



Institute of Engineering & Technology

MID TERM REPORT

On

TWITTER SENTIMENT ANALYSIS

Submitted by

**SAHIL SRIVASTAVA
SHIVANK GARG
KANISHK
SAMARTH AHUJA**

**Department of Computer Engineering & Applications
Institute of Engineering & Technology**



**GLA University
Mathura- 281406, INDIA**

PROBLEM STATEMENT

The problem in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature.

Whether the expressed opinion in a document, a sentence or an entity feature is positive, negative, or neutral.

OBJECTIVE

In this project, we will showcase how to perform Sentiment Analysis on Twitter data using Pig. To begin with, we will be collecting real-time tweets from Twitter using Flume. With the help of AFINN dictionary we can find positive sentiment from the data which we have dumped into our HDFS/FLUME. We can prioritize our own customized words in AFINN dictionary as per the need of Client.

As this project is most advance use case of Hadoop in MNC & has wide impact in sentiment analysis. We need most advance concepts of PIG & MAP-REDUCE for getting into this project.

The data from Twitter is in 'Json' format, so a Pig JsonLoader is required to load the data into Pig. You need to download the required jars for the JsonLoader.

PROJECT FUNCTIONALITIES

The functionalities of the proposed system are very user friendly and attractive. Some of the functionalities are as follows:

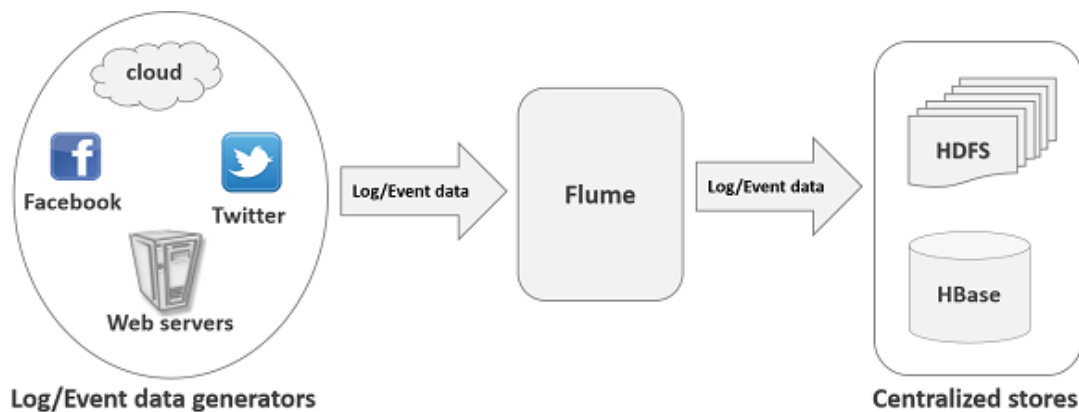
- Add category
- Add keyword
- Select File
- Collect tweets
- Pre-process tweets
- Analysis
- Web Application
- Providing the resultant tweets
- Providing suggestions

INTRODUCTION TO APACHE FLUME

What is flume?

Apache Flume is a tool/service/data ingestion mechanism for collecting aggregating and transporting large amounts of streaming data such as log files, events (etc...) from various sources to a centralized data store.

Flume is a highly reliable, distributed, and configurable tool. It is principally designed to copy streaming data (log data) from various web servers to HDFS.



Applications of Flume

Assume an e-commerce web application wants to analyze the customer behavior from a particular region. To do so, they would need to move the available log data in to Hadoop for analysis. Here, Apache Flume comes to our rescue.

Flume is used to move the log data generated by application servers into HDFS at a higher speed.

Advantages of Flume

1. Using Apache Flume, we can store the data in to any of the centralized stores (HBase, HDFS).
2. Flume is reliable, fault tolerant, scalable, manageable, and customizable.
3. When the rate of incoming data exceeds the rate at which data can be written to the destination, Flume acts as a mediator between data producers and the centralized stores and provides a steady flow of data between them.

4. Flume provides the feature of contextual routing.
5. The transactions in Flume are channel-based where two transactions (one sender and one receiver) are maintained for each message. It guarantees reliable message delivery.

Features of Flume

1. Flume ingests log data from multiple web servers into a centralized store (HDFS, HBase) efficiently.
2. Using Flume, we can get the data from multiple servers immediately into Hadoop.
3. Along with the log files, Flume is also used to import huge volumes of event data produced by social networking sites like Facebook and Twitter, and e-commerce websites like Amazon and Flipkart.
4. Flume supports a large set of sources and destinations types.
5. Flume supports multi-hop flows, fan-in fan-out flows, contextual routing, etc.

How Flume helps Hadoop to get data from live streaming?

1. Flume allows the user to do the following:
2. Stream data into Hadoop from multiple sources for analysis.
3. Collect high-volume web logs in real-time.
4. It acts as a buffer when the rate of incoming data exceeds the rate at which the data can be written. Thereby preventing data loss.
5. Guarantees data delivery.
6. Scales horizontally (connects commodity system in parallel) to handle additional data volume.

Essential Components Involved in Getting Data from a Live-Streaming Source

There are 3 major components, namely: Source, Channel, and Sink, which are involved in ingesting data, moving data and storing data, respectively. Below is the breakdown of the parts applicable in this scenario:

1. **Event** – A singular unit of data that is transported by Flume (typically a single log entry).
2. **Source** – The entity through which data enters into the Flume. Sources either actively samples the data or passively waits for data to be delivered to them. A variety of sources such as log4j logs and syslog's, allows data to be collected.
3. **Sink** – The unit that delivers the data to the destination. A variety of sinks allow data to be streamed to a range of destinations. Example: HDFS sink writes events to the HDFS.
4. **Channel** – It is the connection between the Source and the Sink. The Source ingests Event into the Channel and the Sink drains the Channel.
5. **Agent** – Any physical Java virtual machine running Flume. It is a collection of Sources, Sinks and Channels.
6. **Client** – It produces and transmits the Event to the Source operating within the Agent

INTRODUCTION TO HADOOP

Hadoop is a framework which helps in storing and processing huge datasets and Sqoop component is used to transfer files from traditional databases like RDBMS to HDFS and vice versa when the data is of the structured type.

What if we want to load the data which is of type semi-structured and unstructured into the HDFS cluster, or else capture the live streaming data which is generated, from different sources like twitter, weblogs and more into the HDFS cluster, which component of Hadoop ecosystem will be useful to do this kind of job. The solution is FLUME.

Micro blogging today has become a very popular communication tool among internet users. Twitter, one of the largest social media sites receives millions of tweets every day on variety of important issues. Authors of those messages write about their life, share opinions on variety of topics and discuss current issues. These posts analysis can be used for decision making in different areas like government, Electronics, Business, Product review etc. Also, sentiment analysis is one of the important areas of analysis of twitter posts that can be very helpful in decision making.

Performing Sentiment Analysis on Twitter is trickier than doing it for large reviews. This is because the tweets are very short (only about 140 characters) and usually contain slangs, emoticons, hash tags and other twitter specific jargon. For the development purpose twitter provides streaming API which allows the developer an access to 1% of tweets tweeted at that time bases on the particular keyword. The object about which we want to perform sentiment analysis is submitted to the twitter API's which does further mining and provides the tweets related to only that object. Twitter data is generally unstructured i.e. use of abbreviations is very high. Also, it allows the use of emoticons which are direct indicators of the author's view on the subject. Tweet messages also consist of a timestamp and the user name.

INTRODUCTION TO BIG DATA

What is Data?

The quantities, characters, or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media.

What is Big Data?

Big Data is also data but with a huge size. Big Data is a term used to describe a collection of data that is huge in size and yet growing exponentially with time. In short, such data is so large and complex that none of the traditional data management tools are able to store it or process it efficiently.

Examples of Big Data

The New York Stock Exchange generates about one terabyte of new trade data per day.



Social media

The statistics shows that 500+ terabytes of new data ingested into the databases of social media site Facebook every day



A single jet engine can generate 10+terabytes of data in 30 minutes of flight time. It turns up to petabytes.



Types of Big Data

1. Structured
2. Unstructured
3. Semi-structured

Structured

Any data that can be stored, accessed and processed in the form of fixed format is termed as a 'structured' data. Over the period of time, talent in computer science has achieved greater success in developing techniques for working with such kind of data (where the format is well known in advance) and also deriving value out of it. However, nowadays, we are foreseeing issues when a size of such data grows to a huge extent, typical sizes are being in the rage of multiple zettabytes

Ex- An Employee table in the database

Unstructured

Any data with unknown form or the structure is classified as unstructured data. In addition to the size being huge, un-structured data poses multiple challenges in terms of its processing for deriving value out of it. A typical example of unstructured data is a heterogeneous data source containing a combination of simple text files, images, videos etc. Now day organizations have wealth of data available with them but unfortunately, they don't know how to derive value out of it since this data is in its raw form or unstructured format

Ex-the output returned by google search

Semi-structured

Semi-structured data can contain both the forms of data. We can see semi-structured data as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS. Example of semi-structured data is a data represented in an XML file.

Ex-personal data stored in xml file

Characteristics of Big Data

- (i) **Volume** – The name Big Data itself is related to a size which is enormous. Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, **volume** is one characteristic which needs to be considered while dealing with Big Data.
- (ii) **Variety** – The next aspect of Big Data is its **variety**. Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications. Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.
- (iii) **Velocity** – The term **velocity** refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data.

(iv) Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensor devices, etc. The flow of data is massive and continuous.

(v) **Variability** – This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.

Benefits of Big Data Processing

Ability to process Big Data brings in multiple benefits, such as-

Businesses can utilize outside intelligence while taking decisions

Improved customer services

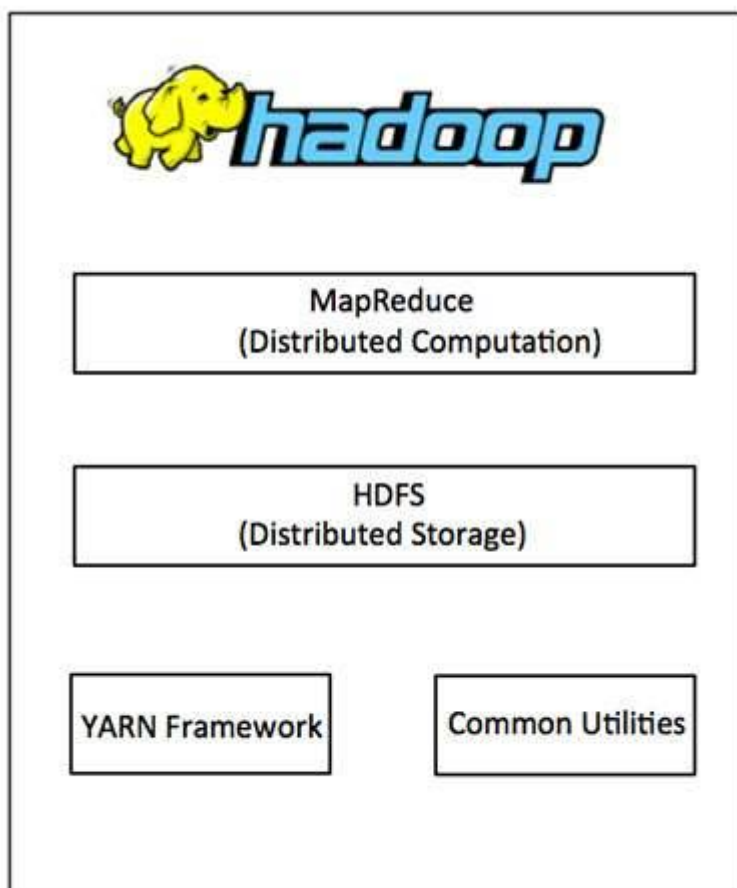
INTRODUCTION TO HADOOP

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Hadoop Architecture

At its core, Hadoop has two major layers namely –

1. Processing/Computation layer (MapReduce)
2. Storage layer (Hadoop Distributed File System)



MapReduce

MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open-source framework.

Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. It is highly fault-tolerant and is designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications having large datasets.

Apart from the above-mentioned two core components, Hadoop framework also includes the following two modules –

Hadoop Common – These are Java libraries and utilities required by other Hadoop modules.

Hadoop YARN – This is a framework for job scheduling and cluster resource management.

How Does Hadoop Work?

It is quite expensive to build bigger servers with heavy configurations that handle large scale processing, but as an alternative, you can tie together many commodity computers with single-CPU, as a single functional distributed system and practically, the clustered machines can read the dataset in parallel and provide a much higher throughput. Moreover, it is cheaper than one high-end server. So, this is the first motivational factor behind using Hadoop that it runs across clustered and low-cost machines. Hadoop runs code across a cluster of computers. This process includes the following core tasks that Hadoop performs –

- Data is initially divided into directories and files. Files are divided into uniform sized blocks of 128M and 64M (preferably 128M).
- These files are then distributed across various cluster nodes for further processing.

- HDFS, being on top of the local file system, supervises the processing.
- Blocks are replicated for handling hardware failure.
- Checking that the code was executed successfully.
- Performing the sort that takes place between the map and reduce stages.
- Sending the sorted data to a certain computer.
- Writing the debugging logs for each job.

Advantages of Hadoop

1. Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
2. Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
3. Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
4. Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

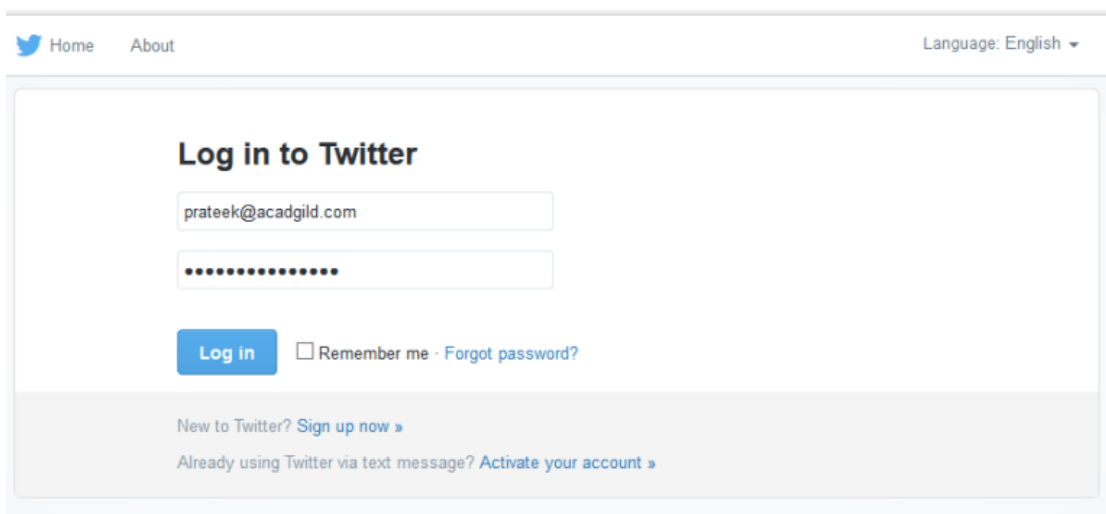
IMPLEMENTATION

Let's look at the necessary pre-requisites:

- Twitter account
- Install Hadoop/Start Hadoop

Data Streaming from Twitter to HDFS

Step 1: Open a Twitter developer account

A screenshot of the Twitter login page. At the top, there are links for 'Home' and 'About' on the left, and 'Language: English' with a dropdown arrow on the right. The main content area is titled 'Log in to Twitter'. It contains two input fields: the first for a username (pre-filled with 'prateek@acadgild.com') and the second for a password (masked with dots). Below these fields is a blue 'Log in' button, a checkbox for 'Remember me', and a link for 'Forgot password?'. At the bottom of the login area, there are two links: 'New to Twitter? Sign up now »' and 'Already using Twitter via text message? Activate your account »'.

Step 2: Go to the following link and click on 'create app'.
<https://apps.twitter.com/app>



Step 3: Fill in the necessary details.

Create an application

Application Details

Name *

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

Description *

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

Website *

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

Where should we return after successful authentication? OAuth 1.0a applications should explicitly specify their oauth_callback in the request token step.

Step 4: Accept the agreement and click on 'create your Twitter application'.

regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Agreement

Effective: May 18, 2015.

This Twitter Developer Agreement ("**Agreement**") is made between you (either an individual or an entity, referred to herein as "**you**") and Twitter, Inc. and Twitter International Company (collectively, "**Twitter**") and governs your access to and use of the Licensed Material (as defined below).

PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING WITHOUT LIMITATION ANY LINKED TERMS AND CONDITIONS APPEARING OR REFERENCED BELOW, WHICH ARE HEREBY MADE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU HAVE READ, AND THAT YOU AGREE TO COMPLY WITH AND TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALL APPLICABLE LAWS AND REGULATIONS IN THEIR ENTIRETY WITHOUT LIMITATION OR QUALIFICATION. IF YOU DO NOT AGREE TO BE BOUND BY THIS AGREEMENT, THEN YOU MAY NOT ACCESS OR OTHERWISE USE THE LICENSED MATERIAL. THIS AGREEMENT IS EFFECTIVE AS OF THE FIRST DATE THAT YOU USE THE LICENSED MATERIAL ("**EFFECTIVE DATE**").

IF YOU ARE AN INDIVIDUAL REPRESENTING AN ENTITY, YOU ACKNOWLEDGE THAT YOU HAVE THE APPROPRIATE AUTHORITY TO ACCEPT THIS AGREEMENT ON BEHALF OF SUCH ENTITY. YOU MAY NOT USE THE LICENSED MATERIAL AND MAY NOT ACCEPT THIS AGREEMENT IF YOU ARE NOT OF LEGAL AGE TO FORM A BINDING CONTRACT WITH TWITTER, OR YOU ARE

☒ Yes, I agree

Create your Twitter application

Step 5: Go to 'Keys and Access Token' tab.

Your application has been created. Please take a moment to review and adjust your application's settings.

acadgildApp

[Test OAuth](#)

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)



This app will help me do analysis in flume
<http://www.yahoo.com>

Organization

Information about the organization or company associated with your application. This information is optional.

Organization None

Organization website None

Application Settings

Step 6: Copy the consumer key and the consumer secret

Step 7: Scroll down further and click on 'create my access token'

[Regenerate Consumer Key and Secret](#)[Change App Permissions](#)

Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

Token Actions

[Create my access token](#)

You will now receive a message that says that you have successfully generated your application access token.

Status

Your application access token has been **successfully generated**. It may take a moment for changes you've made to reflect.
[Refresh](#) if your changes are not yet indicated.

acadgildApp

[Test OAuth](#)

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)



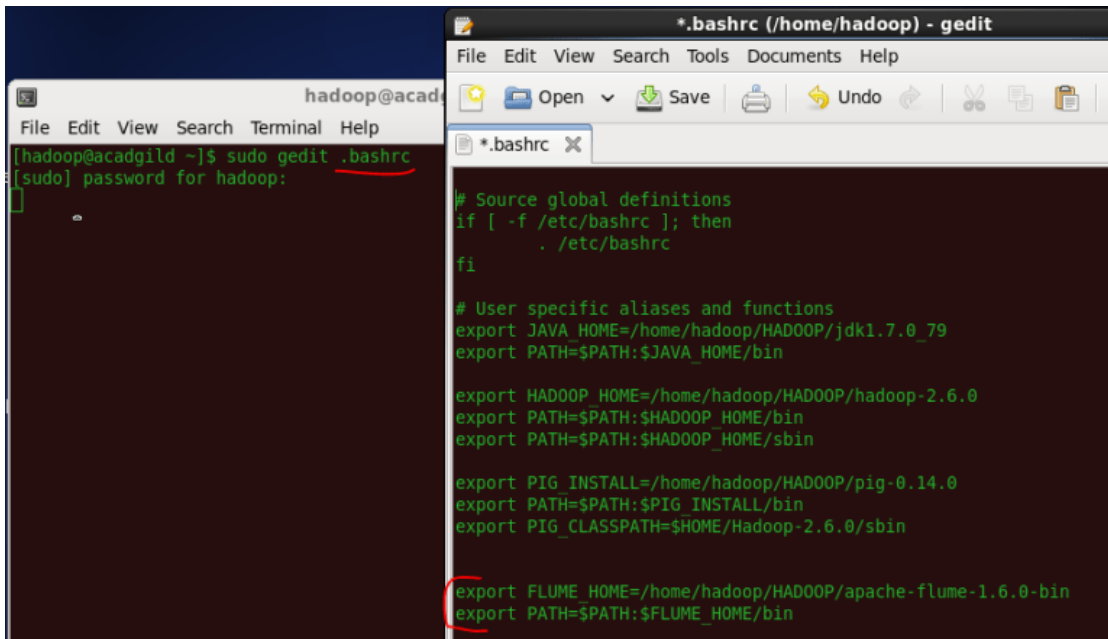
Step 8: Copy the Access Token and Access token Secret.

Step 9: Download flume tar file from below link and extract it

<https://drive.google.com/open?id=0B2nmxAJLHEE8ZGILeE05TEUtdEE>

Extract the flume.tar file and update the path of extracted file in .bashrc

NOTE: keep the path same as where the extracted file exists.



```
hadoop@acadgild ~]$ sudo gedit .bashrc
[sudo] password for hadoop:

*.bashrc (/home/hadoop) - gedit
File Edit View Search Tools Documents Help
*.bashrc
# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

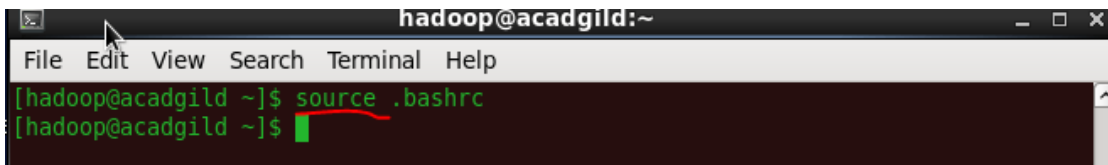
# User specific aliases and functions
export JAVA_HOME=/home/hadoop/HADOOP/jdk1.7.0_79
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/home/hadoop/HADOOP/hadoop-2.6.0
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

export PIG_INSTALL=/home/hadoop/HADOOP/pig-0.14.0
export PATH=$PATH:$PIG_INSTALL/bin
export PIG_CLASSPATH=$HOME/Hadoop-2.6.0/sbin

export FLUME_HOME=/home/hadoop/HADOOP/apache-flume-1.6.0-bin
export PATH=$PATH:$FLUME_HOME/bin
```

Update the bashrc file with source command.



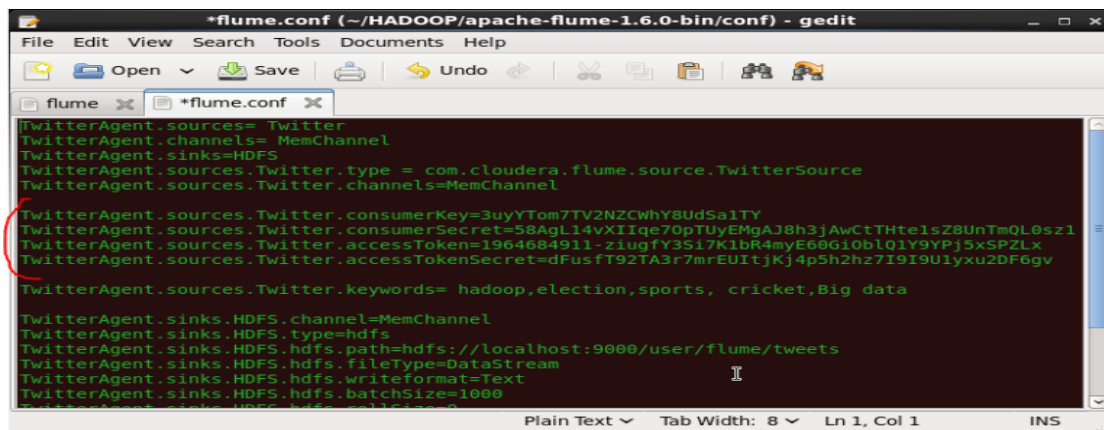
```
hadoop@acadgild:~
File Edit View Search Terminal Help
[hadoop@acadgild ~]$ source .bashrc
[hadoop@acadgild ~]$
```

Step 10: Create a new file inside the 'conf' directory inside the Flume-extracted directory.

Step 11: Copy the information from the below link and paste it inside the newly created file.

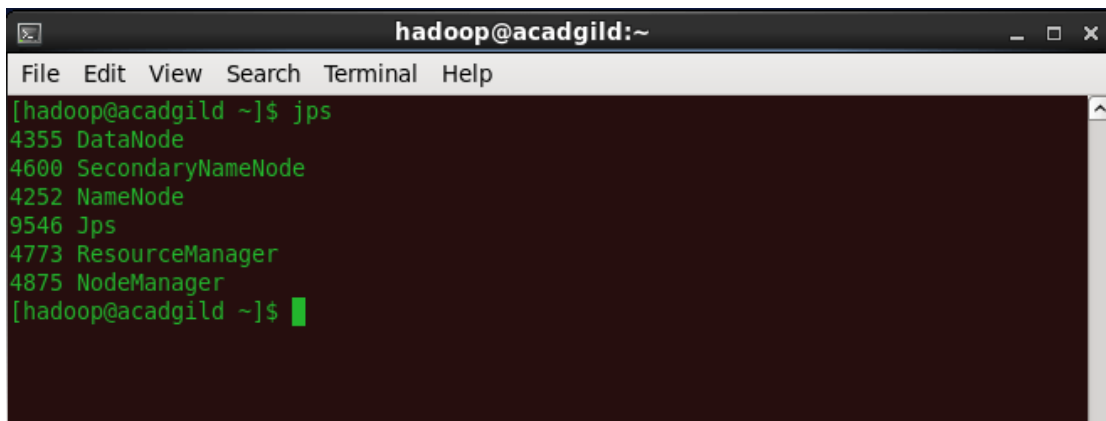
<https://drive.google.com/open?id=0B1QaXx7tpw3Sb3U4LW9SWlNidkk>

Step 12: Change the twitter api keys with the keys generated as shown in the step no 6 and step number



```
+flume.conf (~/.HADOOP/apache-flume-1.6.0-bin/conf) - gedit
File Edit View Search Tools Documents Help
flume *flume.conf
TwitterAgent.sources= Twitter
TwitterAgent.channels= MemChannel
TwitterAgent.sinks=HDFS
TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels=MemChannel
TwitterAgent.sources.Twitter.consumerKey=3uyYTom7TV2NZCWhY8UdSa1TY
TwitterAgent.sources.Twitter.consumerSecret=58Agl14vXIIqe70pTUYEMgAJ8h3jAwCtTHtelsZ8UnTmQL0sz1
TwitterAgent.sources.Twitter.accessToken=1964684911-zlugfY35i7K1bR4myE60G10bl01Y9YPj5xSPZLx
TwitterAgent.sources.Twitter.accessTokenSecret=dFusfT92TA3r7mrEUItjKj4p5h2hz7I9I9U1yxu2DF6gv
TwitterAgent.sources.Twitter.keywords= hadoop,election,sports, cricket,Big data
TwitterAgent.sinks.HDFS.channel=MemChannel
TwitterAgent.sinks.HDFS.type=hdfs
TwitterAgent.sinks.HDFS.hdfs.path=hdfs://localhost:9000/user/flume/tweets
TwitterAgent.sinks.HDFS.hdfs.fileType=DataStream
TwitterAgent.sinks.HDFS.hdfs.writeformat=Text
TwitterAgent.sinks.HDFS.hdfs.batchSize=1000
TwitterAgent.sinks.HDFS.hdfs.minSize=1000
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

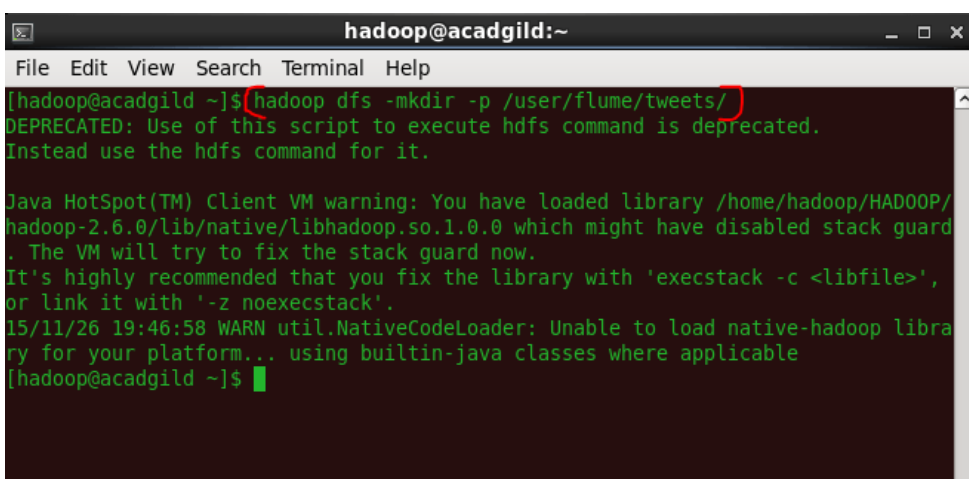
Step 13: Open the terminal to check for all Hadoop daemons running, by using the 'jps' command.



```
hadoop@acadgild:~
File Edit View Search Terminal Help
[hadoop@acadgild ~]$ jps
4355 DataNode
4600 SecondaryNameNode
4252 NameNode
9546 Jps
4773 ResourceManager
4875 NodeManager
[hadoop@acadgild ~]$
```

Step 14: Using the below command, create a directory inside HDFS where Twitter data will be stored.

Hadoop dfs -mkdir -p /user/flume/tweets



```
hadoop@acadgild:~
File Edit View Search Terminal Help
[hadoop@acadgild ~]$ hadoop dfs -mkdir -p /user/flume/tweets/
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Java HotSpot(TM) Client VM warning: You have loaded library /home/hadoop/HADOOP/
hadoop-2.6.0/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard
. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>',
or link it with '-z noexecstack'.
15/11/26 19:46:58 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
[hadoop@acadgild ~]$
```

Step 15: For fetching data from Twitter, give the below command in the terminal.

flume-ng agent -n TwitterAgent -f <location of created/edited conf file>

```
hadoop@acadgild:~  
File Edit View Search Terminal Help  
[hadoop@acadgild ~]$ flume-ng agent -n TwitterAgent -f /home/hadoop/HADOOP/apache-flume-1.6.0-bin/conf/flume.conf
```

This will start fetching data from Twitter and send it to the HDFS.

```
hadoop@acadgild:~  
File Edit View Search Terminal Help  
} }} channels:{MemChannel=org.apache.flume.channel.MemoryChannel{name: MemChannel}} }  
15/11/26 20:48:57 INFO node.Application: Starting Channel MemChannel  
15/11/26 20:48:57 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: CHANNEL, name: MemChannel: Successfully registered new MBean.  
15/11/26 20:48:57 INFO instrumentation.MonitoredCounterGroup: Component type: CHANNEL, name: MemChannel started  
15/11/26 20:48:57 INFO node.Application: Starting Sink HDFS  
15/11/26 20:48:57 INFO node.Application: Starting Source Twitter  
15/11/26 20:48:57 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: SINK, name: HDFS: Successfully registered new MBean.  
15/11/26 20:48:57 INFO instrumentation.MonitoredCounterGroup: Component type: SINK, name: HDFS started  
15/11/26 20:48:57 INFO twitter4j.TwitterStreamImpl: Establishing connection.  
15/11/26 20:49:17 INFO twitter4j.TwitterStreamImpl: stream.twitter.com  
15/11/26 20:49:17 INFO twitter4j.TwitterStreamImpl: Waiting for 250 milliseconds  
15/11/26 20:49:17 INFO twitter4j.TwitterStreamImpl: Establishing connection.  
15/11/26 20:49:17 INFO twitter4j.TwitterStreamImpl: stream.twitter.com  
15/11/26 20:49:17 INFO twitter4j.TwitterStreamImpl: Waiting for 500 milliseconds  
15/11/26 20:49:18 INFO twitter4j.TwitterStreamImpl: Establishing connection.  
15/11/26 20:49:18 INFO twitter4j.TwitterStreamImpl: stream.twitter.com  
15/11/26 20:49:18 INFO twitter4j.TwitterStreamImpl: Waiting for 1000 milliseconds
```

To stop fetching data, press '*Ctrl+c*'. This will end the process of fetching the data.

Step 16: To check the contents of the Tweets folder, use the following command:

```
hadoop dfs -ls /user/flume/tweets
```

```
hadoop@acadgild:~  
File Edit View Search Terminal Help  
[hadoop@acadgild ~]$ hadoop fs -ls /user/flume/tweets  
Java HotSpot(TM) Client VM warning: You have loaded library /home/hadoop/HADOOP/  
hadoop-2.6.0/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard  
. The VM will try to fix the stack guard now.  
It's highly recommended that you fix the library with 'execstack -c <libfile>',  
or link it with '-z noexecstack'.  
15/12/24 12:45:25 WARN util.NativeCodeLoader: Unable to load native-hadoop libra  
ry for your platform... using builtin-java classes where applicable  
Found 1 items  
-rw-r--r-- 1 hadoop supergroup 316294 2015-12-24 12:39 /user/flume/tweets/  
FlumeData.1450940925854  
[hadoop@acadgild ~]$
```

Step 17: To see the data inside this file, type the following command:
hadoop dfs -cat /user/flume/tweets/<flumeData file name>

```
hadoop@acadgild:~  
File Edit View Search Terminal Help  
[hadoop@acadgild ~]$ hadoop dfs -cat /user/flume/tweets/FlumeData.1450940925854  
DEPRECATED: Use of this script to execute hdfs command is deprecated.  
Instead use the hdfs command for it.  
  
Java HotSpot(TM) Client VM warning: You have loaded library /home/hadoop/HADOOP/  
hadoop-2.6.0/lib/native/libhadoop.so.1.0.0 which might have disabled stack guard
```

```
hadoop@acadgild:~  
File Edit View Search Terminal Help  
... https://t.co/rKcFWFoUcc #education", "contributors": null, "geo": null, "entities  
": { "symbols": [], "urls": [ { "expanded_url": "http://bit.ly/1QJreHR", "indices": [ 103, 1  
26 ], "display_url": "bit.ly/1QJreHR", "url": "https://t.co/rKcFWFoUcc" } ], "hashtags":  
[ { "text": "education", "indices": [ 127, 137 ] } ], "user_mentions": [], "is_quote_status"  
: false, "source": "<a href=\"http://twitterfeed.com\" rel=\"nofollow\">twitterfeed  
</a>", "favorited": false, "in_reply_to_user_id": null, "retweet_count": 0, "id_str": "  
679921891398660096", "user": { "location": "Coventry", "default_profile": false, "profile  
background_tile": false, "statuses_count": 33000, "lang": "en", "profile_link_color"  
: "2FC2EF", "profile_banner_url": "https://pbs.twimg.com/profile_banners/254678901  
4/1441620563", "id": 2546789014, "following": null, "protected": false, "favourites_cou  
nt": 508, "profile_text_color": "666666", "verified": false, "description": "Amateur tv  
nerd . Working", "contributors_enabled": false, "profile_sidebar_border_color": "18  
1A1E", "name": "Angela Dubravski", "profile_background_color": "1A1B1F", "created_at"  
: "Wed May 14 06:28:35 +0000 2014", "default_profile_image": false, "followers_count  
": 729, "profile_image_url_https": "https://pbs.twimg.com/profile_images/6595025668  
54627328/sfQgy6LM_normal.jpg", "geo_enabled": false, "profile_background_image_url"  
: "http://abs.twimg.com/images/themes/theme9/bg.gif", "profile_background_image_ur  
l_https": "https://abs.twimg.com/images/themes/theme9/bg.gif", "follow_request_sen  
t": null, "url": null, "utc_offset": null, "time_zone": null, "notifications": null, "prof  
ile_use_background_image": true, "friends_count": 1192, "profile_sidebar_fill_color"  
: "252429", "screen_name": "citaCrowl", "id_str": "2546789014", "profile_image_url": "h  
ttp://pbs.twimg.com/profile_images/659502566854627328/sfQgy6LM_normal.jpg", "list  
ed_count": 91, "is_translator": false } }  
[hadoop@acadgild ~]$
```

We have completed the action of fetching live-streaming data from Twitter and loaded it to the HDFS, using Flume after streaming data from twitter, we need to analysis of this data and give polarity accordingly.