Architecture Documentation

Overview:

- This application will read and parse tweets from Twitter in real time, split the tweets into individual words, perform a word count, and transfers and stores the results in a postgres database.
- Technology used:
    - AWS EC2
    - Apache Storm
    - Tweepy
    - Streamparse
    - Twitter
    - Postgres
    - Python

File Structure:

- Topologies
  extweetwordcount.clj

  An application in apache storm requires a topology file (clojure) because this file will outline what the bolts and spouts are and more importantly how they are connected. The topology is essentially a pipeline for the data.

- Src
    - Spouts
      tweets.py

      The spout file outlines the components that will emit raw data. In this case we have 3 elements in this spout folder that will eventually feed into 3 elements in the parse-tweet-bolt.

      The specific data used in this exercise is twitter data so we use the python library tweepy to gather data from Twitter.

    - Bolts
      parse.py
      wordcount.py

      The bolt is a file that outlines which components will consume the data, transfer the data, send the data to other bolts, or store data in external storage systems. For this specific exercise, the 3 elements of parse-tweet-bolt will pass data along to 2 elements in count-bolt which will ultimately store the appropriately transformed data into a postgres database. Parse.py basically cleans up the raw twitter data to prepare clean twitter data for wordcount.py to receive. Wordcount.py uses

psycopg2 and stores the words from a tweet appropriately into postgres. Pyscog2 is required to establish a connection with postgres.

-