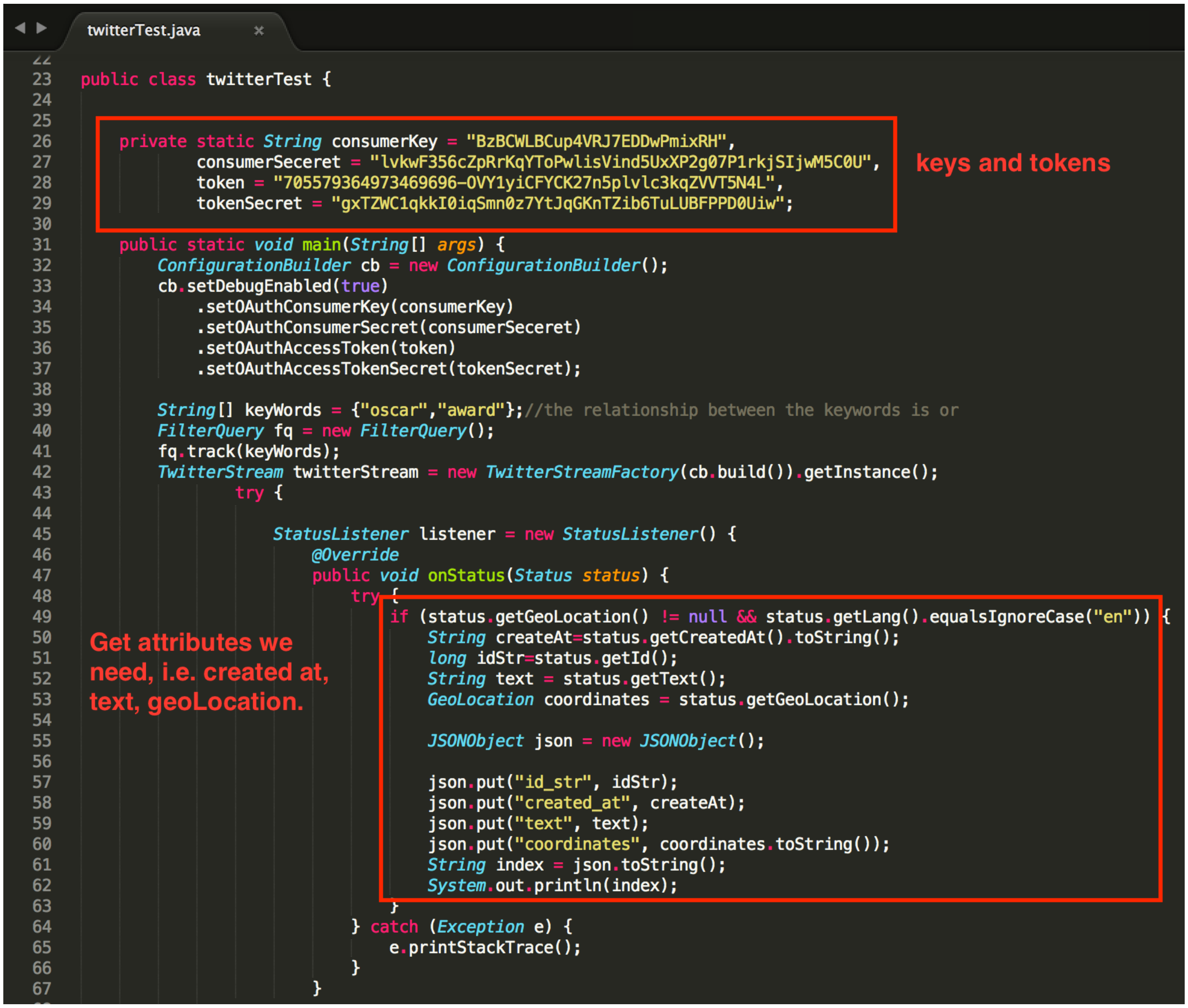
TwittMap Report

Team members: Xiaolong Jiang (N18261025), Guoqing Xiong (N17801169)

1. Twitter API

In the Twitter Streaming API part, we applied a account to configure Key and Secret. We used twitter4j package and call TwitterStreamFactory class to get instance of twitterStream.

Then we call the listener in twitterStream. We used Filter Query to search word and override onStatus method to get elements such ad id,coordinates in JSON file. Then we upload these JSON file to cloudSearch platform.



```java
public class twitterTest {

    private static String consumerKey = "BzBCWLBCup4VRJ7EDDwPmixRH",
            consumerSeceret = "lvkwF356cZpRrKqYToPwlisVind5UxXP2g07P1rkjSIjwM5C0U",
            token = "705579364973469696-0VY1yiCFYCK27n5plvlc3kqZVVT5N4L",
            tokenSecret = "gxTZWC1qkkI0iqSmn0z7YtJqGKnTZib6TuLUBFPPD0Uiw";

    public static void main(String[] args) {
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true)
            .setOAuthConsumerKey(consumerKey)
            .setOAuthConsumerSecret(consumerSeceret)
            .setOAuthAccessToken(token)
            .setOAuthAccessTokenSecret(tokenSecret);

        String[] keyWords = {"oscar","award"};//the relationship between the keywords is or
        FilterQuery fq = new FilterQuery();
        fq.track(keyWords);
        TwitterStream twitterStream = new TwitterStreamFactory(cb.build()).getInstance();
                try {

                    StatusListener listener = new StatusListener() {
                        @Override
                        public void onStatus(Status status) {
                            try {
                                if (status.getGeoLocation() != null && status.getLang().equalsIgnoreCase("en")) {
                                    String createAt=status.getCreatedAt().toString();
                                    long idStr=status.getId();
                                    String text = status.getText();
                                    GeoLocation coordinates = status.getGeoLocation();

                                    JSONObject json = new JSONObject();

                                    json.put("id_str", idStr);
                                    json.put("created_at", createAt);
                                    json.put("text", text);
                                    json.put("coordinates", coordinates.toString());
                                    String index = json.toString();
                                    System.out.println(index);
                                }
                            } catch (Exception e) {
                                e.printStackTrace();
                            }
                        }
```

keys and tokens

Get attributes we need, i.e. created at, text, geoLocation.

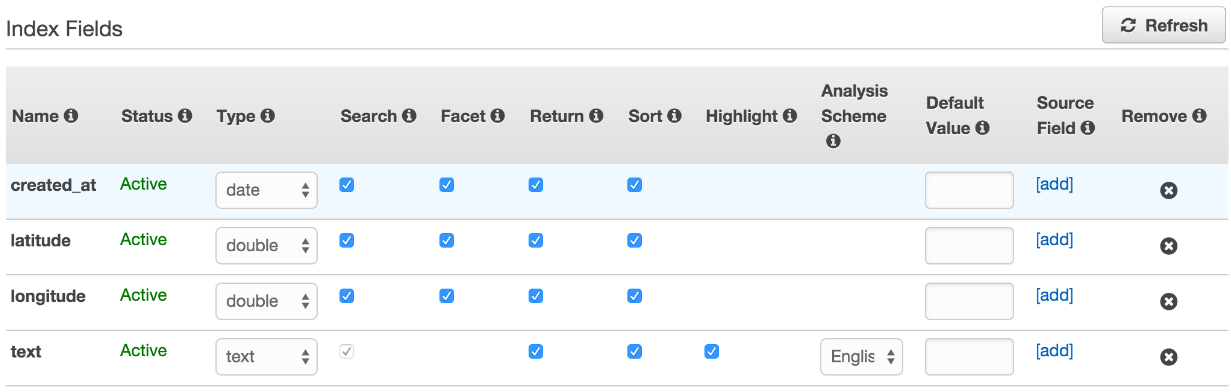
2. Deploy AWS CloudSearch

Amazon CloudSearch is a managed service in the AWS Cloud that makes it simple and cost-effective to set up, manage, and scale a search solution for your website or application.

2.1 Index Construction

To search files/informations in structured or semi-structured data, we have to build index of files first. In each tweet, the indices we need for now are created_at, latitude, longitude, and text. Besides, tweed ID is needed to uniquely identify each tweet file in CloudSearch.

Figure below is the index fields of CloudSearch.



Index Fields

Name	Status	Type	Search	Facet	Return	Sort	Highlight	Analysis Scheme	Default Value	Source Field	Remove
created_at	Active	date	☑	☑	☑	☑				[add]	⊗
latitude	Active	double	☑	☑	☑	☑				[add]	⊗
longitude	Active	double	☑	☑	☑	☑				[add]	⊗
text	Active	text	✓		☑	☑	☑	Englis		[add]	⊗

2.2 Upload File

Because the file standard formats of Twitter API and AWS CloudSearch don't match. We need a Tweet Format Transfer program, which will be mention at Chapter 3.

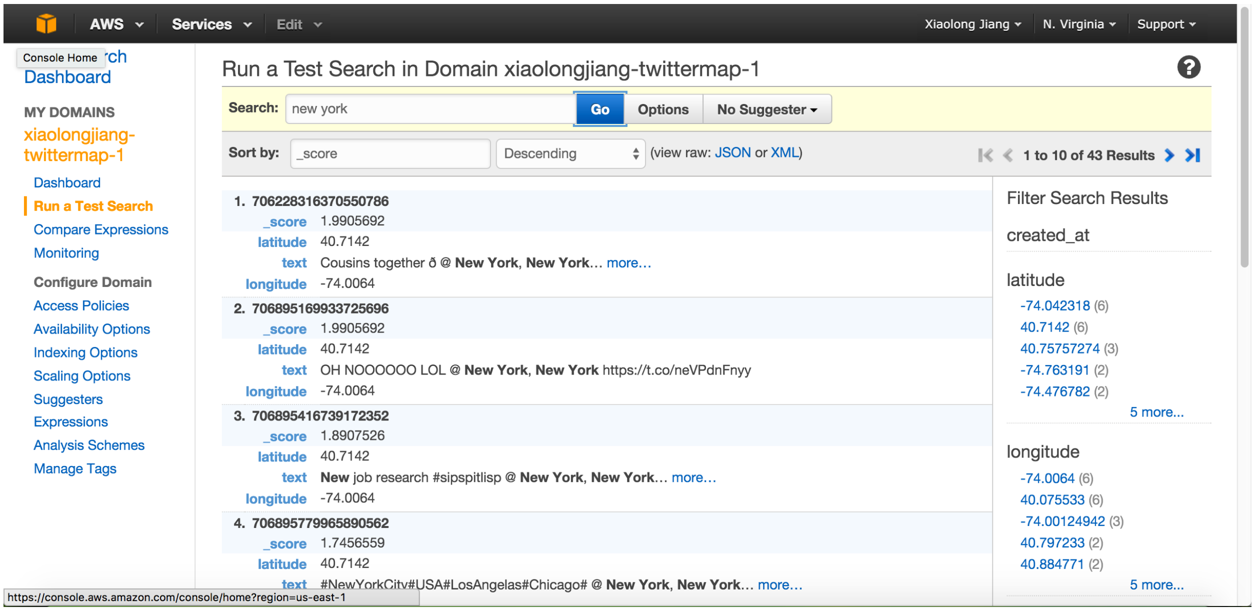
2.3 Search

After upload file onto AWS CloudSearch, we can search through AWS CloudSearch Console or HTTP Request.

2.3.1 Search through AWS CloudSearch Console

Click on 'Run a Test Search' in the navigation bar, and type in search query, then click 'GO' button. The relevant file will be returned by CloudSearch.

Figure below is searching 'new york' through CS console.



Run a Test Search in Domain xiaolongjiang-twittermap-1

Search: new york

Sort by: _score Descending (view raw: JSON or XML)

1 to 10 of 43 Results

1. 706228316370550786
_score 1.9905692
latitude 40.7142
text Cousins together ð @ **New York**, **New York**... more...
longitude -74.0064

2. 706895169933725696
_score 1.9905692
latitude 40.7142
text OH NOOOOOO LOL @ **New York**, **New York** https://t.co/neVPdnFnyy
longitude -74.0064

3. 706895416739172352
_score 1.8907526
latitude 40.7142
text **New** job research #sipspitlisp @ **New York**, **New York**... more...
longitude -74.0064

4. 706895779965890562
_score 1.7456559
latitude 40.7142
text #NewYorkCity#USA#LosAngelas#Chicago# @ **New York**, **New York**... more...

Filter Search Results

created_at

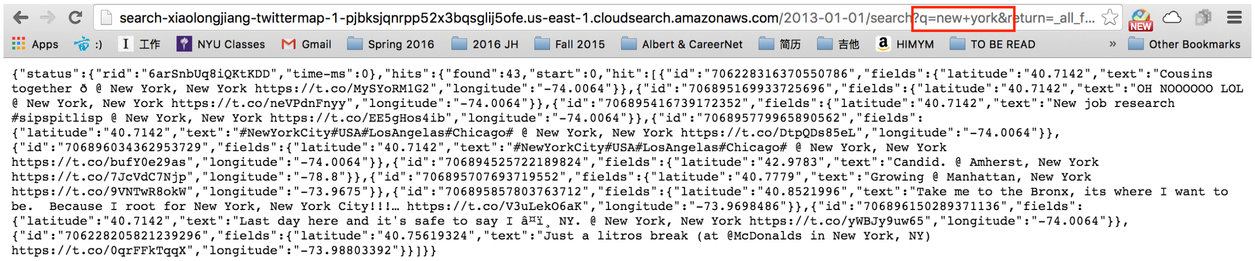
latitude
-74.042318 (6)
40.7142 (6)
40.75757274 (3)
-74.763191 (2)
-74.476782 (2)
5 more...

longitude
-74.0064 (6)
40.075533 (6)
-74.00124942 (3)
40.797233 (2)
40.884771 (2)
5 more...

2.3.2 Search through HTTP Request

Using <"http://" + Search Endpoint + "search?q=" + query> to get response from CloudSearch. It supports multiple words query with "+" between words instead of whitespace.

Figure below is searching "new york" using http request.


search-xiaolongjiang-twittermap-1-pjbksjqnrpp52x3bqsglij5ofe.us-east-1.cloudsearch.amazonaws.com/2013-01-01/search?q=new+york&return=_all_f...
Apps
:)
工作
NYU Classes
Gmail
Spring 2016
2016 JH
Fall 2015
Albert & CareerNet
简历
吉他
HIMYM
TO BE READ
Other Bookmarks
NEW

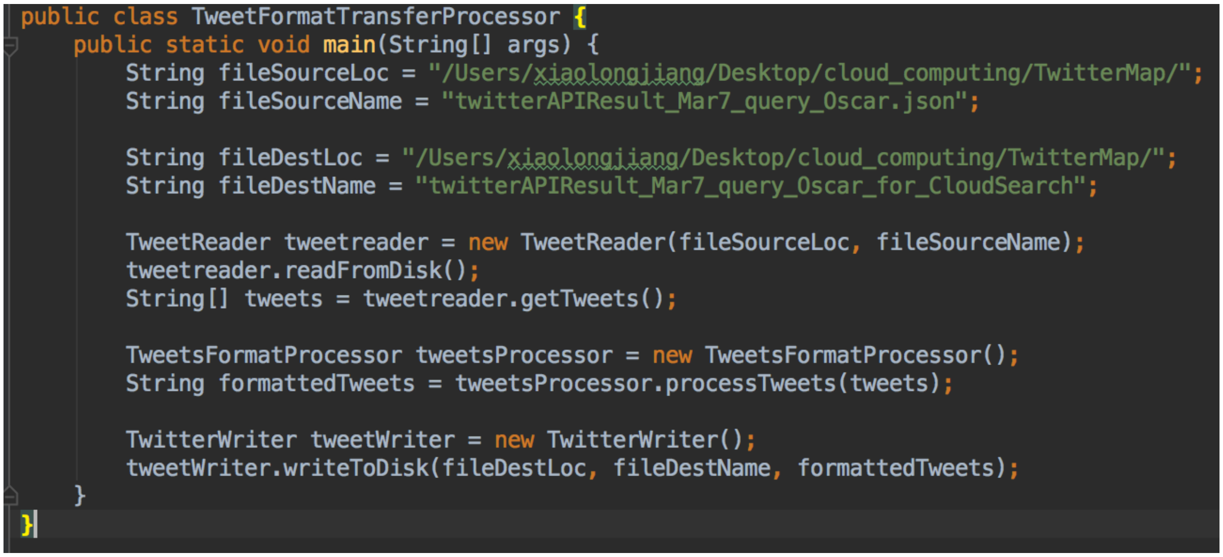
```
{"status":{"rid":"6arSnbUq8iQKtKDD","time-ms":0},"hits":{"found":43,"start":0,"hit":[{"id":"706228316370550786","fields":{"latitude":"40.7142","text":"Cousins
together ð @ New York, New York https://t.co/MySYoRM1G2","longitude":"-74.0064"}},{"id":"706895169933725696","fields":{"latitude":"40.7142","text":"OH NOOOOOO LOL
@ New York, New York https://t.co/neVPdnFnyy","longitude":"-74.0064"}},{"id":"706895416739172352","fields":{"latitude":"40.7142","text":"New job research
#sipspitlisp @ New York, New York https://t.co/EE5gHos4ib","longitude":"-74.0064"}},{"id":"706895779965890562","fields":
{"latitude":"40.7142","text":"#NewYorkCity#USA#LosAngelas#Chicago# @ New York, New York https://t.co/DtpQDs85eL","longitude":"-74.0064"}},
{"id":"706896034362953729","fields":{"latitude":"40.7142","text":"#NewYorkCity#USA#LosAngelas#Chicago# @ New York, New York
https://t.co/bufY0e29as","longitude":"-74.0064"}},{"id":"706894525722189824","fields":{"latitude":"42.9783","text":"Candid. @ Amherst, New York
https://t.co/7JcVdC7Njp","longitude":"-78.8"}},{"id":"706895707693719552","fields":{"latitude":"40.7779","text":"Growing @ Manhattan, New York
https://t.co/9VNTwR8okW","longitude":"-73.9675"}},{"id":"706895857803763712","fields":{"latitude":"40.8521996","text":"Take me to the Bronx, its where I want to
be.  Because I root for New York, New York City!!!… https://t.co/V3uLekO6aK","longitude":"-73.9698486"}},{"id":"706896150289371136","fields":
{"latitude":"40.7142","text":"Last day here and it's safe to say I â¤ï¸ NY. @ New York, New York https://t.co/yWBJy9uw65","longitude":"-74.0064"}},
{"id":"706228205821239296","fields":{"latitude":"40.75619324","text":"Just a litros break (at @McDonalds in New York, NY)
https://t.co/0qrFFkTqqX","longitude":"-73.98803392"}}]}}
```

3. Data Processor from Twitter API JSON File to AWS CloudSearch JSON File

Since file format returned from Twitter API is different from upload format in AWS CloudSearch. We need to write a program to transfer Tweets format. The program reads original tweet file from disk, parses original tweet file in each index and put indices into standard format of CloudSearch, and write the result to disk.

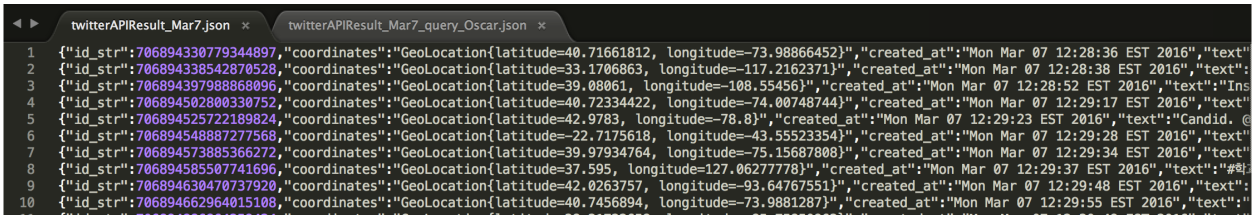
Moreover, S3 can be used in future to process data on the cloud, EC2 can be used in future to run parse program, and the data output from EC2 could upload to CloudSearch directly.

Figure below is main program of Tweet Format Transfer Professor.



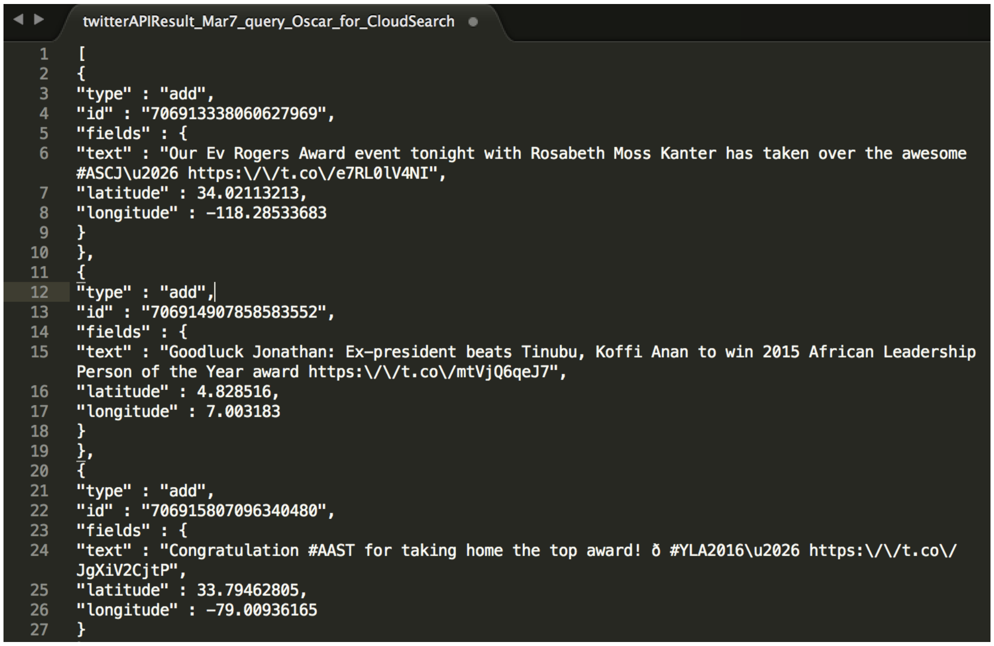
```java
public class TweetFormatTransferProcessor {
    public static void main(String[] args) {
        String fileSourceLoc = "/Users/xiaolongjiang/Desktop/cloud_computing/TwitterMap/";
        String fileSourceName = "twitterAPIResult_Mar7_query_Oscar.json";

        String fileDestLoc = "/Users/xiaolongjiang/Desktop/cloud_computing/TwitterMap/";
        String fileDestName = "twitterAPIResult_Mar7_query_Oscar_for_CloudSearch";

        TweetReader tweetreader = new TweetReader(fileSourceLoc, fileSourceName);
        tweetreader.readFromDisk();
        String[] tweets = tweetreader.getTweets();

        TweetsFormatProcessor tweetsProcessor = new TweetsFormatProcessor();
        String formattedTweets = tweetsProcessor.processTweets(tweets);

        TwitterWriter tweetWriter = new TwitterWriter();
        tweetWriter.writeToDisk(fileDestLoc, fileDestName, formattedTweets);
    }
}
```

Figure below is Tweets returned from Twitter API


twitterAPIResult_Mar7.json
twitterAPIResult_Mar7_query_Oscar.json

```
{"id_str":706894330779344897,"coordinates":"GeoLocation{latitude=40.71661812, longitude=-73.98866452}","created_at":"Mon Mar 07 12:28:36 EST 2016","text"
{"id_str":706894338542870528,"coordinates":"GeoLocation{latitude=33.1706863, longitude=-117.2162371}","created_at":"Mon Mar 07 12:28:38 EST 2016","text":
{"id_str":706894397988868096,"coordinates":"GeoLocation{latitude=39.08061, longitude=-108.55456}","created_at":"Mon Mar 07 12:28:52 EST 2016","text":"Ins
{"id_str":706894502800330752,"coordinates":"GeoLocation{latitude=40.72334422, longitude=-74.00748744}","created_at":"Mon Mar 07 12:29:17 EST 2016","text"
{"id_str":706894525722189824,"coordinates":"GeoLocation{latitude=42.9783, longitude=-78.8}","created_at":"Mon Mar 07 12:29:23 EST 2016","text":"Candid. @
{"id_str":706894548887277568,"coordinates":"GeoLocation{latitude=-22.7175618, longitude=-43.55523354}","created_at":"Mon Mar 07 12:29:28 EST 2016","text"
{"id_str":706894573885366272,"coordinates":"GeoLocation{latitude=39.97934764, longitude=-75.15687808}","created_at":"Mon Mar 07 12:29:34 EST 2016","text"
{"id_str":706894585507741696,"coordinates":"GeoLocation{latitude=37.595, longitude=127.06277778}","created_at":"Mon Mar 07 12:29:37 EST 2016","text":"#확
{"id_str":706894630470737920,"coordinates":"GeoLocation{latitude=42.0263757, longitude=-93.64767551}","created_at":"Mon Mar 07 12:29:48 EST 2016","text":
{"id_str":706894662964015108,"coordinates":"GeoLocation{latitude=40.7456894, longitude=-73.9881287}","created_at":"Mon Mar 07 12:29:55 EST 2016","text":"
```

Figure below is Tweet file after ProcessTwitterFile Program.



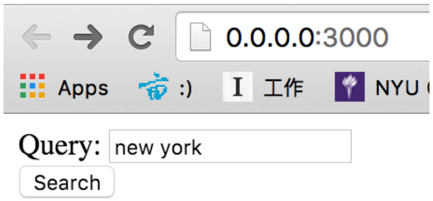
```
[
{
"type" : "add",
"id" : "706913338060627969",
"fields" : {
"text" : "Our Ev Rogers Award event tonight with Rosabeth Moss Kanter has taken over the awesome #ASCJ\u2026 https:\/\/t.co\/e7RL0lV4NI",
"latitude" : 34.02113213,
"longitude" : -118.28533683
}
},
{
"type" : "add",
"id" : "706914907858583552",
"fields" : {
"text" : "Goodluck Jonathan: Ex-president beats Tinubu, Koffi Anan to win 2015 African Leadership Person of the Year award https:\/\/t.co\/mtVjQ6qeJ7",
"latitude" : 4.828516,
"longitude" : 7.003183
}
},
{
"type" : "add",
"id" : "706915807096340480",
"fields" : {
"text" : "Congratulation #AAST for taking home the top award! ð #YLA2016\u2026 https:\/\/t.co\/JgXiV2CjtP",
"latitude" : 33.79462805,
"longitude" : -79.00936165
}
```

4. Web Interface Using Ruby on Rails

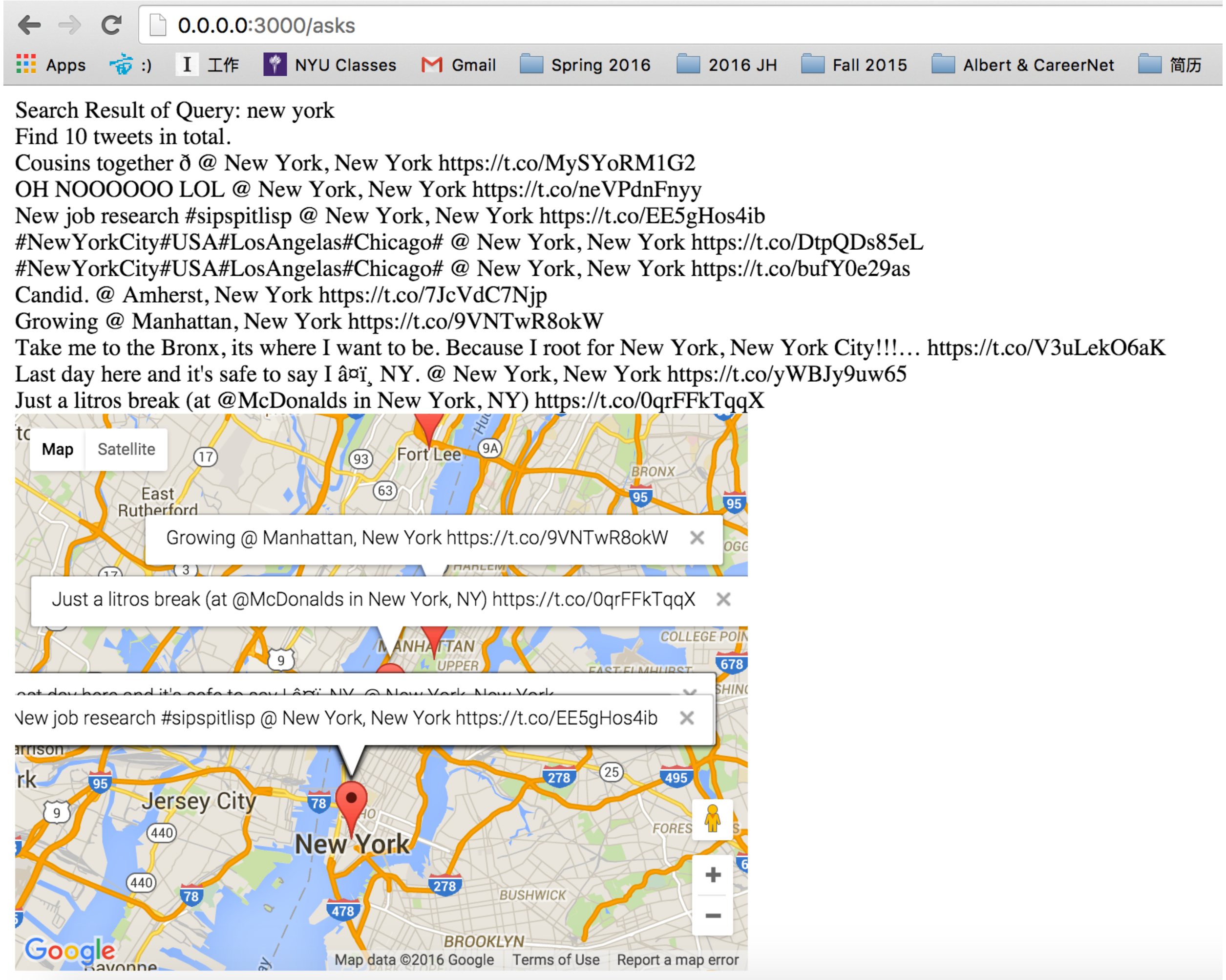
4.1 User Interface

We are using Ruby on Rails to build the UI of twitter map. The UI is simple and clear— typing query into text filed of 'query', and then click the 'search' button. Then a web page of relevant tweets, and a map of tweets' text content and location are.

Figure below is searching User Interface. User could type query into textfield and then click search.



Query: new york



Search Result of Query: new york
Find 10 tweets in total.
Cousins together ð @ New York, New York https://t.co/MySYoRM1G2
OH NOOOOOO LOL @ New York, New York https://t.co/neVPdnFnyy
New job research #sipspitlisp @ New York, New York https://t.co/EE5gHos4ib
#NewYorkCity#USA#LosAngelas#Chicago# @ New York, New York https://t.co/DtpQDs85eL
#NewYorkCity#USA#LosAngelas#Chicago# @ New York, New York https://t.co/bufY0e29as
Candid. @ Amherst, New York https://t.co/7JcVdC7Njp
Growing @ Manhattan, New York https://t.co/9VNTwR8okW
Take me to the Bronx, its where I want to be. Because I root for New York, New York City!!!… https://t.co/V3uLekO6aK
Last day here and it's safe to say I â¤ï¸ NY. @ New York, New York https://t.co/yWBJy9uw65
Just a litros break (at @McDonalds in New York, NY) https://t.co/0qrFFkTqqX

Growing @ Manhattan, New York https://t.co/9VNTwR8okW ×

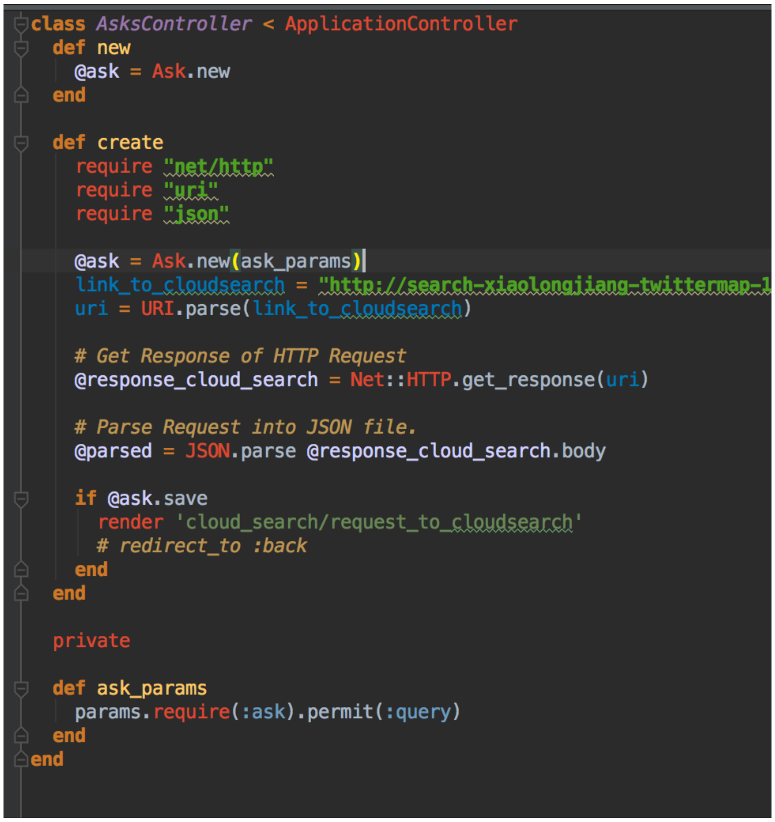
Just a litros break (at @McDonalds in New York, NY) https://t.co/0qrFFkTqqX ×

New job research #sipspitlisp @ New York, New York https://t.co/EE5gHos4ib ×

4.2 Send Request to CloudSearch and Get Result

At the UI, we get User's query input. First, we store the query into database. Then concatenate query string with CloudSearch Endpoint. After that, we can send a HTTP Request to CloudSearch and get response.

Figure below is ruby code of Request Controller.



```ruby
class AsksController < ApplicationController
  def new
    @ask = Ask.new
  end

  def create
    require "net/http"
    require "uri"
    require "json"

    @ask = Ask.new(ask_params)
    link_to_cloudsearch = "http://search-xiaolongjiang-twittermap-1
    uri = URI.parse(link_to_cloudsearch)

    # Get Response of HTTP Request
    @response_cloud_search = Net::HTTP.get_response(uri)

    # Parse Request into JSON file.
    @parsed = JSON.parse @response_cloud_search.body

    if @ask.save
      render 'cloud_search/request_to_cloudsearch'
      # redirect_to :back
    end
  end

  private

  def ask_params
    params.require(:ask).permit(:query)
  end
end
```

5. Google Map API and Rails

We are using Javascript to access Google Map API. The UI get query from user, and then sends query http request to CloudSearch, the returned JSON file is parsed to a Hash data structure by ruby. Then the tweet text and geological information is passed to Google Map API. We initialized Google map object and configured its view center, zoom degree and mapType. Then we inserted tweet makers into map by its coordinates and inserted infoWindow into map by its text. We do it to all tweet data set in a for loop. At last, we call map's event to addDomListener to load all these data.

Since Rails is embedded with HTML and Javascript, we can put HTML/Ruby/Javascript code together.

6. Deploy Rails Application on Elastic Beanstalk

To deploy Rails web application, we need to:

(1)Setup git repository

```
$ git init
$ git add -A
$ git commit -m “default rails project”
```

(2) Configure the EB CLI

```
$ eb init
$ git commit -am “updated .gitignore”
```

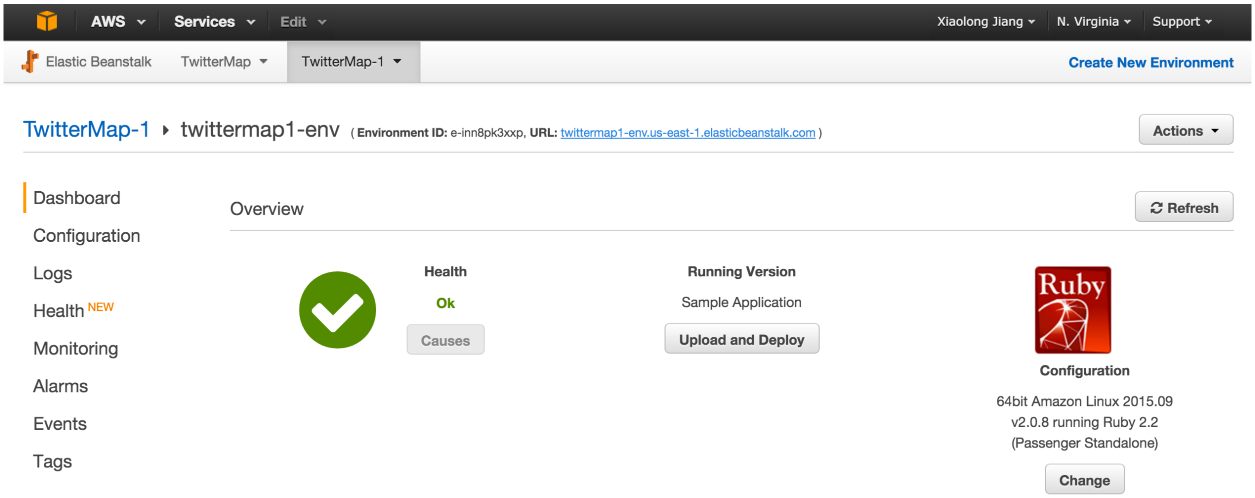
(3) add a gem ‘puma’ into Gemfile

```
$ git commit -am “Add Puma to Gemfile”
```

(4) Deploy the project

```
$ eb create rails-beanstalk-env
```

Figure below is Rails application successfully deployed on Elastic Beanstalk



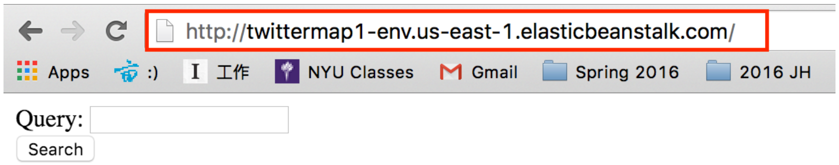
TwitterMap-1 ▸ twittermap1-env (**Environment ID:** e-inn8pk3xxp, **URL:** twittermap1-env.us-east-1.elasticbeanstalk.com)

Health
Ok

Running Version
Sample Application

Configuration
64bit Amazon Linux 2015.09 v2.0.8 running Ruby 2.2 (Passenger Standalone)

Figure below is accessing Rails application through EB



http://twittermap1-env.us-east-1.elasticbeanstalk.com/

Query:

7. Refinement

In this assignment, we could implement S3 and EC2 into Twitter crawling process and uploading files onto CloudSearch process. If we can do crawling and update in a time period, say 1 hour per update, then real-time twitter information crawl/search platform can be established purely on AWS cloud.

Moreover, we’re crawling twitter in a single thread program. If we can build a multi-threads crawling program, we can do much faster than single thread since the processor wouldn’t be idle during the response waiting time.

8. Conclusion

AWS is very useful and powerful platform. We can build projects based on the services AWS provided. In this assignment, we used AWS CloudSearch and AWS Beanstalk in this project. AWS CloudSearch is more like a powerful search engine for structured and semi-structured file, and Beanstalk can deploy Web application in a second. These services could save a lot time and money for startup even mid-sized companies.