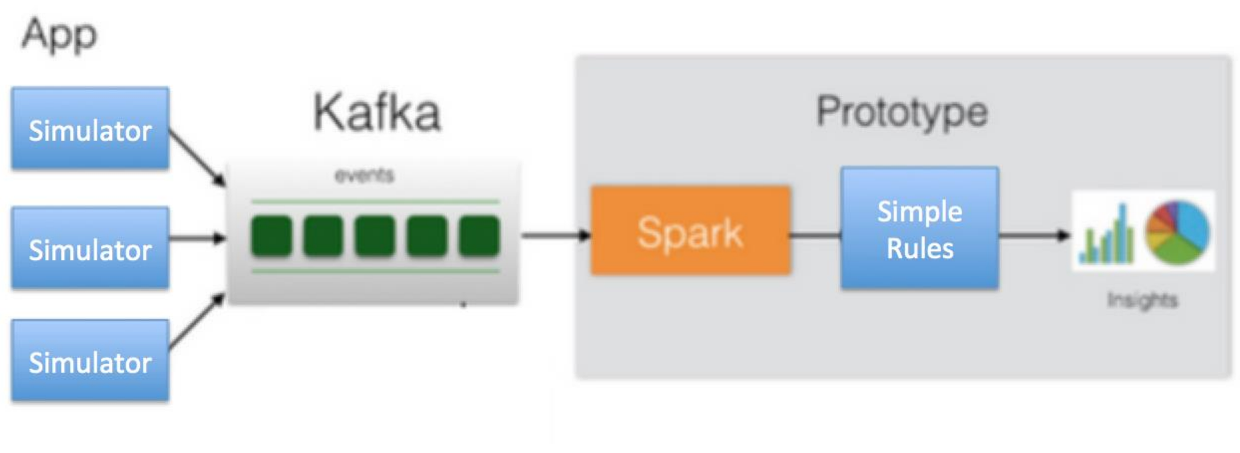
CMPE 239 – Moonshot Assignment

Name: Shounak Gujarathi

Project Title: Twitter stream analysis and visualization using Apache Kafka, Spark and Java



As given in the requirements, this assignment is made up of 3 parts and the initial Setup.

Part 0: Initial Setup [Setting up Java, Zookeeper, Kafka and Spark]

Part I: Simulating Streaming data, or getting actual streams and sending ti via Kafka Producer.

Part II: Consuming kafka producer stream via Kafka Consumer and converting to Spark Dstream.

Part III: Analysis on the incoming data via Kafka Producer using Apache Spark and Generating graphs.

I'll be explaining each step in detail in this report.

Part 0: Initial Setup [Setting up Java, Zookeeper, Kafka and Spark]

I’ve implemented this assignment on Windows 64 bit and will be detailing the steps required to setup Zookeeper, Kafka and Spark.

Step 1:

i. Download and install Java SE Runtime Environment 8 from http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html. I downloaded the Windows x64 version.

ii. Download & extract Zookeeper from http://zookeeper.apache.org/releases.html

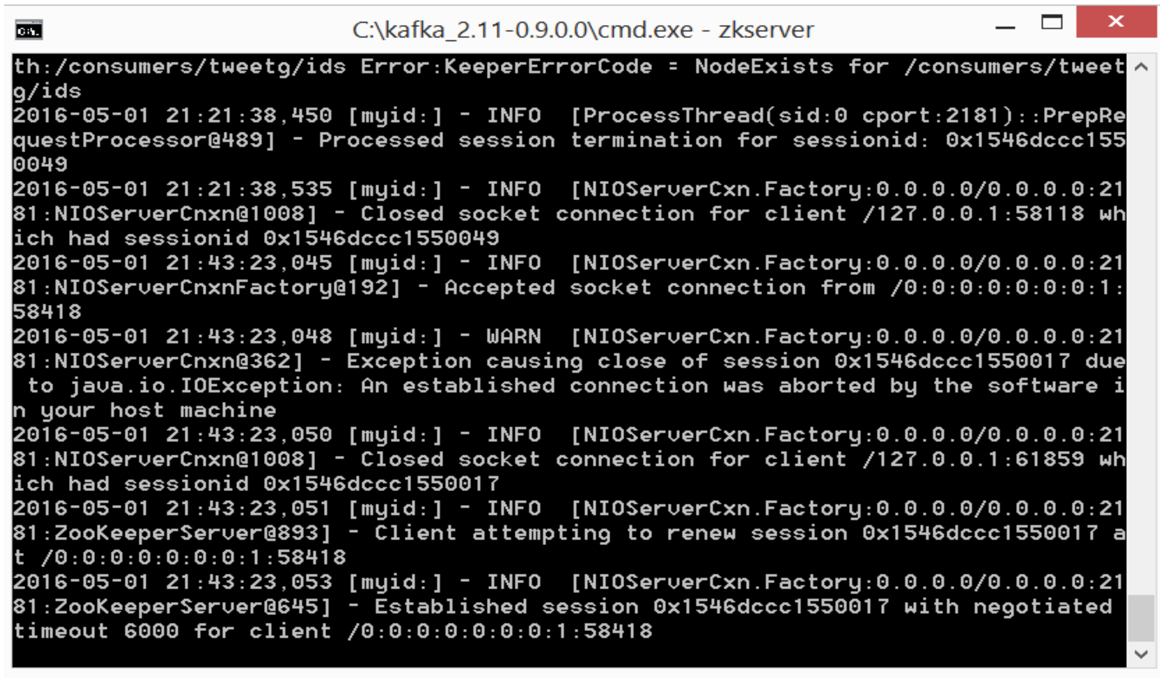
iii. Download & extract Kafka from http://kafka.apache.org/downloads.html

Step 2: Starting Zookeeper Server

i. Add system environment variables for Zookeeper as follows:
   a. ZOOKEEPER_HOME = *C:\zookeeper-3.4.7*
   b. Append to PATH varaible: %ZOOKEEPER_HOME%\bin;

ii. Go to zookeeper-3.4.8\conf directory. In the zoo.cfg (zoo_sample.cfg in some cases, where it must be renamed) replace *dataDir=/tmp/zookeeper* with a desired folder path. We change this path as we need to persist zookerper data. 2181 is the default port number that can be changed in this file.

iii. Run zookeeper by typing zkserver in cmd. You should see a window similar to the one given below.



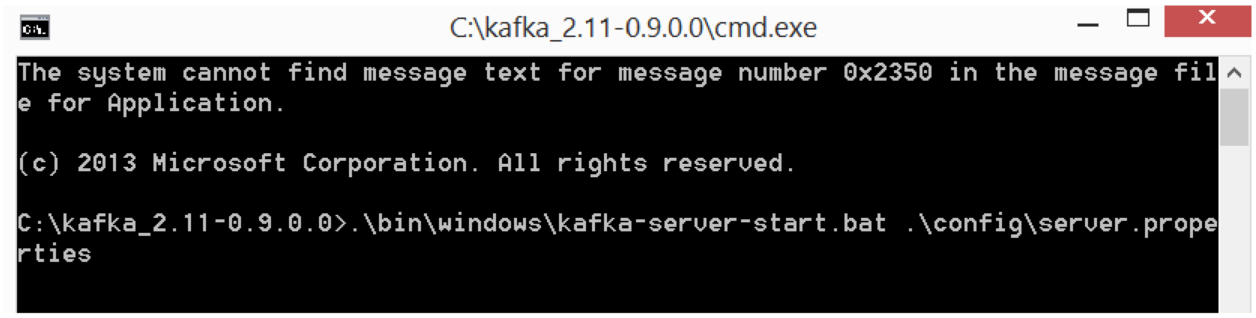
```
th:/consumers/tweetg/ids Error:KeeperErrorCode = NodeExists for /consumers/tweet
g/ids
2016-05-01 21:21:38,450 [myid:] - INFO  [ProcessThread(sid:0 cport:2181)::PrepRe
questProcessor@489] - Processed session termination for sessionid: 0x1546dccc155
0049
2016-05-01 21:21:38,535 [myid:] - INFO  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxn@1008] - Closed socket connection for client /127.0.0.1:58118 wh
ich had sessionid 0x1546dccc1550049
2016-05-01 21:43:23,045 [myid:] - INFO  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxnFactory@192] - Accepted socket connection from /0:0:0:0:0:0:0:1:
58418
2016-05-01 21:43:23,048 [myid:] - WARN  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxn@362] - Exception causing close of session 0x1546dccc1550017 due
 to java.io.IOException: An established connection was aborted by the software i
n your host machine
2016-05-01 21:43:23,050 [myid:] - INFO  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:21
81:NIOServerCnxn@1008] - Closed socket connection for client /127.0.0.1:61859 wh
ich had sessionid 0x1546dccc1550017
2016-05-01 21:43:23,051 [myid:] - INFO  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:21
81:ZooKeeperServer@893] - Client attempting to renew session 0x1546dccc1550017 a
t /0:0:0:0:0:0:0:1:58418
2016-05-01 21:43:23,053 [myid:] - INFO  [NIOServerCxn.Factory:0.0.0.0/0.0.0.0:21
81:ZooKeeperServer@645] - Established session 0x1546dccc1550017 with negotiated
timeout 6000 for client /0:0:0:0:0:0:0:1:58418
```

Step 3: Starting Kafka Server

Once the zkserver starts, we can proceed with Kafka server.

i. Extract the kafka folder and goto it's root.

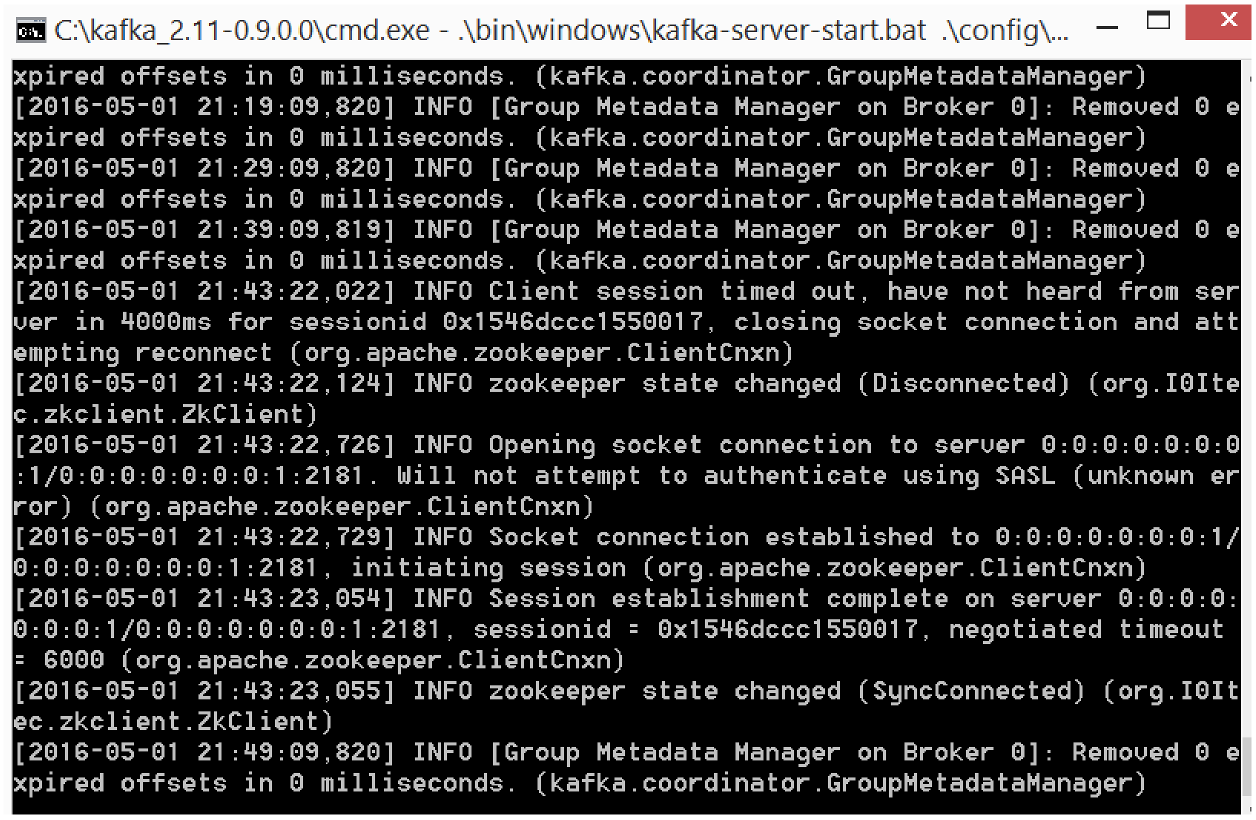
ii. You need to run .\bin\windows\kafka-server-start.bat with .\config\server.properties. It can be done as shown below.



C:\kafka_2.11-0.9.0.0\cmd.exe

```
The system cannot find message text for message number 0x2350 in the message fil
e for Application.

(c) 2013 Microsoft Corporation. All rights reserved.

C:\kafka_2.11-0.9.0.0>.\bin\windows\kafka-server-start.bat .\config\server.prope
rties
```

iii. Once run successfully the kafka server will look like this



C:\kafka_2.11-0.9.0.0\cmd.exe - .\bin\windows\kafka-server-start.bat .\config\...

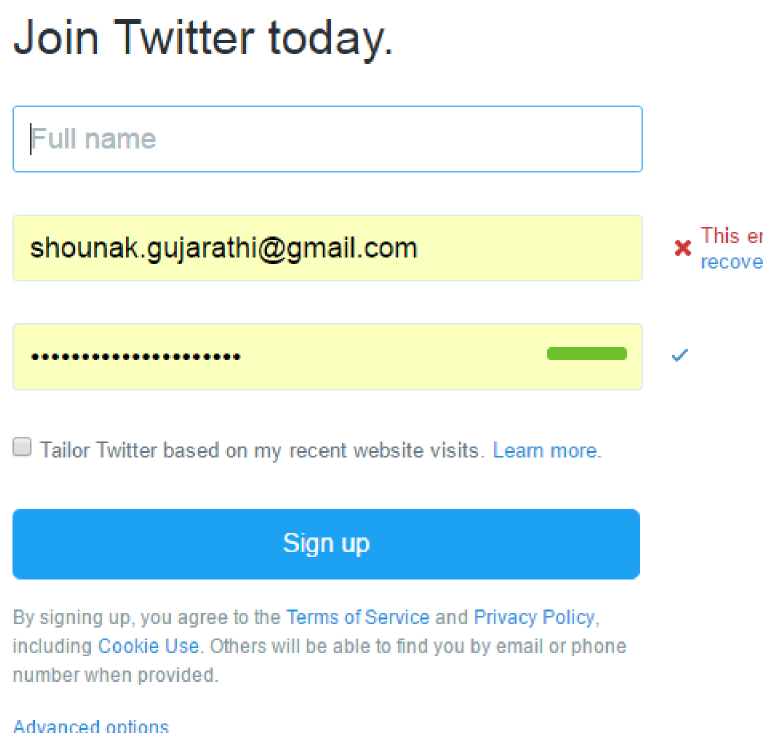
```
xpired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
[2016-05-01 21:19:09,820] INFO [Group Metadata Manager on Broker 0]: Removed 0 e
xpired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
[2016-05-01 21:29:09,820] INFO [Group Metadata Manager on Broker 0]: Removed 0 e
xpired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
[2016-05-01 21:39:09,819] INFO [Group Metadata Manager on Broker 0]: Removed 0 e
xpired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
[2016-05-01 21:43:22,022] INFO Client session timed out, have not heard from ser
ver in 4000ms for sessionid 0x1546dccc1550017, closing socket connection and att
empting reconnect (org.apache.zookeeper.ClientCnxn)
[2016-05-01 21:43:22,124] INFO zookeeper state changed (Disconnected) (org.I0Ite
c.zkclient.ZkClient)
[2016-05-01 21:43:22,726] INFO Opening socket connection to server 0:0:0:0:0:0:0
:1/0:0:0:0:0:0:0:1:2181. Will not attempt to authenticate using SASL (unknown er
ror) (org.apache.zookeeper.ClientCnxn)
[2016-05-01 21:43:22,729] INFO Socket connection established to 0:0:0:0:0:0:0:1/
0:0:0:0:0:0:0:1:2181, initiating session (org.apache.zookeeper.ClientCnxn)
[2016-05-01 21:43:23,054] INFO Session establishment complete on server 0:0:0:0:
0:0:0:1/0:0:0:0:0:0:0:1:2181, sessionid = 0x1546dccc1550017, negotiated timeout
= 6000 (org.apache.zookeeper.ClientCnxn)
[2016-05-01 21:43:23,055] INFO zookeeper state changed (SyncConnected) (org.I0It
ec.zkclient.ZkClient)
[2016-05-01 21:49:09,820] INFO [Group Metadata Manager on Broker 0]: Removed 0 e
xpired offsets in 0 milliseconds. (kafka.coordinator.GroupMetadataManager)
```

iv. These are all the steps required for starting kafka server.

Part I: Simulating Streaming data, or getting actual streams and sending them via Kafka Producer.

What I'm doing is getting live streaming data from twitter. Following are the steps needed to get Twitter streaming data.

Step 1: Set up apps.twitter account https://twitter.com/signup



Join Twitter today.

Full name

shounak.gujarathi@gmail.com

This er
recove

••••••••••••••••••••

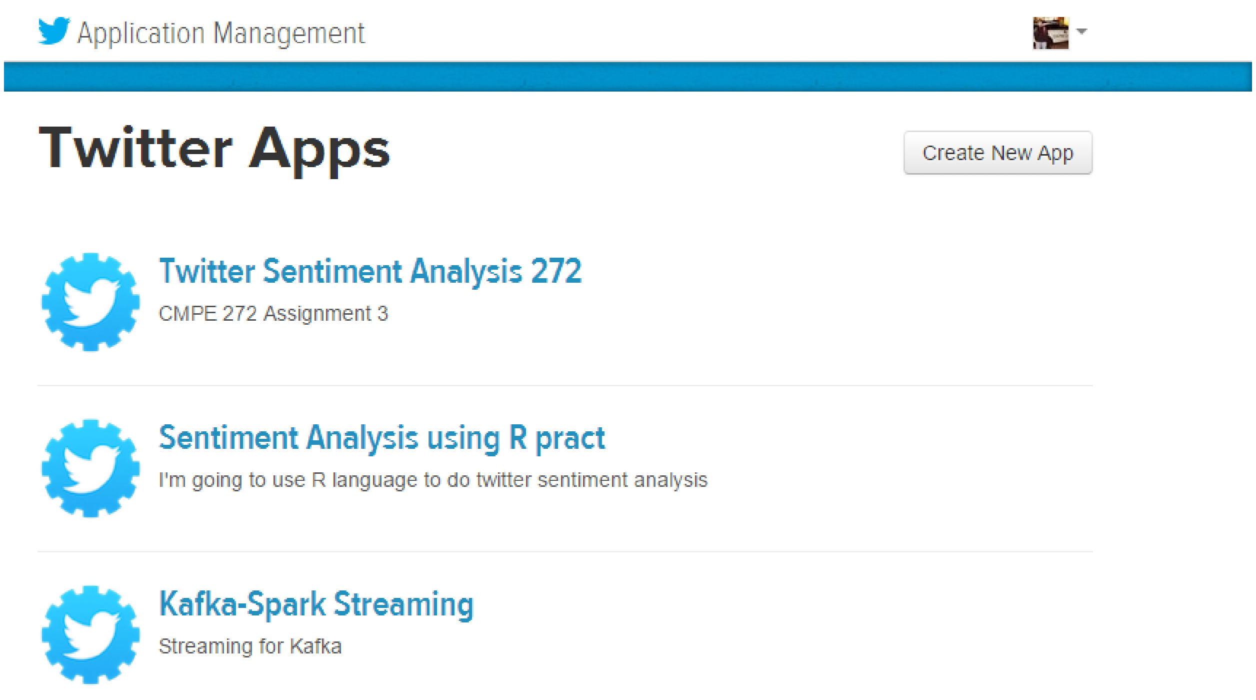
Tailor Twitter based on my recent website visits. Learn more.

Sign up

By signing up, you agree to the Terms of Service and Privacy Policy, including Cookie Use. Others will be able to find you by email or phone number when provided.

Advanced options

Step 2: Login to apps.twitter account at https://apps.twitter.com/ and click on create a new app


Application Management

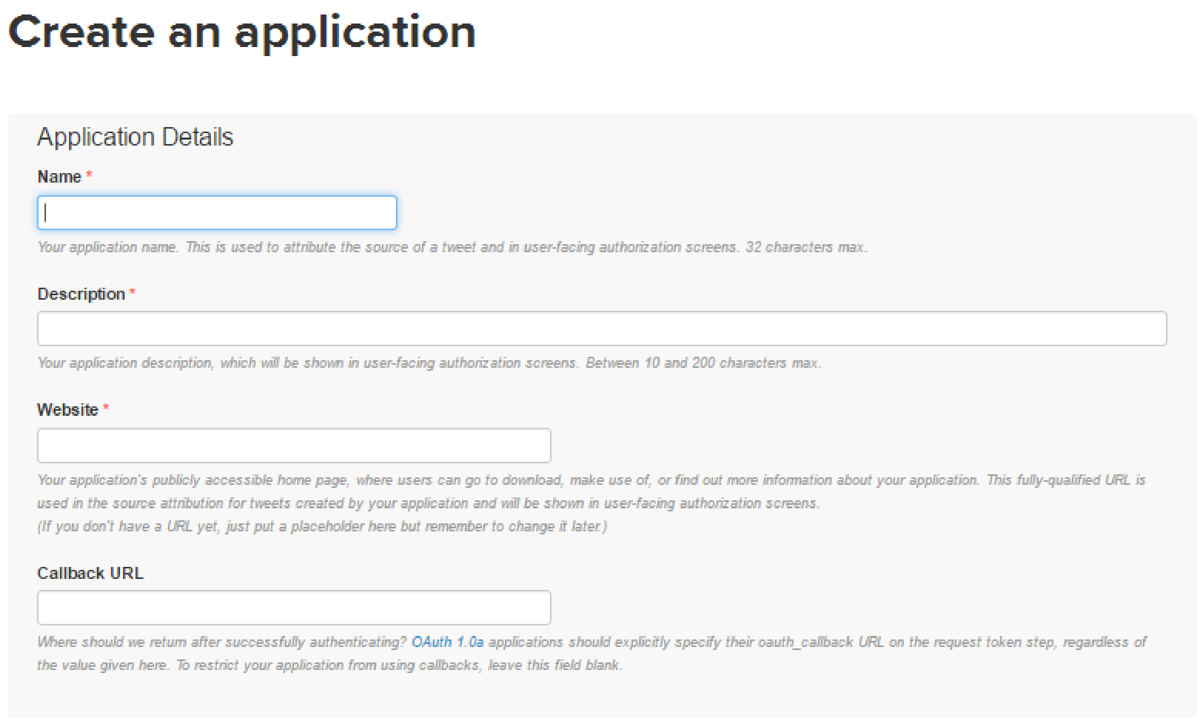
Twitter Apps

Create New App

Twitter Sentiment Analysis 272
CMPE 272 Assignment 3

Sentiment Analysis using R pract
I'm going to use R language to do twitter sentiment analysis

Kafka-Spark Streaming
Streaming for Kafka


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.

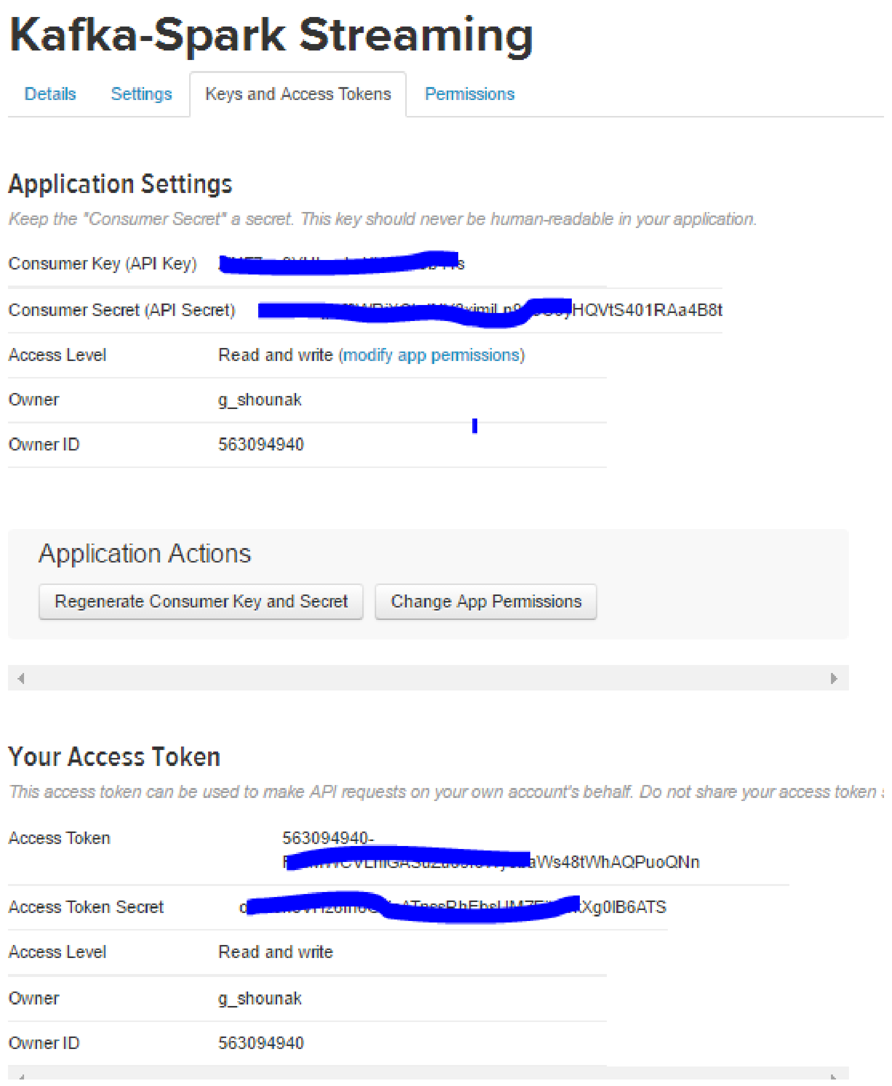
Website *

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 successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Fill in the required details and your application will be created. My app name is Kafka-Spark Streaming. Now in order to call twitter api, we will need the Access_Key and Token given here: https://apps.twitter.com/app/12299898/keys



Kafka-Spark Streaming

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	[redacted]
Consumer Secret (API Secret)	[redacted]HQVtS401RAa4B8t
Access Level	Read and write (modify app permissions)
Owner	g_shounak
Owner ID	563094940

Application Actions

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token

Access Token	563094940-[redacted]aWs48tWhAQPuoQNn
Access Token Secret	[redacted]Xg0lB6ATS
Access Level	Read and write
Owner	g_shounak
Owner ID	563094940

These keys will be used in the Java Source Code. For accepting incoming stream, 2 Maven dependencies are required in the project.

```
<!-- Apache Kafka dependency -->
<dependency>
  <groupId>org.apache.kafka</groupId>
  <artifactId>kafka_2.10</artifactId>
  <version>0.9.0.1</version>
</dependency>
```

AND

```xml
<dependency>
  <groupId>com.twitter</groupId>
  <artifactId>hbc-core</artifactId> <!-- or hbc-twitter4j -->
  <version>2.2.0</version> <!-- or whatever the latest version is -->
</dependency>
```

The source code for the Kafka producer sending twitter stream can be found in the source I provided in the class TwitterKafkaProducer. What this class does is it accepts twitter streaming data from twitter using OAuth authentication and then sends this stream on the kafka channel named “tweet”.

I’ve printed the output of this stream to get an idea of how tweet looks. Here, the “text” filed contains the actual tweet and there is a lot of other information that can be used.

```
{
   "created_at":"Mon May 02 04:22:06 +0000 2016",
   "id":726990122416136192,
   "id_str":"726990122416136192",
   "text":"Unade mis canciones favoritas @DiegoAGarmendia @GermanGarmendia Me encanto\nAncud -
 Cambia (lyric video) https:\/\/t.co\/yp3vPHuvT4 v\u00eda @YouTube",
   "source":"\u003ca href=\"http:\/\/twitter.com\" rel=\"nofollow\"\u003eTwitter Web Client\u003c\/a\u003e",
   "truncated":false,
   "in_reply_to_status_id":null,
   "in_reply_to_status_id_str":null,
   "in_reply_to_user_id":null,
   "in_reply_to_user_id_str":null,
   "in_reply_to_screen_name":null,
   "user":{
      "id":2202260851,
      "id_str":"2202260851",
      "name":"Brenda Tomlinson",
      "screen_name":"Brenda151369",
      "location":null,
.

.

.

   "timestamp_ms":"1462162926664"
}
```

I’ve limited the number of tweets to 1000 for this assignment.

Part II: Consuming kafka producer stream via Kafka Consumer and converting to Spark Dstream.

I’ve setup Kafka Consumer and Spark in the same class. The dependencies required for both are:

```xml
<!-- Basic Apache Spark dependency -->
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-core_2.10</artifactId>
  <version>1.6.1</version>
</dependency>

<!-- Apache Spark Streaming dependency -->
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-streaming_2.10</artifactId>
  <version>1.6.1</version>
</dependency>

<!-- Apache Spark & Kafka Integration dependency -->
<dependency>
  <groupId>org.apache.spark</groupId>
  <artifactId>spark-streaming-kafka_2.10</artifactId>
  <version>1.1.0</version>
</dependency>
```

To run spark a SparkConf class instance needs to be created.

```
SparkConf sparkConf = new
SparkConf().setAppName(sparkAppName).setMaster("local[4]");
JavaStreamingContext jsc = new JavaStreamingContext(sparkConf, new
Duration(3000));
```

To accept stream from kafka kafkaUtils.createStream is used

```
JavaPairReceiverInputDStream<String, String> messages =
        KafkaUtils.createStream(jsc, "localhost:2181", "tweetg", topicMap);
```

This code basically converts a Kafka Stream to Spark Dstream.

Detailed code is given in the KafkaStreaming.java directory.

Part III: Analysis on the incoming data via Kafka Producer using Apache Spark and Generating graphs.

What I'm doing is performing a word count on the text element of each incoming tweet.

```
{
   "created_at":"Mon May 02 04:22:06 +0000 2016",
   "id":726990122416136192,
   "id_str":"726990122416136192",
   "text":"Unade mis canciones favoritas @DiegoAGarmendia @GermanGarmendia Me encanto\nAncud -
 Cambia (lyric video) https:\/\/t.co\/yp3vPHuvT4 v\u00eda @YouTube",
   "source":"\u003ca href=\"http:\/\/twitter.com\" rel=\"nofollow\"\u003eTwitter Web Client\u003c\/a\u003e",
   "truncated":false,
   "in_reply_to_status_id":null,
   "in_reply_to_status_id_str":null,
   "in_reply_to_user_id":null,
   "in_reply_to_user_id_str":null,
   "in_reply_to_screen_name":null,
   "user":{
      "id":2202260851,
      "id_str":"2202260851",
      "name":"Brenda Tomlinson",
      "screen_name":"Brenda151369",
      "location":null,
.

.

   "timestamp_ms":"1462162926664"
}
```

So my word count program will go through all 1000 tweets and count the words in the “text” element of the json object one at a time and reduce it to give aggregated results for all 1000 streams.

```
JavaDStream<String> line = messages.map(new StreamMessages());
JavaDStream<String> words = line.flatMap(new splitWords());
JavaPairDStream<String, Integer> wordCount = words.mapToPair(new
wordMapper()).reduceByKey(new WordCountReducer());
```

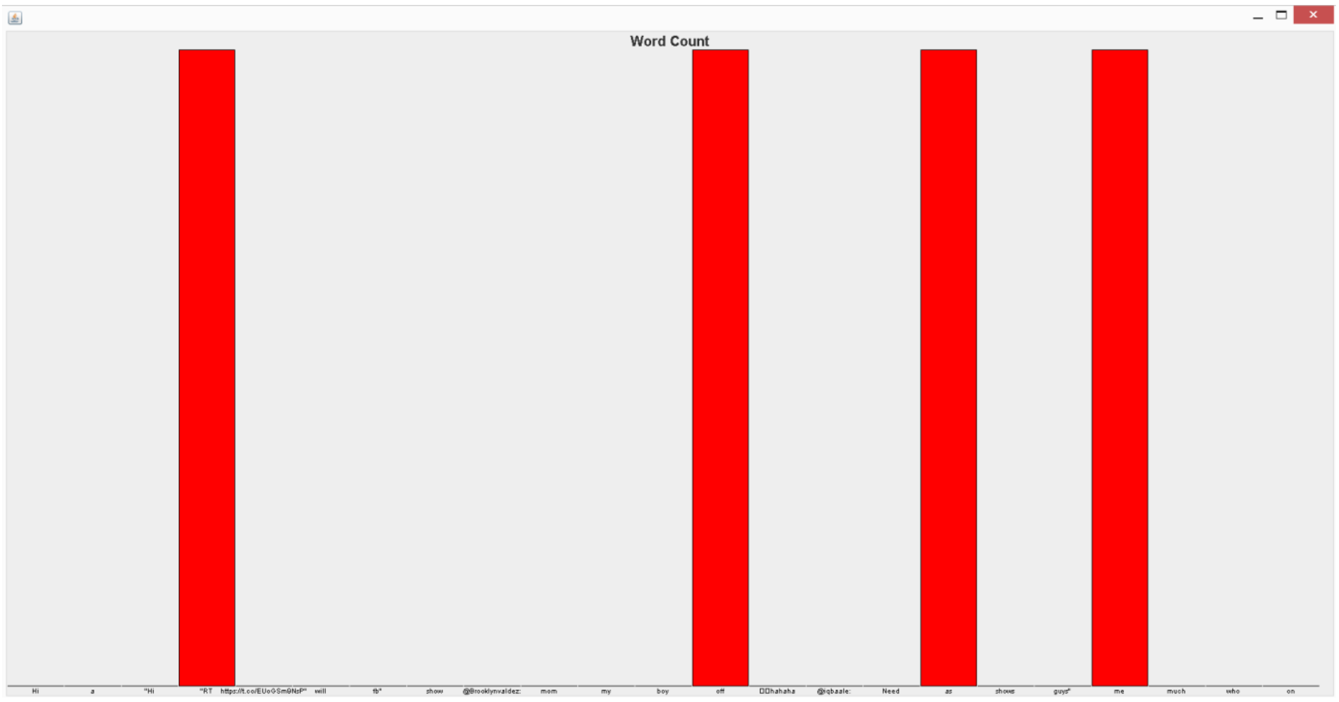
That’s all you need to perform a word count in Spark. Spark has a rich library of inbuilt functions like map, reduce, mapToPair that provide quick in-memory processing for real time results.

The console output looks like this:

```
(Premier,1)

(paper,1)

(Seattle,1)

(House,1)

(opening,2)

(RT http://t.co/EU.GSm9NsF”,14)

(hatapendezwa,1)

(otoño,1)

(off,14)

(MAINE,1)

…
```

The … signifies that there are more words but aren’t been shown. This can be verified from the graph generated in the next page

I've also visualized the results using Java appeletes.



The bar graphs shows that words RT, off, as and me were repeated a lot more than the others.

Future Scope: While doing this project I made attempts to save this tweet data in cassandra and then readinf from it using golang and visualizting the data. Due to time contrainst I wasn't able to see it through so I'm submitting visualiztion only based on Java Appelets. However, I will work on this in the future to make a better visualization experience.

Conslusion: Thus I've successfully read srtreaming data(Twitter) via Apache Kafka, performed analysis on this data using Apache Spark and visualized this information.