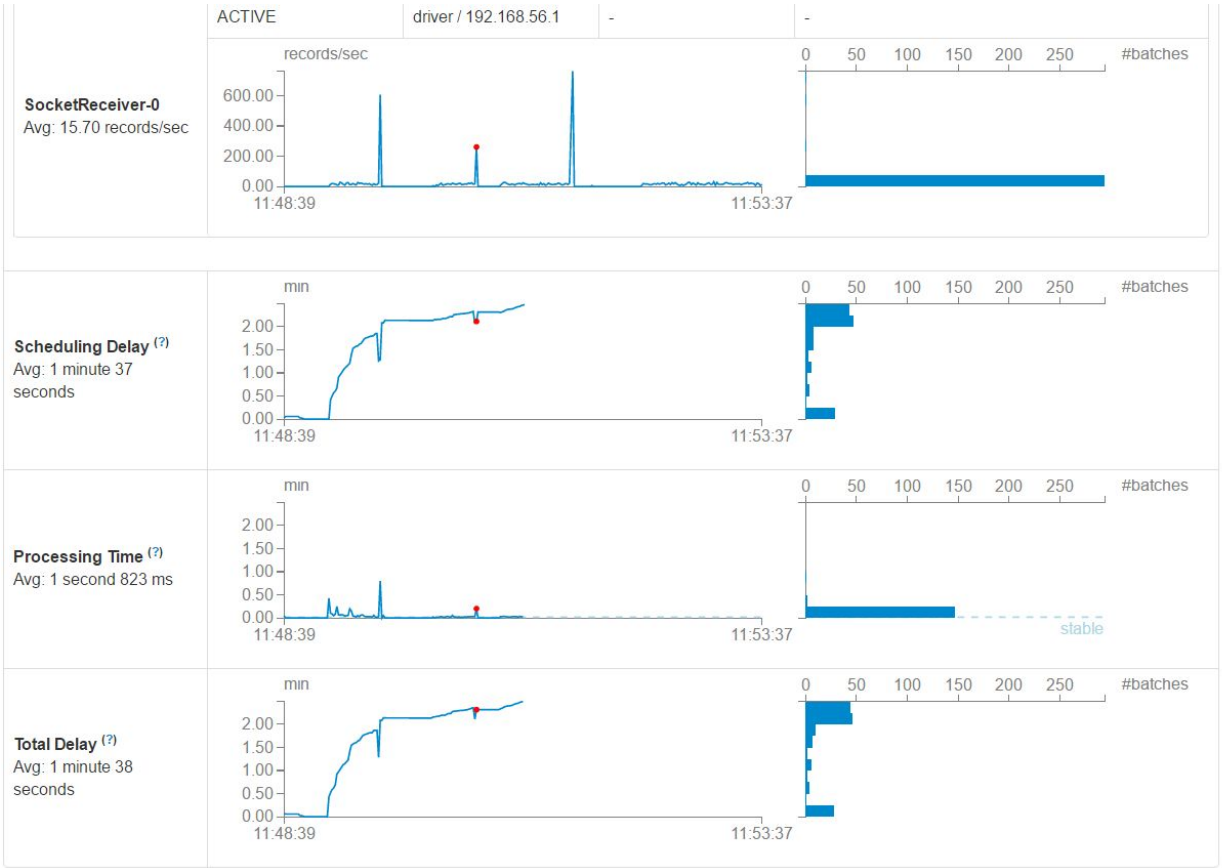
Spark is an open source cluster based framework by Apache used mainly to deal with Big data. Now, Spark Streaming is a subset or an extension of Spark or even a subset of Spark which is used to deal with live data.
In our project, we use Spark for doing analysis or pre-processing data, etc. The data collected using Twitter Streaming API is pulled by Spark using Socket Text Streaming. This data is then processed, analysis is done, etc. converted. Results are then converted into JSON objects and passed onto Elasticsearch over to Kibana.
Spark is very powerful tool which is used to do extremely complex type of analysis. These kinds of analysis, sometimes, yield results that are not transformable to JSON. Since elasticsearch holds all the data, even the past data. There is no concept of windowing in Kibana. Hence it uses all the historical data. We can also write their own search query to limit current data.

One of the most important thing in Spark is taking care of how efficiently we use it. Now, we can run any number of algorithms in any order, misuse space complexity as well and get analysis done. But that affects Spark usage. This needs to taken into account and design should be made accordingly. One can monitor this on Spark UI.



First graph shows amount of data coming in batches. Now, most of the time Spark Streaming struggles with efficiency because of amount of data coming in. This can be avoided by setting a threshold to amount of data coming in. We capped that threshold to 10000 tweets. This means Spark will parallely try to process 10000 tweets. Now, if we increase the batch to more than that

our performance suffers. We can be sure that our methodology used is going good by checking other 3 graphs.