Data Analysis with Elasticsearch and Kibana

Ll. Tobarra, A. Robles, R. Pastor
Dpto. Sistemas de Comunicación y Control
UNED
{llanos,arobles,rpastor}@scc.uned.es

SNOLA

LASI Spain 2018

UNED

Departamento de Sistemas de Comunicación y Control

Outline

- Data visualization
- Installation and introduction
- Data handle: Elasticsearch
- Data visualization: Kibana
- Machine Learning: X-Pack

DATA VISUALIZATION

More than graphs

Data visualization

Data base

OPEN DATA

Data aggregation
EDAs

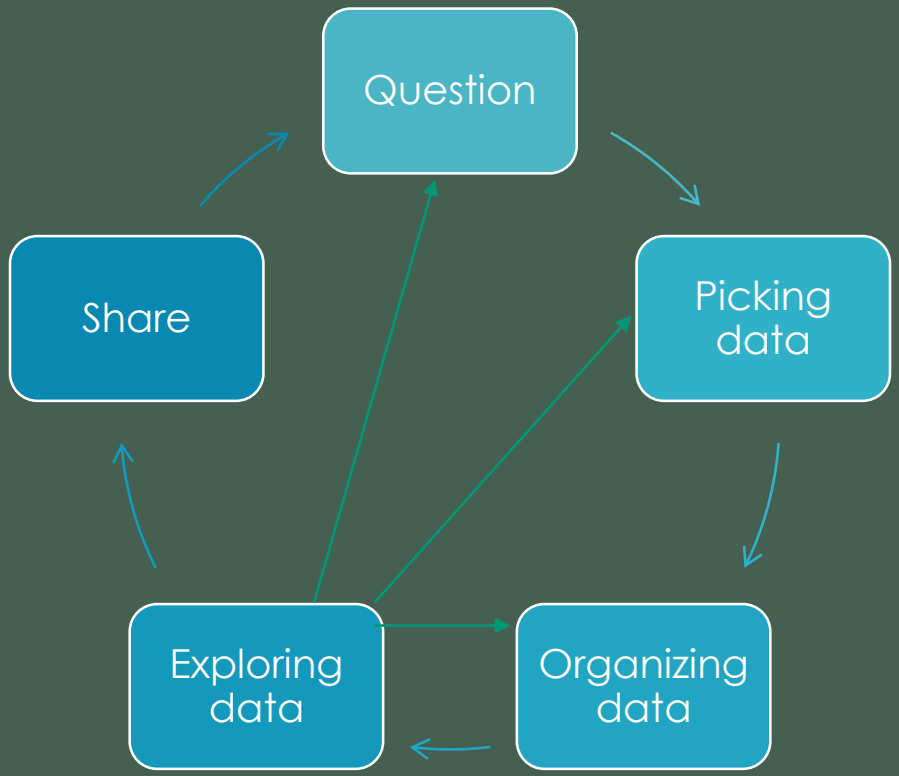
Machine Learning
Statistics

Models

Tracking

Dashboards

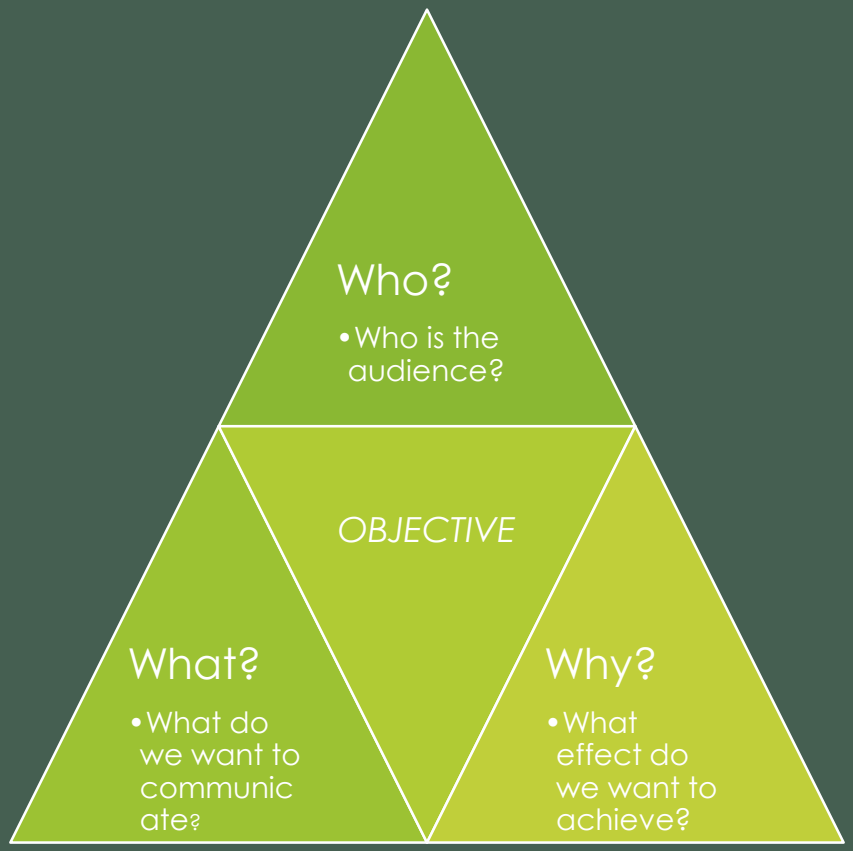
Creating visualizations



Visualization principles

1. Clarify your objective
2. Use the suitable data
3. Select the suitable visualizations
4. Design in an attractive way
5. Choose the most suitable channel of communication
6. Verify results

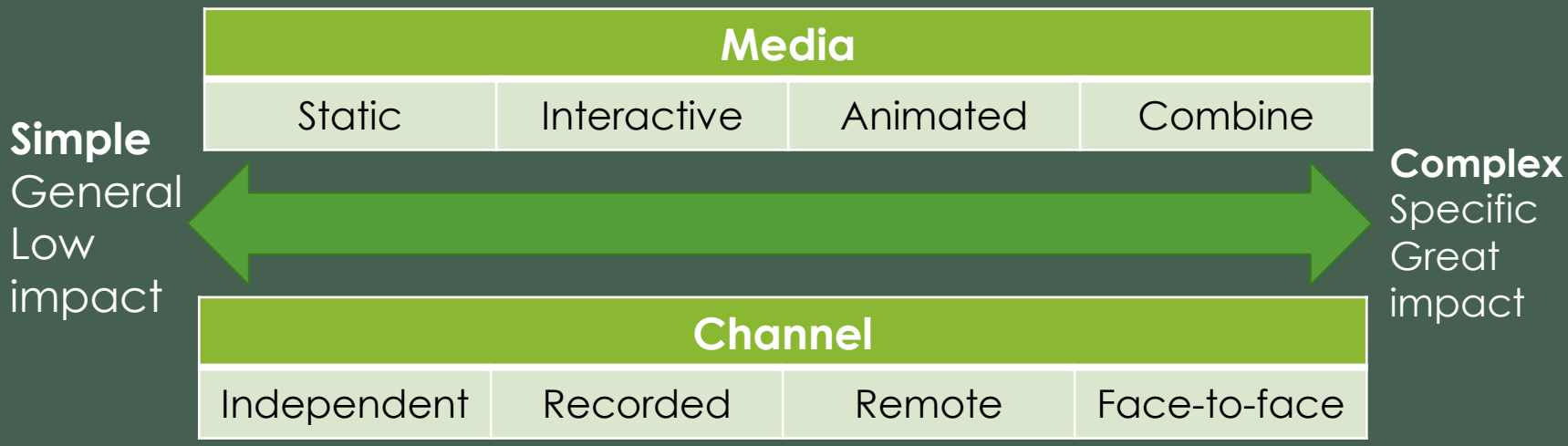
Objective of visualization



Efective codification

Type	Cuantitative	Ordinal	Nominal
Definition	Precise numerical values	Elements with an order	Group or class members
Effective codification	Position Length Angle Area Gradient gray Gradient color Color tone	Position Gradient gray Gradient color Color tone Length Angle Area	Position Shape Color tone Gradient gray Gradient color Length Angle Area

Communication channels



Evaluation of visualizations: RUI

- Include feedback:
  - Reach: Has the audience understood the message? Who does and does not?
  - Understanding: Have you interpreted the message in the same way that we proposed it?
  - Impact: Has the expected reaction been achieved?

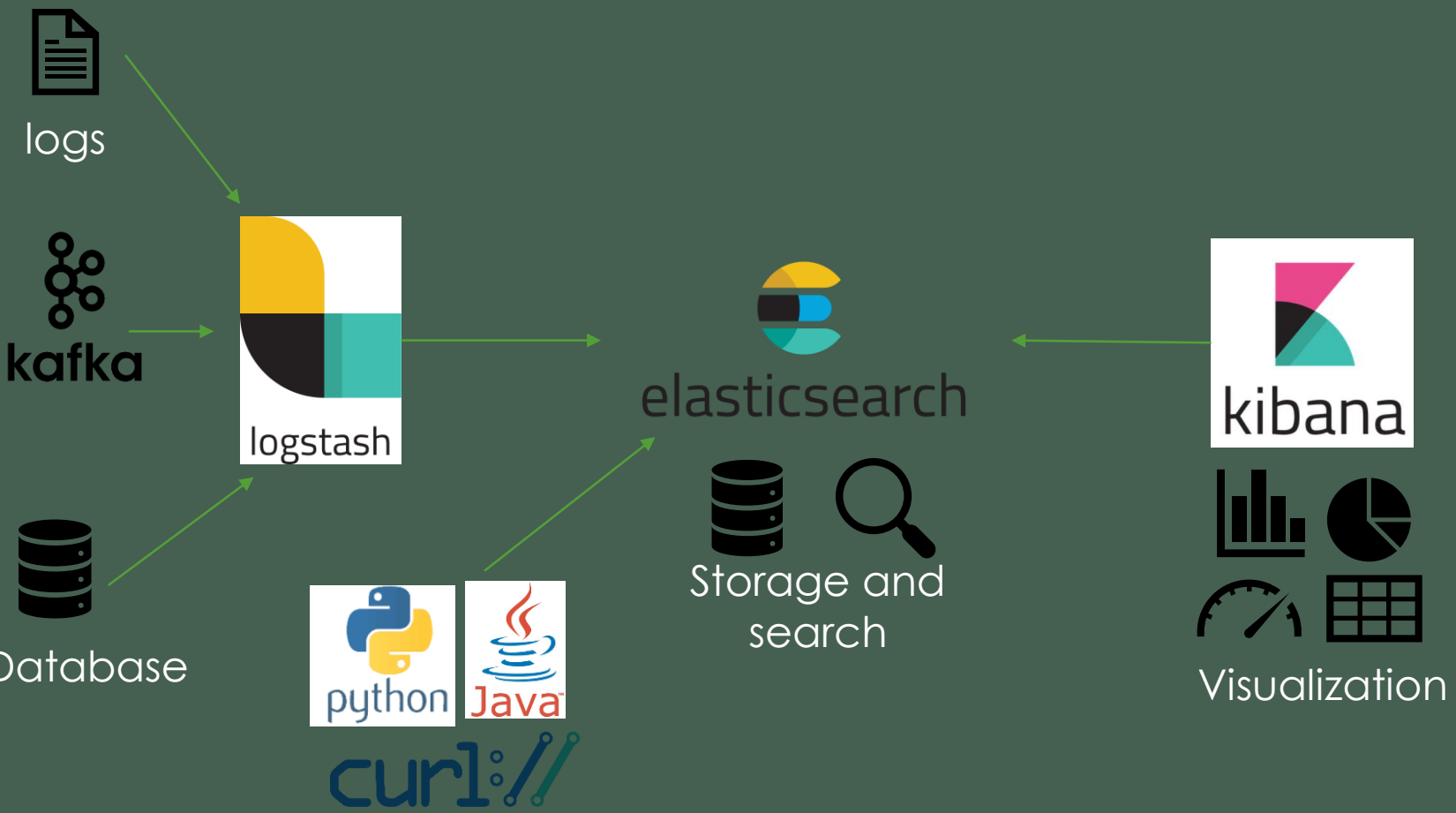
INSTALLATION & INTRODUCTION

I LOVE LOGS

LOGS as a source of information

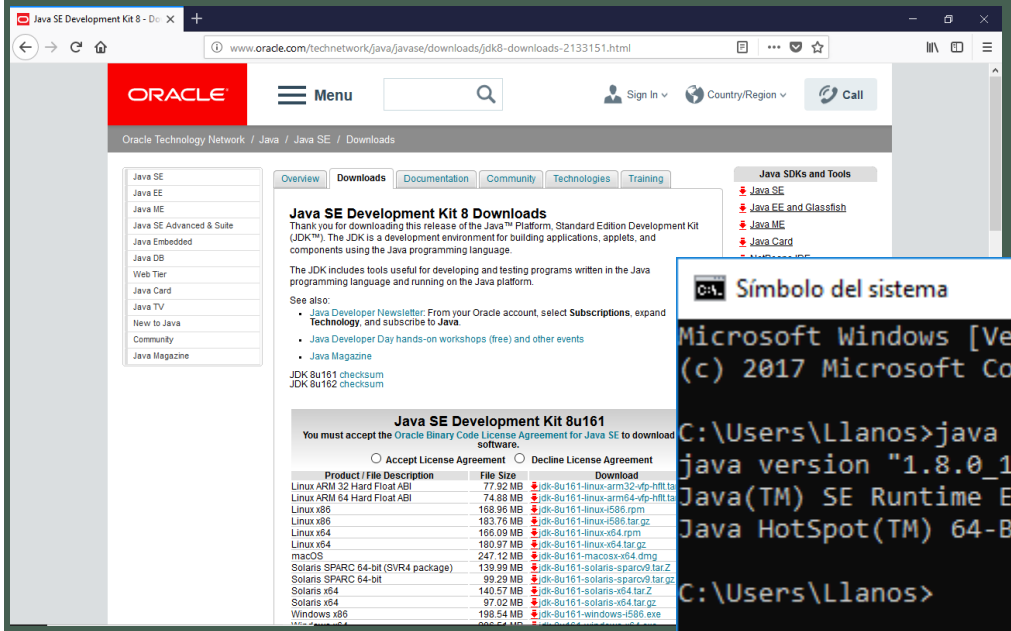
- Distributed semi-structures data
- Huge amount of data that is not easy
- Relevant information

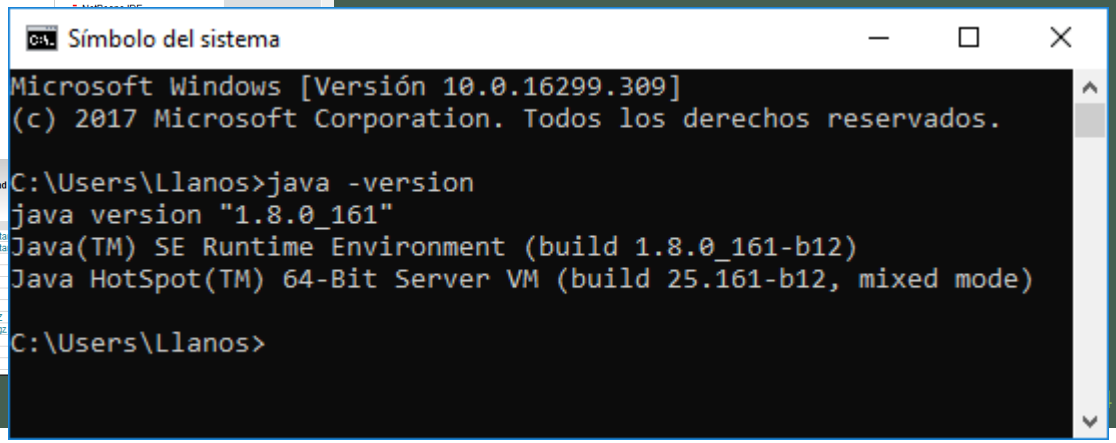
ELK Stack


logs
kafka
Database
logstash
python
Java
curl://
elasticsearch
Storage and search
kibana
Visualization

Installation: prerequisites

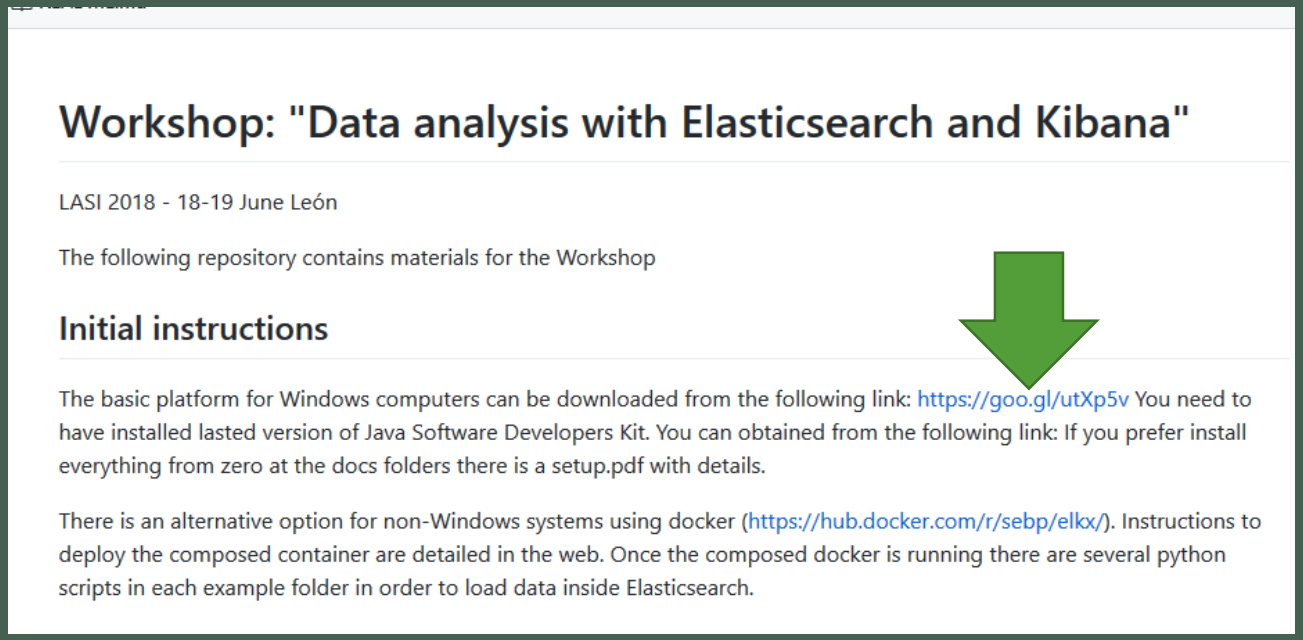
- We need install Java 7 or higher in JDK version (development kit)
  - Download: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html





Download zip and extract

- URL https://github.com/LlanosTobarra/LASI2018
- The zip contains a full installation ready to use



Workshop: "Data analysis with Elasticsearch and Kibana"

LASI 2018 - 18-19 June León

The following repository contains materials for the Workshop

Initial instructions

The basic platform for Windows computers can be downloaded from the following link: https://goo.gl/utXp5v You need to have installed lasted version of Java Software Developers Kit. You can obtained from the following link: If you prefer install everything from zero at the docs folders there is a setup.pdf with details.

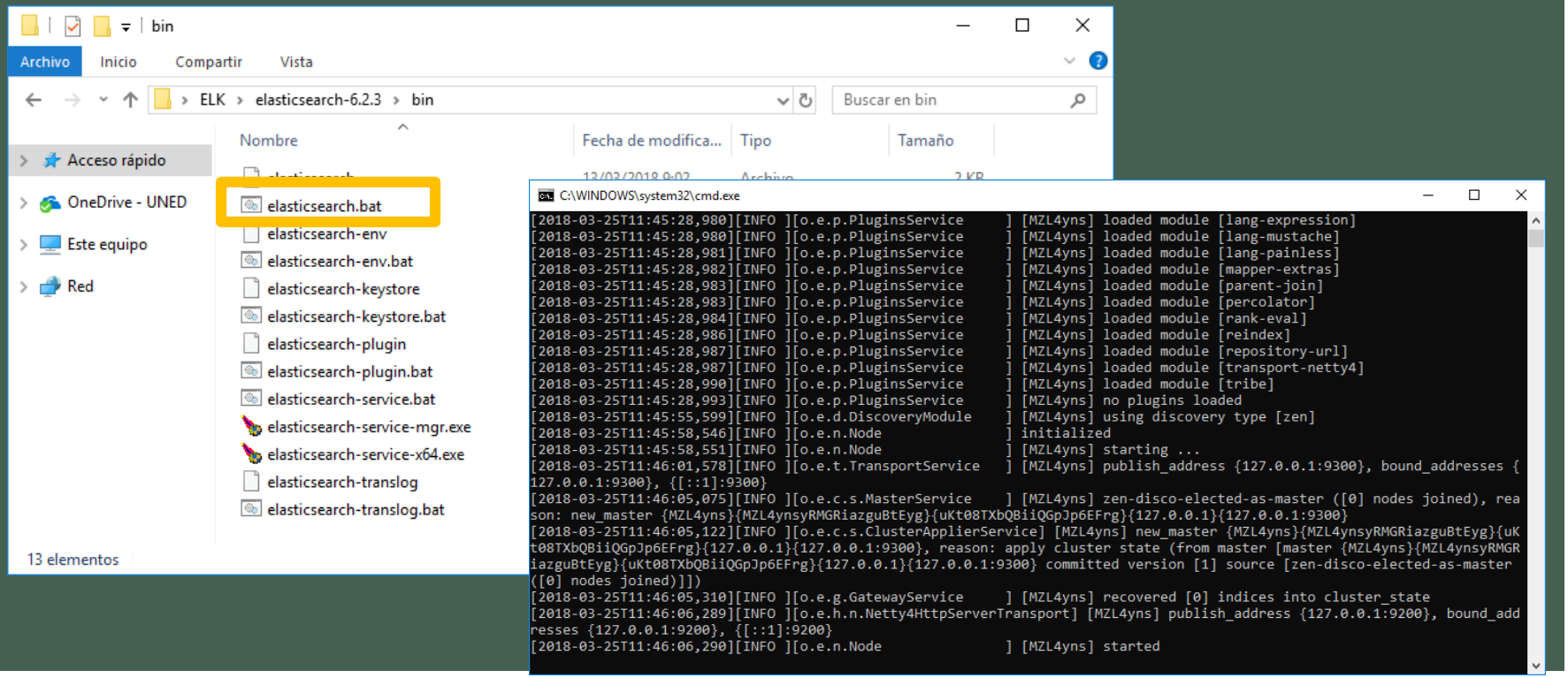
There is an alternative option for non-Windows systems using docker (https://hub.docker.com/r/sebp/elkx/). Instructions to deploy the composed container are detailed in the web. Once the composed docker is running there are several python scripts in each example folder in order to load data inside Elasticsearch.

ALTERNATIVE

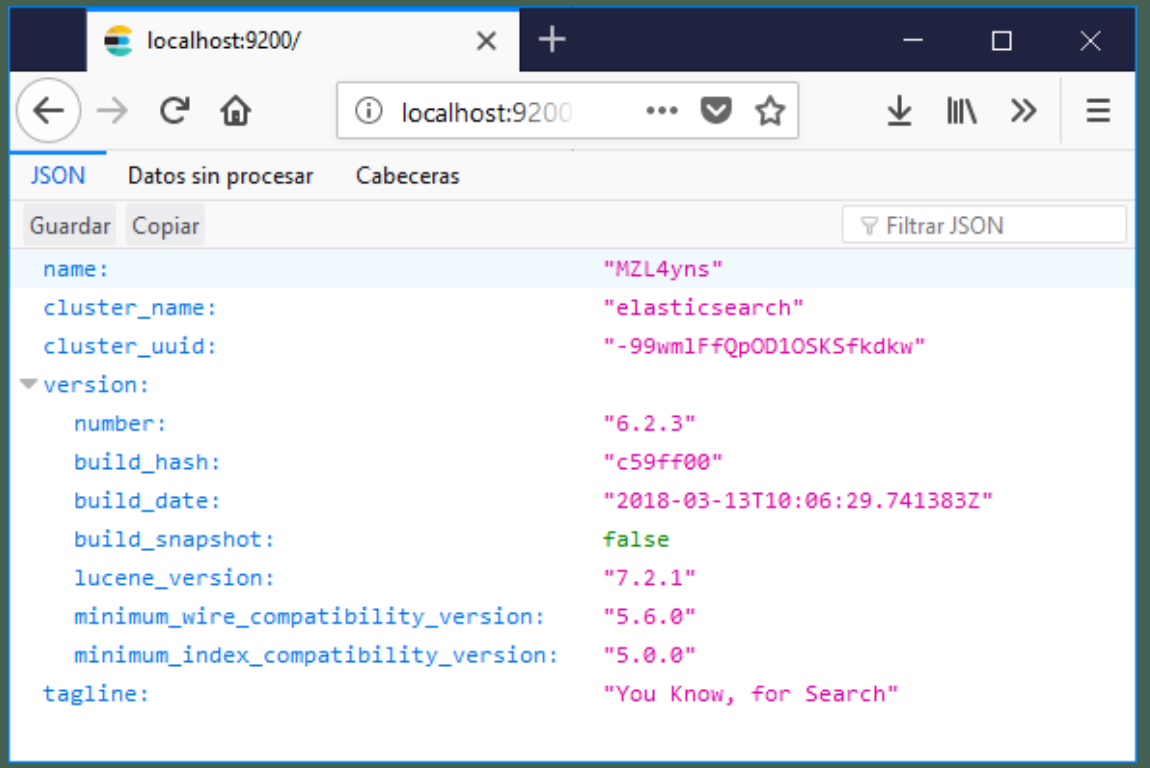
Raise your hand and we will provide you with a username and password to an online version

First steps

- Running ELK/elasticsearch-X.X/bin/elasticsearch.bat



Checking

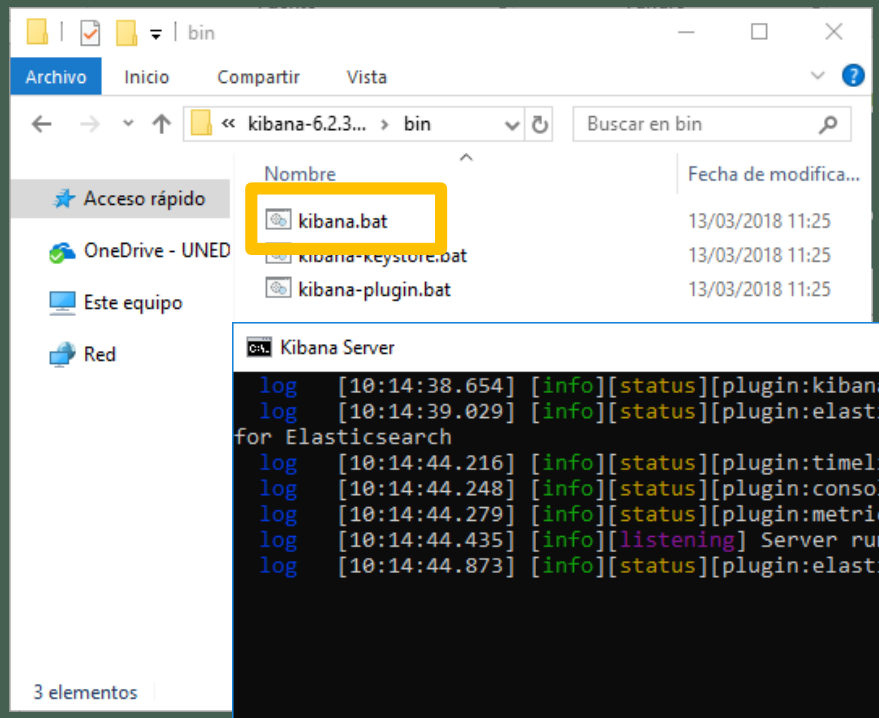


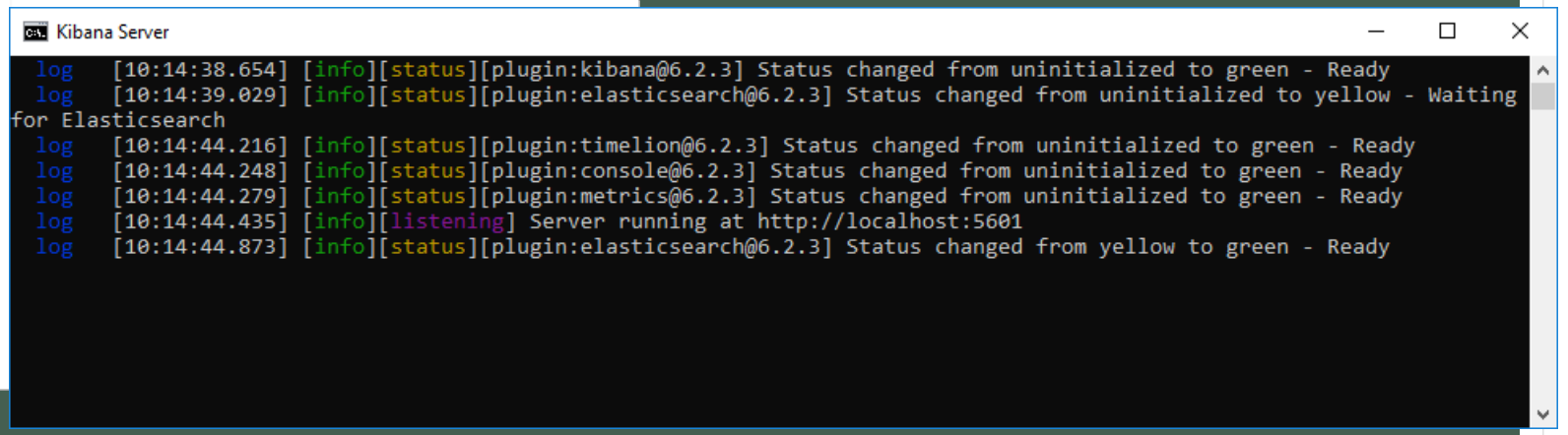
OPEN HTTP://LOCALHOST:9200

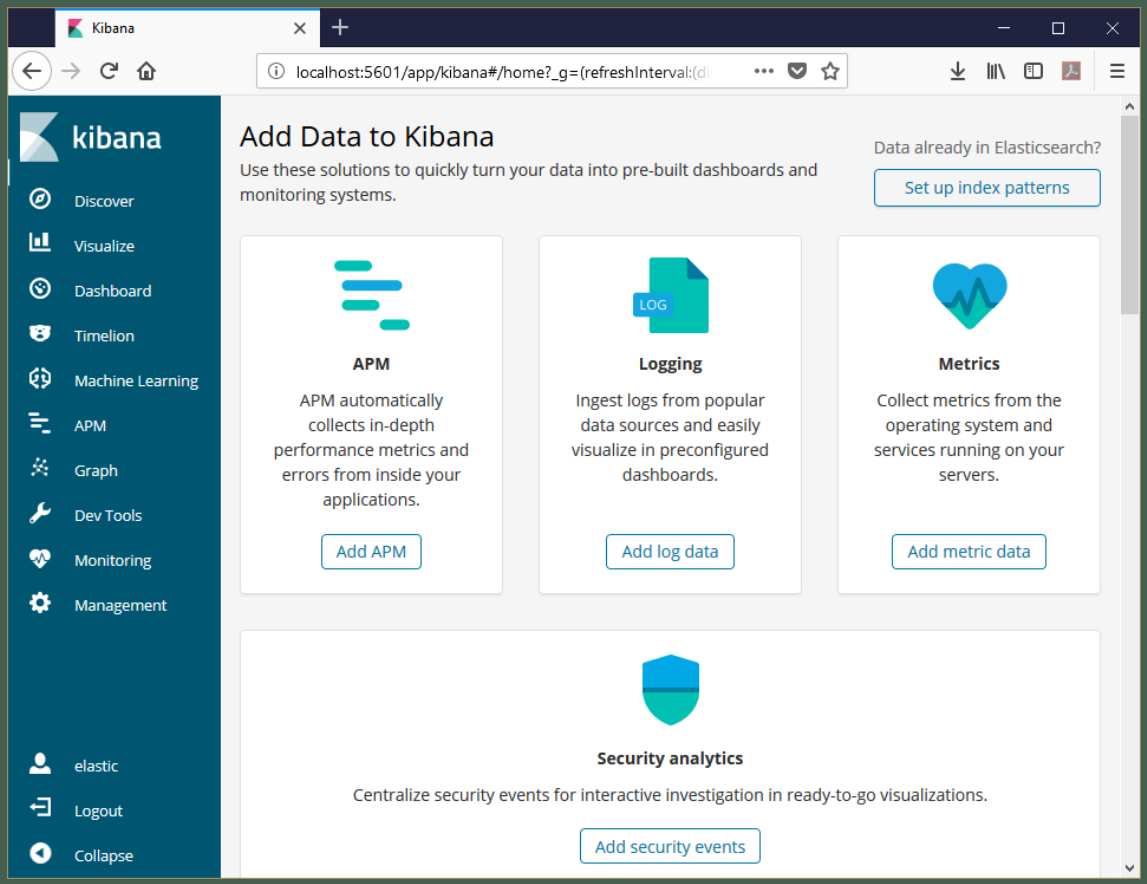
CREDENTIALS ARE NEEDED

First steps

- Running ELK/Kibana-X.X./bin/kibana.bat



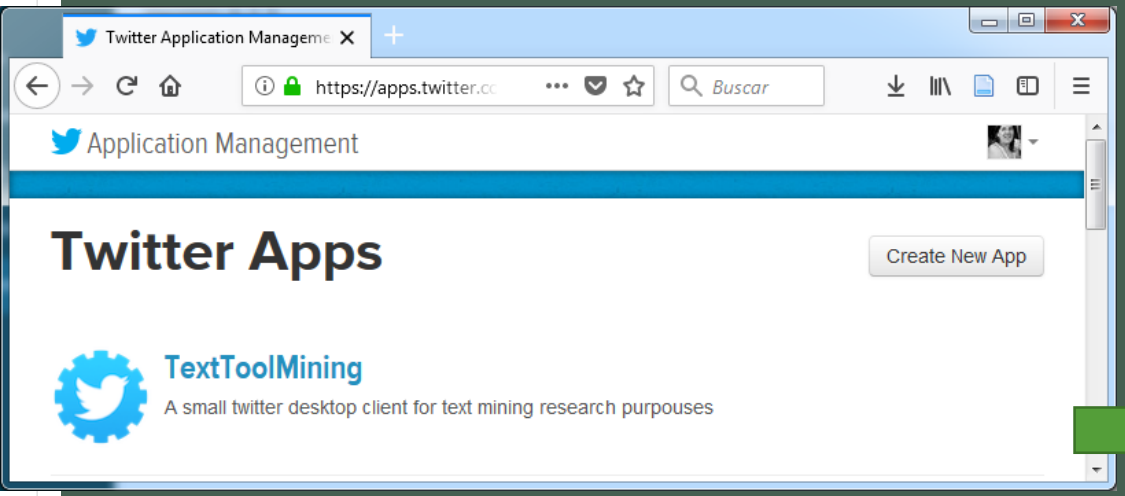


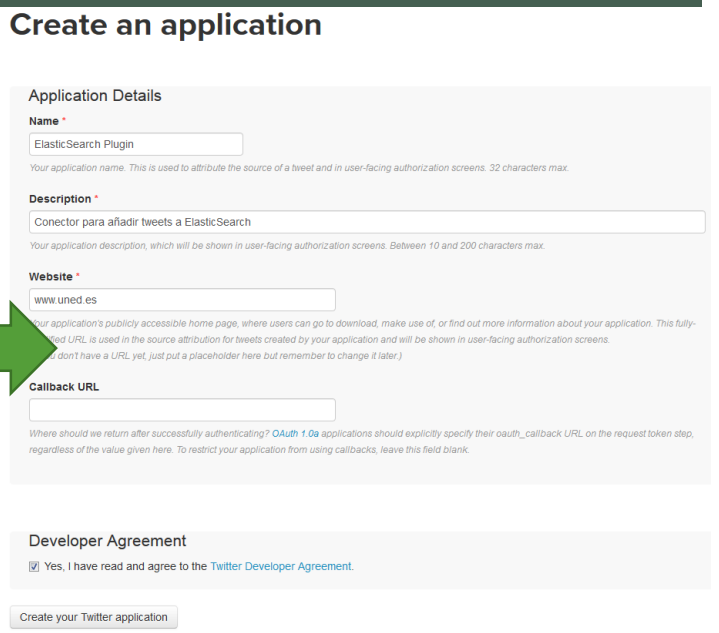


Accessing to Kibana: http://localhost:5601
Use “elastic” user

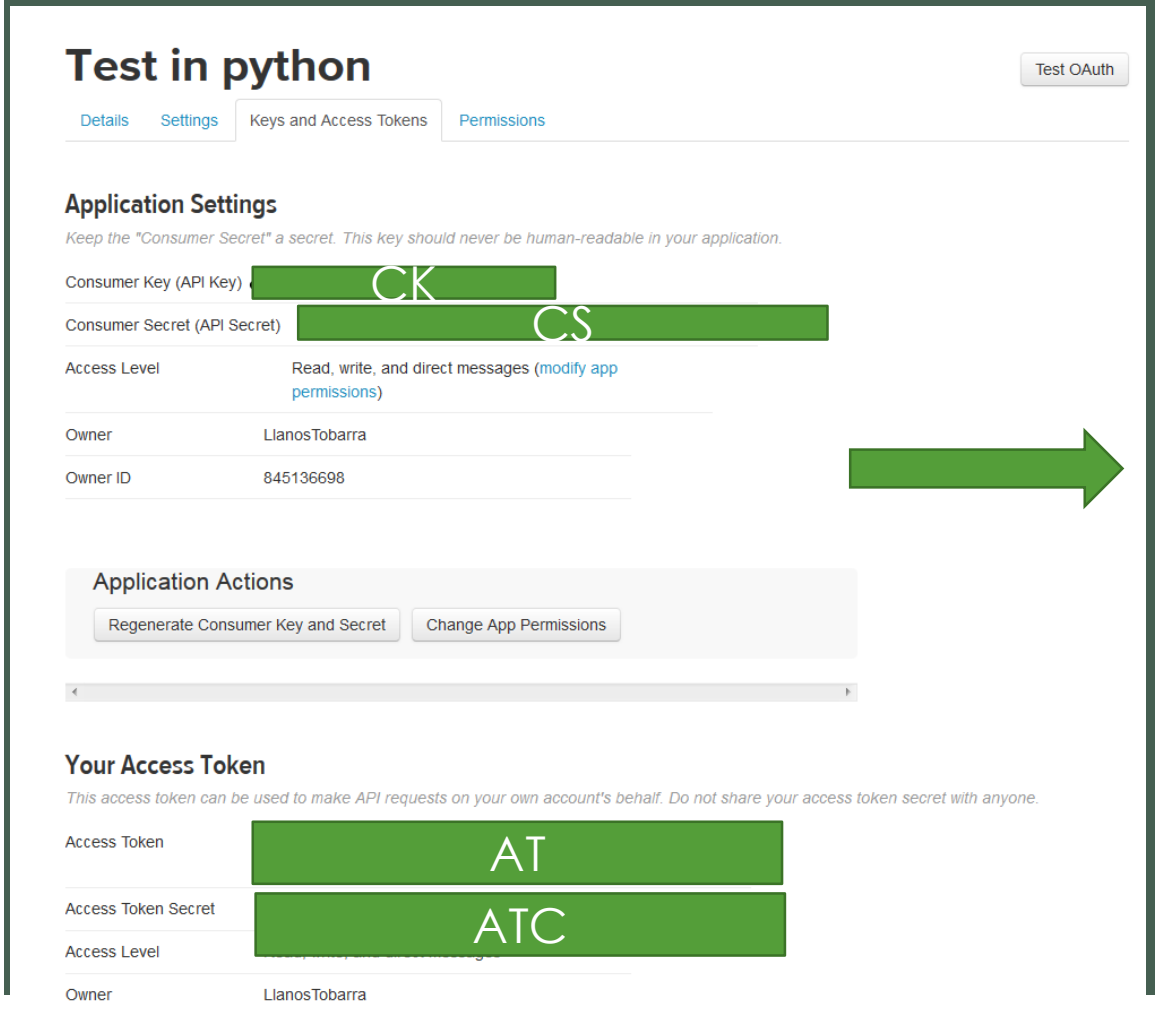
Real time data ingestion: Twitter

- Connecting with the Twitter API Streaming to add tweets in real time
- We need credential of our Twitter account
- Go to https://apps.twitter.com/,and "Create new App"





Twitter: Security tokens



Add to the Python script our credentials

consumer_key=CK
consumer_secret=CS
access_token=AT
access_token_secret=ATC

Educational datasets

- Student's Academic Performance Dataset: **xAPI-Educational Mining Dataset**
  - **Source:** https://www.kaggle.com/aljarah/xAPI-Edu-Data/version/2
  - Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining Educational Data to Predict Student's academic Performance using Ensemble Methods. International Journal of Database Theory and Application, 9(8), 119-136
  - Amrieh, E. A., Hamtini, T., & Aljarah, I. (2015, November). Preprocessing and analyzing educational data set using X-API for improving student's performance. In Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on (pp. 1-5). IEEE

Data fields:

- Timestamp
- Actor name. In case of student is an integer
- Role: staff or student
- Action performed: submit, create, post ,and view
- Object instance identificator
- Object type: forum, link, LTI, wiki, blog, assessment, scorm, quiz, book, video
- Topic: subject topic
- Course id

Syntectic generated data:

- Dataset generated using the script: https://github.com/jiscdev/lakhak
- Used in LAK'2017 and LAK'2018 Hackaton

Non-educational dataset: Star-Wars

- Data extracted from https://swapi.co
- Data from:
  - Characters
  - Planets
  - Films
  - Starships
- More info: https://swapi.co/documentation

ELASTICSEARCH

A distributed database oriented to search

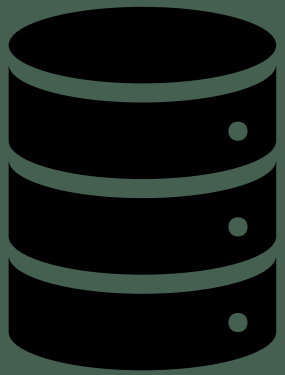
ELASTICSEARCH

- Distributed engine for real-time data analysis
- Open Source solution developed in Java
- Built on the Apache Lucene project as a search engine on which there is a RESTful web access API
- Documentary database with search in all the text instead of tables and columns
- Use for single-page projects

elasticsearch

Funcionality

- Queries
  - Perform and combine the results of queries on structured, semi-structured and unstructured data of different nature (geo, metrics, ...)
  - Free searches: ask what you want
- Analysis
  - Comprehension of logs of millions of lines in a simple way
  - It provides data aggregation mechanisms that facilitate the analysis of trends and existing patterns within the data



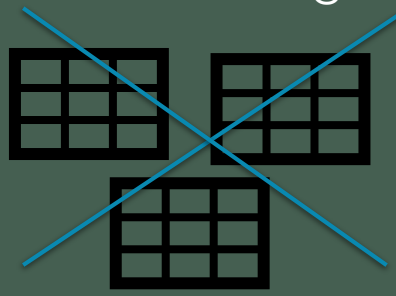
Advantages

Scalabilty

Very fast

Multi-language

Free searches and search suggestions



Non schema

Basic concepts

- Almost real time
  - There is a small delay between when a document is indexed and available for your search (depends on the platform)
- Cluster
  - It is defined as a set of nodes that store the data in a distributed way
  - It provides a federated index and the ability to perform searches across all nodes
  - It is identified with a single name: elasticsearch
- Node
  - It is a single server that is part of the cluster, stores the data and participates in the cluster index and searches

Basic concepts

- Index
  - It is a collection of documents with similar characteristics
  - It is assigned a name, which is used when indexing, searching, updating and deleting the documents contained therein
- Type
  - It is a logical category or a partition of an index whose semantics is complete
  - It is defined by a set of documents that have a set of fields in common
  - You can define more a type within the same index

Basic concepts

- Document
  - It is the basic unit of indexable information
  - The format of the documents is JSON
- Shards
  - Elasticsearch allows you to divide the index into several segments called fragments (shards)
  - Each fragment is a fully functional and independent index that can be stored in a cluster node
- Replica
  - Elasticsearch allows the creation of one or more copies of the index fragments, which are called replicas

API

- The RESTful API of Elasticsearch is accessible using JSON together with the HTTP methods
- Characteristics:
  - Multiple indexes
  - Support of dates in the name of the indexes
  - Common options
  - Access control based on URL

Document
- Load one or several documents

Search
- From several indexes
- Several URIs

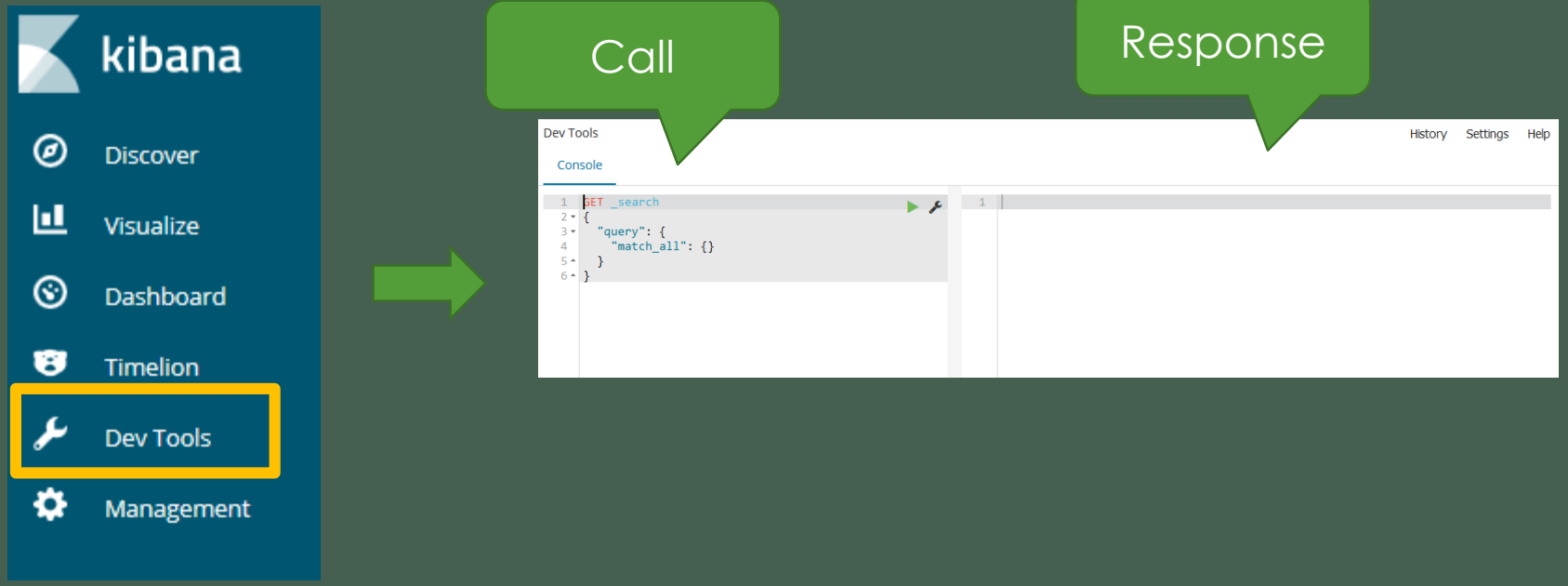
Aggregation
- From data searches

Index
- Management of indexes

Cluster
- Management of nodes
- Heath of nodes

CONSOLE UI

- We will work with the console to communicate with Elasticsearch and execute instructions against its API



CRUD

Operation	Description	Example

PUT /index/type/id { <JSON>}	Add/Update a document to the index	PUT /list/song/6
		{
		"title": "Billie Jean"
		"album":"Thriller"
		"year" : "1984"
		"artist": "Michael Jackson"
		}

GET	Obtain a document from an index	GET /list/song/6
DELETE	Remove a document	DELETE /list/song/6

MAPPING

- It is the process that describes how a document is stored and indexed
- Dynamic Mapping - We can add documents to an index that does not have an associated mapping and ElasticSearch associates the default types with it
- An index can have associated one or more mappings, which are used to divide the documents into logical groups. They consist of Meta-fields, which provide information about mapping and other objects: _index, _type, _id, _source
- Fields defined in the documents and associated types

Type	Supported
Basic	integer, long, double, short, byte, double, float, string, date, Boolean y binary, keyword
Complex	object, nested
Geo	geo_point, geo_shape,…
Special	IPv4, token_count,…

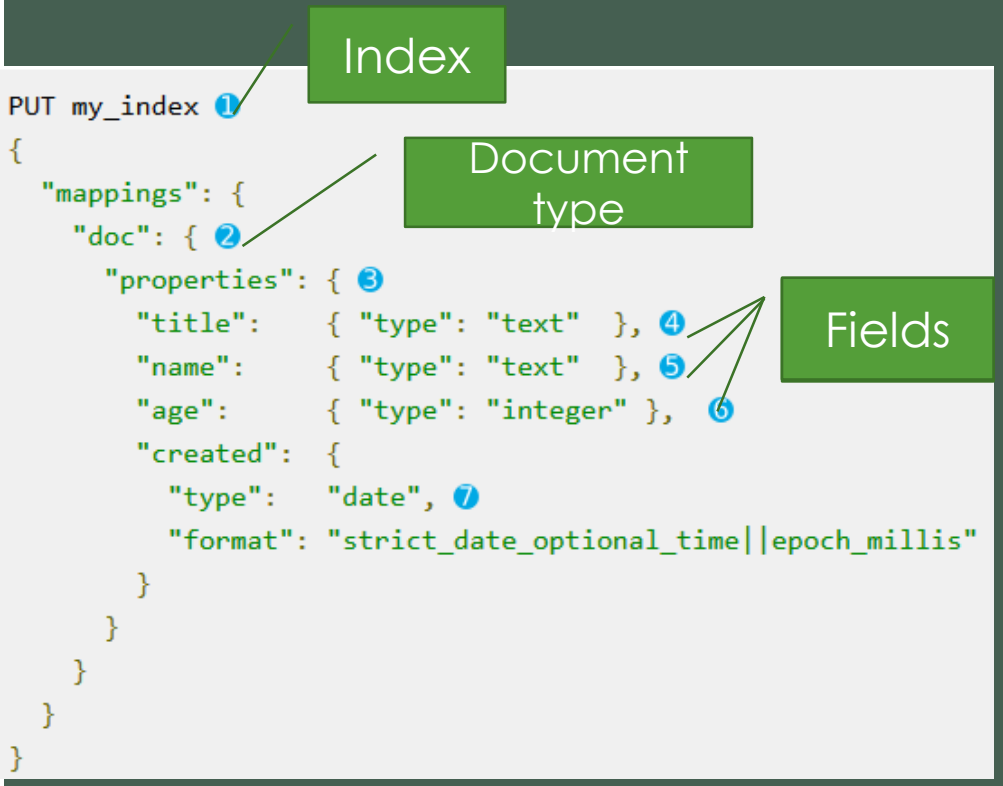
INDEX and MAPPING

```
GET cheyenne
```

```
{
  "cheyenne": {
    "aliases": {},
    "mappings": {
      "registry": {
        "properties": {
          "Account-Name": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
            }
          },
          "Acct": {
            "properties": {
              "5": {
                "type": "long"
              },
              "6": {
                "type": "long"
              }
            }
          },
          "AcctGroup": {
            "type": "text",
            "fields": {
              "keyword": {
                "type": "keyword",
                "ignore_above": 256
              }
```

- GET <index>
  - GET /<index>/_mapping/_doc
- HEAD <index>
- DELETE <index>

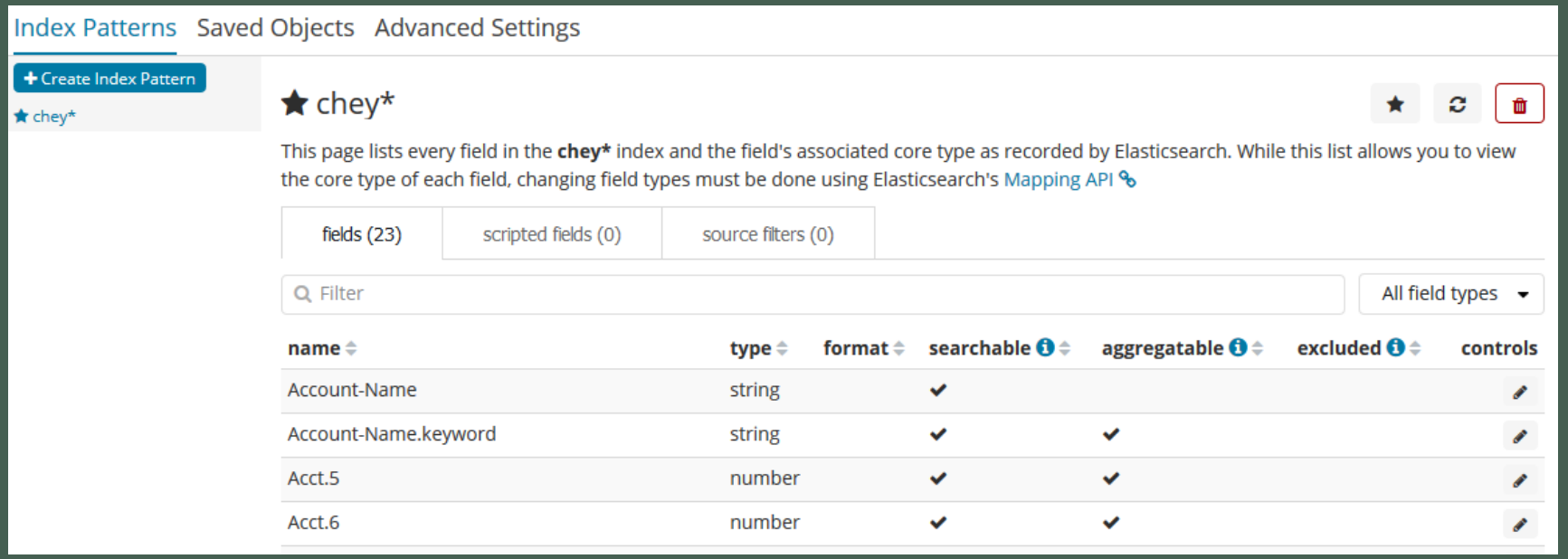
MAPPING update



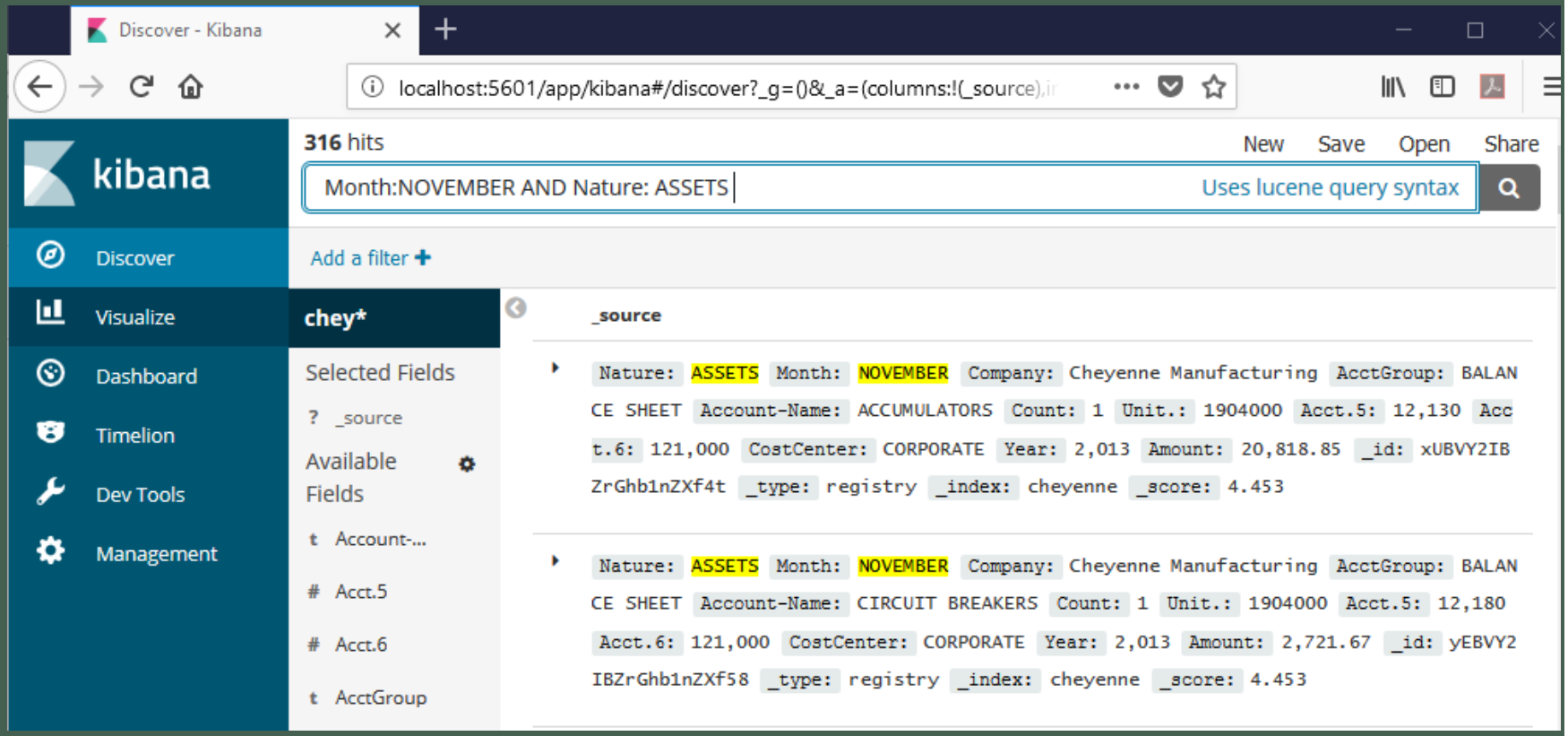
- A text field can be marked as text for textual searches and as a keyword for aggregations and sorting
- It is defined when the index is created. We can redefine it with another PUT command
- Beware of defining too many types for a field: we can generate a state explosion and cause a lot of errors

KIBANA edition

Management > Index Patterns



Text queries



Queries: QUERY_STRING

- Search data in:
  - Operators AND, OR, NOT. For example, Madrid OR Barcelona
  - We can also use + to indicate that it should appear and - in that it should not appear. For example, "restaurant + cheap -vegetarian"
  - We can specify the attribute to search with field: value. For example city: Madrid
  - We can use wildcards and regular expressions (between / and /)
  - _exists_: field checks if that field exists in the document
  - Similar terms using ~, for example camp ~ (distance Damerau-Levenshtein)
  - We can indicate a range by [min TO max] for example date: [2013-01-01 TO 2013-12-01]

Regular expressions

Expression	Description	Example
.	Any character	Abc. → Abcd, Abce,…
+	One or more occurrences of the regular expression	A+ → A, AA, AAA,AAAA,…
*	None or more occurrences of the regular expression	A* → ‘’, A, AA, AAA,AAAA,…
?	None or one occurrences of the regular expression	A? → ‘’, A
{min,max}	Minimum and maximum number of occurrences	A{2,5} → AA,AAA, AAAA,AAAAA
()	Expression aggrupation	(AB)+→AB, ABAB,ABABAB,…
|	One element or another	A | B → A , B
[a-b]	Range of elements	[A-Z] → A, B, C, …, Z
~	Denial, which does not contain the regular expression	~A→ B, C, …,Z
<i-z>	Interval	A<1-5> → A1, A2, A3, A4, A5
@	Any chain	

Reverse index

- For the text fields, an inverse index is created that speeds up searches

1) Mondays are very hard
2) Fridays are better

Word	Document
mondays	1
fridays	2
are	1,2
very	1
hard	1
better	2

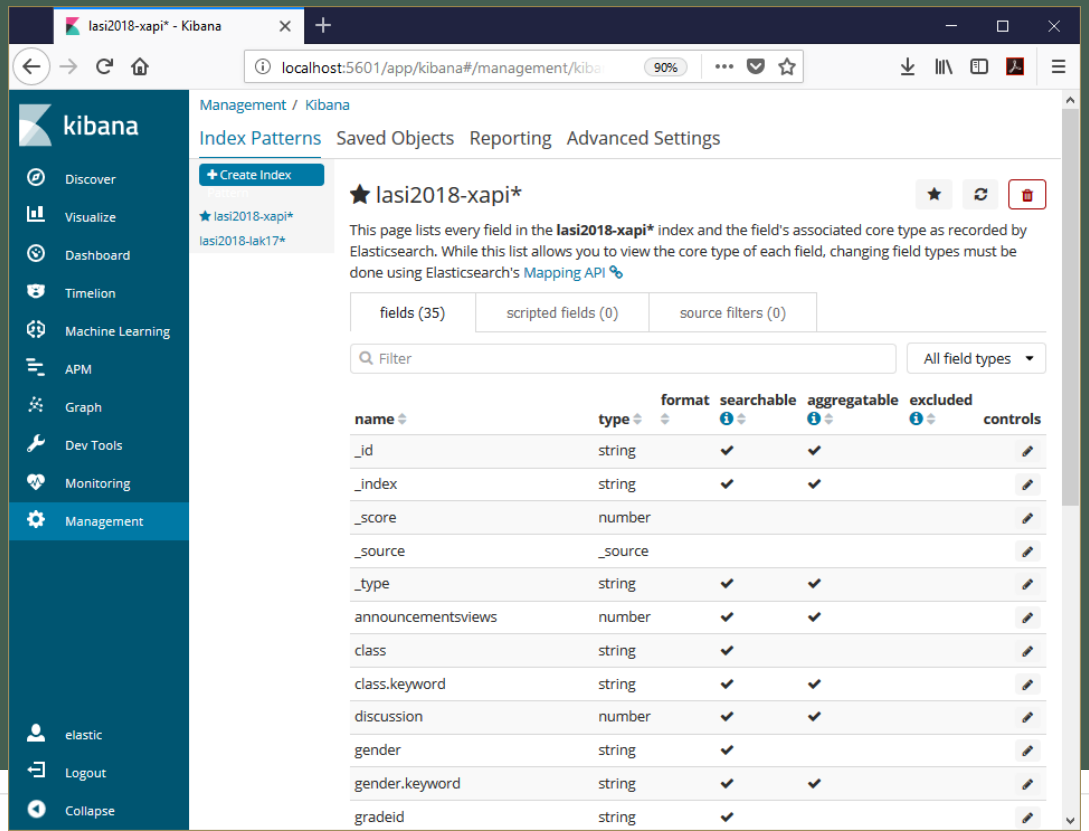
There will be terms that are considered significant because they are not frequently found in all documents

KIBANA

Data visualization

Management

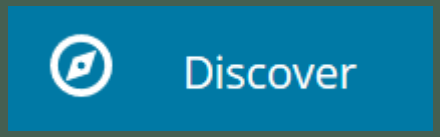
- First time, we must create index patterns
- Each index that matches the regular expression of a index pattern is included in this pattern name



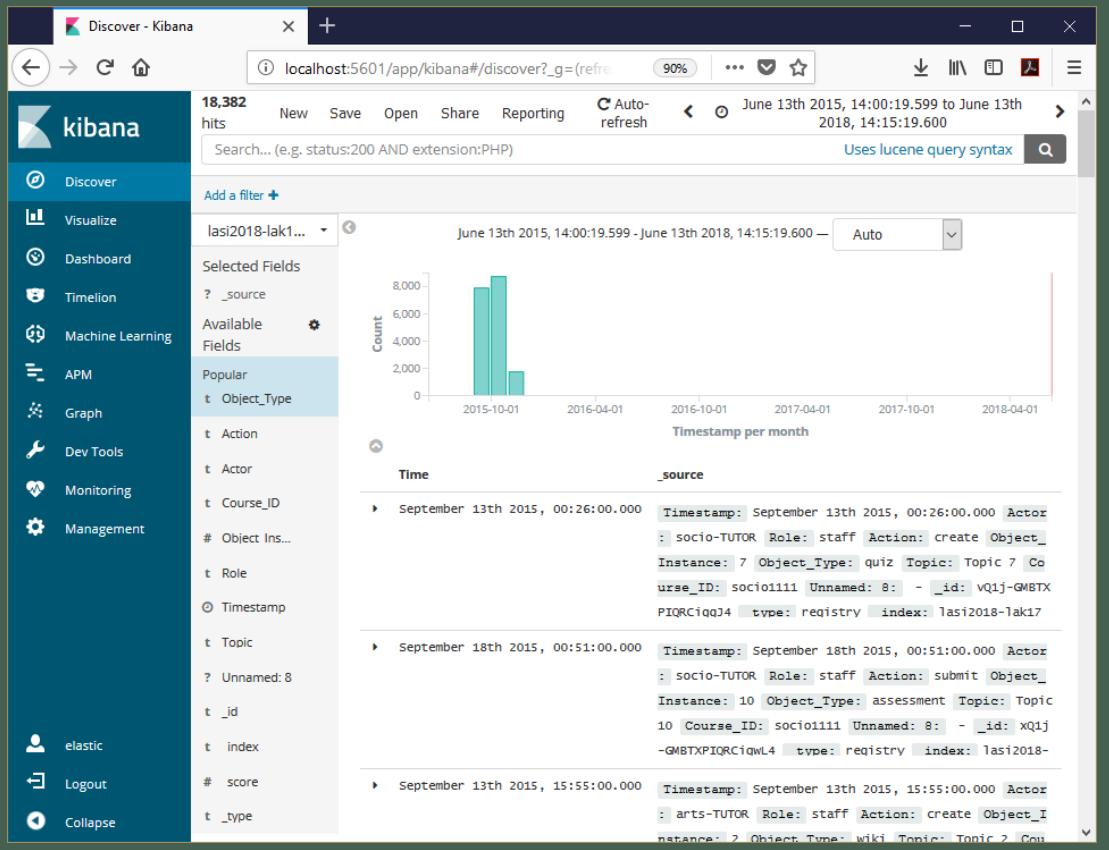
In addition, from management, we can handle users and roles

And review the plugin configuration

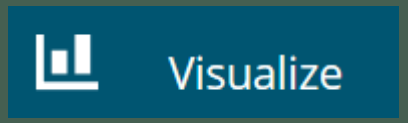
Discover



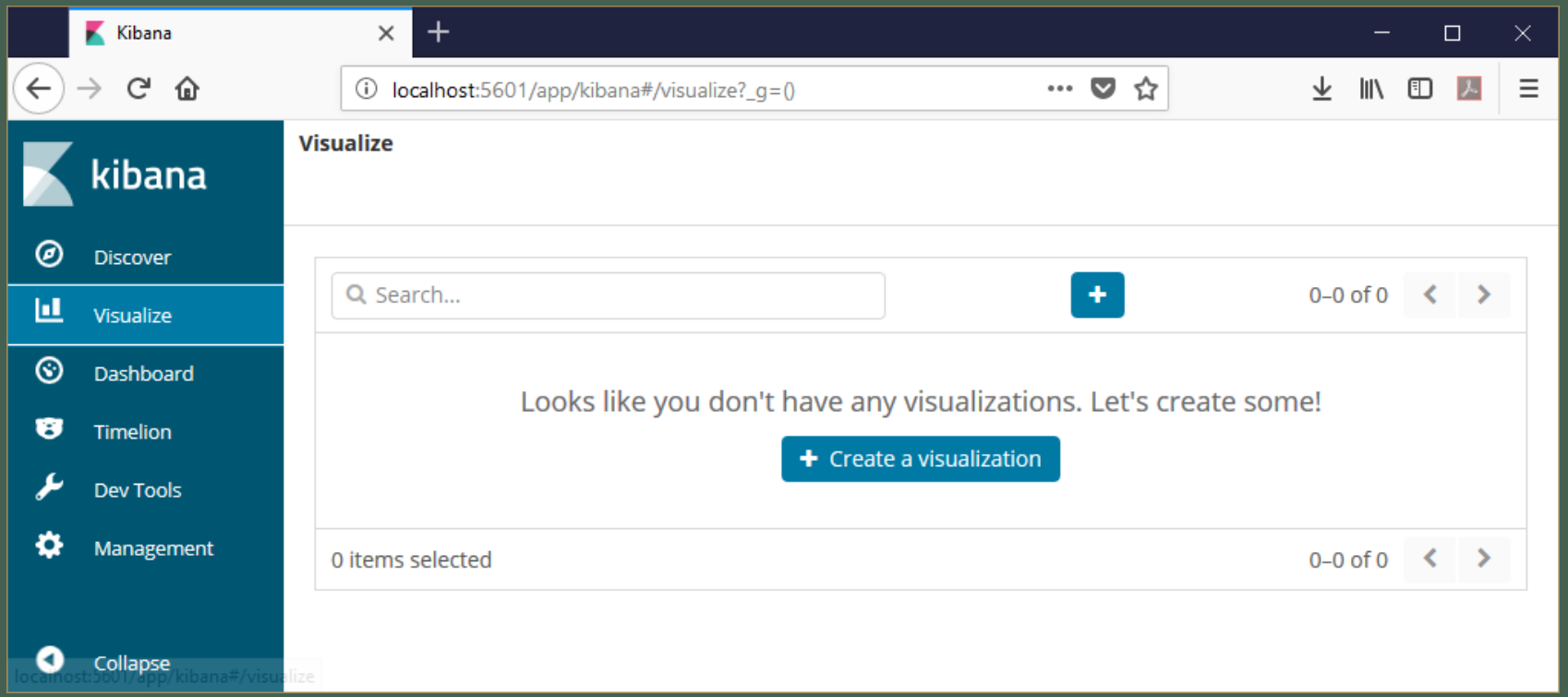
- Once we have created at least one index pattern we can use Discover tool in order to know more about the data
- Filters and queries



Visualize

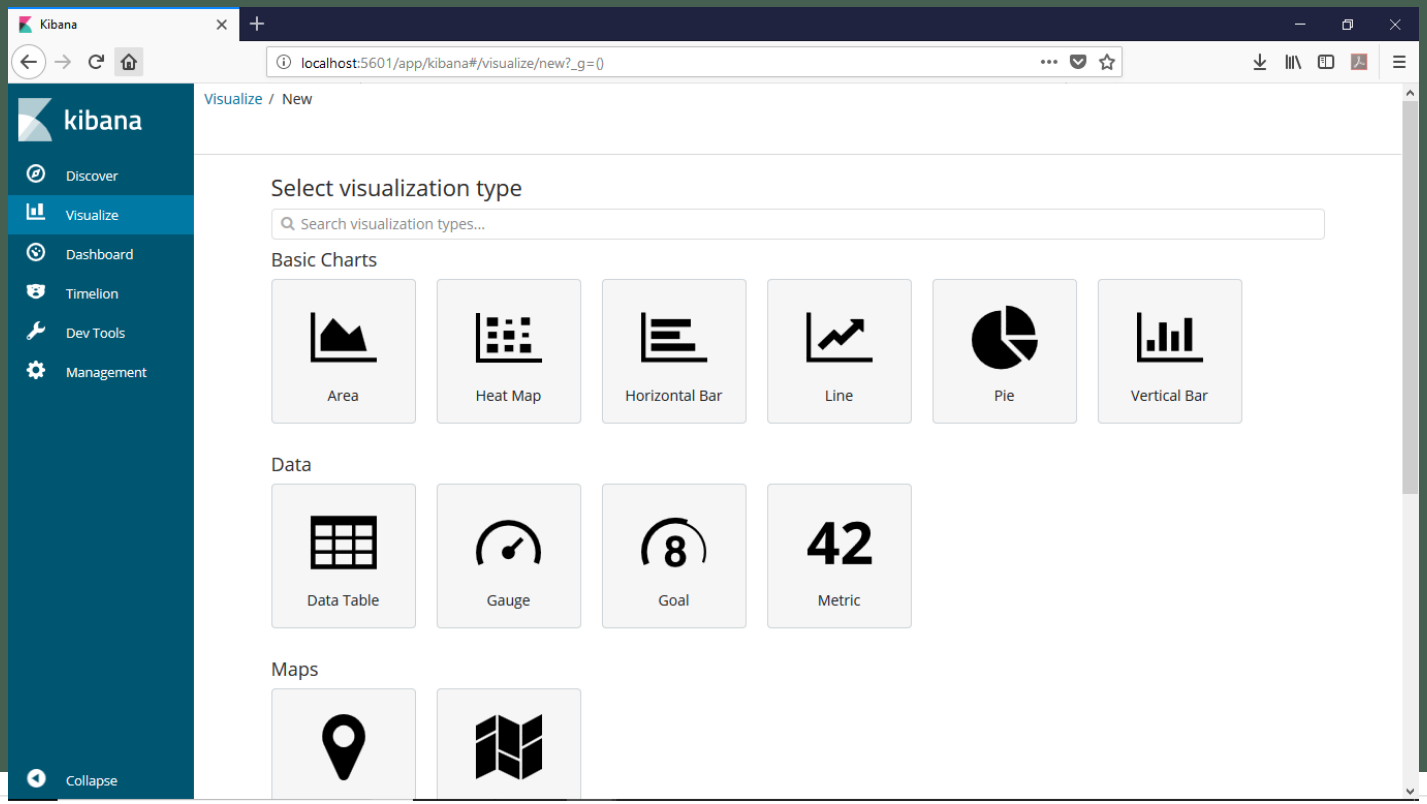


- Visualize option allows us to create basic graphs



Types of graphs

- Once we have choosen "Create a graph", we should select the type of graph



Compare values

- Columns
- Bars
- Circular area
- Line
- Scatter plot
- Bullet

Elements compositions

- Circular
- Stacked bars
- Columns stacked
- Area
- Waterfall

Data distribution

- Dispersion
- Lines
- Columns
- Bars

Data tendencies

- Lines
- Lines with two axes
- Columns

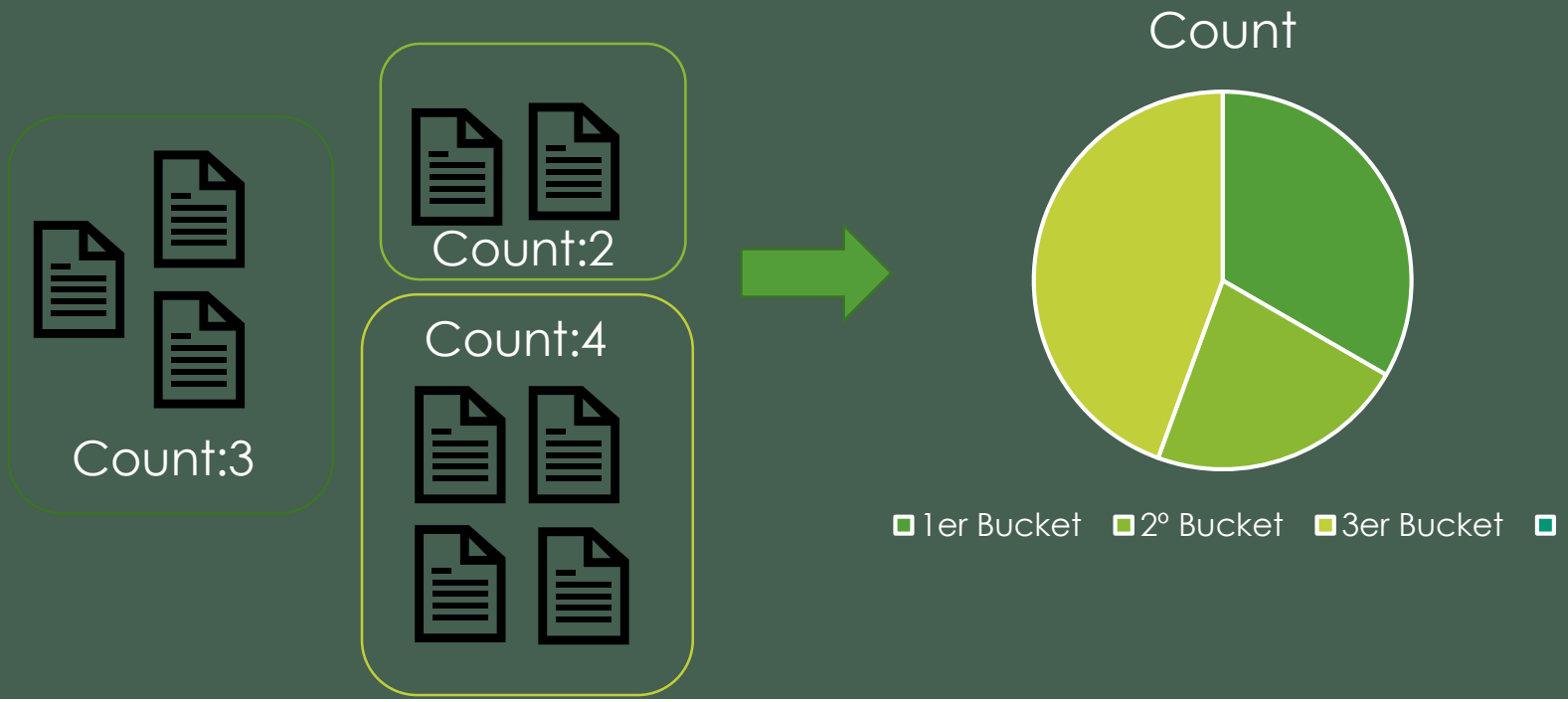
Relation among data groups

- Dispersion
- Bubbles
- Lines

A type of graph for each use case

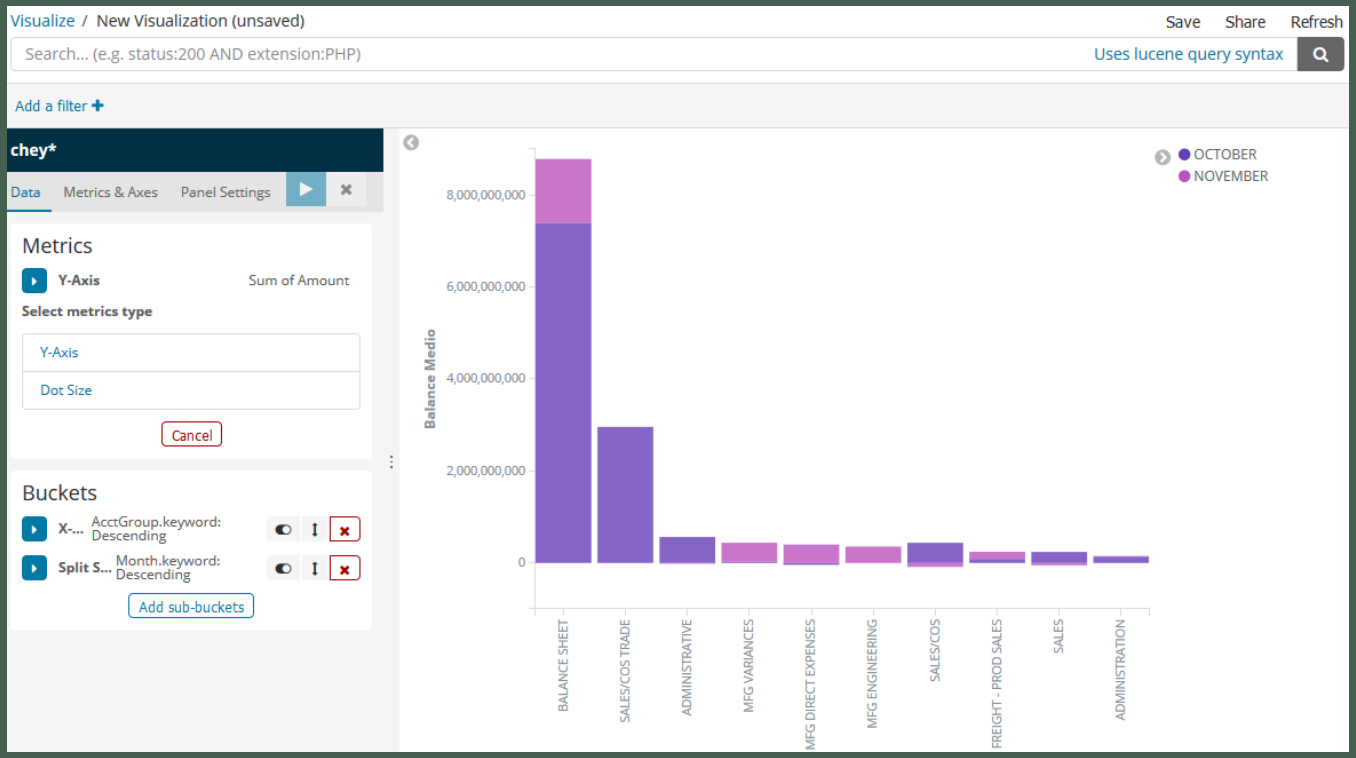
Aggregations

- Bucket: is a criteria that allows us to group documents
- Metrics: is the function that must perfomr inside each document bucket



Visualize: graph definition

- Save: store graph
- Share: HTML is provided so we can integrate the graph into a webpage

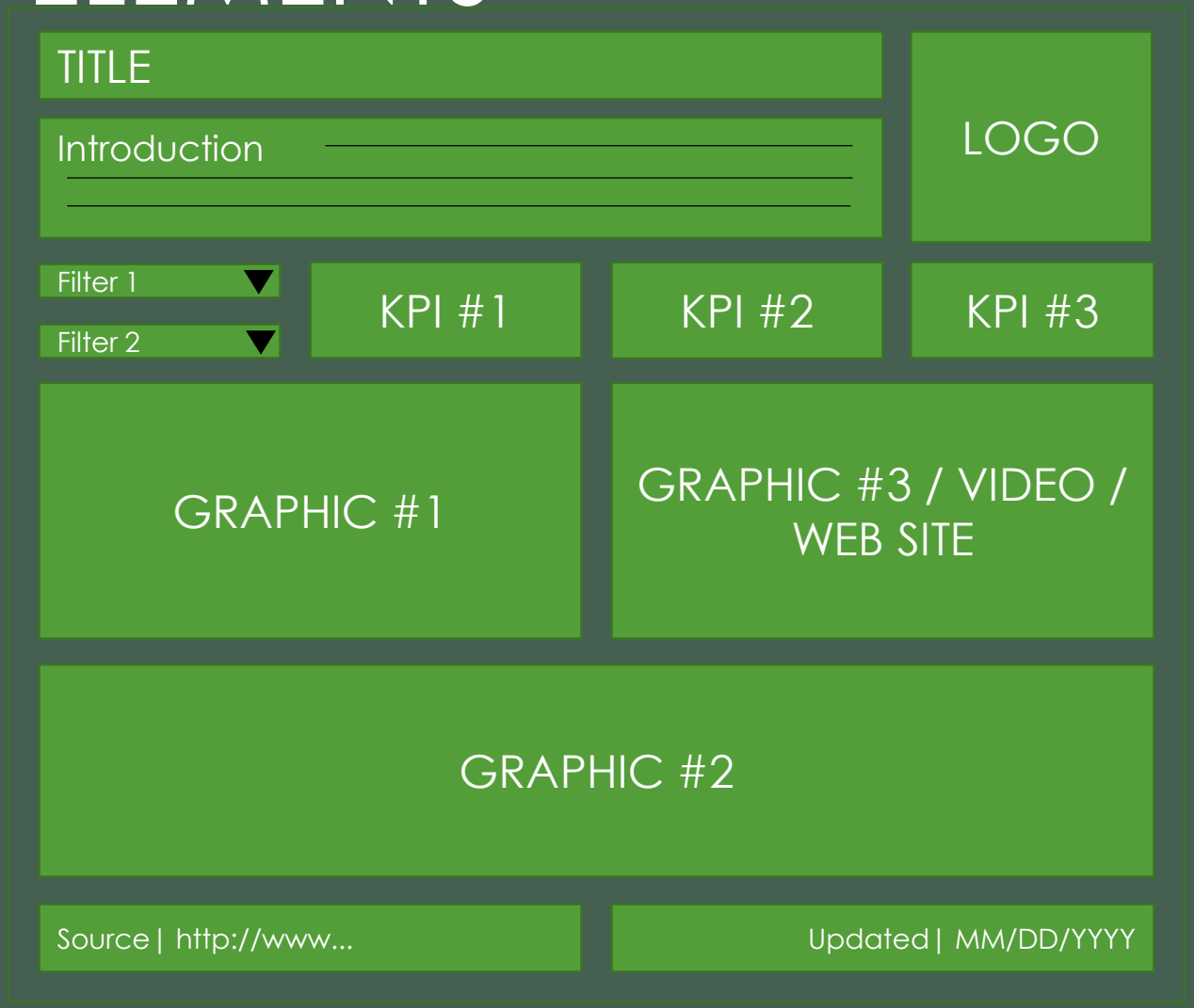


DASHBOARDS

- Definition:

It is a graphic representation of the main indicators (KPI) that intervene in the achievement of business objectives, and is aimed at making decisions to optimize the strategy of the company

ELEMENTS



- Descriptive texts: titles, paragraphs, references and attributions
- KPIs or summary values
- Filters, to allow questions
- Actions, filter or highlight sheets
- Multimedia: logos, videos, web pages

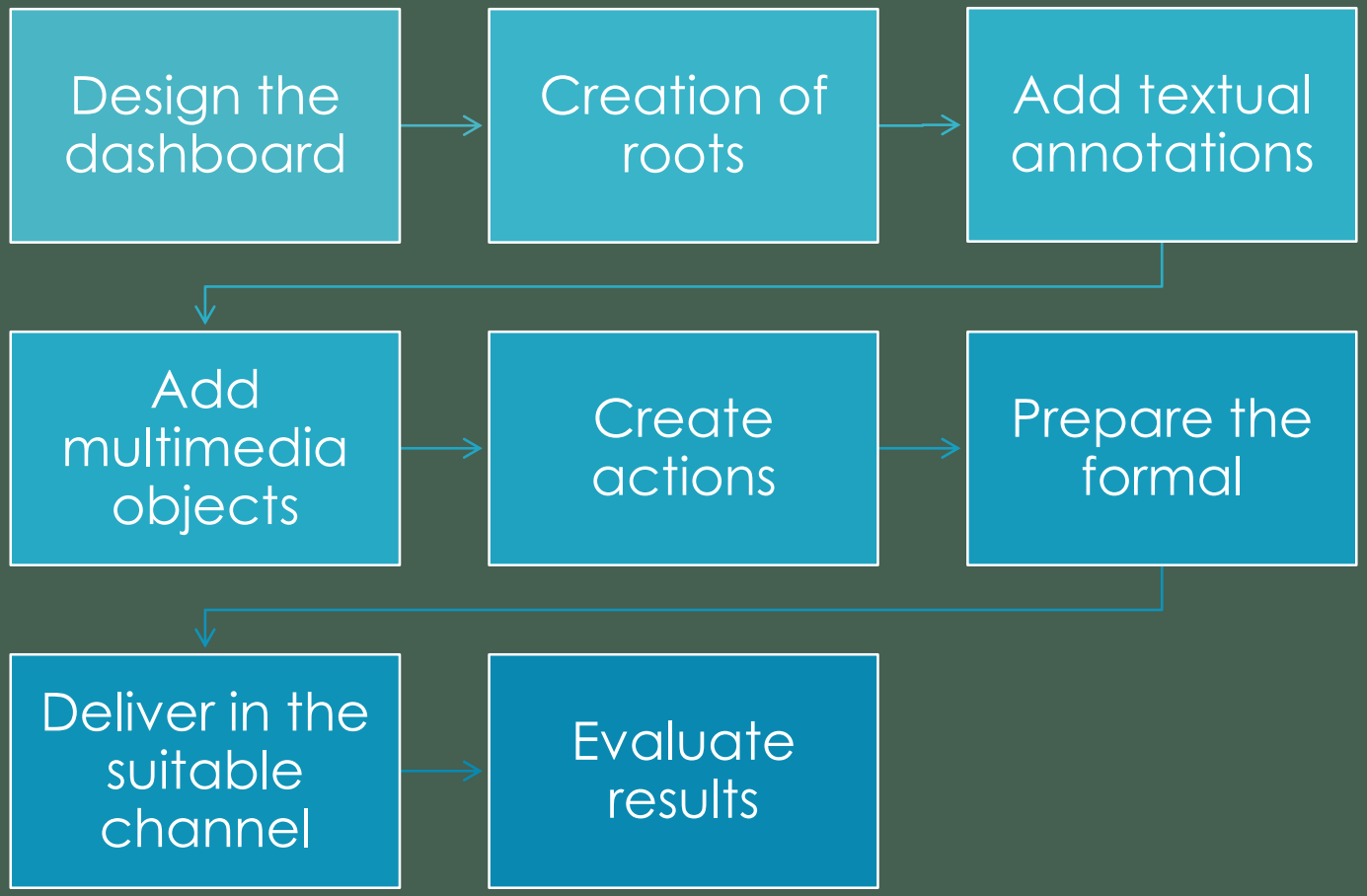
Dashboards types

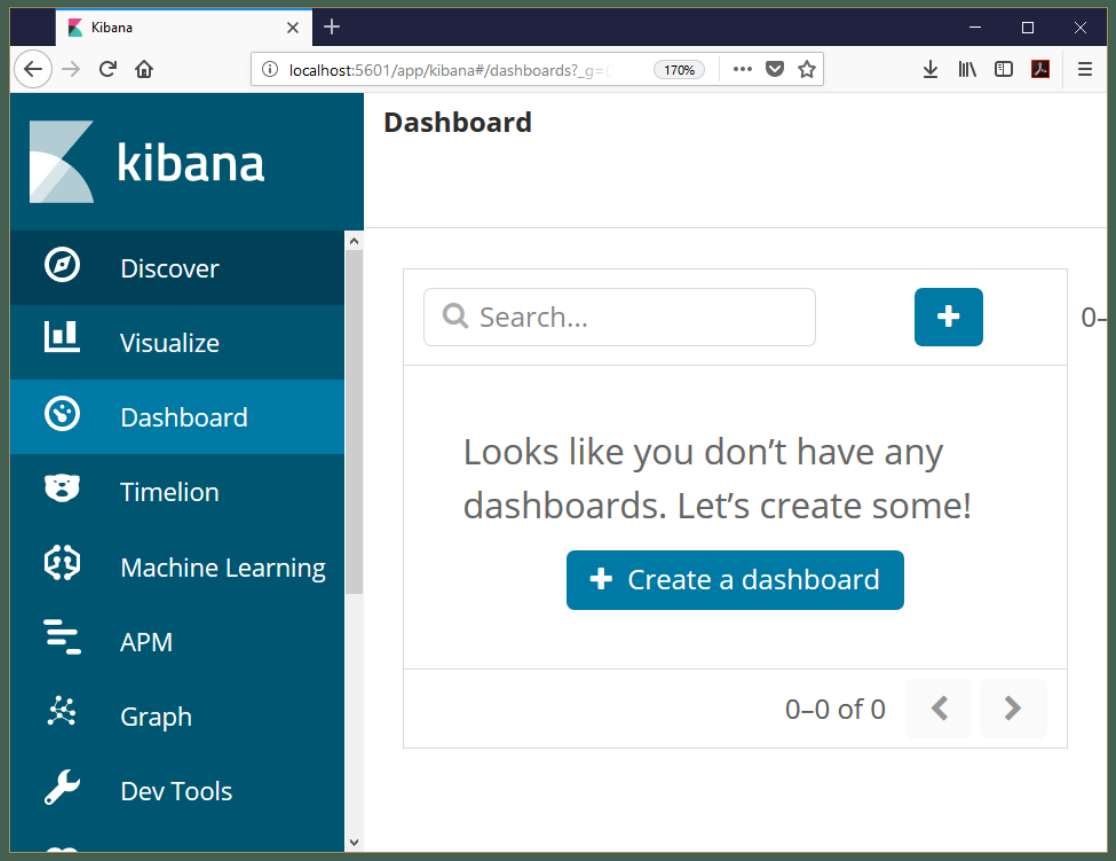
Explanation	Facts about a subject to educate the audience
	Statics
Exploration	Start in a subject and answer the questions that arise
	Very interactive
Both	
Historical	Flow of events or evolution of a situation over time
Infographics	Set of facts about a subject, in column format

Context

Decisions/Analysis	Data journalism	Open data
• Achieve objectives • Business Intelligence • Ej: Web traffic to a blog	• Report about news • Delve into a hot topic • Attractive	• Repository converted in visualization • Educate readers

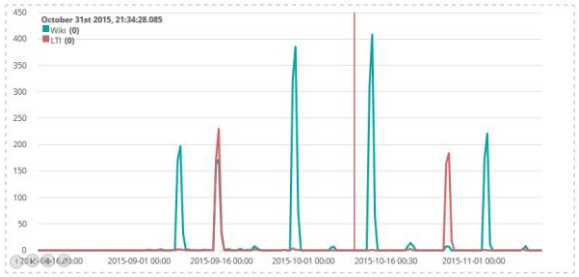
Workflow


Design the dashboard
Creation of roots
Add textual annotations
Add multimedia objects
Create actions
Prepare the formal
Deliver in the suitable channel
Evaluate results



Dashboard

Dashboards



Timelion

Timelion

- Timelion is an visualization tool for time series in Kibana
- Time series visualizations are visualizations, that analyze data in time order
- Parameters:
  - Index: pattern index in order to represent data
  - Timefield: is the name of the field inside the index that contains dates
  - q: Lucene query that reduces the amount of data to represent
  - metric: is the aggregation function (count, max, min, avg…)
- Example: .es(index=lasi2018-*lak17*,timefield=Timestamp,q='wiki',metric=count:Topic)

Timelion

- Other functions:
  - Label: name of the series
  - Title: graph title
  - Color: changes the color of the line. It accepts an hexadecimal color value
  - Legend: determines the shape and the position of the graph legend
- Example:

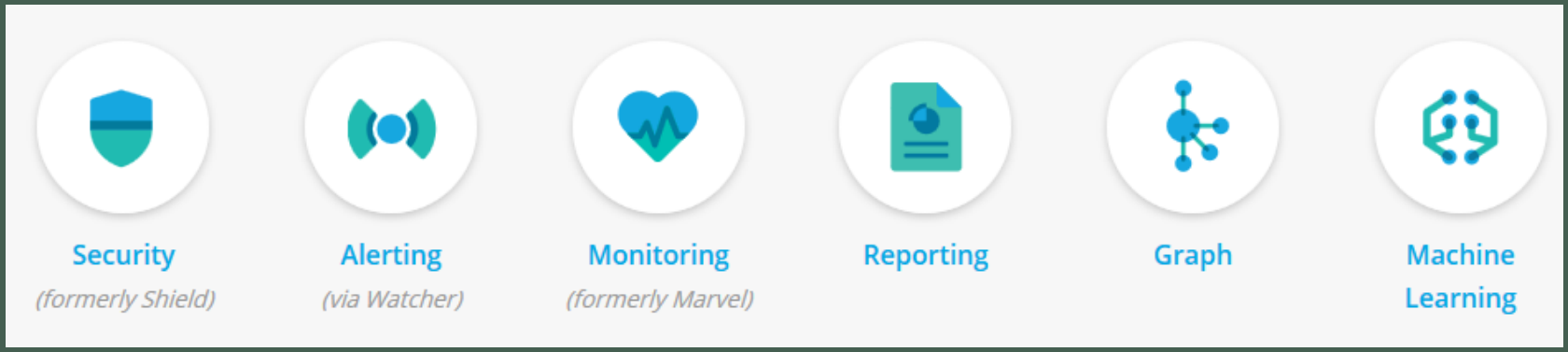
*.es(index=lasi2018-*lak17*,timefield=Timestamp,q='wiki',metric=count:Topic).label("Wiki").title('Activity related to different objects').color(#1E90FF).legend(columns=2,position=nw)*

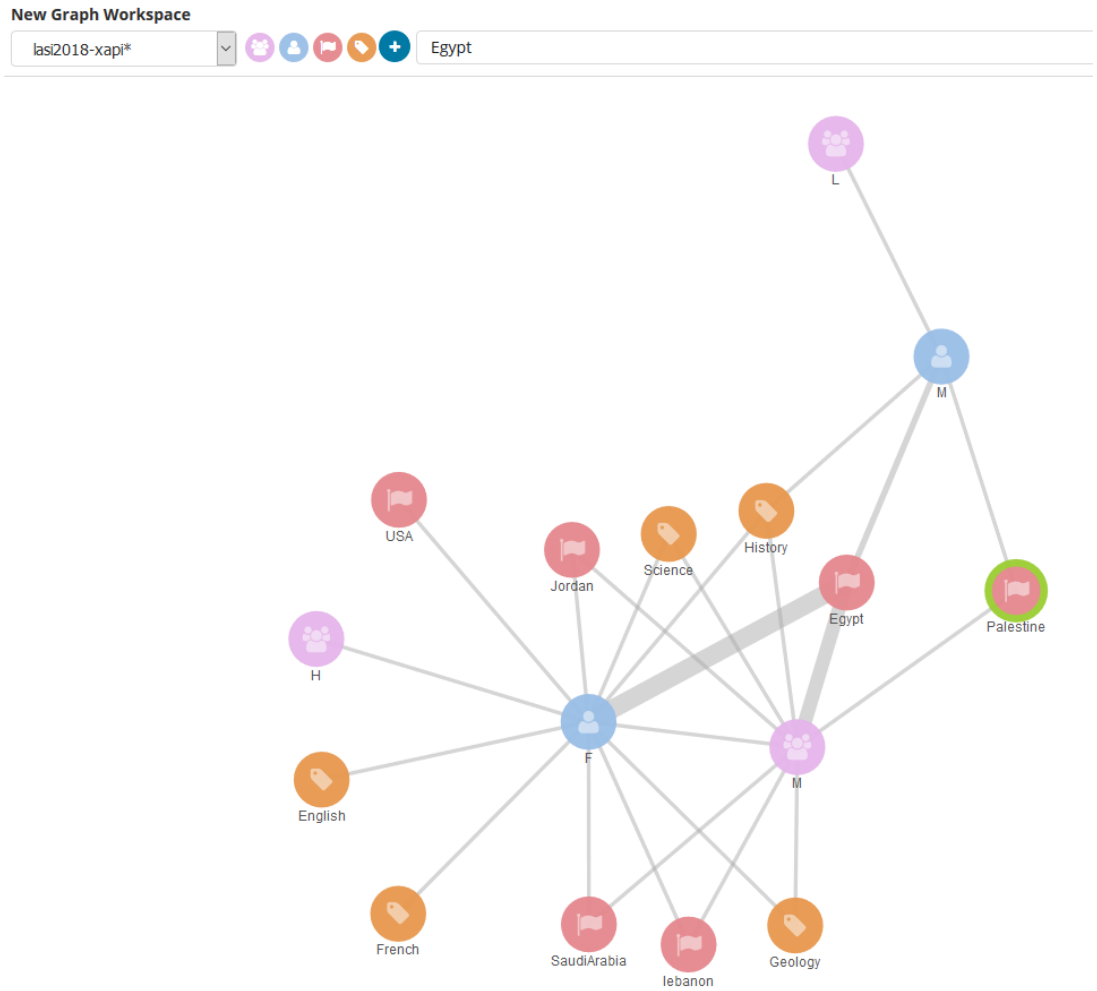
MACHINE LEARNING

Plugin X-Pack

X-PACK

- Non-free Elasticsearch plugin that increases the functionality with anomaly detecction, reports, monitoring, security and graph analysis





GRAPH

Graph

Related terms

Machine Learning

Machine Learning

Single metric
Detect anomalies in a single time series.

Multi metric
Detect anomalies in multiple metrics by splitting a time series by a categorical field.

Population
Detect activity that is unusual compared to the behavior of the population.

Advanced
Use the full range of options to create a job for more advanced use cases.

Further questions:

llanos@scc.uned.es
arobles@scc.uned.es
rpastor@scc.uned.es