



CIS5900 Term Project Tutorial



Authors: [Aakanksha Tasgaonkar](#); [Amogh Mahesh](#); [Louis Tan](#); [Nanjesh Gowda](#); [Tanvi Gidwani](#)

Instructor: [Jongwook Woo](#)

Date: 08/10/2019

Lab Tutorial

Aakanksha Tasgaonkar (atasgao@calstatela.edu)

Amogh Mahesh (amahesh3@calstatela.edu)

Louis Tan (ltan3@calstatela.edu)

Nanjesh Gowda (nmandya@calstatela.edu)

Tanvi Gidwani (tgawade@calstatela.edu)

08/10/2016

Trending YouTube Video Data Analysis using Elasticsearch and Kibana

Objectives

YouTube (the world-famous video sharing website) maintains a list of the top trending videos on the platform. According to Variety magazine, “To determine the year’s top-trending videos, YouTube uses a combination of factors including measuring users’ interactions (number of views, shares, comments and likes). This dataset is a daily record of the top trending YouTube videos. The dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the US, GB, DE, CA, and FR regions (USA, Great Britain, Germany, Canada, and France, respectively), with up to 200 listed trending videos per day. We would analyze this data to get insights into YouTube trending videos, to see

what is common between these videos. Those insights might also be used by people who want to increase popularity of their videos. Goal of this analysis is to include an insight into YouTube statistics such as the trending videos, the most liked/viewed categories, trending YouTube channels based on these categories. This also includes a well-planned and mapped statistical analysis of the data over a given period. We have used cutting edge technology like Elastic Search and Kibana for the visualization of the procured data.

- Download Logstash.exe, Elasticsearch, Kibana on local.
- Download and map the dataset into Elastic Search with appropriate mapping types.
- Configure Logstash.conf file to ingest the dataset in the form of CSV to Elastic Search.
- Verify successful loading at API console of Elasticsearch.
- Define Index Patterns in Kibana.

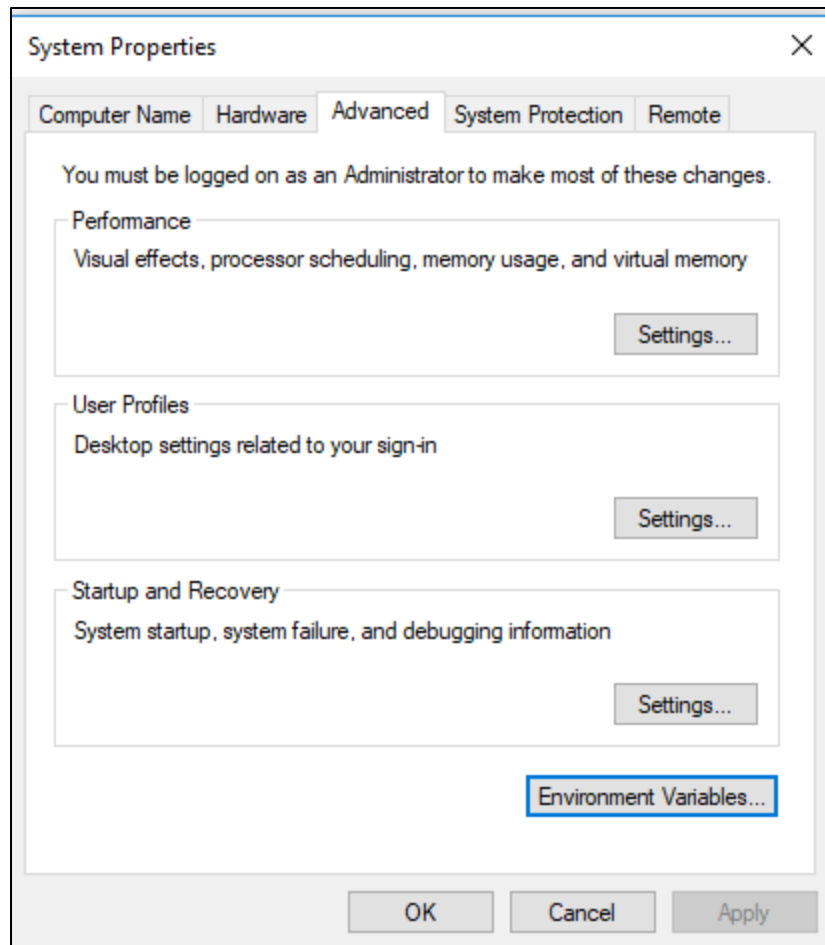
Platform Spec

- Elasticsearch Logstash Kibana
- Server's OS: Windows OS
- Memory Size: Elastic search- 931.5 GB; Kibana- 1.4 GB
- CPU Speed: 1.9 Ghz
- Total Memory Size: Elastic search- 931.5 GB; Kibana- 1.4 GB

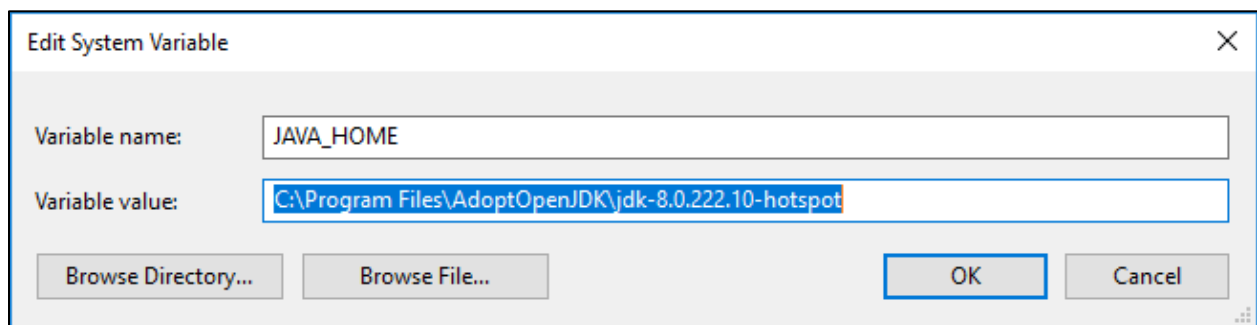
Step 1: Download and Install Java Development Kit (AdoptOpenJDK)

Elasticsearch is built using Java and includes a bundled version of OpenJDK from the JDK maintainers (GPLv2+CE) within each distribution. The bundled JVM is the recommended JVM and is located within the JDK directory of the Elasticsearch home directory.

1. Download JDK kit on local if you don't have one using the below given link:
 - i) <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Once downloaded and installed we need to set the environment variable shown in steps below.
 - i) System Properties → Environment Variables



- ii) New System Variable → Variable Name: JAVA_HOME & variable value : C:\Program Files\AdoptOpenJDK\jdk-8.0.222.10-hotspot (Path to your JDK file)
- Click ok and apply changes



3. Verify JAVA_HOME is set by using command in Windows command line interface:

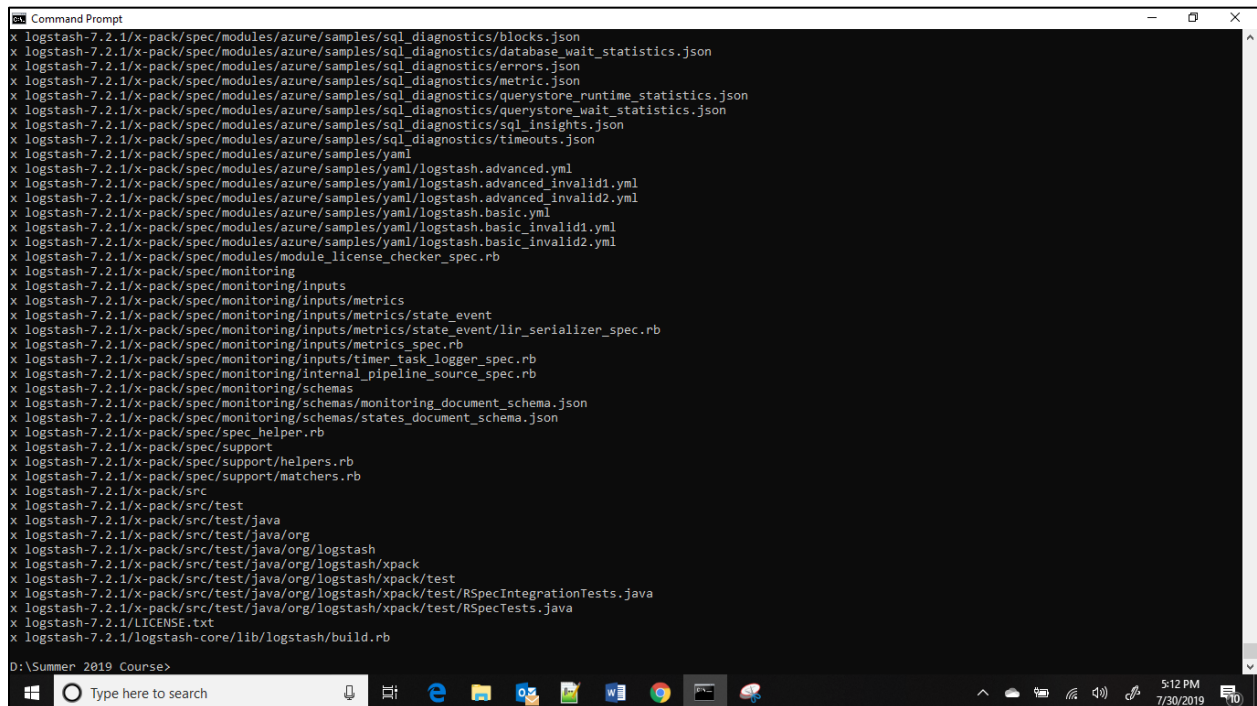
```
echo %JAVA_HOME%
```

```
C:\Users\kunal>echo %JAVA_HOME%  
C:\Program Files\AdoptOpenJDK\jdk-8.0.222.10-hotspot  
C:\Users\kunal>
```

Step 2: Download and Extract ELK Stack on Local

To start with the lab, we need to download and extract Elasticsearch, Logstash and Kibana on local.

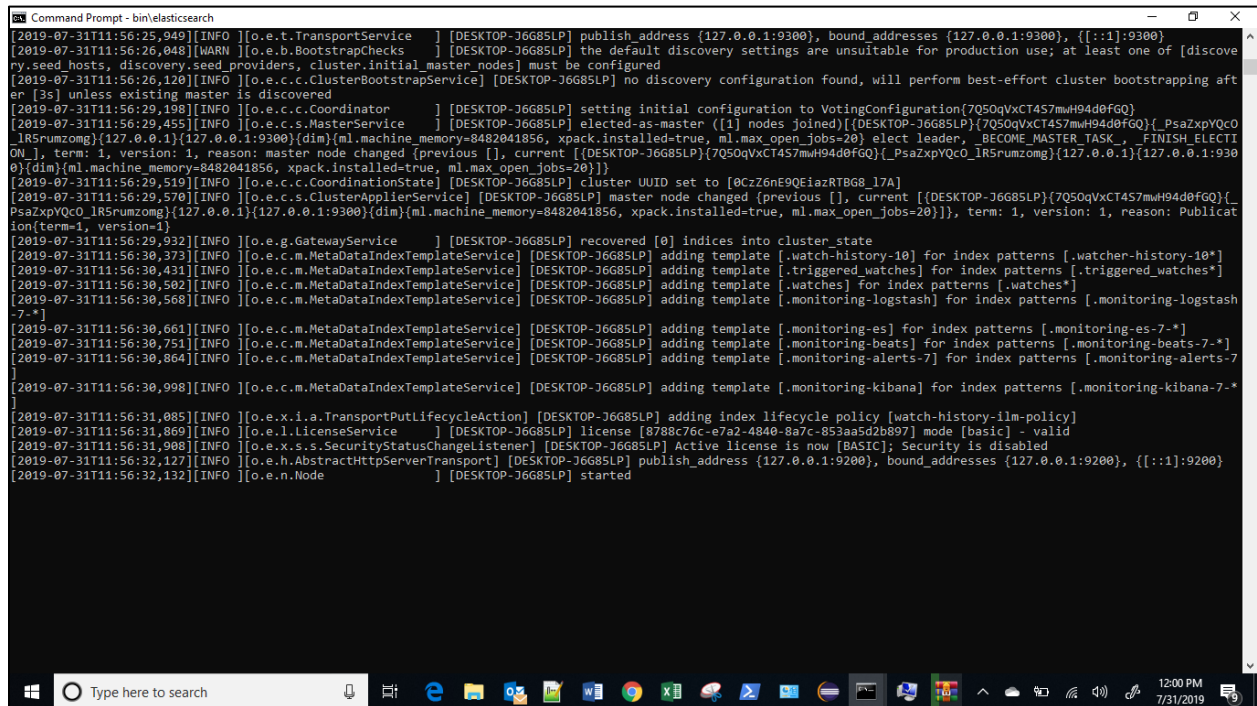
1. Download Elastic search, Logstash, Kibana from the below given links:
 - i) <https://www.elastic.co/downloads/elasticsearch>
 - ii) <https://www.elastic.co/downloads/kibana>
 - iii) <https://www.elastic.co/downloads/logstash>
2. Extract Logstash using the following commands on Windows Command Line Interface.
 - i) Unzip xvf file.gz



```
Command Prompt  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/blocks.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/database_wait_statistics.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/errors.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/metric.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/querystore_runtime_statistics.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/querystore_wait_statistics.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/sql_insights.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/sql_diagnostics/timeouts.json  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml/logstash.advanced.yml  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml/logstash.advanced_invalid1.yml  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml/logstash.advanced_invalid2.yml  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml/logstash.basic.yml  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml/logstash.basic_invalid1.yml  
x logstash-7.2.1/x-pack/spec/modules/azure/samples/yaml/logstash.basic_invalid2.yml  
x logstash-7.2.1/x-pack/spec/modules/module_license_checker_spec.rb  
x logstash-7.2.1/x-pack/spec/monitoring  
x logstash-7.2.1/x-pack/spec/monitoring/inputs  
x logstash-7.2.1/x-pack/spec/monitoring/inputs/metrics  
x logstash-7.2.1/x-pack/spec/monitoring/inputs/metrics/state_event  
x logstash-7.2.1/x-pack/spec/monitoring/inputs/metrics/state_event/lir_serializer_spec.rb  
x logstash-7.2.1/x-pack/spec/monitoring/inputs/metrics_spec.rb  
x logstash-7.2.1/x-pack/spec/monitoring/inputs/timer_task_logger_spec.rb  
x logstash-7.2.1/x-pack/spec/monitoring/internal_pipeline_source_spec.rb  
x logstash-7.2.1/x-pack/spec/monitoring/schemas  
x logstash-7.2.1/x-pack/spec/monitoring/schemas/monitoring_document_schema.json  
x logstash-7.2.1/x-pack/spec/monitoring/schemas/states_document_schema.json  
x logstash-7.2.1/x-pack/spec/spec_helper.rb  
x logstash-7.2.1/x-pack/spec/support  
x logstash-7.2.1/x-pack/spec/support/helpers.rb  
x logstash-7.2.1/x-pack/spec/support/matchers.rb  
x logstash-7.2.1/x-pack/src  
x logstash-7.2.1/x-pack/src/test  
x logstash-7.2.1/x-pack/src/test/java/org  
x logstash-7.2.1/x-pack/src/test/java/org/logstash  
x logstash-7.2.1/x-pack/src/test/java/org/logstash/xpack  
x logstash-7.2.1/x-pack/src/test/java/org/logstash/xpack/test  
x logstash-7.2.1/x-pack/src/test/java/org/logstash/xpack/test/RSpecIntegrationTests.java  
x logstash-7.2.1/x-pack/src/test/java/org/logstash/xpack/test/RSpecTests.java  
x logstash-7.2.1/LICENSE.txt  
x logstash-7.2.1/logstash-core/lib/logstash/build.rb  
D:\Summer 2019 Course>
```

3. Similarly extract Elasticsearch and Kibana using the command line interface.
4. Once the ELK stack is downloaded, we need to start Elastic as well as Kibana in order to ingest the configuration file through Logstash and set up the mappings in Elasticsearch.
5. Initiate Elasticsearch through Windows command line interface by using the following commands:

- i) Go to your elastic folder on local: `cd elasticsearch-7.3.0`
- ii) `bin\elasticsearch`



```
Command Prompt - bin\elasticsearch
[2019-07-31T11:56:25,949][INFO ][o.e.t.TransportService] [DESKTOP-J6G85LP] publish_address {127.0.0.1:9300}, bound_addresses {127.0.0.1:9300}, {[::1]:9300}
[2019-07-31T11:56:26,048][WARN ][o.e.b.BootstrapChecks] [DESKTOP-J6G85LP] the default discovery settings are unsuitable for production use; at least one of [discovery.seed_hosts, discovery.seed_providers, cluster.initial_master_nodes] must be configured
[2019-07-31T11:56:26,120][INFO ][o.e.c.c.ClusterBootstrapService] [DESKTOP-J6G85LP] no discovery configuration found, will perform best-effort cluster bootstrapping after [3s] unless existing master is discovered
[2019-07-31T11:56:29,198][INFO ][o.e.c.c.Coordinator] [DESKTOP-J6G85LP] setting initial configuration to VotingConfiguration{7Q50qVxCT457mmH94d0fGQ}
[2019-07-31T11:56:29,455][INFO ][o.e.c.s.MasterService] [DESKTOP-J6G85LP] elected-as-master ([1] nodes joined)[{DESKTOP-J6G85LP}{7Q50qVxCT457mmH94d0fGQ}{_PsaZxpVQcO_IR5rumzomg}{127.0.0.1}{127.0.0.1:9300}(dim){ml.machine_memory=8482041856, xpack.installed=true, ml.max_open_jobs=20} elect leader, BECOME_MASTER_TASK, FINISH_ELECTION, term: 1, version: 1, reason: master node changed [previous [], current [{DESKTOP-J6G85LP}{7Q50qVxCT457mmH94d0fGQ}{_PsaZxpVQcO_IR5rumzomg}{127.0.0.1}{127.0.0.1:9300}(dim){ml.machine_memory=8482041856, xpack.installed=true, ml.max_open_jobs=20}]]
[2019-07-31T11:56:29,519][INFO ][o.e.c.c.CoordinationState] [DESKTOP-J6G85LP] cluster UUID set to {0Cz26nE9QEiazRTB8_17A}
[2019-07-31T11:56:29,570][INFO ][o.e.c.s.ClusterApplierService] [DESKTOP-J6G85LP] master node changed [previous [], current [{DESKTOP-J6G85LP}{7Q50qVxCT457mmH94d0fGQ}{_PsaZxpVQcO_IR5rumzomg}{127.0.0.1}{127.0.0.1:9300}(dim){ml.machine_memory=8482041856, xpack.installed=true, ml.max_open_jobs=20}]], term: 1, version: 1, reason: Publication{term=1, version=1}
[2019-07-31T11:56:29,932][INFO ][o.e.g.GatewayService] [DESKTOP-J6G85LP] recovered [0] indices into cluster state
[2019-07-31T11:56:30,373][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.watch-history-10] for index patterns [.watcher-history-10*]
[2019-07-31T11:56:30,431][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.triggered_watches] for index patterns [.triggered_watches*]
[2019-07-31T11:56:30,502][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.watches] for index patterns [.watches*]
[2019-07-31T11:56:30,568][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.monitoring-logstash] for index patterns [.monitoring-logstash-7-*]
[2019-07-31T11:56:30,661][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.monitoring-es] for index patterns [.monitoring-es-7-*]
[2019-07-31T11:56:30,751][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.monitoring-beats] for index patterns [.monitoring-beats-7-*]
[2019-07-31T11:56:30,864][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.monitoring-alerts-7] for index patterns [.monitoring-alerts-7-*]
[2019-07-31T11:56:30,998][INFO ][o.e.c.m.MetaDataIndexTemplateService] [DESKTOP-J6G85LP] adding template [.monitoring-kibana] for index patterns [.monitoring-kibana-7-*]
[2019-07-31T11:56:31,085][INFO ][o.e.x.i.a.TransportPutLifecycleAction] [DESKTOP-J6G85LP] adding index lifecycle policy [watch-history-ilm-policy]
[2019-07-31T11:56:31,869][INFO ][o.e.l.licenseService] [DESKTOP-J6G85LP] license [8788c70c-e7a2-4840-8a7c-853aa5d2b897] mode [basic] - valid
[2019-07-31T11:56:31,908][INFO ][o.e.x.s.s.SecurityStatusChangeListener] [DESKTOP-J6G85LP] Active license is now [BASIC]; Security is disabled
[2019-07-31T11:56:32,127][INFO ][o.e.h.AbstractHttpServerTransport] [DESKTOP-J6G85LP] publish_address {127.0.0.1:9200}, bound_addresses {127.0.0.1:9200}, {[::1]:9200}
[2019-07-31T11:56:32,132][INFO ][o.e.n.Node] [DESKTOP-J6G85LP] started
```

6. Similarly, initiate Kibana through Windows command line interface by using the following commands:

- i) Go to your kibana folder on local: `cd kibana`
- ii) `bin\kibana`

```
Command Prompt - bin\kibana
log [19:07:14.199] [info][status][plugin:oss_telemetry@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:14.204] [info][status][plugin:file_upload@7.3.0] Status changed from uninitialized to yellow - Waiting for Elasticsearch
log [19:07:14.210] [warning][encrypted_saved_objects] Generating a random key for xpack.encrypted_saved_objects.encryptionKey. To be able to decrypt encrypted saved
objects attributes after restart, please set xpack.encrypted_saved_objects.encryptionKey in kibana.yml
log [19:07:14.212] [info][status][plugin:encrypted_saved_objects@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:14.225] [info][status][plugin:snapshot_restore@7.3.0] Status changed from uninitialized to yellow - Waiting for Elasticsearch
log [19:07:14.242] [info][status][plugin:actions@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:14.258] [info][status][plugin:alerting@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:14.265] [info][status][plugin:dates@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:15.383] [info][status][plugin:timelion@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:15.392] [info][status][plugin:ui_metric@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:15.397] [info][status][plugin:visualizations@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:19.128] [info][status][plugin:elasticsearch@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.489] [info][license][xpack] Imported license information from Elasticsearch for the [data] cluster: mode: basic | status: active
log [19:07:19.499] [info][status][plugin:xpack_main@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.500] [info][status][plugin:graph@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.504] [info][status][plugin:searchprofiler@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.507] [info][status][plugin:ml@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.509] [info][status][plugin:tilemap@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.510] [info][status][plugin:watcher@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.511] [info][status][plugin:grokdebugger@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.515] [info][status][plugin:logstash@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.518] [info][status][plugin:beats_management@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.520] [info][status][plugin:index_management@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.521] [info][status][plugin:index_lifecycle_management@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.523] [info][status][plugin:rollup@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.527] [info][status][plugin:remote_clusters@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.530] [info][status][plugin:cross_cluster_replication@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.530] [info][status][plugin:file_upload@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.532] [info][status][plugin:snapshot_restore@7.3.0] Status changed from yellow to green - Ready
log [19:07:19.534] [info][kibana-monitoring][monitoring] Starting monitoring stats collection
log [19:07:19.551] [info][status][plugin:maps@7.3.0] Status changed from yellow to green - Ready
log [19:07:23.750] [warning][reporting] Generating a random key for xpack.reporting.encryptionKey. To prevent pending reports from failing on restart, please set xp
ack.reporting.encryptionKey in kibana.yml
log [19:07:23.763] [info][status][plugin:reporting@7.3.0] Status changed from uninitialized to green - Ready
log [19:07:23.910] [info][task_manager] Installing .kibana_task_manager index template version: 7030099.
log [19:07:23.989] [info][task_manager] Installed .kibana_task_manager index template: version 7030099 (API version 1)
log [19:07:25.580] [info][migrations] Creating index .kibana_1.
log [19:07:25.981] [info][migrations] Pointing alias .kibana to .kibana_1.
log [19:07:26.066] [info][migrations] Finished in 498ms.
log [19:07:26.072] [info][listening] Server running at http://localhost:5601
log [19:07:26.114] [info][server][Kibana][http] http server running
log [19:07:26.868] [info][status][plugin:spaces@7.3.0] Status changed from yellow to green - Ready
```

7. Now we need to navigate to our Kibana host URL as shown in the above screenshot to access

Kibana through web browser on local:

i) log [19:07:26.072] [info][listening] Server running at <http://localhost:5601>

```
log [19:07:26.066] [info][migrations] Finished in 498ms.
log [19:07:26.072] [info][listening] Server running at http://localhost:5601
log [19:07:26.114] [info][server][Kibana][http] http server running
log [19:07:26.868] [info][status][plugin:spaces@7.3.0] Status changed from yellow to green - Ready
```

Step 3: Download and Map the Datasets

This tutorial requires Trending YouTube Video Statistics data set to be downloaded on to your local. You can download the dataset from Kaggle on the below given link:

i) Dataset link: <https://www.kaggle.com/datasnaek/youtube-new>

Before you load the data set, you must set up mappings for the fields. Mappings divide the documents in the index into logical groups and specify the characteristics of the fields. These characteristics include the searchability of the field and whether it's tokenized or broken up into separate words.

In Kibana Dev Tools > Console, set up a mapping for the YouTube data set by copying and pasting the following. Then, select Play button:

```
PUT global/

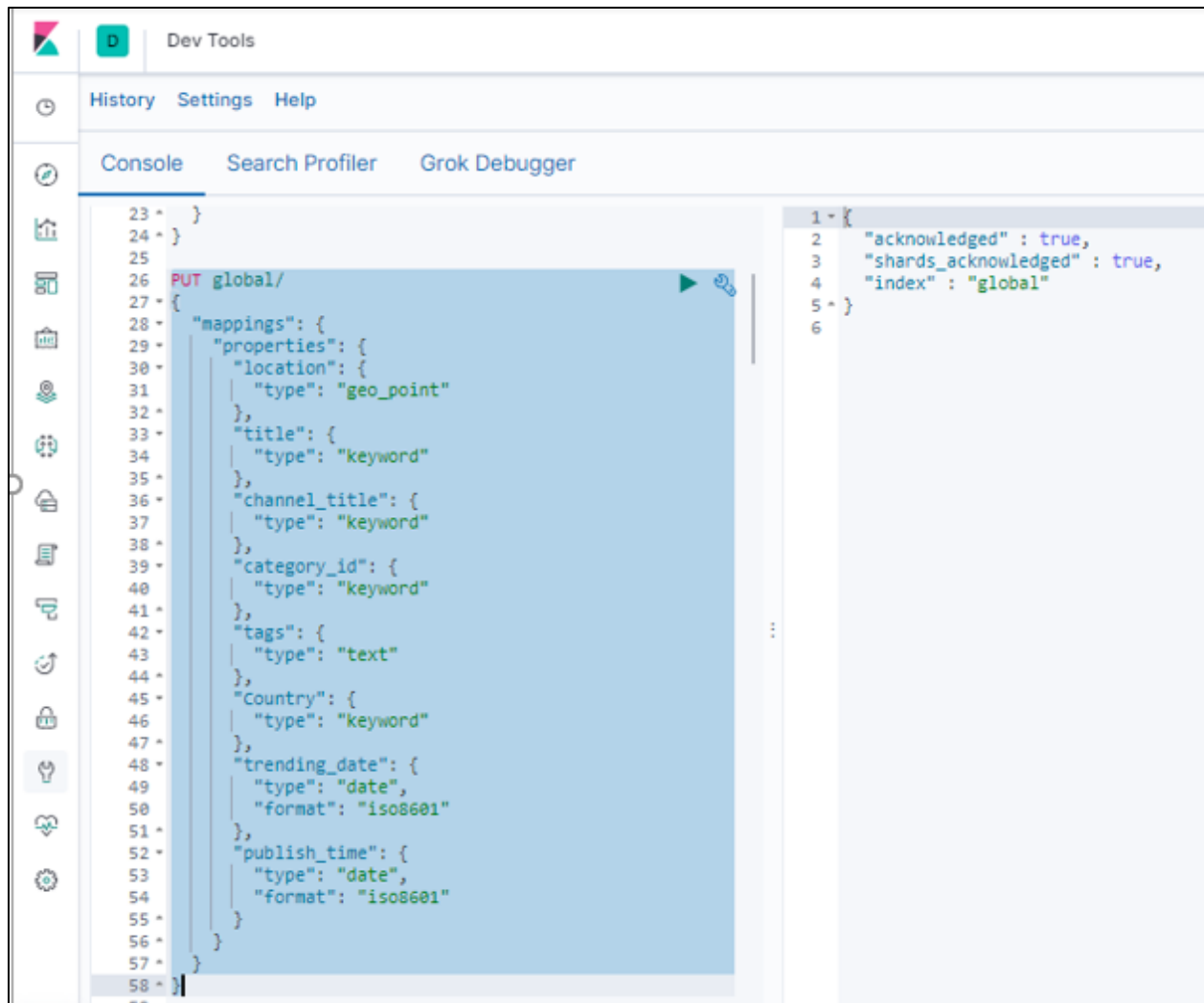
{
  "mappings": {
    "properties": {
      "location": {
        "type": "geo_point"
      },
      "title": {
        "type": "keyword"
      },
      "channel_title": {
        "type": "keyword"
      },
      "category_id": {
        "type": "keyword"
      },
      "tags": {
        "type": "text"
      },
      "Country": {
        "type": "keyword"
      },
      "trending_date": {
        "type": "date",
```

```
        "format": "iso8601"
    },
    "publish_time": {
        "type": "date",
        "format": "iso8601"
    }
}
}
```

This mapping specifies field characteristics for the data set:

- i) The title, channel_title, category_id and Country fields are keyword fields. These fields are not analyzed. The strings are treated as a single unit even if they contain multiple words.
- ii) The location is mapped in the form of geo_point to accept latitude-longitude pairs and to aggregate documents geographically.

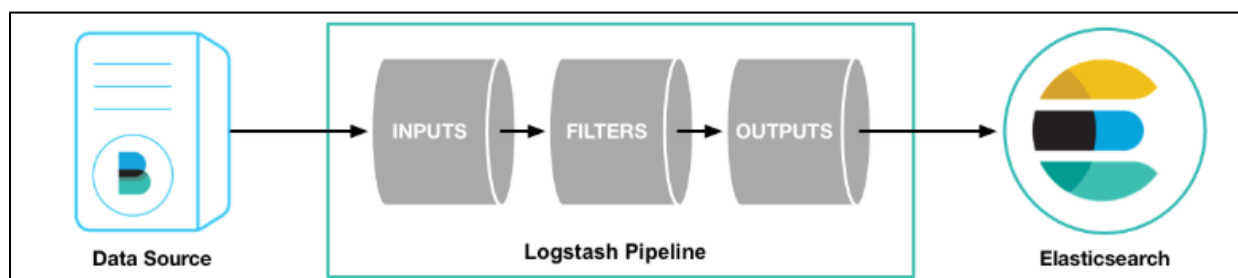
The trending_date and publish_time are the date fields in iso8601 format.



Step 4: Parsing Logs with Logstash

We now need to ingest our data from the dataset into Elasticsearch for visualizations through Logstash which is a primary part of ELK stack that collects, parses and transforms our data into Elasticsearch cluster.

Processing of Logstash as given below in pictorial representation:



- i) We need to define the input and filters in the form of Logstash configuration file. Open notepad++ and define the code as given below and save it as .conf file:

```
input{
  file{
    path => "D:/Summer_2019_Course/Project/global_youtube.csv"
    start_position => "beginning"
    sincedb_path => "NUL"
  }
}

filter {
  csv {
    separator => ","
    columns => ["video_id", "trending_date", "title",
"channel_title", "category_id", "publish_time", "tags", "views", "likes",
"dislikes", "comment_count", "latitude", "longitude",
"Country"]
  }

  mutate {convert => ["views", "integer"]}
  mutate {convert => ["likes", "integer"]}
  mutate {convert => ["dislikes", "integer"]}
  mutate {convert => ["comment_count", "integer"]}
  mutate {convert => {"latitude" => "float"}}
  mutate {convert => {"longitude" => "float"}}
  mutate {rename => {"latitude" => "[location][lat]"}}
  mutate {rename => {"longitude" => "[location][lon]"}}
  date {match => [ "trending_date","ISO8601" ]
```

```

        timezone => "America/Chicago"}

        date {match => [ "publish_time","ISO8601" ]

        timezone => "America/Chicago"}

        if [message] =~ "^?video_id" { drop {} }

    }

output {

    elasticsearch {

        hosts => "localhost"

        index => "publish_time_global"

        document_type => "_doc"

    }

    stdout {}

}

```

ii) The configuration file would take the input i.e. YouTube dataset which is in the form of a comma separated file, define its columns and transform data into integer, location and match the date in its correct format.

iii) Now go to the Logstash folder through windows command line:

```
cd logstash-7.2.1
```

iv) Execute the below command in order to ingest our Final.conf i.e. Logstash configuration file into our Elasticsearch cluster

```
bin\logstash -f D:\Summer_2019_Course\logstash-7.2.1\final.conf
```

```
D:\Summer_2019_Course\logstash-7.2.1>bin\logstash -f D:\Summer_2019_Course\logstash-7.2.1\final.conf
```

v) Once executed the data from the dataset will start getting loaded into our Elasticsearch cluster as shown below:

```
Command Prompt - bin\logstash -f D:\Summer_2019_Course\logstash-7.2.1\final.conf

"@version" => "1",
"message" => "GfXNGjfsKRY,2018-06-13T00:00:00.000Z,James Bay - Delicate (Taylor Swift cover) in the Live Lounge,BBCRadio1VEVO,Music,2018-05-24T17:00:17.000Z,\
James Bay\\\"Taylor Swift\\\"\\\"Delicate\\\"\\\"BBC\\\"\\\"Radio 1\\\"\\\"Live Lounge\\\"\\\",807683,28245,536,934,51.509865,-0.118092,United Kingdom\r",
"views" => 807683,
"path" => "D:/Summer_2019_Course/Project/global_youtube.csv",
"Country" => "United Kingdom",
"location" => {
  "lon" => -0.118092,
  "lat" => 51.509865
},
"@timestamp" => 2018-05-24T17:00:17.000Z,
"host" => "DESKTOP-J6G85LP",
"dislikes" => 536,
"likes" => 28245,
"video_id" => "GfXNGjfsKRY"
},
{
  "tags" => "music\\\"hiphop\\\"\\\"hip\\\"\\\"hop\\\"\\\"rap\\\"\\\"mix\\\"\\\"remix\\\"\\\"free\\\"\\\"download\\\"\\\"mixtape\\\"\\\"album\\\"\\\"leak\\\"\\\"drake\\\"\\\"drake lyrics\\\"\\\"lil
baby lyrics\\\"\\\"lil baby\\\"\\\"yes indeed\\\"\\\"yes indeed lyrics\\\"\\\"drake lil baby\\\"\\\"drake yes ineed\\\"\\\"lil baby yes indeed\\\"\\\"freestyle\\\"\\\"southside\\\"",
"channel_title" => "FutureHype",
"publish_time" => "2018-05-18T17:35:05.000Z",
"trending_date" => "2018-06-13T00:00:00.000Z",
"comment_count" => 1863,
"title" => "Drake & Lil Baby - Yes Indeed (Lyrics)",
"category_id" => "Music",
"@version" => "1",
"message" => "AbEHRrq7xwU,2018-06-13T00:00:00.000Z,Drake & Lil Baby - Yes Indeed (Lyrics),FutureHype,Music,2018-05-18T17:35:05.000Z,\"music\\\"\\\"hiphop\\\"\\\"\\\"
hip\\\"\\\"\\\"hop\\\"\\\"\\\"rap\\\"\\\"\\\"mix\\\"\\\"\\\"remix\\\"\\\"\\\"free\\\"\\\"\\\"download\\\"\\\"\\\"mixtape\\\"\\\"\\\"album\\\"\\\"\\\"leak\\\"\\\"\\\"drake\\\"\\\"\\\"drake lyrics\\\"\\\"\\\"
lil baby lyrics\\\"\\\"\\\"lil baby\\\"\\\"\\\"yes indeed\\\"\\\"\\\"yes indeed lyrics\\\"\\\"\\\"drake lil baby\\\"\\\"\\\"drake yes ineed\\\"\\\"\\\"lil baby yes indeed\\\"\\\"\\\"free
style\\\"\\\"\\\"southside\\\"\\\"\",5579404,55587,2083,1863,51.509865,-0.118092,United Kingdom\r",
"views" => 5579404,
"path" => "D:/Summer_2019_Course/Project/global_youtube.csv",
"Country" => "United Kingdom",
"location" => {
  "lon" => -0.118092,
  "lat" => 51.509865
},
"@timestamp" => 2018-05-18T17:35:05.000Z,
"host" => "DESKTOP-J6G85LP",
"dislikes" => 2083,
"likes" => 55587,
"video_id" => "AbEHRrq7xwU"
}
```

vi) In Kibana Dev Tools > Console, verify that the data from the dataset has been ingested into our Elasticsearch cluster by using the API console in Kibana

```
173 GET /global/_count
174
```

Command will give you the count of documents uploaded in Elasticsearch.

```
History Settings Help

Console Search Profiler Grok Debugger

143     "type": "text",
144     "fields": {
145       "keyword": {
146         "type": "keyword",
147         "ignore_above": 256
148       }
149     },
150   },
151   "Country": {
152     "type": "text",
153     "fields": {
154       "keyword": {
155         "type": "keyword",
156         "ignore_above": 256
157       }
158     }
159   }
160 }
161 }
162 :
163
164 GET /global/_search
165 {
166   "query": {
167     "match_all": {}
168   }
169 }
170
171
172
173 GET /global/_count
174
```

```
1- {
2   "count" : 292841,
3   "_shards" : {
4     "total" : 1,
5     "successful" : 1,
6     "skipped" : 0,
7     "failed" : 0
8   }
9 }
10
```

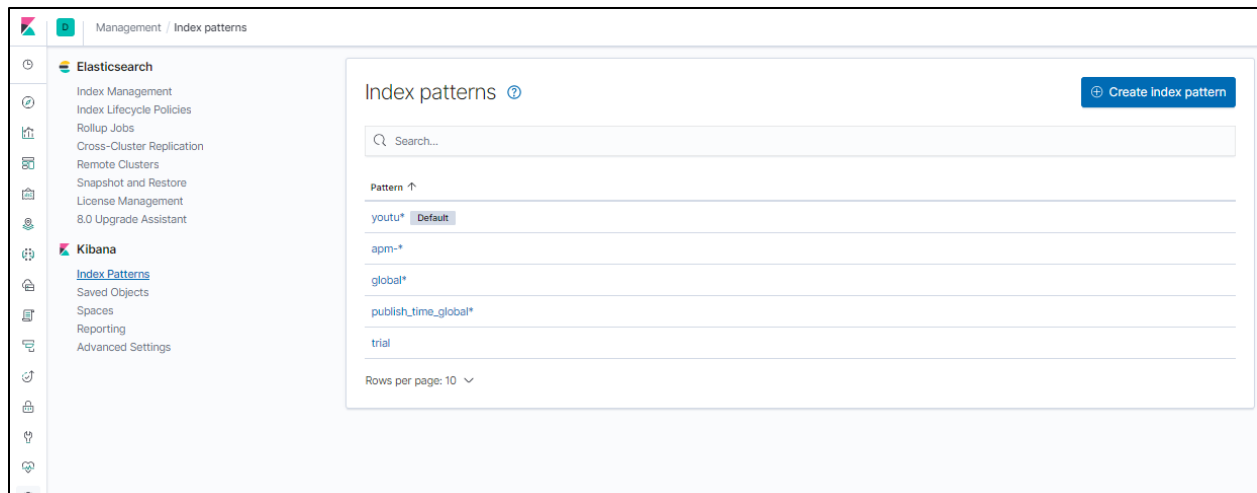
Step 4: Defining your index patterns

Index patterns tell Kibana which Elasticsearch indices you want to explore. An index pattern can match the name of a single index or include a wildcard (*) to match multiple indices.

You'll create patterns for the YouTube data set, which has an index named global

This data sets contain time-series data.

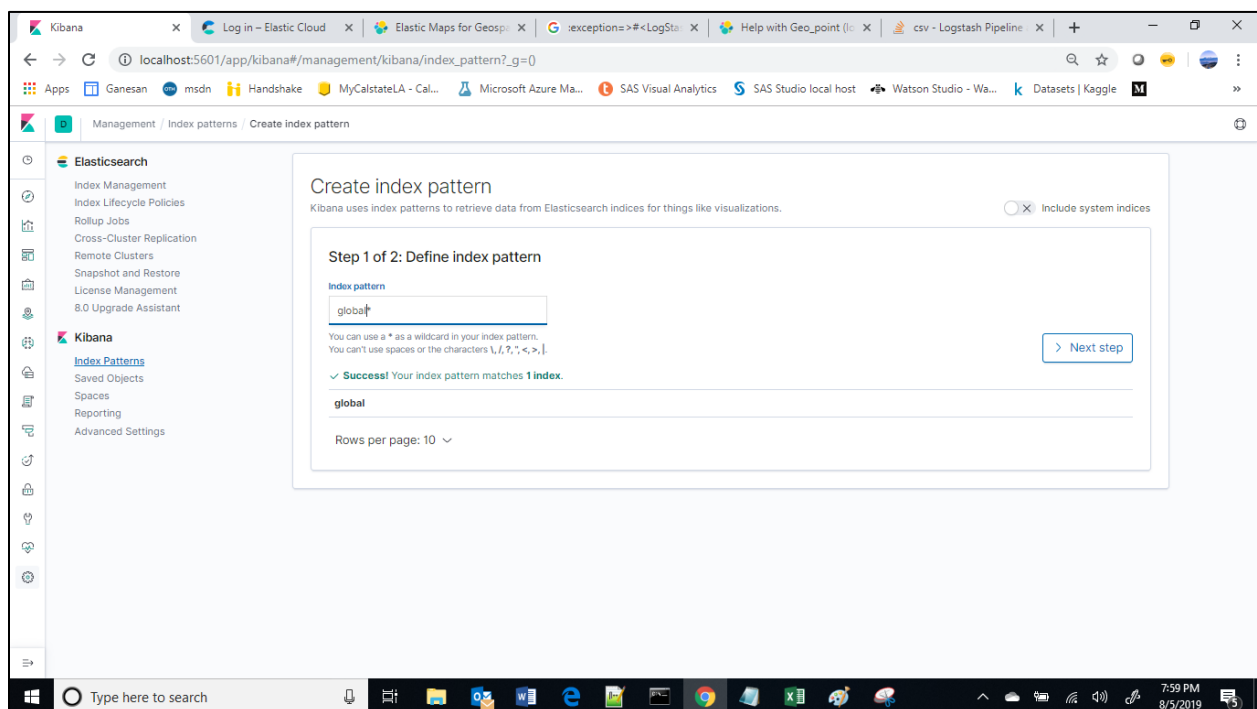
- i) In Kibana, open Management, and then click Index Patterns.



ii) If this is your first index pattern, the Create index pattern page opens automatically.

Otherwise, click Create index pattern in the upper left.

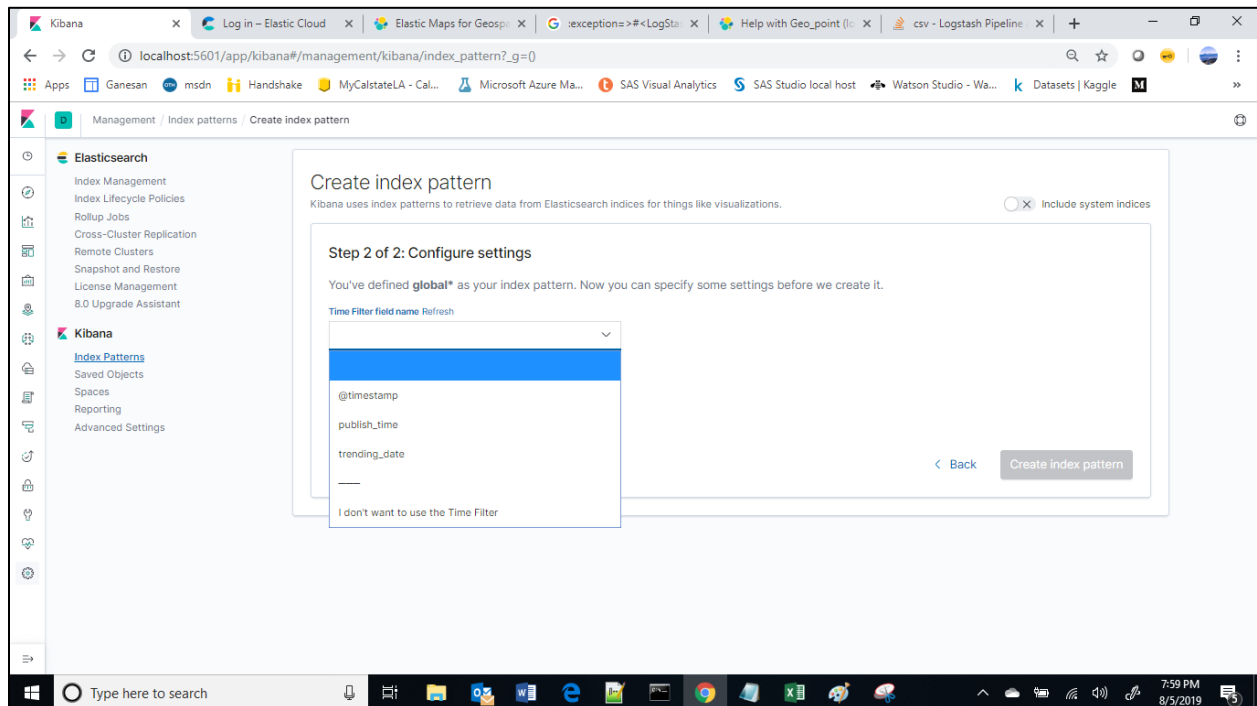
iii) Enter global* in the Index pattern field, which should show “Success!”. If not, your data uploading has not worked.



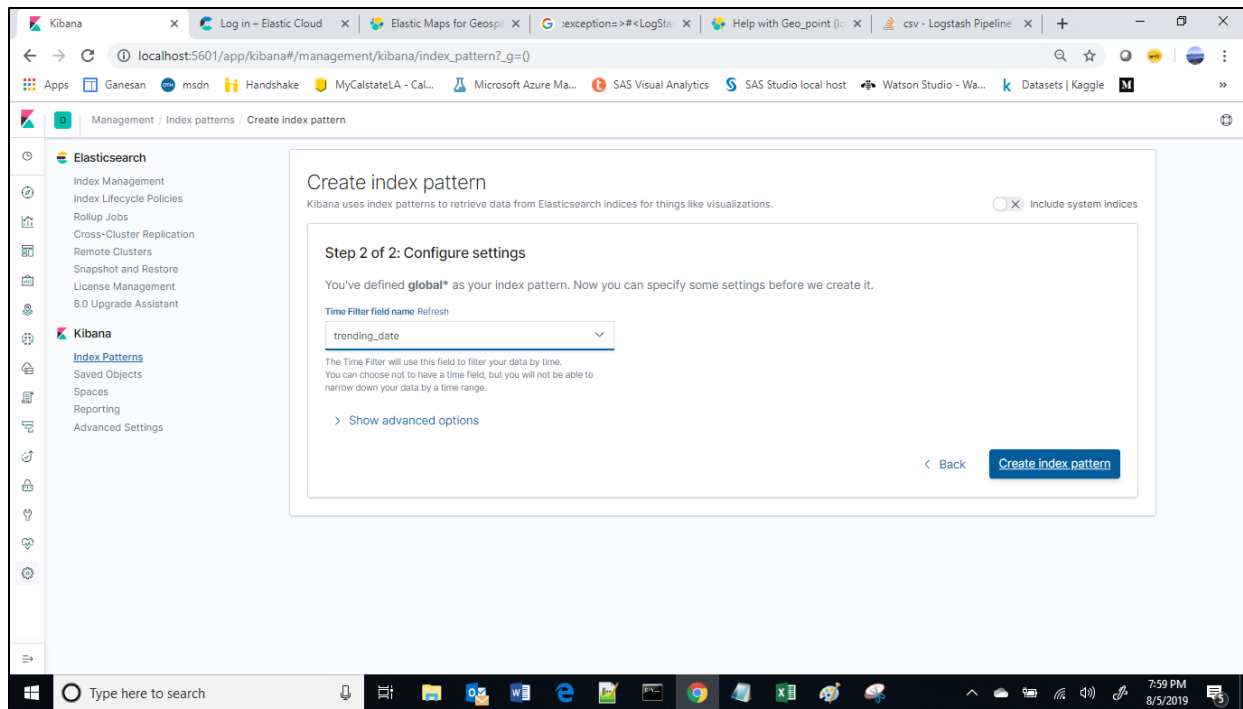
iv) Click Next step.

v) In Configure settings, click Create index pattern. For this pattern, you don't need to configure any settings.

vi) This data set contains time-series data. In Configure settings, select @timestamp in the Time Filter field name dropdown menu. Click create index pattern.



vii) Click create index pattern.



viii) You will see that Timestamp field is indexed as Time/date field. It will show the following fields that are indexed with the data types you defined at Mapping:

global*

Time Filter field name: trending_date

This page lists every field in the **global*** index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch [Mapping API](#).

| Name | Type | Format | Searchable | Aggregatable | Excluded |
|------------------|---------|--------|------------|--------------|----------|
| @timestamp | date | | • | • | |
| @version | string | | • | | |
| @version.keyword | string | | • | • | |
| Country | string | | • | • | |
| _id | string | | • | • | |
| _index | string | | • | • | |
| _score | number | | | | |
| _source | _source | | | | |
| _type | string | | • | • | |

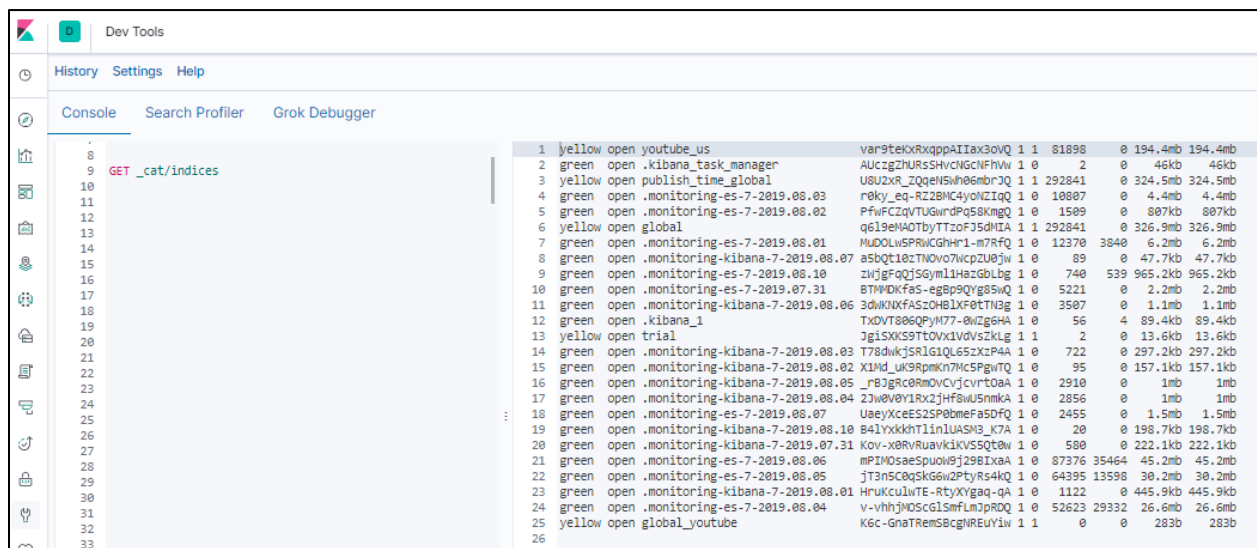
ix) Similarly create one more index to define the 2nd timeline that YouTube dataset has i.e.

Publish_time with the index pattern name as publish_time_global.

x) When you define an index pattern, the indices that match that pattern must exist in

Elasticsearch and they must contain data.

xi) To check which indices are available, go to Dev Tools > Console and enter GET _cat/indices.



| id | status | type | name | mapping | settings | size |
|----|--------|------|---------------------------------|------------------------|----------|---------------------------|
| 1 | yellow | open | youtube_us | var9teKXrXappAIiax3oVQ | 1 1 | 81898 0 194.4mb 194.4mb |
| 2 | green | open | .kibana_task_manager | AUczgZhuRS5hVcNGcNFHvw | 1 0 | 2 0 46kb 46kb |
| 3 | yellow | open | publish_time_global | USU2zXR_ZQqEN5wh6mbrJQ | 1 1 | 292841 0 324.5mb 324.5mb |
| 4 | green | open | .monitoring-es-7-2019.08.03 | r8ky_eq-RZ2BHC4yoNZIQ | 1 0 | 18887 0 4.4mb 4.4mb |
| 5 | green | open | .monitoring-es-7-2019.08.02 | PfwFCZqVTUGwrdPqS8KmgQ | 1 0 | 1589 0 807kb 807kb |
| 6 | yellow | open | global | q619eHAOTbyTTzoF35dHIA | 1 1 | 292841 0 326.9mb 326.9mb |
| 7 | green | open | .monitoring-es-7-2019.08.01 | MU0DLw5PRwCGHm1-m7RfQ | 1 0 | 12370 3948 6.2mb 6.2mb |
| 8 | green | open | .monitoring-kibana-7-2019.08.07 | a5uqt18zTN0vo7wcp2U0Jw | 1 0 | 89 0 47.7kb 47.7kb |
| 9 | green | open | .monitoring-es-7-2019.08.10 | zhJgFqQj5GymLIHaZ0L0g | 1 0 | 740 539 965.2kb 965.2kb |
| 10 | green | open | .monitoring-es-7-2019.07.31 | 8TWKDKfas-eg8p9QygsSwQ | 1 0 | 5221 0 2.2mb 2.2mb |
| 11 | green | open | .monitoring-kibana-7-2019.08.06 | 3dWkXKFASz0HBLX0T7N0g | 1 0 | 3507 0 1.1mb 1.1mb |
| 12 | green | open | .kibana_1 | TxDVT806QPyH77-0wZg6HA | 1 0 | 56 4 89.4kb 89.4kb |
| 13 | yellow | open | trial | Jgi5KX59Tt0VxIVdV5zKlg | 1 1 | 2 0 13.6kb 13.6kb |
| 14 | green | open | .monitoring-kibana-7-2019.08.03 | T78dwkj5RLG1QL65zKzP4A | 1 0 | 722 0 297.2kb 297.2kb |
| 15 | green | open | .monitoring-kibana-7-2019.08.02 | X1Md_UK9RpmKn7Mc5PgwTQ | 1 0 | 95 0 157.1kb 157.1kb |
| 16 | green | open | .monitoring-kibana-7-2019.08.05 | _r8Jgrc0Rm0VcVjcvrT0aA | 1 0 | 2910 0 1mb 1mb |
| 17 | green | open | .monitoring-kibana-7-2019.08.04 | 2Jw8VBY1RXzjHf8uU5nmkA | 1 0 | 2856 0 1mb 1mb |
| 18 | green | open | .monitoring-es-7-2019.08.07 | UaeYXceES2SP0bneFa5DfQ | 1 0 | 2455 0 1.5mb 1.5mb |
| 19 | green | open | .monitoring-kibana-7-2019.08.10 | B4lyxkkhTlinLUASm3_K7A | 1 0 | 20 0 198.7kb 198.7kb |
| 20 | green | open | .monitoring-kibana-7-2019.07.31 | Kov-x8RvRuavkiKV5SQ0w | 1 0 | 580 0 222.1kb 222.1kb |
| 21 | green | open | .monitoring-es-7-2019.08.06 | mPIW0SaeSpUw0j298IXaA | 1 0 | 87376 35464 45.2mb 45.2mb |
| 22 | green | open | .monitoring-es-7-2019.08.05 | jT3N5C0qSk6wQ2PtyRs4kQ | 1 0 | 64395 13598 30.2mb 30.2mb |
| 23 | green | open | .monitoring-kibana-7-2019.08.01 | HrukCulWTE-RtyXYgaq-QA | 1 0 | 1122 0 445.9kb 445.9kb |
| 24 | green | open | .monitoring-es-7-2019.08.04 | V-vhhjW0ScG1SmfLm3pRDQ | 1 0 | 52623 29332 26.6mb 26.6mb |
| 25 | yellow | open | global_youtube | K6c-GnaTremSBcgNREUYiw | 1 1 | 0 0 283b 283b |
| 26 | | | | | | |

Step 5: Discovering your data

Using the Discover application, you can enter an Elasticsearch query to search your data and filter the results.

i) Open Discover. The current index pattern appears below the filter bar, in this case global*.

You might need to click New in the menu bar to refresh the data.

ii) In the Time filter field, enter the following dates and select Update/Refresh button:

Nov 6, 2017 – May 6, 2018.

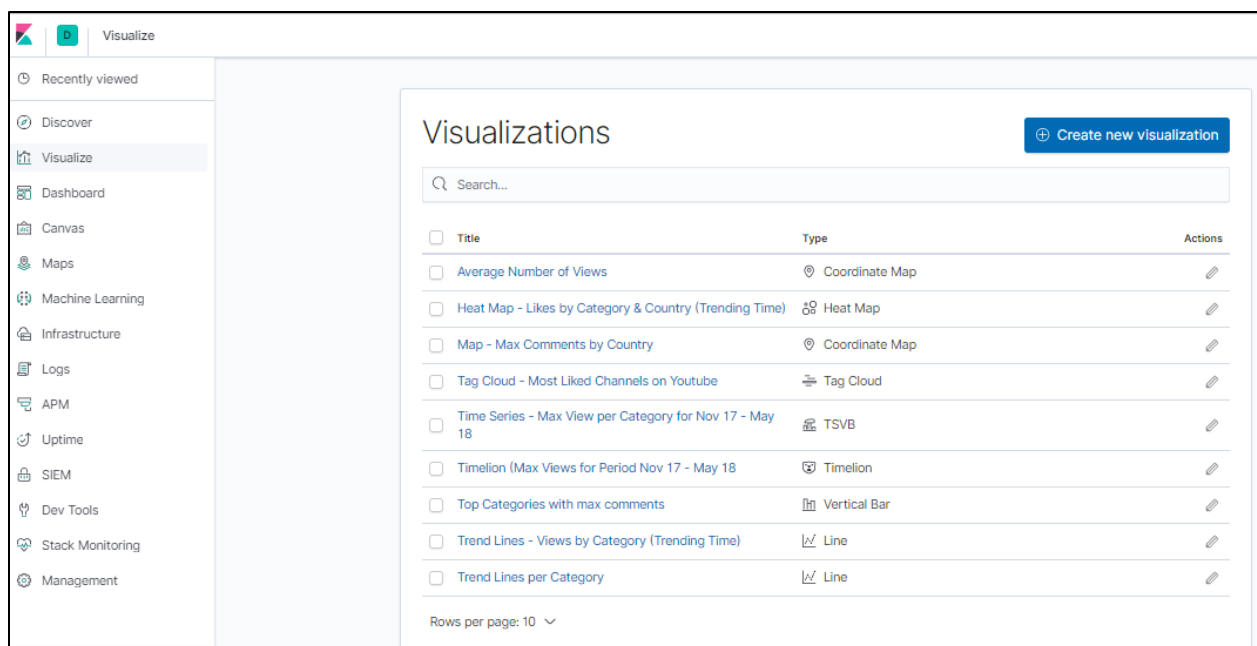
iii) The search returns all YouTube documents between the above given dates.



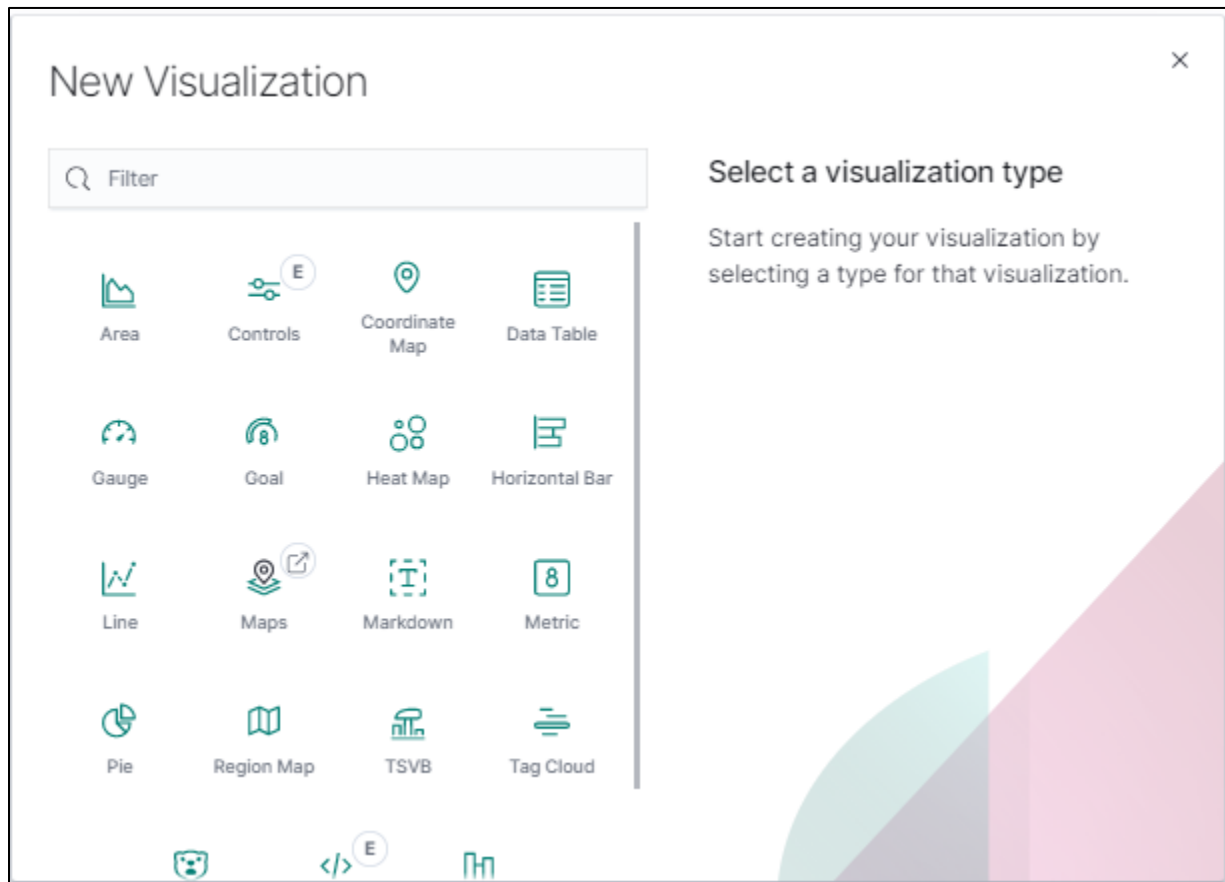
Step 5: Visualizing your data

In the Visualize application, you can shape your data using a variety of charts, tables, and maps, and more. You'll create six visualizations in the lab: a timelion chart, time-series chart, trend line chart, tag cloud, heat map and geo spatial chart.

- i) In Kibana page, open Visualize.



ii) Click Create a visualization or the + button. You'll see all the visualization types in Kibana.



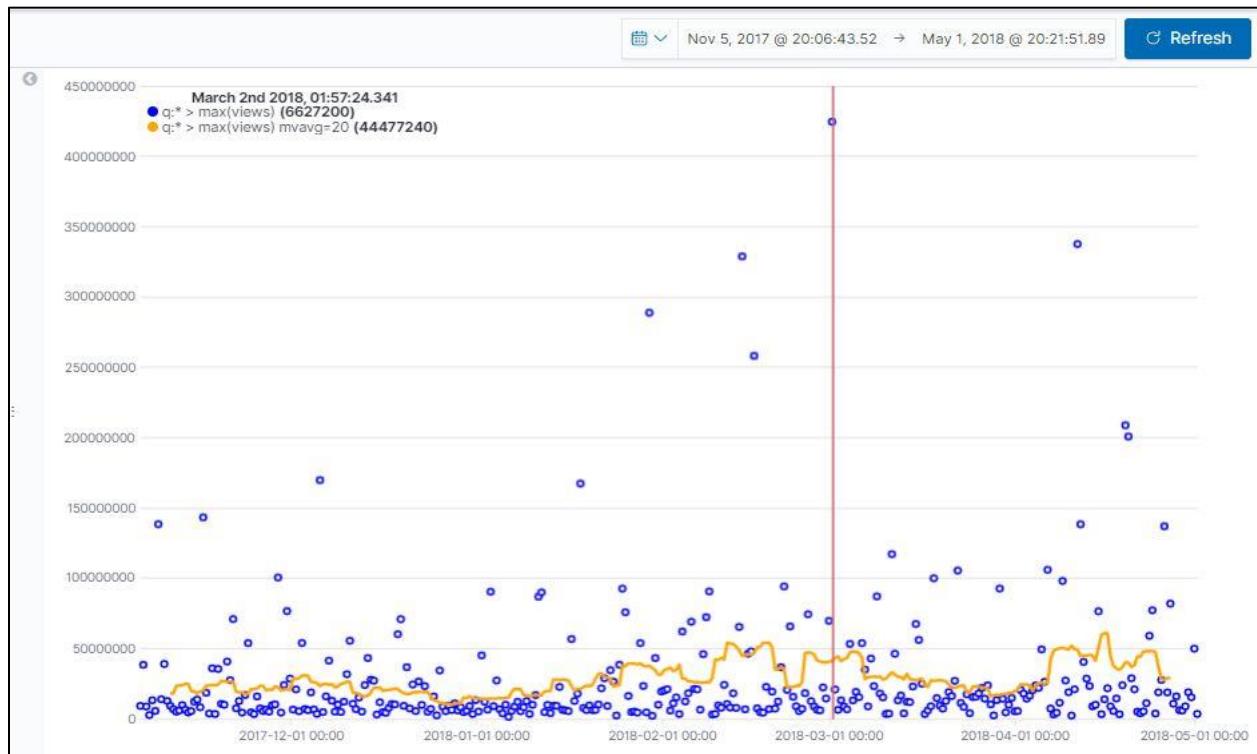
iii) In New Search, select the global* index pattern. You'll use Timelion chart to gain insight into the time line of the YouTube views.

Timelion

Timelion is a time series data visualizer that enables you to combine totally independent data sources within a single visualization. It's driven by a simple expression language you use to retrieve time series data, perform calculations to tease out the answers to complex questions, and visualize the results. We would now visualize to see the time series for maximum Views of YouTube users over the time period – Nov 2017 to May 2018

Code:

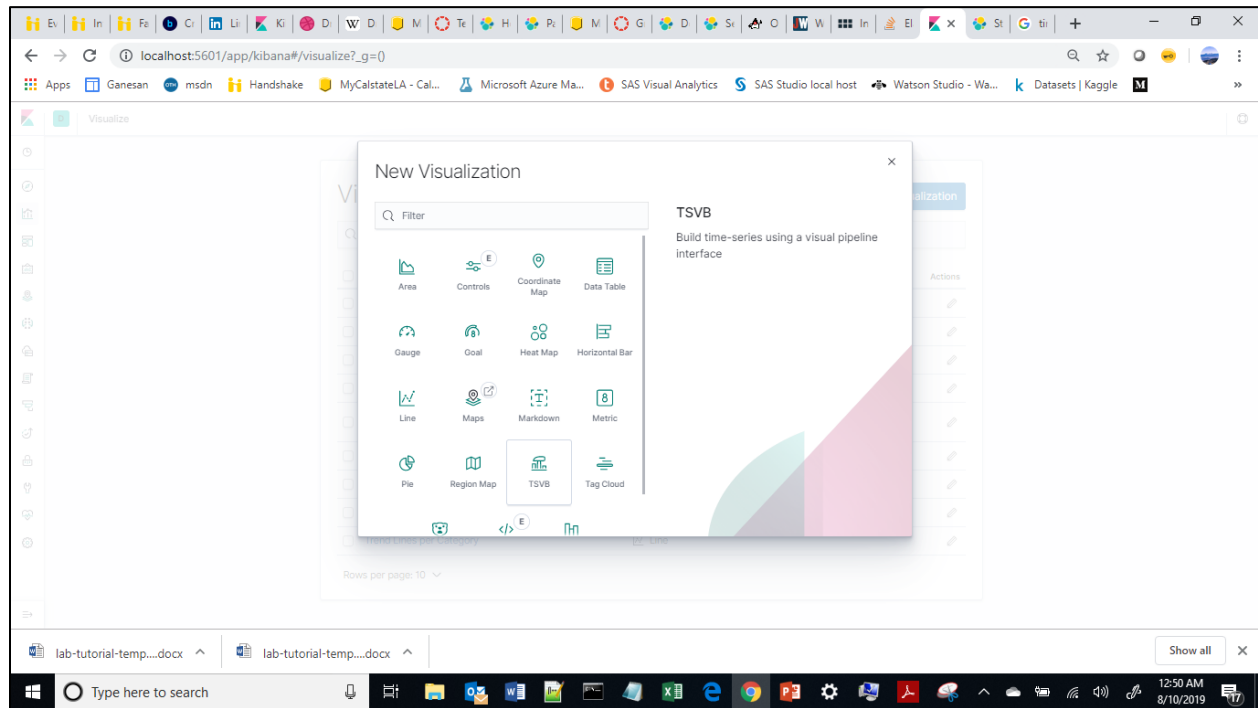
```
.es(index=publish_time_global*,metric=max:views).points().color(blue),  
.es(index=publish_time_global*,metric=max:views).mvavg(20).color(orange)  
e)
```



The above graph shows the time series graph analysis of the pattern of YouTube videos viewed by public on a monthly basis. With the help of this analysis we can understand the pattern as to when the audience are viewing videos and how many users are actively viewing YouTube channels in a particular time frame. From the graph it is evident that maximum viewership was clocked by YouTube in the months of March and April.

TSVB

TSVB is a time series data visualizer that allows you to use the full power of the Elasticsearch aggregation framework.



Aggregation

The aggregations framework helps provide aggregated data based on a search query. It is based on simple building blocks called aggregations, that can be composed in order to build complex summaries of the data.

Settings

- Aggregations: Max

- Field: views
- Group: terms
- By: category_id.keyword
- Top: 10
- Order by: Terms
- Direction: Descending

Panel options configuration:

- Label:** [Empty]
- Metrics:**
 - Aggregation: Max
 - Field: views
- Options:**
 - Group by: Terms
 - By: category_id.keyword
 - Include: [Empty]
 - Exclude: [Empty]
 - Top: 10
 - Order by: Terms
 - Direction: Descending

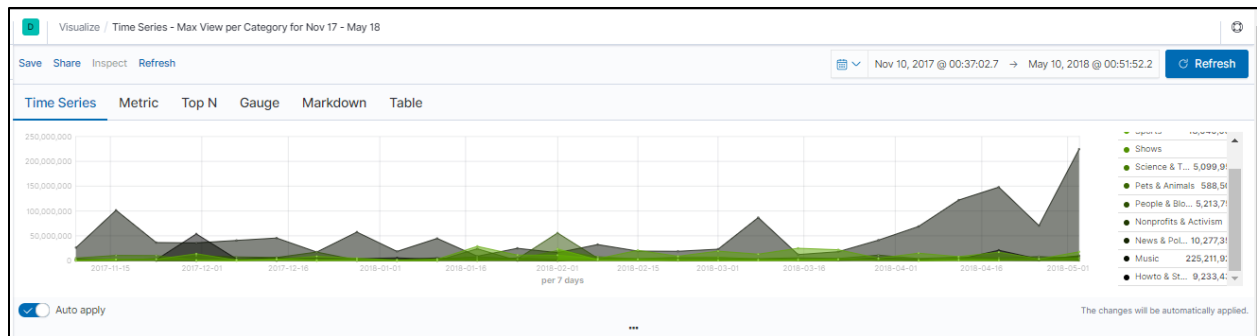
- Panel Options: Interval → 7d

This will represent out data on weekly basis.

Panel options configuration:

- Data:**
 - Index pattern: logstash-*
 - Time field: publish_time
 - Interval: 7d
 - Drop last bucket?: Yes
- Panel filter:** Search
- Ignore global filter?:** No

Visualization



Trend Lines

Trend lines are used to predict the continuation of a certain trend of a variable. It also helps to identify the correlation between two variables by observing the trend in both simultaneously.

Settings:

Metric

Buckets

Buckets

⌵ X-axis

👁️ ⓘ ✕

Aggregation

[Date Histogram help](#)

Date Histogram

⌵

Field

trending_date

⌵

Minimum interval

Auto

✕ ⌵

Select an option or create a custom value.
Examples: 30s, 20m, 24h, 2d, 1w, 1M

☐ ✕ Drop partial buckets

Custom label

Split Series

Split series

Sub aggregation

Terms

Field

category_id

Order by

Metric: Average views

Order

Descending

Size

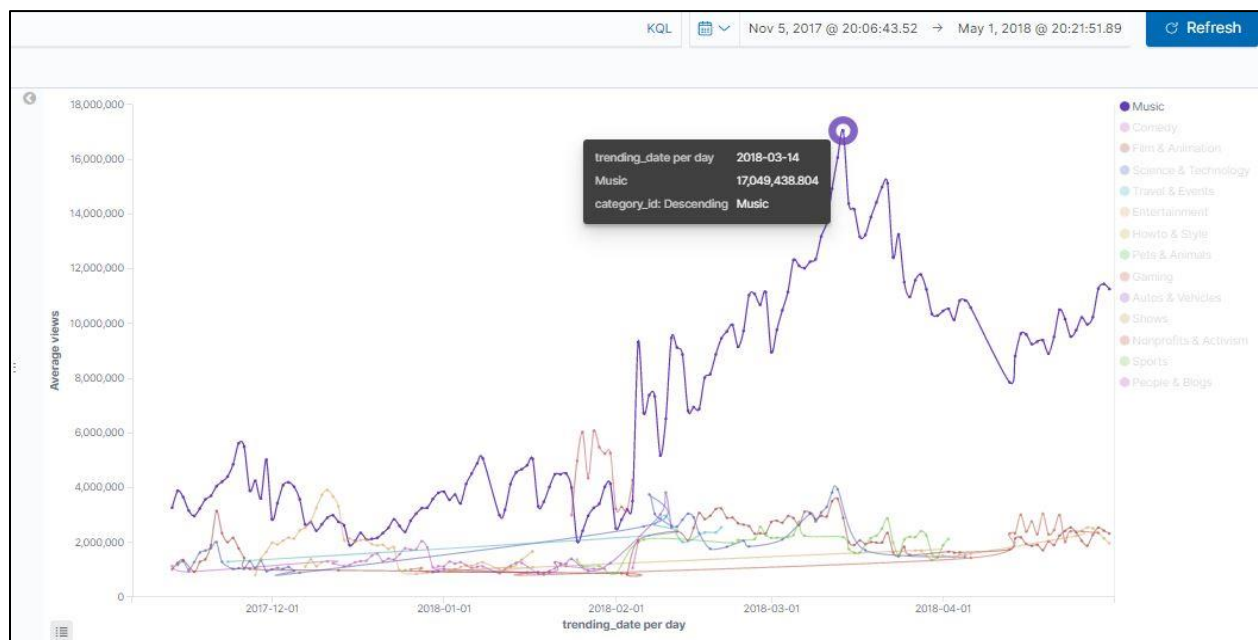
3

☐ Group other values in separate bucket

☐ Show missing values

Custom label

Visualization

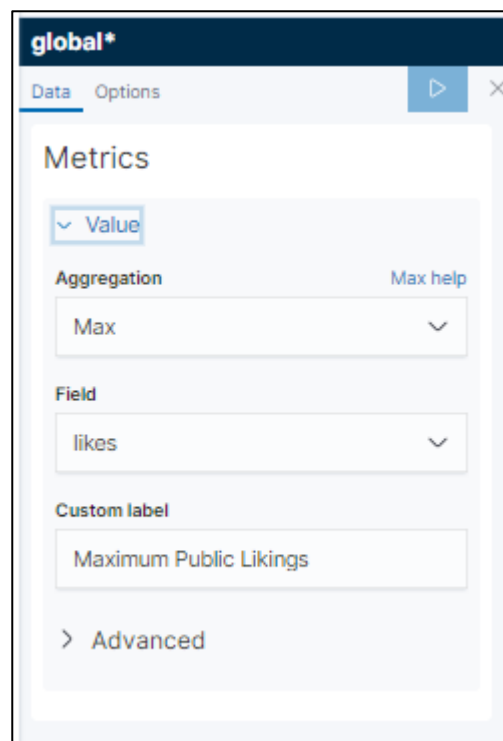


This graphical representation gives insight about the categories that are trending daily. The above analysis depicts that Music is the most preferred YouTube category which is viewed daily by many audiences.

Heat Map

A heat map is a graphical representation of data where the individual values contained in a matrix are represented as colors. The color for each matrix position is determined by the metrics aggregation.

Settings:



The image shows a settings panel for a visualization tool, titled "global*" in a dark blue header. Below the header, there are two tabs: "Data" (which is selected and underlined) and "Options". To the right of the tabs are a play button icon and a close button icon (an 'X'). The main content area is titled "Metrics" and contains several settings:

- A dropdown menu labeled "Value" with a downward arrow, currently showing "Value".
- An "Aggregation" section with a dropdown menu showing "Max" and a "Max help" link to its right.
- A "Field" section with a dropdown menu showing "likes" and a downward arrow.
- A "Custom label" section with a text input field containing "Maximum Public Likings".
- A section labeled "> Advanced" at the bottom.

Bucket

~ X-axis

Terms help

Aggregation

Terms

Field

category_id

Order by

Metric: Maximum Public Likings

Order

Descending

Size

10

☒ Group other values in separate bucket

☒ Show missing values

Custom label

Top Categories by Public Likings

> Advanced

~ Y-axis

Terms help

Sub aggregation

Terms

Field

Country

Order by

Metric: Maximum Public Likings

Order

Descending

Size

10

☒ Group other values in separate bucket

☒ Show missing values

Custom label

Country

Visualization



We have also analyzed the country-wise preference of a YouTube category using the heat map. From the above analysis countries such as Europe and United States have a greater viewership for the category of Music which is followed by Entertainment.

Tag Cloud

A tag cloud visualization is a visual representation of text data, typically used to visualize free form text.

Tags are usually single words, and the importance of each tag is shown with font size or color. The font size for each word is determined by the metrics aggregation.

Settings:

publish_time_global*

Data Options ▶ ×

Metrics

▼ Tag size

Aggregation [Max help](#)

Max ▼

Field

likes ▼

Custom label

Public Likes

Buckets

Buckets

▼ Tags

Aggregation [Terms help](#)

Terms ▼

Field

channel_title ▼

Order by

Metric: Public Likes ▼

Order **Size**

Descending ▼ 200

☐ ☒ Group other values in separate bucket

☐ ☒ Show missing values

Visualization

publish_time_global*

Data Options ▶ ×

Metrics

▼ Value

Aggregation Max help

Max ▼

Field

comment_count ▼

Custom label

Maximum Comment Count

> Advanced

Bucket

Buckets

▼ Geo coordinates

Aggregation Geohash help

Geohash ▼

Field

location ▼

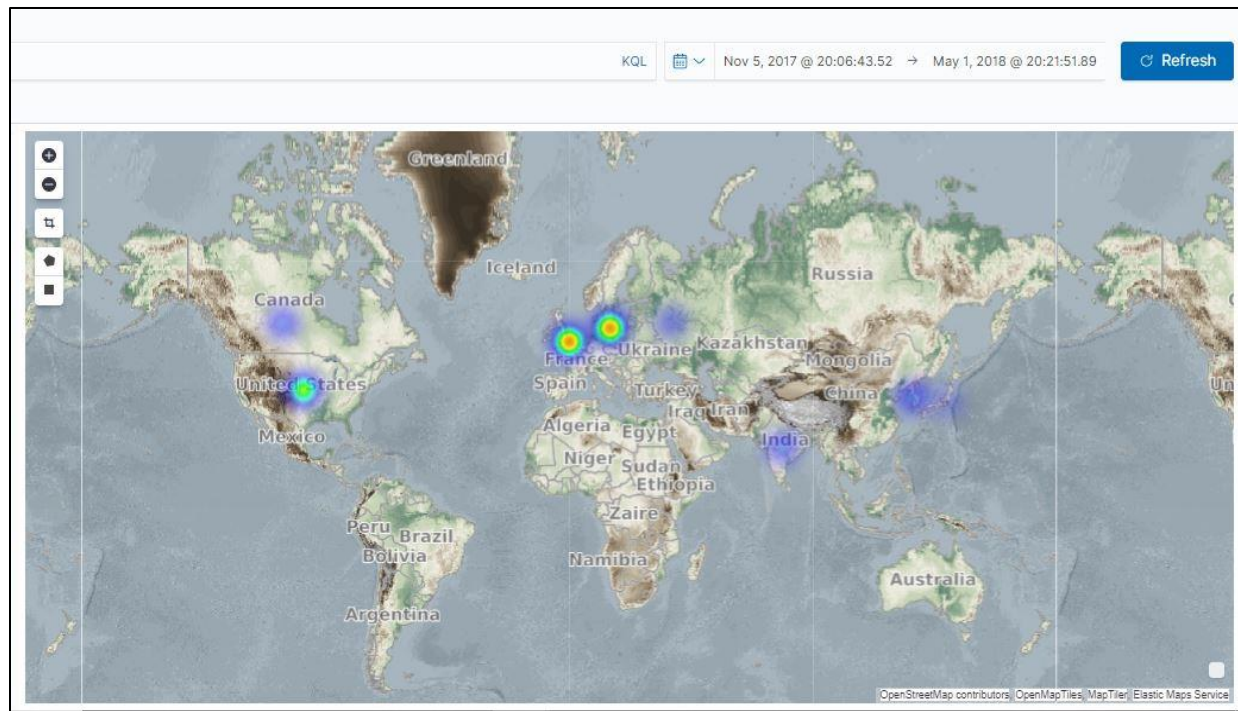
☒ Change precision on map zoom

☒ Place markers off grid (use geocentroid)

☒ Only request data around map extent

Custom label

Visualization



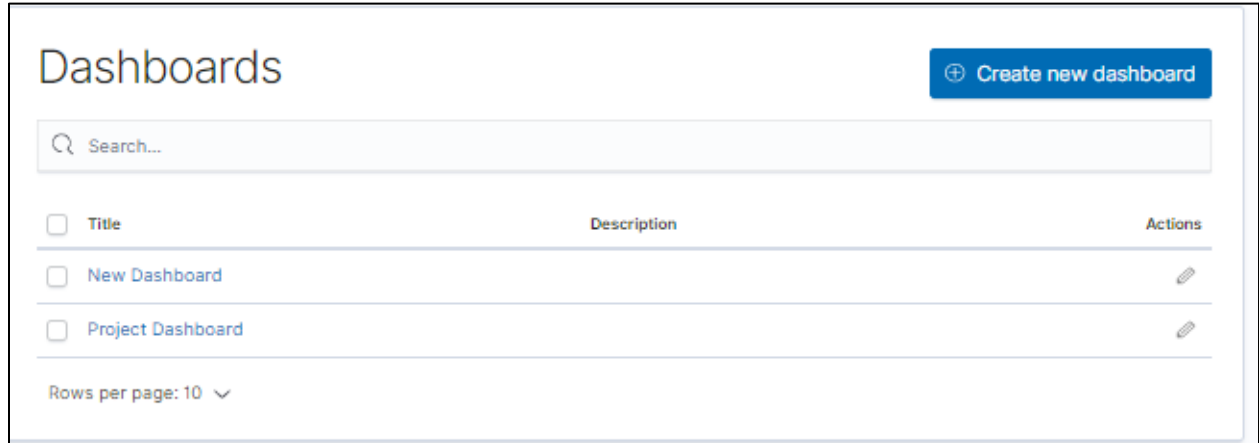
The above geographical heat map represents the number of comments per location. From this representation we can find out the location having maximum number of users who comment on YouTube videos. And it can be seen that in countries like United Kingdom, France and Germany the users have commented a greater number of times on the YouTube videos. Whereas the trend gets weaker as we come across countries like India and Japan.

Dashboard

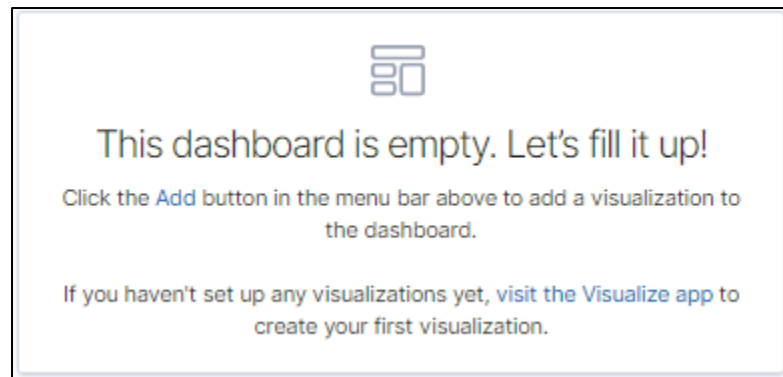
A Kibana dashboard is a collection of visualizations, searches, and maps, typically in real-time. Dashboards provide at-a-glance insights into your data and enable you to drill down into details.

1. To start working with dashboards, click Dashboard in the side navigation.
2. In the side navigation, click Dashboard.

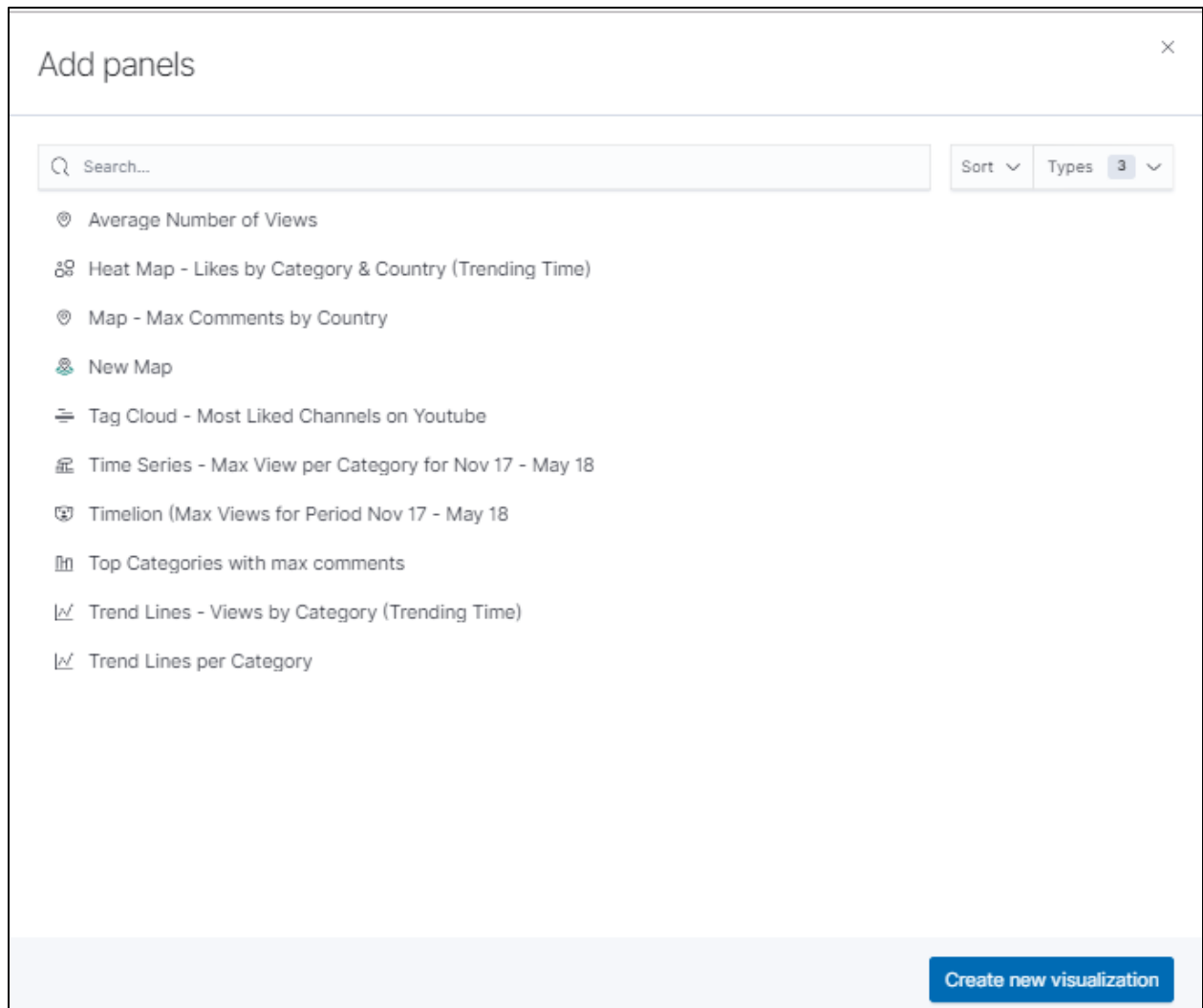
3. Click Create new dashboard.



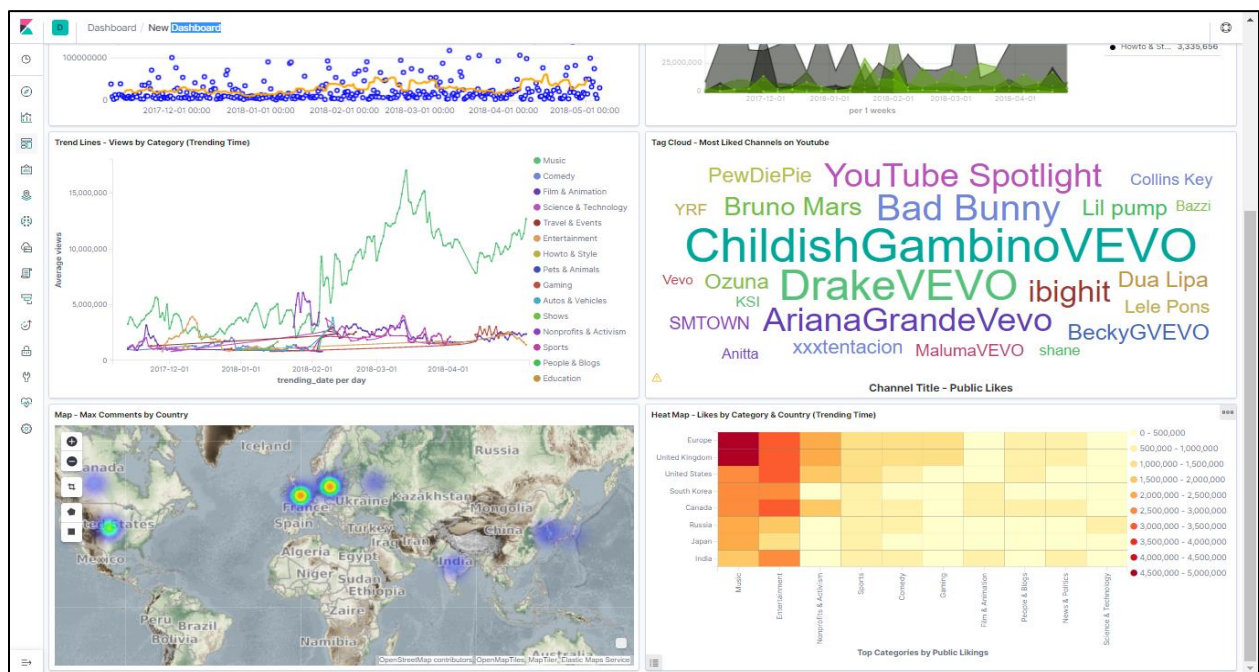
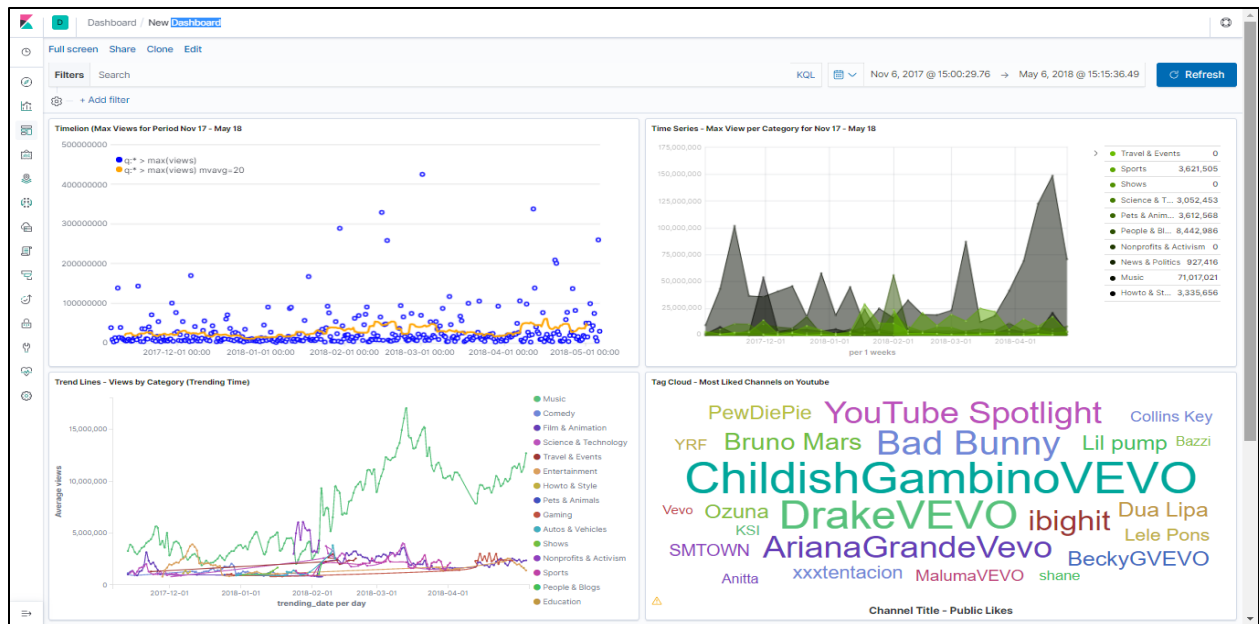
4. Click Add.



5. To add a visualization, select its name from the list of visualizations or click Add new visualization to create one. If you have many visualizations, you can filter the list.



6. To add a saved search, click the Saved Search tab, and then select a name from the list.
7. When you're finished adding and arranging the dashboard content, go to the menu bar, click Save, and enter a name. Optionally, you can store the time period specified in the time filter by selecting Store time with dashboard.



References

1. URL of Data Source, <https://www.kaggle.com/datasnaek/youtube-new>
2. URL of your Github : <https://github.com/tanvigawade/ElasticSearch-Kibana.git>
3. <https://www.elastic.co/guide/en/kibana/current/tutorial-visualizing.html>
4. <https://www.elastic.co/blog/getting-started-with-hosted-elasticsearch-and-a-sample-dataset>
5. <https://www.elastic.co/guide/en/kibana/current/tutorial-load-dataset.html>
6. <https://www.elastic.co/guide/en/kibana/current/tilemap.html>
7. <https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations.html>
8. <https://discuss.elastic.co/t/how-to-import-data-to-elasticsearch/58100/2>
9. <https://www.elastic.co/guide/en/logstash/current/advanced-pipeline.html>