Lab - 2

Objectives:

1. Introduction to Virtualized Storage. How does HDFS work?
2. Distributed resource management using Apache Ambari.
3. Loading data into HDFS.
4. Visualizing loaded data in Zeppelin Notebook, using Spark SQL.

Instructions:

Downloading required files:

1. Download the files from the following URLs
    a. https://github.com/synchon/IITM-CS4830/raw/master/lab2/omniture-logs.tsv
    b. https://raw.githubusercontent.com/synchon/IITM-CS4830/master/lab2/season-1213_csv.csv
    c. https://raw.githubusercontent.com/synchon/IITM-CS4830/master/lab2/trucks.csv
2. SSH into your VM.
3. Once done, you need to add **127.0.0.1** as an entry to the **/etc/hosts** file, like so:

```
echo '127.0.0.1 sandbox-hdp.hortonworks.com' | sudo tee -a /etc/hosts
```

Loading data into HDFS:

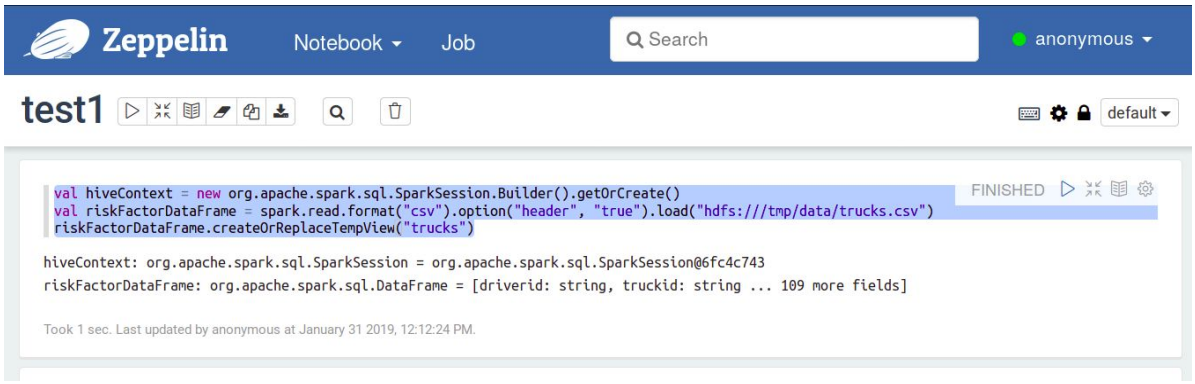
1. In the **Ambari Services** (in the left sidebar), click on **HDFS**.
2. Access the Hadoop Dashboard on your browser at: **<public-ip>:50070** (change the IP address to the public IP of the VM).
3. On the navbar, click on **Utilities** and then select **Browse the file system**.
4. Play around with the interface and get yourself familiar with files residing in the HDFS.

Visualizing data using Zeppelin Notebook and Spark SQL:

1. Note down the metrics like number of blocks, under-replicated blocks/ namenode Heap etc…
2. From the **Ambari Services**, click on **blue icon** (shown below) and open File-view

3. Upload each of the files using the interface to the "/tmp/data" directory (create a folder called /data and upload each of the files)
4. Note down the change in metrics like block size etc.. (explain these changes in the report)
5. To open Zeppelin Notebook, open your browser and access : **<public-ip>:9995** (change the IP address to the public IP of the VM).
6. Click **Create new note** and name the note as **Lab 2**. Leave the **Default Interpreter** as **spark2**.

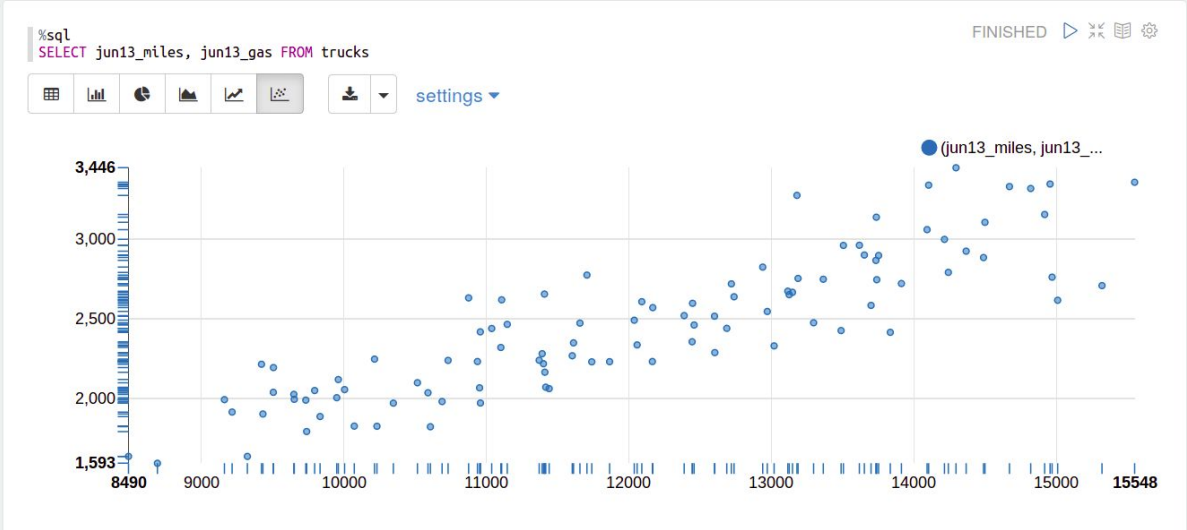
Below is an example of scala code that reads data from hdfs and displays the same




```
val ans = hiveContext.sql("SELECT driverid, jun13_miles FROM trucks LIMIT 15");
ans.show();
```

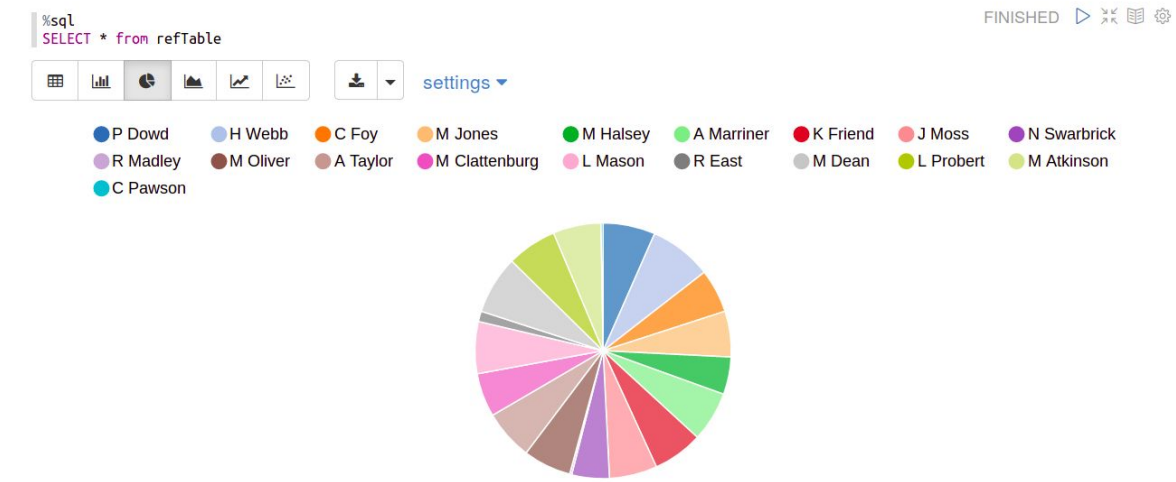
```
+--------+-----------+
|driverid|jun13_miles|
+--------+-----------+
|      A1|       9217|
|      A2|      12058|
|      A3|      13652|
|      A4|      12687|
|      A5|      10233|
|      A6|      14488|
|      A7|      10938|
|      A8|      11392|
|      A9|      12601|
|     A10|      13699|
|     A11|      12447|
```

The below snapshot reads data in Pyspark and visualizes the same consequently



The above code is also available at:
https://github.com/synchon/IITM-CS4830/blob/master/lab2/codeSample.md

7) Now read the data present in the **season-1213_csv.csv** from the HDFS. Understand the data and attempt to generate the following plot (For those unfamiliar with SQL have look at the groupby command). This plot describes the number of matches officiated by each referee



In the report, explain how the computations used to generate the above table is executed in the map-reduce framework (what are the map/reduce operations)

Questions to answer:

1. What is the relation of HDFS to CAP theorem
2. We saw Hive,Pig are one of the many services that deals with Yarn. What is the funciton of these two service
3. Note down metrics like (number of blocks, heap size, under-replicated blocks). Explain the change in metrics after uploading each of the files.
4. Describe the task (of referee vs matches officiated) as a map-reduce task
5. Attach code snippets/plot screenshots of the completed Zeppelin task