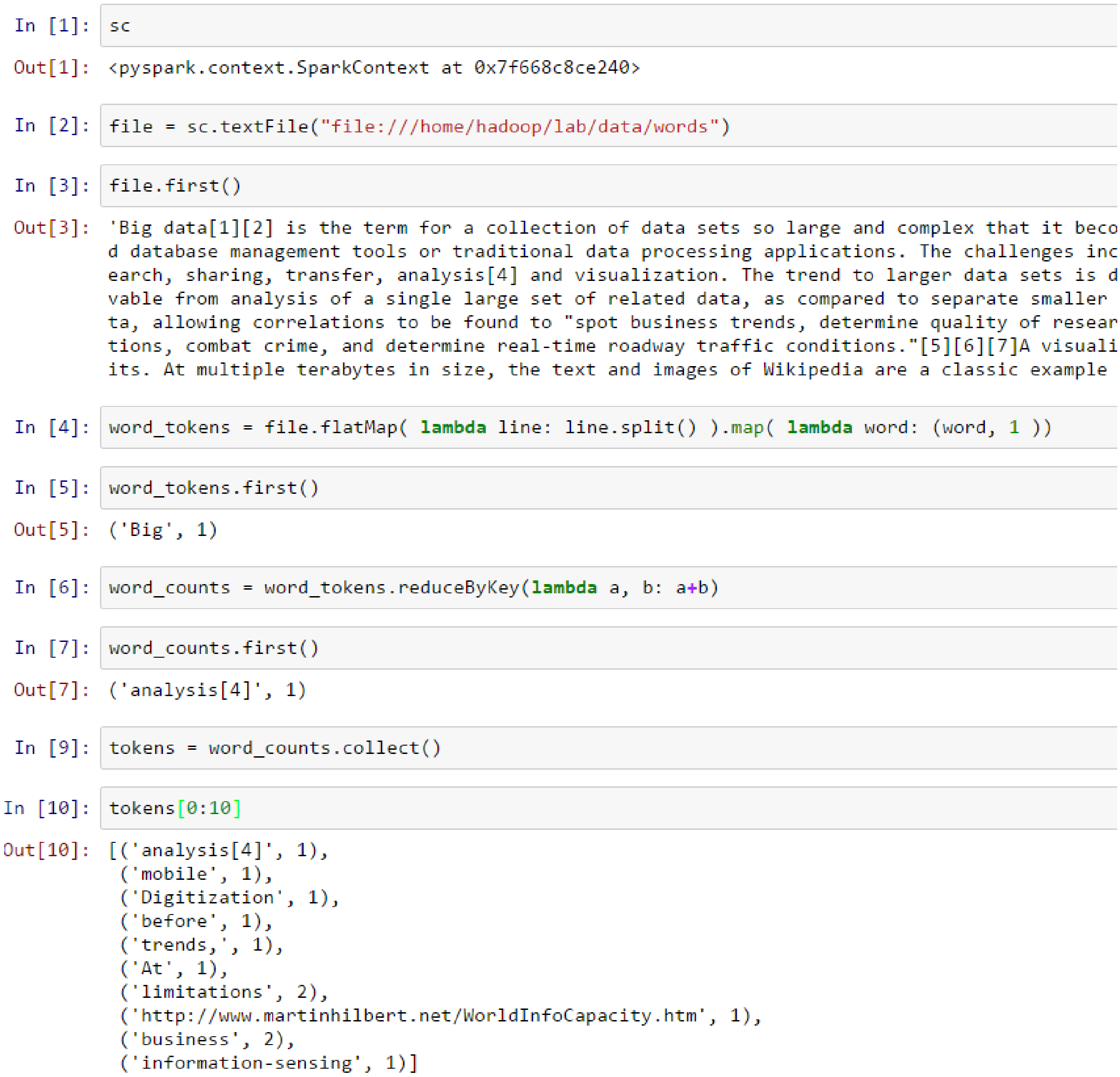
  This should list the directories under /home/hadoop under the VM.



- 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)

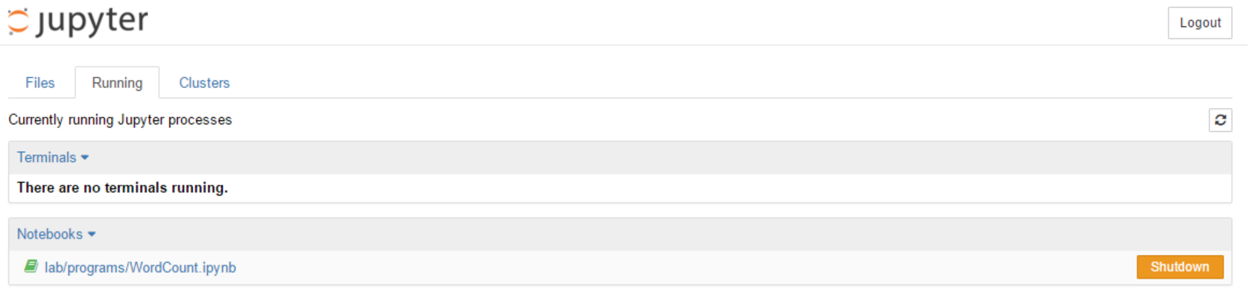


- It should open a new notebook. Click on “Untitled” on top of the page and rename it to “WordCount”.



```
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.



Currently running Jupyter processes

There are no terminals running.

lab/programs/WordCount.ipynb

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

The guide for this will be shared before the workshop.

5) 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)
  ```

6) Participant's Assignment

Use the RDD APIs to analyse these data.

7) Working with Spark DataFrames

The guide for this will be shared before the workshop.

8) Start Hadoop Services

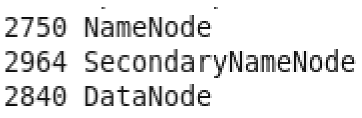
a. Start HDFS and YARN services

- 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*

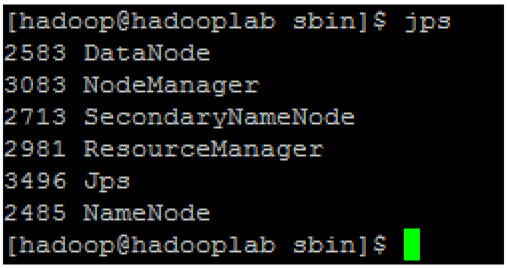
  ./start-dfs.sh

  ## 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

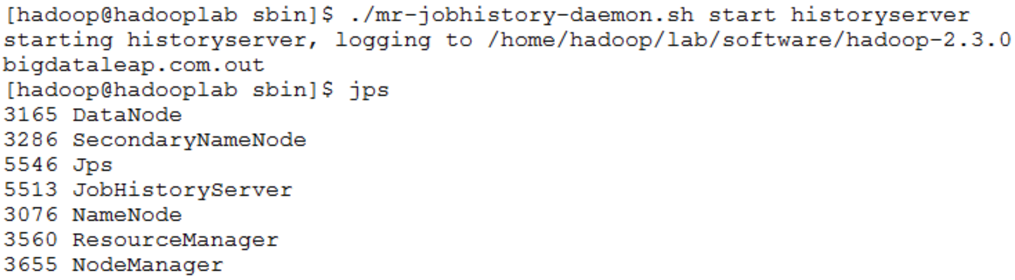


```
[hadoop@hadooplab sbin]$ jps
2583 DataNode
3083 NodeManager
2713 SecondaryNameNode
2981 ResourceManager
3496 Jps
2485 NameNode
[hadoop@hadooplab sbin]$
```

- Run the history server, which will provide information about completed jobs
  Go to ***/home/hadoop/lab/software/hadoop-2.3.0/sbin*** directory and type the following command

  ./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

9) Working with HDFS

- Listing Directories

  hdfs dfs -ls /

- Creating directory

  hdfs dfs -mkdir /sparklab

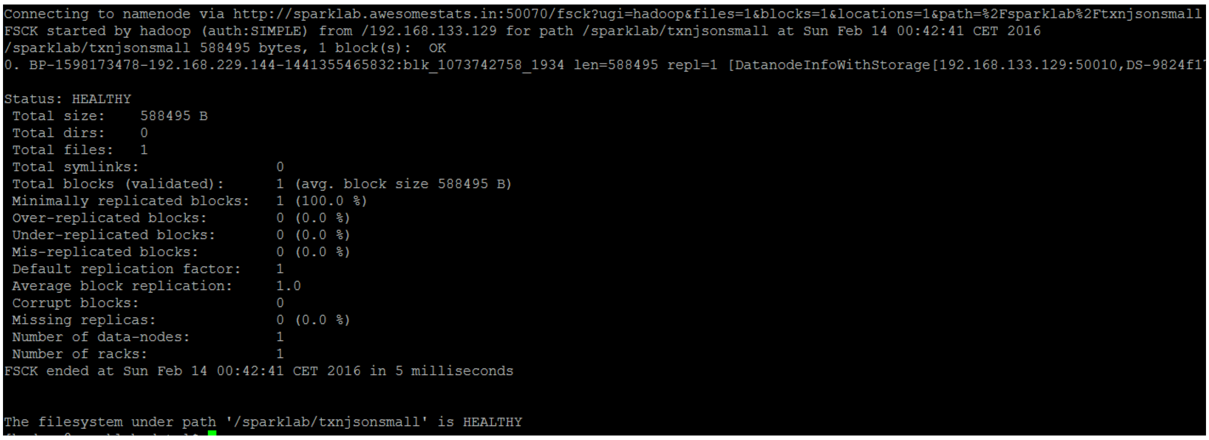
- Copying files

  Using ***WinSCP***, first transfer the file ***txnjsonsmall*** from your desktop ***/spark/data*** folder to the directory in VM ***/home/hadoop/lab/data/txnjsonsmall***

  Then copy the file 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



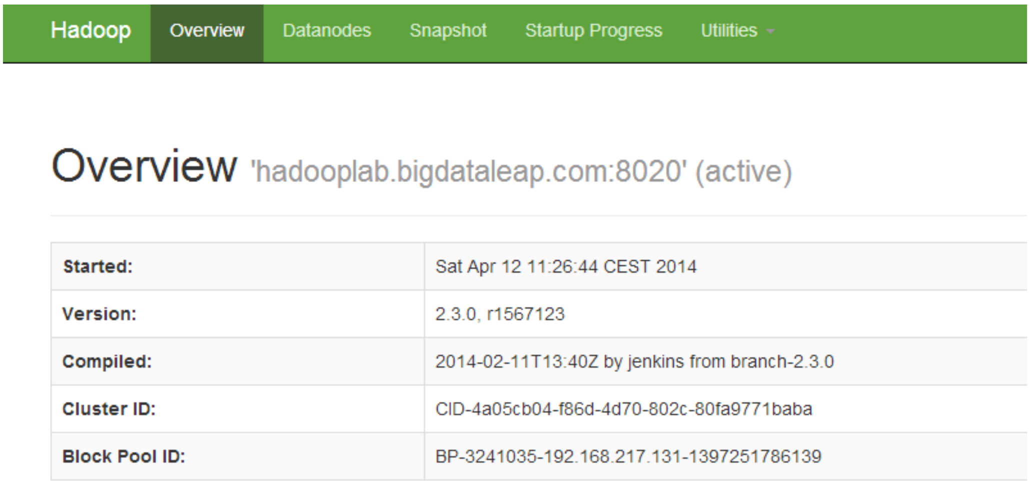
```
Connecting to namenode via http://sparklab.awesomestats.in:50070/fsck?ugi=hadoop&files=1&blocks=1&locations=1&path=%2Fsparklab%2Ftxnjsonsmall
FSCK started by hadoop (auth:SIMPLE) from /192.168.133.129 for path /sparklab/txnjsonsmall at Sun Feb 14 00:42:41 CET 2016
/sparklab/txnjsonsmall 588495 bytes, 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

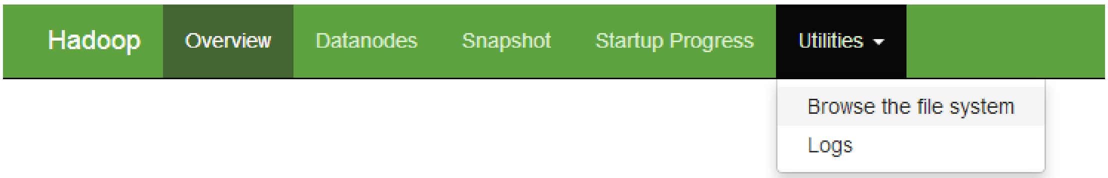
  http://hadooplab.bigdataleap.com:50070/



Overview 'hadooplab.bigdataleap.com:8020' (active)

Started:	Sat Apr 12 11:26:44 CEST 2014
Version:	2.3.0, r1567123
Compiled:	2014-02-11T13:40Z by jenkins from branch-2.3.0
Cluster ID:	CID-4a05cb04-f86d-4d70-802c-80fa9771baba
Block Pool ID:	BP-3241035-192.168.217.131-1397251786139

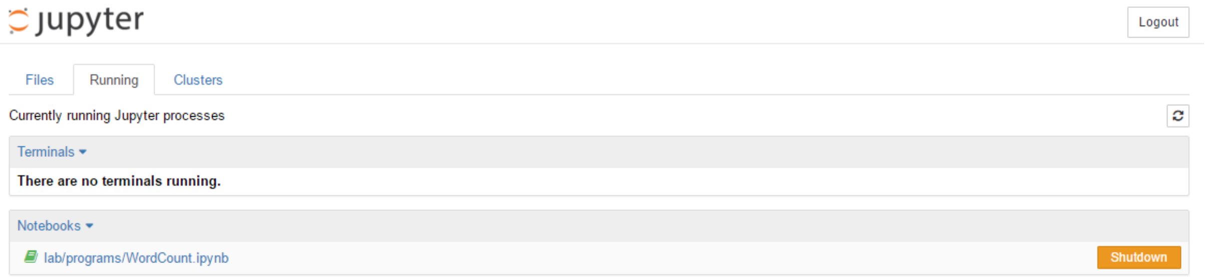
- File system explorer and log explorer is available under utilities menu



10) Configuring Spark to run on YARN mode

- **Stop IPython Notebook Process**

Shut down all the notebooks running



Currently running Jupyter processes

Terminals

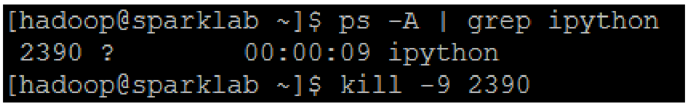
There are no terminals running.

Notebooks

lab/programs/WordCount.ipynb

Shutdown all the notebooks running.

- 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>

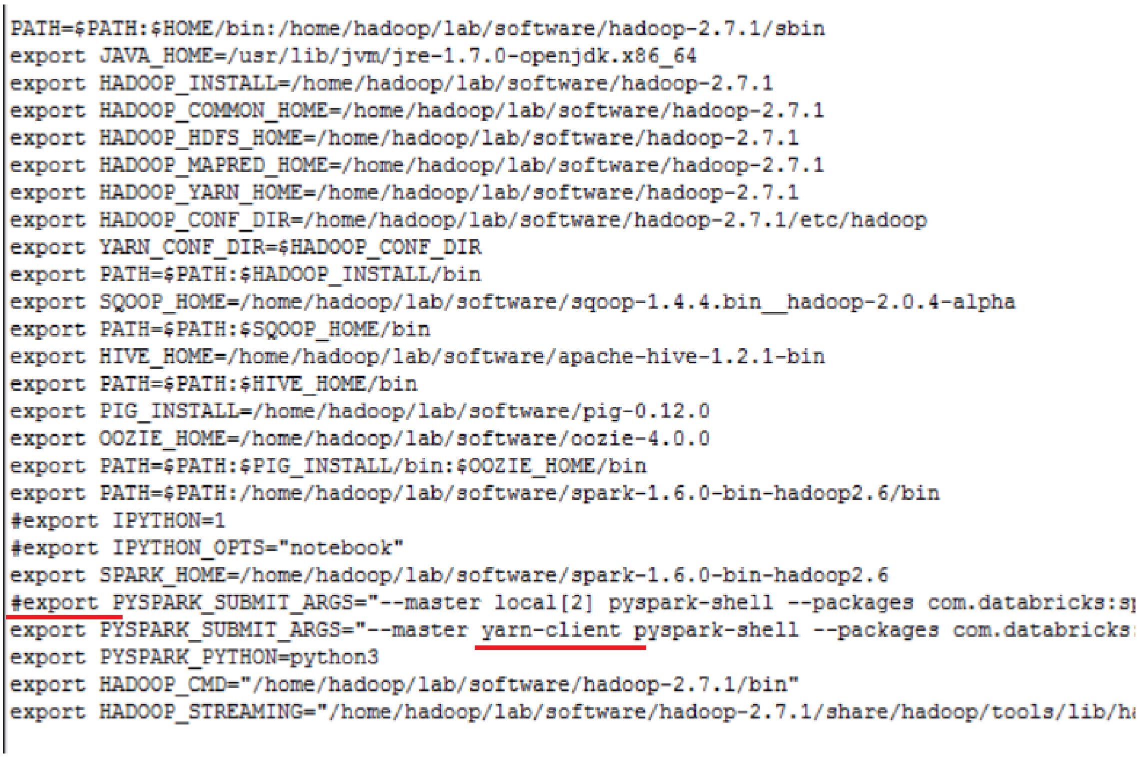
- **Configure .bash_profile**

  Open the .bash_profile file from WinSCP

  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)*



```
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:s
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/h
```

  save and close.

- **Run the bash_profile**

  Run the following command at the command prompt.

  . .bash_profile

- **Configure spark-defaults.conf**

  Go to ***/lab/reference*** directory of your desktop

Copy and replace the file ***spark-defaults.conf*** onto the following directory in VM
/home/hadoop/lab/software/spark-1.6.0-bin-hadoop2.6/conf

- Go the putty terminal

  cd /home/hadoop

  Then run .bash_profile

  . .bash_profile

  Run ***export*** command at the linux prompt. It should show the new settings for PYSPARK_SUBMIT_ARGS

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

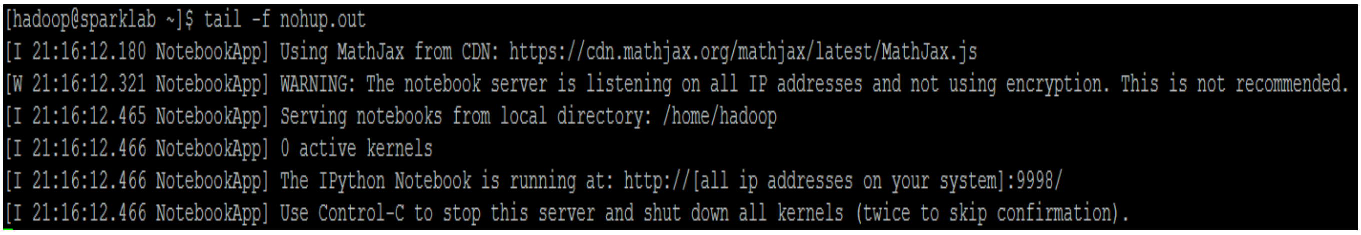
  cd /home/hadoop

  rm nohup.out

  nohup ipython notebook --profile=pyspark &

- Check the port number of the ipython notebook server port

  tail -f nohup.out



```
[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

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

The guide for this will be shared before the workshop.

12) Monitoring & Debugging

The guide for this will be shared before the workshop.

13) 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.

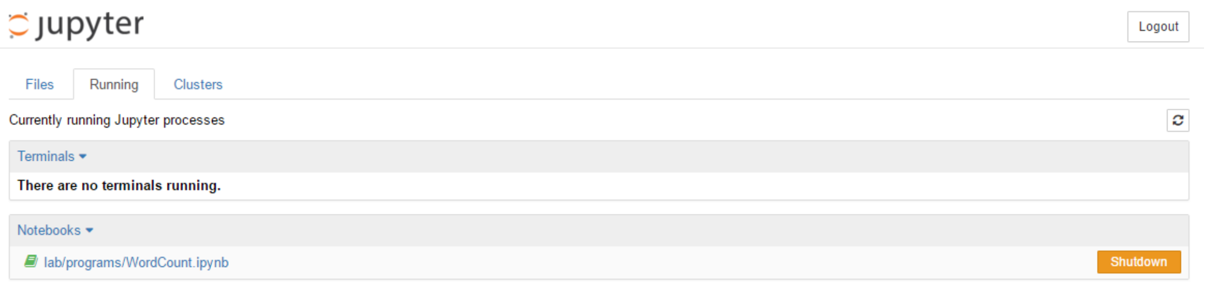
- **Run the bash_profile**

Run the following command at the command prompt.

. .bash_profile

c. Stop and restart IPython Notebook Process

Shut down all the notebooks running



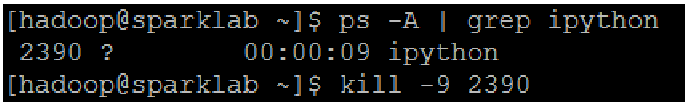
Currently running Jupyter processes

There are no terminals running.

lab/programs/WordCount.ipynb

Shutdown all the notebooks running.

- 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**

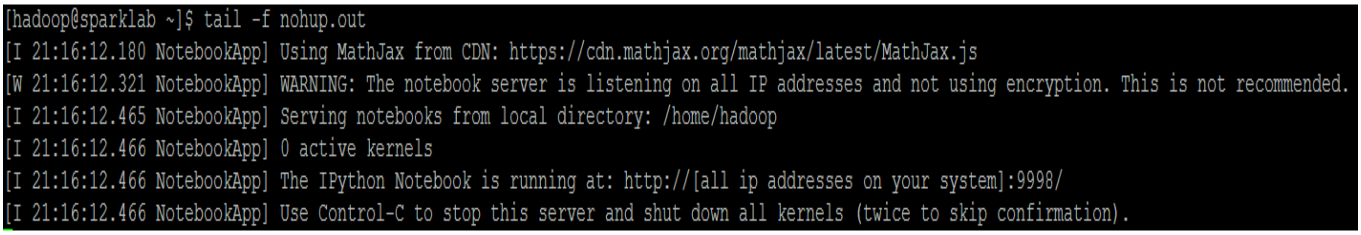
  cd /home/hadoop

  rm nohup.out

  nohup ipython notebook --profile=pyspark &

- Check the port number of the ipython notebook server port

  tail -f nohup.out



```
[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

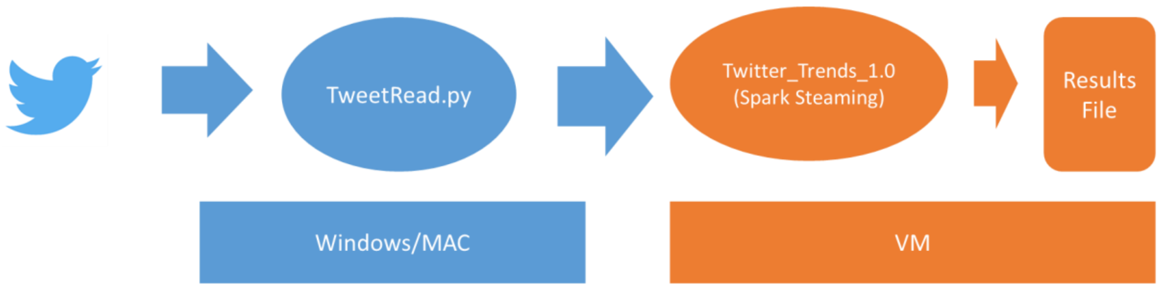
14) Participant’s Assignment 2: Working with Spark DataFrames

15) Using Spark Streaming

Preparation Step:

- 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

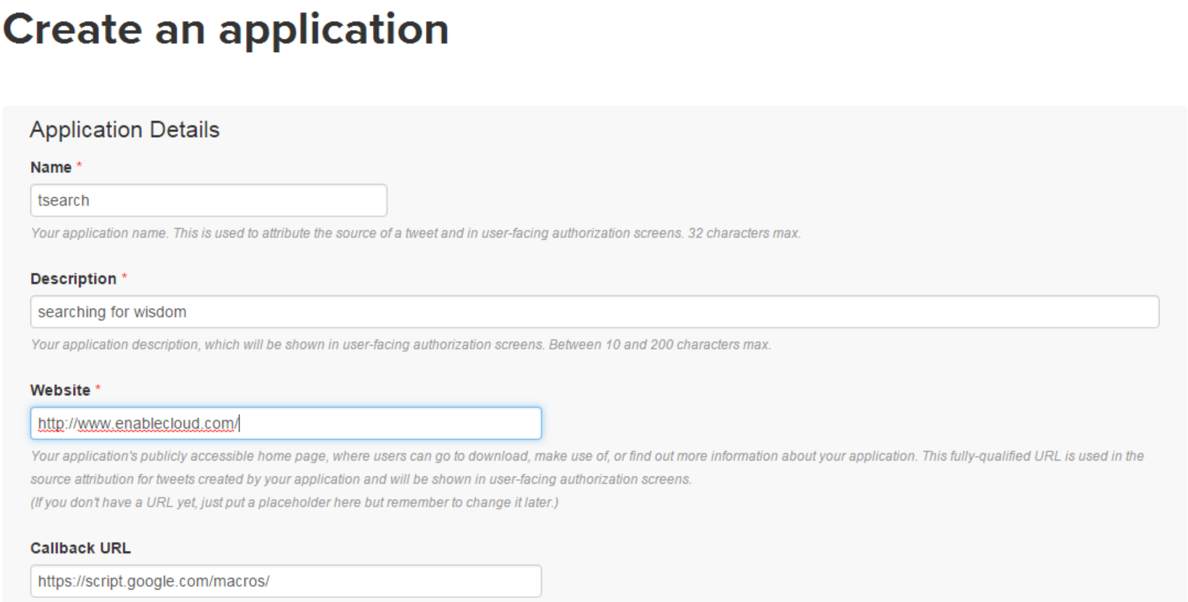
Step 2:

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.



Create an application

Application Details

Name *

tsearch

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

searching for wisdom

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

http://www.enablecloud.com/

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

https://script.google.com/macros/

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

Step 4:

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

  python TweetRead.py

Step 5:

Go to putty session of your VM and go to the following directory.

cd /home/hadoop/lab/programs/trends

Create a file called results and listen to the end of the file

touch results

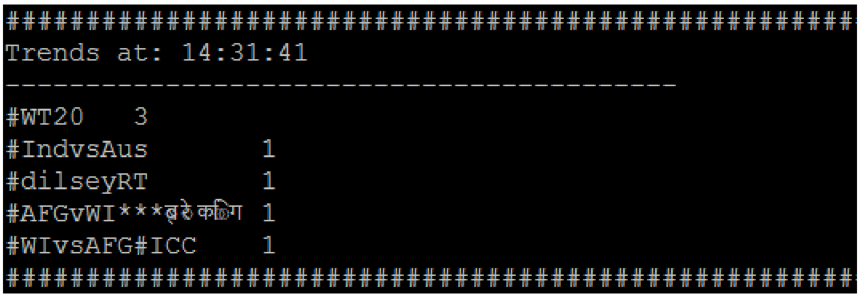
tail -f results

Step 6:

- Go to the ipython notebook
- Traverse to the directory lab/programs
- Click on the file ***Twitter_Trends_1.0.ipynb***
- 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)
- Then execute the steps one by one
- The last step will run indefinitely.

Step 7:

Once these steps are executed, go to the putty session (as step 5) and wait for tweets tags and their counts (as shown below)



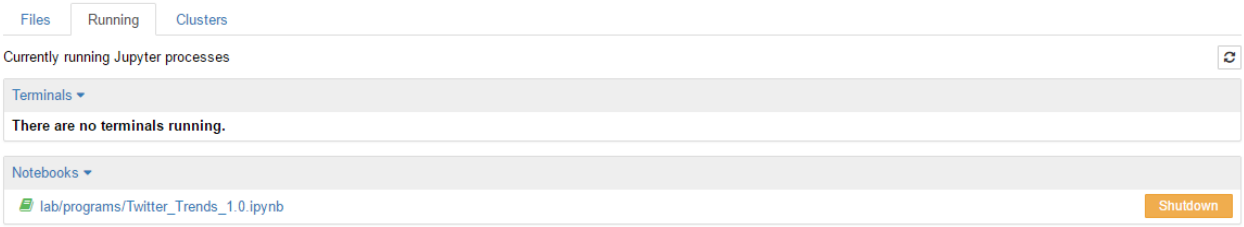
```
#####################################################
Trends at: 14:31:41
------------------------------------------
#WT20    3
#IndvsAus        1
#dilseyRT        1
#AFGvWI***ब्रेकठिग 1
#WIvsAFG#ICC     1
#####################################################
```

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

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



Currently running Jupyter processes

There are no terminals running.

lab/programs/Twitter_Trends_1.0.ipynb

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

16) Visualization, Statistics & Machine Learning Libraries

The guide for this will be shared before the workshop.

17) Assignment: Visualization, Statistics & Machine Learning Library

The guide for this will be shared before the workshop.

18) 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
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/

19) 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.)

➢ 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

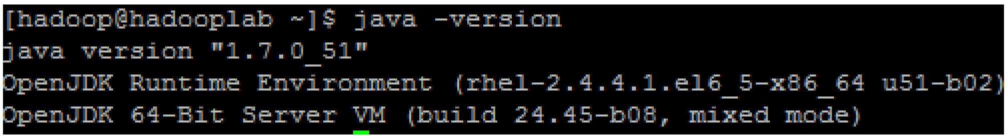
. .bash_profile

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

export

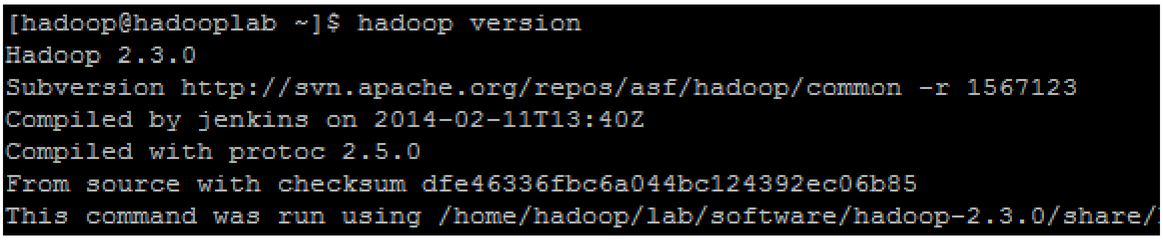
- ➢ Check the following versions

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

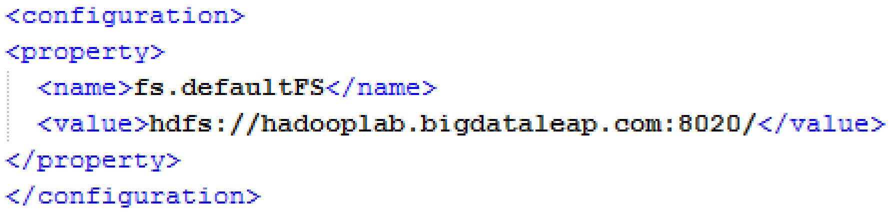
Go to conf directory of hadoop installation folder

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

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.

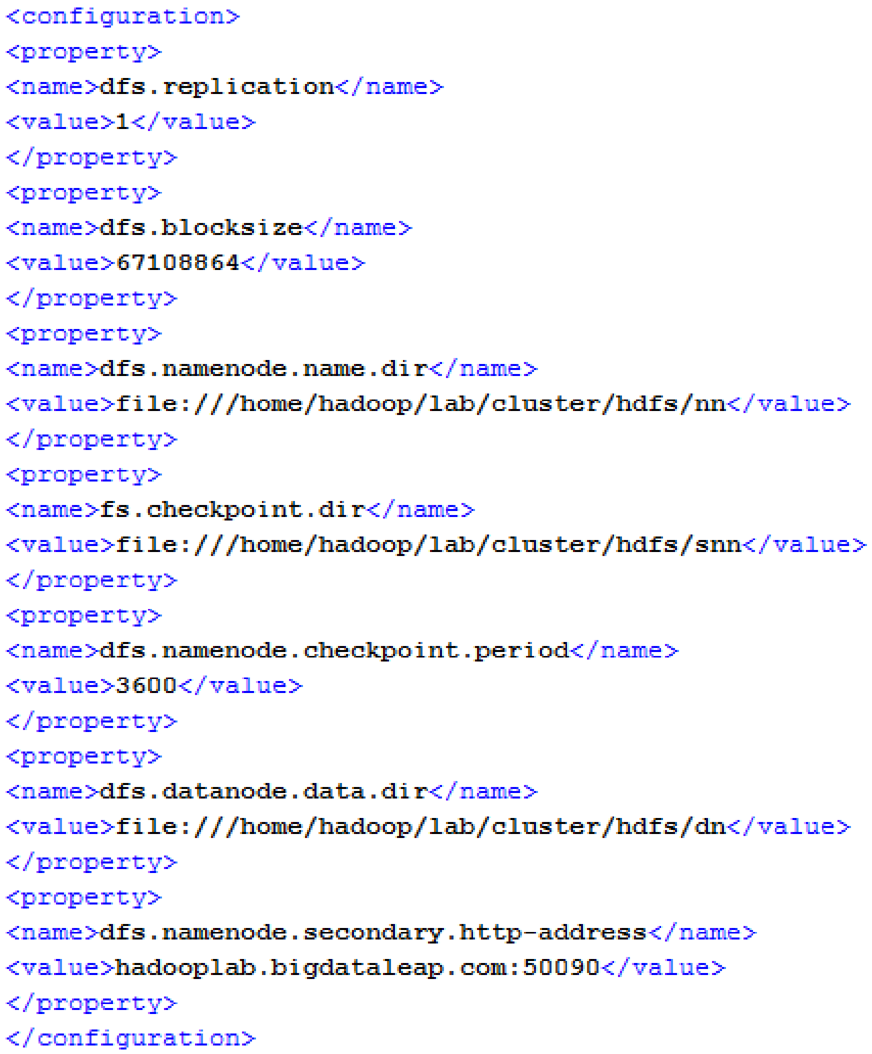
Note for MAC Users: People using MAC machine can use scp command

- core-site.xml



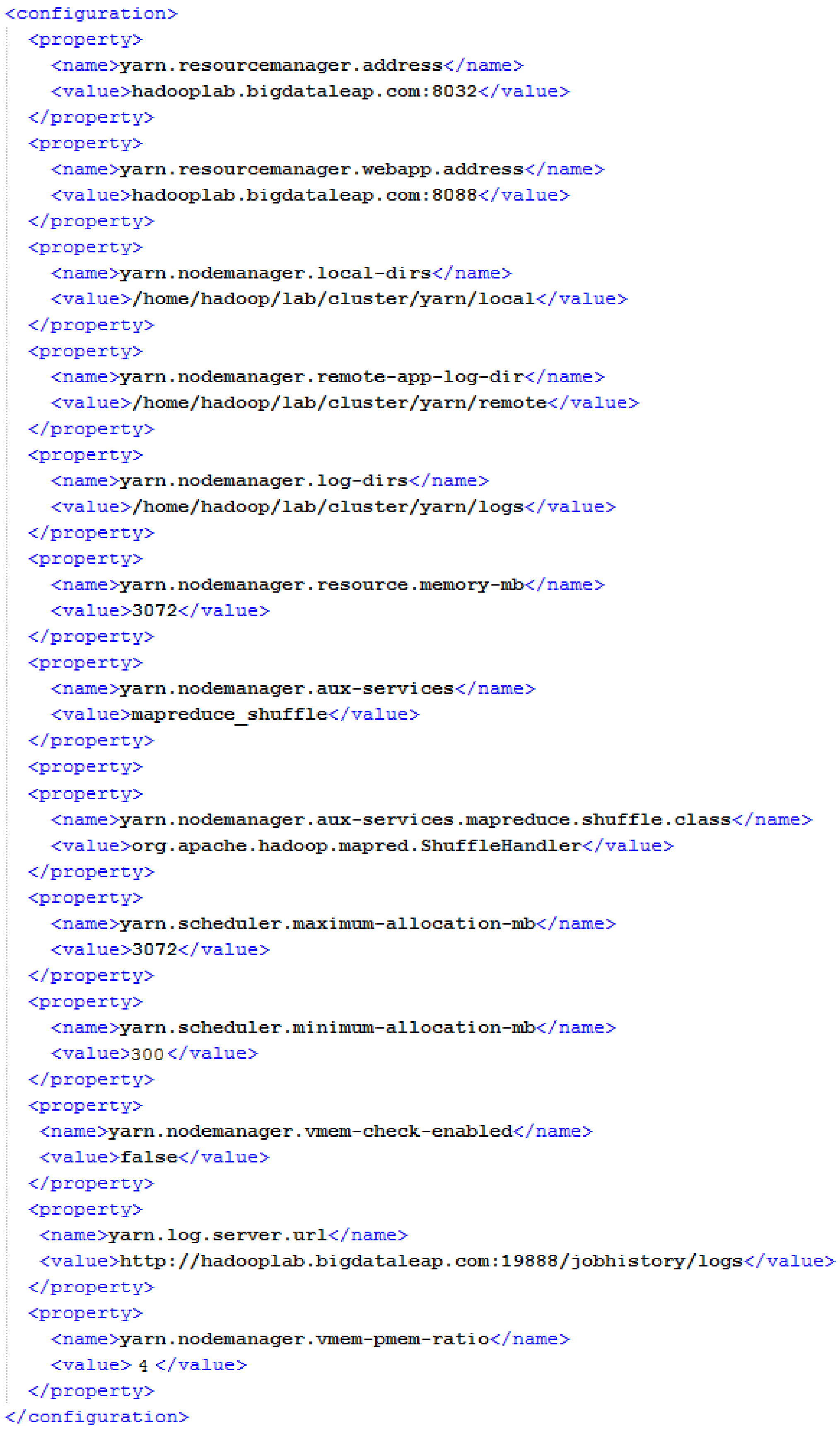
```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://hadooplab.bigdataleap.com:8020/</value>
</property>
</configuration>
```

- hdfs-site.xml



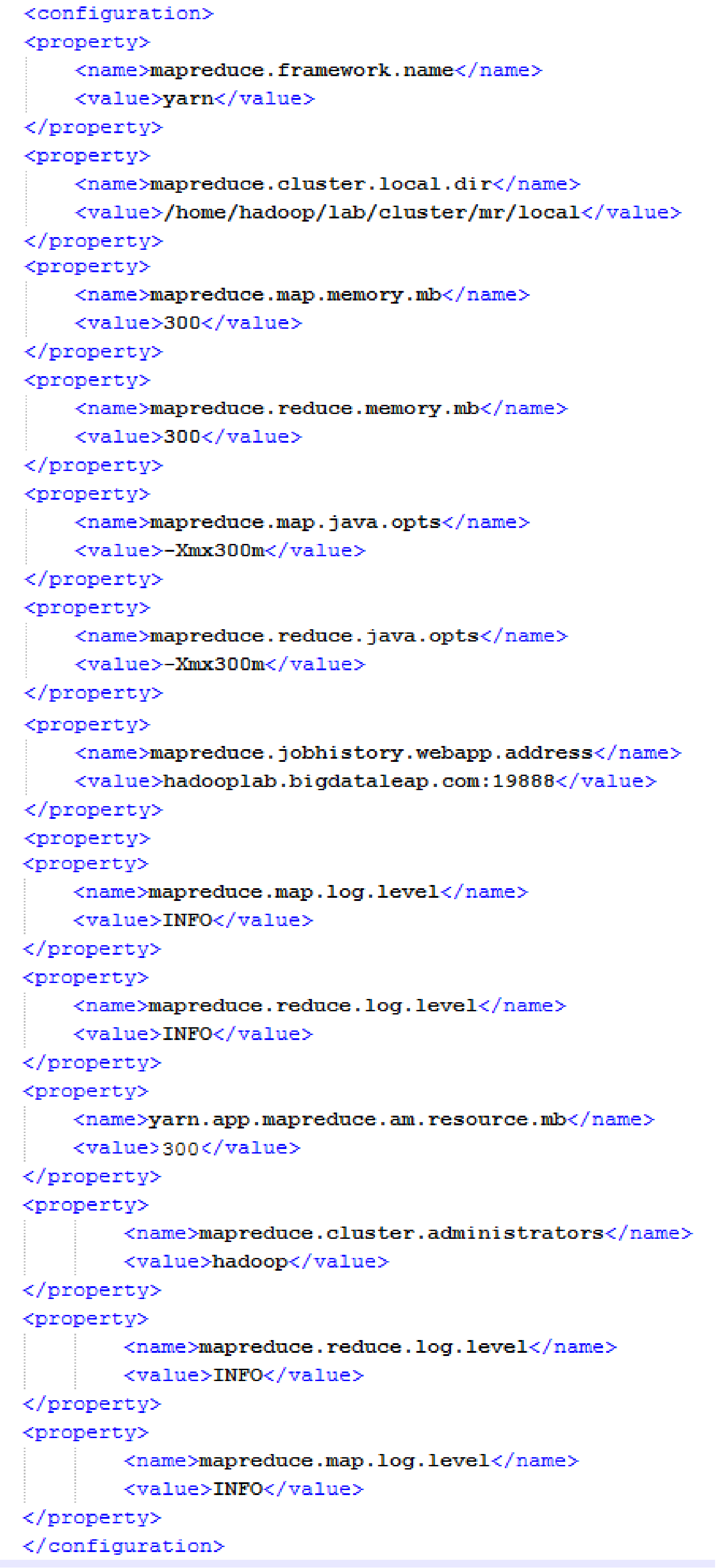
```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.blocksize</name>
<value>67108864</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///home/hadoop/lab/cluster/hdfs/nn</value>
</property>
<property>
<name>fs.checkpoint.dir</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



```
<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>
```

- 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>
```

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

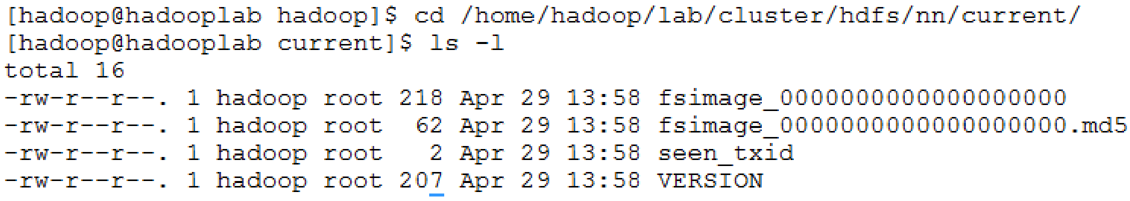
  Note: Comment the existing JAVA_HOME line if already there.

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
```