#### The following framework is implemented using Apache Spark Streaming, Kafka, Elasticsearch and Kibana. It performs real-time sentiment analysis of particular hash tags in twitter data.

#### Sentiment analysis for all the tweets for #trump, #obama and show them (e.g., positive, neutral, negative, etc. tweets) on a map. When we show tweets on a map, we plot them using their latitude and longitude.

Sentiment Analyzer

Spark Streaming

Consumer.py

Visualizer

Elasticsearch

Tweet scrapper

tweets.py

Kafka

Kibana

Sentiment analysis framework

1. Scrapper(tweets.py):

The scrapper will collect all tweets and sends them to Kafka for analytics. The scraper will be a standalone program and will perform the followings:

1. Collect tweets in real-time with particular hash tags. (**#trump**, **#obama**)
2. After getting tweets we will filter them to check if location field is present or not.
3. After filtering, we will send them (tweets) to Kafka.
4. Kafka API (producer) is used. (<https://kafka.apache.org/090/documentation.html#producerapi>)
5. Scrapper program will run infinitely and should take hash tag as input parameter while running.
6. Kafka:

You need to install Kafka and run Kafka Server with Zookeeper. You should create a dedicated channel/topic for data transport.

(<https://kafka.apache.org/quickstart)>

1. Sentiment Analyzer:

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or attitude of a speaker.

For example,

“President Donald Trump approaches his first big test this week from a position of unusual weakness.” - has positive sentiment.

“Trump has the lowest standing in public opinion of any new president in modern history.” - has neutral sentiment.

“Trump has displayed little interest in the policy itself, casting it as a thankless chore to be done before getting to tax-cut legislation he values more.” - has negative sentiment.

You can use any third party sentiment analyzer like Stanford CoreNLP (java/scala), nltk(python) for sentiment analyzing. For example, you can add Stanford CoreNLP as an external library using SBT/Maven in your scala/java project. In python you can import nltk by installing it using pip.

Sentiment analysis using Spark Streaming:

In Spark Streaming, create a Kafka consumer and periodically collect filtered tweets from scrapper. For each hash tag, perform sentiment analysis using Sentiment Analyzing tool (discussed above). Then for each hash tag, send the output to Elasticsearch for visualization.

1. Visualizer:

Install Elasticsearch and Kibana. Create an index for visualization. Create a map to show all kinds of tweets. After that, create a dashboard to show the map. Dashboard requires data to be time stamped. So, when you send data from Spark to Elasticsearch add a time stamp. In the dashboard set a Map refresh time to 2 min as an example.

ELK tutorial:

<https://www.oreilly.com/learning/a-guide-to-elasticsearch-5-and-the-elkelastic-stack>

<https://www.digitalocean.com/community/tutorials/how-to-use-kibana-dashboards-and-visualizations>