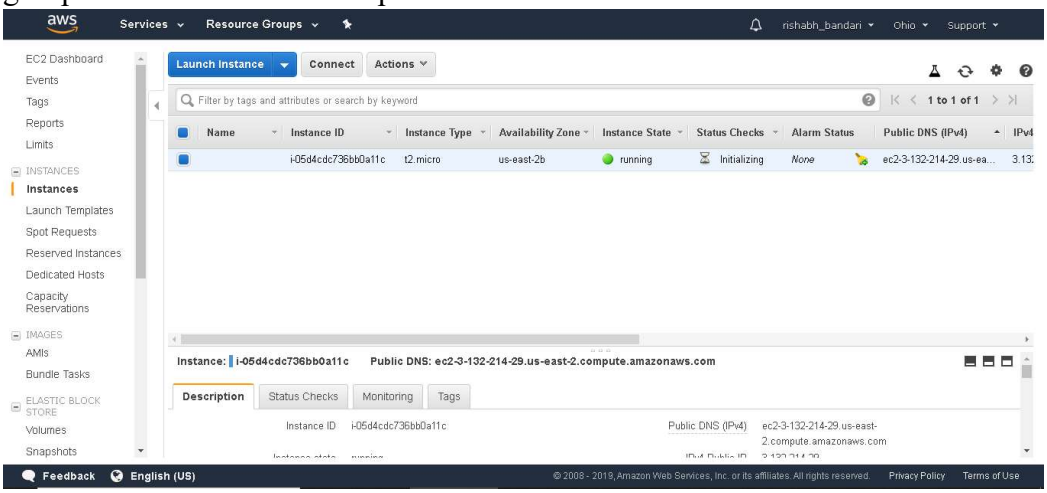
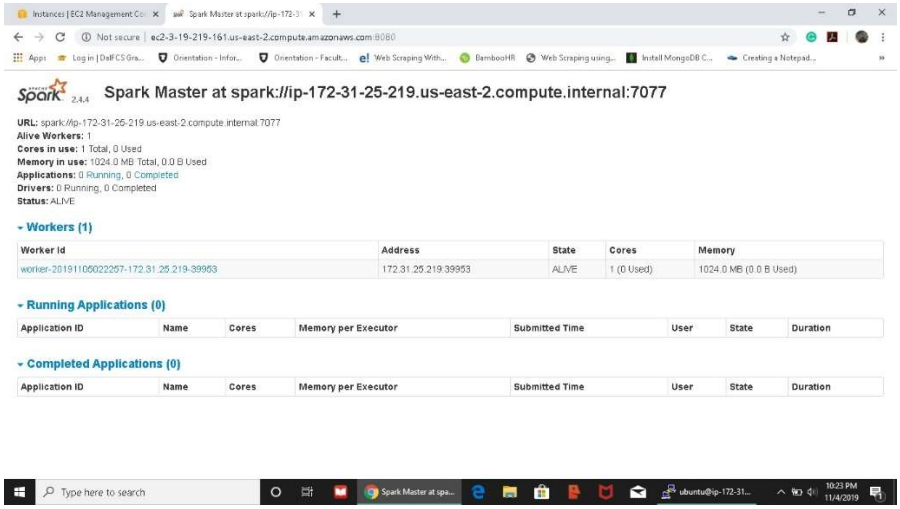
DATA MANAGEMENT WAREHOUSING AND ANALYTICS – ASSIGNMENT 2

A. Cluster Setup

- The cloud setup was done with the help of Amazon Elastic Compute Cloud (EC2). The steps in Lab: Tutorial 3 was followed to create the AWS instance. Finally, the security groups were edited to enable ports 8080 and 8081.



- After creating the instance, the installation and setup process of Apache Spark was done with the help of Tutorial 5 from the Labs. (Version 2.4.4 was installed)
    1. Using the ubuntu virtual machine, the installation file was downloaded in the new server directory and unpacked using the following commands.
        - wget http://apache.forsale.plus/spark/spark-2.4.4/spark-2.4.4-bin-hadoop2.7.tgz
        - sudo tar zxvf spark-2.4.4-bin-hadoop2.7.tgz
    2. 'export PYSPARK_PYTHON=python3' was used to install python.
        - After starting the master and slave, the number of workers was checked.



    3. Finally, the 'sudo ./spark-2.4.0-bin-hadoop2.7/bin/pyspark' command was run to start the python shell

B. Data Extraction Process

The extraction of tweets was done with the help of Python. The Tweepy API was used in retrieving and cleaning the data. In the case of the NewsAPI, the open source python code was used to extract the articles. The jupyter notebook was used for all the processes. The tweepy API gives access to the RESTful API of Twitter.

- Twitter Data Extraction & Transformation:
    1. Firstly, with the help of the Twitter developer account, a Twitter application was created. This application gives us the required Consumer Keys and Access Tokens that will be required to access the API.
    2. Tweepy was installed with the pip install command.
    3. Search API: This allows us to retrieve historical data of tweets. In simple words, it works and retrieves the same way the twitter search bar searches for tweets and related keywords. The major function used in the program uses just the Consumer Key, Consumer Secret, Access Token and Access Token Secret as parameters. As seen in the python file attached for search tweets, the program accesses the API, retrieves the tweets and stores them in an output csv file. I have only included the timestamps, tweet text and username in this retrieval process. Attention was given to excluding out special characters, links and other unwanted text.
    4. Stream API: Streaming of tweets involves retrieval of live Twitter data. The streaming API is used to push data into a particular session on a real time basis. The tweepy.Stream creates a streaming session and sends data to the stream listener. The method, on_data receives the messages depending on the features required. This the streaming of data was done with the help of the StreamListener, Stream object and the Stream Connection.

        (Please find the respective ipynb files attached as part of the complete zip file.)

- News API Data Extraction
    1. A developer account was first created in newsapi.org and the API key was generated.
    2. The newsapi python library was installed using - pip install newsapi-python
    3. Using the open source modules, the data was extracted based on the keyword. I had restricted the source to CBC News. (cbc.ca)
    4. The request parameter q is used to search for specific keywords or phrases.
    5. Since a free version of the news API was used, a request limitation was encountered. Hence, only the first 100 records of data were retrieved.

C. Cleaning Process

- The data retrieved from the News API did not require much cleaning as it was mostly relevant and only limited records retrieved.
- Much of the cleaning was done while streaming and searching for the tweets with the help of the tweepy library.

- While searching for the tweets, the ‘encoding="utf-8"’ was while creating the output file. This removes all the unwanted characters and links.
- In the streaming process,

  links = re.sub(r"https\S+", " ", tweet) was used to remove the unwanted links.

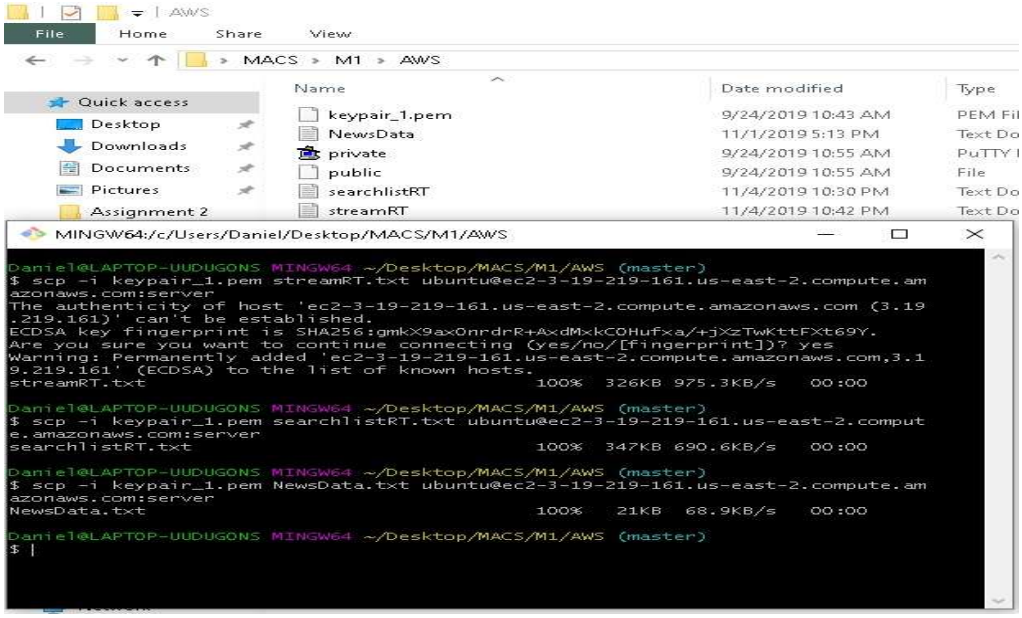
  chars = re.sub(r'[^\x00-\x7f]+', ' ', links) was used to remove the special characters.

- Finally, all the data was collected and structured manually with the help of Notepad and Excel.

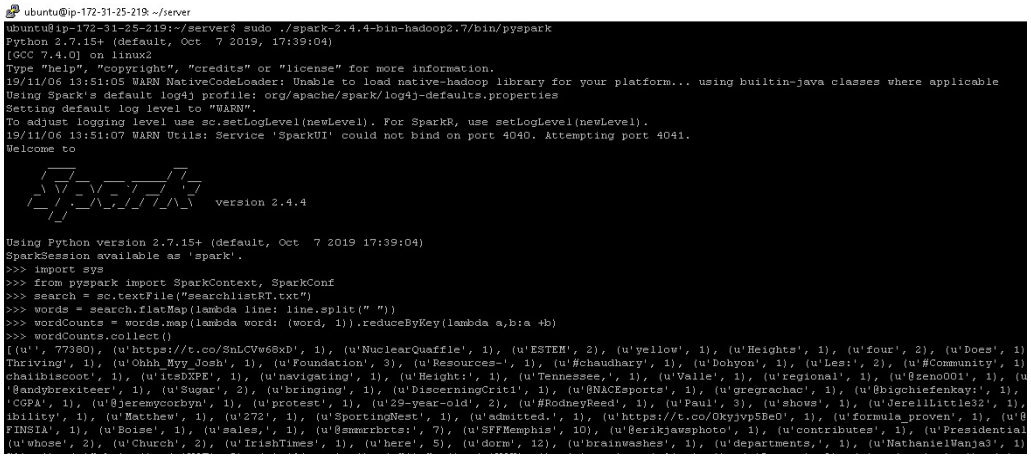
(Please find attached the respective files for retrieved tweets.)

D. Data Processing

- The cleaned tweets were uploaded to Spark with the help of the scp command, which allowed for copying files between two different directory locations (Local to remote system).



- The text file is read, and the count of every unique word is calculated with the help of the Map-Reduce.



- The last step involved performing a frequency count of the following substrings or words.

Word	Count
Education	378
Canada	495
University	1791
Dalhousie	2
expensive	3
good school/schools	20
bad school/schools	0
faculty	13
Computer Science	2
Graduate	586

It can be inferred that the word University, had the highest frequency whereas there were no tweets with the phrase bad schools.

E. References

[1] ritvikmath, "GitHub," GitHub, [Online]. Available: https://github.com/ritvikmath/ScrapingData/blob/master/Scraping%20Twitter%20Data.ipynb. [Accessed 25th October 2019].

[2] J. Roesslein, "Tweepy.org," [Online]. Available: http://docs.tweepy.org/en/v3.4.0/streaming_how_to.html. [Accessed 28 October 2019].

[3] Harrison, "pythonprogramming," [Online]. Available: https://pythonprogramming.net/twitter-api-streaming-tweets-python-tutorial/. [Accessed 26 October 2019].

[4] "StackOverflow," [Online]. Available: https://pythonprogramming.net/twitter-api-streaming-tweets-python-tutorial/. [Accessed 30 October 2019].

[5] "Python Examples," [Online]. Available: https://pythonexamples.org/pyspark-word-count-example/. [Accessed 2 November 2019].

[6] "ApacheSpark.org," [Online]. Available: https://spark.apache.org/examples.html. [Accessed 3 November 2019].

[7] "Tweepy," [Online]. Available: http://docs.tweepy.org/en/v3.4.0/index.html. [Accessed October 2019].