

# **SOCIAL MEDIA SENTIMENT ANALYSIS**

**REALTIME TWITTER ANALYSIS**

**DEEPA. A**

**DEPT OF CSE,EPCET, 2018-19**

---

---

## USECASE

Twitter is a micro-blogging website that has become increasingly popular with the network community. Users update short messages, also known as Tweets, which are limited to 140 characters. Users update their personal opinions on many subjects, discuss current topics and write about life events through tweets. This platform is favoured by many users because it has no political and economic restrictions and is easily available to large number of people. As the number of users increase, micro-blogging platforms are becoming a place to find strong viewpoints and sentiment. People use twitter to forecast and analyse in a lot of different areas.

Sentiment analysis deals with identifying and classifying opinions or sentiments which are present in source text. Social media is generating a huge amount of sentiment rich data in the form of tweets, status updates, reviews and blog posts etc. Sentiment analysis of this user generated data is very useful in knowing the opinion of the crowd.

# DIFFRENCIATOR:

In this project, Twitter Tweets are analyzed Realtime than the Historical Based.

While most previous analysis uses batch processing of social media data, but now days there is a potential need of real time data processing.

This analytics allows an organization the ability to take immediate action for those times when acting within seconds or minutes is significant. The goal is to obtain the insight required to act prudently at the right time - which increasingly means immediately.

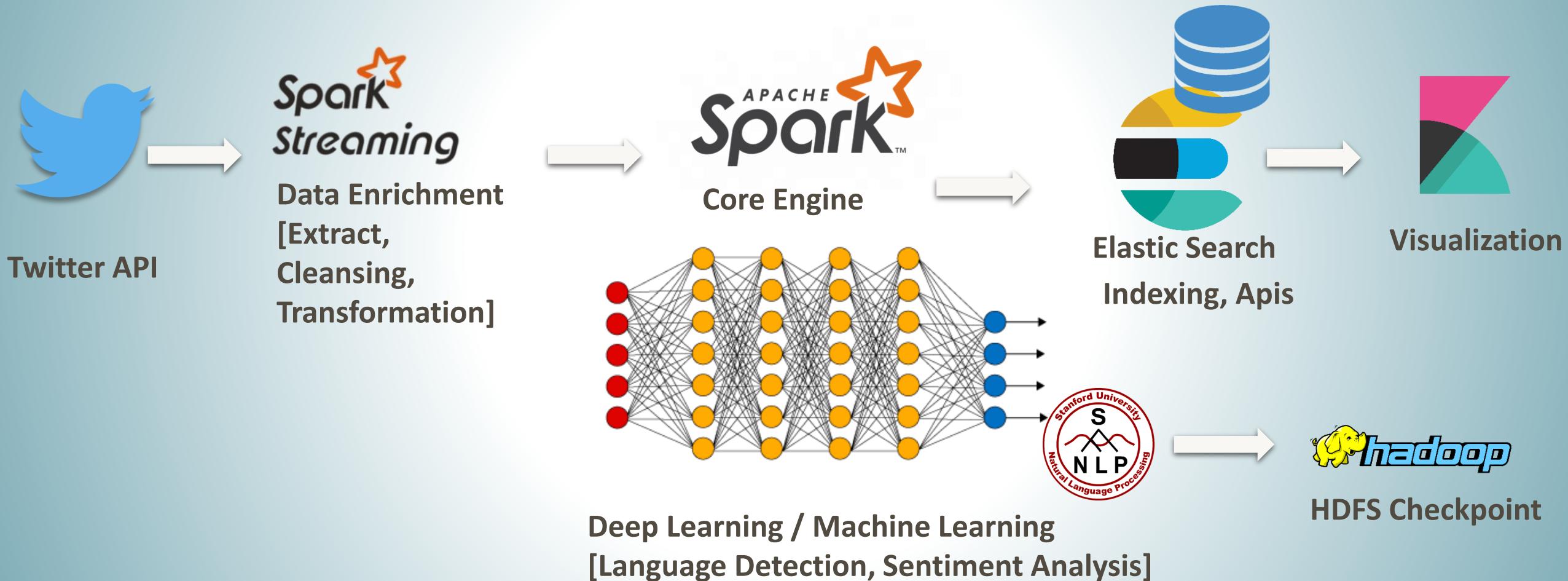
Complex event processing (CEP) combines data from multiple sources to detect patterns and attempt to identify either opportunities or threats. The goal is to identify significant events and respond fast. Sales leads, orders or customer service calls are examples.

Operational Intelligence (OI) uses real time data processing and CEP to gain insight into operations by running query analysis against live feeds and event data such as tweets. OI is near real time analytics over operational data and provides visibility over social media data. The goal is to obtain near real time insight using continuous analytics to allow the organization to take immediate action.

Batch data processing is an efficient way of processing high volumes of data is where a group of transactions is collected over a period of time. Data is collected, entered, processed and then the batch results are produced.

In contrast, real time data processing involves a continual input, process and output of data. Data must be processed in a small time period (or near real time). Radar systems, customer services and bank ATMs are examples.

# APPROACH





# TWITTER

# TWITTER APIs

The screenshot shows the Twitter Developer API documentation website at <https://developer.twitter.com/en/docs.html>. The page has a purple header with navigation links for Developer, Use cases, Products, Docs, More, and Labs. On the right side of the header are buttons for Apply, Apps, a search icon, and a user profile icon. The main content area features a "Stay Informed" section with a "Learn how >" button. Below it are sections for "Search Tweets", "Filter realtime Tweets", "Account Activity API", "Direct Message API", "Twitter for websites", and "Ads API", each with a "Learn more" button. A sidebar on the left lists categories: Tweets, Direct Messages, Media, and Trends. The "Tweets" category is expanded, showing sub-sections like Post, retrieve and engage with Tweets; Get Tweet timelines; Curate a collection of Tweets; Optimize Tweets with Cards; Search Tweets; Filter realtime Tweets; Sample realtime Tweets; Get batch historical Tweets; Rules and filtering; Data enrichments; Tweet objects; Tweet compliance; and Tweet updates.

## Stay Informed

Staying informed about changes to our APIs is important for those developing on the platform and can be critical to maintaining your applications. We have a number of channels to help you stay in-the-loop.

[Learn how >](#)

### Search Tweets

Use the Search API to find historical Tweets. Free to enterprise versions available.

[Learn more](#)

### Filter realtime Tweets

Get only the Tweets you need by using advanced filtering tools with the realtime streaming API.

[Learn more](#)

### Account Activity API

Have 15+ account activities delivered to you in realtime via a webhook connection.

[Learn more](#)

### Direct Message API

Build personalized customer experiences with our Direct Message platform.

[Learn more](#)

### Twitter for websites

### Ads API

### Tweets

Post, retrieve and engage with Tweets

Get Tweet timelines

Curate a collection of Tweets

Optimize Tweets with Cards

Search Tweets

Filter realtime Tweets

Sample realtime Tweets

Get batch historical Tweets

Rules and filtering

Data enrichments

Tweet objects

Tweet compliance

Tweet updates

### Direct Messages

### Media

### Trends

# TWITTER DEVELOPER ACCOUNT

Get access to the Twitter API

#welcome

We're excited you want to use Twitter APIs and data!

As a developer platform, our first responsibility is to our users: to provide a place that supports the health of conversation on Twitter.

This application process helps us to:

1. Prevent abuse of the Twitter platform.
2. Better understand and serve our developer community.

Thank you for your time and thoughtful responses.  
Applications are final once submitted and can't be edited.

What is your primary reason for using Twitter developer tools?

We'll help you on your path to getting the most out of Twitter APIs and data.

Professional	Hobbyist	Academic	Other
...for commercial uses	...for a personal project	...for education or research	I don't fit any of those
Building B2B products	Making a bot	Doing academic research	Embedding Tweets on a website
Building consumer products	Building tools for Twitter users	Teaching	Doing something else
Publishing ads programmatically	Exploring the API	Student	

Next

# TWITTER DEVELOPER ACCOUNT - REVIEW

The screenshot shows a Gmail inbox with a search bar at the top containing "twitter". A list of emails is displayed, with the first one being an application review from "developer-accounts <developer-accounts@twitter.com>". The email subject is "Twitter developer account application [ ref:\_0ODAOKOA8.\_5004A1iDyr9:ref ]". The message body starts with "Hello," and "Thanks for your interest in building on Twitter." It continues by asking for more details about the use case and the types of information needed for review. At the bottom, it encourages replying with these details and expresses appreciation for help.

Gmail

Compose

Inbox 9,124

Starred

Snoozed

Important

Sent

Drafts 1

deepa

anjali Subramanian There is no need to introduce the love

Nithyanantha Babu Sundaram

Twitter developer account application [ ref:\_0ODAOKOA8.\_5004A1iDyr9:ref ]

developer-accounts <developer-accounts@twitter.com>  
to me

Fri, Jun 21, 1:24 AM (2 days ago)

Hello,

Thanks for your interest in building on Twitter.

Before we can finish our review of your developer account application, we need some more details about your use case.

The types of information that are valuable for our review include:

- *The core use case, intent, or business purpose for your use of the Twitter APIs.*
- *If you intend to analyze Tweets, Twitter users, or their content, share details about the analyses you plan to conduct, and the methods or techniques.*
- *If your use involves Tweeting, Retweeting, or liking content, share how you'll interact with Twitter accounts, or their content.*
- *If you'll display Twitter content off of Twitter, explain how, and where, Tweets and Twitter content will be displayed with your product or service, including whether Tweets and Twitter content will be displayed at row level, or aggregated.*

Just reply to this email with these details. Once we've received your response, we'll continue our review. We appreciate your help!

Thanks

# TWITTER DEVELOPER ACCOUNT - REVIEW

The screenshot shows the 'Keys and tokens' section of a Twitter developer account. The URL is https://developer.twitter.com/en/apps/12422614. The page has a purple header with links for Developer, Use cases, Products, Docs, More, and Labs. On the right, there are 'Apply' and 'Apps' buttons and a user profile icon. The main content area has tabs for App details, Keys and tokens (which is selected), and Permissions. The 'Keys and tokens' section contains sections for Consumer API keys and Access token & access token secret, each with regenerate and revoke buttons.

**Keys and tokens**  
Keys, secret keys and access tokens management.

**Consumer API keys**  
GWT4av6gSsQqqqtCnNfhQw6Wm (API key)  
Nncna7TBqmylNuKAjbGOh5Y1637wwj1VciWiSUmpyUiUbKJQjZ (API secret key)  
[Regenerate](#)

**Access token & access token secret**  
887145308-BqFurnbkue3NGzEwUUmAEokxgdF4NWGsqEuzHiem (Access token)  
FZYvC42i6zQ3DdoGFwcrbdXKXI1CiUzr15yw7S9oQbjTe (Access token secret)  
Read, write, and Direct Messages (Access level)  
[Revoke](#) [Regenerate](#)

[Developer policy and terms](#) [Follow @twitterdev](#) [Subscribe to developer news](#)

**About**  
Let's go Twitter Company

**Business**  
About Twitter Ads Targeting

**Developers**  
Documentation Forums

**Help Center**  
Using Twitter Managing your account

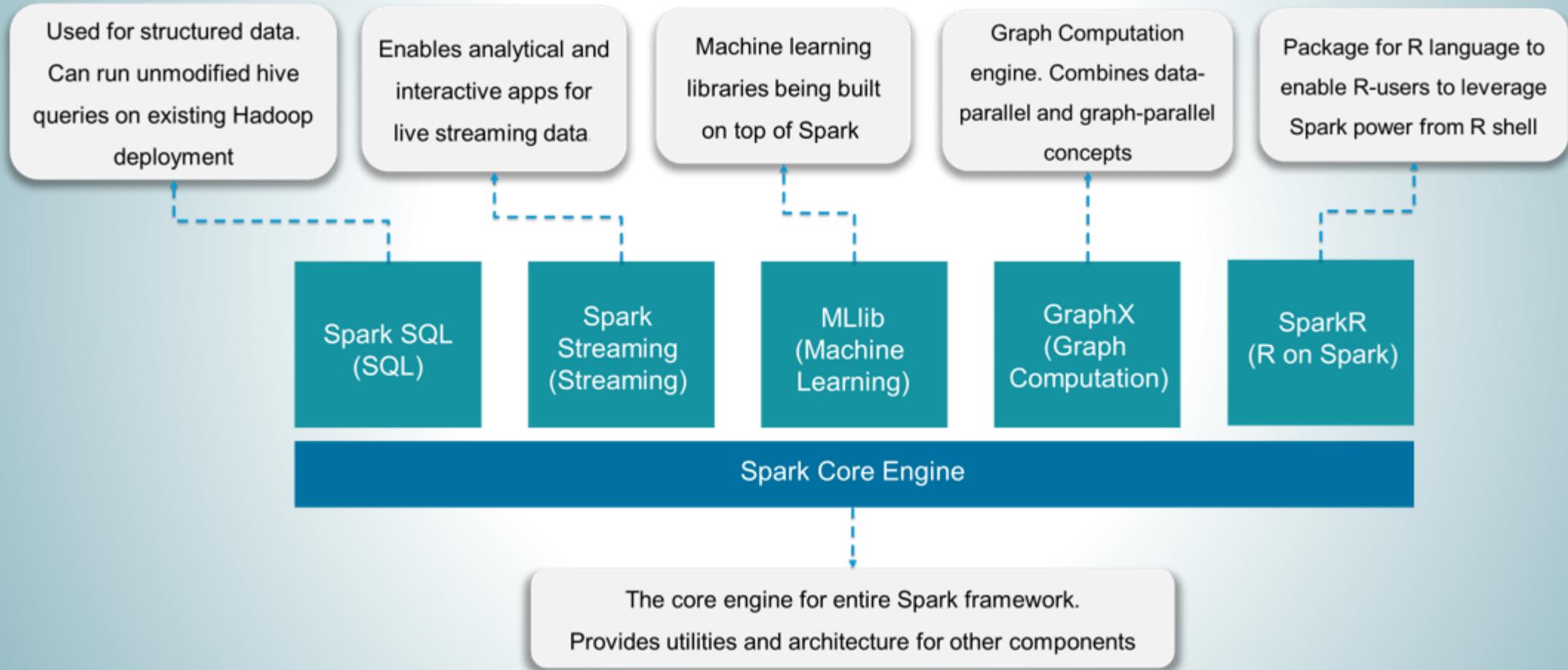
**Marketing**  
Insights Success Stories



# SPARK STREAMING

# SPARK

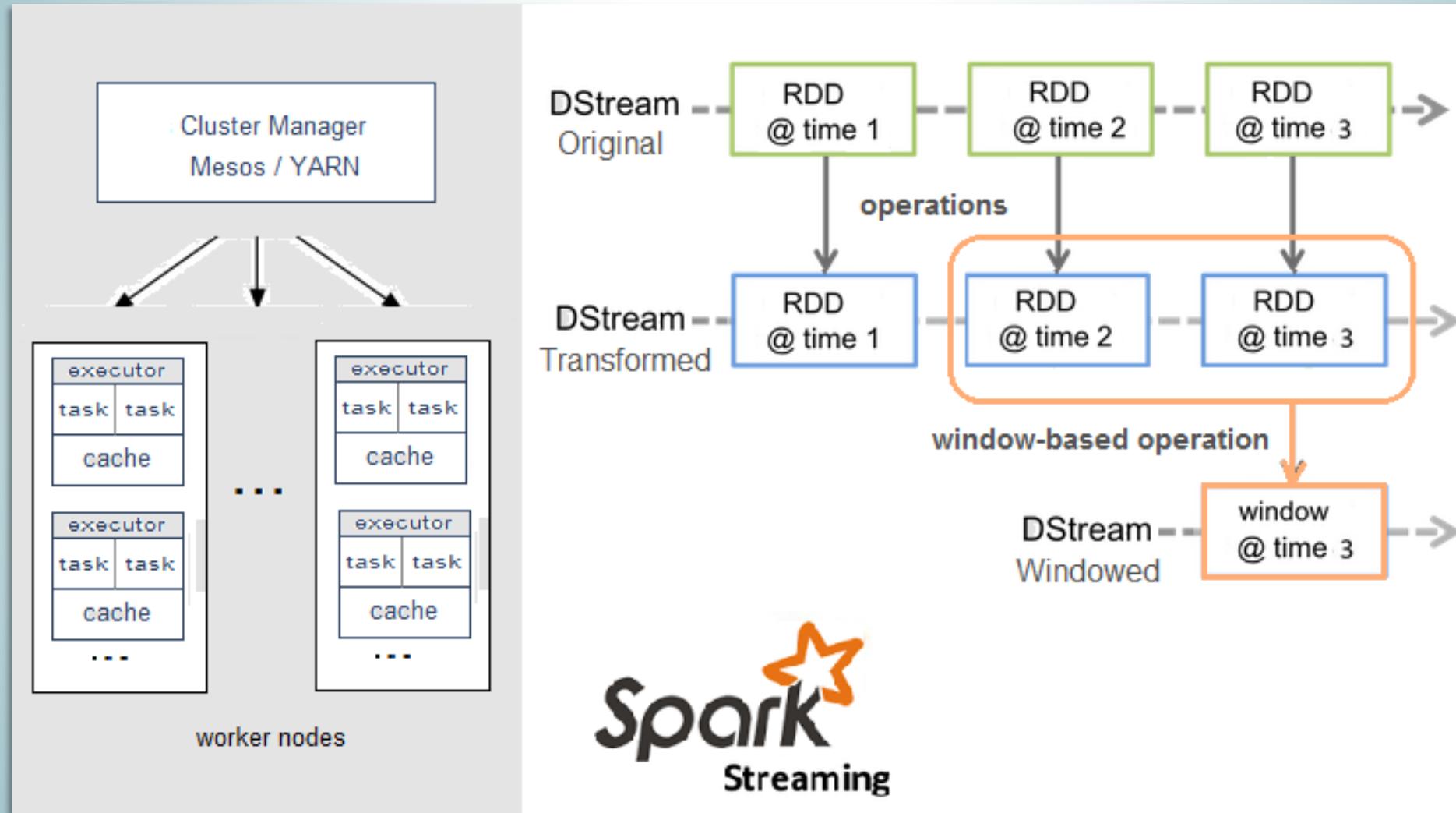
- Apache Spark™ is a unified analytics engine for large-scale data processing.



# WHY SPARK ?

Apache Spark	Hadoop MapReduce
Parallel Processing Framework	Parallel Processing Framework
Real-time processing	Batch Processing
Stores data in-memory	Read and writes from disk
Up to 100x faster in memory [OR] 10x faster on disk than MapReduce	Fast, but slower than spark
Written in Scala	Written in Java
Real-time or batch processing for big data analysis	Batch Processing for big data analysis
Cost is higher due to RAM Storage requirement	Cost is lower due to disk memory

# SPARK STREAMING



# STANFORD NLP



- Stanford CoreNLP provides a set of natural language analysis ML Framework written in Java.
- It can take raw human language text input and give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize and interpret dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases or word dependencies, and indicate which noun phrases refer to the same entities.
- It was originally developed for English, but now also provides varying levels of support for (Modern Standard) Arabic, (mainland) Chinese, French, German, and Spanish.
- Stanford CoreNLP is an integrated framework, which make it very easy to apply a bunch of language analysis tools to a piece of text. Starting from plain text, you can run all the tools with just two lines of code.
- Its analyses provide the foundational building blocks for higher-level and domain-specific text understanding applications.
- Stanford CoreNLP is a set of stable and well-tested natural language processing tools, widely used by various groups in academia, industry, and government. The tools variously use rule-based, probabilistic machine learning, and deep learning components.



# ELASTIC SEARCH

# ELASTIC SEARCH

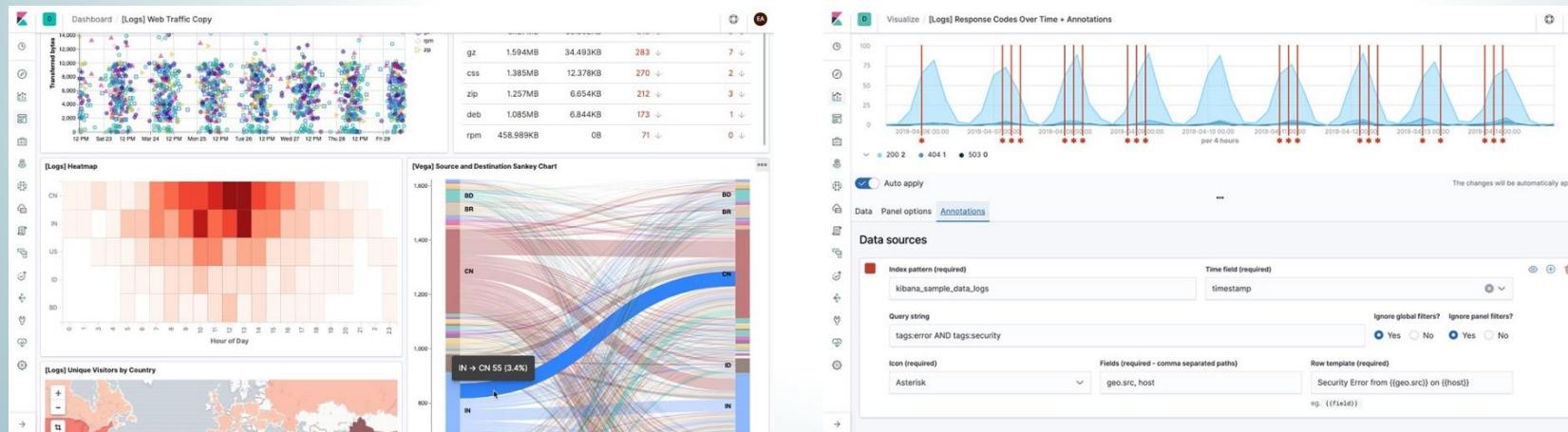
- Elasticsearch is an open-source, RESTful, distributed search and analytics engine built on Apache Lucene.
- It is commonly used for log analytics, full-text search, security intelligence, business analytics, and operational intelligence use cases.
- Elasticsearch combines the speed of search instances with the power of analytics for better decision making. It gives insights that make your business streamlined and improves your products by interactive search and other analyzing features.
- ElasticSearch is a schema-less database that has powerful search capabilities and is easy to scale horizontally.



# KIBANA

# KIBANA

- Kibana lets you visualize our Elasticsearch Analysis data and navigate the entire transformed modeled data.
- Kibana core ships with the classics: histograms, line graphs, pie charts, sunbursts, and more. All leverage the full aggregation capabilities of Elasticsearch.
- Perform advanced time series analysis on your Elasticsearch data with our curated time series UIs. Describe queries, transformations, and visualizations with powerful, easy-to-learn expressions.



<>

# CODE WALKTHROUGH

# LIBRARIES USED

```
libraryDependencies ++= {  
    Seq(  
        "org.apache.spark" %% "spark-core" % "2.4.3",  
        "org.apache.spark" %% "spark-sql" % "2.4.3",  
        "org.apache.spark" %% "spark-streaming" % "2.4.3",  
        "org.apache.bahir" %% "spark-streaming-twitter" % "2.3.3",  
        "edu.stanford.nlp" % "stanford-corenlp" % "3.5.1",  
        "edu.stanford.nlp" % "stanford-corenlp" % "3.5.1" classifier "models",  
        "org.elasticsearch" %% "elasticsearch-spark-20" % "7.1.1"  
    )  
}
```

# TWITTER -> SPARK STREAMING

```
System.setProperty("twitter4j.oauth.consumerKey", "GWT4av6gSsQqqqtCnNfhQw6Wm")
System.setProperty("twitter4j.oauth.consumerSecret", "Nncna7TBqmyINuKAjbGOh5Y1637wwj1VcIWiSUmpyUiUbKJQjZ")
System.setProperty("twitter4j.oauth.accessToken", "887145308-BqFurnbkue3NGzEwUUmAEokxgdF4NWGsqEuzHiem")
System.setProperty("twitter4j.oauth.accessTokenSecret", "FZYvC42i6zQ3DdoGFwcrbdXKXI1CiUZr15yw759oQbJTe")
val sparkConf = new SparkConf().setAppName("twitterSentiment").setMaster("local[2]")
val ssc = new StreamingContext(sparkConf, Seconds(5))
val stream = TwitterUtils.createStream(ssc, None, filters)
```

# NLP SENTIMENT ANALYSER - CODE WALKTHROUGH

```
def detectSentiment(message: String): SENTIMENT_TYPE = {  
  
    val pipeline = new StanfordCoreNLP(nlpProps)  
  
    val annotation = pipeline.process(message)  
    var sentiments: ListBuffer[Double] = ListBuffer()  
    var sizes: ListBuffer[Int] = ListBuffer()  
  
    var longest = 0  
    var mainSentiment = 0  
  
    for (sentence <- annotation.get(classOf[CoreAnnotations.SentencesAnnotation])) {  
        val tree = sentence.get(classOf[SentimentCoreAnnotations.AnnotatedTree])  
        val sentiment = RNNCoreAnnotations.getPredictedClass(tree)  
        val partText = sentence.toString  
  
        if (partText.length() > longest) {  
            mainSentiment = sentiment  
            longest = partText.length()  
        }  
  
        sentiments += sentiment.toDouble  
        sizes += partText.length  
  
        println("debug: " + sentiment)  
        println("size: " + partText.length)  
  
    }  
}
```

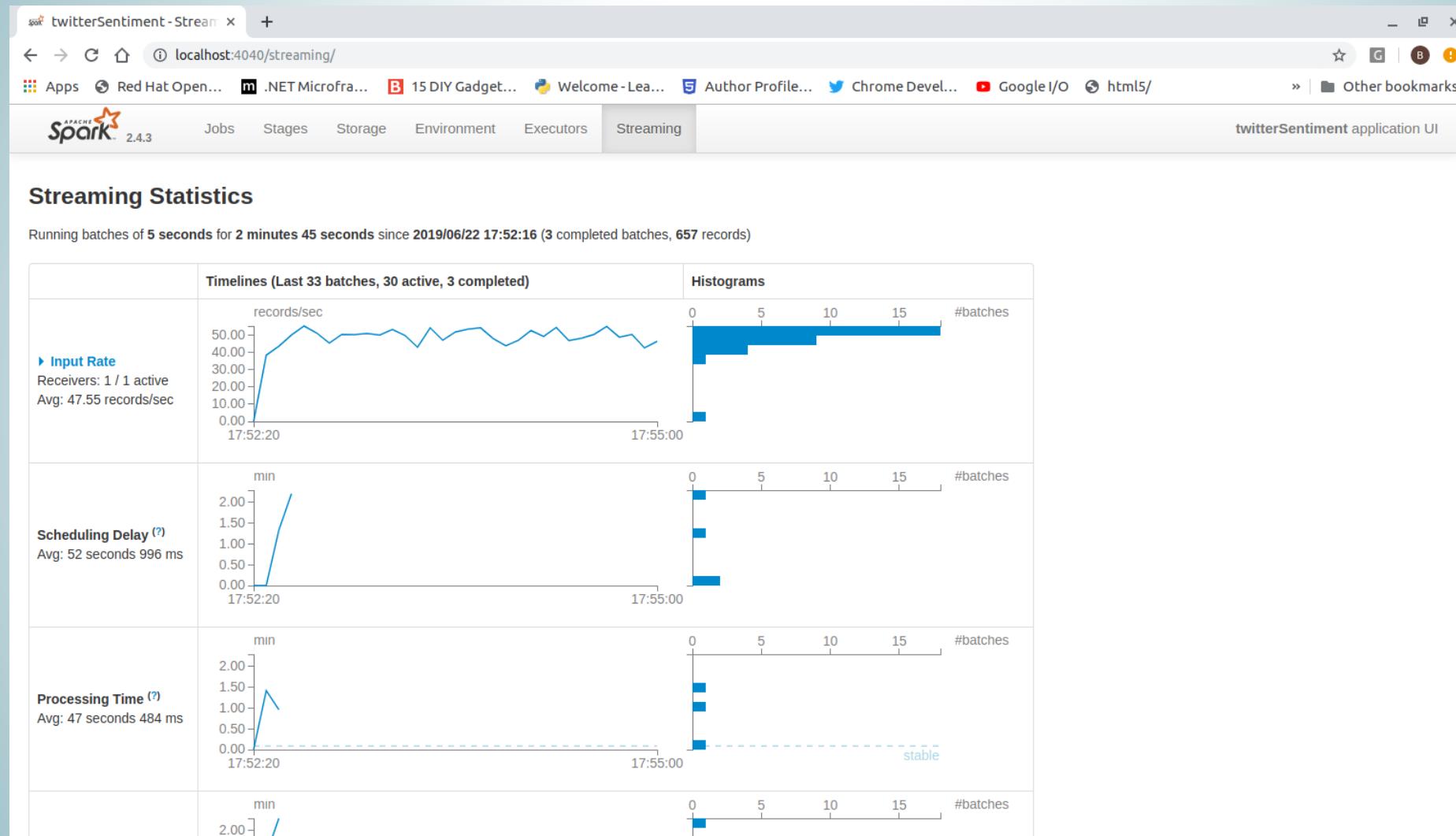
```
    weightedSentiment match {  
        case s if s <= 0.0 => NOT_UNDERSTOOD  
        case s if s < 1.0 => VERY_NEGATIVE  
        case s if s < 2.0 => NEGATIVE  
        case s if s < 3.0 => NEUTRAL  
        case s if s < 4.0 => POSITIVE  
        case s if s < 5.0 => VERY_POSITIVE  
        case s if s > 5.0 => NOT_UNDERSTOOD  
    }
```

# LANGUAGE DETECTION / SEND TO ELASTIC SEARCH

```
stream.foreachRDD { (rdd, time) =>
  rdd.map(t => {
    Map(
      "user" -> t.getUser.getScreenName,
      "created_at" -> t.getCreatedAt.toInstant.toString,
      "location" -> Option(t.getGeoLocation).map(geo => {
        s"${geo.getLatitude},${geo.getLongitude}")
      }),
      "text" -> t.getText,
      "hashtags" -> t.getHashtagEntities.map(_.getText),
      "retweet" -> t.getRetweetCount,
      "language" -> detectLanguage(t.getText),
      "sentiment" -> com.twitter.stream.util.SentimentAnalysisUtils.detectSentiment(t.getText).toString
    )
  }).saveToEs("twitter/tweet")
}
```



# SPARK STREAMING EXECUTION



# SPARK STREAMING EXECUTION

twitterSentiment - Spark UI

localhost:4040/jobs/

APACHE Spark 2.4.3

Jobs Stages Storage Environment Executors Streaming

twitterSentiment application UI

## Spark Jobs (?)

User: deepa  
Total Uptime: 3.4 min  
Scheduling Mode: FIFO  
Active Jobs: 2  
Completed Jobs: 6

Event Timeline

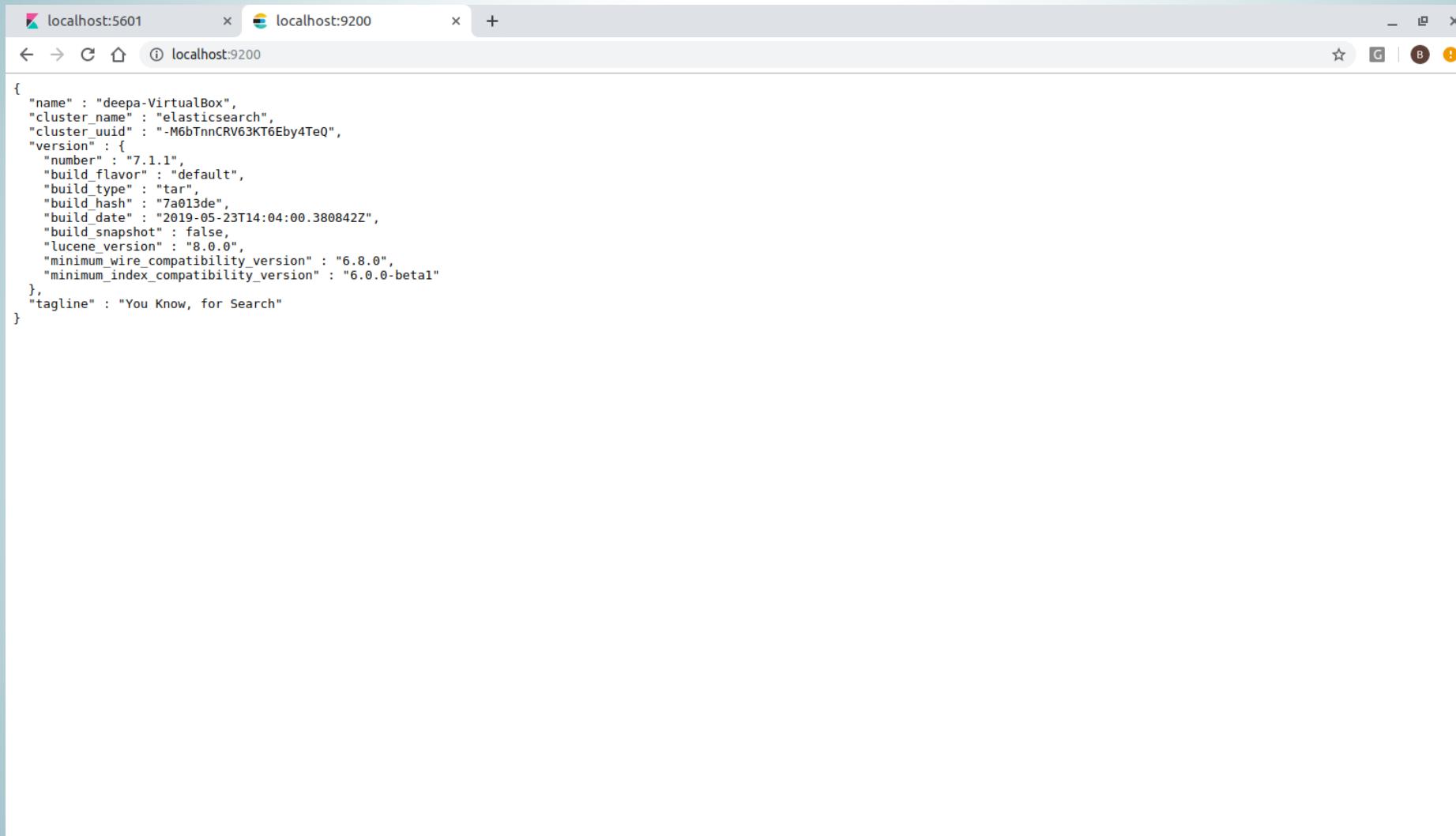
Active Jobs (2)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
7	Streaming job from [output operation 1, batch time 17:52:35] runJob at EsSpark.scala:108 (kill)	2019/06/22 17:54:47	45 s	0/1	8/13 (2 running)
0	Streaming job running receiver 0 start at analytics.scala:84 (kill)	2019/06/22 17:52:16	3.3 min	0/1	0/1 (1 running)

Completed Jobs (6)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
6	Streaming job from [output operation 0, batch time 17:52:35] print at analytics.scala:65	2019/06/22 17:54:47	63 ms	1/1	1/1
5	Streaming job from [output operation 1, batch time 17:52:30] runJob at EsSpark.scala:108	2019/06/22 17:53:50	57 s	1/1	14/14
4	Streaming job from [output operation 0, batch time 17:52:30] print at analytics.scala:65	2019/06/22 17:53:50	0.1 s	1/1	1/1
3	Streaming job from [output operation 1, batch time 17:52:25] runJob at EsSpark.scala:108	2019/06/22 17:52:28	1.3 min	1/1	11/11
2	Streaming job from [output operation 0, batch time 17:52:25] print at analytics.scala:65	2019/06/22 17:52:25	1.0 s	1/1	4/4
1	Streaming job from [output operation 0, batch time 17:52:25]	2019/06/22 17:52:25	0.6 s	1/1	1/1

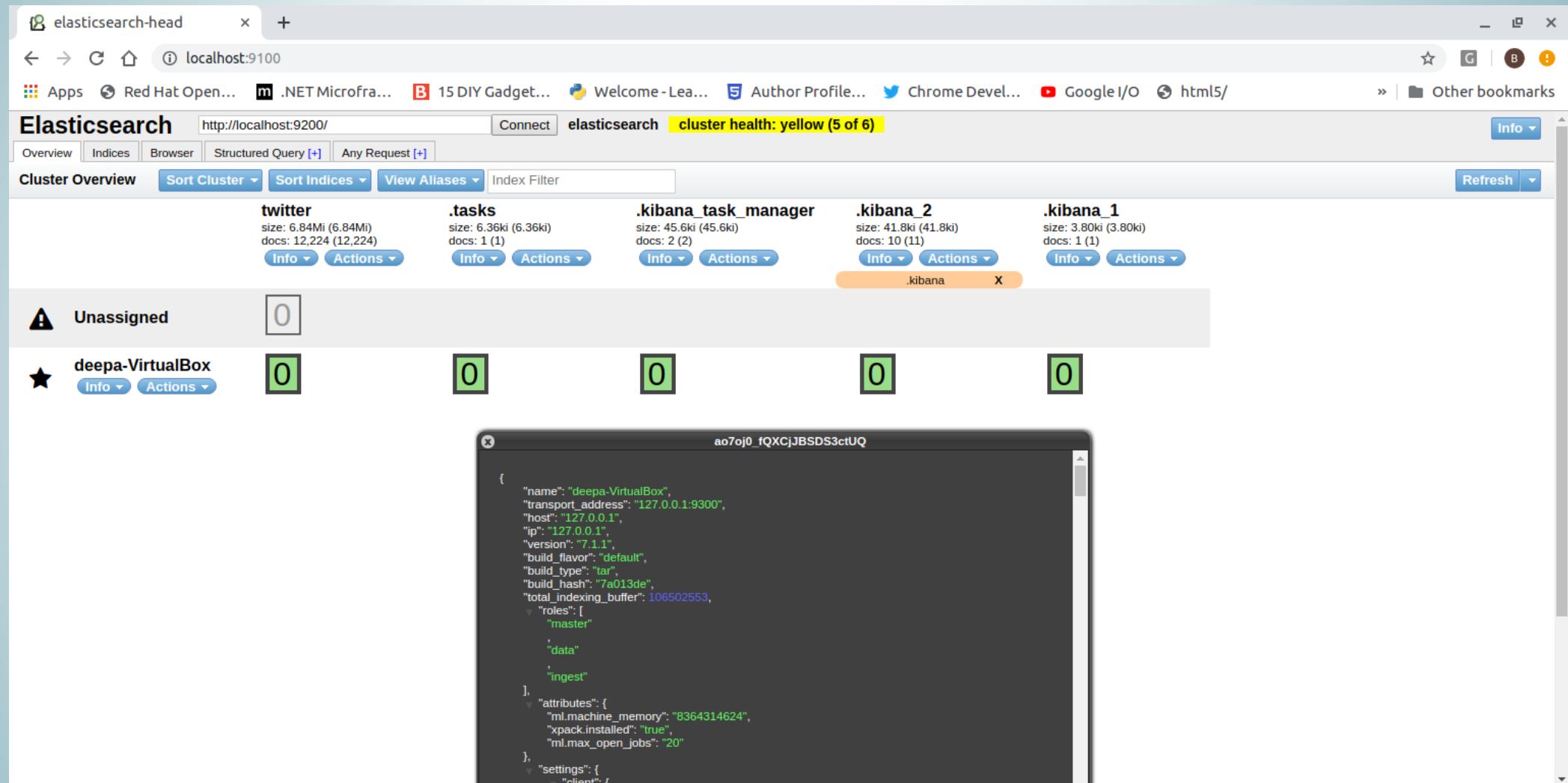
# ELASTIC SEARCH - API



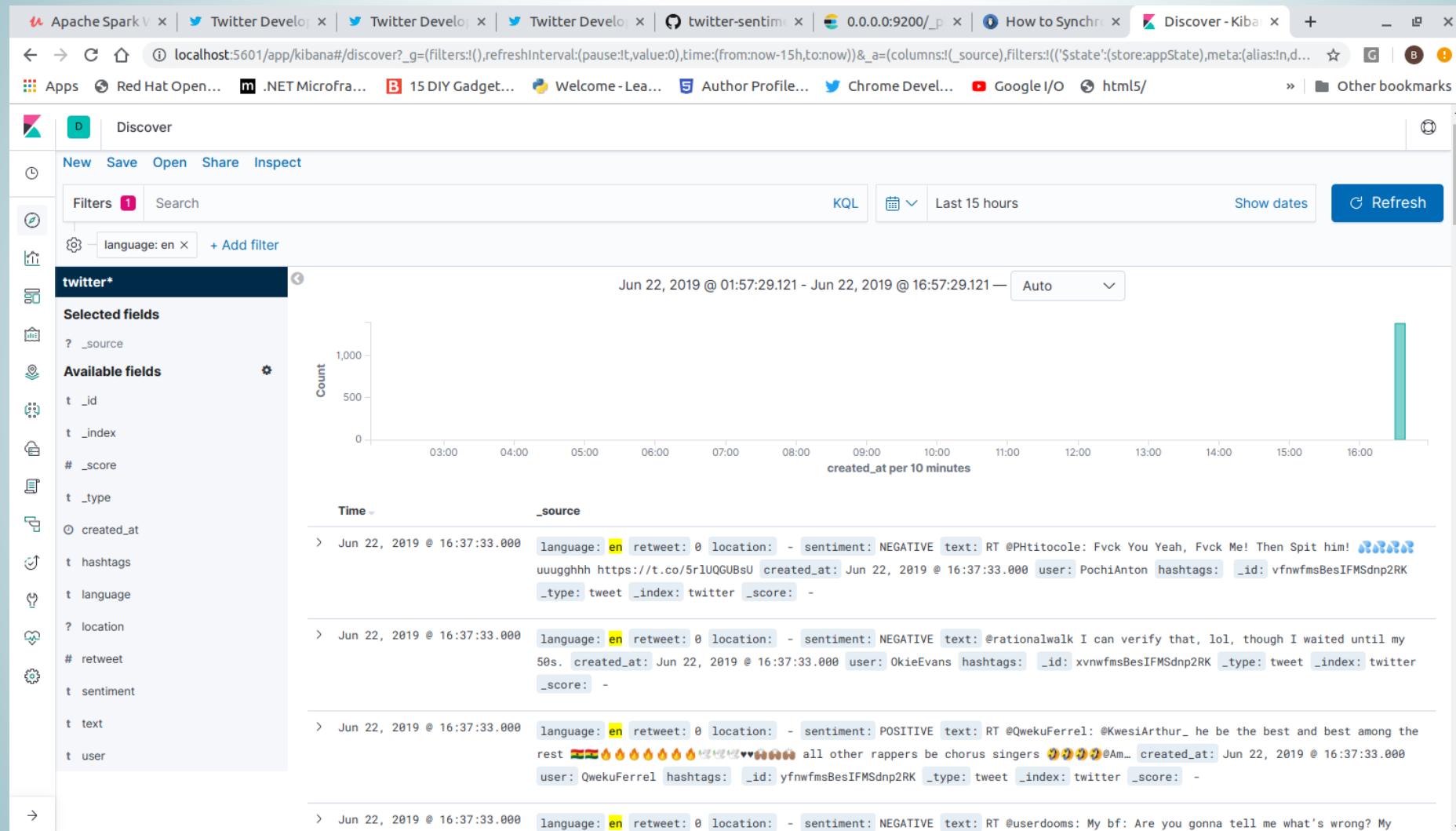
A screenshot of a web browser window displaying the Elasticsearch cluster state API response. The browser has two tabs open: 'localhost:5601' and 'localhost:9200'. The 'localhost:9200' tab shows the JSON output of the API call.

```
{
  "name" : "deepa-VirtualBox",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "-M6bTnnCRV63KT6Eby4TeQ",
  "version" : {
    "number" : "7.1.1",
    "build_flavor" : "default",
    "build_type" : "tar",
    "build_hash" : "7a013de",
    "build_date" : "2019-05-23T14:04:00.380842Z",
    "build_snapshot" : false,
    "lucene_version" : "8.0.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

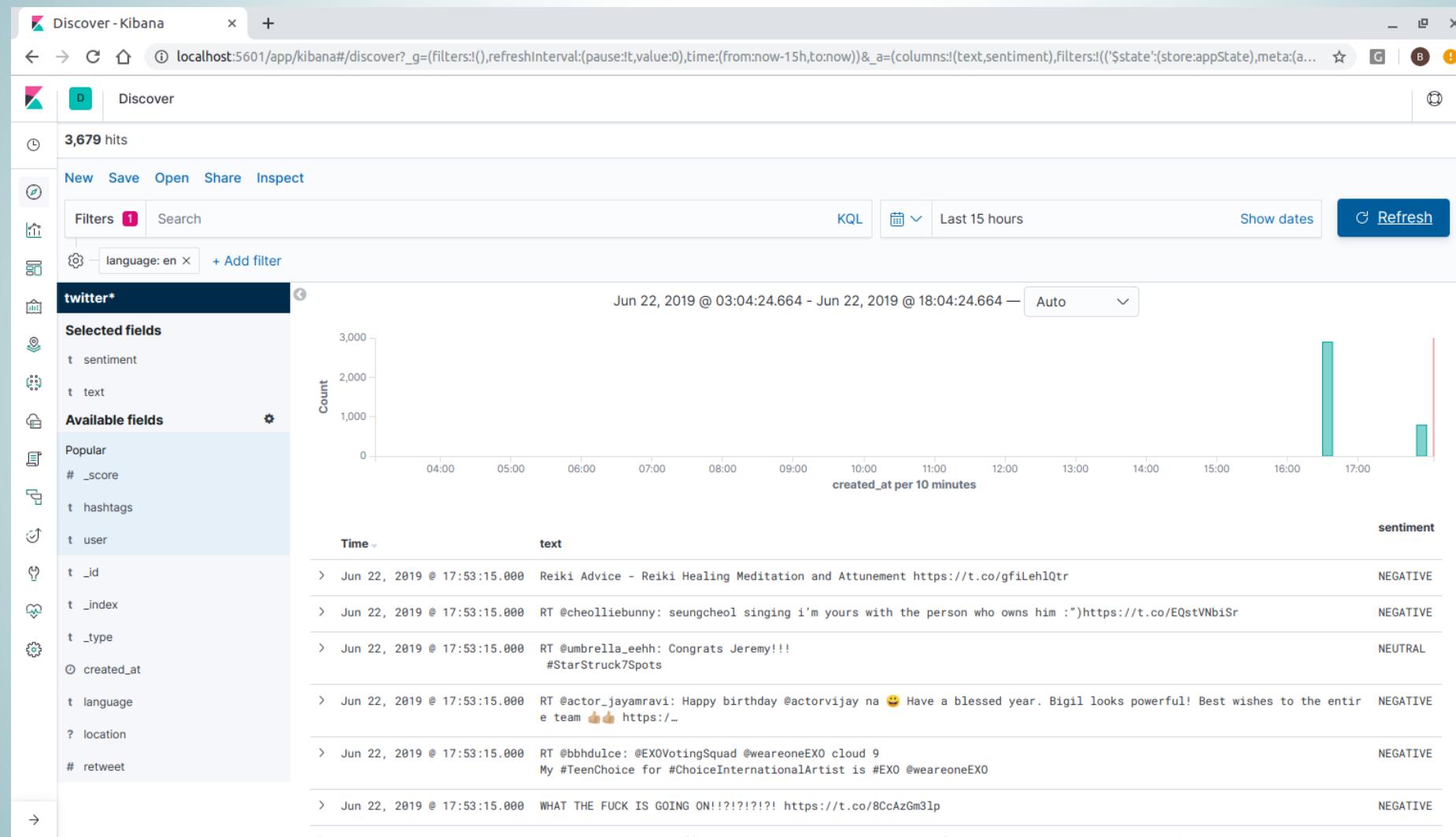
# ELASTIC SEARCH – SENTIMENTS INGESTION



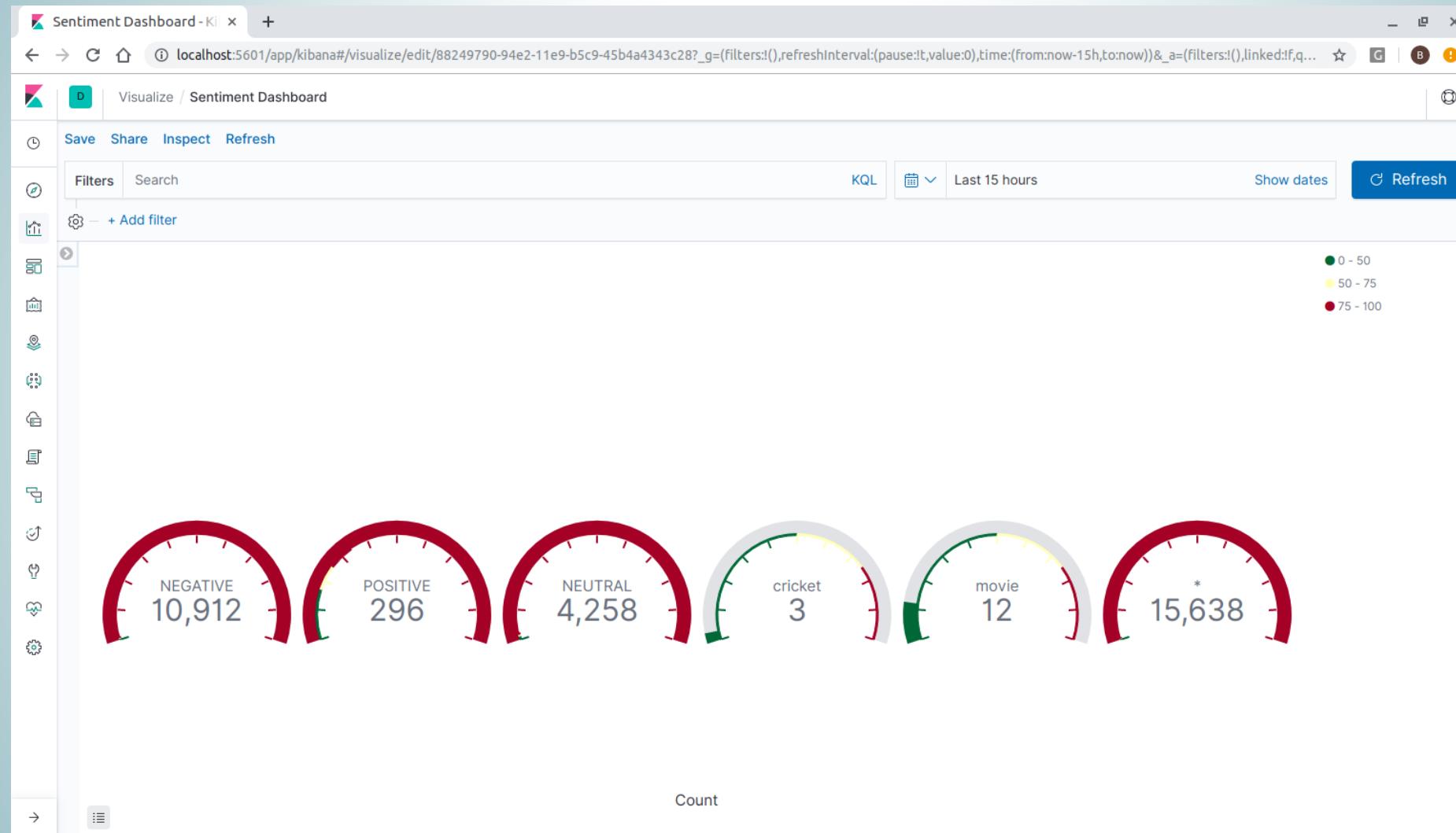
# KIBANA – TWEETS INDEX



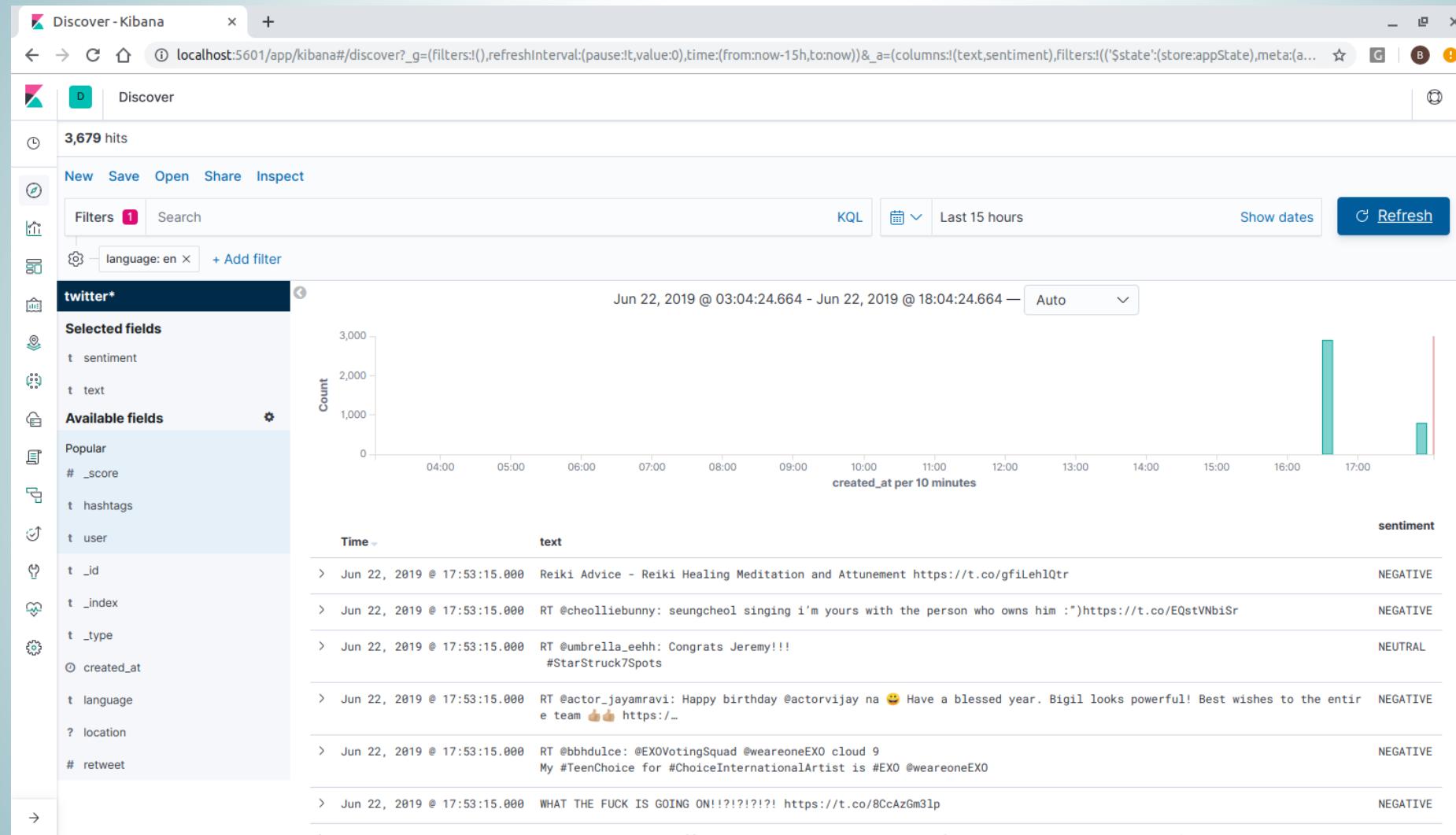
# KIBANA – TWEETS INDEX BY SENTIMENTS



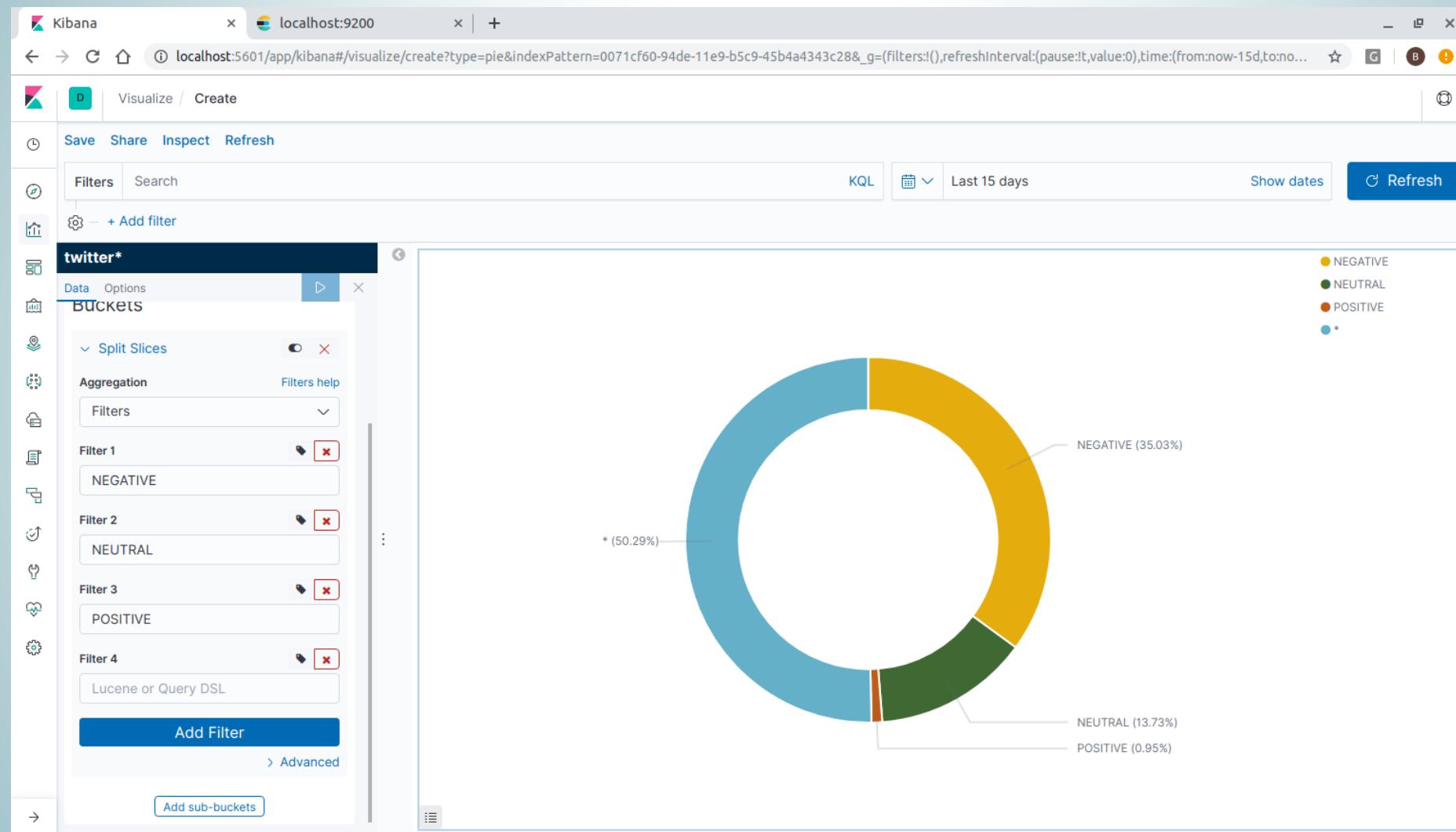
# KIBANA – REALTIME SENTIMENTS – GUAGE VIZUALIZATION



# KIBANA – TWEETS INDEX BY SENTIMENTS



# KIBANA – TWEETS SENTIMENTS BY %



# CONCLUSION

- This project has demonstrated the optimized way of extracting the sentiment information from twitter's social media tweet sources and using it as a general trend predictor.
- While opinion data remain a useful sort of information for determining overall market sentiment, they are often difficult to analyse and, since they are often focused on conveying nuanced information, may contain mixed messages. By leveraging deep learning and neural networks as trained in the project will result in business intelligence to understand the subjective reasons why people are or are not responding to popular events in the market.
- This project will be useful for creating inputs to a model that recognize context, tone, and previous indications of sentiment can help increase accuracy and get a better overall sense of the people opinions on the matter.