

Introducción al stack de Elastic y sus capacidades geoespaciales

Jorge Sanz

2022-06-09, Kibana, Elastic,

 xurxosanz  jsanz  jorge.sanz@elastic.co

<https://ela.st/siglibre2022-taller>

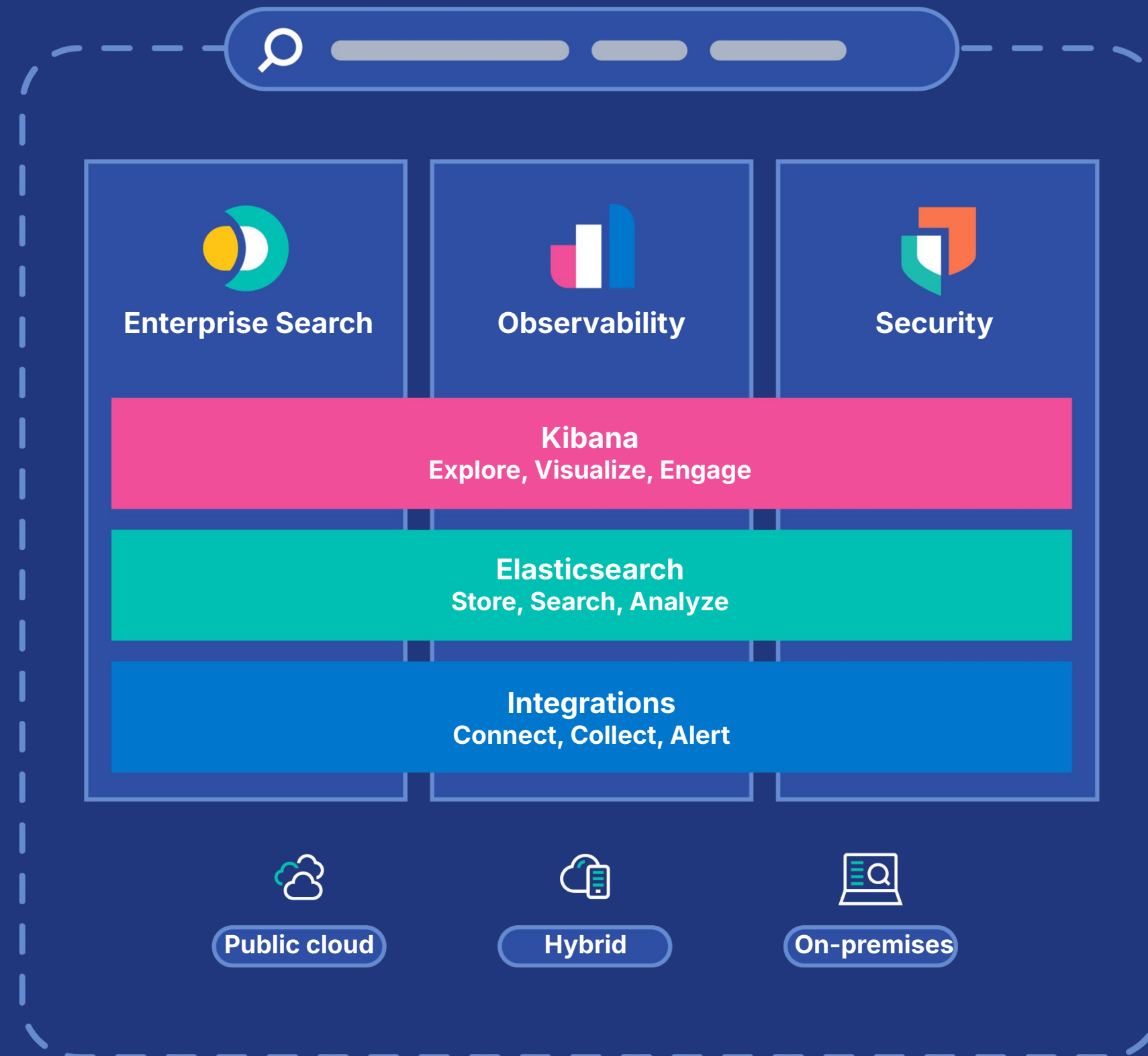
Objetivos de la sesión

- ✓ **Ejecutar** el stack de Elastic de manera sencilla
- ✓ Qué son **Kibana** y sus diferentes componentes
- ✓ Cómo usar **Elastic Maps** para visualizar datos geoespaciales
- ✓ Cuáles son las características **geoespaciales** básicas de Elasticsearch
- ✓ Cómo encaja Elastic en la **arquitectura** de una aplicación geoespacial

Agenda

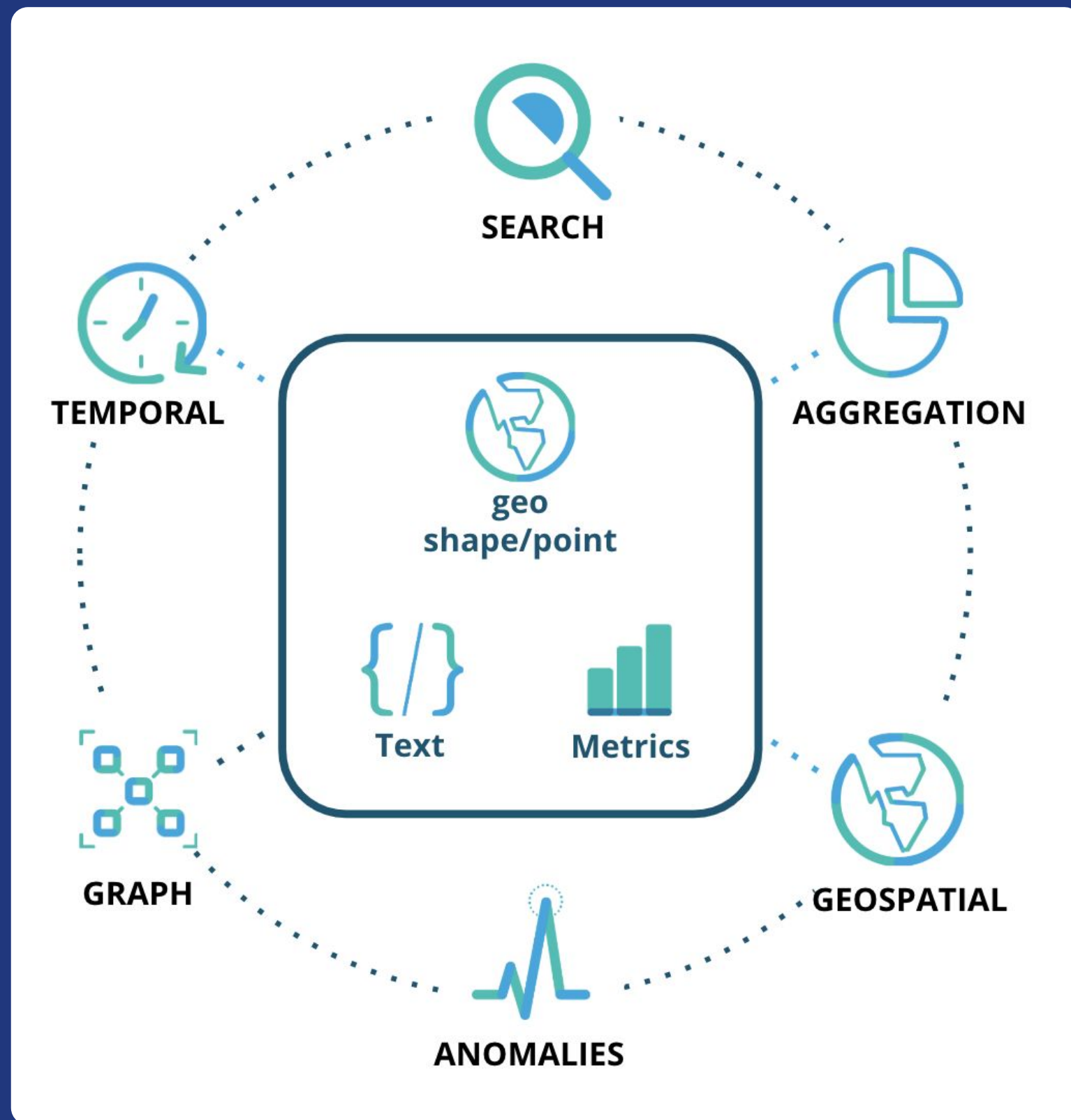
- 1 Introducción al stack de Elastic
- 2 Preparación del laboratorio
- 3 Introducción a Kibana
- 4 Elastics Maps
- 5 Elasticsearch Geo
- 6 Web mapping con Elasticsearch

La plataforma Elastic

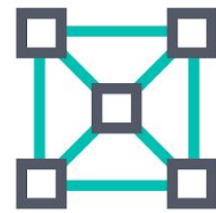


Elasticsearch

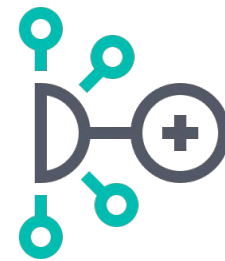
Todos los datos son bienvenidos



Componentes de Elasticsearch



Cluster



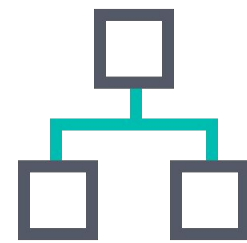
Node



Shard



Index



Mapping



Document



Field

Communicating with Elasticsearch

- All communication through HTTP endpoints 

```
→ http --verify=/tmp/ca.crt --auth elastic:changeme https://localhost:9203/
HTTP/1.1 200 OK
X-elastic-product: Elasticsearch
content-length: 531
content-type: application/json

{
  "cluster_name": "docker-cluster",
  "cluster_uuid": "kIQy28mqRlyOvBis_FELXg",
  "name": "es01",
  "tagline": "You Know, for Search",
  "version": {
    "build_date": "2022-04-20T10:35:10.180408517Z",
    "build_flavor": "default",
    "build_hash": "b174af62e8dd9f4ac4d25875e9381ffe2b9282c5",
    "build_snapshot": false,
    "build_type": "docker",
    "lucene_version": "9.1.0",
    "minimum_index_compatibility_version": "7.0.0",
    "minimum_wire_compatibility_version": "7.17.0",
    "number": "8.2.0"
  }
}
```

Communicating with Elasticsearch

- All communication through HTTP endpoints 
- JSON

```
→ http --verify=/tmp/ca.crt --auth elastic:changeme\  
  "https://localhost:9203/flight_tracking*/_search?size=1"  
HTTP/1.1 200 OK  
X-elastic-product: Elasticsearch  
content-length: 582  
content-type: application/json  
  
{  
  "_shards": {  
    "failed": 0,  
    "skipped": 0,  
    "successful": 2,  
    "total": 2  
  },  
  "hits": {  
    "hits": [  
      {  
        "_id": "_VcQtIAB4XM30LHfsxTV",  
        "_index": "flight_tracking_2022-05-11",  
        "_score": 1.0,  
        "_source": {  
          "@timestamp": 1652288434988,  
          "baroAltitude": 8206.74,  
          "callsign": "PDT6046",  
          "geoAltitude": 8564.88,  
          "heading": 56.6,  
          "icao24": "a808c4",  
          "lastContact": 1652288393000,  
          "location": {  
            "lat": 34.129,  
            "lon": -82.6954  
          },  
          "onGround": false,  
          "originCountry": "United States",  
          "spi": false,  
        }  
      }  
    ]  
  }  
}
```

Communicating with Elasticsearch

- All communication through HTTP endpoints 
- JSON
- REST methods: GET, PUT, POST, DELETE

```
○ echo -n '{"hello": "world"}' | http --verify=/tmp/ca.crt --auth elastic:changeme \
  POST "https://localhost:9203/my_test/_doc"
HTTP/1.1 201 Created
Location: /my_test/_doc/KLCJxoAB0CoZ6d_Z8ZDc
X-elastic-product: Elasticsearch
content-length: 159
content-type: application/json

{
  "_id": "KLCJxoAB0CoZ6d_Z8ZDc",
  "_index": "my_test",
  "_primary_term": 1,
  "_seq_no": 0,
  "_shards": {
    "failed": 0,
    "successful": 1,
    "total": 2
  },
  "_version": 1,
  "result": "created"
}

→ http --verify=/tmp/ca.crt --auth elastic:changeme DELETE \
  "https://localhost:9203/my_test"
HTTP/1.1 200 OK
X-elastic-product: Elasticsearch
content-length: 21
content-type: application/json

{
  "acknowledged": true
}
```

Communicating with Elasticsearch

- All communication through HTTP endpoints 
- JSON
- REST methods: GET, PUT, POST, DELETE
- _cat API for human readable display 

```
~  
→ http --verify=/tmp/ca.crt --auth elastic:changeme\  
  "https://localhost:9203/_cat/health?v&h=cluster,status,node.total,pri,shards"  
HTTP/1.1 200 OK  
X-elastic-product: Elasticsearch  
content-length: 88  
content-type: text/plain; charset=UTF-8
```

cluster	status	node.total	pri	shards
docker-cluster	green	2	15	30

```
~  
→ http --verify=/tmp/ca.crt --auth elastic:changeme\  
  "https://localhost:9203/_cat/nodes?v&h=name,ip,ram.percent,cpu,node.role"  
HTTP/1.1 200 OK  
X-elastic-product: Elasticsearch  
content-length: 130  
content-type: text/plain; charset=UTF-8
```

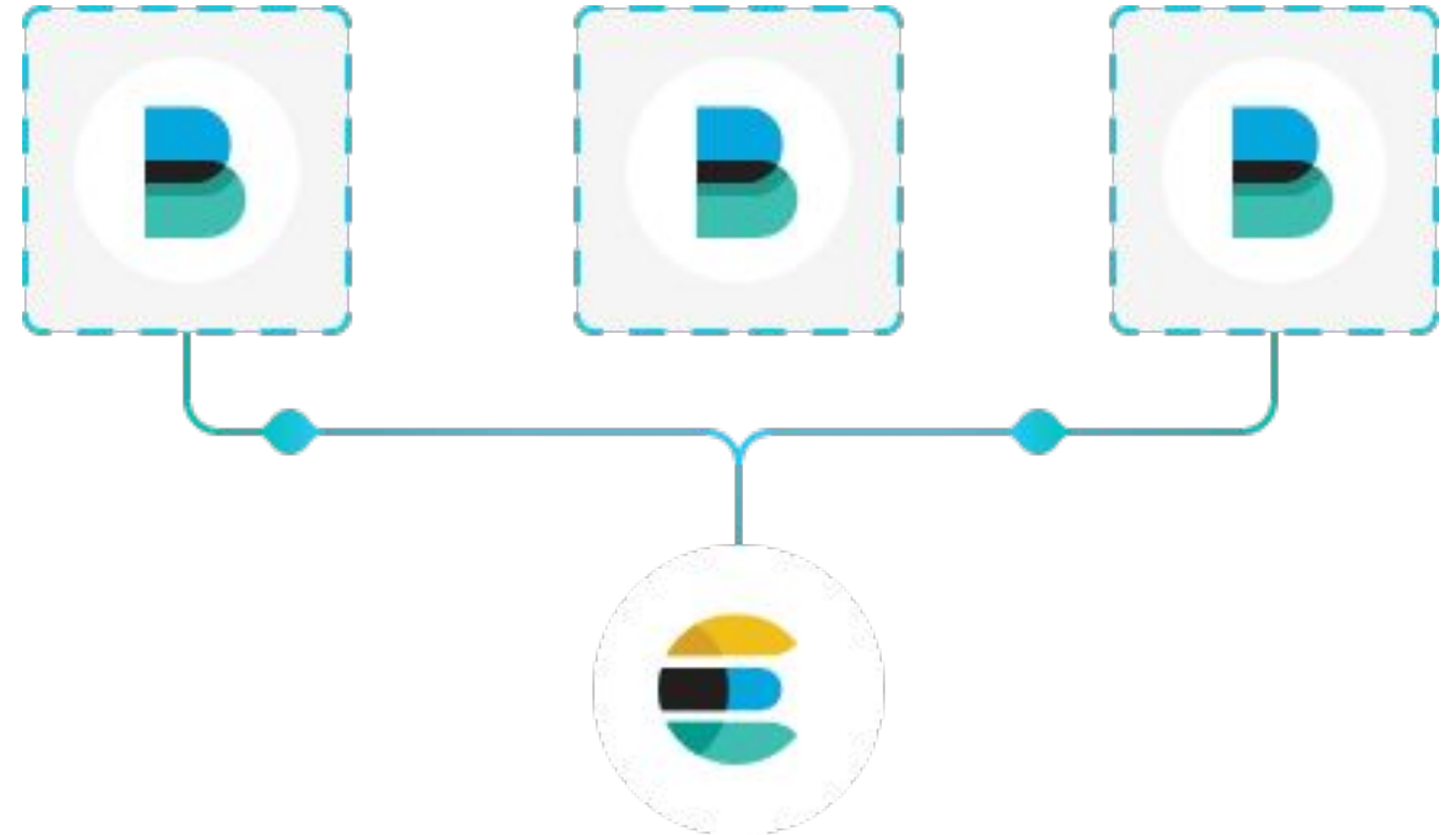
name	ip	ram.percent	cpu	node.role
es01	172.27.0.3	100	7	cdfhilmrstw
es02	172.27.0.4	96	7	cdfhilmrstw

```
~  
→ http --verify=/tmp/ca.crt --auth elastic:changeme\  
  "https://localhost:9203/_cat/indices?v&h=index,health,docs.count,store.size"  
HTTP/1.1 200 OK  
X-elastic-product: Elasticsearch  
content-length: 224  
content-type: text/plain; charset=UTF-8
```

index	health	docs.count	store.size
flight_tracking_2022-05-12	green	299029	108.6mb
flight_tracking_2022-05-11	green	438453	155.2mb
flight_tracking_2022-05-15	green	4070	3.4mb



Lightweight data shippers



Ship data from the source	Ship and centralize in Elasticsearch	Ship to Logstash for transformation and parsing
Ship to Elastic Cloud	Libbeat: API framework to build custom beats	70+ community Beats



Beats

All the modules



FileBeat
Log Files



MetricBeat
Metrics



PacketBeat
Network Data



WinLogBeat
Window Events



HeartBeat
Uptime Monitoring



AuditBeat
Audit Data



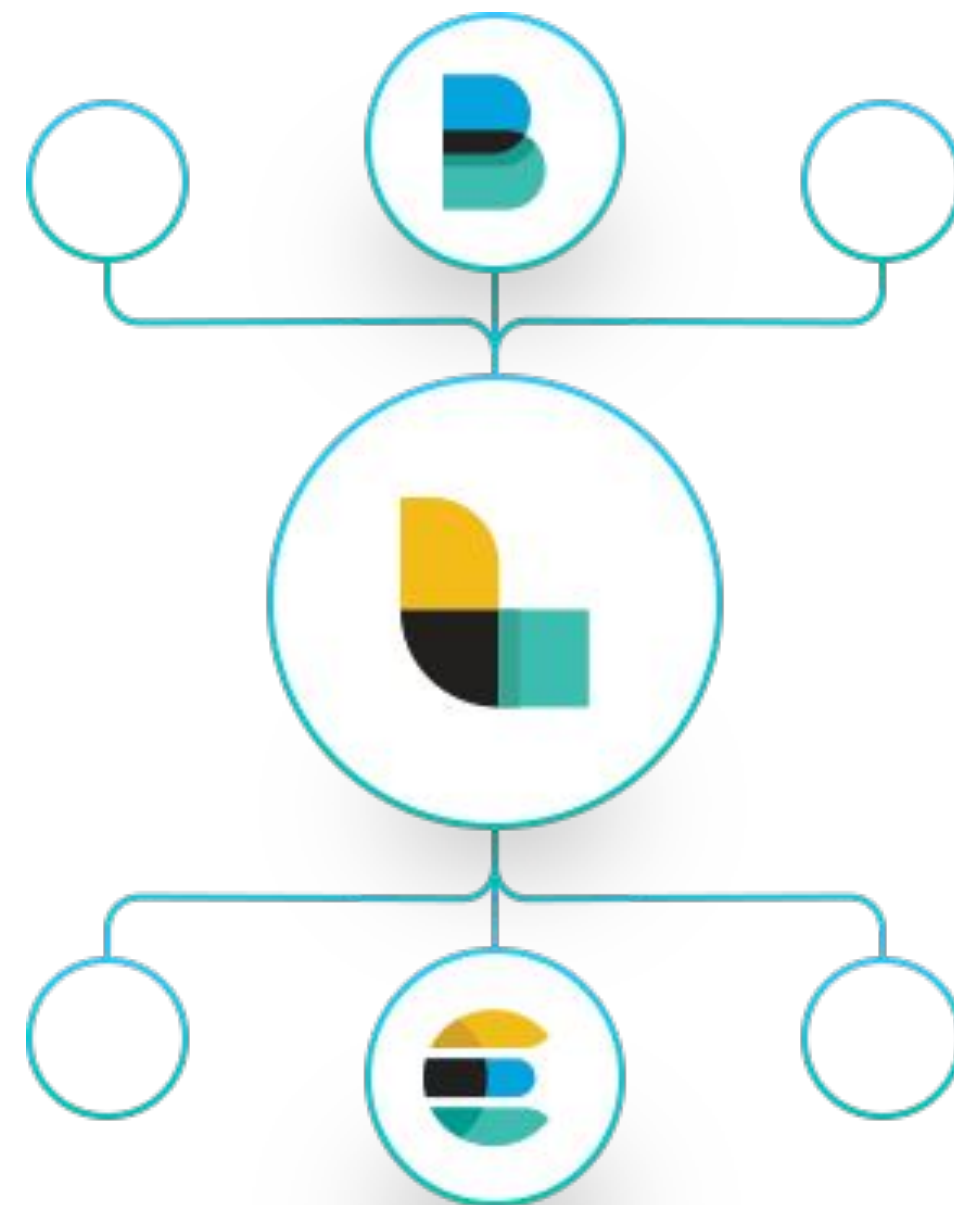
FunctionBeat
Serverless Shipper

Plus, more than 70 community Beats and growing...



Logstash

ETL for Elasticsearch



Ingest data of all shapes,
sizes, and sources

Parse and dynamically
transform data

Transport data to any
output

Secure and encrypt data
inputs

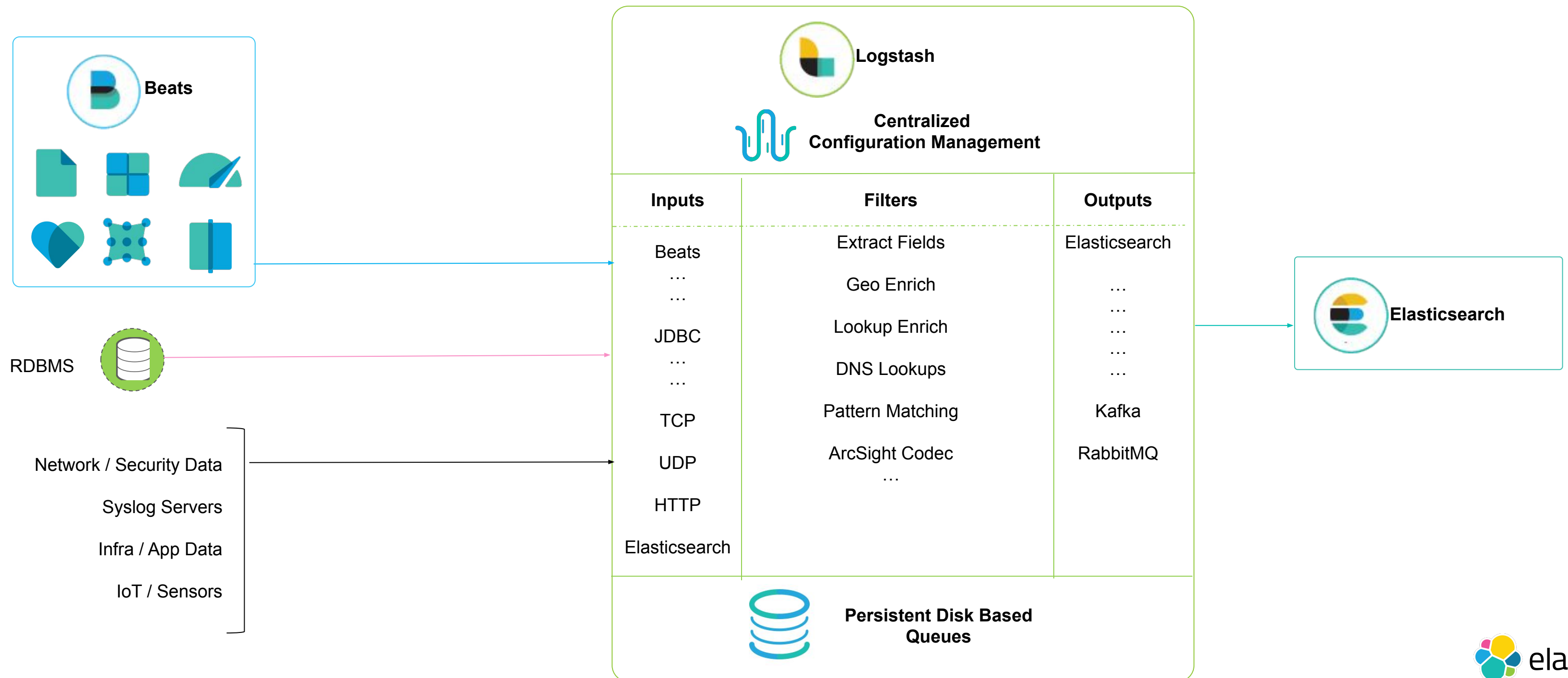
Build your own pipelines

Lots of plugins



Logstash

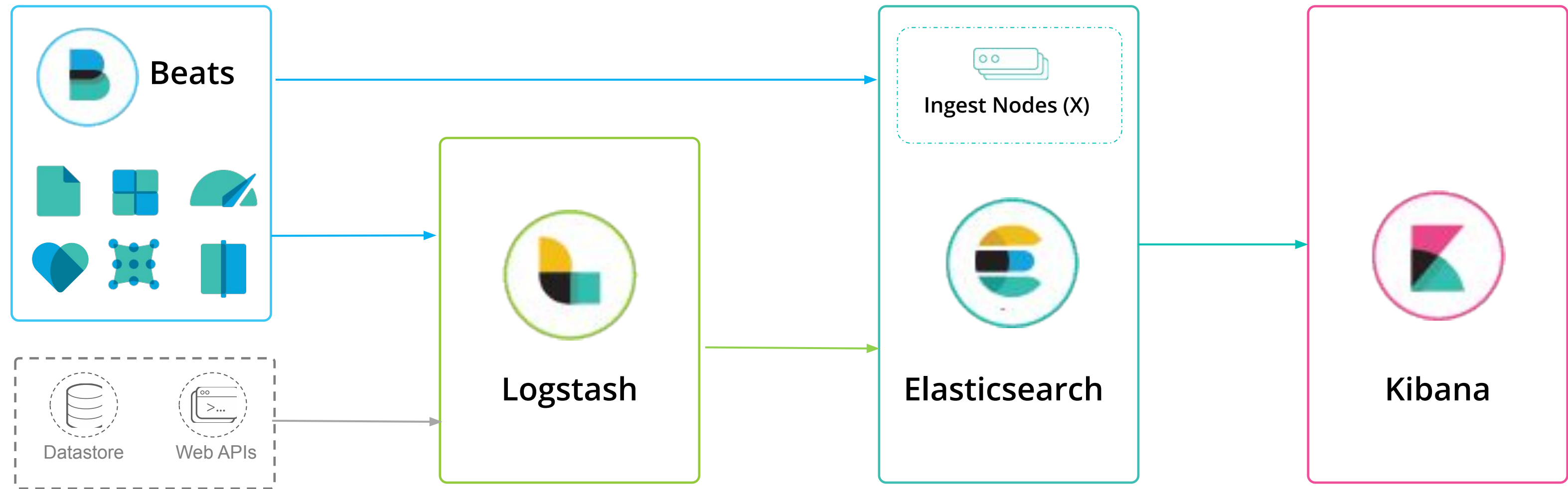
Normalize and Enrich Data before Indexing





Elastic **Stack**

Ingest, Store, Search, Visualise



Agenda

- 1 Introducción al stack de Elastic
- 2 Preparación del laboratorio
- 3 Introducción a Kibana
- 4 Elastics Maps
- 5 Elasticsearch Geo
- 6 Web mapping con Elasticsearch

Dónde podéis ejecutar el laboratorio



Cuenta de prueba en Elastic Cloud



- Crea una cuenta gratuita de dos semanas en cloud.elastic.co
- Crear un *deployment* nuevo (**detalles**)
 - los valores por defecto están bien
 - para este taller puedes eliminar del despliegue **Integrations Server** y **Enterprise Search**
- Para cargar los datos de prueba (open sky):
 - Puedes subir datos en tiempo real con la imagen docker o script opensky-loader (**detalles**)
 - Puedes usar un **dataset estático** en formato GeoJSON o generar uno nuevo (**detalles**)



Usando Docker Compose en OSGeo Live

- Prerrequisito: instalar **Docker** y **Docker Compose**
- Descargar laboratorio: <https://github.com/jsanz/elastic-workshop>

```
git clone https://github.com/jsanz/elastic-workshop.git
```

- Ajustar parámetros en **.env**
 - Puerto de OpenSky viewer (puerto 80 no disponible en OSGeo Live)
 - **OPENSKY_VIEWER_PORT=8080**
 - (Opcional) contraseñas de usuario
 - **ELASTIC_PASSWORD=changeme**
 - **KIBANA_PASSWORD=changeme**

Arrancar aplicaciones



```
$ cd elastic-workshop/lab  
$ sudo sysctl -w vm.max_map_count=262144 # ajusta la memoria virtual  
$ docker compose pull # descarga las imágenes del stack  
$ docker compose build # construye las imágenes opensky  
$ docker compose up -d # arranca todos los contenedores
```

Otros comandos útiles

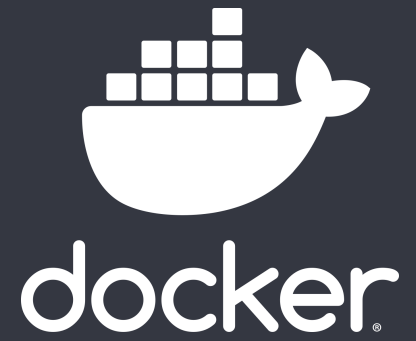
```
$ docker compose ps # muestra información de estado  
$ docker compose logs -f kibana # mostrar logs  
$ docker compose down # apaga todos los contenedores y servicios  
$ docker compose restart opensky-viewer # resetea un contenedor  
$ docker compose stop opensky-viewer # para un contenedor  
$ docker volume ls # lista los volúmenes (discos duros de docker)  
$ docker volume rm [nombre-volume] # borra un volumen
```

Comprobar estado de los contenedores

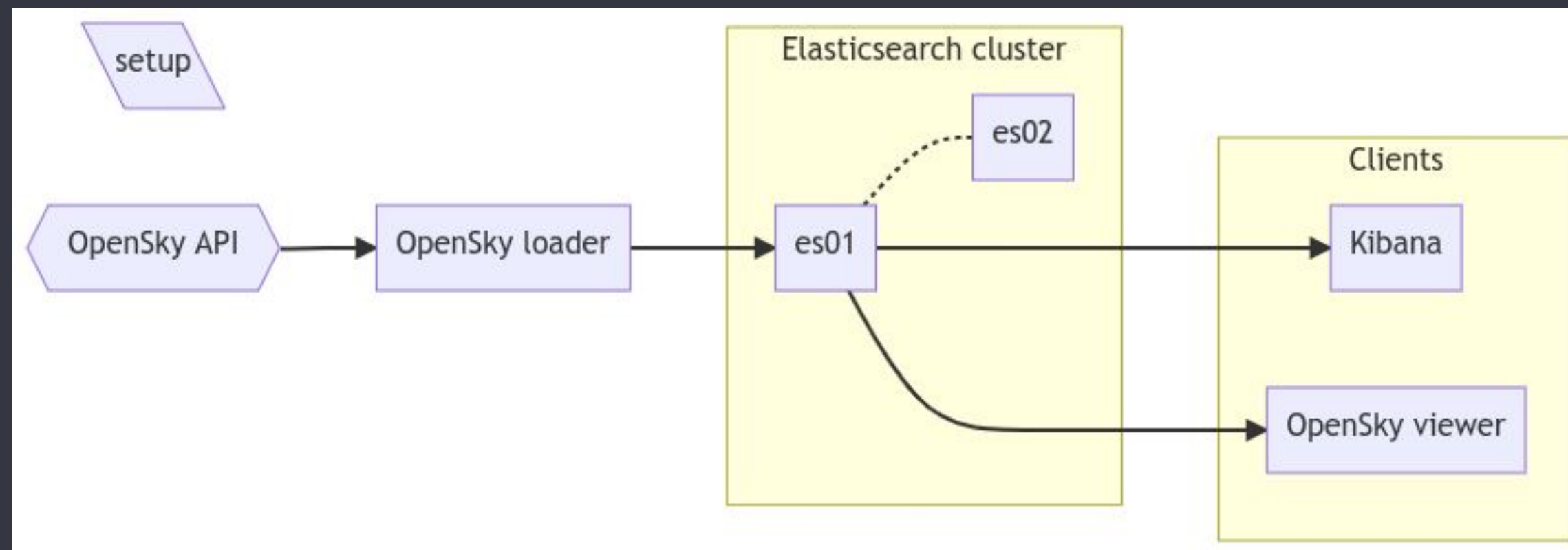


```
user@osgeolive: ~/elastic-workshop/lab
user@osgeolive:~/elastic-workshop/lab$ docker compose ps
NAME                                COMMAND                                SERVICE    STATUS    PORTS
user@osgeolive:~/elastic-workshop/lab$ docker compose up -d
[+] Running 7/7
  :: Network lab_default             Created
  :: Container lab-setup-1           Healthy
  :: Container lab-es01-1            Healthy
  :: Container lab-es02-1            Healthy
  :: Container lab-kibana-1          Started
  :: Container opensky-loader        Started
  :: Container opensky-viewer        Started
user@osgeolive:~/elastic-workshop/lab$ docker compose ps
NAME                                COMMAND                                SERVICE    STATUS    PORTS
lab-es01-1                          "/bin/tini -- /usr/l..."           es01       running (healthy) 0.0.0.0:9200->9200/tcp, :::9200->9200/tcp
lab-es02-1                          "/bin/tini -- /usr/l..."           es02       running (healthy) 9300/tcp
lab-kibana-1                        "/bin/tini -- /usr/l..."           kibana     running (healthy) 0.0.0.0:5601->5601/tcp, :::5601->5601/tcp
lab-setup-1                         "/bin/tini -- /usr/l..."           setup      exited (0)
opensky-loader                      "docker-entrypoint.s..."           opensky-loader running
opensky-viewer                      "docker-entrypoint.s..."           opensky-viewer running          0.0.0.0:8080->3000/tcp, :::8080->3000/tcp
user@osgeolive:~/elastic-workshop/lab$
```

¿Qué tenemos?



- **es01** y **es02**: cluster de Elasticsearch, <https://localhost:9200>
- **opensky-loader**: cargador de datos
- **kibana**: <http://localhost:5601>
- **opensky-viewer**: visualizador de datos, <http://localhost:8080>



Agenda

- 1 Introducción al stack de Elastic
- 2 Preparación del laboratorio
- 3 Introducción a Kibana
- 4 Elastics Maps
- 5 Elasticsearch Geo
- 6 Web mapping con Elasticsearch



Kibana

Algunos conceptos básicos del
funcionamiento de Kibana



Developer Tools

Consola

Permite ejecutar consultas a Elasticsearch con autocompletado, indentación, histórico, etc.

Search profiler

Estadísticas de rendimiento de consultas

Grok debugger

Ayuda para crear expresiones grok para Logstash y otras herramientas de carga

Painless lab

Entorno para probar scripts escritos en painless

The screenshot displays the Elastic Developer Tools interface. At the top, the Elastic logo and a search bar are visible. Below the navigation bar, the 'Console' tab is active, showing a REST client interface. The left pane contains a REST client request: a GET request to 'flight_tracking*/_search' with a JSON body. The right pane shows the corresponding JSON response, which includes search statistics and a list of hits. The response is formatted with syntax highlighting and line numbers.

```
1 GET flight_tracking*/_search
2 {
3   "query": {
4     "bool": {
5       "must": [
6         {
7           "match_all": {}
8         }
9       ],
10      "filter": {
11        "geo_bounding_box": {
12          "location": {
13            "top_left": {
14              "lat": 40.666,
15              "lon": -73.824
16            },
17            "bottom_right": {
18              "lat": 40.62,
19              "lon": -73.744
20            }
21          }
22        }
23      }
24    }
25  }
26 }
```

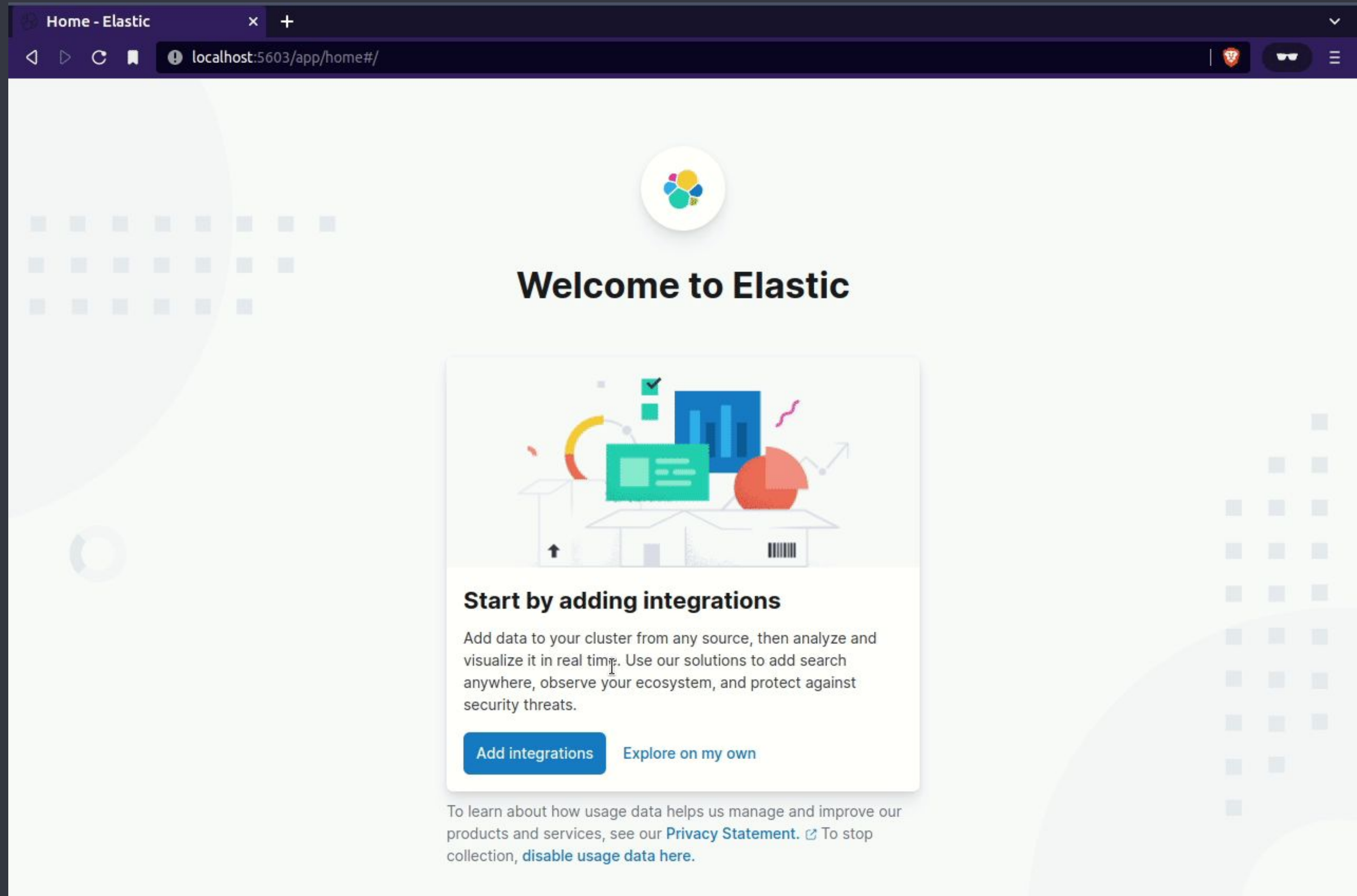
```
1 {
2   "took" : 1185,
3   "timed_out" : false,
4   "_shards" : {
5     "total" : 4,
6     "successful" : 4,
7     "skipped" : 0,
8     "failed" : 0
9   },
10  "hits" : {
11    "total" : {
12      "value" : 216,
13      "relation" : "eq"
14    },
15    "max_score" : 1.0,
16    "hits" : [
17      {
18        "_index" : "flight_tracking_2022-05-11",
19        "_id" : "flcStIAB4XM30LHftGv_",
20        "_score" : 1.0,
21        "_source" : {
22          "@timestamp" : 1652288566382,
23          "onGround" : false,
24          "spi" : false,
25          "icao24" : "aa2ca9",
26          "callsign" : "AAL235",
27          "originCountry" : "United States",
28          "timePosition" : 1652288530000,
29          "lastContact" : 1652288530000,
30          "location" : {
31            "lat" : 40.6218,
32            "lon" : -73.7733
33          },
34          "baroAltitude" : -45.72,
35          "velocity" : 65.03,
36          "heading" : 30.42,
37          "verticalRate" : -3.58,
38          "geoAltitude" : 76.2,
39          "transponderCode" : "2504"
40        }
41      },
42      {
43        "_index" : "flight_tracking_2022-05-11",
44        "_id" : "jFcTtIAB4XM30LHfr4QC",
45        "_score" : 1.0,
46        "_source" : {
47          "@timestamp" : 1652288630372,
48          "onGround" : false,
49          "spi" : false,
50          "icao24" : "a24720",
51          "callsign" : "RPA5658",
```



Data Views

- Componente lógico que **agrupa** índices usando un **patrón**
 - `my_application_logs_*`
- Definir **formato** a campos: números, monedas, imágenes, URL, ...
- Definir **campo temporal** para filtros (opcional)
- **Campos dinámicos (runtime)** para cálculos en tiempo de consulta

Crear un Data View

A screenshot of a web browser showing the Elastic home page. The browser's address bar displays 'localhost:5603/app/home#/' and the page title is 'Home - Elastic'. The Elastic logo is centered at the top. Below it, the text 'Welcome to Elastic' is displayed. A central card features an illustration of data visualizations (bar charts, pie charts, line graphs) and a server rack. The card is titled 'Start by adding integrations' and contains a paragraph of text. At the bottom of the card are two buttons: 'Add integrations' and 'Explore on my own'. Below the card, there is a link to the 'Privacy Statement' and a link to 'disable usage data here'.

Home - Elastic

localhost:5603/app/home#/

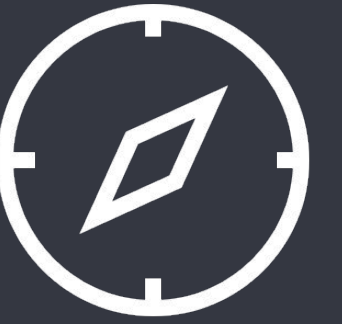
Welcome to Elastic

Start by adding integrations

Add data to your cluster from any source, then analyze and visualize it in real time. Use our solutions to add search anywhere, observe your ecosystem, and protect against security threats.

[Add integrations](#) [Explore on my own](#)

To learn about how usage data helps us manage and improve our products and services, see our [Privacy Statement](#). To stop collection, [disable usage data here](#).



Discover

- Herramienta de **exploración** rápida
- **Rango temporal** y **actualización** automática*
- **Barra de búsqueda** usando Kibana Query Language o Lucene*
- **Filtros***
- Campos como **columnas**
- **Estadísticas** de los campos
- **Inspect** tool: estadísticas, petición completa y respuesta
- **Guardar** tu búsqueda para usarla más tarde en cuadros de mando

* común a muchos otros elementos de Kibana

RYP*

KQL Refresh

+ Add filter

flight_tracking* ▾

Search field names

Filter by type 0 ▾

Selected fields 5

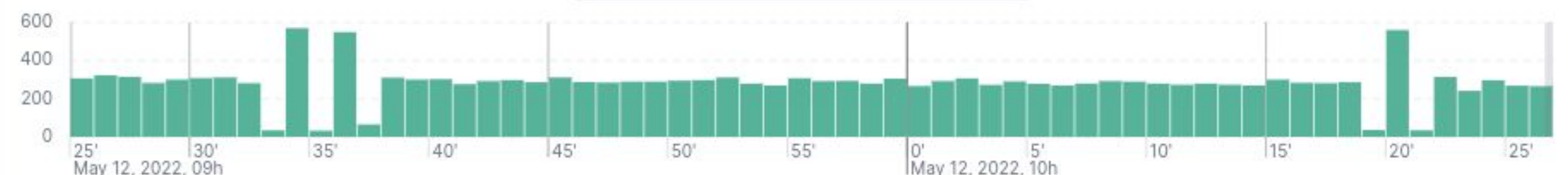
- callsign
- onGround
- originCountry
- geoAltitude
- velocity

Available fields 13

- _id
- _index
- _score
- @timestamp
- baroAltitude
- heading
- icao24

17,490 hits

Documents Field statistics BETA



May 12, 2022 @ 09:25:07.803 - May 12, 2022 @ 10:26:39.729

Columns 1 field sorted

	timePosition	callsign	onGround	originCountry	geoAltitude	velocity
☐	May 12, 2022 @ 10:26:33.000	RYP74LE	false	Ireland	198.12	68.07
☐	May 12, 2022 @ 10:26:09.000	RYP2MM	false	Ireland	10,302.24	204.44
☐	May 12, 2022 @ 10:26:09.000	RYP89JF	false	Ireland	11,612.88	204.44
☐	May 12, 2022 @ 10:26:09.000	RYP2KQ	false	Ireland	11,010.9	206.36
☐	May 12, 2022 @ 10:26:09.000	RYP4QZ	false	Ireland	9,631.68	202.93

Rows per page: 100

< 1 2 3 4 5 >

Lens

Tus datos delante tuyo

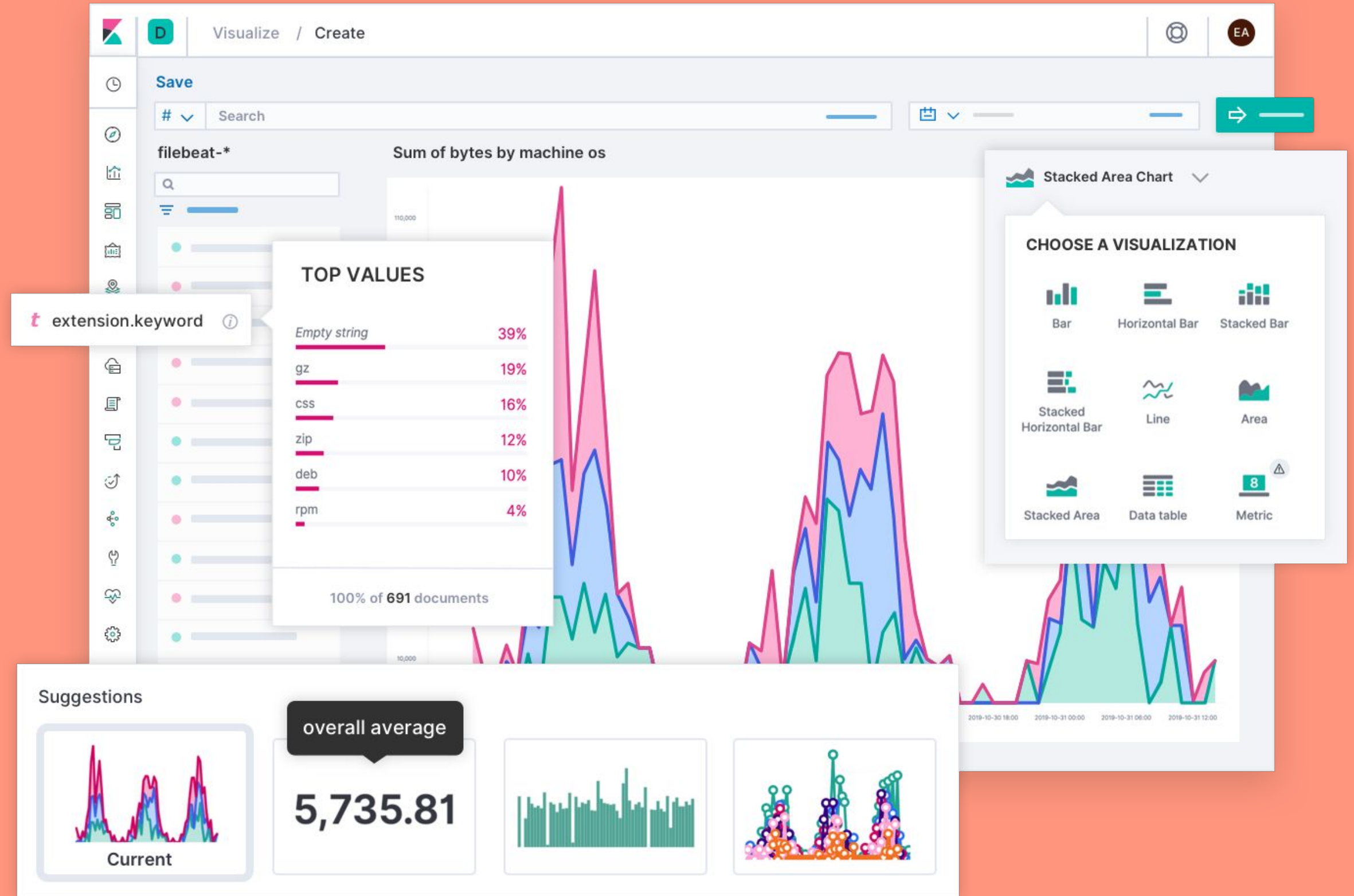
Exploración de campos con un clic

Arrastrar y soltar

De cero a gráficas útiles con un simple gesto de ratón

Sugerencias inteligentes

Lens analiza tus datos y te sugiere gráficas apropiadas a su distribución



New visualization



Lens

Create visualizations with our drag and drop editor. Switch between visualization types at any time. *Recommended for most users.*



Maps

Create and style maps with multiple layers and indices.



TSVB

Perform advanced analysis of your time series data.



Custom visualization

Use Vega to create new types of visualizations. *Requires knowledge of Vega syntax.*



Aggregation based

Use our classic visualize library to create charts based on aggregations.

[Explore options →](#)

Tools



Text

Add text and images to your dashboard.



Controls

Add dropdown menus and range sliders to your dashboard.

Want to learn more? [Read documentation](#)

... y más



Dashboards

- Combinación de múltiples visualizaciones: **paneles**
- Time Range + Search Bar + Filters
- Los paneles permiten definir filtros para hacer **drill down**
- Los paneles pueden tener rangos temporales **personalizados**
- **Compartir**
- **Exportar** como PDF o PNG

Cumbre Vieja eruption

- Location: La Palma, Canary Islands
- Data sources: Lava flows and buildings: La Palma GeoData; Earthquakes: IGN
- Use webcams
- Dashboard map



Source: @volcanos
Compiled by Jorge Saez, Kibera Team, Elcivir

Total count

8,602

earthquakes

Average magnitude

2.7

Drone surveys

72

Surveys

Lava footprints

84

days

Total lava affected area

1,190

Hectare

Total affected buildings

12,253

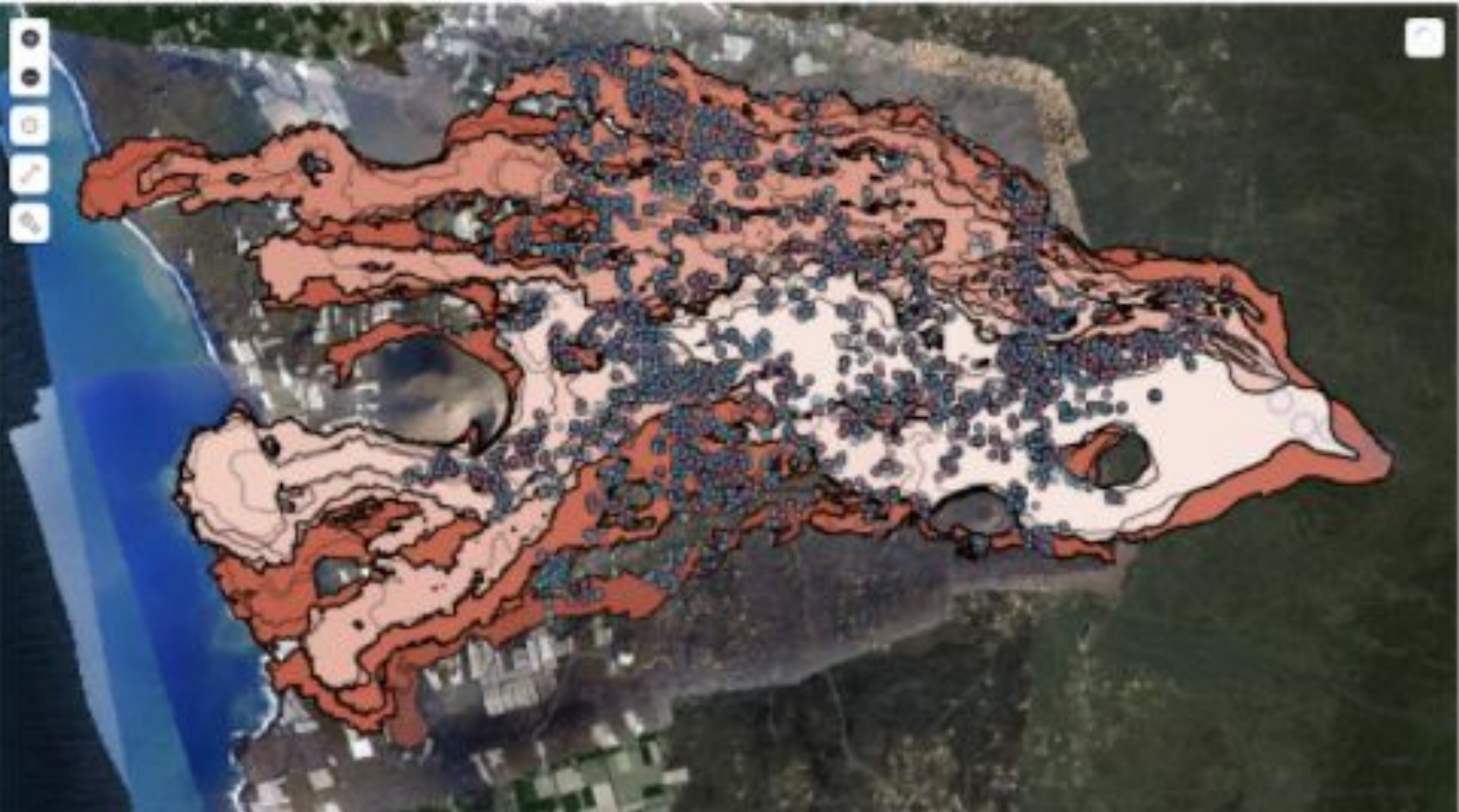
Buildings

Total affected built area

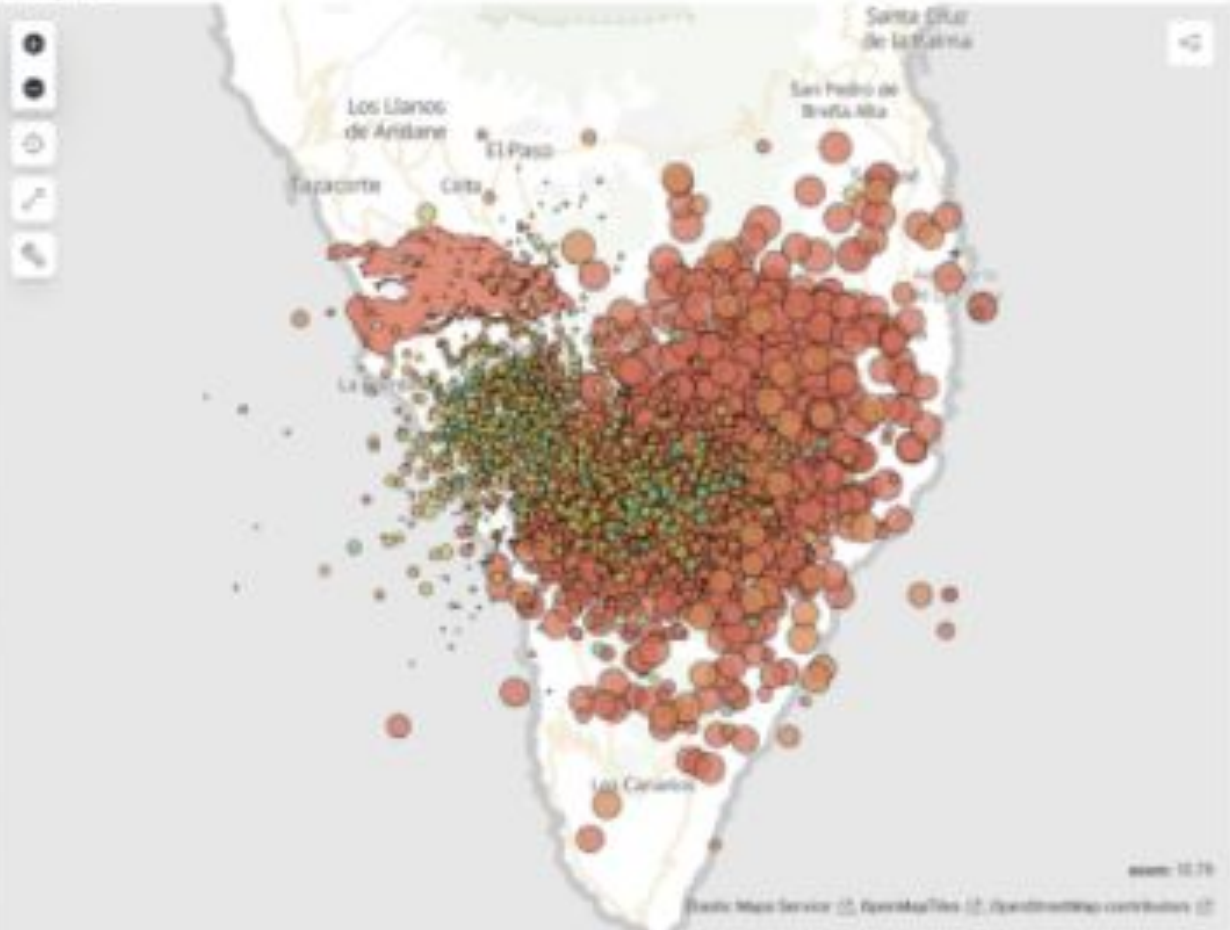
139

Hectare

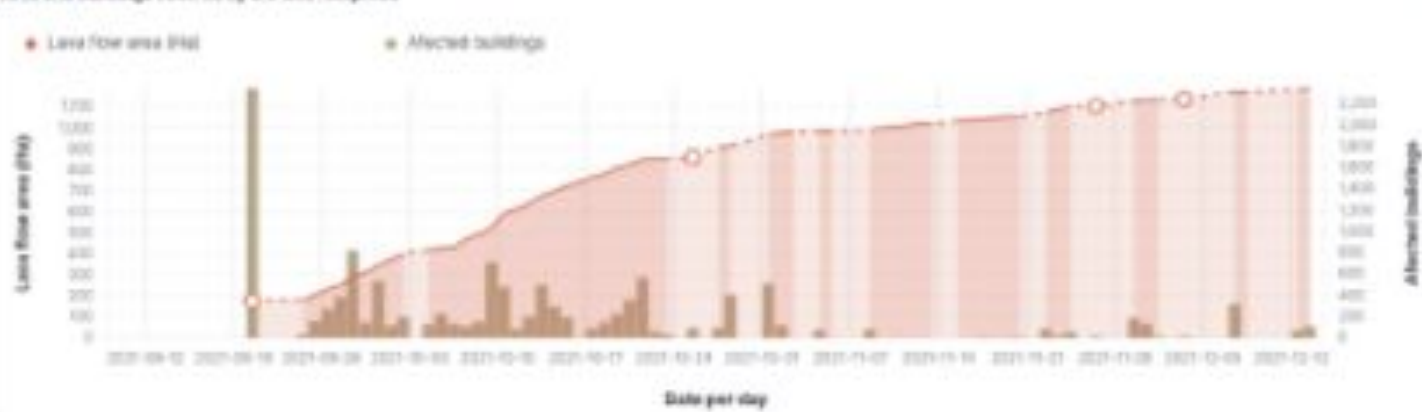
Lava flow footprints, affected buildings by type and area, and drone imagery



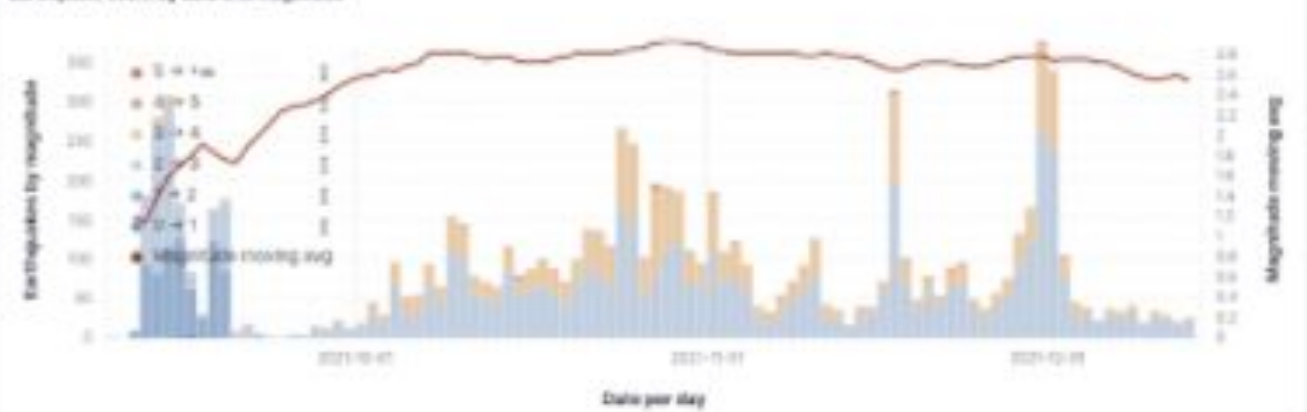
Earthquakes



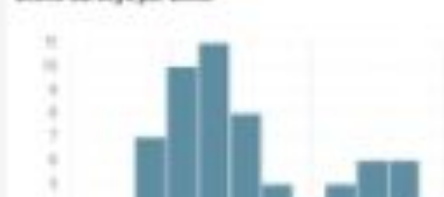
Area and buildings covered by the lava footprints



Earthquake count by date and magnitude



Drone surveys per week



Buildings count by type and number of floors

Type	1+ floors	2+ floors	≥3 floors
Building	4,745	701	94
Flat roof	3,290	369	61
Shed-like roof	2,645	332	33

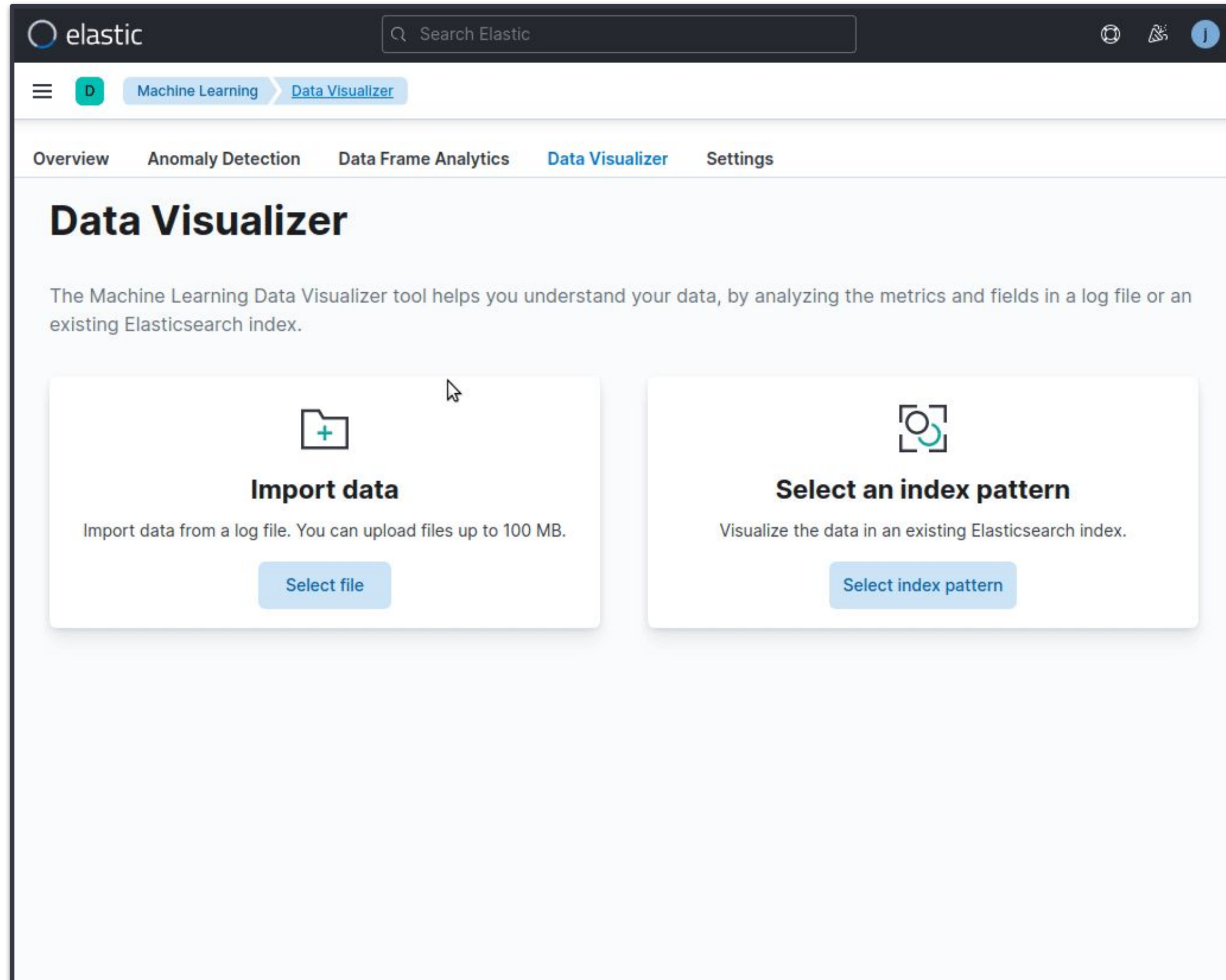
Earthquake count by depth (km) and magnitude



Earthquakes depth over time (km)



Ingest with Kibana: CSV file upload



Agenda

- 1 Introducción al stack de Elastic
- 2 Preparación del laboratorio
- 3 Introducción a Kibana
- 4 **Elastics Maps**
- 5 Elasticsearch Geo
- 6 Web mapping con Elasticsearch

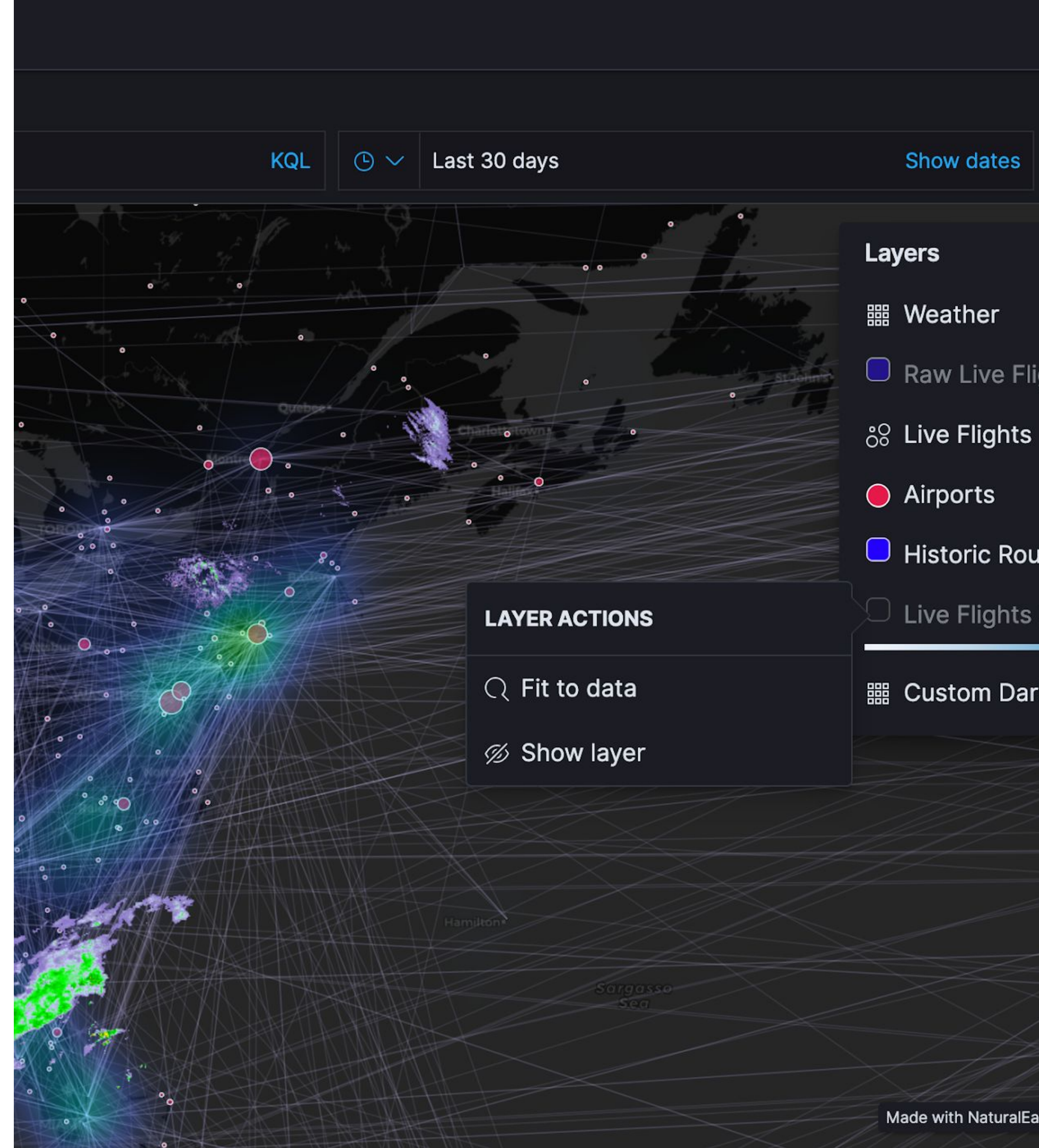
Elastic Maps

Acercando los Sistemas de
Información Geográfica a Kibana



Elastic Maps

- Interfaz amigable
- MapLibre
- `geo_point` y `geo_shape`
- Agregaciones: heat map, clustering, grids, geoline
- Estilos desde los datos
- Herramientas para dibujar, filtrar, medir
- Uso independiente o en dashboards y canvas
- Componente para otras aplicaciones de Kibana



Elastic Maps Service

maps.elastic.co

- 18 niveles de zoom
 - Mapas de todo el mundo o de tu ciudad
- Tres estilos de mapas base
 - Muestra tus datos sobre un mapa base claro, oscuro, o colorido
- Límites Administrativos
 - Más de 60 capas de regiones administrativas de países de todo el mundo

 Elastic Maps Service

elastic.co

 Tile Layers

Road map

Road map - desaturated

[Road map - dark](#)

 Vector Layers

World Countries

Australia States

Austria States

Belarus Regions

Belgium Provinces

Belgium Regions

Brazil States

Canada Provinces

China Provinces

Croatia Counties

Denmark Regions

Estonia Counties

Finland Regions

France Departments

Germany States

Hungary Counties

India States and Territo...

Ireland Counties

Italy Provinces

[Japan Prefectures](#)

Luxembourg Cantons

Netherlands Provinces

Norway Counties

Poland Voivodeships

Portugal Districts

Slovakia Regions



Japan Prefectures

© OpenStreetMap contributors, Elastic Maps Service



Search...

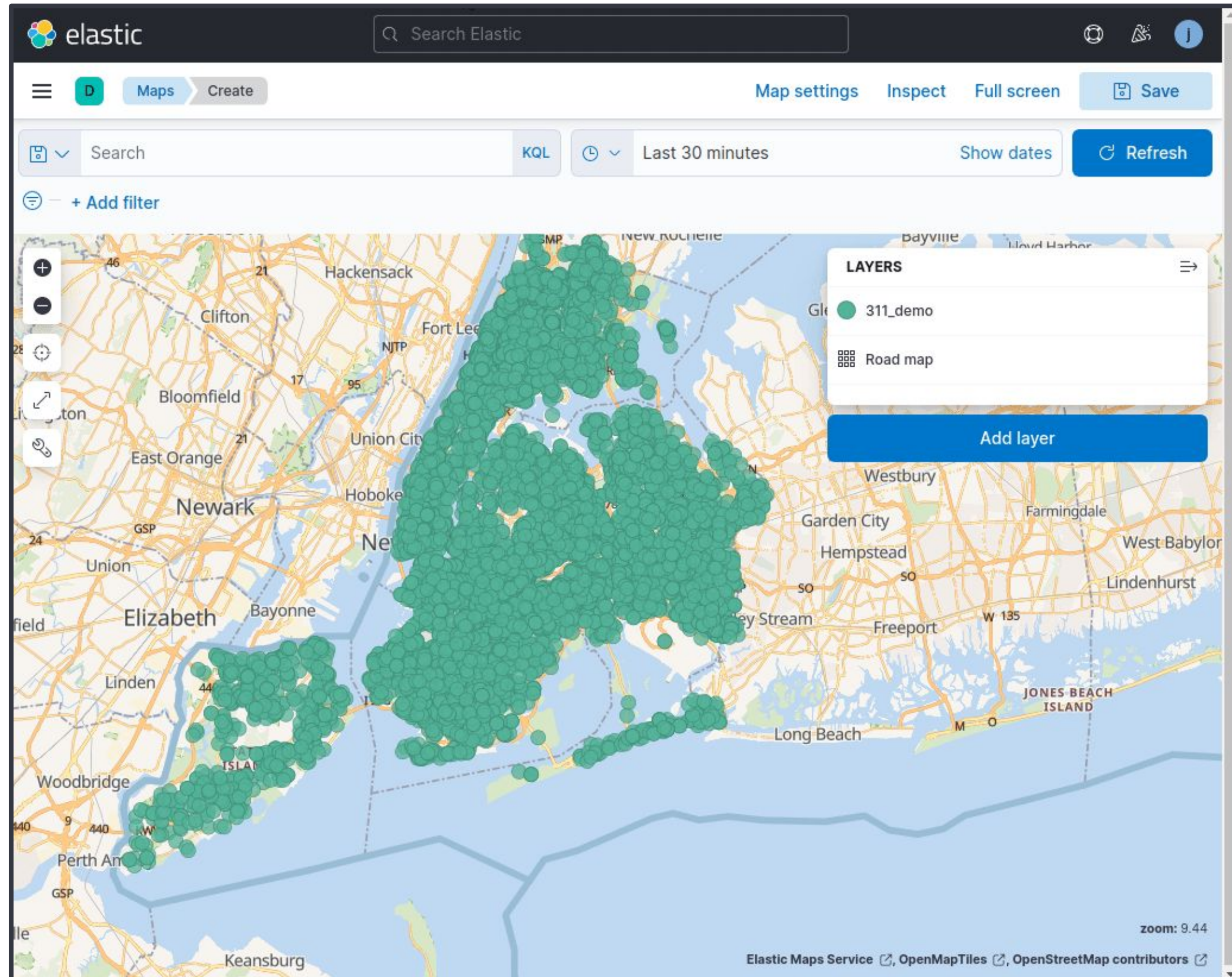
ISO 3166-2 code (iso_3166_2)

JP-01

Dantai code (dantai)

010006

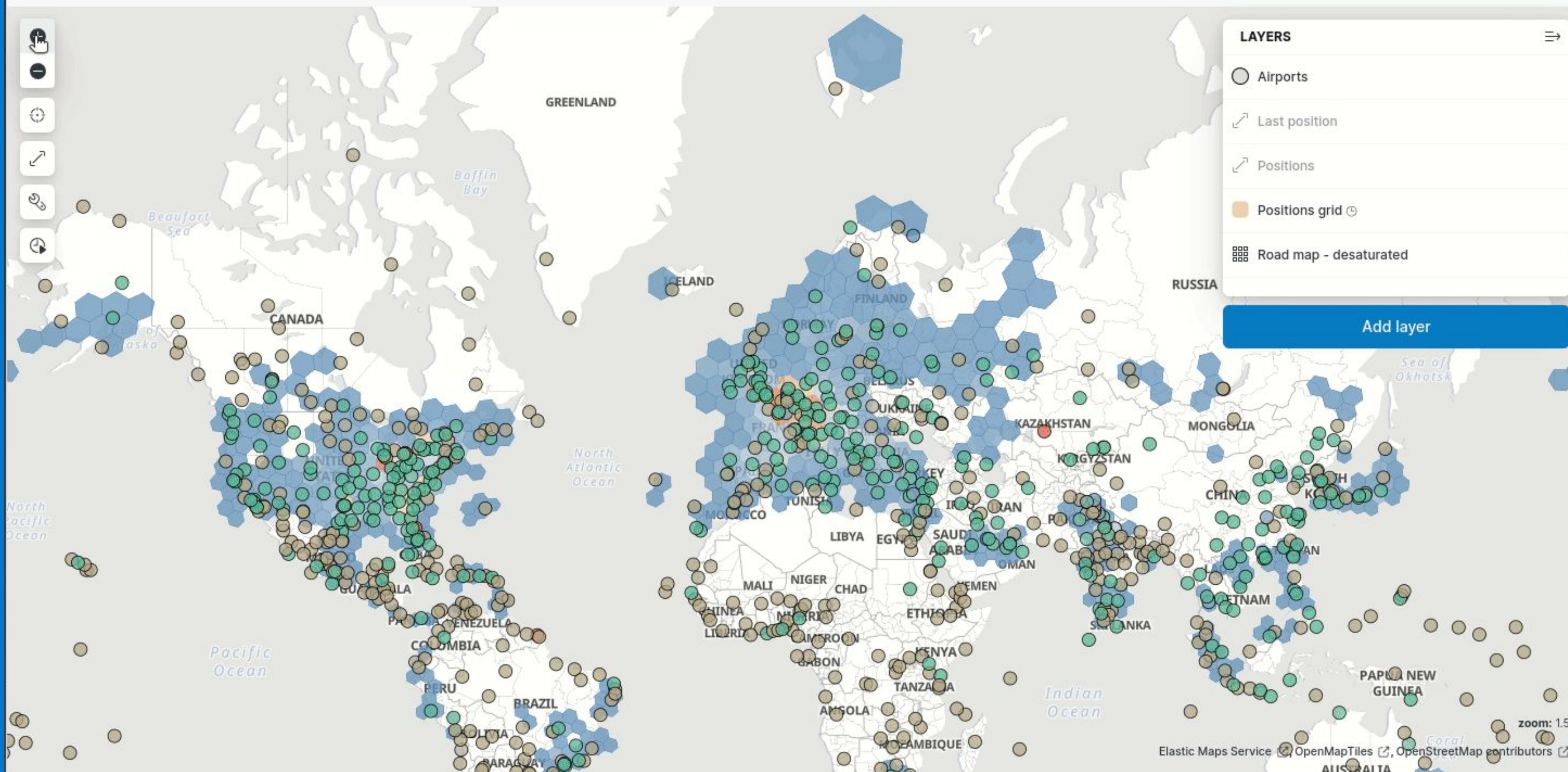
Ingest with Kibana: GeoJSON and Shapefile upload



- Docs 
- Tutorial 

Hora de practicar

- Crea un **mapa** nuevo
- Sube datos a tu *cluster*
 - **Aeropuertos** (desde DevTools o GeoJSON [detalles](#))
 - **Posiciones** (vía cargador o fichero estático)
- Capa **aeropuertos**
- Capa de **grid (hex, tile o cluster) de posiciones** (zoom 0 a 6)
- Capa de **posiciones individuales** (zoom 7 en adelante)
- Añadir **tooltips**, jugar con la **simbología**, etc
- Añadir capa con la agregación de línea (**tracks**)



Agenda

- 1 Introducción al stack de Elastic
- 2 Preparación del laboratorio
- 3 Introducción a Kibana
- 4 Elastics Maps
- 5 Elasticsearch Geo
- 6 Web mapping con Elasticsearch

Elasticsearch geospatial data types

- `geo_point` 
 - A single pair of latitude and longitude **coordinates**
 - Can be inserted as an object, WKT, array, geohash
- `geo_shape` 
 - Supports any **lat/lon** geometry type, incl. envelope and circle
 - Inserted with GeoJSON or WKT notation
- `shape` 
 - Supports any **cartesian** geometry type
 - Inserted with GeoJSON or WKT notation

Geo Enrichment

Adding a geo dimension to your data

- Transform data at ingest time
- Enrich IP addressed with estimated location
- Lookup location based on another index e.g. postcodes
- Tag documents by [matching with polygons in another index](#) e.g. local authority boundaries
- Convenient UI in Kibana

The screenshot shows the 'Edit pipeline' interface in Kibana for a pipeline named '311_demo-pipeline'. The interface includes a sidebar with navigation links for 'Stack Management', 'Ingest Node Pipelines', and 'Edit pipeline'. The main content area displays the pipeline configuration. At the top, there are fields for 'Name' (311_demo-pipeline), 'Description' (Ingest pipeline created by text structure), and a checkbox for 'Add version number'. Below this is the 'Processors' section, which lists several processors: 'CSV' (Extracts CSV values from 'message' to 'Unique Key', 'Created Date', 'Closed Date'), 'Convert' (Converts 'BBL' to type 'long'), 'Convert' (Converts 'Incident Zip' to type 'long'), 'Convert' (Converts 'Latitude' to type 'double'), 'Convert' (Converts 'Longitude' to type 'double'), 'Convert' (Converts 'Unique Key' to type 'long'), and 'Remove' (Removes 'message'). The 'Set' processor at the bottom is highlighted with a red box, showing its configuration: 'Sets value of 'location' to '{{Latitude}},{{Longitude}}''. Below the processors is a section for 'Failure processors' with an 'Add a processor' button. At the bottom right, there are buttons for 'Save pipeline' and 'Cancel'.

Ingest with ogr2ogr

<https://gdal.org/drivers/vector/elasticsearch.html>

- ogr2ogr can read and write into Elasticsearch
- Support for custom mapping definitions
- Blog posts:
 - [How to ingest geospatial data into Elasticsearch with GDAL](#)
 - [Import OSM data into Elasticsearch with ogr2ogr and Docker](#)



Have you used [Elastic Maps](#) in Kibana yet? I am very excited about multi-layer support. Heat maps, vector layers from the Elastic Maps Service, individual documents all in the same interface! What a fantastic way to and visualize your data.

But what about geospatial data that's not in Elasticsearch? Maybe you overlay a shapefile of regional sales territories with sales aggregations. you have a CSV file of distribution center locations, and you want to get data into Elasticsearch, but configuring Filebeat or Logstash is not ideal for ingesting static datasets. Well, we have the perfect solution for you: GDAL.

[GDAL](#) (Geospatial Data Abstraction Library) contains command line tools that can convert geospatial data between over 75 different geospatial file formats including [Elasticsearch](#). GDAL can be [compiled from source](#) or [installed via package managers](#). GDAL can also be installed via [Homebrew OSGeo4Mac](#) (ex. `brew tap osgeo/osgeo4mac` && `brew install osgeo-gdal`). Note, you need to have GDAL v3.1 or later to ingest data into Elasticsearch 7.x.

Connecting to Elasticsearch

Once you've installed GDAL, open your command line or terminal window and try connecting to your Elasticsearch cluster using the `ogrinfo` tool. We'll use the URL with "ES:" to tell GDAL to use the Elasticsearch driver.

```
ogrinfo ES:http://localhost:9200/
```

Search

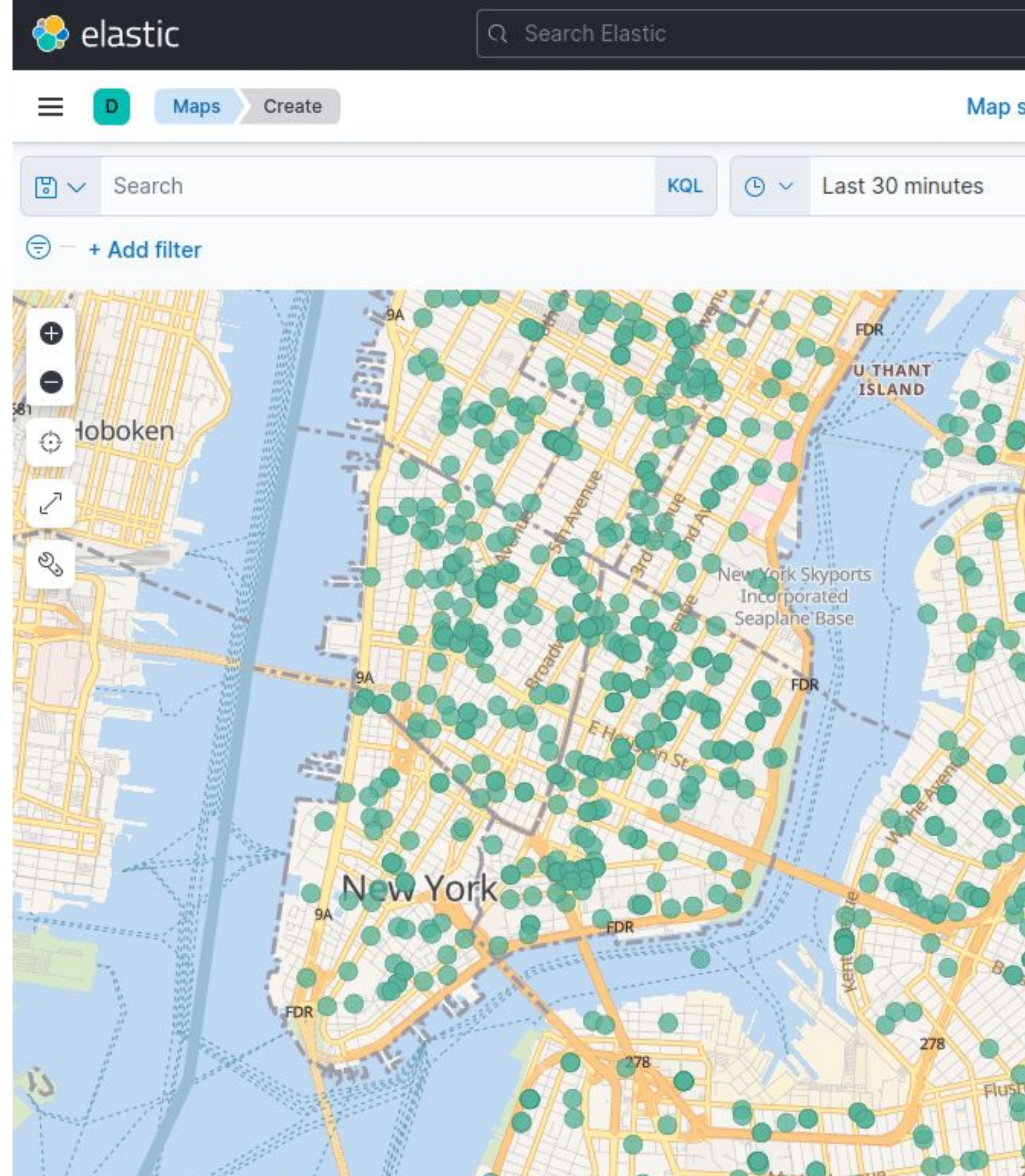
Filter documents with geospatial relationships

Geo Filters

- Bounding box
- Point and radius
- Polygon
- An indexed geo_shape

Plus every other Elasticsearch filter

- Boolean
- Range (numeric, date, IP)
- Unstructured text (stemming, fuzzy ...)



Aggregate

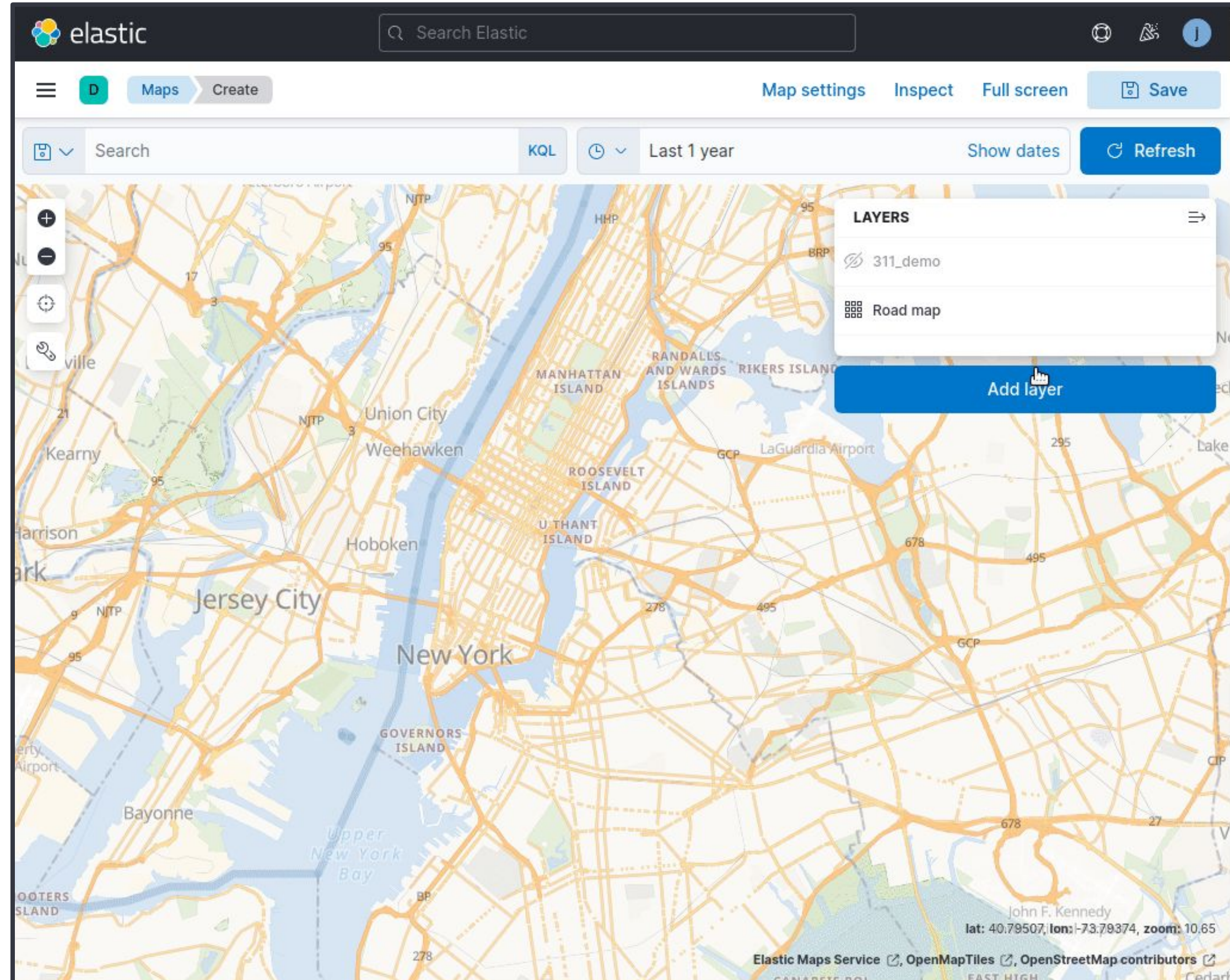
Geo Bucket

- Distance (rings) 📖
- Hash 📖
- Geotile 📖
- Geohex 📖

Geo Metric

- Centroid 📖
- Bounds 📖
- Geoline 📖

Many more 📖



OGC servers and Elasticsearch

Expose Elasticsearch indices as OGC services

GeoServer



GeoServer

[About](#) | [Blog](#) | [Download](#) |

GeoServer 2.20.x User Manual » Community modules » Elasticsearch data store

[previous](#) | [next](#) | [modules](#)

Elasticsearch data store

Elasticsearch is a popular distributed search and analytics engine that enables complex search features in near real-time. Default field type mappings support string, numeric, boolean and date types and allow complex, hierarchical documents. Custom field type mappings can be defined for geospatial document fields. The `geo_point` type supports point geometries that can be specified through a coordinate string, geohash or coordinate array. The `geo_shape` type supports Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon and GeometryCollection GeoJSON types as well as envelope and circle types. Custom options allow configuration of the type and precision of the spatial index.

This data store allows features from an Elasticsearch index to be published through GeoServer. Both `geo_point` and `geo_shape` type mappings are supported. OGC filters are converted to Elasticsearch queries and can be combined with native Elasticsearch queries in WMS and WFS requests.

Contents:

- [Elasticsearch data store](#)
 - [Configuration](#)
 - [Configuring data store](#)
 - [Configuring authentication](#)
 - [Configuring HTTPS/SSL](#)

Table Of Contents

- Elasticsearch data store
 - » Configuration
 - » Configuring data store
 - » Configuring authentication
 - » Configuring HTTPS/SSL
 - » Configuring layer
 - » Configuring logging
 - » Filtering
 - » Native queries
 - » Examples
 - » Aggregations
 - » Geohash grid aggregations
 - » Grid Strategy
 - » Basic
 - » Metric
 - » Nested
 - » Implementing a custom Grid Strategy
 - » FAQ

Continue Reading

- » Previous: Optimize rendering of complex polygons
- » Next: GeoMesa data store

pygeoapi

OpenAPI

Data publishing

Providers overview

Publishing vector data to OGC API - Features

Providers

Connection examples

Data access examples

Publishing raster data to OGC API - Coverages

Publishing map tiles to OGC API - Tiles

Publishing processes via OGC API - Processes

Publishing metadata to OGC API - Records

Publishing data to OGC API - Environmental Data Retrieval

Publishing files to a SpatioTemporal Asset Catalog

Customizing pygeoapi: plugins

Elasticsearch

Note

Elasticsearch 7 or greater is supported.

To publish an Elasticsearch index, the following are required in your index:

indexes must be documents of valid GeoJSON Features

index mappings must define the GeoJSON `geometry` as a `geo_shape`

```
providers:

- type: feature
- name: Elasticsearch
- data: http://localhost:9200/ne_110m_populated_places_simple
- id_field: geonameid
- time_field: datetimefield

```

This provider has the support for the CQL queries as indicated in the table above.

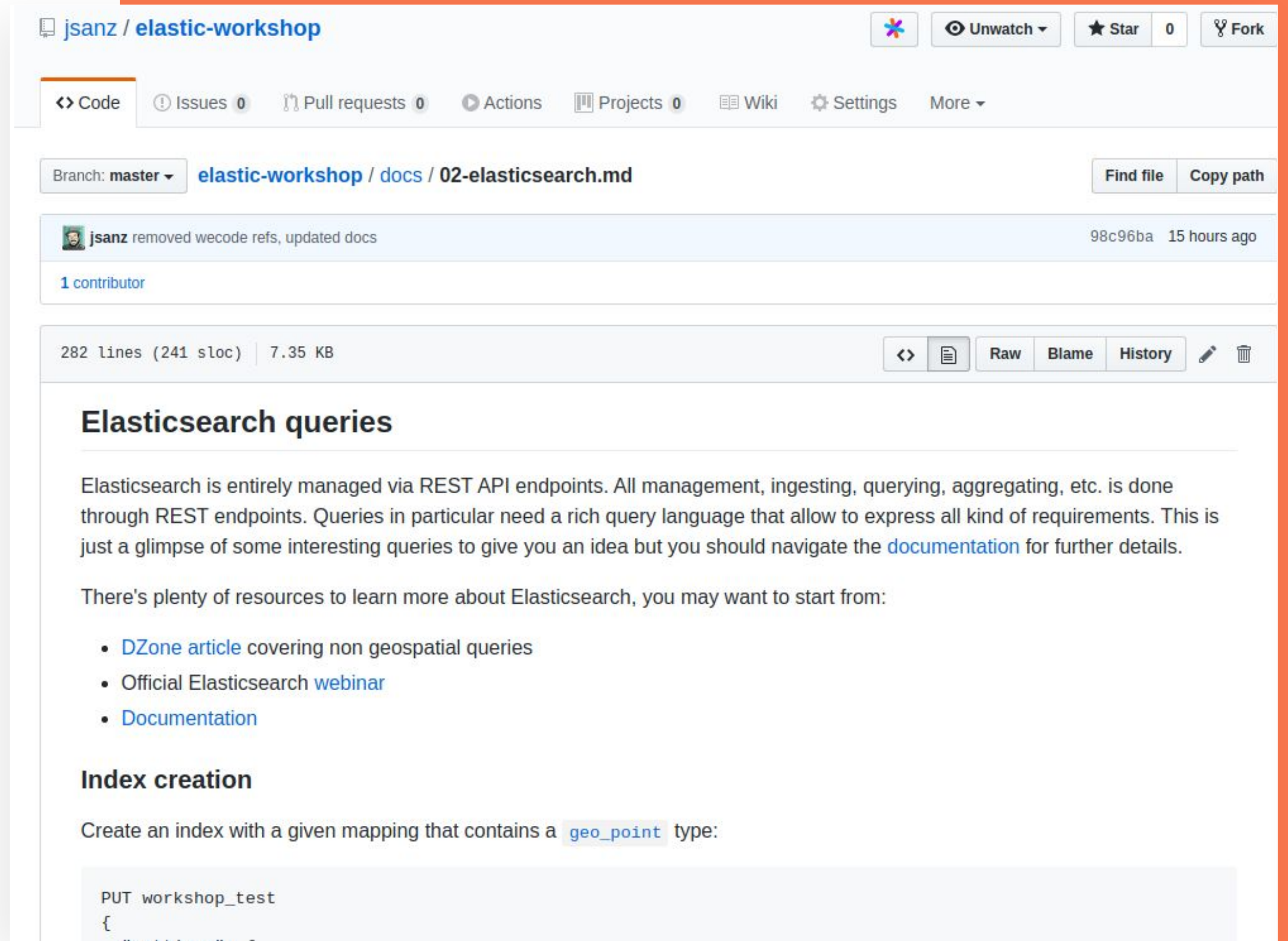


Hora de practicar

- Accede a las Dev Tools de Kibana
- Prueba las diferentes consultas del gui3n
 - Consultas b3sicas con Elasticsearch
 - Consultas geoespaciales

DSL Elasticsearch

- Kibana DevTools Console
 - curl, postman, ...
- Crear índices
 - Definir tipos, incluyendo campos geoespaciales
- Insertar
 - Añadir datos a tu índice
- Búsquedas
 - Obteniendo documentos
- Agregaciones
 - Agrupa tus datos



The screenshot shows a GitHub repository page for 'jsanz / elastic-workshop'. The repository has 0 issues, 0 pull requests, 0 actions, 0 projects, 0 wiki pages, and 0 settings. A commit by 'jsanz' is shown, titled 'removed wecode refs, updated docs', with commit hash '98c96ba' and a timestamp of '15 hours ago'. The file '02-elasticsearch.md' is selected, showing 282 lines (241 sloc) and 7.35 KB. The file content includes a section titled 'Elasticsearch queries' and another titled 'Index creation'.

Elasticsearch queries

Elasticsearch is entirely managed via REST API endpoints. All management, ingesting, querying, aggregating, etc. is done through REST endpoints. Queries in particular need a rich query language that allow to express all kind of requirements. This is just a glimpse of some interesting queries to give you an idea but you should navigate the [documentation](#) for further details.

There's plenty of resources to learn more about Elasticsearch, you may want to start from:

- [DZone article](#) covering non geospatial queries
- Official Elasticsearch [webinar](#)
- [Documentation](#)

Index creation

Create an index with a given mapping that contains a `geo_point` type:

```
PUT workshop_test
{
  "mappings": {
    "properties": {
      "location": {
        "type": "geo_point"
      }
    }
  }
}
```

Elasticsearch geo

Consultas geoespaciales

- Búsquedas
 - Encontrar documentos por punto/radio, bounding box, polígonos arbitrarios
- Agregación de métricas
 - Encontrar el centroide o la caja envolvente de tu búsqueda
- Agregación en grupos
 - Agrupa tus resultados por características geográficas como anillos o cuadrículas

The screenshot shows a GitHub repository page for 'jsanz / elastic-workshop'. The file '03-elasticsearch-geo.md' is selected, showing its content. The file has 191 lines (154 sloc) and is 5.17 KB. The content is titled 'Elasticsearch Geospatial Queries' and includes a 'Search' section with the instruction 'Find documents in your index using geospatial conditions.' Below this, there is a 'Point and radius query' section with the instruction 'With the `geo_distance` query type get the positions near Barajas airport:'. A code block shows a GET request for 'flight_tracking*/_search' with a query object containing 'geo_distance' parameters: 'distance' (5km) and 'location' (lat: 40.469674, lon: -3.559828).

jsanz / elastic-workshop

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Settings More

Branch: master elastic-workshop / docs / 03-elasticsearch-geo.md Find file Copy path

jsanz added geo queries content 375beef on Feb 5

1 contributor

191 lines (154 sloc) 5.17 KB

Elasticsearch Geospatial Queries

Search

Find documents in your index using geospatial conditions.

Point and radius query

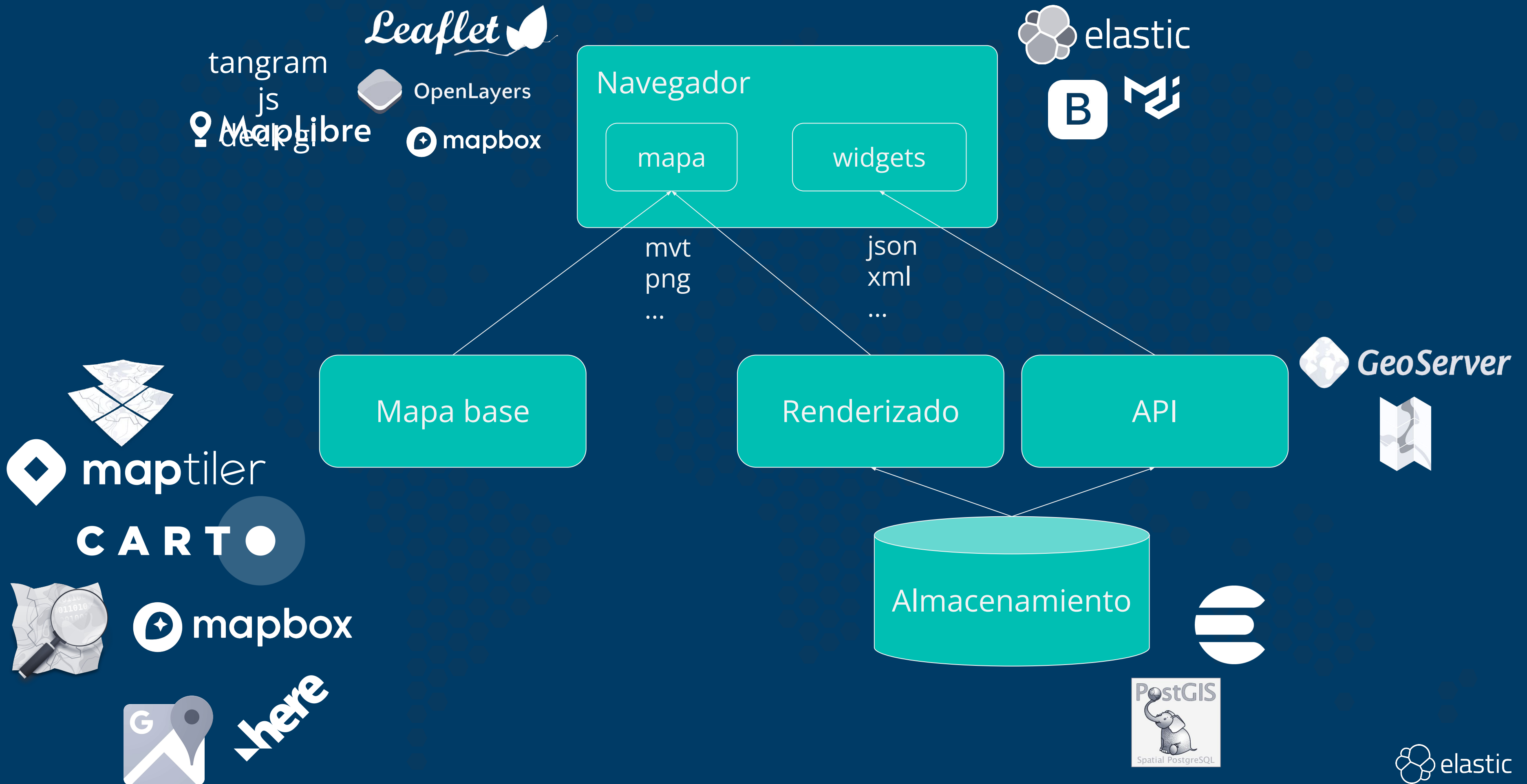
With the `geo_distance` query type get the positions near Barajas airport:

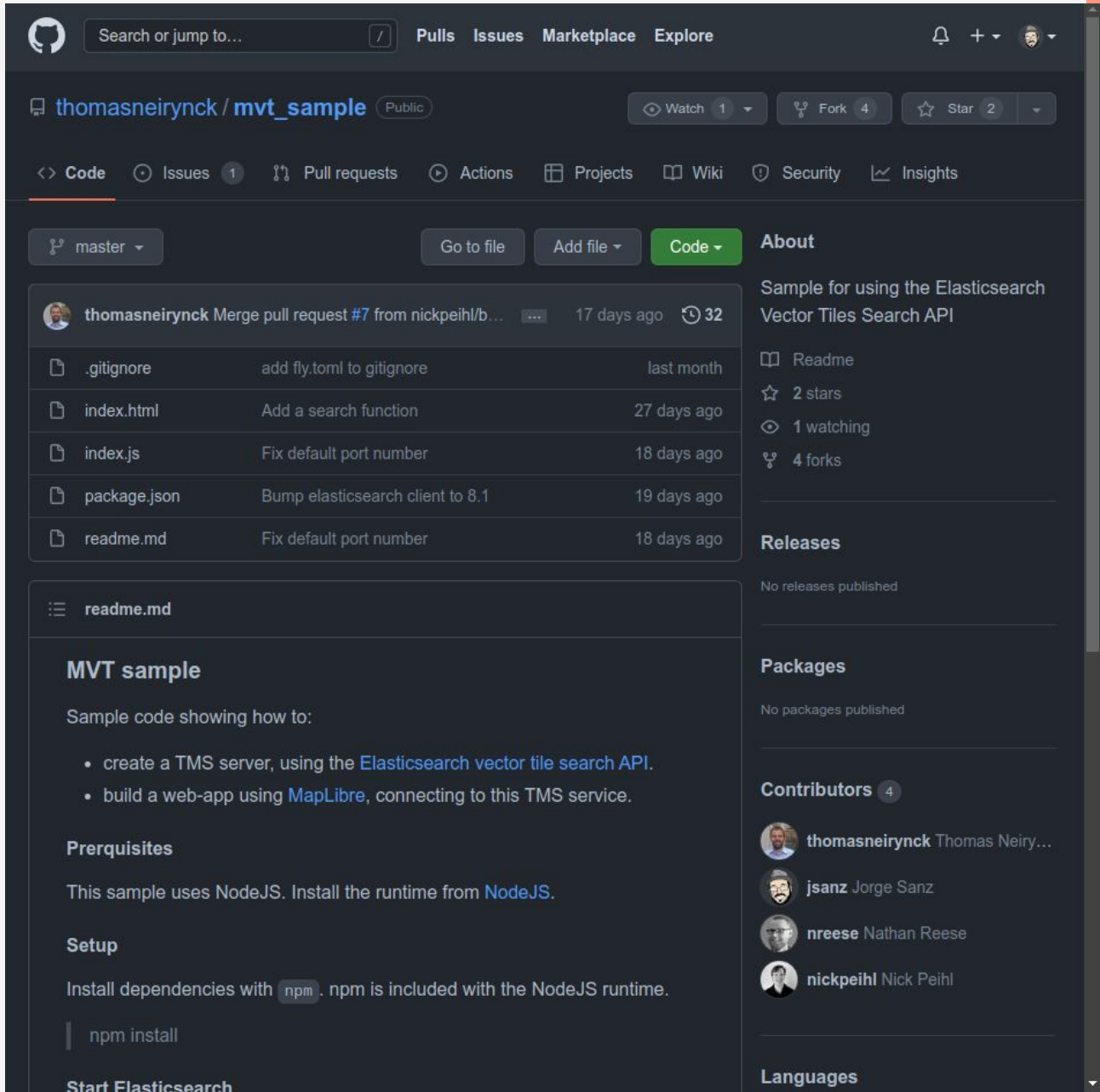
```
GET flight_tracking*/_search
{
  "query": {
    "geo_distance": {
      "distance": "5km",
      "location": {
        "lat": 40.469674,
        "lon": -3.559828
      }
    }
  }
}
```

Agenda

- 1 Introducción al stack de Elastic
- 2 Preparación del laboratorio
- 3 Introducción a Kibana
- 4 Elastics Maps
- 5 Elasticsearch Geo
- 6 Web mapping con Elasticsearch

Pequeña intro al *webmapping*

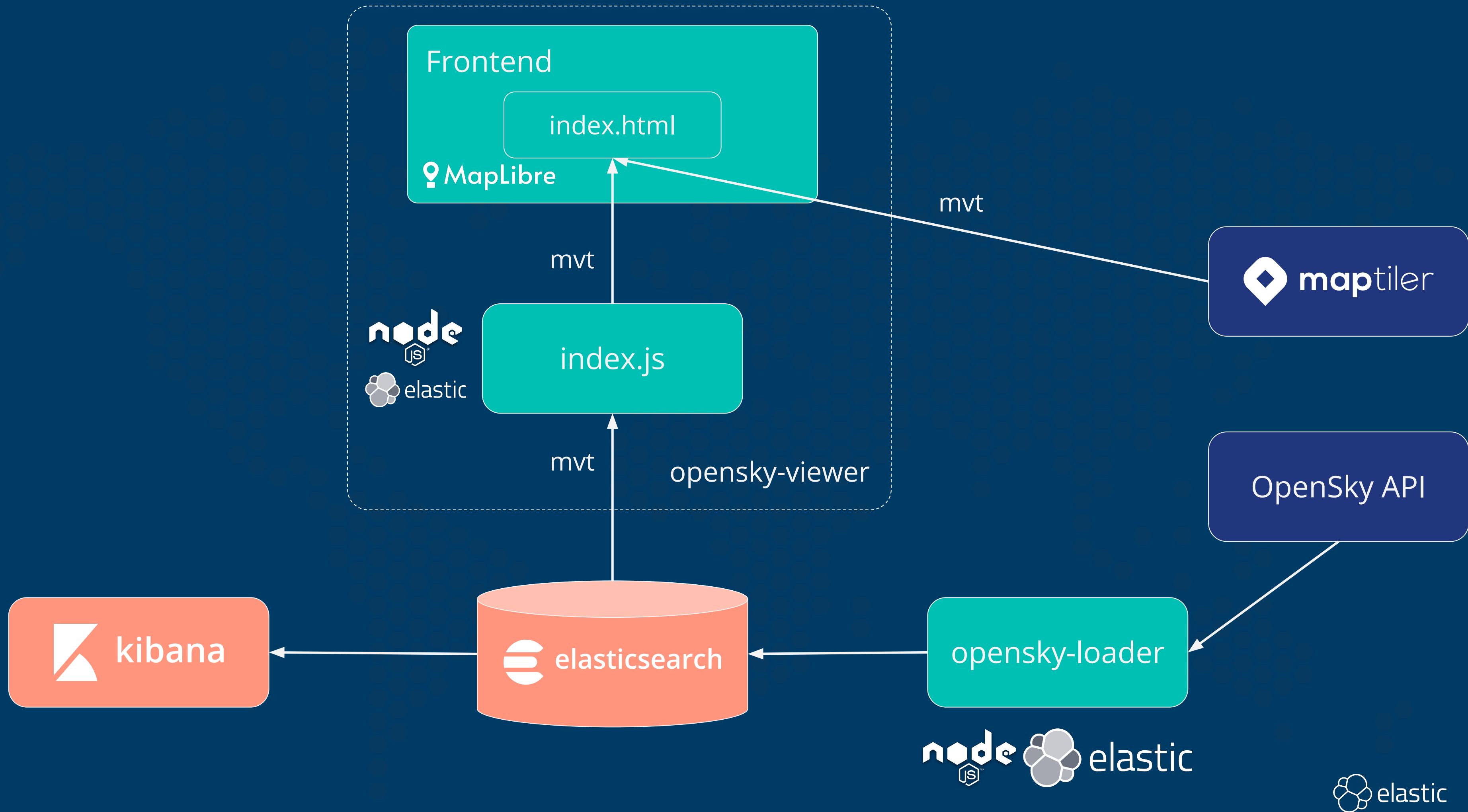




- Un servidor de teselas: nodejs
- Un visor genérico: Maplibre

https://github.com/thomasneiryck/mvt_sample





Hora de practicar: cargador de datos

- </lab/opensky-loader/index.js>
- Aplicación nodejs con el cliente de elasticsearch
- Revisar el flujo de carga de datos
- Cómo se crea el índice
- Cómo se formatean los datos antes de subirlos
- Cómo se cargan los datos usando la **Bulk API**

Hora de practicar: visualizador

https://github.com/thomasneiryneck/mvt_sample

Backend

- </lab/opensky-viewer/index.js>
- La ruta raíz sirve el *frontend*
- Controlador para la ruta </tile>
- Revisar parámetros
- Construcción de la query a Elasticsearch
- Cabeceras adicionales

Frontend

- </lab/opensky-viewer/index.html>
- Maplibre y un formulario sencillo
- Capas para polígonos, líneas y puntos
- Revisar *source*: el servidor del *backend*
- Revisar estilos
- Revisar *callback* para el contador de geometrías

¡Gracias!

Jorge Sanz

2022-06-09, Kibana, Elastic,

 xurxosanz  jsanz  jorge.sanz@elastic.co

<https://ela.st/siglibre2022-taller>