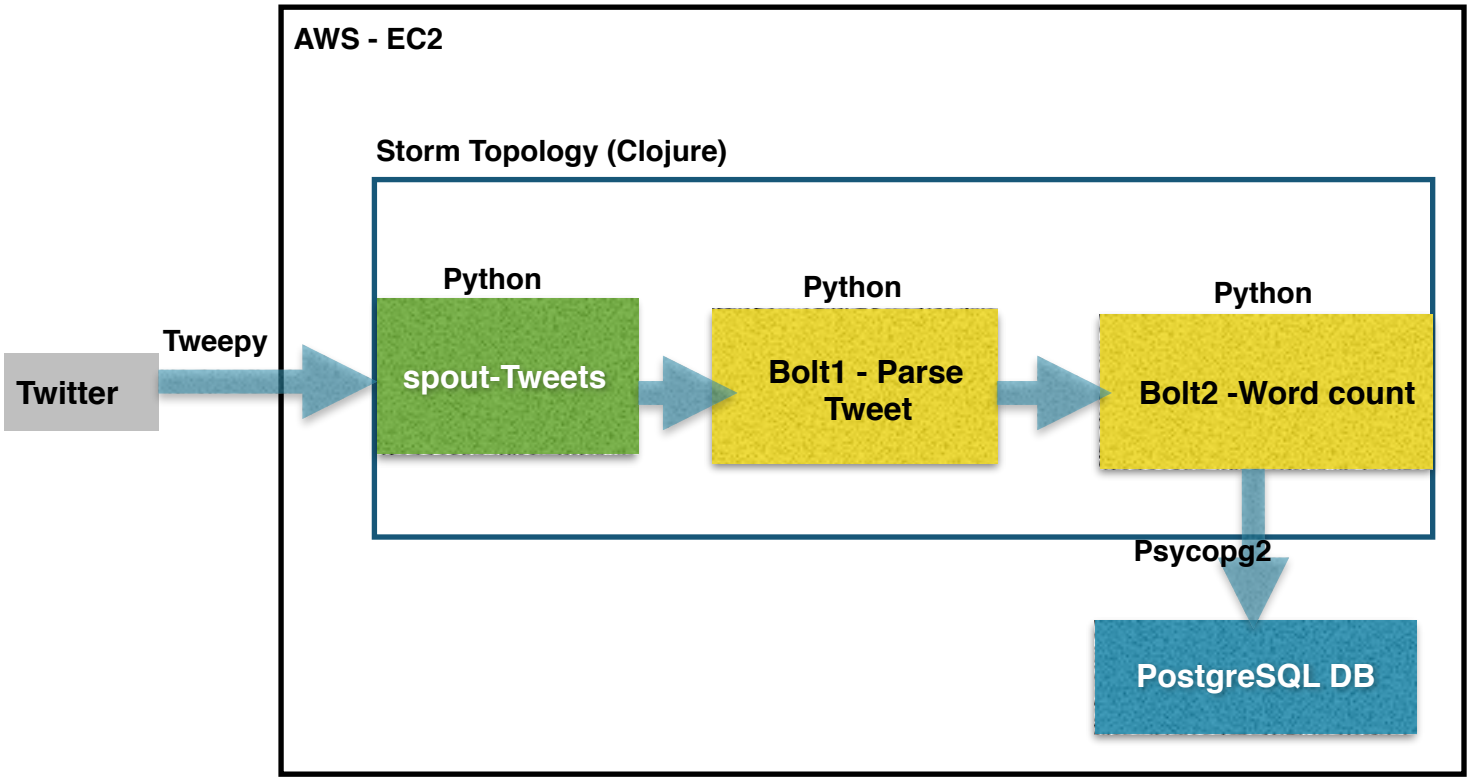
Application Summary

The application captures live data from Tweeter stream , processes and analyzes live and stores data in a relational database for further analysis. The database stores each word captured from Tweeter stream and the count of their occurrences . The data thus stored can be used for numerous analysis surrounding the popularity of usage of words in Twitter.
As example, this application out of the box provides reports on Top 20 most frequently occurring words and frequency of occurrence of any word of interest. It can also be used to analyze the relative usage of words over time and provide valuable insight on sentiment surrounding a product or any other item of interest.

Application Architecture



Live Tweets are captured in real time by using tweepy application . Tweepy creates a streaming session after authenticating user access token and secrets.
Tweet-spout emits tuples of tweets.
Parse tweet bolt accepts these tweet tuples ,split the tweets into words , filter out certain non-words and re-tweets and emits the non-empty tweet words.
Count -bolt takes the tweet words and increments the count of that word in a PostgreSQL database.

Important application components

Storm Topology - It is written in Clojure language and defines the network of spouts and bolts. In this application, one spout for collecting tweets is connected to a bolt for parsing the tweets to extract words which in turn is connected to another bolt for counting those words.

Spouts

tweets.py - It is written in Python. It uses tweepy to interact with Twitter's Streaming API and emits tuples of tweets.

Bolts

i. parse.py - It is written in Python. It accepts the tweet tuples emitted from spout tweets.py and splits the tweets into words. It filters out certain non-words and emits the tweet words.

ii. wordcount.py - It is written in Python. It accepts the tweet words from parse.py , counts those words and update the word count in PostgreSQL database. It connects with PostgreSQL DB using psycopg2 package.

PostgreSQL

Database : Tcount
Table : tweetwordcount

Serving scripts

i. **finalresults.py** - returns all the words in the table and their total count of occurrences if run without any argument. It also accepts a word as an argument and returns its count .
ii. **histogram.py** - returns all the words from the table whose total number of occurrences in the table is between the two integers passed as arguments.
iii. **barplot_top20.py** - creates a horizontal bar plot of top 20 words and their number of occurrences as captured in the tweetwordcount table