GEOG 594: AWS Lab4
Submitted by: Harshal Sanap
Dated: 10/18/2018

1. What are the advantages and disadvantage of using Amazon EC2 for building a research project website?

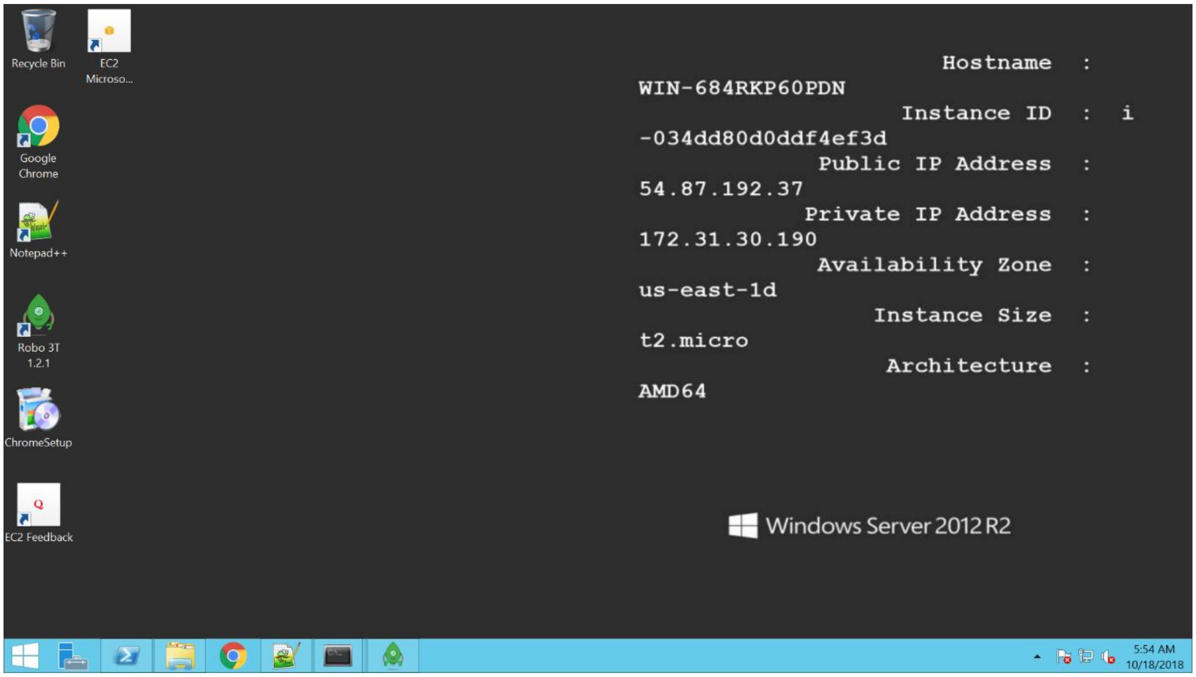
Advantages of using Amazon Elastic Cloud Computing (EC2):

- Ease of use: AWS provides an easy to use platform to host already existing applications or new SaaS based applications. The only additional step required is to log into a virtual machine via a remote desktop.
- Flexible: AWS allows you to select the RAM, hard disk and processing unit as required by the applications need. It also allows you to select the operating system, programming languages, database required for your applications. These options can be updated at any time.
- Cost Effective: As we need to pay only for the resources that we use on the go the AWS platform is very cost effective. It doesn't require any upfront commitments or long-term contracts.
- Reliable: As Amazon is a well established multi billion-dollar company, AWS is known to be highly reliable platform available in the market currently.
- Scalable & High performance: Auto scaling and elastic load balancing is provided by the AWS platform making it easy for your applications to scale up or down based on the demand.

Disadvantages of using Amazon Elastic Cloud Computing (EC2):

- Internet Downtime: As AWS is a cloud computing service, it requires continuous stream of internet. As its internet dependent, we can not perform any work incase the internet is down for some reason.
- Security: As AWS is a remote login platform it can become prone to being hacked via internet though AWS has high security measures.
- Vendor lock in: It is very difficult to migrate from one cloud platform to another.
- Cost: the cloud solution on a small scale and for short term projects can be expensive.

2. Attach the screenshot of your EC2 Virtual Server.



```
                         Hostname  :
WIN-684RKP60PDN
                      Instance ID  :  i
-034dd80d0ddf4ef3d
                Public IP Address  :
54.87.192.37
               Private IP Address  :
172.31.30.190
                Availability Zone  :
us-east-1d
                    Instance Size  :
t2.micro
                     Architecture  :
AMD64
```

GEOG 594: AWS Lab4

Submitted by: Harshal Sanap

Dated: 10/18/2018

3. What are the differences between traditional SQL databases and NoSQL databases?

SQL	NoSQL
1. It's a relational database (RDBMS)	1. It's a non-relational database
2. Database is table based	2. Database is in the form of key-value pairs
3. They have predefined schema	3. They have dynamic schema for unstructured data
4. They are generally vertically scalable. Meaning you can manage the increasing load by increasing the RAM, SSD, CPU etc.	4. They are horizontally scalable. Meaning, you can manage the increasing loads by simply adding a new server
5. It uses SQL as a query language	5. It uses Unstructured Query Language (UnQL) which varies from database to database
6. It is a good fit for using complex queries	6. It is not good for complex queries
7. Data is stored in 2 dimensional table format	7. Data is stored in JSON file format
8. Generally used incase of non hierarchical data	8. Mostly used incase of hierarchical data
9. Examples: MySQL, SQLite, Oracle, Postgres	9. Examples: MongoDB, Cassandra, HBase, Redis

4. What is your selected keyword in the **get_tweets-yourname.py** to collect tweets? How many tweets did you get? **List the content of first 3 tweets in your report. (Please include all metadata elements in the example, including ID, source, text, location, etc.) Discuss the search results and how to improve the keyword search for your subject.**

Response:

I used "**Halloween**" as my search query.

I received 100 results as output of the search query as the max output record limit was set to 100.

The contents of the tweets can be seen as follows:

1st Tweet:

```
User: lunmione
Tweet: b'if any hot witches want to kidnap me and take me to their swamp s
hack for halloween please feel free'
Time: b'Fri Oct 19 02:39:05 +0000 2018'
Location: b'e/em'
```

2nd Tweet:

```
User: MorningJournal
Tweet: b'KISS FM Halloween Bash is Oct. 27 at JACK Thistledown Racino http
s://t.co/Ye7W9kQmgL via @foodandfunneo'
Time: b'Fri Oct 19 02:39:05 +0000 2018'
Location: b'T: 41.35179,-82.119916'
```

3rd Tweet:

```
User: yu83720313
Tweet: b'RT @ViSULOG: HYDEHALLOWEEN PARTY 2018 https://t.co/sEIipGdrHu #vi
sulog'
```

GEOG 594: AWS Lab4
Submitted by: Harshal Sanap
Dated: 10/18/2018

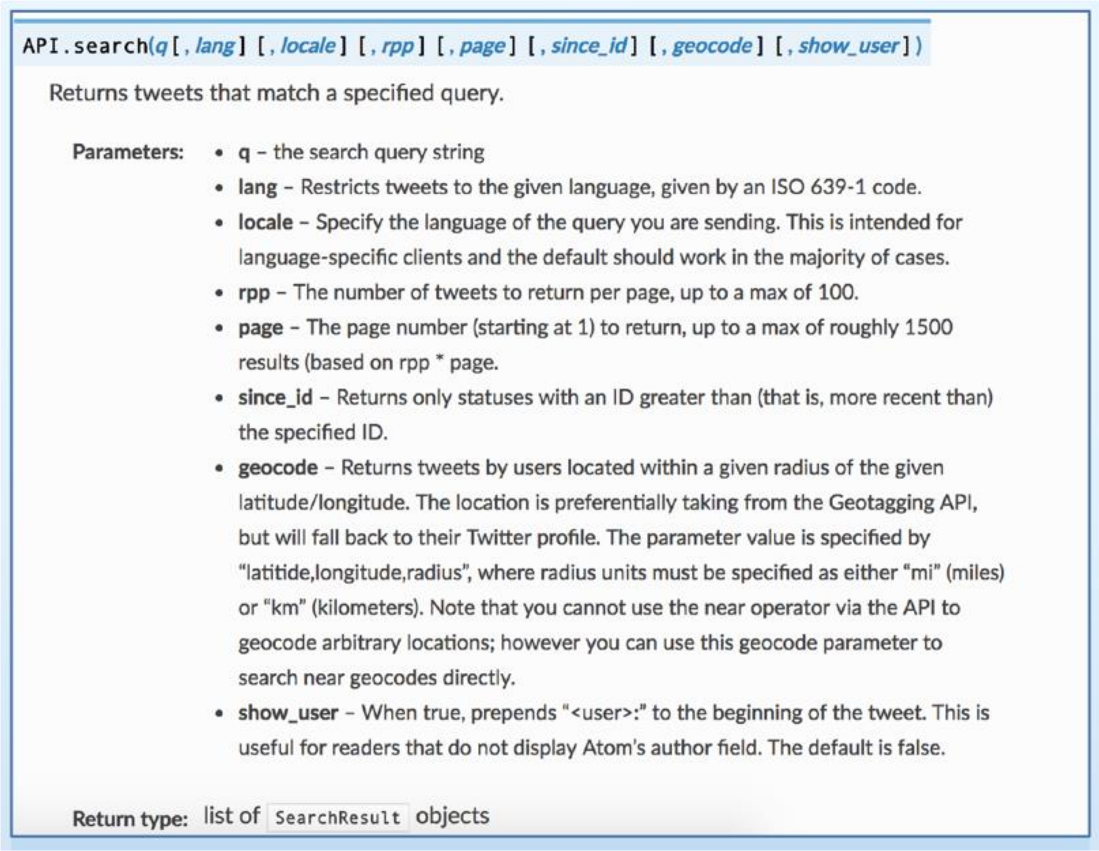
```
Time: b'Fri Oct 19 02:39:05 +0000 2018'
Location: b''
```

The detailed information of the above tweets in JSON format can be viewed from the attached document.

Tweet_json.docx

Observations:

- The tweepy package used in this assignment's python code simply returns the output of a search query without any kind of filtering.
- In my search query I also received few tweets with Chinese language which I didn't understand thus to avoid this problem we can refine the search query by inputting the language parameter. On researching google I found the below attached list of parameters.



```
API.search(q[, lang] [, locale] [, rpp] [, page] [, since_id] [, geocode] [, show_user])
```

Returns tweets that match a specified query.

Parameters:
- **q** – the search query string
- **lang** – Restricts tweets to the given language, given by an ISO 639-1 code.
- **locale** – Specify the language of the query you are sending. This is intended for language-specific clients and the default should work in the majority of cases.
- **rpp** – The number of tweets to return per page, up to a max of 100.
- **page** – The page number (starting at 1) to return, up to a max of roughly 1500 results (based on rpp * page.
- **since_id** – Returns only statuses with an ID greater than (that is, more recent than) the specified ID.
- **geocode** – Returns tweets by users located within a given radius of the given latitude/longitude. The location is preferentially taking from the Geotagging API, but will fall back to their Twitter profile. The parameter value is specified by "latitide,longitude,radius", where radius units must be specified as either "mi" (miles) or "km" (kilometers). Note that you cannot use the near operator via the API to geocode arbitrary locations; however you can use this geocode parameter to search near geocodes directly.
- **show_user** – When true, prepends "<user>:" to the beginning of the tweet. This is useful for readers that do not display Atom's author field. The default is false.

Return type: list of `SearchResult` objects

Source: https://www.toptal.com/python/twitter-data-mining-using-python

- Personally, I think we can collect information related to tweets around different regions by giving the geocode information. This information can later be compared to understand how the interest of a particular topic varies from one region to another.