**twitterStreamingtoBigQuery-Code Documentation**

**Real-time sentiment analysis of tweets in GCP**

**This module will stream tweets based on a specified # into a Pub/Sub topic. An Apache Beam pipeline will then conduct sentiment analysis on these tweets and stream into a Big Query table. From here it is up to the user to use a visualisation tool of their choice to power a live dashboard. The Beam job will run on Dataflow. Much of the workflow has been automated but there are some configuration tasks that the user needs to complete before running the programs. The only file in the source code the user need to intact with is config.py. This guide was written assuming the user has a basic understanding of the Google Cloud Platform suite of products and can navigate the web console.**

**Contents**

**Config.py – configuration file for twitter-2-pubsub.py and stream-tweets-dataflow.py E**

**listofcities.py – twitter-2-pubsub dependency DNE**

**requirements.py – packages used by stream-tweets-dataflow.py DNE**

**requirements.txt – Dataflow worker dependency DNE**

**twitter-setup.py – packages used by twitter-2-pubsub.py DNE**

**twitter-2-pubsub – source code for connecting to twitter API and streaming into pubsub DNE**

**stream-tweets-dataflow – source code for apache beam job to run on dataflow DNE**

**DNE = DO NOT EDIT**

**E = EDIT**

**Step 1 setting config.py variables**

These variables are pretty self-explanatory but I will give a short description for each

PROJECT\_ID – this is the project ID you with run the application in please ensure you have the appropriate IAM access. If using a service account, ensure you have been assigned the role of admin on Pub/Sub and dataflow. example ‘warm-airline-2323’

TOPIC\_NAME = this is name of the Pub/Sub topic that will be the ingestion point of the tweets. Even if you have already created this topic please fill in this variable as the program will ask you to verify this. Example ‘twitter-stream’

TRACK\_TERMS - this is the # you wish to follow. You can provide a list of #. Example [‘trump’, ‘clinton’]

STREAM\_TIME – this is the amount of time you want to run the stream for in seconds

Twitter credentials much be generated from a Twitter account.

Please follow the guide in this link to get your twitter credentials

<https://themepacific.com/how-to-generate-api-key-consumer-token-access-key-for-twitter-oauth/994/>

STAGING – this is the location in your bucket that will be used to store files while running the dataflow job. Must be in the form 'gs://<BUCKET\_NAME>/staging'

TEMP – this is the location in your bucket that will be used to store temporary files during the dataflow job Must be in the form 'gs://<BUCKET\_NAME>//temp’

**STEP 2 – Create the big query table**

Please create a big query table with the following column names and schema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Username** | STRING | NULLABLE | Describe this field... |
| **Tweet** | STRING | NULLABLE | Describe this field... |
| **Time** | TIMESTAMP | NULLABLE | Describe this field... |
| **Followers** | INTEGER | NULLABLE | Describe this field... |
| **Location** | STRING | NULLABLE | Describe this field... |
| **Source** | STRING | NULLABLE | Describe this field... |
| **Sentiment** | FLOAT | NULLABLE | Describe this field... |

**STEP 3 – Set up virtual environment for twitter-2-pubsub.py**

Twitter-2-pubsub.py has some python 3 dependencies and at the time of writing the default interpreter used in cloud shell and compute engine is python 2.7.x

So we need to set up a VE to run the streamer in.

Please run the following commands in order one by one

sudo apt-get install -y python3-pip

sudo apt-get install build-essential libssl-dev libffi-dev python3-dev

sudo apt-get install -y python3-venv

pyvenv my\_env\_for\_streaming

source my\_env\_for\_streaming/bin/activate

sudo apt-get install -y git

clone your repo using - git clone '<THIS REPO URL>'

**Step 4 running twitter-2-pubsub.py**

It is recommended to run twitter-2-pubsub.py in your cloud shell as the shell comes pre-installed with many packages.

First we need to run the set-up file. Ensure you are inside the correct directory.

Python twitter-setup.py

Then run the program

Python twitter-2-pubsub.py

You will receive a prompt to confirm if the topic you specified has been created already. If reply ‘no’ this topic will be created for you. Please verify this by navigating to the Pub/Sub section in your web console and searching for the topic under the tab Topics.

The stream is now running and is being ingested into your Pub/Sub topic!

**Step 5 running the Dataflow pipeline**

For stream-tweets-dataflow.py we do not need a virtual environment.

It is recommended to run this in shell for testing and in a micro-VM for prolonged use. Open a new shell environment by clicking the + sign and ‘add new shell session’.

Once inside the new session we need to run the requirements.py file

Python requirements.py

Next we need to run the stream-tweets-dataflow.py. There are 3 required arguments this program needs to run:

Requirements\_file – is the requiremtns.txt file that come with this distribution that is used by the dataflow worker to locally install any decencies that the Beam pipeline uses.

Input\_topic – is the topic that the Beam job ingests the stream from. It must be given in the form projects/<PROJECT\_ID>/topics/<TOPIC\_NAME>

e.g projects/warm-airline-207713/topics/twitter-stream

Output\_table - is the big query table that the Beam job will write to. It must be given in the form <PROJECT\_ID>:<DATASET>.<TABLE\_NAME>

e.g warm-airline-207713:Tweets\_raw.Donald\_Trump\_Tweets

Now we run the program

python stream-tweets-dataflow.py --requirements\_file requirements.txt \--input\_topic projects/warm-airline-207713/topics/twitter-stream \--output\_table warm-airline-207713:Tweets\_raw.Donald\_Trump\_Tweets\_D

You should now be able to see a dataflow streaming job running in the Dataflow tab on the web console. Please wait around 1 min for the job to auto scale. Please check tweets are arriving in your Big query table by selecting an appropriate query. E.g:

SELECT count(time) as records FROM [warm-airline-207713:Tweets\_raw.Donald\_Trump\_Tweets\_D@-3600000-]

This will give you the last hour worth of data.

Please note that the dataflow job will only ingest the tweets that are streamed after the job was created so if your tweets stream has stopped you need to re-run it.

Also ensure you stop the dataflow job when you have finished.