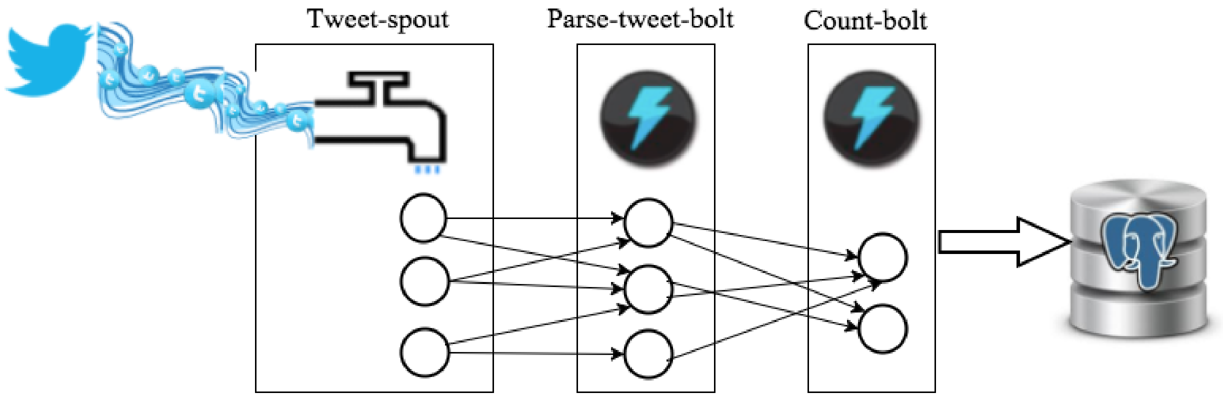
# Exercise-2 Architecture

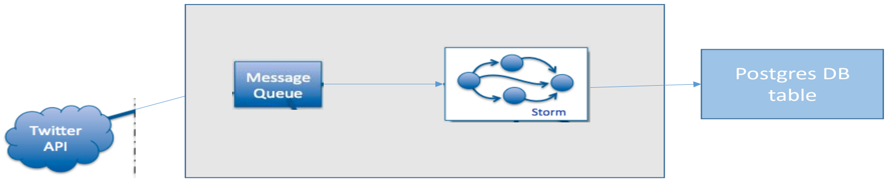
Ganesh Sundararajan

The architecture of exercise-2 is shown below.



The twitter feed is fed into the Apache Storm where the feed is broken into words. The broken tweeter feed words are then counted. The word and its frequency count is stored in the postgres database for subsequent processing by the user.

The higher level architecture of the application is shown below



The source code for the application can be obtained from git hub https://github.com/ganeshsberkeley/EC2_STEPS

Please clone the repo using the command git clone https://github.com/ganeshsberkeley/EC2_STEPS

Once the repo is clone please follow the instructions in the README.md file

The application requires a few pre-steps (assuming that you have already created the proper EC2 instance. Please refer to the PDF file in docs folder for more details)

1. Follow the README.md to setup all the tools required for running Apace Storm on Amazon EC2
2. Create a twitter account as well as application for streaming tweets (if you don't have one. Please refer to the PDF file in the docs folder for details on steps for creating an account and application)
3. Modify the **EC2_STEPS**/tweetwordcount/src/spouts/**tweets.py** file to add your credentials as outlined in the PDF file in the docs folder
4. Create the database and tweetwordcount table in postgres as outlined in the README.md file
5. In order to test the tweeter feed connection, please follow the instructions in the PDF file to modify the **EC2_STEPS**/tweetwordcount/**Twittercredentials.py** and run the **EC2_STEPS**/**tweetwordcount**/hello-stream-twitter.py. Once the python script runs successfully, you can run the application (please refer to the README.md files for instructions on how to run the application).

The EC2_STEPS/tweetword/count/src folder has the source files for the spouts and bolts used by the application.

The spout receives twitter stream from the twitter application and extracts the English language tuples and passes it on the bolt. We receive the twitter stream using three spouts.

The output of the three spouts are received by the parser-tweet-bolt(s). The 3 parser-tweet-bolts break the tuple into words **(**Filtering out the hash tags, RT, @ and urls) and sends the words to the count bolt.

The 2 count bolts keep a running count of each word received from the parser-bolt and updates the information in the postgres database (if a new word is received an insertion is done instead of update).

The topology of this application is present in the EC2_STEPS/tweetwordcount/topologies directory.