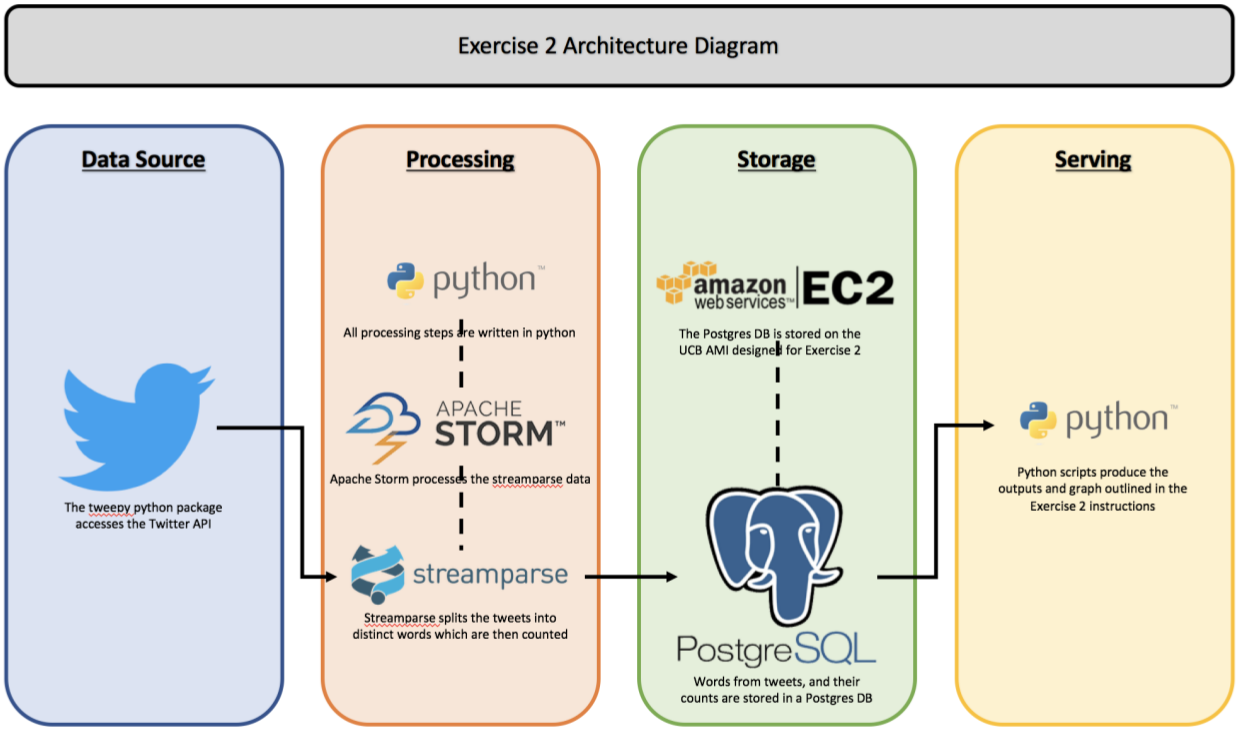
Application Idea

The idea behind this application is simply to read tweets and count the frequency of individual words. This is an end-to-end solution as we are grabbing data from the source, processing and storing the scrubbed data, and serving results about that data back to anyone who runs one of the several python scripts in the exercise_2 directory.

Architecture


Exercise 2 Architecture Diagram
Data Source
Processing
Storage
Serving
The tweepy python package accesses the Twitter API
python
All processing steps are written in python
APACHE STORM
Apache Storm processes the streamparse data
streamparse
Streamparse splits the tweets into distinct words which are then counted
amazon web services EC2
The Postgres DB is stored on the UCB AMI designed for Exercise 2
PostgreSQL
Words from tweets, and their counts are stored in a Postgres DB
python
Python scripts produce the outputs and graph outlined in the Exercise 2 instructions

All of the scripting was done in python, and Amazon EC2 will be used to host the Postgres database and all other scripts. The data source for this application is tweets from Twitter. We will be utilizing a python package called tweepy to access the tweets. Then Apache Storm and streamparse will split each tweet into individual words and count their frequencies. The results will be displayed to the users' command line while they are also stored in the Postgres database. There are several scripts in the home exercise (Plot.py, finalresults.py, histogram.py) that will provide some basic analysis on the database, but there are many other potential use cases.

Getting Started

The following are suggested steps for you to take in order to get this application up and running.

1) Start a fresh instance of the UCB Exercise 2 FULL AMI

2) Mount an EBS Volume to that instance
3) Start postgres by running “/data/start_postgres.sh”
4) Using pip, install the following python packages
    - tweepy
    - psycopg2
    - pandas
    - matplotlib
        - When importing into python, use the following code:

          ```
          import matplotlib
          matplotlib.use(‘Agg’)
          import matplotlib.pyplot as plt
          ```
5) Clone the github repository you’re reading this file from
6) Enter your Twitter API Credentials into the file /extweetwordcount/src/spouts/tweets.py
    - Test whether your credentials work by entering them again in the Twittercredentials.py file in the home directory and running hello-stream-twitter.py
7) Run the psycopg-sample.py file, which will create the Postgres database tcount and the table tweetwordcount
8) To load data into the database, cd into /extweetwordcount and run the command “sparse run”
    - Assuming everything works properly, you should see words and their counts start showing up in your window after a few seconds.
    - Enter “Ctrl + C” to stop the streamparse after you have the desired sample of words
9) In the home exercise_2 directory, you should now be able to run the files Plot.py, finalresults.py, and histogram.py. Each file contains commented code that I encourage you to read to understand the logic used to produce the output.

Directory

I encourage you to read the README.txt file in the home directory as I describe the purpose of several files, as well as any dependencies. The files that I don’t mention in that README are relating to the Apache Storm and streamparse applications. These files are listed in the table below (copied from the Exercise instructions).

Name of the program	**Location**	**Description**
tweets.py	exercise_2/tweetwordcount/src/spouts/	tweet-spout
parse.py	exercise_2/tweetwordcount/src/bolts/	parse-tweet-bolt
wordcount.py	exercise_2/tweetwordcount/src/bolts	count-bolt

Twittercredentials.py	exercise_2/	Twitter API Keys
hello-stream-twitter.py	exercise_2/	Sample Twitter stream program
tweetwordcount.clj	exercise_2/tweetwordcount/topologies/	Topology for the application
psycopg-sample.py	exercise_2/	Sample code for connecting to psycopg