**COMS 6998 - Cloud Computing and Big Data**

During the term the class will present three assignments. In these assignments you will work in pairs to build a complete Cloud based application. This means your homeworks are cummulative, and you will be building on top at every stage.

Initially you will build a web application on the AWS cloud stack, and you will notice that your application will get a de facto load balancing capability provided by the AWS platform. After this stage you application will live in an Elastic Beanstalk (EB) container.

The next step will be to distribute that applications so that several EB-EC2 instances may collaborate to achieve results. For this you will be using distributed messaging to decouple your application.

Finally you will add some intelligence to your application. Your distributed application may be generating data from usage, and consuming data from different APIs. In this stage you will use BigData tools like, Hadoop, HBase, PIG, etc. to gain insights on the data that's traversing your application.

We will use the Twitter Streaming API throughout the different homeworks. The first step is to get API keys that you can obtain [here](https://dev.twitter.com/).

**First Assignment**

**Assignment Objectives:**

* Use the [Elastic Beanstalk API](http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/elasticbeanstalk/AWSElasticBeanstalkClient.html) to create, configure, and deploy an application instance programmatically.
* Use the [Elastic LoadBalancing API](http://docs.aws.amazon.com/AWSJavaSDK/latest/javadoc/com/amazonaws/services/elasticloadbalancing/AmazonElasticLoadBalancingClient.html) to configure load balancing on Elastic Beanstalk created.

For this assignment you will develop an application that:

* Reads a stream of tweets from the Twitter Live API (Code provided)
* Records the tweet ID, time, and other relevant elements into a DB (SQL or NoSQL)
* Presents the Tweet in a map that is being updated in Near Real Time (Consider evaluating WebSockets, or Server Side Events for your implementation)
* The map clusters tweets as to show where is people tweeting the most, according to the sample tweets you get from the streaming API.

**Second Assignment**

**Assignment Objectives:**

* Use the [Amazon SQS](http://aws.amazon.com/sqs/) service to create a processing queue for the Tweets that are delivered by the Twitter Streaming API
* Use [Amazon SNS](http://aws.amazon.com/sns/) service to update the status processing on each tweet so the UI can refresh.
* Integrate a third party cloud service API into the Tweet processing flow.

For this assignment you will develop an application that:

* Reads a stream of tweets from the Twitter Live API (Code provided). **Note:** you might follow a specific topic on the API or get the complete stream
* Records the tweet ID, time, and other relevant elements into a DB (SQL or NoSQL) **(2)**
* After the tweet is recorded in the DB send a message to the Queue for Asynchronous processing on the text of the tweet **(3)**
* Presents the Tweet in a map that is being updated in Near Real Time (Consider evaluating WebSockets, or Server Side Events for your implementation)
* The map clusters tweets as to show where is people tweeting the most, according to the sample tweets you get from the streaming API.
* Define a worker pool that will pick up messages from the queue to process. These workers should each run on a separate pool thread.
* Make a call to the sentiment API off your preference (e.g. [Alchemy](http://www.alchemyapi.com/api/sentiment-analysis/)). This can return a positive or negative sentiment evaluation for the text of the submitted Tweet. **(4)**
* As soon as the tweet is processed send a notification -using SNS- to an HTTP endpoint that will update the UI with the new information about the Tweet. **(5)**
* Using this information your application should display the Tweet clusters and the overall sentiment. **(6)**

**Architecture Diagram:**

Below you will find an Architecture diagram that shows the main differences between Assignment 1 and Assignment 2. We will use this diagram to explain the modifications that you will introduce into your application (in bold on the steps above).

