



# Data Analysis with Elasticsearch and Kibana

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

SNOLA



LASI  
Spain 2018



Departamento de  
Sistemas de  
Comunicación  
y Control

# Outline

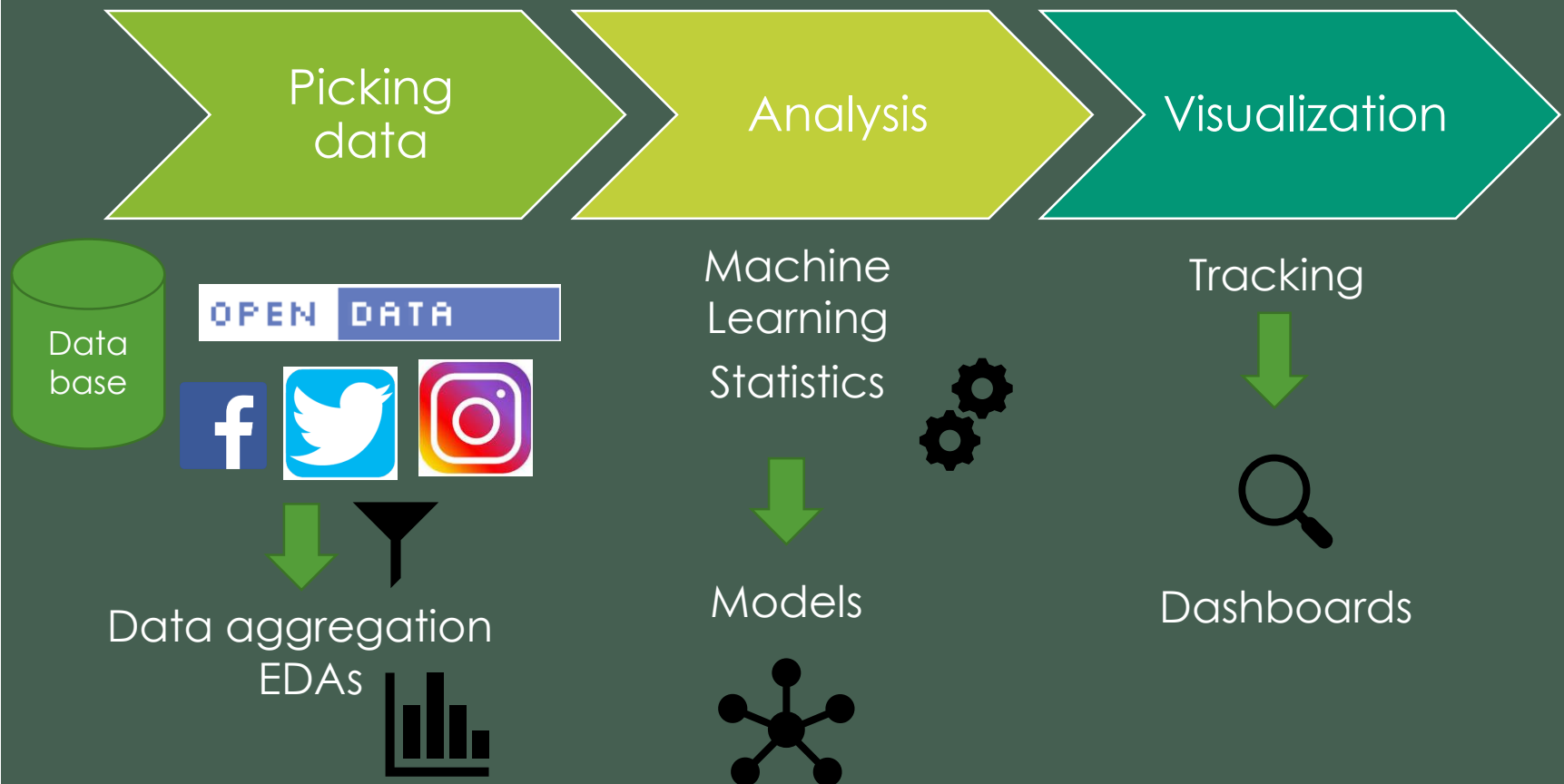
- Data visualization
- Installation and introduction
- Data handle: Elasticsearch
- Data visualization: Kibana
- Machine Learning: X-Pack



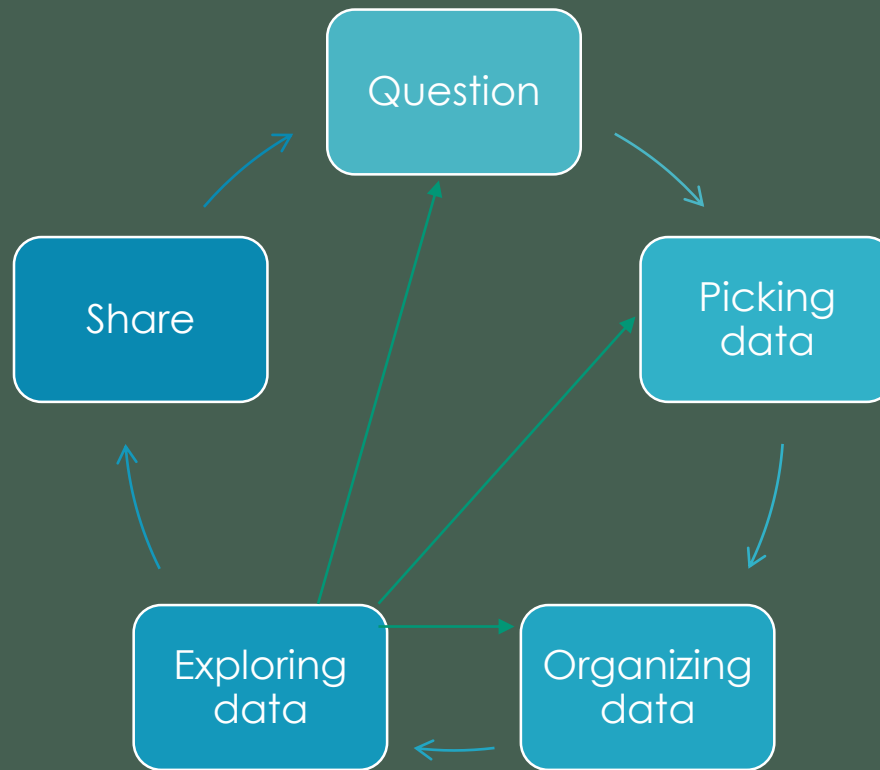
# DATA VISUALIZATION

More than graphs

# Data visualization



# 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

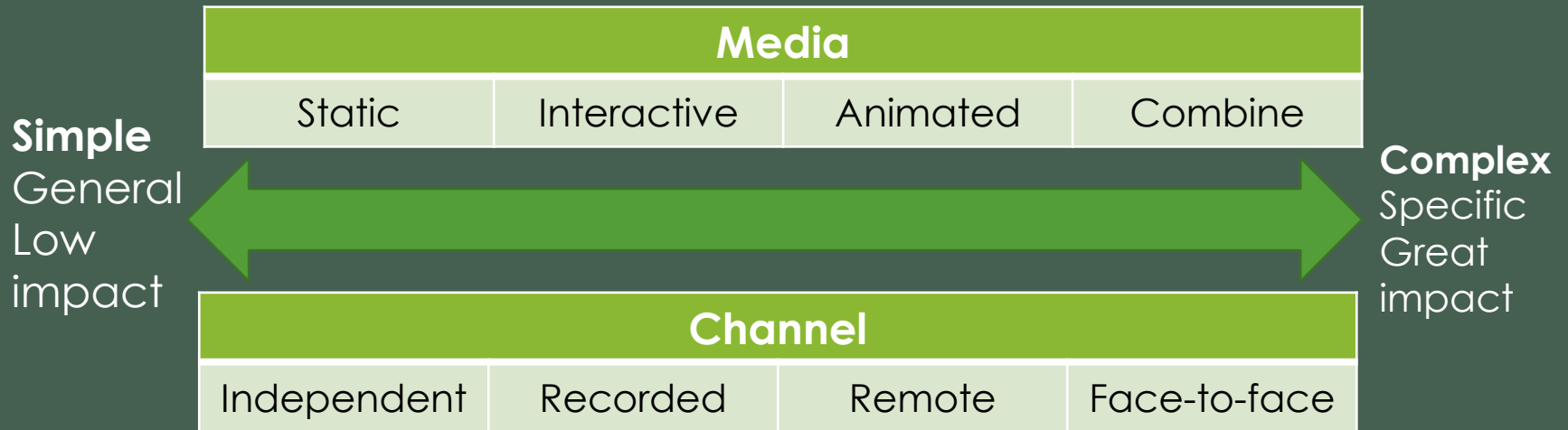


# Effective codification

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

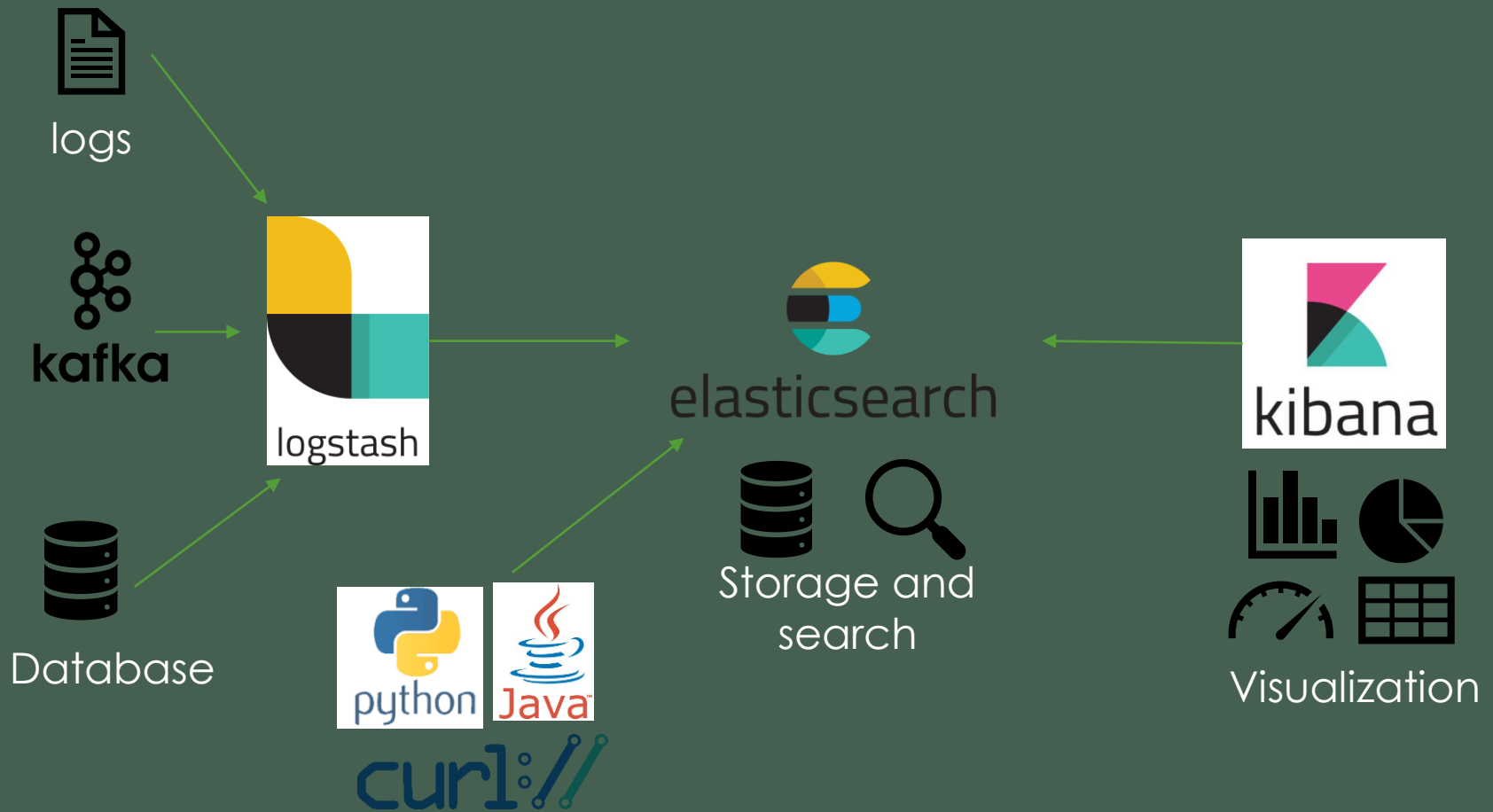
I LOVE LOGS

# LOGS as a source of information

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



# ELK Stack



# 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>

The screenshot shows the Oracle Java SE Development Kit 8 Downloads page. The page has a navigation menu on the left with links to Java SE, Java EE, Java ME, Java SE Advanced & Suite, Java Embedded, Java DB, Web Tier, Java Card, Java TV, New to Java, Community, and Java Magazine. The main content area is titled "Java SE Development Kit 8 Downloads" and includes a "Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language." It also mentions that the JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform. Below this, there is a section for "See also:" with links to the Java Developer Newsletter, Java Developer Day hands-on workshops, and Java Magazine. At the bottom, there is a table of download links for various operating systems and architectures, including Linux, Solaris, and Windows.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.92 MB	<a href="#">jdk-8u101-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.88 MB	<a href="#">jdk-8u101-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	168.96 MB	<a href="#">jdk-8u101-linux-i586.rpm</a>
Linux x86	183.76 MB	<a href="#">jdk-8u101-linux-i586.tar.gz</a>
Linux x64	166.09 MB	<a href="#">jdk-8u101-linux-x64.rpm</a>
Linux x64	180.97 MB	<a href="#">jdk-8u101-linux-x64.tar.gz</a>
macOS	247.12 MB	<a href="#">jdk-8u101-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	139.99 MB	<a href="#">jdk-8u101-solaris-sparcv9.tar.gz</a>
Solaris SPARC 64-bit	99.29 MB	<a href="#">jdk-8u101-solaris-sparcv9.tar.gz</a>
Solaris x64	140.57 MB	<a href="#">jdk-8u101-solaris-x64.tar.gz</a>
Solaris x64	97.02 MB	<a href="#">jdk-8u101-solaris-x64.tar.gz</a>
Windows x86	198.54 MB	<a href="#">jdk-8u101-windows-i586.exe</a>

The screenshot shows a Windows command prompt window titled "Símbolo del sistema". The window displays the output of the command `java -version`. The output is as follows:

```
Microsoft Windows [Versión 10.0.16299.309]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Llanos>java -version
java version "1.8.0_161"
Java(TM) SE Runtime Environment (build 1.8.0_161-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)

C:\Users\Llanos>
```

# 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 latest version of Java Software Developers Kit. You can obtain it 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.



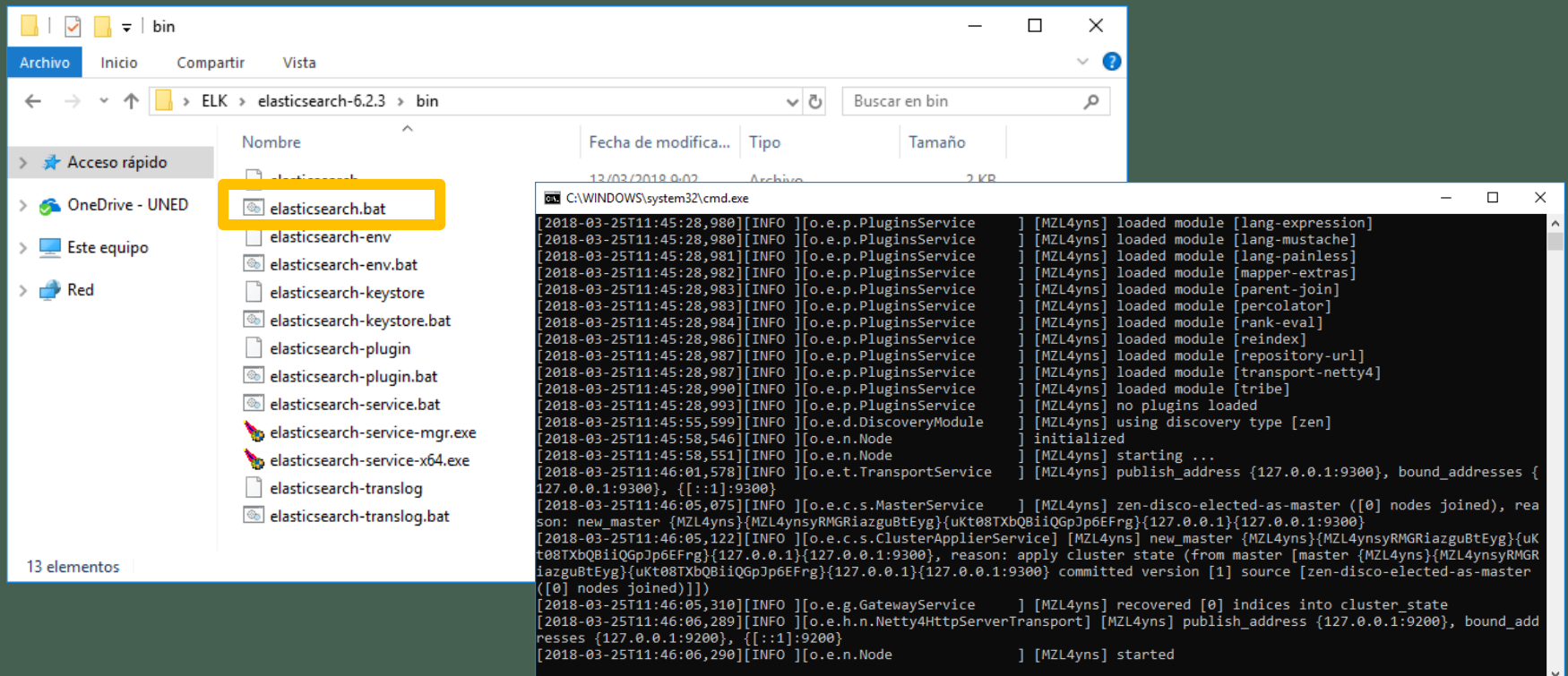
# ALTERNATIVE

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

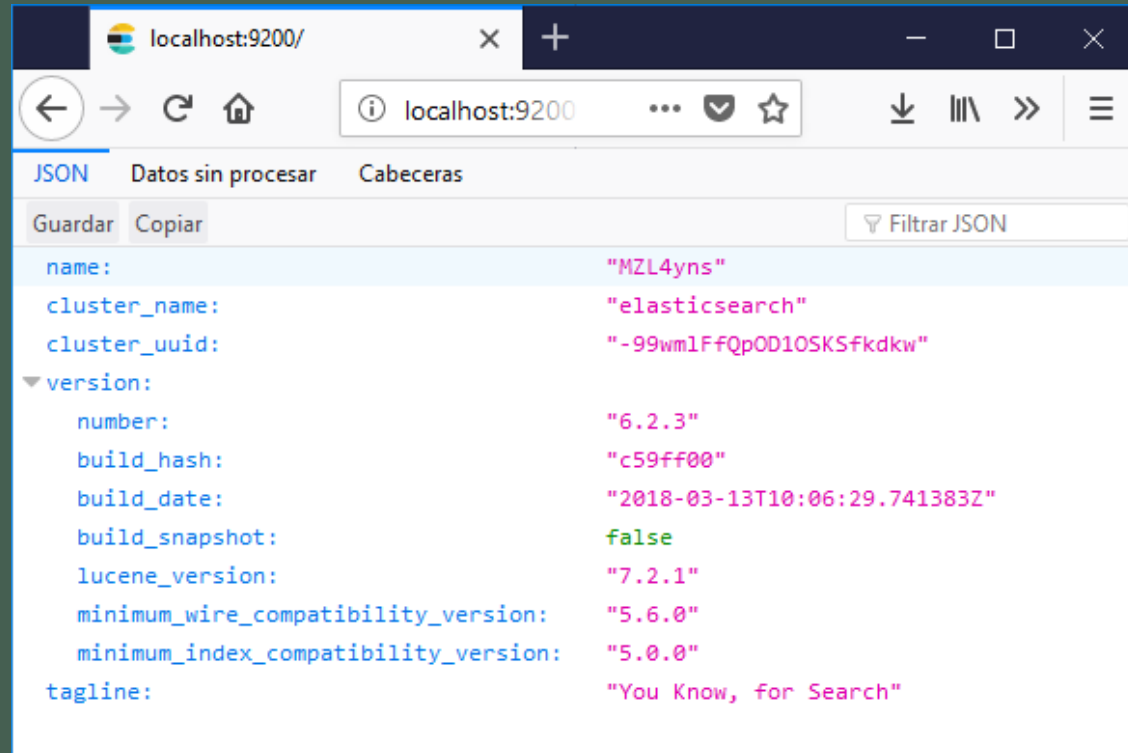


# First steps

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



# Checking

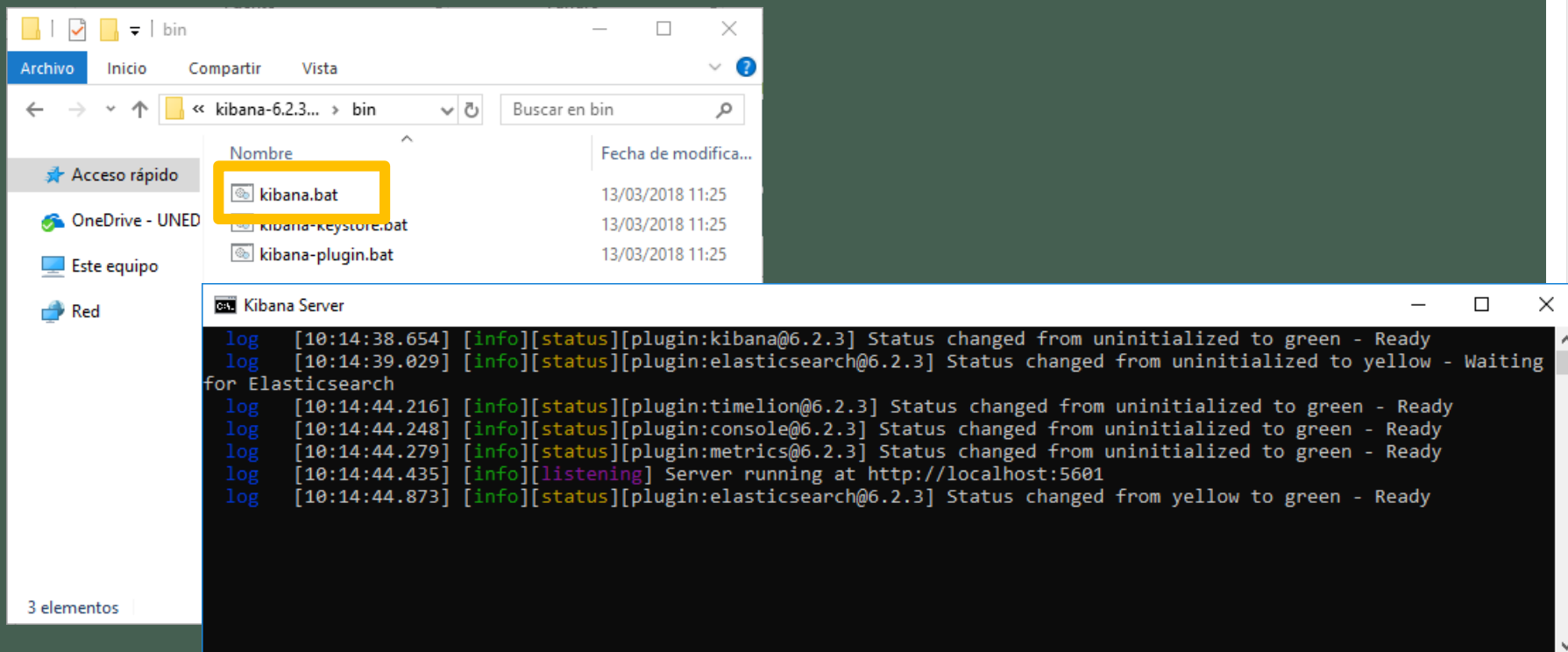


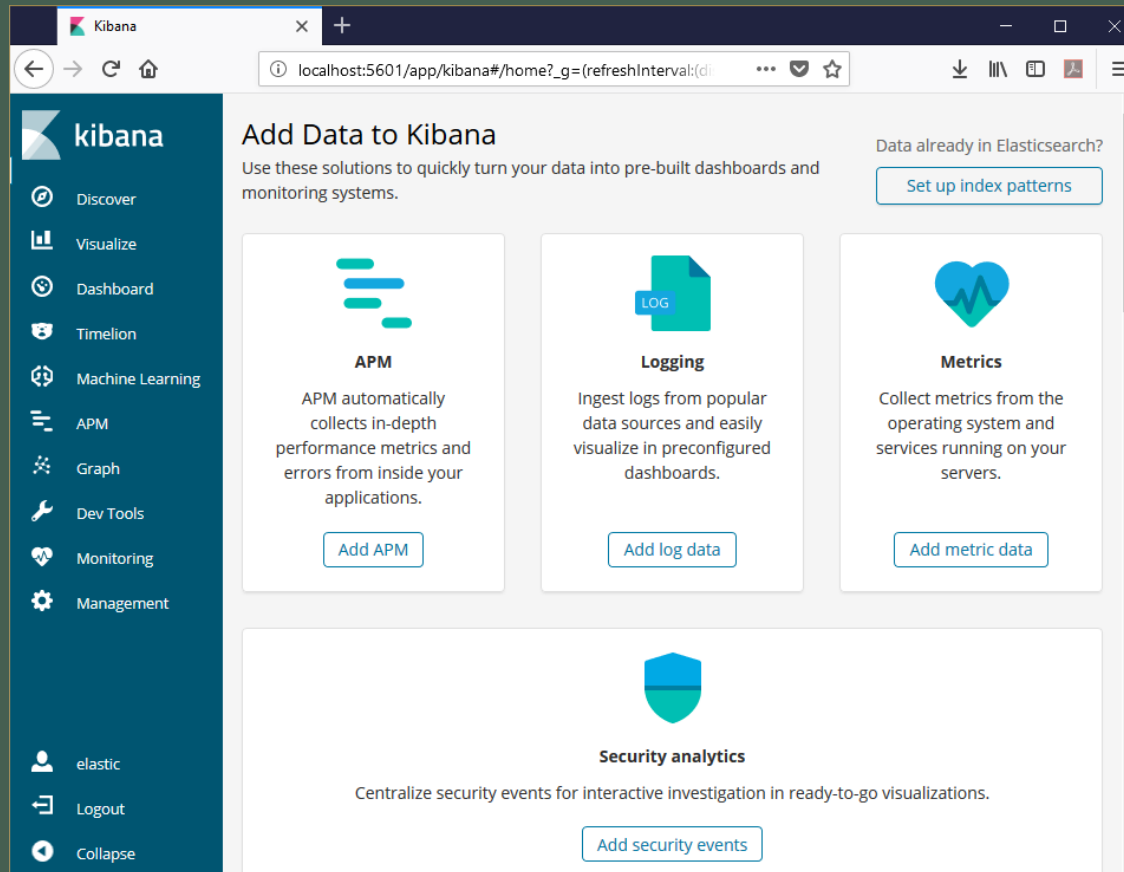
OPEN [HTTP://LOCALHOST:9200](http://localhost:9200)

CREDENTIALS ARE NEEDED

# First steps

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

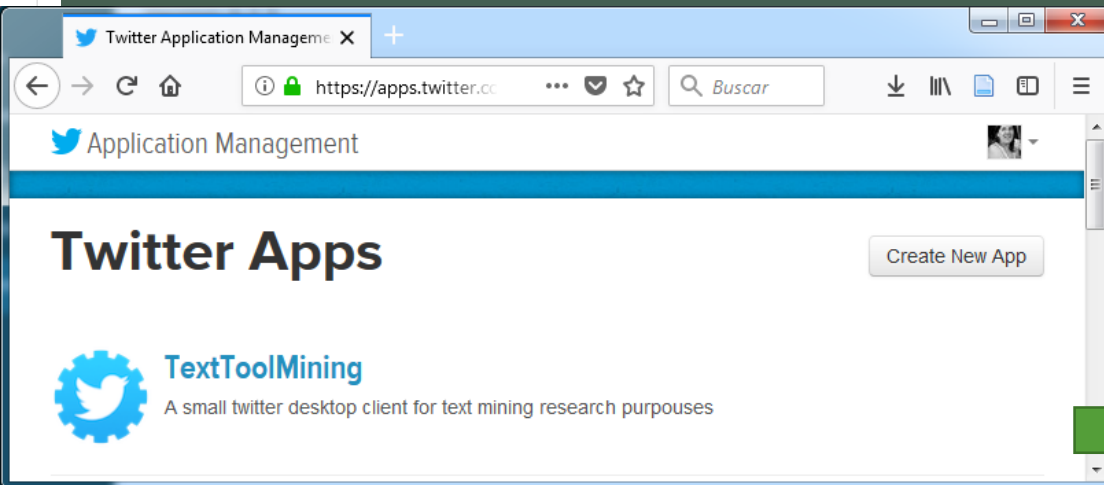




Accessing to Kibana: <http://localhost:5601>  
Use “elastic” user

# 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”



## Create an application

### Application Details

#### Name \*

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

#### Description \*

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

#### Website \*

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. If you don't have a URL yet, just put a placeholder here but remember to change it later.)

#### Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

### Developer Agreement

☒ Yes, I have read and agree to the [Twitter Developer Agreement](#).

Create your Twitter application

# Twitter: Security tokens

## Test in python

Test OAuth

[Details](#) [Settings](#) [Keys and Access Tokens](#) [Permissions](#)

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	CK
Consumer Secret (API Secret)	CS
Access Level	Read, write, and direct messages ( <a href="#">modify app permissions</a> )
Owner	LlanosTobarra
Owner ID	845136698



### Application Actions

[Regenerate Consumer Key and Secret](#) [Change App Permissions](#)

### Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	AT
Access Token Secret	ATC
Access Level	
Owner	LlanosTobarra

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 identifier
- 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



{RESTful API}

{JSON}

# 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



Scalability



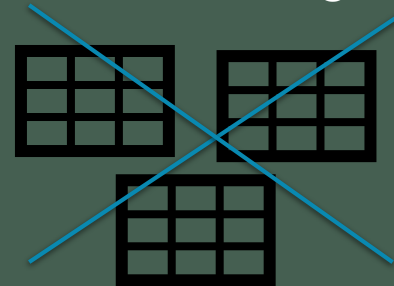
Very fast



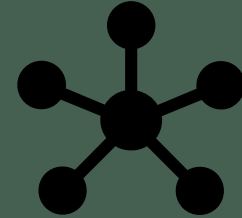
Multi-language



Free searches  
and search  
suggestions



Non schema

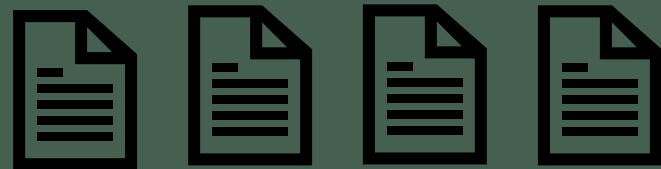


# Basic concepts

- 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

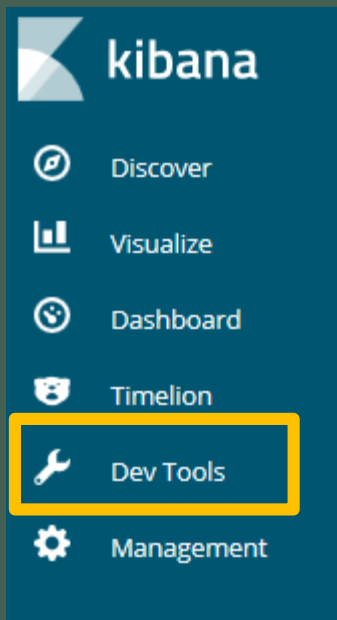
- Management of indexes

## Cluster

- Management of nodes
- Health of nodes

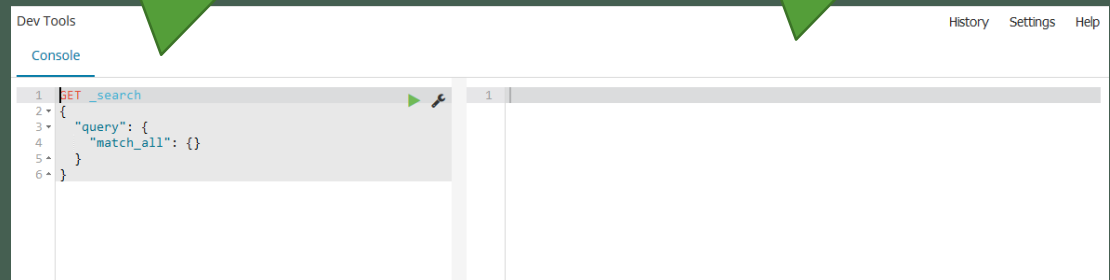
# CONSOLE UI

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



Call

Response



# 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
1 {
2   "cheyenne": {
3     "aliases": {},
4     "mappings": {
5       "registry": {
6         "properties": {
7           "Account-Name": {
8             "type": "text",
9             "fields": {
10              "keyword": {
11                "type": "keyword",
12                "ignore_above": 256
13              }
14            }
15          },
16          "Acct": {
17            "properties": {
18              "5": {
19                "type": "long"
20              },
21              "6": {
22                "type": "long"
23              }
24            }
25          },
26          "AcctGroup": {
27            "type": "text",
28            "fields": {
29              "keyword": {
30                "type": "keyword",
31                "ignore_above": 256
32              }
33            }
34          }
35        }
36      }
37    }
38  }
```

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

# MAPPING update

Index

Document  
type

Fields

```
PUT my_index ①
{
  "mappings": {
    "doc": { ②
      "properties": { ③
        "title": { "type": "text" }, ④
        "name": { "type": "text" }, ⑤
        "age": { "type": "integer" }, ⑥
        "created": {
          "type": "date", ⑦
          "format": "strict_date_optional_time|epoch_millis"
        }
      }
    }
  }
}
```

- 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

Index Patterns Saved Objects Advanced Settings

+ Create Index Pattern

★ chey\*

★ chey\*



This page lists every field in the **chey\*** index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's [Mapping API](#)

fields (23)

scripted fields (0)

source filters (0)

Filter

All field types

name	type	format	searchable	aggregatable	excluded	controls
Account-Name	string		✓			
Account-Name.keyword	string		✓	✓		
Acct.5	number		✓	✓		
Acct.6	number		✓	✓		

# Text queries

The screenshot shows the Kibana Discover interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timelion, Dev Tools, and Management. The main area displays a search query: `Month:NOVEMBER AND Nature:ASSETS`. Below the query bar, there are two search results. Each result is a JSON object representing a document from the 'chey\*' index. The first result has a score of 4.453 and the second has a score of 4.453. Both results are for documents of type 'registry' and index 'cheyenne'.

Discover - Kibana

localhost:5601/app/kibana#/discover?\_g=()&\_a=(columns:!(source),ir

316 hits

New Save Open Share

Month:NOVEMBER AND Nature:ASSETS

Uses lucene query syntax

Discover

Add a filter +

Visualize

chey\*

Selected Fields

? \_source

Available Fields

t Account...

# Acct.5

# Acct.6

t AcctGroup

\_source

Nature: ASSETS Month: NOVEMBER Company: Cheyenne Manufacturing AcctGroup: BALAN  
CE SHEET Account-Name: ACCUMULATORS Count: 1 Unit.: 1904000 Acct.5: 12,130 Acc  
t.6: 121,000 CostCenter: CORPORATE Year: 2,013 Amount: 20,818.85 \_id: xUBVY2IB  
ZrGhb1nZXf4t \_type: registry \_index: cheyenne \_score: 4.453

Nature: ASSETS Month: NOVEMBER Company: Cheyenne Manufacturing AcctGroup: BALAN  
CE SHEET Account-Name: CIRCUIT BREAKERS Count: 1 Unit.: 1904000 Acct.5: 12,180  
Acct.6: 121,000 CostCenter: CORPORATE Year: 2,013 Amount: 2,721.67 \_id: yEBVY2  
IBZrGhb1nZXf58 \_type: registry \_index: cheyenne \_score: 4.453



# 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. $\rightarrow$ Abcd, Abce,...
+	One or more occurrences of the regular expression	A+ $\rightarrow$ A, AA, AAA,AAAA,...
*	None or more occurrences of the regular expression	A* $\rightarrow$ '', A, AA, AAA,AAAA,...
?	None or one occurrences of the regular expression	A? $\rightarrow$ '', A
{min,max}	Minimum and maximum number of occurrences	A{2,5} $\rightarrow$ AA,AAA, AAAA,AAAAA
()	Expression aggrupation	(AB)+ $\rightarrow$ AB, ABAB,ABABAB,...
	One element or another	A   B $\rightarrow$ A , B
[a-b]	Range of elements	[A-Z] $\rightarrow$ A, B, C, ..., Z
~	Denial, which does not contain the regular expression	~A $\rightarrow$ B, C, ...,Z
<i-z>	Interval	A<1-5> $\rightarrow$ 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



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

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

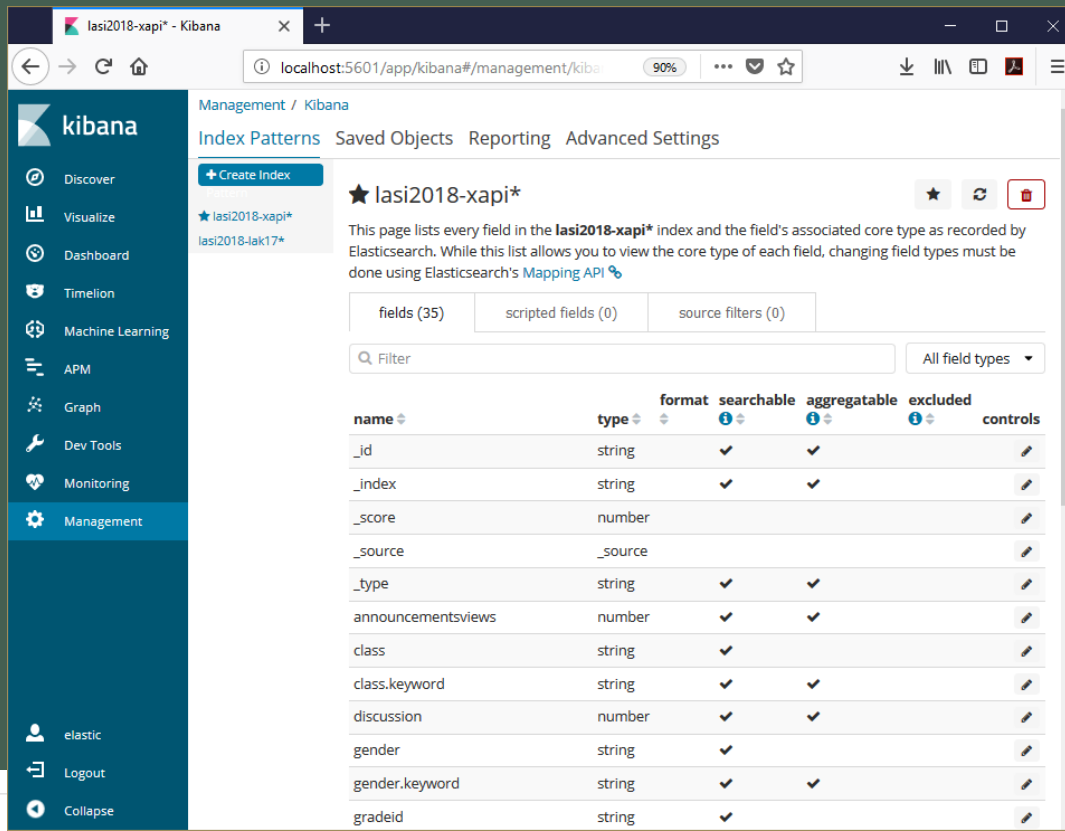


# KIBANA

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



The screenshot shows the Kibana Management interface. The left sidebar contains navigation links: Discover, Visualize, Dashboard, Timelion, Machine Learning, APM, Graph, Dev Tools, Monitoring, and Management (highlighted). The main content area is titled 'Management / Kibana' and shows the 'Index Patterns' section. A 'Create Index' button is visible. Below it, the 'lasi2018-xapi\*' index is selected. A description states: 'This page lists every field in the lasi2018-xapi\* index and the field's associated core type as recorded by Elasticsearch. While this list allows you to view the core type of each field, changing field types must be done using Elasticsearch's Mapping API'. Below the description are filters for 'fields (35)', 'scripted fields (0)', and 'source filters (0)'. A search bar and a dropdown for 'All field types' are also present. The main table lists fields with columns: name, type, format, searchable, aggregatable, excluded, and controls. The table contains 15 rows of field data.

name	type	format	searchable	aggregatable	excluded	controls
_id	string		✓	✓		
_index	string		✓	✓		
_score	number					
_source	_source					
_type	string		✓	✓		
announcementsviews	number		✓	✓		
class	string		✓			
class.keyword	string		✓	✓		
discussion	number		✓	✓		
gender	string		✓			
gender.keyword	string		✓	✓		
gradeid	string		✓			

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

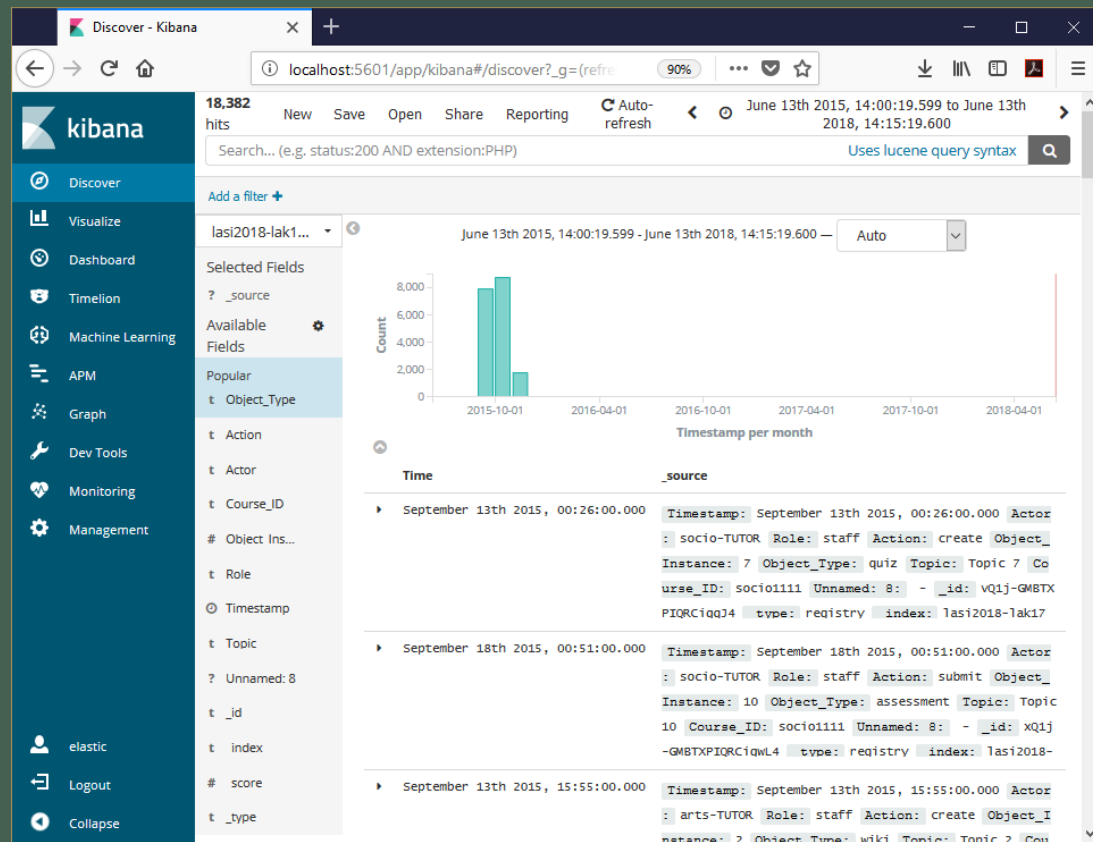
And review the plugin configuration

# Discover

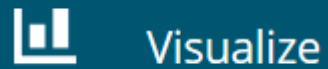


Discover

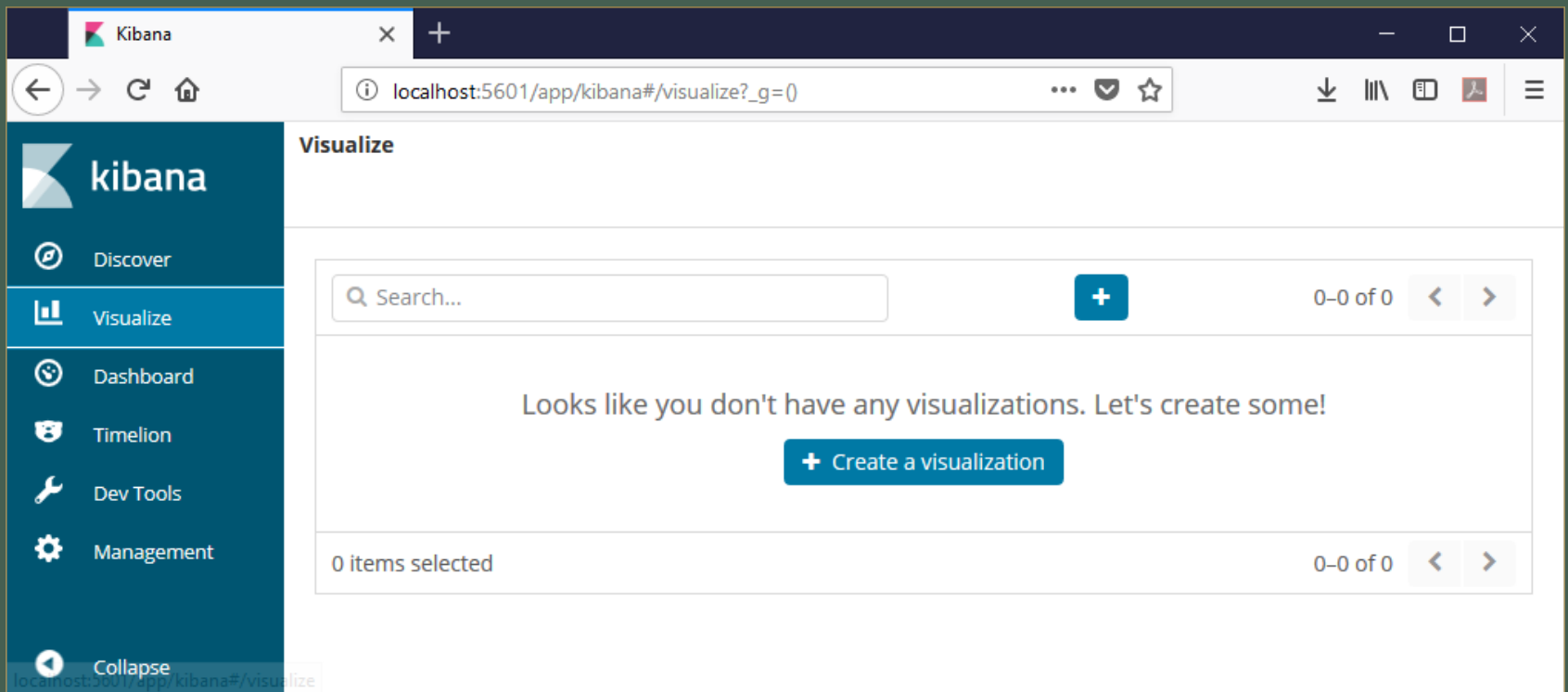
- Once we have created at least one index pattern we can use Discover tool in order to know more about the data
- Filters and queries



# Visualize

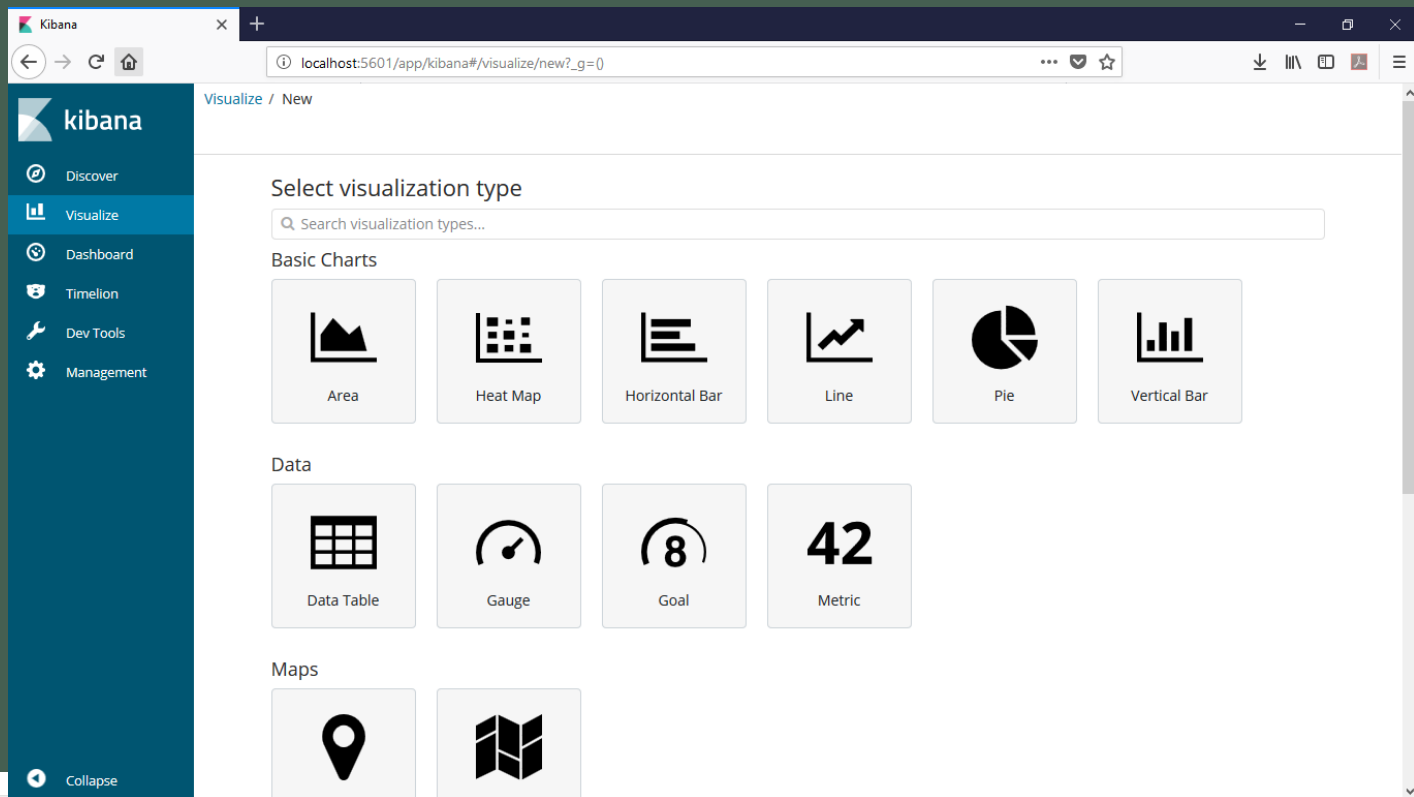


- Visualize option allows us to create basic graphs



# Types of graphs

- Once we have chosen “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

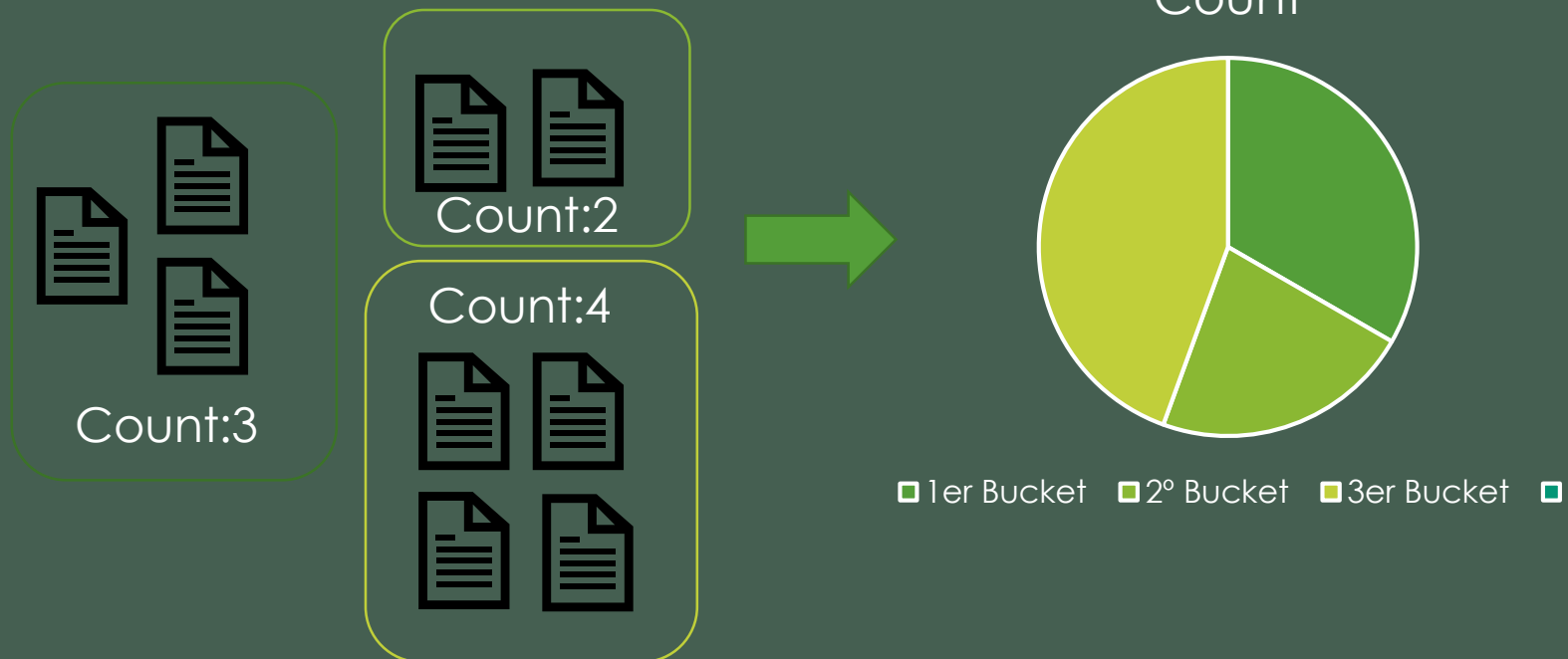
### Relation among data groups

- Dispersion
- Bubbles
- Lines

A type of graph for each use case

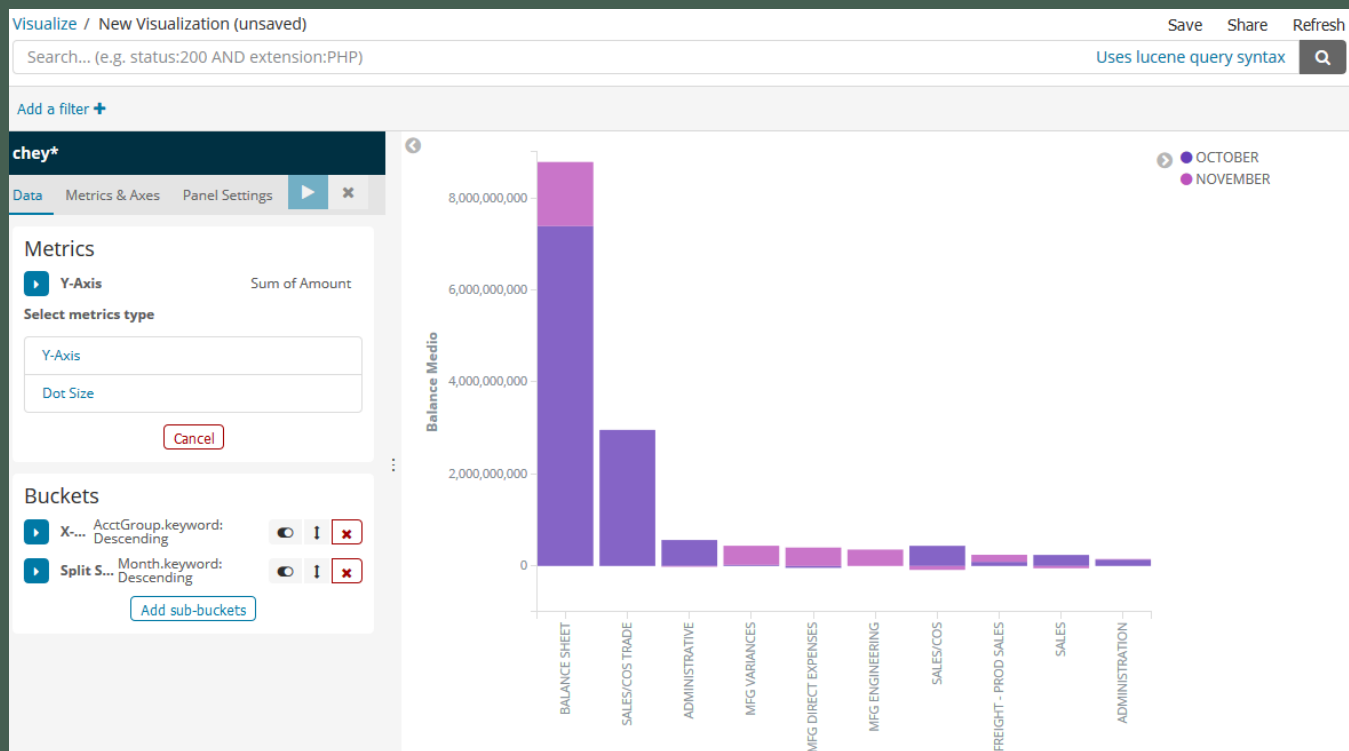
# Aggregations

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



# Visualize: graph definition

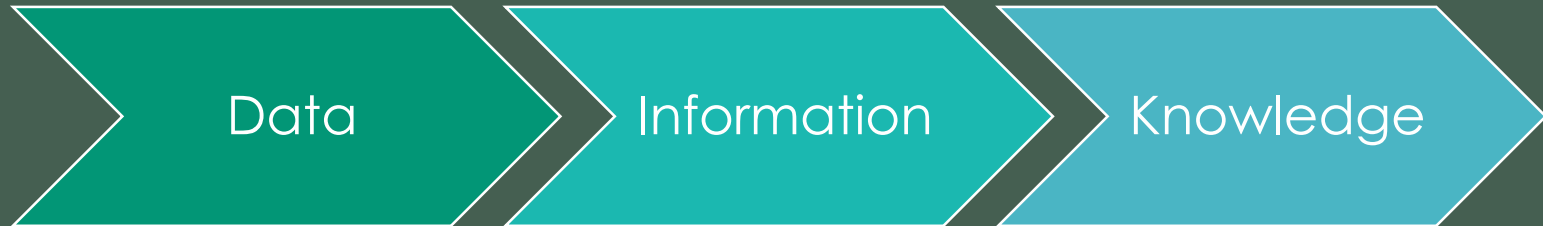
- Save: store graph
- Share: HTML is provided so we can integrate the graph into a webpage



# 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

TITLE		LOGO	
Introduction _____ _____			
Filter 1 ▼	KPI #1	KPI #2	KPI #3
Filter 2 ▼			
GRAPHIC #1		GRAPHIC #3 / VIDEO / WEB SITE	
GRAPHIC #2			
Source   <a href="http://www...">http://www...</a>		Updated   MM/DD/YYYY	

- 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

- Achieve objectives
- Business Intelligence
- Ej: Web traffic to a blog

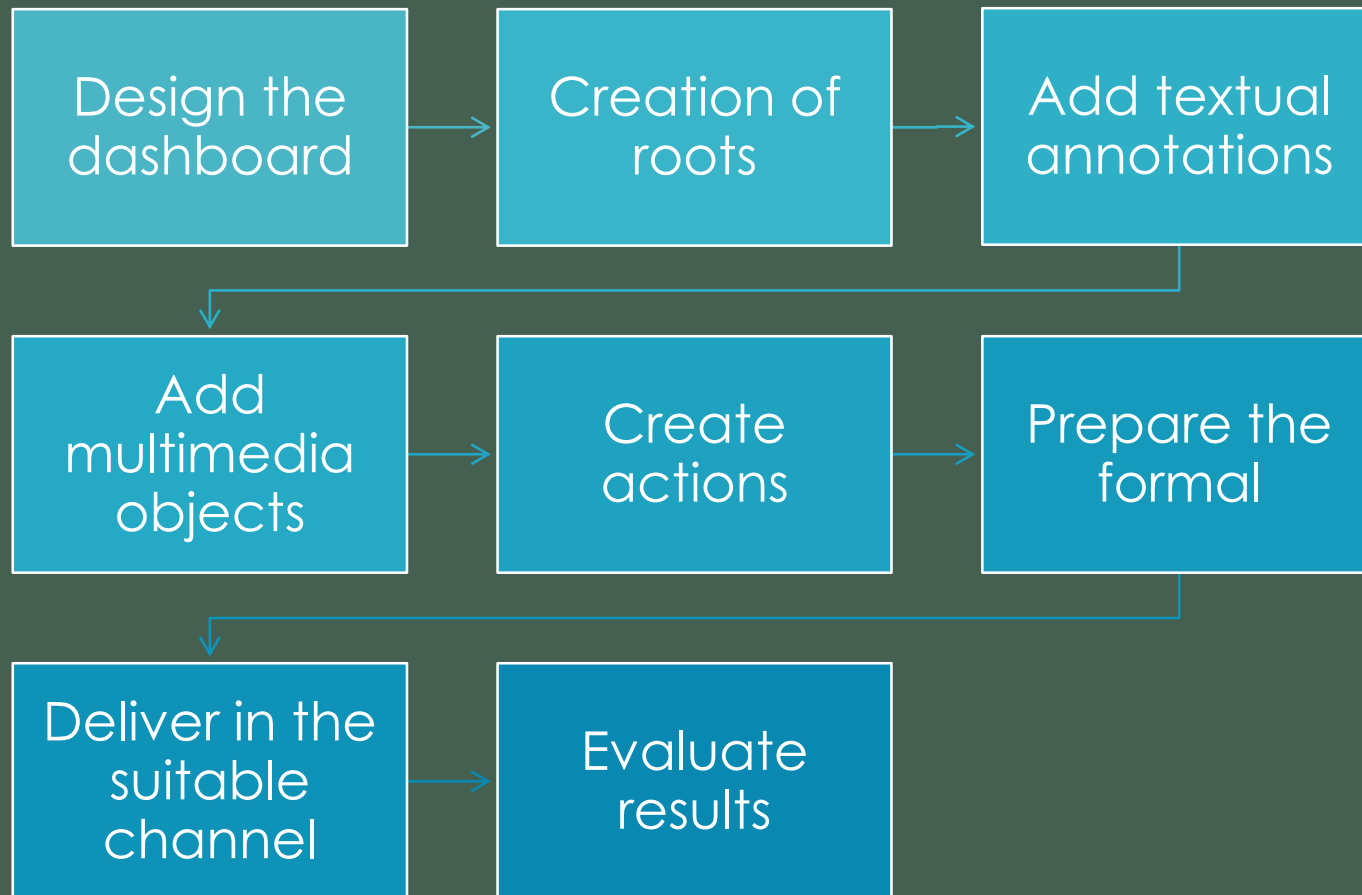
## Data journalism

- Report about news
- Delve into a hot topic
- Attractive

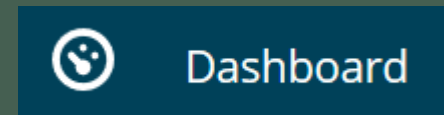
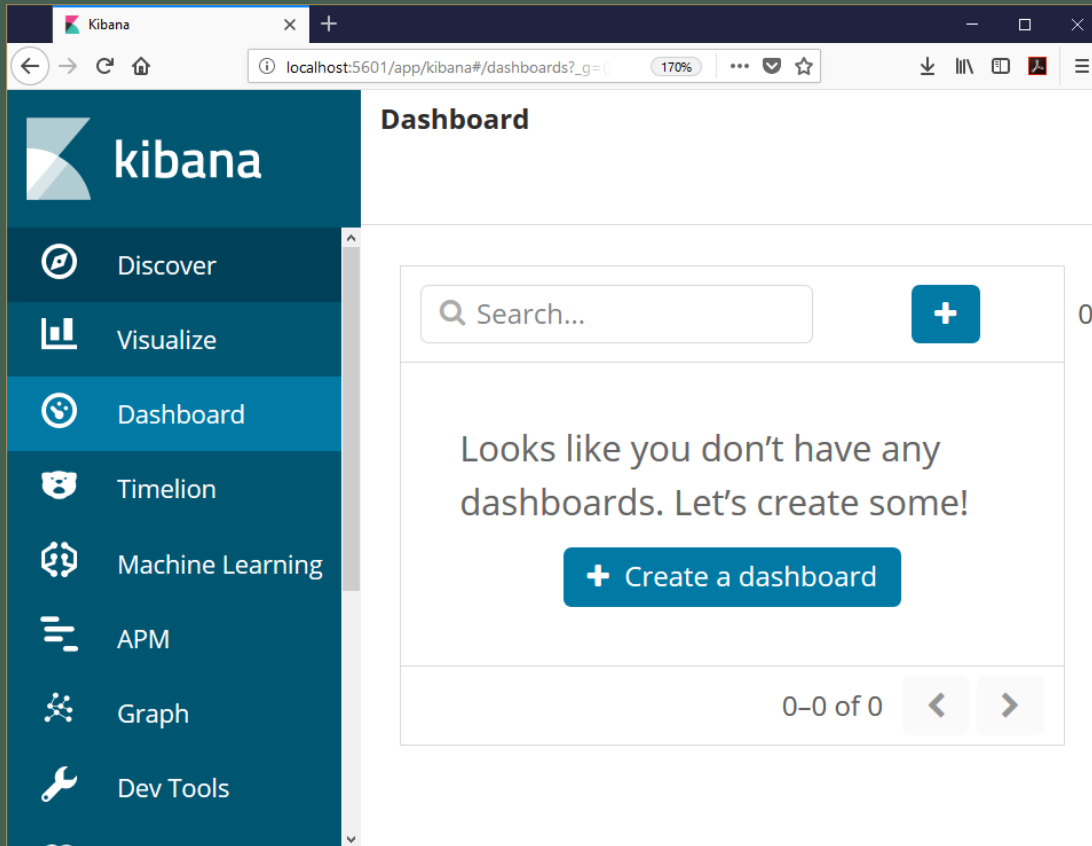
## Open data

- Repository converted in visualization
- Educate readers

# Workflow







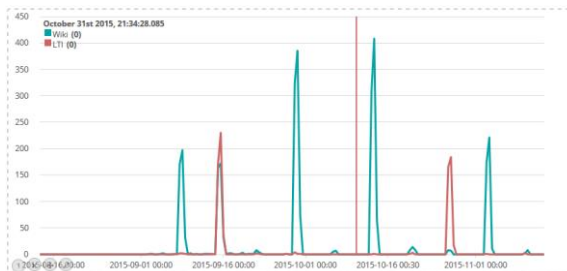
# Dashboards



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  
i").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 detection, reports, monitoring, security and graph analysis



Security

*(formerly Shield)*



Alerting

*(via Watcher)*



Monitoring

*(formerly Marvel)*



Reporting



Graph



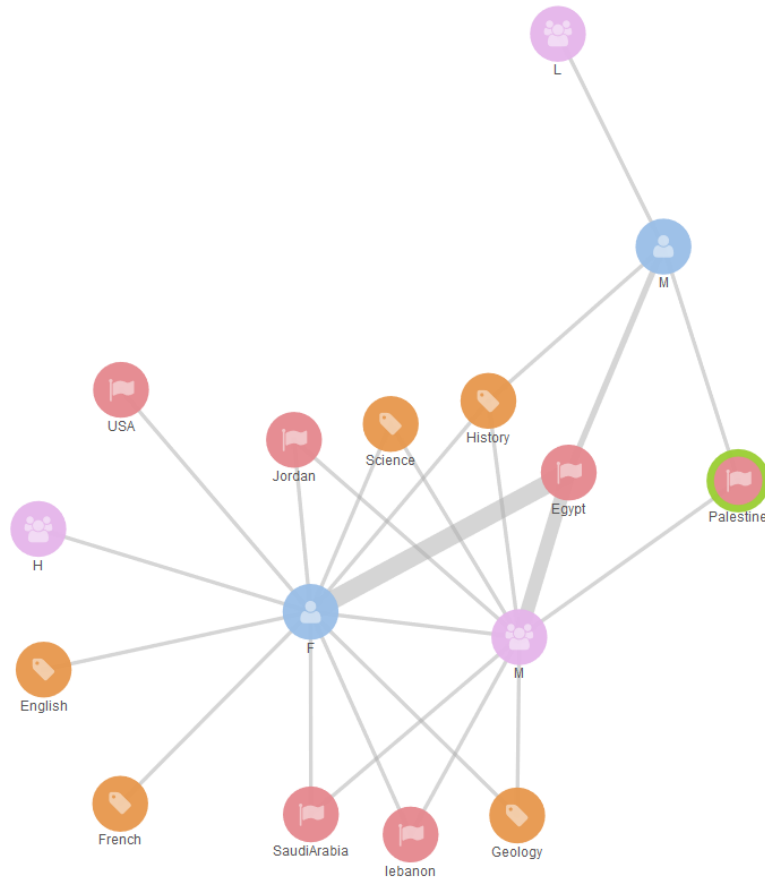
Machine  
Learning

## New Graph Workspace

lasl2018-xapi\*



Egypt



# 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](mailto:llanos@scc.uned.es)  
[arobles@scc.uned.es](mailto:arobles@scc.uned.es)  
[rpastor@scc.uned.es](mailto:rpastor@scc.uned.es)

