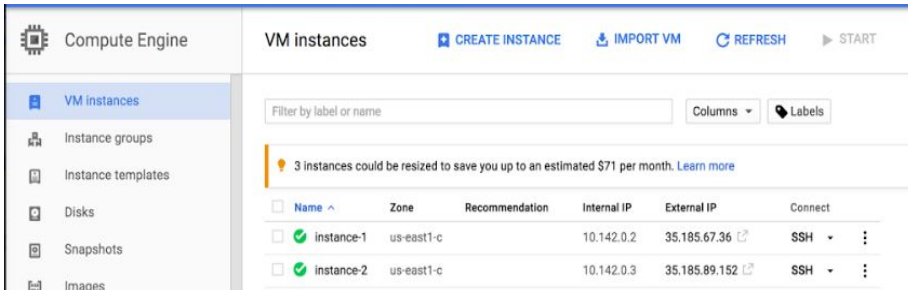
# Team 6 - TWITTER DATA ANALYTICS IMPLEMENTATION ON GOOGLE CLOUD

## Screenshots of each step

1. Creation of Google Instances :

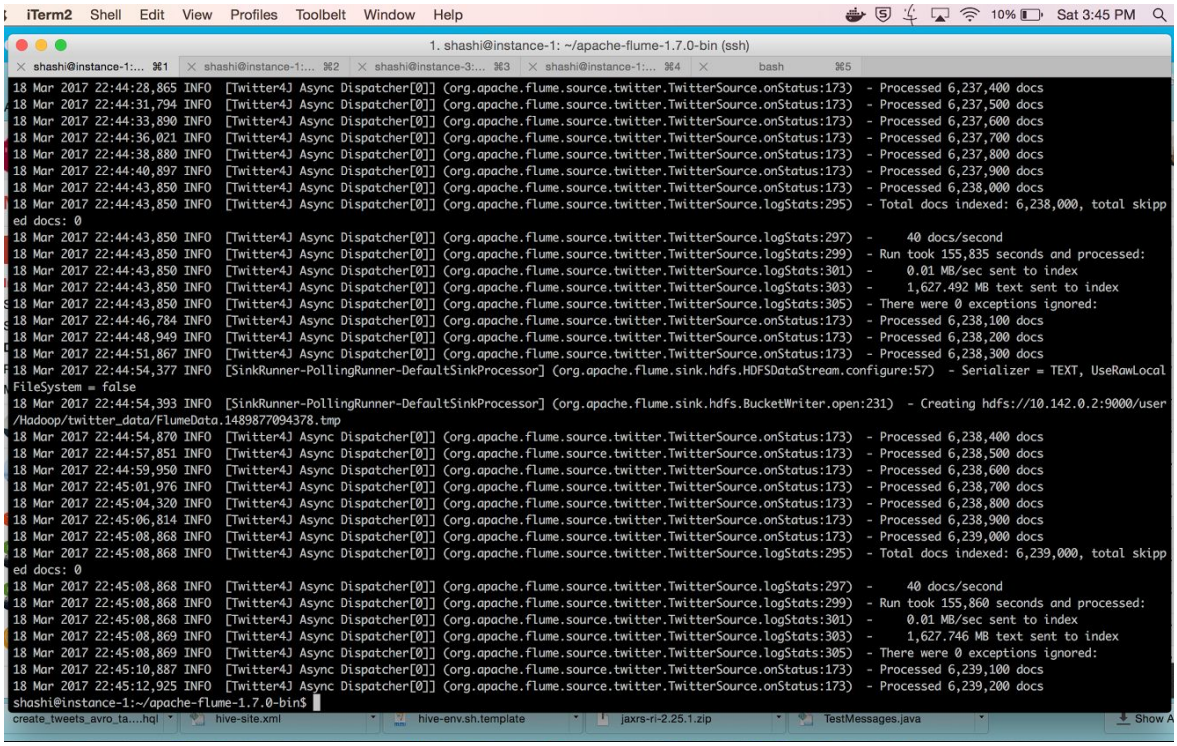



2. Apache Flume Configuration file

```
#### TwitterAgent for collecting Twitter data to Hadoop HDFS #####

TwitterAgent.sources = Twitter
TwitterAgent.channels = FileChannel
TwitterAgent.sinks = HDFS

#TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource
TwitterAgent.sources.Twitter.channels = FileChannel

TwitterAgent.sources.Twitter.consumerKey = SqWu0uWAtrg6p6ROqAptbJlva
TwitterAgent.sources.Twitter.consumerSecret = bPtncFTPBUwvbzLnW5yv0VmtMMPkcnst4r8HrAkZwxG5uuB1xo
TwitterAgent.sources.Twitter.accessToken = 2235585415-RLCVufXlDhzcXsAe4DknXnmoQV7ihlXpvB5Tevk
TwitterAgent.sources.Twitter.accessTokenSecret = XFbV51FS0BtkBXe7jOA2qe8Sd8kirNTkur9hwb5JZdqCy

TwitterAgent.sources.Twitter.maxBatchSize = 50000
TwitterAgent.sources.Twitter.maxBatchDurationMillis = 100000

#TwitterAgent.sources.Twitter.keywords = Apache, Hadoop, Mapreduce, hadooptutorial, Hive, Hbase, MySql

TwitterAgent.sinks.HDFS.channel = FileChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path = hdfs://10.142.0.2:9000/user/Hadoop/twitter_data
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
TwitterAgent.sinks.HDFS.hdfs.batchSize = 200000
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
TwitterAgent.sinks.HDFS.hdfs.rollCount = 2000000

TwitterAgent.channels.FileChannel.type = file
TwitterAgent.channels.FileChannel.checkpointDir =  /home/shashi/apache-flume-1.7.0-bin/var/log/flume/checkpoint/
TwitterAgent.channels.FileChannel.dataDirs = /home/shashi/apache-flume-1.7.0-bin/var/log/flume/data/
```

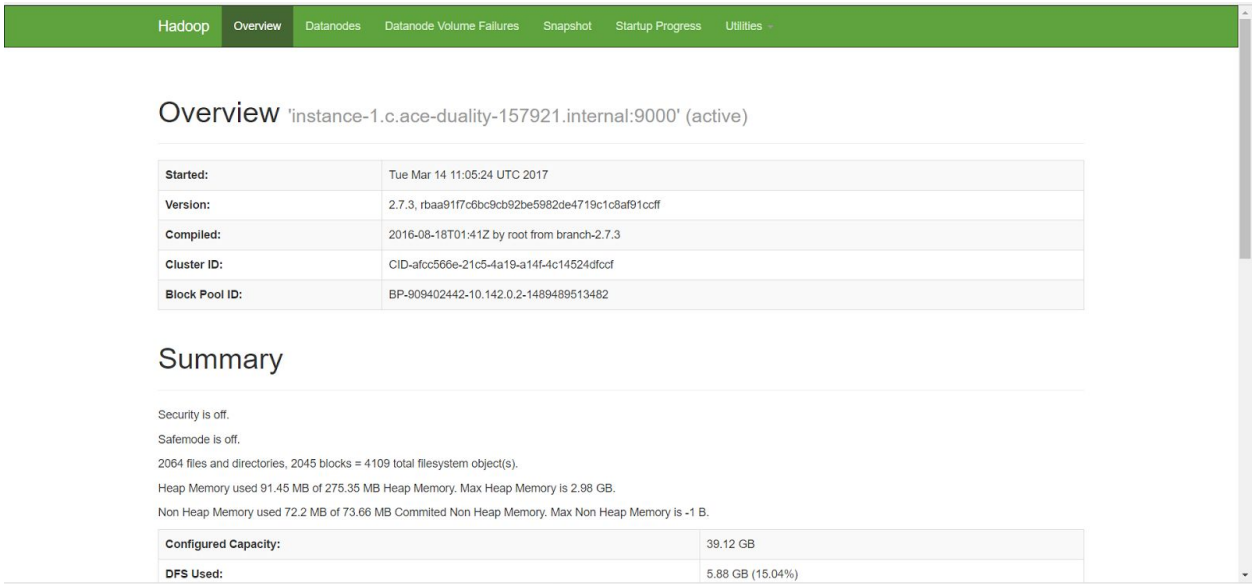
3. Execution of command to load data from Twitter source and loading into HDFS

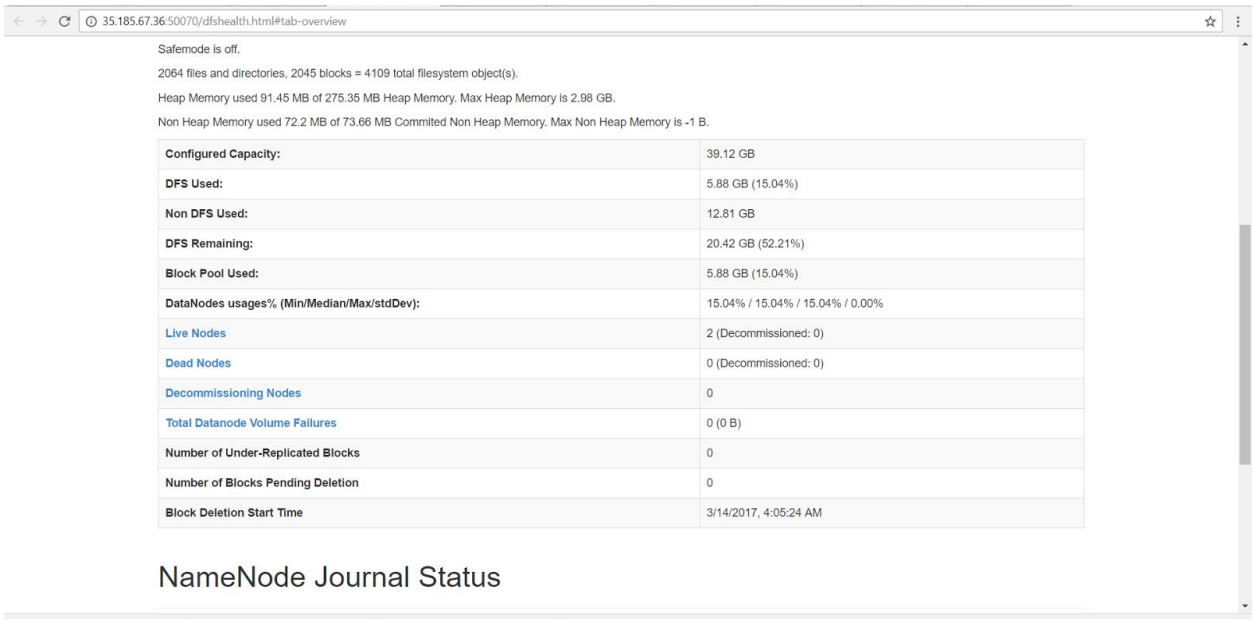
bin/flume-ng agent --conf ./conf/ -f  conf/twit.conf  Dflume.root.logger=DEBUG,console -n TwitterAgent  >> log1 2>&1  &




# Namenode Server Check(35.185.67.36:50070)

http://35.185.67.36:50070/dfshealth.html#tab-overview






Hadoop daemon process  by command line
jps (command)

```
shashi@instance-1:~$ jps
9648 SecondaryNameNode
9812 ResourceManager
9909 NodeManager
6406 Bootstrap
17879 Jps
9352 NameNode
9454 DataNode
shashi@instance-1:~$ clear
shashi@instance-1:~$ jps
9648 SecondaryNameNode
9812 ResourceManager
9909 NodeManager
6406 Bootstrap
17895 Jps
9352 NameNode
9454 DataNode
shashi@instance-1:~$ hdfs dfsadmin -report
17/03/25 22:21:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Configured Capacity: 42003021824 (39.12 GB)
Present Capacity: 25071865856 (23.35 GB)
DFS Remaining: 18753622016 (17.47 GB)
DFS Used: 6318243840 (5.88 GB)
DFS Used%: 25.20%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (2):

Name: 10.142.0.3:50010 (instance-2.c.ace-duality-157921.internal)
Hostname: instance-2.c.ace-duality-157921.internal
Decommission Status : Normal
Configured Capacity: 21001510912 (19.56 GB)
DFS Used: 3159126016 (2.94 GB)
Non DFS Used: 4421447680 (4.12 GB)
DFS Remaining: 13420937216 (12.50 GB)
DFS Used%: 15.04%
DFS Remaining%: 63.90%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
```

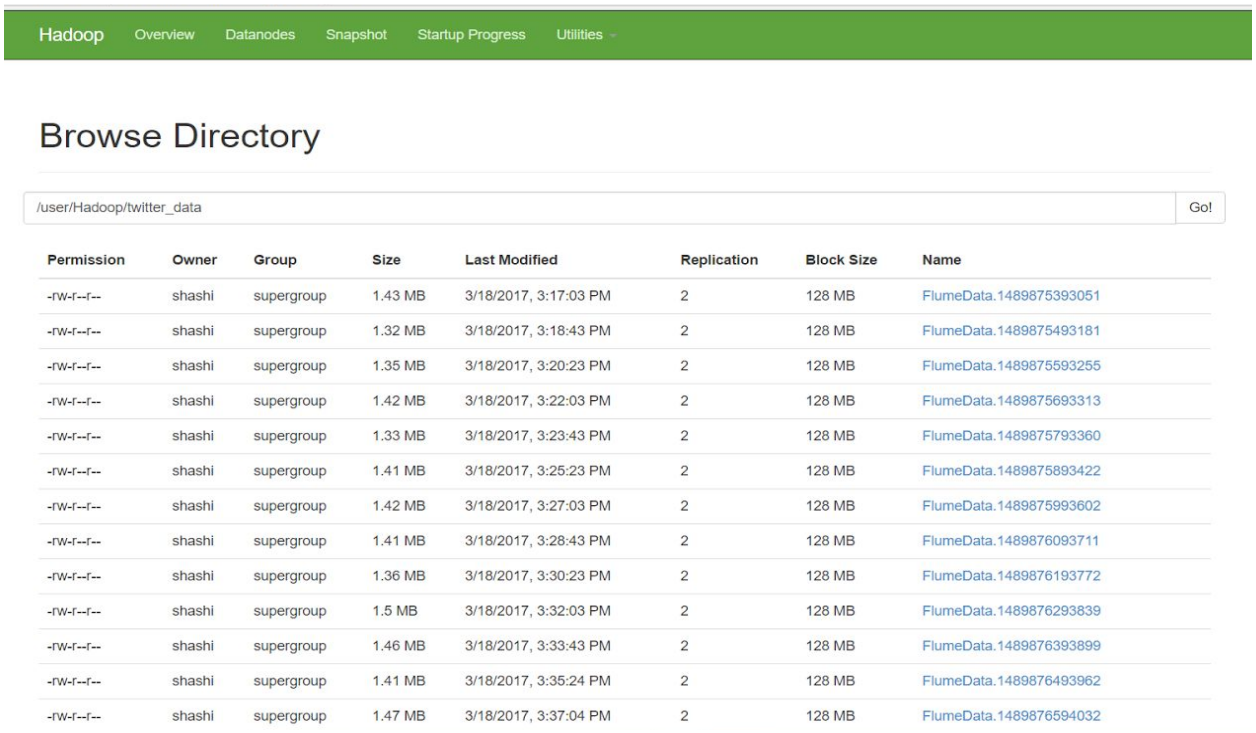
hdfs dfsadmin -report
Report of Hadoop cluster

```
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (2):

Name: 10.142.0.3:50010 (instance-2.c.ace-duality-157921.internal)
Hostname: instance-2.c.ace-duality-157921.internal
Decommission Status : Normal
Configured Capacity: 21001510912 (19.56 GB)
DFS Used: 3159126016 (2.94 GB)
Non DFS Used: 4421447680 (4.12 GB)
DFS Remaining: 13420937216 (12.50 GB)
DFS Used%: 15.04%
DFS Remaining%: 63.90%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Mar 25 22:21:10 UTC 2017


Name: 10.142.0.2:50010 (instance-1.c.ace-duality-157921.internal)
Hostname: instance-1.c.ace-duality-157921.internal
Decommission Status : Normal
Configured Capacity: 21001510912 (19.56 GB)
DFS Used: 3159117824 (2.94 GB)
Non DFS Used: 12509708288 (11.65 GB)
DFS Remaining: 5332684800 (4.97 GB)
DFS Used%: 15.04%
DFS Remaining%: 25.39%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Sat Mar 25 22:21:11 UTC 2017
```

## Checking files loaded in HDFS

http://35.185.67.36:50070/explorer.html#/user/Hadoop/twitter_data

Click the link for seeing the data




| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| -rw-r--r-- | shashi | supergroup | 1.43 MB | 3/18/2017, 3:17:03 PM | 2 | 128 MB | FlumeData.1489875393051 |
| -rw-r--r-- | shashi | supergroup | 1.32 MB | 3/18/2017, 3:18:43 PM | 2 | 128 MB | FlumeData.1489875493181 |
| -rw-r--r-- | shashi | supergroup | 1.35 MB | 3/18/2017, 3:20:23 PM | 2 | 128 MB | FlumeData.1489875593255 |
| -rw-r--r-- | shashi | supergroup | 1.42 MB | 3/18/2017, 3:22:03 PM | 2 | 128 MB | FlumeData.1489875693313 |
| -rw-r--r-- | shashi | supergroup | 1.33 MB | 3/18/2017, 3:23:43 PM | 2 | 128 MB | FlumeData.1489875793360 |
| -rw-r--r-- | shashi | supergroup | 1.41 MB | 3/18/2017, 3:25:23 PM | 2 | 128 MB | FlumeData.1489875893422 |
| -rw-r--r-- | shashi | supergroup | 1.42 MB | 3/18/2017, 3:27:03 PM | 2 | 128 MB | FlumeData.1489875993602 |
| -rw-r--r-- | shashi | supergroup | 1.41 MB | 3/18/2017, 3:28:43 PM | 2 | 128 MB | FlumeData.1489876093711 |
| -rw-r--r-- | shashi | supergroup | 1.36 MB | 3/18/2017, 3:30:23 PM | 2 | 128 MB | FlumeData.1489876193772 |
| -rw-r--r-- | shashi | supergroup | 1.5 MB | 3/18/2017, 3:32:03 PM | 2 | 128 MB | FlumeData.1489876293839 |
| -rw-r--r-- | shashi | supergroup | 1.46 MB | 3/18/2017, 3:33:43 PM | 2 | 128 MB | FlumeData.1489876393899 |
| -rw-r--r-- | shashi | supergroup | 1.41 MB | 3/18/2017, 3:35:24 PM | 2 | 128 MB | FlumeData.1489876493962 |
| -rw-r--r-- | shashi | supergroup | 1.47 MB | 3/18/2017, 3:37:04 PM | 2 | 128 MB | FlumeData.1489876594032 |

## 4. Loading data from HDFS to Hive tables by using Avro and schema definition

Creating tweets table

```
CREATE TABLE tweets
  ROW FORMAT SERDE
     'org.apache.hadoop.hive.serde2.avro.AvroSerDe'
  STORED AS INPUTFORMAT
     'org.apache.hadoop.hive.ql.io.avro.AvroContainerInputFormat'
  OUTPUTFORMAT
     'org.apache.hadoop.hive.ql.io.avro.AvroContainerOutputFormat'
  TBLPROPERTIES ('avro.schema.url'='file://
  /home/shashi/TwitterDataAvroSchema.avsc') ;

LOAD DATA INPATH '/user/Hadoop/twitter_data/FlumeData.*' OVERWRITE INTO
TABLE tweets;
```

Schema is defined below

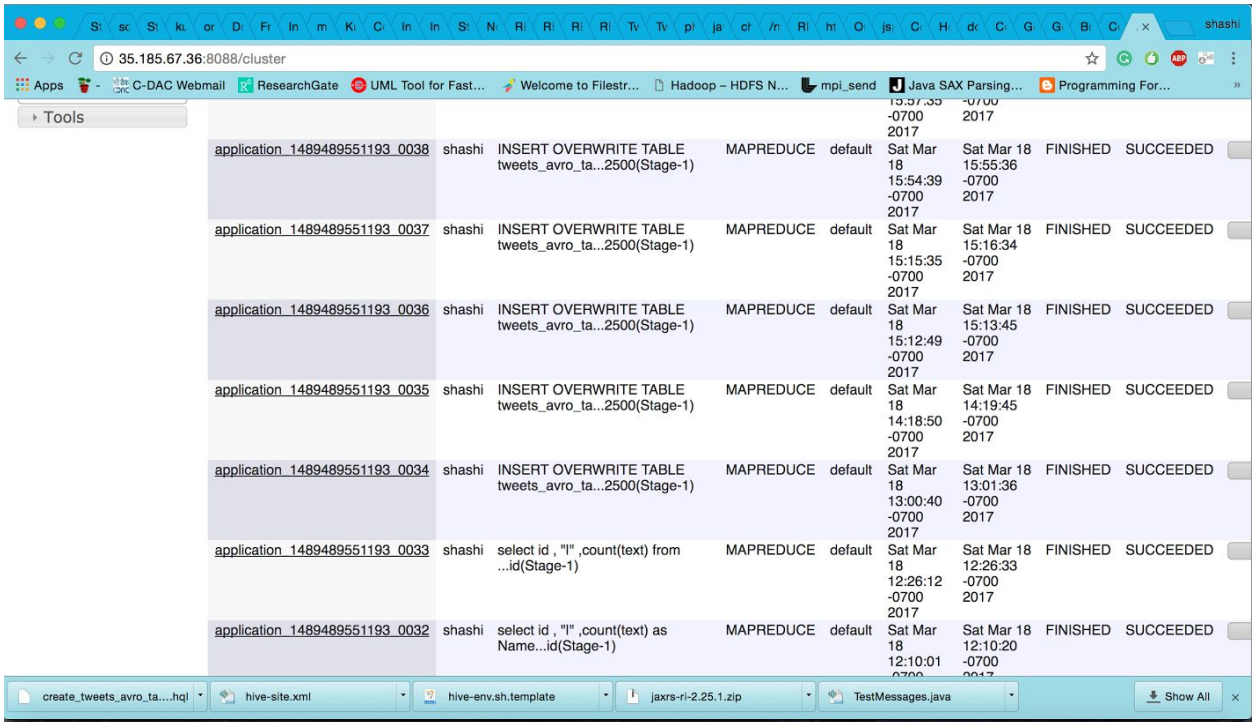
```
{"type":"record",
  "name":"Doc",
  "doc":"adoc",
  "fields":[{"name":"id","type":"string"},
            {"name":"user_friends_count","type":["int","null"]},
            {"name":"user_location","type":["string","null"]},
            {"name":"user_description","type":["string","null"]},
            {"name":"user_statuses_count","type":["int","null"]},
            {"name":"user_followers_count","type":["int","null"]},
            {"name":"user_name","type":["string","null"]},
            {"name":"user_screen_name","type":["string","null"]},
            {"name":"created_at","type":["string","null"]},
            {"name":"text","type":["string","null"]},
            {"name":"retweet_count","type":["long","null"]},
            {"name":"retweeted","type":["boolean","null"]},
            {"name":"in_reply_to_user_id","type":["long","null"]},
            {"name":"source","type":["string","null"]},
            {"name":"in_reply_to_status_id","type":["long","null"]},
            {"name":"media_url_https","type":["string","null"]},
            {"name":"expanded_url","type":["string","null"]}
           ]
}
```

## 5. Mapreduce :

Creation of second hive table where all the structured data will be stored and where all the queries will be operated on. This will have all processed data with Mapreduce operation invoked.

```
CREATE EXTERNAL TABLE tweets_avro_table (
    id                      string,
    user_friends_count      int    ,
    user_location           string,
    user_description        string,
    user_statuses_count     int    ,
    user_followers_count    int    ,
    user_name               string,
    user_screen_name        string,
    created_at              string,
    text                    string,
    retweet_count           bigint,
    retweeted               boolean,
    in_reply_to_user_id     bigint,
    source                  string,
    in_reply_to_status_id   bigint,
    media_url_https         string,
    expanded_url            string
    )
STORED AS AVRO
LOCATION '/user/externaltables/';

INSERT OVERWRITE TABLE tweets_avro_table SELECT * FROM tweets LIMIT
2500;
```

```
Logging initialized using configuration in jar:file:/home/shashi/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
OK
Time taken: 2.528 seconds

Logging initialized using configuration in jar:file:/home/shashi/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
OK
Time taken: 1.65 seconds
Query ID = shashi_20170318225732_d7175625-3f11-4e7b-ae12-11c2f783a92b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1489489551193_0039, Tracking URL = http://instance-1.c.ace-duality-157921.internal:8088/proxy/application_1489489551193_0039/
Kill Command = /home/shashi/hadoop-2.7.3/bin/hadoop job  -kill job_1489489551193_0039
Hadoop job information for Stage-1: number of mappers: 12; number of reducers: 1
2017-03-18 22:57:41,618 Stage-1 map = 0%,  reduce = 0%
2017-03-18 22:58:18,372 Stage-1 map = 8%,  reduce = 0%, Cumulative CPU 46.79 sec
2017-03-18 22:58:22,029 Stage-1 map = 17%,  reduce = 0%, Cumulative CPU 48.93 sec
2017-03-18 22:58:23,189 Stage-1 map = 25%,  reduce = 0%, Cumulative CPU 50.54 sec
2017-03-18 22:58:24,345 Stage-1 map = 42%,  reduce = 0%, Cumulative CPU 56.41 sec
2017-03-18 22:58:25,476 Stage-1 map = 75%,  reduce = 0%, Cumulative CPU 69.51 sec
2017-03-18 22:58:26,641 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 80.86 sec
2017-03-18 22:58:31,015 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 84.82 sec
MapReduce Total cumulative CPU time: 1 minutes 24 seconds 820 msec
Ended Job = job_1489489551193_0039
Loading data to table default.tweets_avro_table
Table default.tweets_avro_table stats: [numFiles=0, numRows=2500, totalSize=0, rawDataSize=0]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 12  Reduce: 1   Cumulative CPU: 84.82 sec   HDFS Read: 28772039 HDFS Write: 975633 SUCCESS
Total MapReduce CPU Time Spent: 1 minutes 24 seconds 820 msec
OK
Time taken: 61.51 seconds

Logging initialized using configuration in jar:file:/home/shashi/apache-hive-1.2.1-bin/lib/hive-common-1.2.1.jar!/hive-log4j.properties
OK
Time taken: 2.799 seconds, Fetched: 36 row(s)
 calling web service1  2  execption
false
 calling web service1  3 execption
```




## 6.Rest Services (SnapShot of Code)

Three rest services are mentioned here

1)Tweets Based on timestamp

2)Number of tweets for particular id

3)Number of retweet count of given id

serManagement/src/com/tutorialspoint/UserService.java - Eclipse - /Users/shashi/Documents/workspace_pyck

Quick Access | Java EE | Java

*User.java | *UserDao.java | *UserService.java

UserManagement ▸ src ▸ com.tutorialspoint ▸ UserService ▸ getUsers() : List<User>

```
1  package com.tutorialspoint;
2
3
4  import java.util.List;
10 @Path("/UserService")
11
12 public class UserService {
13    UserDao userDao = new UserDao();
14
15
16    @GET
17    @Path("/users")
18    @Produces(MediaType.APPLICATION_XML)
19    public List<User> getUsers(){
20       return userDao.getAllUsers();
21    }
22
23    @GET
24    @Path("/users/{timestamp}")
25    @Produces(MediaType.APPLICATION_XML)
26    public List<User> getUser(@PathParam("timestamp") String timestamp){
27
28        System.out.println(" "+ timestamp);
29       return userDao.getTweets(timestamp);
30
31    }
32
33
34    @GET
35    @Path("/user/{userid}")
36    @Produces(MediaType.APPLICATION_XML)
37    public User getUser(@PathParam("userid") long userid ){
38
```

Markers | Propertie | Servers | Data Sour | Snippets | Problems | Console | Git Repos | Git Stagin | History | Terminal

No consoles to display at this time


*User.java | *UserDao.java | *UserService.java

UserManagement ▸ src ▸ com.tutorialspoint ▸ StreamGobbler

```
85 public List<User> getTweets(String timestamp) {
86     // TODO Auto-generated method stub
87
88
89     List<User> al = new ArrayList<User>();
90     try
91     {
92     Runtime rt = Runtime.getRuntime();
93     Process proc = rt.exec(" sh /home/shashi/finalscript1.sh   "+timestamp);
94     //output both stdout and stderr data from proc to stdout of this process
95     StreamGobbler errorGobbler = new StreamGobbler(proc.getErrorStream());
96     StreamGobbler outputGobbler = new StreamGobbler(proc.getInputStream());
97     errorGobbler.start();
98     outputGobbler.start();
99     proc.waitFor();
100
101    }
102    catch(Exception e1)
103    {
104
105        System.out.println(" calling web service1 execption ");
106    }
107
108
109
110        BufferedReader  bfr =null;
111
112        try
113        {
114
115       File F1= new File("/home/shashi/output1");
116        System.out.println(F1.exists());
```


Markers | Propertie | Servers | Data Sour | Snippets | Problems | Console | Git Repos | Git Stagin | History | Terminal

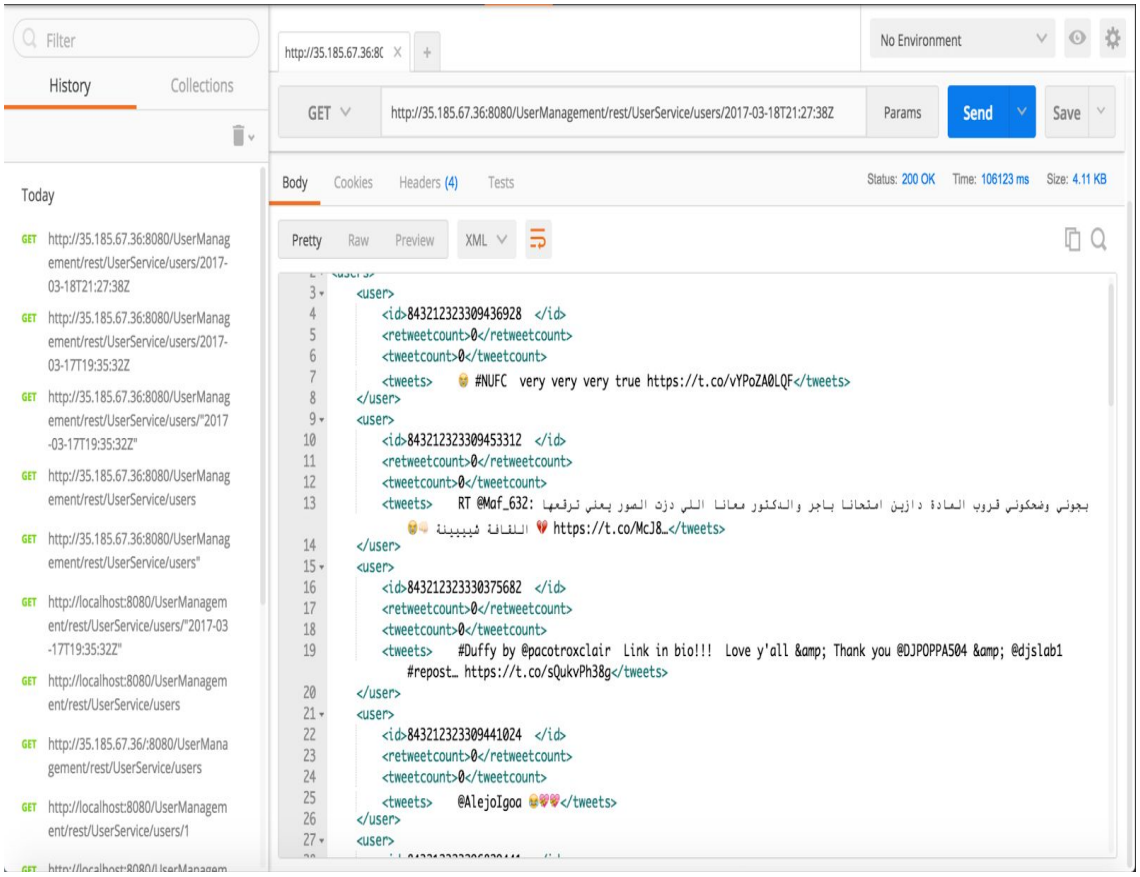
No consoles to display at this time

## 7. Queries Output :

Query 1 :

Input : Timestamp (timestamp passes in url )

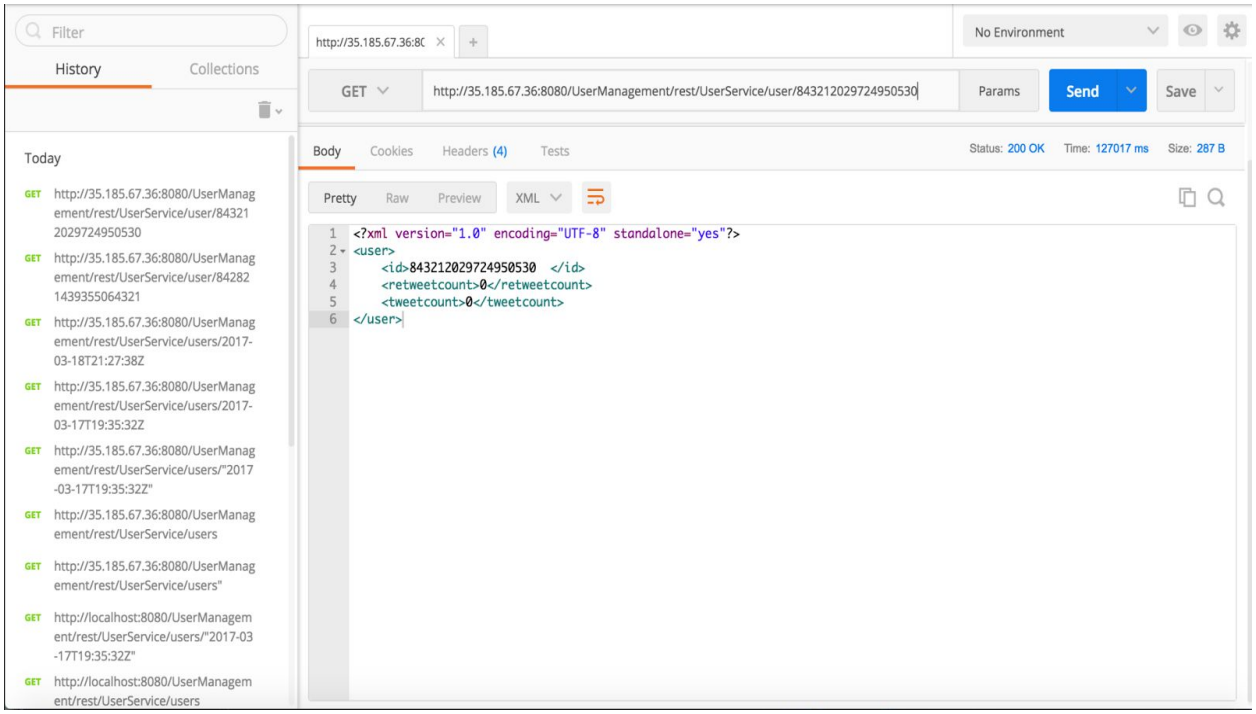
Output : UserId, Tweet text based on timestamp




Query 2

Input : UserId (User Id passed in URL )

Output : UserId, number of tweets for user




Query 3 :

Input : UserId (many User Ids passed in URL)

Output : UserId, Retweet Count