Relayr Test Solution

PYTHON TEST

Environment:

Language Used	Python
Interpreter Version	3.6
IDE	PyCharm
IDE Version	Professional 2017.3
External Dependencies	None

Python Testing Frameworks:

Unittest is used for testing the following program. The benefits of using this framework are:

- Part of the Python standard library
- Flexible test case execution
- Promotes grouping related tests into common test suites
- Very fast test collection
- Very precise test duration reports

However other frameworks could also be used are listed below:

- Robot
- PyTest
- DocTest
- Nose2
- Testify

Assumptions:

- No duplicates would be allowed to be part of the corpus.
- Only strings are added to the corpus.
- A string added to the corpus should not exceed a specific size.

BIG DATA DEEP DIVE

Deliverables:

- Block Diagram
- System Architecture Diagram

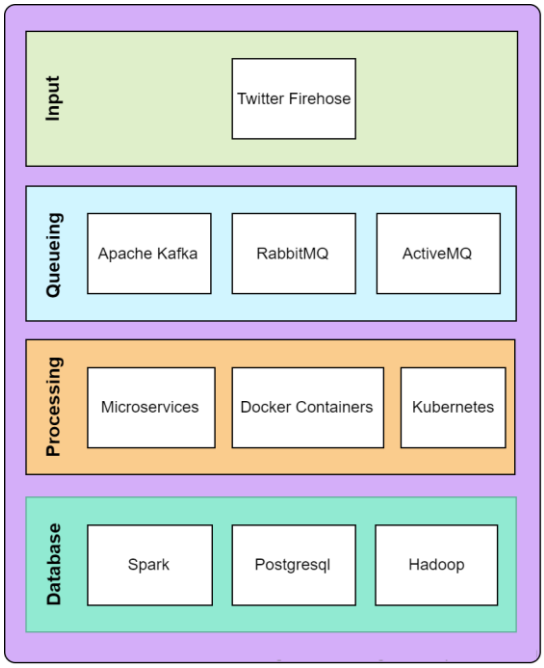
Summary of the Architecture and Technologies Used:

- **Input**: Thinking of our architecture, we are getting a huge amount of data from twitter firehose which is somewhat 6k tweets per second. There is SDK available for that in Python.
- **Queuing System:** To handle this chunk of data, we have to use some **message queue system** in our architecture. There are many message queuing systems available like **RabbitMQ, ActiveMQ**, and **Apache-Kafka**.
  As far as our use case is concerned, we can efficiently use **Apache-Kafka** which will behave as a message queue mechanism for handling the number of incoming tweets. After maintaining a queue, data is being passed on to the processing system for further processing.

- **Processing System:** For routing tweets to the correct model, our processing system might use the **micro-service architecture** efficiently in a **containerized** way to handle the different sentiment analysis features. For making the processing system we can use multiple REST frameworks of Python, we can simply use **Python-Flask** for that purpose.

- **Database/Storage System:** In the end, for storing the data to DB, we can use **PostgreSQL** as our storage system, but horizontally scaling it has few limitations.
  On the other hand, the data we are storing is not much relational, so we can also use the **NoSQL** database with our system like **MongoDB**, as it will be relatively easy to scale it horizontally.

Block Diagram



System Architecture Diagram