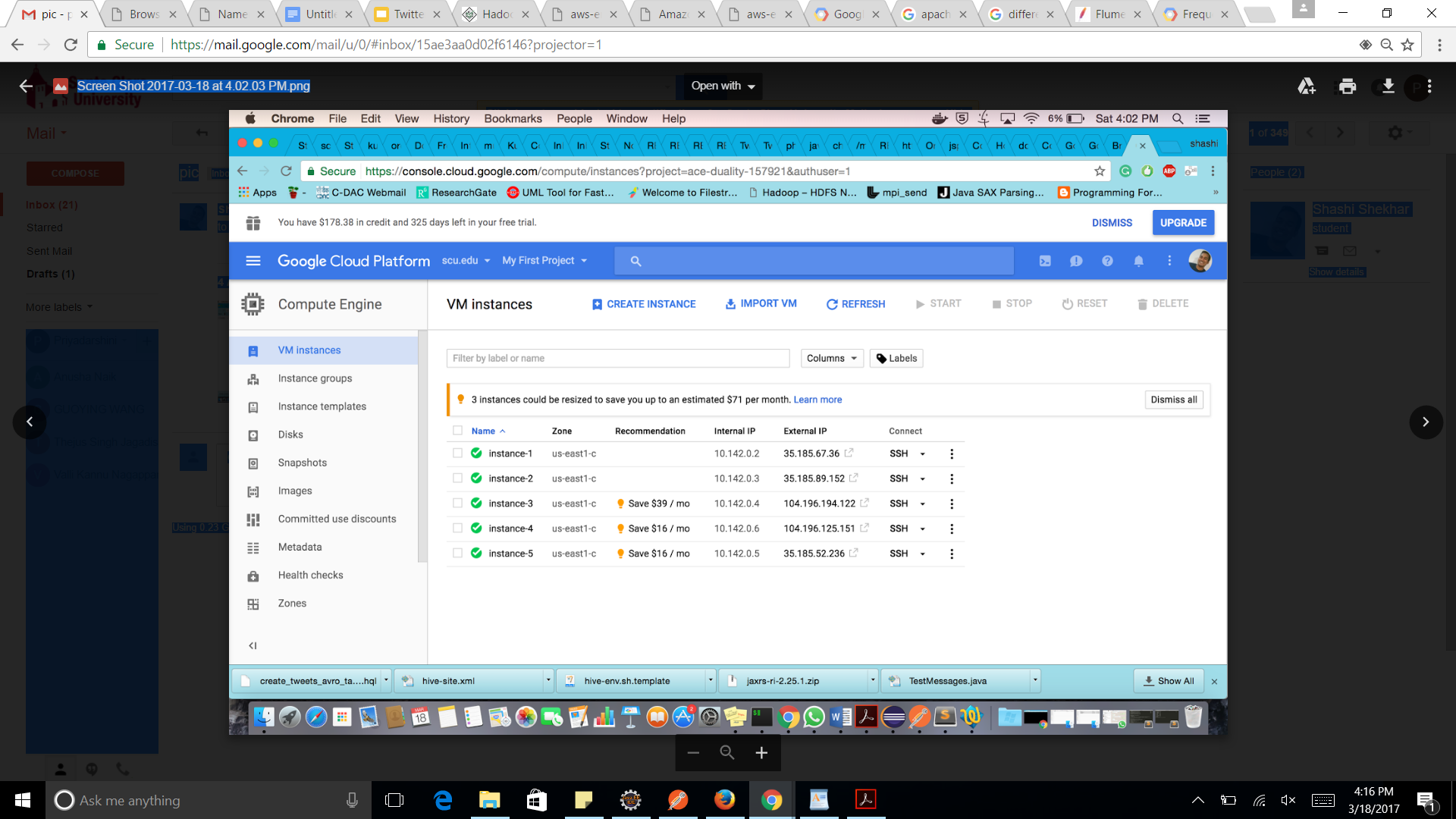
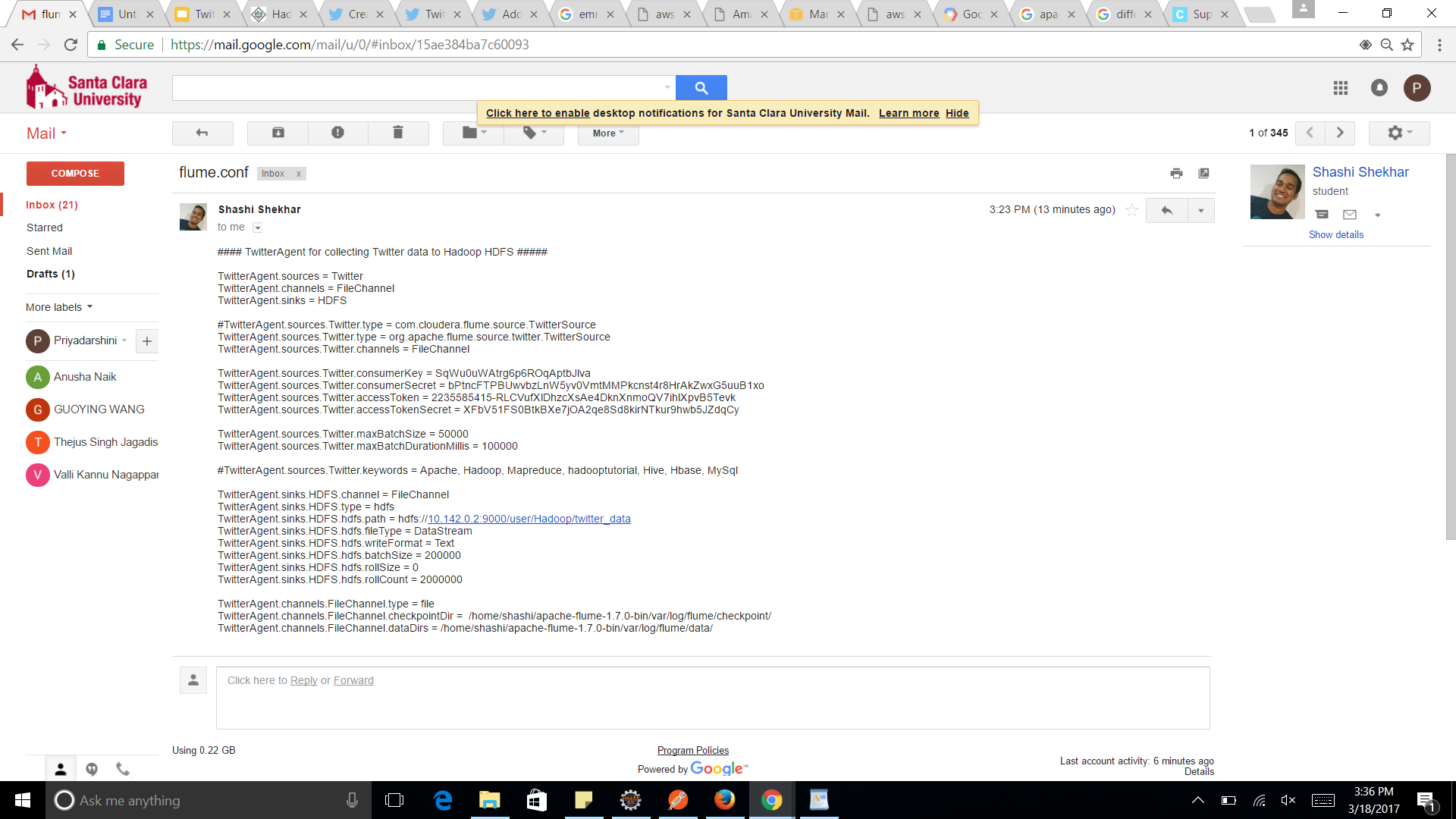
**Team 6 - TWITTER DATA ANALYTICS IMPLEMENTATION ON GOOGLE CLOUD**

**Screenshots of each step**

1. Creation of Google Instances :

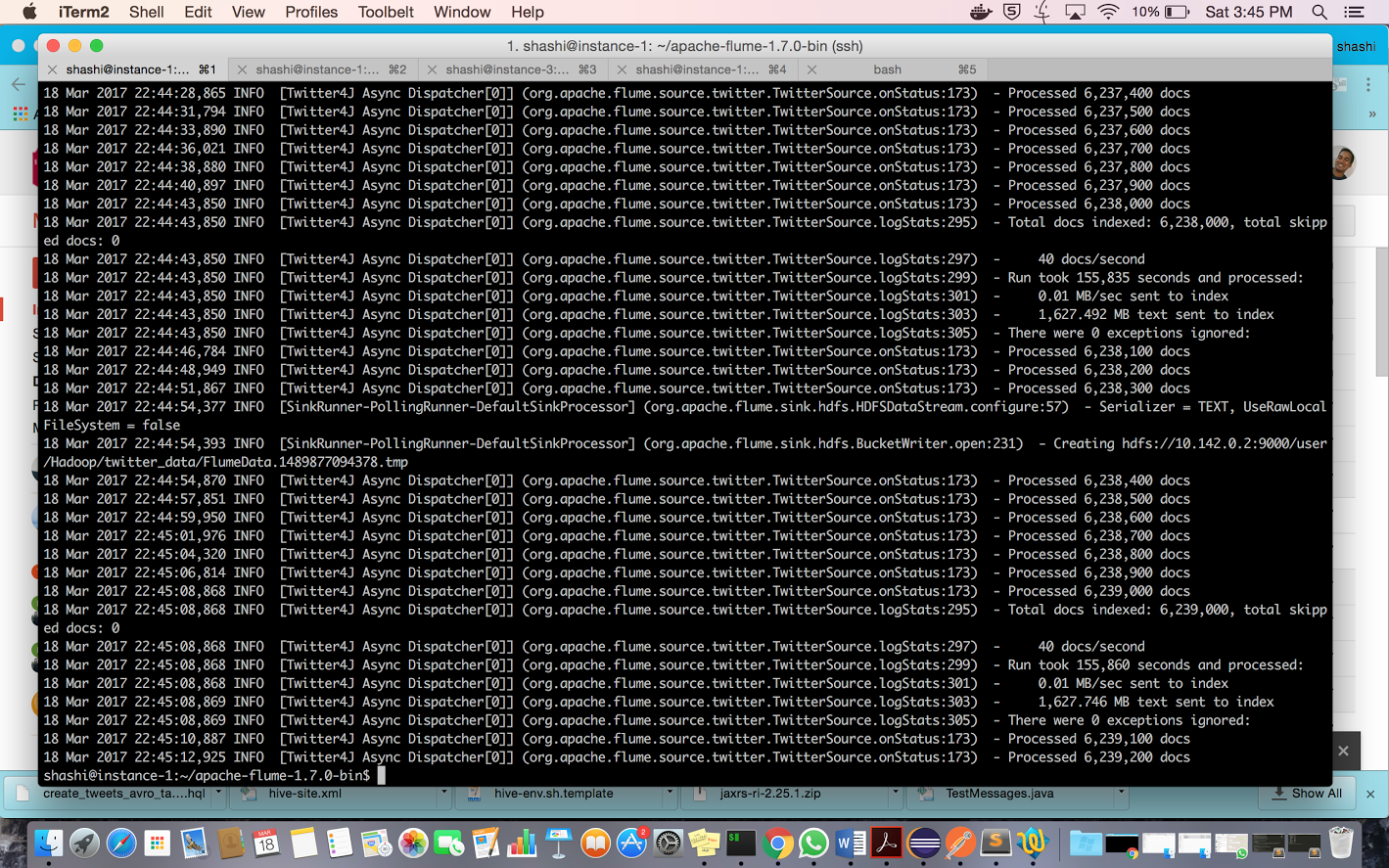


2. Apache Flume Configuration file



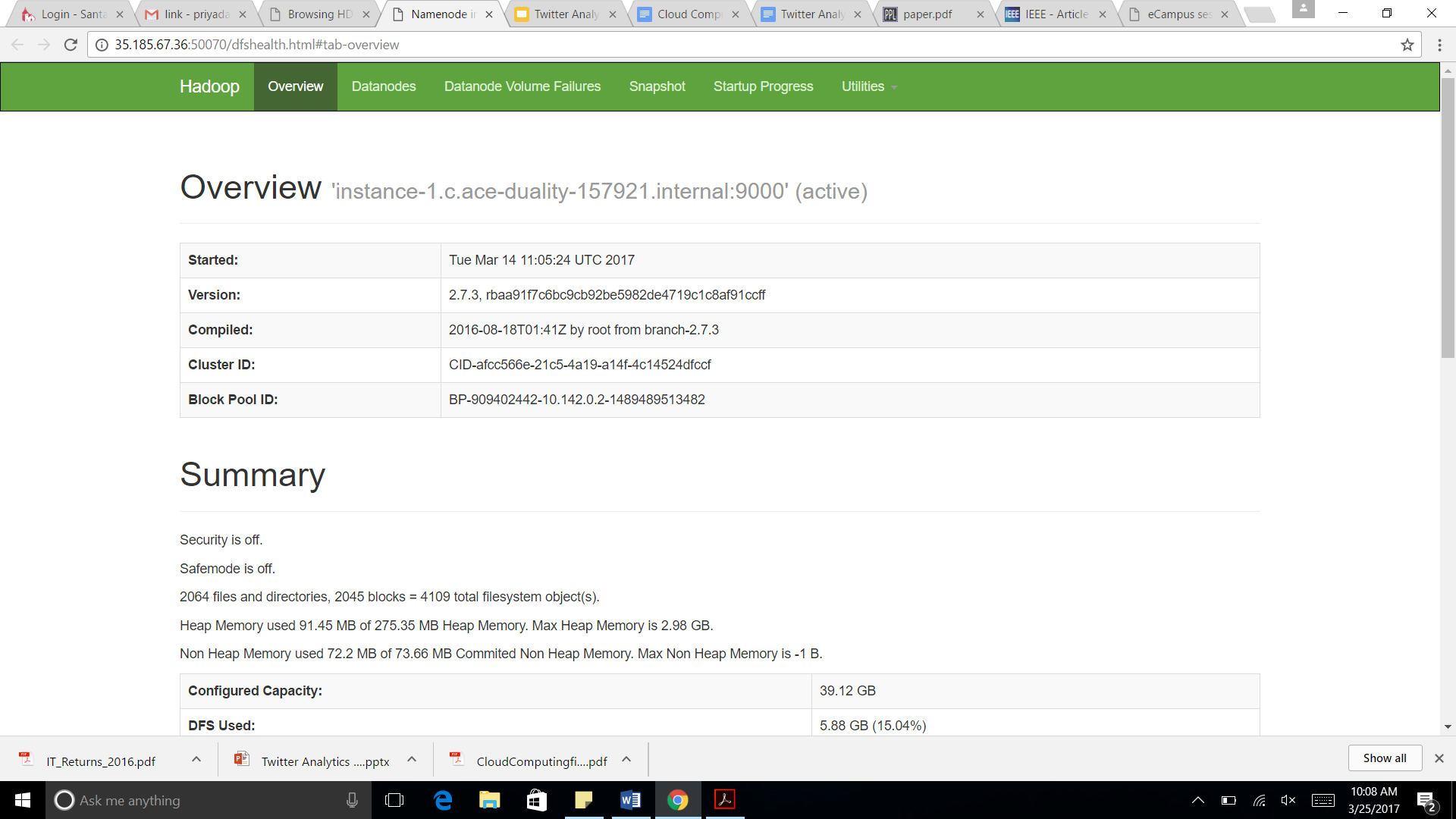
3. Execution of command to load data from Twitter source and loading into HDFS

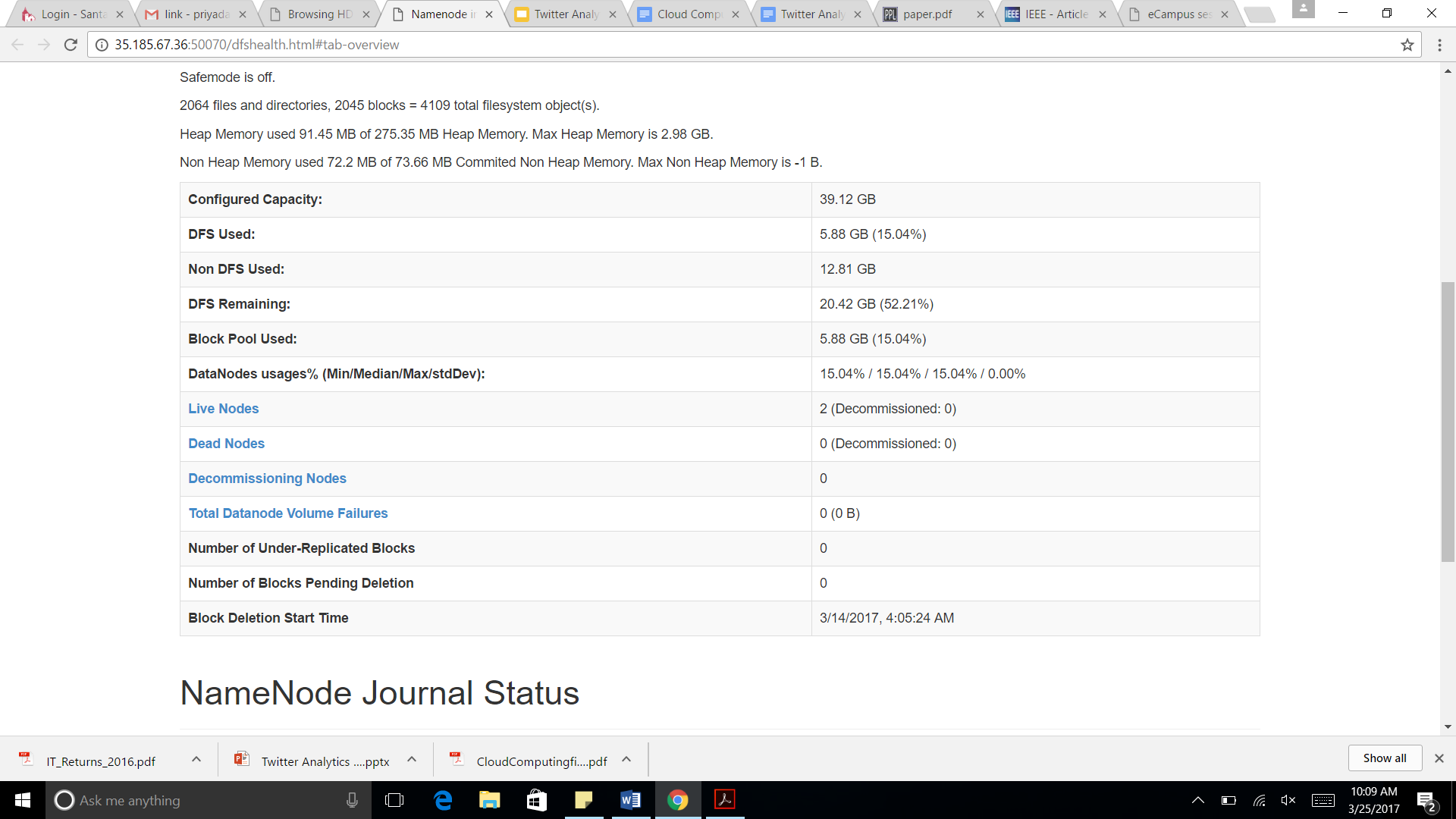
bin/flume-ng agent --conf ./conf/ -f conf/twit.conf Dflume.root.logger=DEBUG,console -n TwitterAgent >> log1 2>&1 &



Namenode Server Check(35.185.67.36:50070)

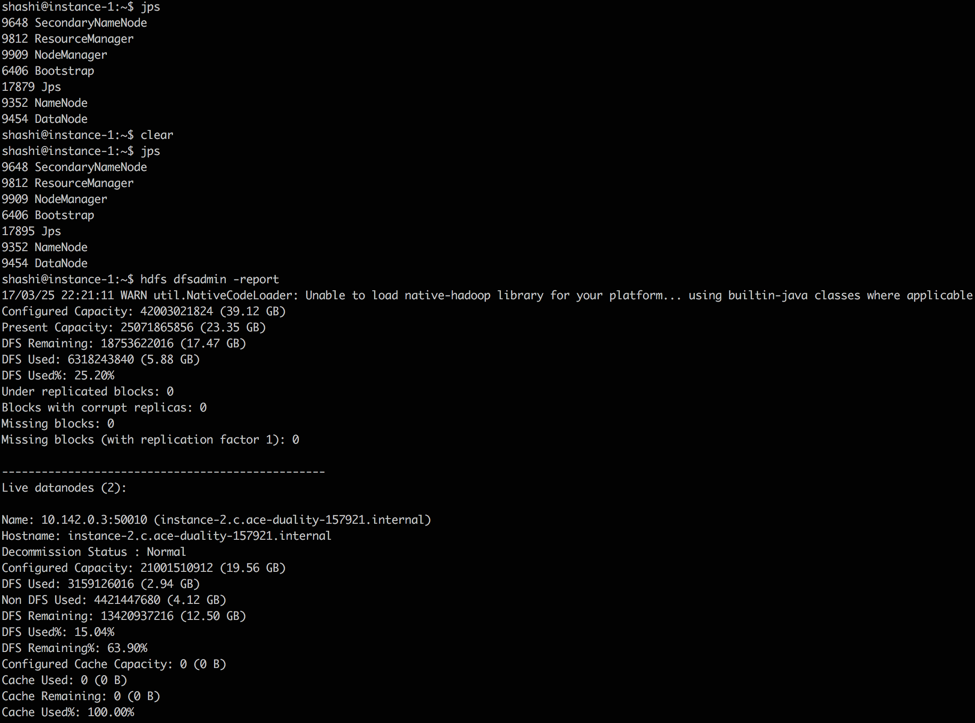
<http://35.185.67.36:50070/dfshealth.html#tab-overview>





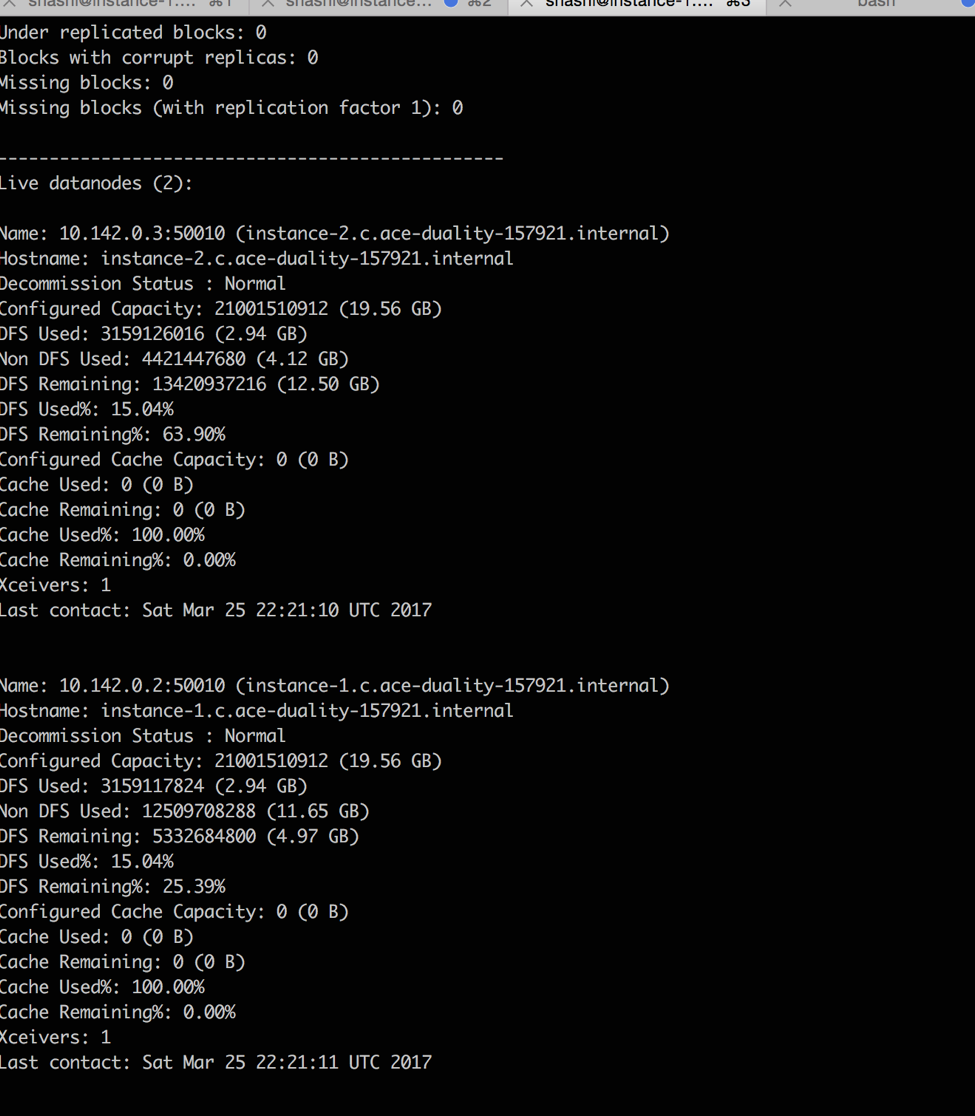
Hadoop daemon process by command line

jps (command)



4. Report of Hadoop cluster from command line

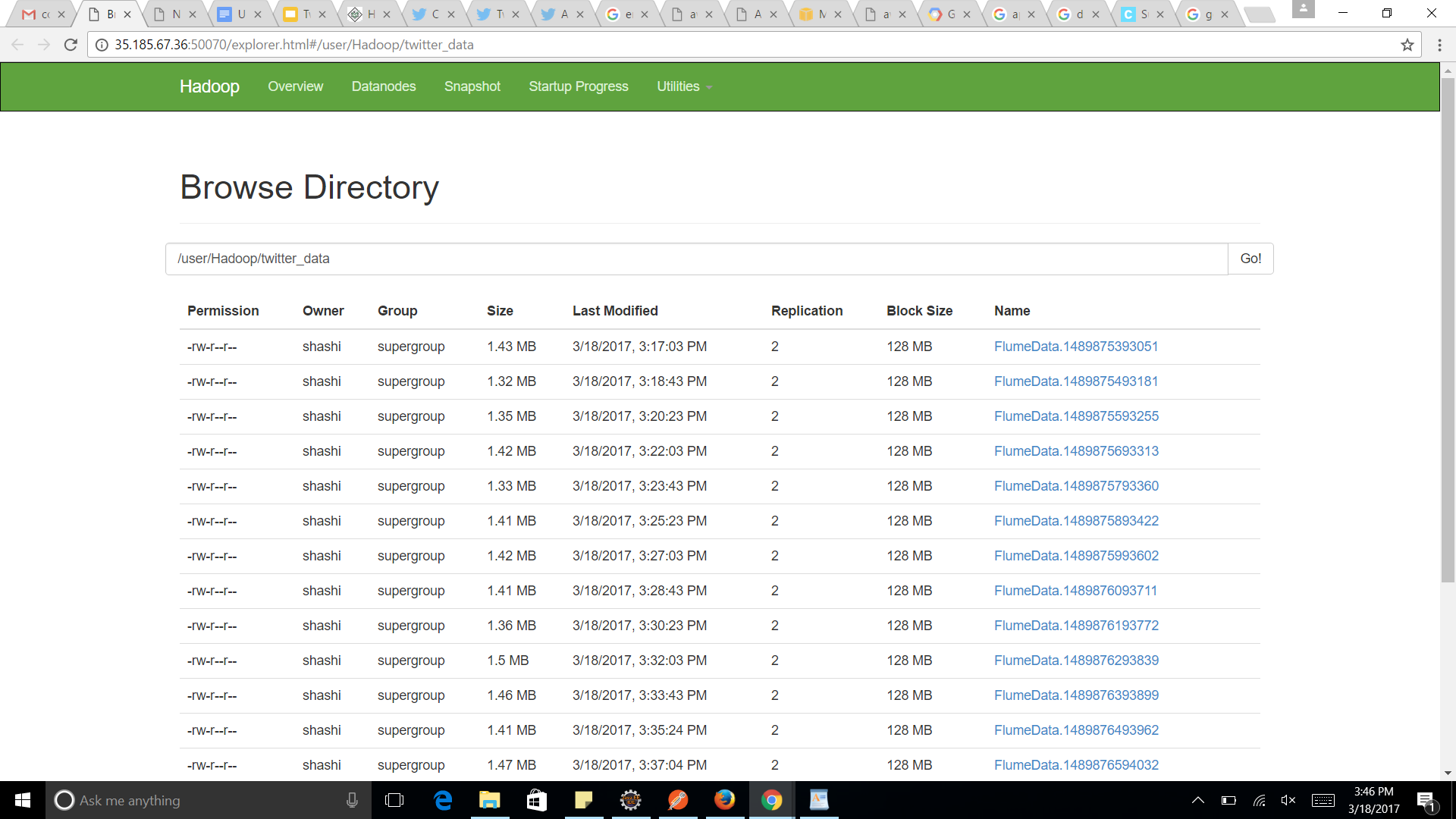
hdfs dfsadmin -report



5. Checking files loaded in HDFS from UI

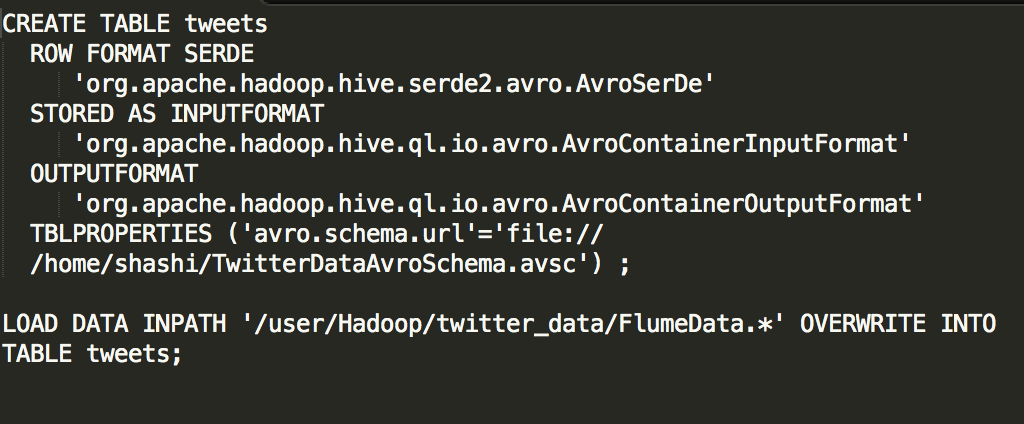
<http://35.185.67.36:50070/explorer.html#/user/Hadoop/twitter_data>

Click the link for seeing the data . We will not be able to see any data file currently as all the files are processed by copying them to the tables. Once the data is copied the files are flushed from HDFS. .



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

Creating tweets table

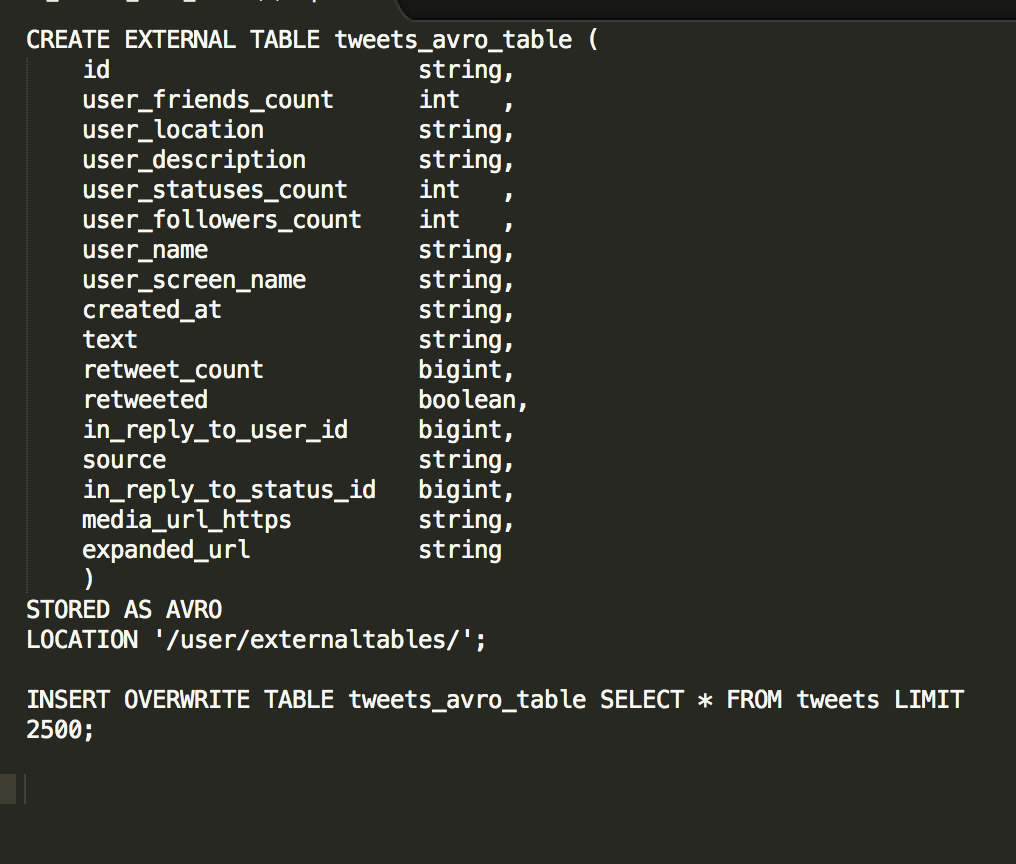


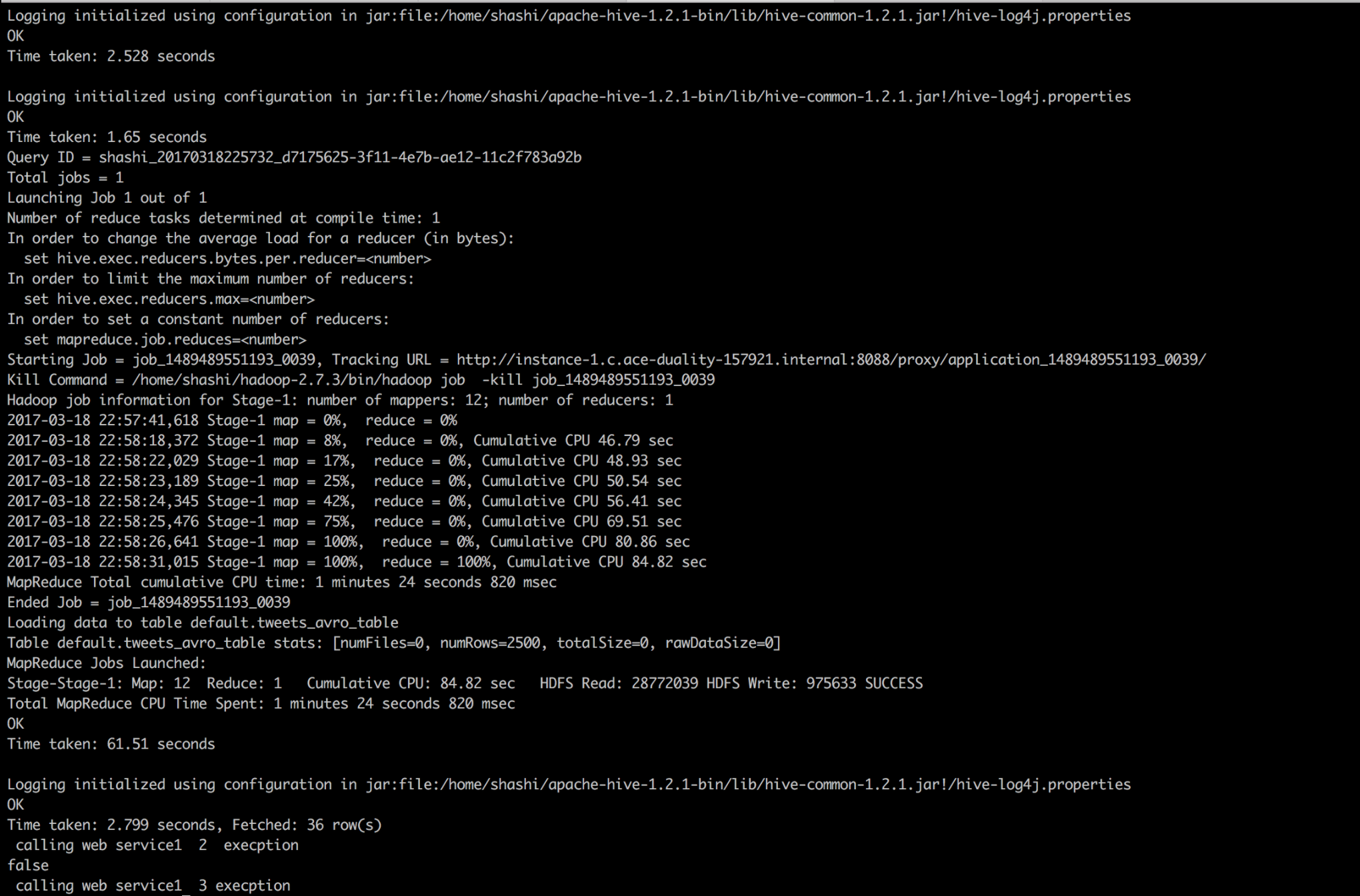
Schema is defined below



7. 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.





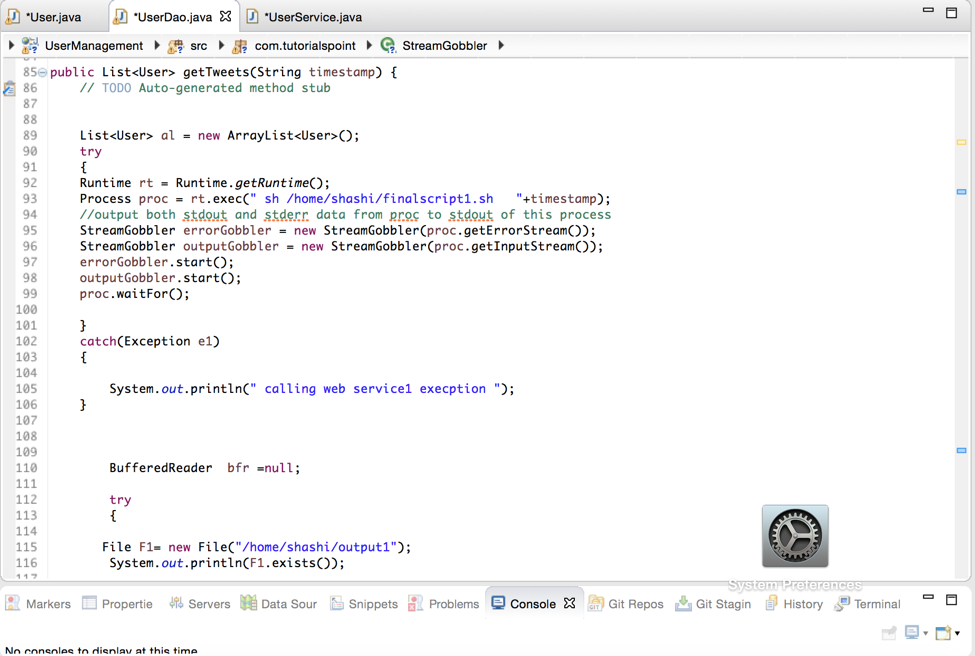


8. Web Services (Rest API) (SnapShot of Code)

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



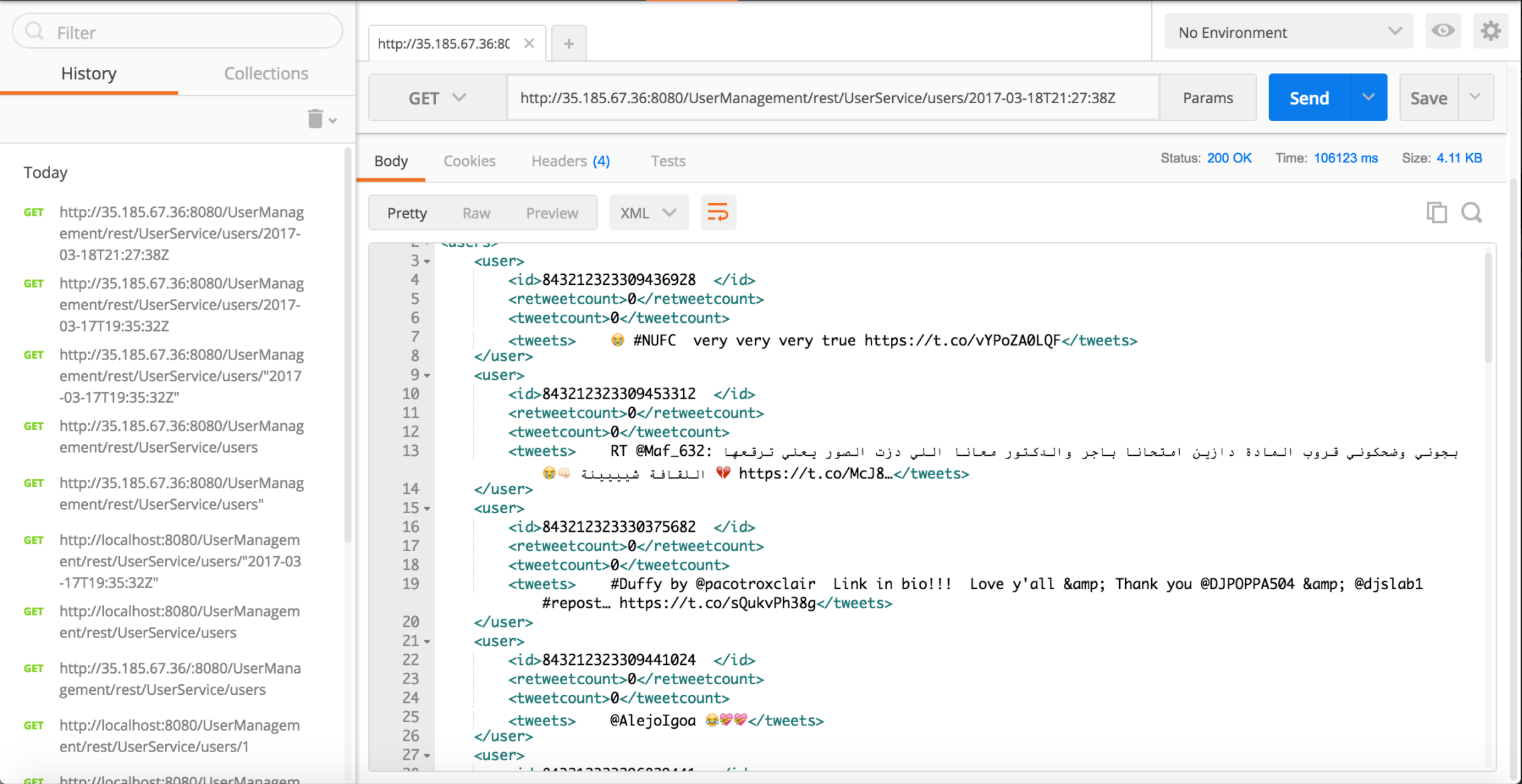


9. Queries Output :

Query 1 :

Input : Timestamp (timestamp passes in url )

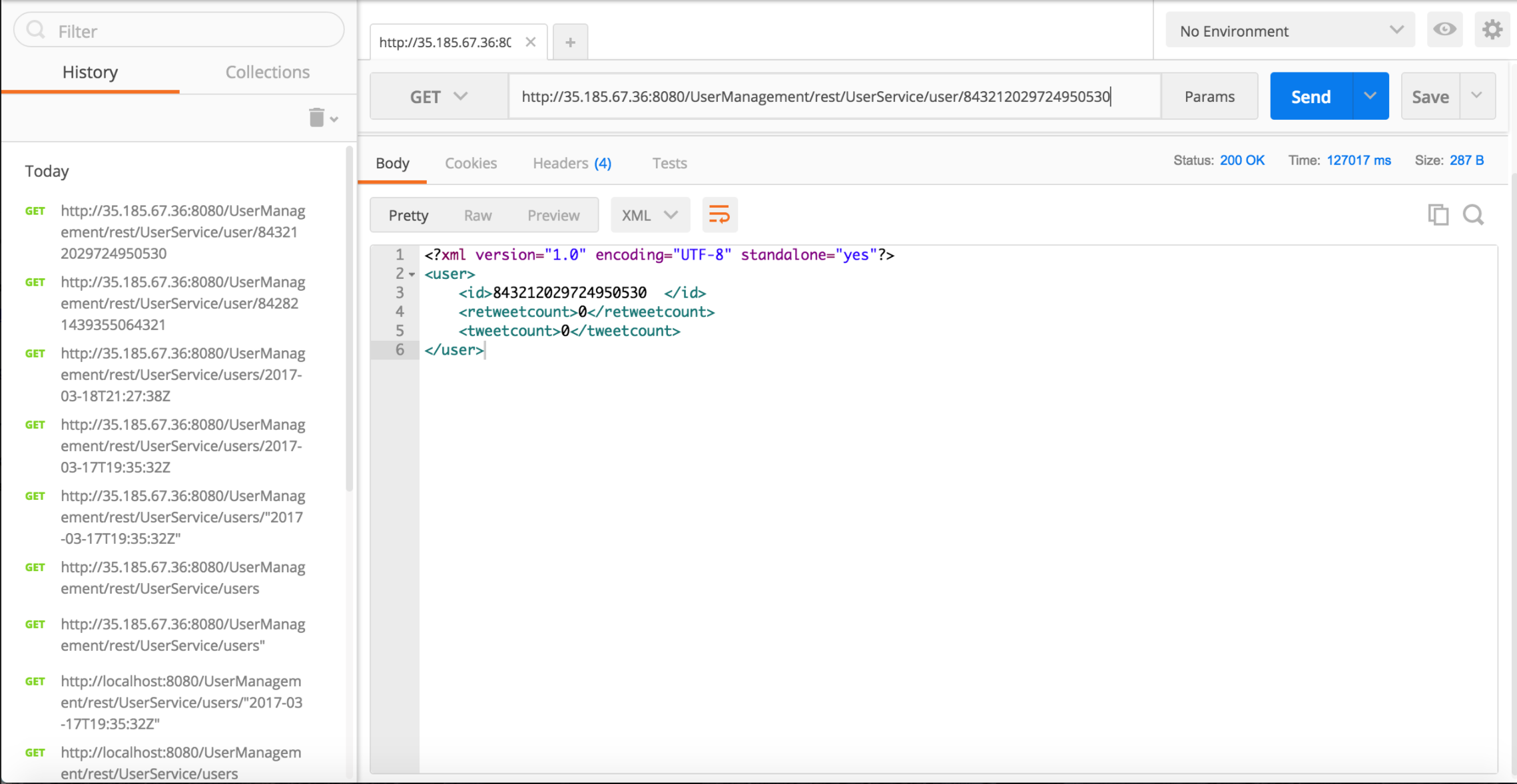
Output : UserId, Tweet text based on timestamp



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

