Exercise 2 - Twitter Streaming Application

Application Purpose

The purpose of the application is to capture a stream of data from the Twitter API and create a database of the words used in the intercepted tweets, including the number of times each word appears. The application can be used to query the database for a specific word, and it will return the number of times that word was captured from the stream. In addition, if given a range of values, the application can return a list of all the word whose count values fall within that range.

Architecture & Topology

The application uses Apache Storm to capture and process the Twitter data and store it in a Postgresql database. The Storm application consists of the following components:

- Twitter spout: Captures tweets from the Twitter API, running 3 processes in parallel.
- Parser bolt: Receives tweets from the Twitter spout parses them into individual words, running 3 parallel processes.
- Counter bolt: Receives words from the Parser bolt and searches the Postgres database. If the word appears in the database, the bolt updates the count. If not, it adds the word to the database and sets the count value to 1.

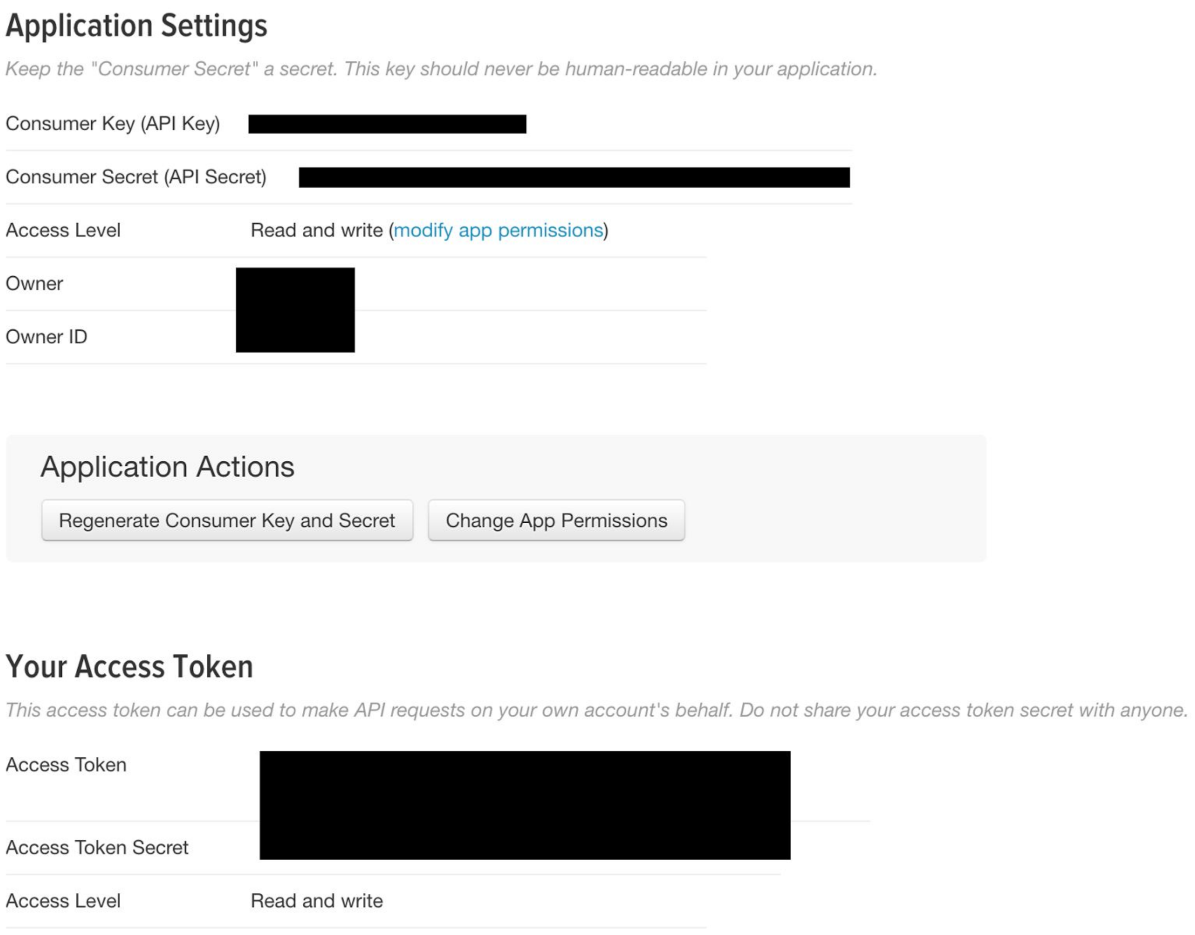
Dependencies

The application is designed to run on an Amazon Web Services EC2 instance based on the *UCB MIDS W205 EX2-FULL* AMI (AMI ID: ami-d4dd4ec3).
To function properly, the application requires installation of the following Python libraries:

- psycopg2 (version 2.6.2)
  - psycopg2.extensions
- tweepy
- __future__
- itertools
- time
- re
- copy
- Queue
- threading
- sys
- os
- collections

To access the Twitter API, the application also requires access tokens obtained from Twitter. The user is required to create an application associated with their Twitter account, which will give them the following four pieces of information:

- A consumer key that identifies your application.
- A consumer secret that acts as a password for your application.
- An access token that identifies your authorized access.
- An access token secret that acts as a password for that authorized access.



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]
Access Level	Read and write (modify app permissions)
Owner	[redacted]
Owner ID	

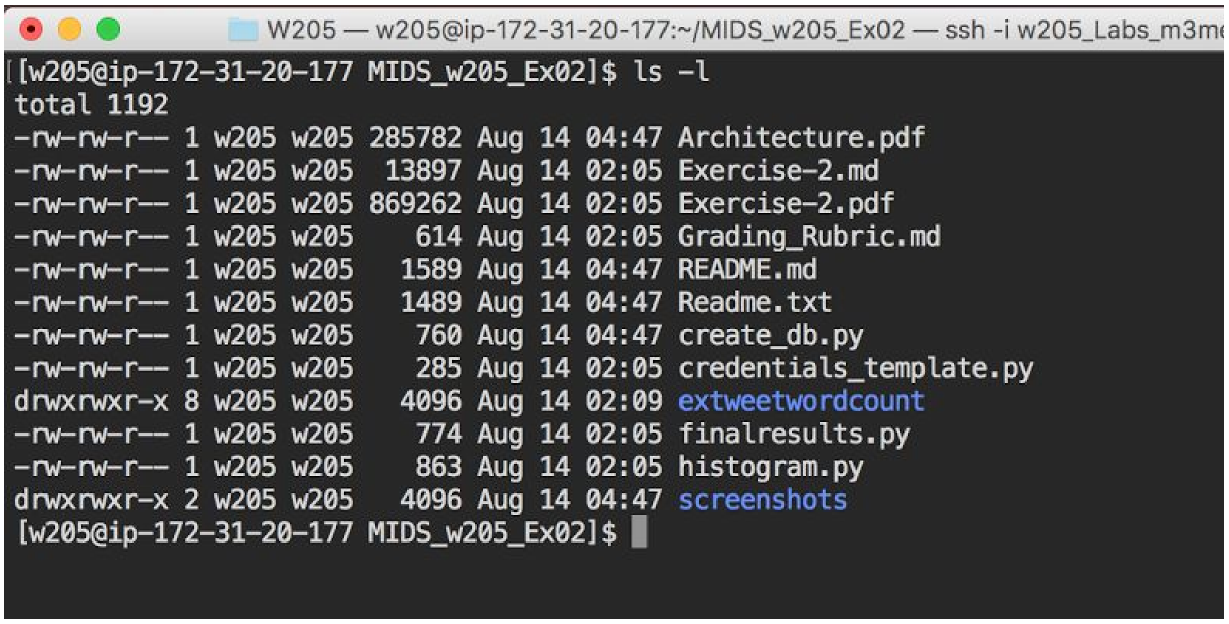
Application Actions

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 secret with anyone.

Access Token	[redacted]
Access Token Secret	
Access Level	Read and write

File Structure


W205 — w205@ip-172-31-20-177:~/MIDS_w205_Ex02 — ssh -i w205_Labs_m3m

```
[[w205@ip-172-31-20-177 MIDS_w205_Ex02]$ ls -l
total 1192
-rw-rw-r-- 1 w205 w205 285782 Aug 14 04:47 Architecture.pdf
-rw-rw-r-- 1 w205 w205  13897 Aug 14 02:05 Exercise-2.md
-rw-rw-r-- 1 w205 w205 869262 Aug 14 02:05 Exercise-2.pdf
-rw-rw-r-- 1 w205 w205    614 Aug 14 02:05 Grading_Rubric.md
-rw-rw-r-- 1 w205 w205   1589 Aug 14 04:47 README.md
-rw-rw-r-- 1 w205 w205   1489 Aug 14 04:47 Readme.txt
-rw-rw-r-- 1 w205 w205    760 Aug 14 04:47 create_db.py
-rw-rw-r-- 1 w205 w205    285 Aug 14 02:05 credentials_template.py
drwxrwxr-x 8 w205 w205   4096 Aug 14 02:09 extweetwordcount
-rw-rw-r-- 1 w205 w205    774 Aug 14 02:05 finalresults.py
-rw-rw-r-- 1 w205 w205    863 Aug 14 02:05 histogram.py
drwxrwxr-x 2 w205 w205   4096 Aug 14 04:47 screenshots
[w205@ip-172-31-20-177 MIDS_w205_Ex02]$ 
```

The application repository is built upon the provided exercise directory (https://github.com/UC-Berkeley-I-School/w205-summer-17-labs-exercises/tree/master/exercise_2)
The Storm components of the application are contained in the “extweetwordcount” folder:

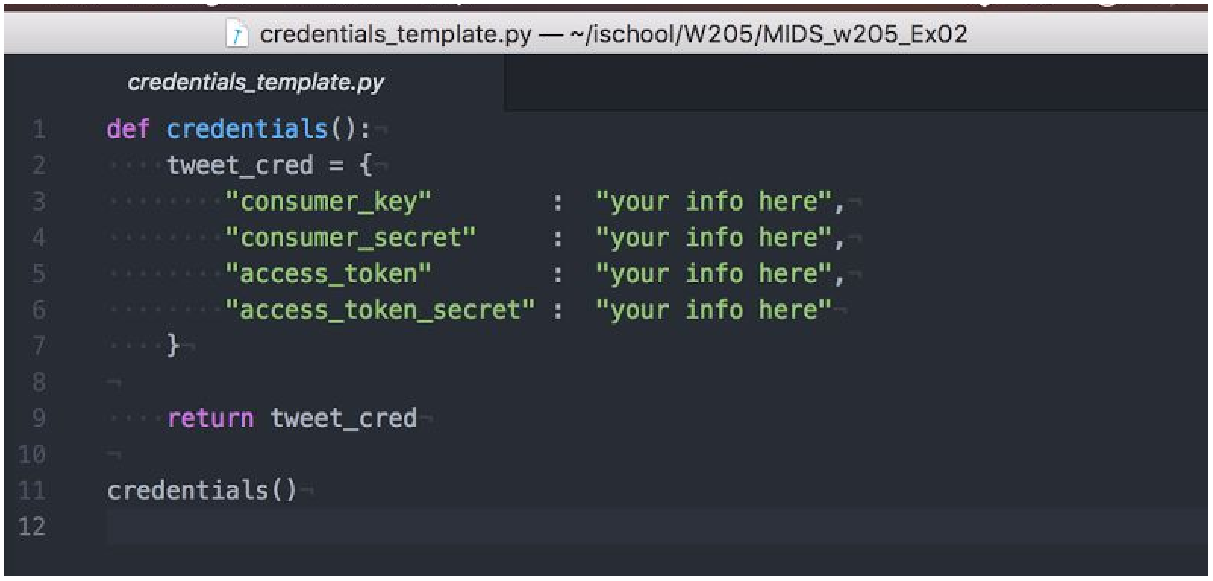
- src - Spout and bolt scripts
- topologies - Contains the topology script that links the spouts and bolts together.

The main MIDS_w205_Ex02 folder contains the following components:

- finalresults.py - Script to query the database for a specific word and return the number of occurrences.
- histogram.py - Given two numbers, this script queries the database and returns all words whose count values fall in that range.
- credentials_template.py - A sample python script for passing Twitter credentials to the Storm application (see next section for details).

Important Notes for Execution

To run properly, the Storm application requires valid Twitter application credentials. The repository contains a python script named “credentials_template.py.” Before launching the application, copy this script to /extweetwordcount/src/spouts and rename it as “tweet_cred.py” Then edit the script, inserting your credentials in the indicated locations and save.



credentials_template.py — ~/ischool/W205/MIDS_w205_Ex02

credentials_template.py

```
def credentials():
    tweet_cred = {
        "consumer_key"        :  "your info here",
        "consumer_secret"     :  "your info here",
        "access_token"        :  "your info here",
        "access_token_secret" :  "your info here"
    }

    return tweet_cred

credentials()
```

IMPORTANT: For security, do NOT insert your credentials until after you have renamed the file. The git repository is configured to not track tweet_cred.py, thereby preventing user credentials from being published to Github. If you insert your credentials before renaming the file, your information could potentially by published, allowing unauthorized access to your Twitter account.