SPARK (SCALA) 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) Access Spark on the Cluster

Participant needs to Login to anyone of the following hosts using their Kerberos id.

- d159101-001.dc.gs.com
- d159101-002.dc.gs.com

After login, execute the script to get hadoop cluster config files:

/gns/software/infra/big-data/hadoop/client-latest/get_client_config.ksh 179663

kinit <Kerberos id>

source <directory chosen>/hadoop/conf/hadoop.client.profile

User should now be able to access the cluster.

Note:
GNS Path :
/gns/software/infra/big-data/spark/spark-1.5.1

2) Starting Spark

A. Start Spark Shell

Enter the command at linux prompt

spark-shell --master local[2]

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

```
16/07/11 23:29:29 INFO spark.HttpServer: Starting HTTP Server
16/07/11 23:29:29 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/07/11 23:29:29 INFO server.AbstractConnector: Started SocketConnector@0.0.0.0:37048
16/07/11 23:29:29 INFO util.Utils: Successfully started service 'HTTP class server' on port 37048.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Scala version 2.10.5 (OpenJDK 64-Bit Server VM, Java 1.7.0_55)
Type in expressions to have them evaluated.
Type :help for more information.
16/07/11 23:29:33 INFO spark.SparkContext: Running Spark version 1.6.0
16/07/11 23:29:33 INFO spark.SecurityManager: Changing view acls to: hadoop
16/07/11 23:29:33 INFO spark.SecurityManager: Changing modify acls to: hadoop
```

B. Check Spark Version

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

>>> sc.version

'1.6.0'

>>> sc.master

'local[2]'

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

Note:

File locations provided in the samples below are only for demo purpose. Data may not necessarily be available at the same location. Please change the file locations as per your lab cluster environment.

3) Download the file lab guides and data from github

Download the following project onto your desktop and untar.

https://github.com/manaranjanp/spark-using-scala

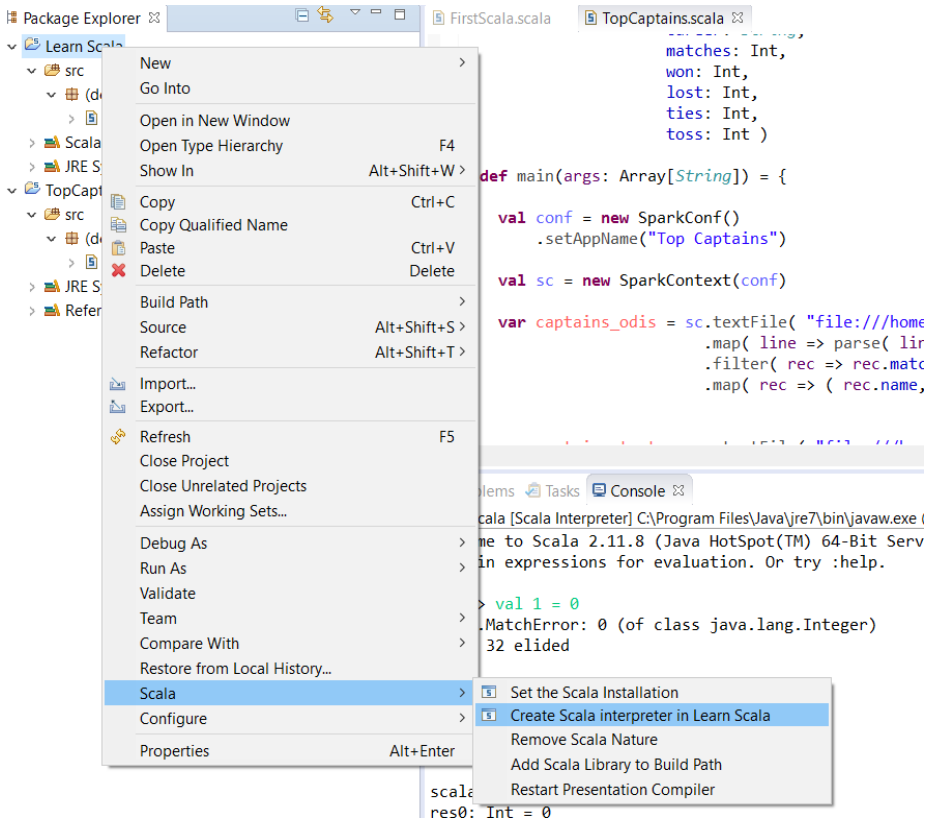
4) Scala Overview

Download the install Eclipse IDE for Scala

http://scala-ide.org/

Create a Scala project called ***Learn Scala***

Then start an Scala interpreter as shown below.



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

To learn Scala basics follow the ***Scala Introduction Ver 1.0.pdf*** tutorial.

5) Running first spark program on command line

The first program will be a word count problem.

Enter the following lines at ***spark-shell*** prompt

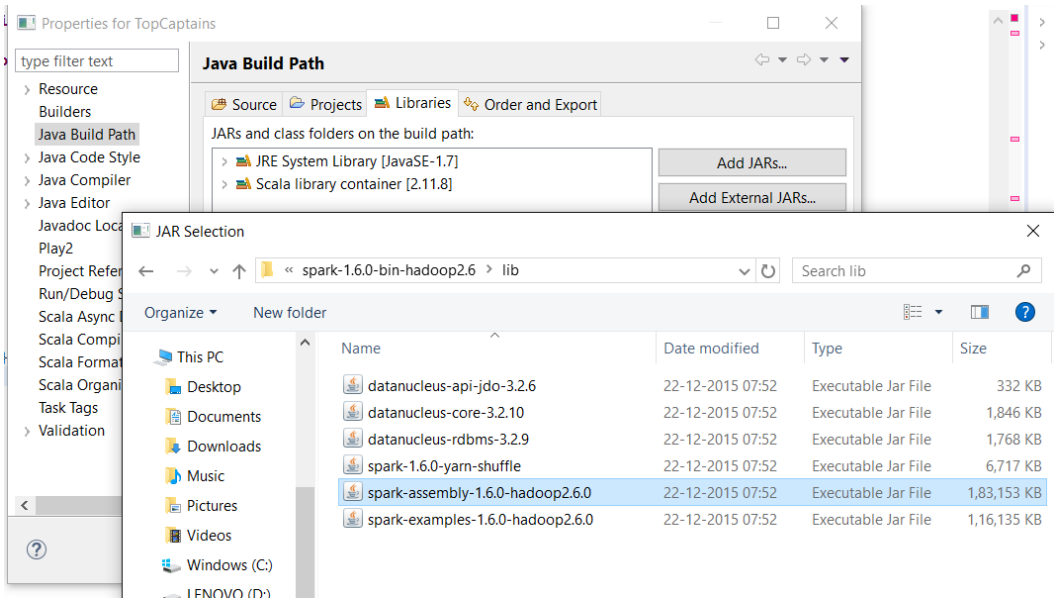
```
var wordfile = sc.textFile( "file:///home/hadoop/lab/data/words")
var words = wordfile.flatMap( line => line.split( " " ) )
words.take( 10 ).foreach( println )
var word_one = words.map( word => ( word, 1 ) )
var word_counts = word_one.reduceByKey( _+_ )
word_counts.take( 10 ).foreach( println )
```

6) Working with Spark APIs – using spark-shell (Interactive)

Follow the ***RDD APIs using Spark Scala - Top Captains.pdf*** tutorial.

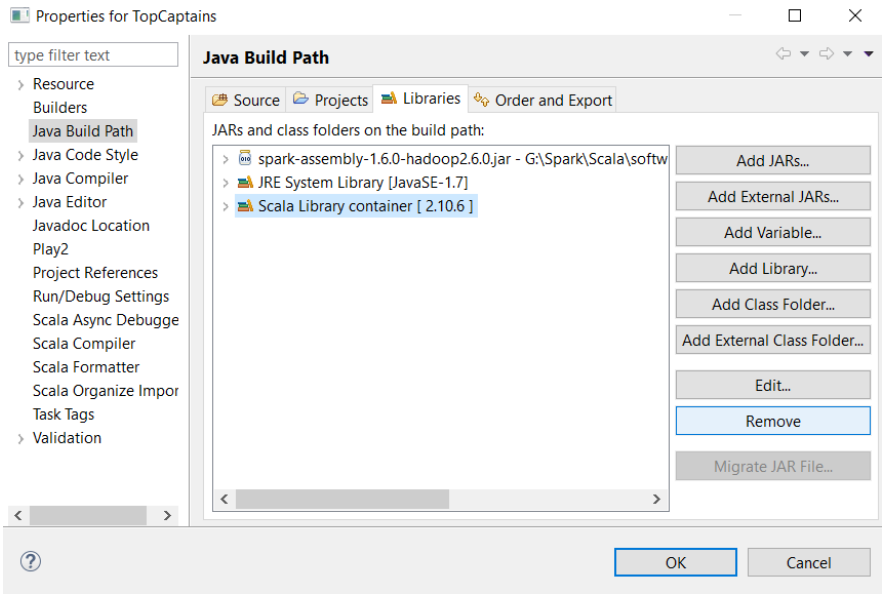
7) Writing a Scala Spark Program for running on Batch Mode

- Start Eclipse
- Create a new Scala Project and give it a new and click on **Finish**
- Right click on the project, select "**Properties**" and Select "**Java Build Path**"
- Click on External Jar and go the lib folder in Spark directory and select ***spark-assembly-1.6.0-hadoop2.6.0.jar*** file

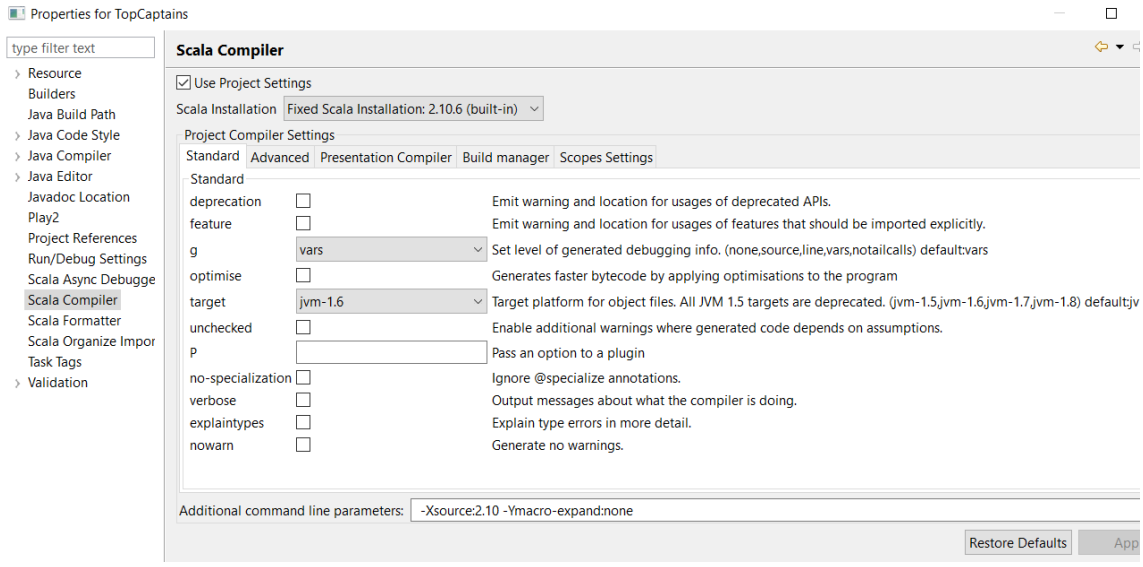


- Then select Scala Library Container in the list and click on "***Remove***". This step need to be done only if a Scala Library container is shown in the list.

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



- Select Scala Container in the left menu and select "**Use Project Settings**" and then select "***Fixed-Scala Installation 2.10.6***"



- Seect "***OK***" and close the dialog.
- Select ***src*** on the project and right click and click new scala class and name is "***TopCaptains***".
- Write the following code in the class

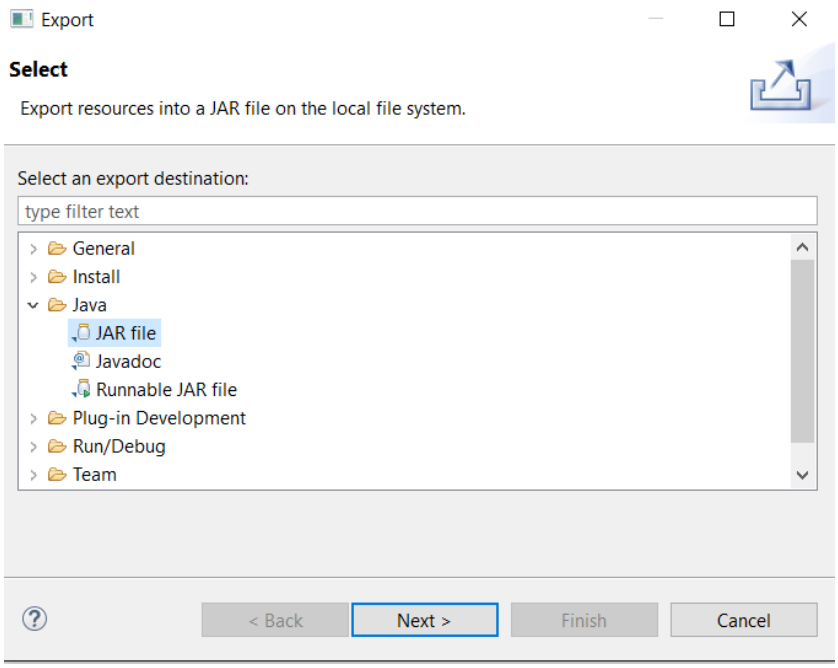
```
import org.apache.spark.{SparkConf, SparkContext}

object TopCaptains {

  def parse( line:String ) = {
```


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

```
  val pieces = line.split(",")
  val name = pieces(0)
  val country = pieces(1)
  val career = pieces(2)
  val matches = pieces(3).toInt
  val won = pieces(4).toInt
  val lost = pieces(5).toInt
  val ties = pieces(6).toInt
  val toss = pieces(7).toInt

  Captain( name, country, career, matches, won, lost, ties, toss )

 }

 case class Captain( name:String,
    country:String,
    career: String,
    matches: Int,
    won: Int,
    lost: Int,
    ties: Int,
    toss: Int )

 def main(args: Array[String]) = {

        val conf = new SparkConf()
              .setAppName("Top Captains")
              .setMaster("local[2]")
        val sc = new SparkContext(conf)

        var captains_odis = sc.textFile( "file:///home/hadoop/lab/data/captains_ODI.csv" )
        var captains = captains_odis.map( line => parse( line ) )
        var captains_100 = captains.filter( rec => rec.matches > 100 )
        var captains_100_percent_wins = captains_100.map( rec => ( rec.name,
           ( rec.won.toFloat / rec.matches.toFloat ) ) )

        var captains_tests = sc.textFile( "file:///home/hadoop/lab/data/captains_Test.csv" )
        var captains_tests_recs = captains_tests.map( line => parse( line ) )
        var captains_tests_50 = captains_tests_recs.filter( rec => rec.matches > 50 )
        var captain_top = captains_tests_50.map( rec => ( rec.name,
           ( rec.won.toFloat / rec.matches.toFloat) ) ).sortBy( rec => rec._2, ascending = false )

        var all_time_best_captains = captains_100_percent_wins.join( captain_top )
        var best_captains = all_time_best_captains.map( rec => ( rec._1,
                              ( rec._2._1 * 0.4 + rec._2._2 * .6 ) ) )

        best_captains .saveAsTextFile("file:///home/hadoop/lab/programs/results/captains")
 }

}
```


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

- Verify if the project compiled. Or go to "***Project***" menu of Eclipse, select "***Clean***" and click on "***OK***".
- Select the scala project and right click, select "***export***" and create a jar file and save it as shown below. Name the jar ***captains.jar***



- Transfer the jar file to VM using WinSCP under the directory "***/home/hadoop/lab/programs***"
- Go to programs directory

  cd /home/hadoop/lab/programs

- Submit the program for execution

  spark-submit --class TopCaptains --master local[2] captains.jar

- Go to /home/hadoop/lab/results directory. The program should have created a directory called ***captains***.

  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

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

[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

Follow ***DataFrames using Spark Scala – MovieLens.pdf*** tutorial

9) 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.

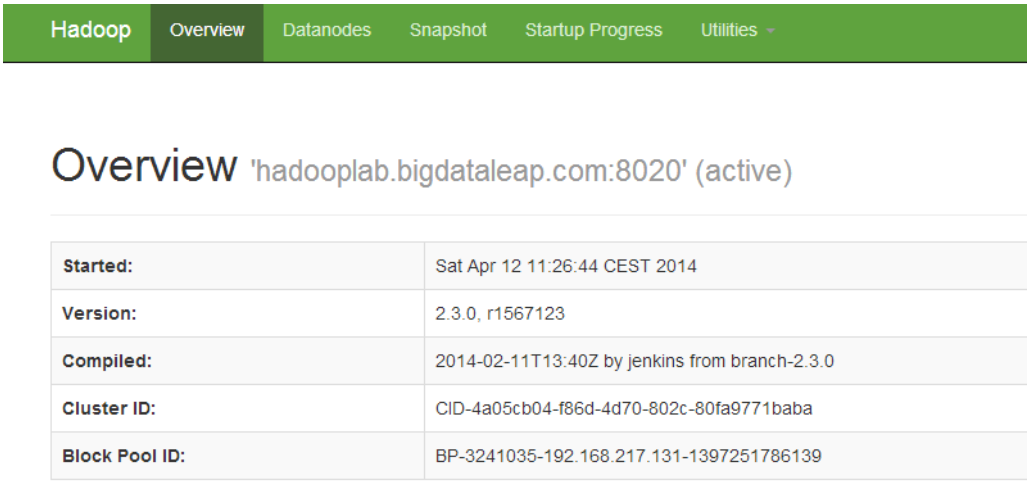
- HDFS Web UI

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

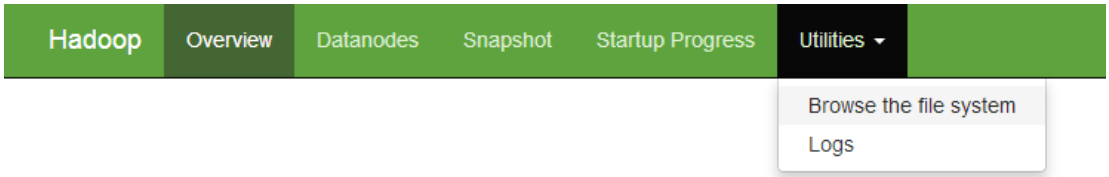
Open your browser & enter the following url

http://<namenode machine dns name>: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



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

Follow ***Working with HDFS and YARN - Retail Analysis.pdf*** tutorial.

11) Integrate with Hive

- Start beeline interface

  beeline -u 'jdbc:hive2://d179663-001.dc.gs.com,d179663-002.dc.gs.com,d179663-003.dc.gs.com:2181/default;principal=dchive/_HOST@GS.COM;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2

- Create hive database and table

  Create a database with your name (use your short name).

  create database <your name>;

  use <your name>;

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

```
CREATE TABLE flights2008 (Year STRING,
Month STRING,
DayofMonth INT,
DayOfWeek INT,
DepTime INT,
CRSDepTime INT,
ArrTime INT,
CRSArrTime INT,
UniqueCarrier STRING,
FlightNum STRING,
TailNum STRING,
ActualElapsedTime FLOAT,
CRSElapsedTime FLOAT,
AirTime FLOAT,
ArrDelay FLOAT,
DepDelay FLOAT,
Origin STRING,
Dest STRING,
Distance FLOAT,
TaxiIn INT,
TaxiOut INT,
Cancelled STRING,
CancellationCode STRING,
Diverted INT,
CarrierDelay FLOAT,
WeatherDelay FLOAT,
NASDelay FLOAT,
SecurityDelay FLOAT,
LateAircraftDelay FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

- Transfer the dataset from Desktop to VM under /home/hadoop/lab/data/ directory.
- Load data

```
LOAD DATA LOCAL INPATH '/home/hadoop/lab/data/2008.csv.bz2' INTO TABLE
flights2008;
```

12) Working with Hive

Follow the steps in ***Working with Hive - Flight Delay Analysis.pdf*** tutorial.

13) Monitoring & Debugging

The guide for this will be shared before the workshop.

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

14) Working with Unstructured data

Follow ***Working with logs & SQL Functions - NASA_logs.pdf*** tutorial

15) Using Spark Streaming

Follow ***Spark Streaming - Twitter Trends.pdf*** tutorial

16) Assignment: Visualization, Statistics & Machine Learning Library

The guide for this will be shared before the workshop.

17) 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-
```


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

```
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/

18) Appendix: Configuring Hadoop

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

a. 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*

b. 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
```


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

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/
```

c. 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

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

```
<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

d. 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/

e. 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.

f. 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