**Installation and Setup instructions**:

**Setting up tweepy**:

1. To stream real-time Twitter data, we use the tweepy API and to listen to tweets. To use tweepy, we need to obtain the Twitter Developer authentication token.
2. To get that follow the instructions in the below link

<https://www.dataquest.io/blog/streaming-data-python/>

1. The content present in “Setting up Tweepy” helps us to get the access token.

**Setting up Kafka**:

1. First, we need to setup Kafka, as our VCL instance already provides an image consisting of Kafka in it we have used that instance.

2) To Start zookeeper and Kafka, execute below commands

**$KAFKA\_HOME/bin/zookeeper-server-start.sh $KAFKA\_HOME/config/zookeeper.properties**

**$KAFKA\_HOME/bin/kafka-server-start.sh $KAFKA\_HOME/config/server.properties**

* **We need to make sure that they are running in separate terminals**

3) Then we need to create a topic “**twitterPost**” in Kafka using the below command

**$KAFKA\_HOME/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 -partitions 1 --topic twitterPost**

* The above command should be executed in a new terminal

**Setting up Elasticsearch**:

1. We can set up on Ubuntu systems by following the below link

<https://tecadmin.net/setup-elasticsearch-on-ubuntu/>

1. We need to install elasticsearch for python so we need to run the following command

**sudo -H python3 -m pip install elasticsearch==7.0.0**

1. Create a new index “**twitterpost**” in elasticsearch using the below curl command

**curl -X PUT "localhost:9200/twitterpost”**

**Setting up Logstash**:

1. For setting up we need to download the Debian(DEB) installation file from the below link

<https://www.elastic.co/downloads/logstash>

1. Further steps on how to install can be found in the below YouTube video

<https://www.youtube.com/watch?v=xHVxlm63VaU>

1. Then we need to modify the configuration file according to our requirement. We have uploaded a configuration file “**logstash-kafka.conf**” just download the file and place it in the path **“/etc/logstash/**
2. Then traverse to the path **“/usr/share/logstash/bin”** and run the below command

**sudo ./logstash -f /etc/logstash/logstash-kafka.conf**

1. You should see Logstash running in the terminal if it shows any FATAL errors kill any previous process running logstash and re-execute the above command.

**Installing Dependencies:**

We need to install the required libraries for running the application, all the libraries are present in **requirements.txt**, go to the folder where this file is present and execute the following command

**sudo pip install -r requirements.txt**

**Setting up a Flask application**:

We have used Flask to interact with front-end UI with the help of REST API’s. Then flask application transforms this request into an elasticsearch query and interacts with the indices in it.

So we need to start the flask server on our VCL instance by the following commands

Copy the **app.py** from Codes **folder** in the drive to your vcl instance.

Make the file executable using the following command: chmod a+x app.py

Now, make sure your 5000 port is free and run the following command to start the server.

**./app.py**

**Hosting our front-end UI: ( Local machine )**

First, we need to map the flask app port on the VCL instance to localhost port. Use the following command to do that:

ssh -N -L 5000:localhost:5000 unityid@RemoteIPAddress

We need to host our front-end application in our local server or any other external server like XAMPP or Tomcat. We can see the UI populating and the user can use any functionalities present.

If you are using XAMPP, unzip the folder in the following location: xampp/htdocs/

Then go to your web browser and access the following address: http://localhost

**Streaming Data:**

To start the twitter streaming, download the **tweets.py** and **twitter.txt** from the Codes folder of the drive.

Replace the key values in **twitter.txt** with the appropriate values of your **Twitter developer account**.

Run the **tweets.py** to start the twitter streaming. It takes care of pushing the tweets to elastic search after preprocessing.

**python tweets.py**

**P.S.**: Please follow the steps in the given order.

**Dataset**:

Here is the list of stop-words used in developing our application.

[a, about, above, after, again, against, all, am, an, and, any, are, arent, as, at, be, because, been, before, being, below, between, both, but, by, cant, cannot, could, couldnt, did, didnt, do, does, doesnt, doing, dont, down, during, each, few, for, from, further, had, hadnt, has, hasnt, have, havent, having, he, hed, hell, hes, her, here, heres, hers, herself, him, himself, his, how, hows, i, id, ill, im, ive, if, in, into, is, isnt, it, its, its, itself, lets, me, more, most, mustnt, my, myself, no, nor, not, of, off, on, once, only, or, other, ought, our, ours, ourselves, out, over, own, same, shant, she, shed, shell, shes, should, shouldnt, so, some, such, than, that, thats, the, their, theirs, them, themselves, then, there, theres, these, they, theyd, theyll, theyre, theyve, this, those, through, to, too, under, until, up, very, was, wasnt, we, wed, well, were, weve, were, werent, what, whats, when, whens, where, wheres, which, while, who, whos, whom, why, whys, with, wont, would, wouldnt, yo, yod, yoll, yore, yove, your, yours, yourself, yourselves, ,t,m,http,https,amp,nt,https,get,got,will,can,don]

**Screenshots**:

We have included a folder named Screenshots which contain pictures of our application in execution.

**References**:

1. <https://www.dataquest.io/blog/streaming-data-python/>
2. <https://medium.com/datascape/twitter-real-time-streaming-api-use-with-python-tweet-mining-3b04a52f18d8>
3. <https://www.geeksforgeeks.org/bloom-filters-introduction-and-python-implementation/>
4. <https://elasticsearch-py.readthedocs.io/en/master/index.html>
5. [https://pyprobables.readthedocs.io/en/latest/index.html#](https://pyprobables.readthedocs.io/en/latest/index.html)
6. <https://pyprobables.readthedocs.io/en/latest/code.html#bloomfilter>
7. <https://pyprobables.readthedocs.io/en/latest/code.html#heavyhitters>
8. <https://www.geeksforgeeks.org/removing-stop-words-nltk-python/>
9. <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-term-query.html>
10. <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-range-query.html>
11. <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-prefix-query.html>
12. <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-terms-set-query.html>
13. <https://www.elastic.co/guide/en/elasticsearch/reference/6.1/_create_an_index.html>
14. <https://discuss.elastic.co/t/elasticsearch-query-with-timestamp-range/97469>