Elasticsearch & Kibana Workshop

FOSS4G Europe, July, 2025

Mostar, Bosnia-Herzegovina

https://ela.st/2025-foss4ge-workshop

elastic | The Search AI Company

FOSS4G EUROPE MOSTAR 2025

Jorge Sanz

Principal Software Engineer
Kibana Presentation and Maps team
jorge.sanz@elastic.co

Craig Taverner

Principal Software Engineer
Elasticsearch Analytics & Geo team
craig.taverner@elastic.co

Agenda

Elastic intro & Elasticsearch and geospatial (~30min)

ES|QL (~90min)

- Source Commands
- Processing Commands: filters, aggregations, calculations
- Geospatial functions

Kibana analytics (~2h)

- Kibana intro
- Discover
- Dashboards
- Lens & ES|QL visualizations
- Maps

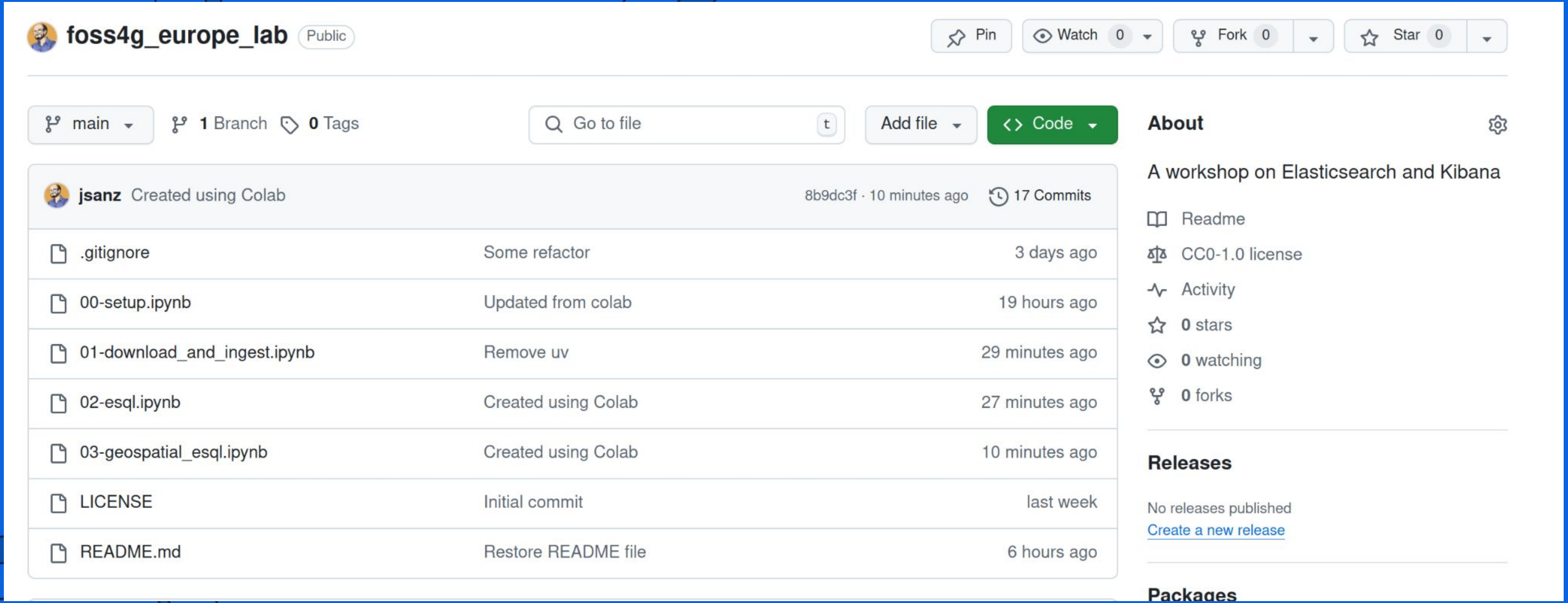
Wednesday at 14:30 at CA01
(TomTom)

“Geospatial ES|QL in Elasticsearch”

with Craig Taverner

details

elastic



jsanz Created using Colab 8b9dc3f · 10 minutes ago 17 Commits

.gitignore	Some refactor	3 days ago
00-setup.ipynb	Updated from colab	19 hours ago
01-download_and_ingest.ipynb	Remove uv	29 minutes ago
02-esql.ipynb	Created using Colab	27 minutes ago
03-geospatial_esql.ipynb	Created using Colab	10 minutes ago
LICENSE	Initial commit	last week
README.md	Restore README file	6 hours ago

About

A workshop on Elasticsearch and Kibana

Readme

CC0-1.0 license

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

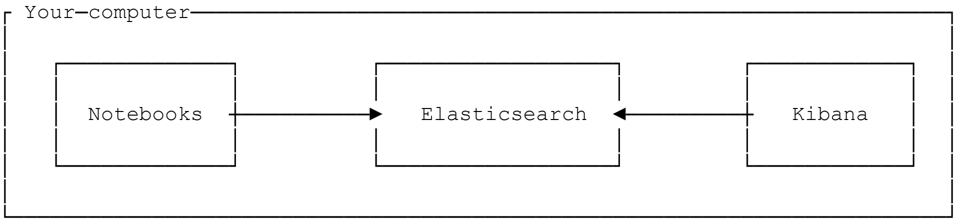
Packages

We'll go through the ES|QL basics and geospatial features using Jupyter notebooks then we'll move to Kibana

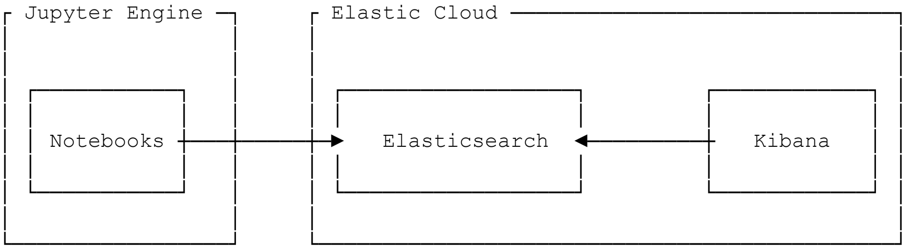
Lab setup

Depending on our connectivity and your preferences

Deploy locally the Notebooks and the Elastic Stack with the `start-local` script



Open the Notebook anywhere and connect to a provided Elastic Stack



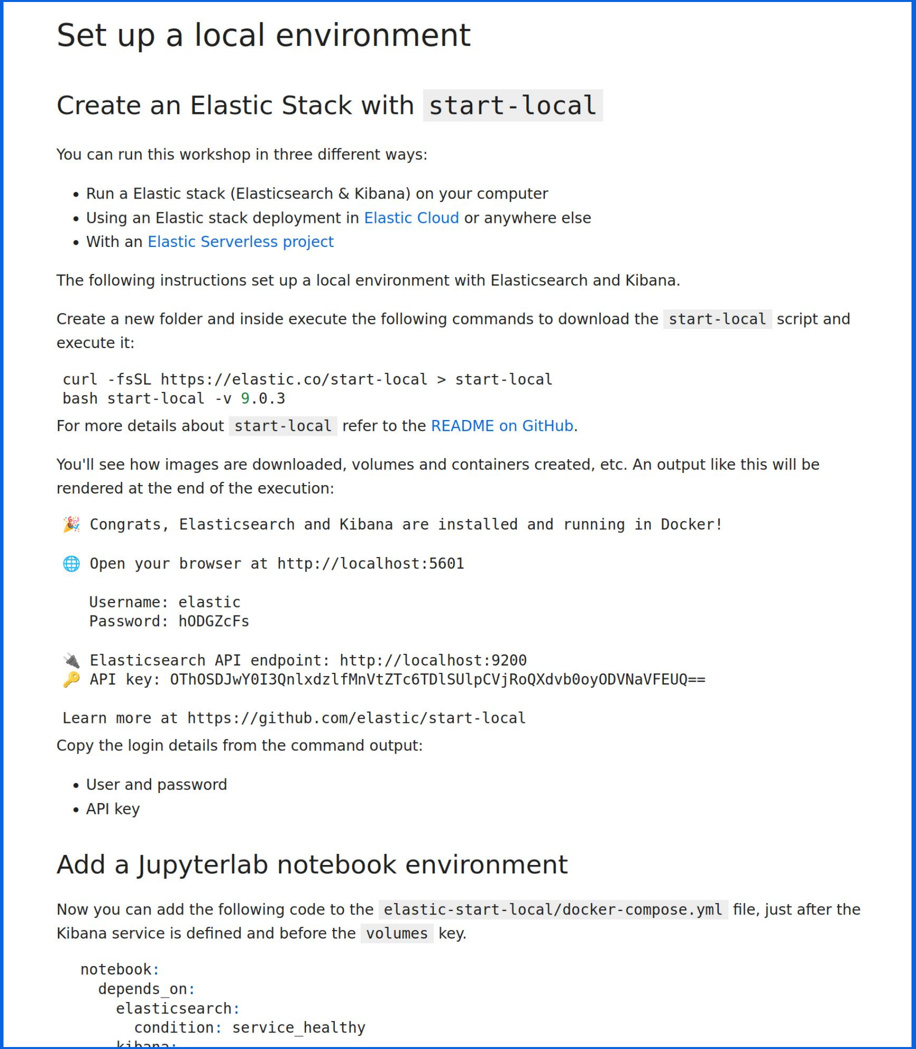
`00-setup.ipynb`

How to download and start an Elastic Stack along with a Jupyter notebook engine.

- Requires a good connectivity to download all the docker images
- Once installed, everything runs in `localhost`
- Fast ingest and download from Elasticsearch
- By default in a *trial* but with instructions to opt out

Alternatively, we provide an Elastic stack cluster for this workshop so you don't need to install anything (now).

- Same features as the local instance (Open Source = Basic license)
- Notebooks can run from any Jupyter engine: locally, Google Colab, Binder, etc.



Set up a local environment

Create an Elastic Stack with `start-local`

You can run this workshop in three different ways:

- Run a Elastic stack (Elasticsearch & Kibana) on your computer
- Using an Elastic stack deployment in Elastic Cloud or anywhere else
- With an Elastic Serverless project

The following instructions set up a local environment with Elasticsearch and Kibana.

Create a new folder and inside execute the following commands to download the `start-local` script and execute it:

```
curl -fsSL https://elastic.co/start-local > start-local
bash start-local -v 9.0.3
```

For more details about `start-local` refer to the README on GitHub.

You'll see how images are downloaded, volumes and containers created, etc. An output like this will be rendered at the end of the execution:

```
🎉 Congrats, Elasticsearch and Kibana are installed and running in Docker!

🌐 Open your browser at http://localhost:5601

   Username: elastic
   Password: hODGZcFs

🔌 Elasticsearch API endpoint: http://localhost:9200
🔑 API key: OThOSDJwY0I3QnlxdzlfMnVtZTc6TDlSUlpCVjRoQXdvb0oyODVNaVFEUQ==

Learn more at https://github.com/elastic/start-local
```

Copy the login details from the command output:

- User and password
- API key

Add a Jupyterlab notebook environment

Now you can add the following code to the `elastic-start-local/docker-compose.yml` file, just after the Kibana service is defined and before the `volumes` key.

```
  notebook:
    depends_on:
      elasticsearch:
        condition: service_healthy
      kibana:
```

Elastic intro

Elastic envisions a world where everyone can unlock new possibilities by harnessing the power of unlimited data.

Elastic — The Search AI Company

Elastic helps everyone transform data into **answers**, **actions**, and **outcomes** with Search AI.

Founded in
2012

3,000+
employees

40+
Countries with employees

5B+
downloads

Used by over **54%**
of the Fortune 500

Publicly traded under
ESTC in the NYSE

Data as of Mar 2025

Used by more than 50% of the Fortune 500 enterprises

TECHNOLOGY	FINANCE	TELCO	CONSUMER	HEALTHCARE	PUBLIC SECTOR	AUTOMOTIVE / TRANSPORTATION	RETAIL
Adobe	BARCLAYS	orange	Uber	VITAS Healthcare	Lawrence Livermore National Laboratory	VOLVO Volvo Group	AutoZone
cisco	ZURICH	dish media	Grab	UCLA Health	OAK RIDGE National Laboratory	Audi	THE HOME DEPOT
workday.	USAA	COMCAST	Miles & More Lufthansa	Yale NewHaven Health	De Watergroep WATER. VANDAAG EN MORGEN.	JAGUAR LAND ROVER	ebay
Microsoft	Swift	verizon	ACTIVISION BLIZZARD	MAYO CLINIC	JPL Jet Propulsion Laboratory	BMW	Kroger
INGRAM MICRO	Postbank	T Mobile	lyft	Pfizer	MENTAT COMPUTE OPTIMIZATION		Walgreens

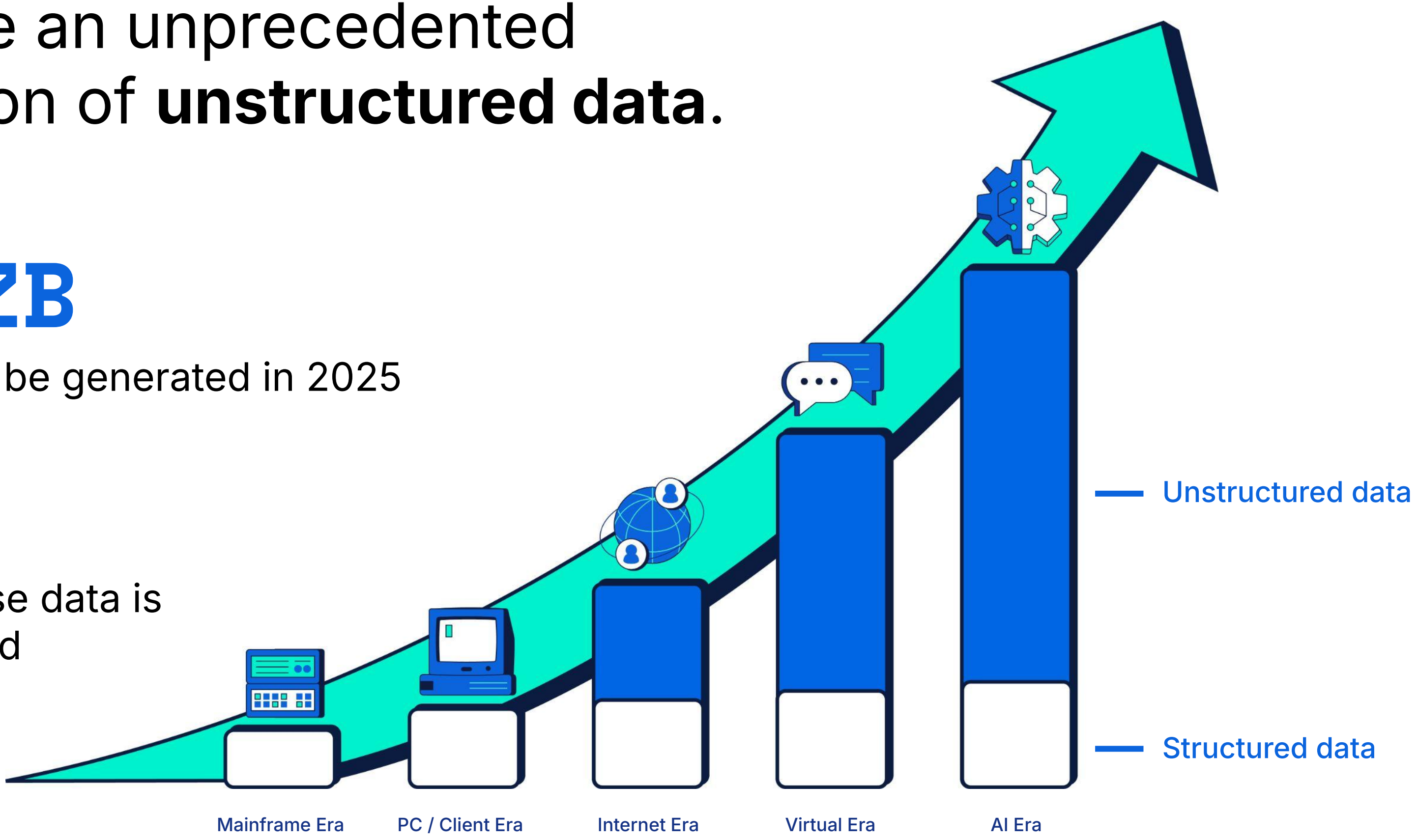
We face an unprecedented explosion of **unstructured data.**

175 ZB

Of data will be generated in 2025

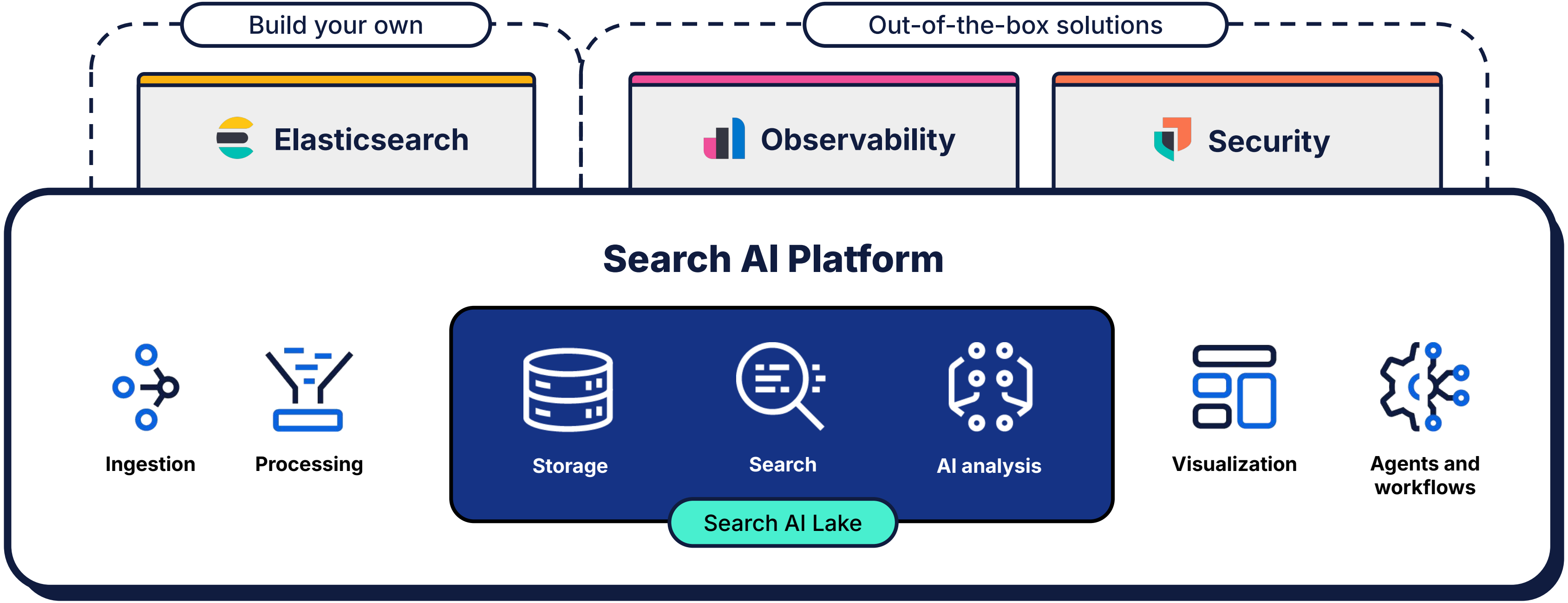
90%

Of enterprise data is unstructured



Source: IDC

One **platform**, two out-of-the-box **solutions**, the freedom to **build anything**

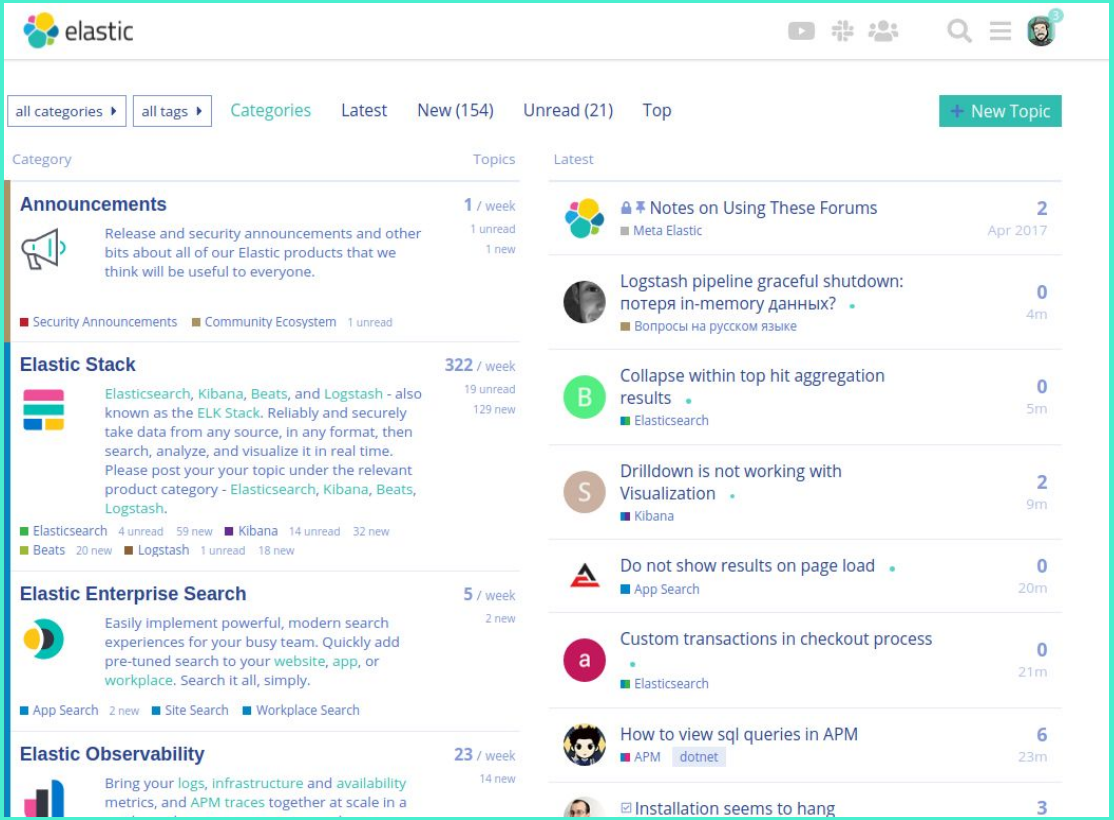


Community

https://github.com/elastic

https://ela.st/slack

https://discuss.elastic.co



elastic

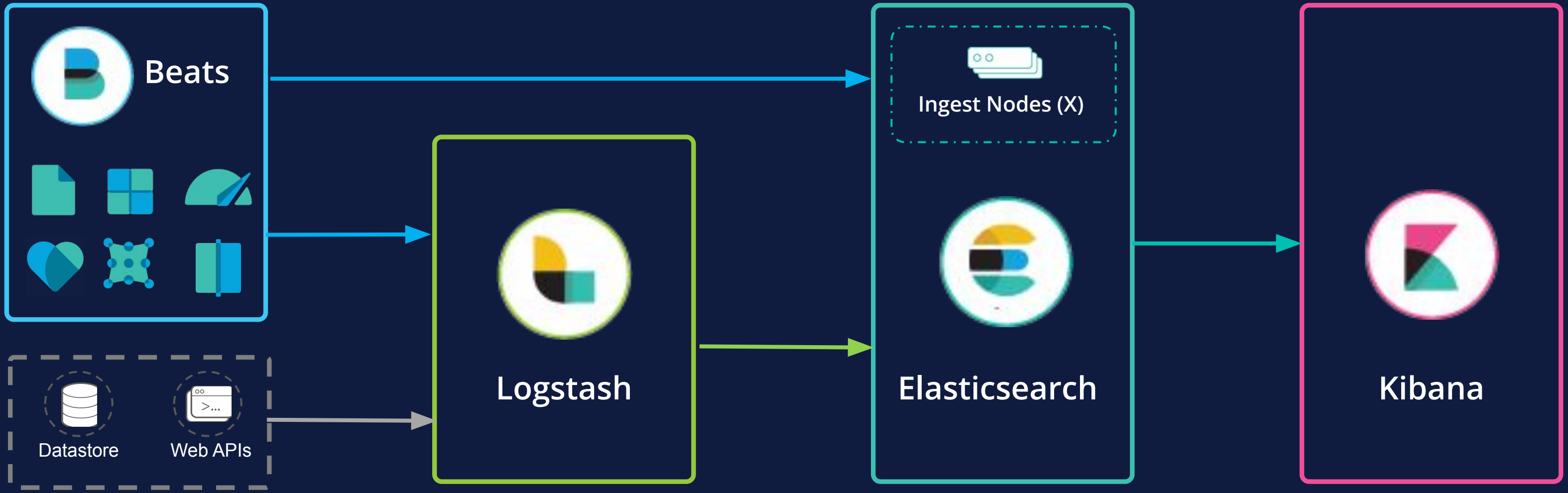
all categories ▸ all tags ▸ Categories Latest New (154) Unread (21) Top + New Topic

Category	Topics
Announcements Release and security announcements and other bits about all of our Elastic products that we think will be useful to everyone. Security Announcements · Community Ecosystem 1 unread	1 / week 1 unread 1 new
Elastic Stack Elasticsearch, Kibana, Beats, and Logstash - also known as the ELK Stack. Reliably and securely take data from any source, in any format, then search, analyze, and visualize it in real time. Please post your your topic under the relevant product category - Elasticsearch, Kibana, Beats, Logstash. Elasticsearch 4 unread 59 new · Kibana 14 unread 32 new · Beats 20 new · Logstash 1 unread 18 new	322 / week 19 unread 129 new
Elastic Enterprise Search Easily implement powerful, modern search experiences for your busy team. Quickly add pre-tuned search to your website, app, or workplace. Search it all, simply. App Search 2 new · Site Search · Workplace Search	5 / week 2 new
Elastic Observability Bring your logs, infrastructure and availability metrics, and APM traces together at scale in a	23 / week 14 new

Latest		
Notes on Using These Forums Meta Elastic	2	Apr 2017
Logstash pipeline graceful shutdown: потеря in-memory данных? Вопросы на русском языке	0	4m
Collapse within top hit aggregation results Elasticsearch	0	5m
Drilldown is not working with Visualization Kibana	2	9m
Do not show results on page load App Search	0	20m
Custom transactions in checkout process Elasticsearch	0	21m
How to view sql queries in APM APM dotnet	6	23m
☑ Installation seems to hang	3	

Elastic Stack

Ingest, Store, Search, Visualise



Elasticsearch

Open by design

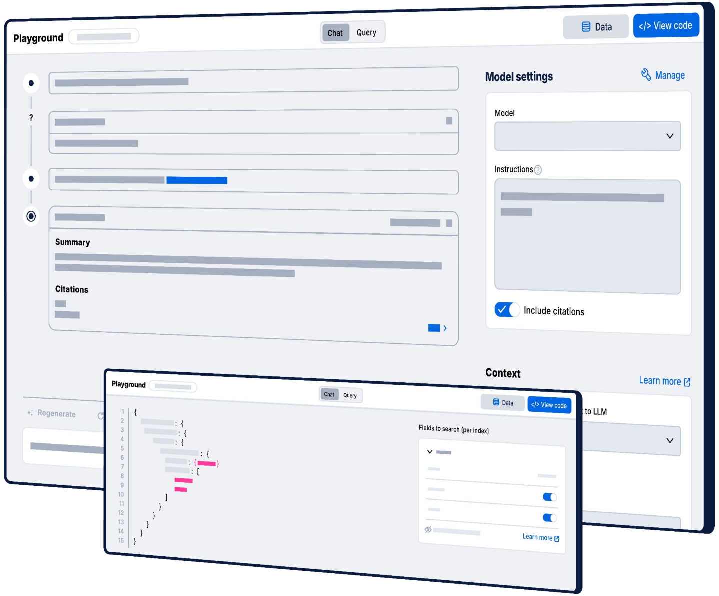
Data comes in all shapes and sizes, and applications and services operate between endpoints and the cloud. Builders win when their tools prioritize flexibility, transparency, and interoperability.

Built for performance

Data volumes are unprecedented, and customer expectations have never been higher. Builders require tools that are able to instantly deliver relevant results at scale.

Wired for innovation

The data landscape is always evolving with new formats, sources, and regulations. Builders need tools that have a comprehensive set of capabilities but never stop pushing the boundaries of what's possible.



Elasticsearch and geospatial

Geospatial timeline

2010-2012

Geo search
Points & Shapes

2014

Geo analysis
Aggregations

2017-2019

Geo at scale
Encoding
upgrades

2020-2021

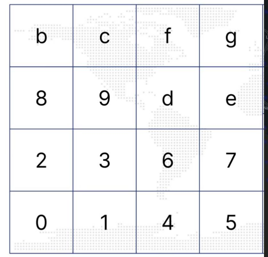
GIS
Maps & analytics

2022-2023

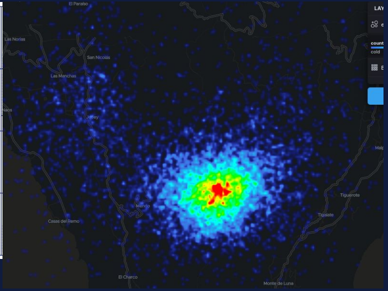
Vector tiles and H3
Fast aggregations
and rendering

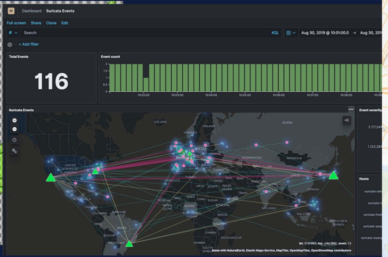
2024

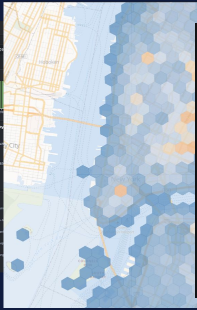
ES|QL
Powerful and
expressive language

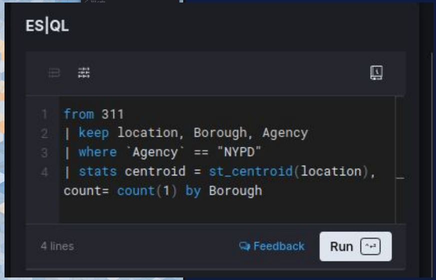


b	c	f	g
8	9	d	e
2	3	6	7
0	1	4	5









ES|QL

```
from 311
| keep location, Borough, Agency
| where `Agency` == "NYPD"
| stats centroid = st_centroid(location),
count= count(1) by Borough
```

elastic

Elasticsearch geospatial data types

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

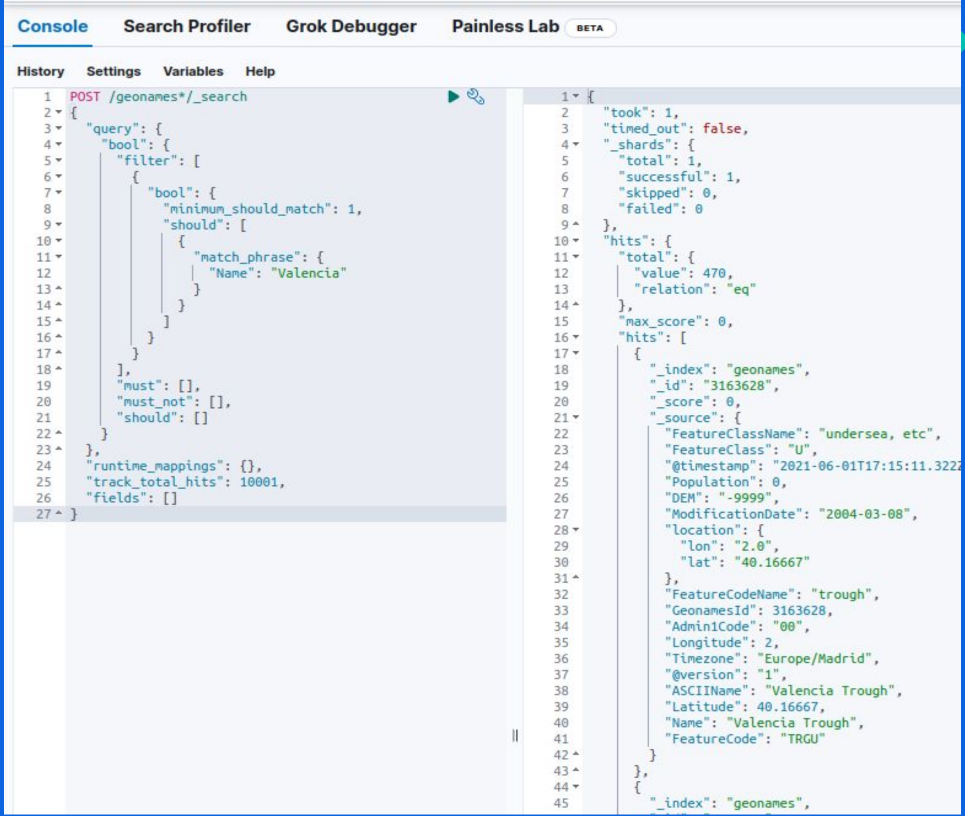
Vector tiles API

Elasticsearch `_search` API

- JSON output format
- Search and aggregate

Elasticsearch `_mvt` API

- *protobuf* output format
- Use queries and aggregations to generate standard vector tiles



```
POST /geonames*/_search
{
  "query": {
    "bool": {
      "filter": [
        {
          "bool": {
            "minimum_should_match": 1,
            "should": [
              {
                "match_phrase": {
                  "Name": "Valencia"
                }
              }
            ]
          }
        }
      ],
      "must": [],
      "must_not": [],
      "should": []
    }
  },
  "runtime_mappings": {},
  "track_total_hits": 10001,
  "fields": []
}
```

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 470,
      "relation": "eq"
    },
    "max_score": 0,
    "hits": [
      {
        "_index": "geonames",
        "_id": "3163628",
        "_score": 0,
        "_source": {
          "FeatureClassName": "undersea, etc",
          "FeatureClass": "U",
          "@timestamp": "2021-06-01T17:15:11.322Z
          "Population": 0,
          "DEM": "-9999",
          "ModificationDate": "2004-03-08",
          "location": {
            "lon": "2.0",
            "lat": "40.16667"
          },
          "FeatureCodeName": "trough",
          "GeonamesId": 3163628,
          "Admin1Code": "00",
          "Longitude": 2,
          "Timezone": "Europe/Madrid",
          "@version": "1",
          "ASCIIName": "Valencia Trough",
          "Latitude": 40.16667,
          "Name": "Valencia Trough",
          "FeatureCode": "TRGU"
        }
      },
      {
        "_index": "geonames",
```

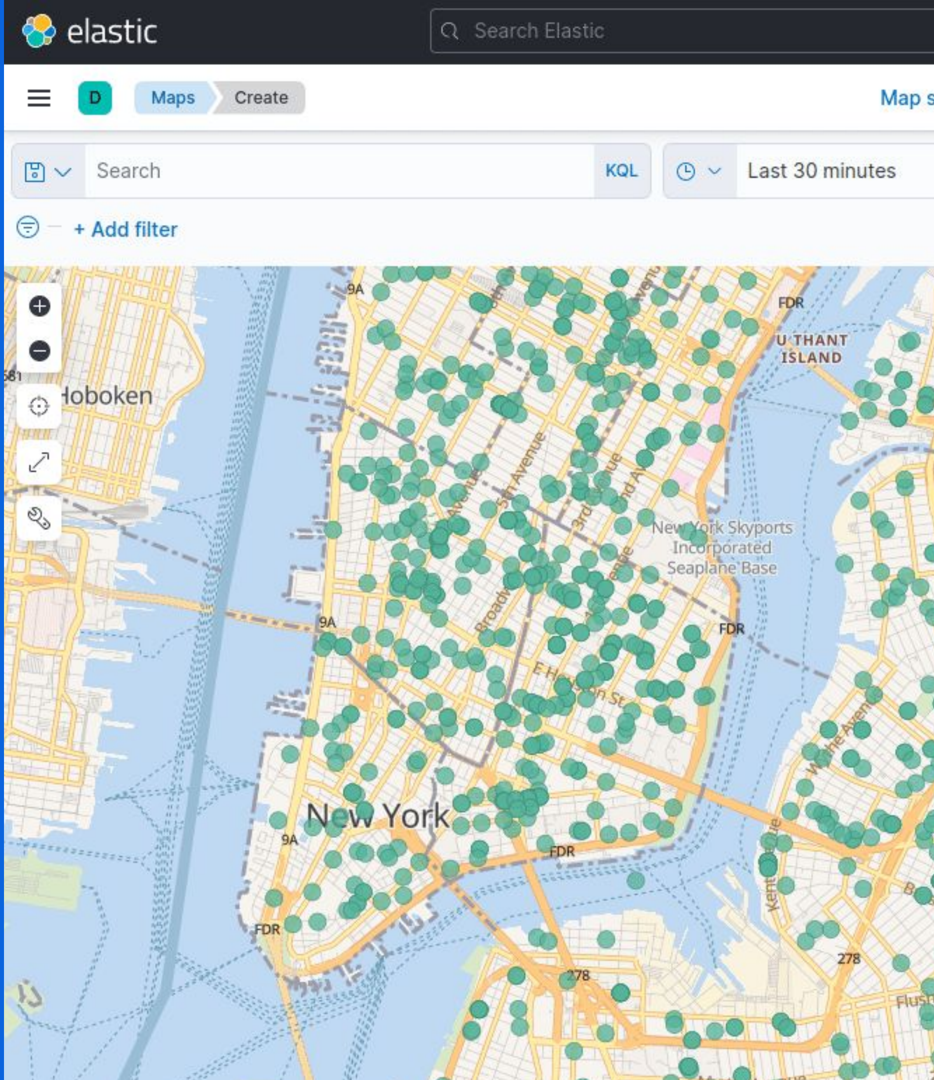
Search

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

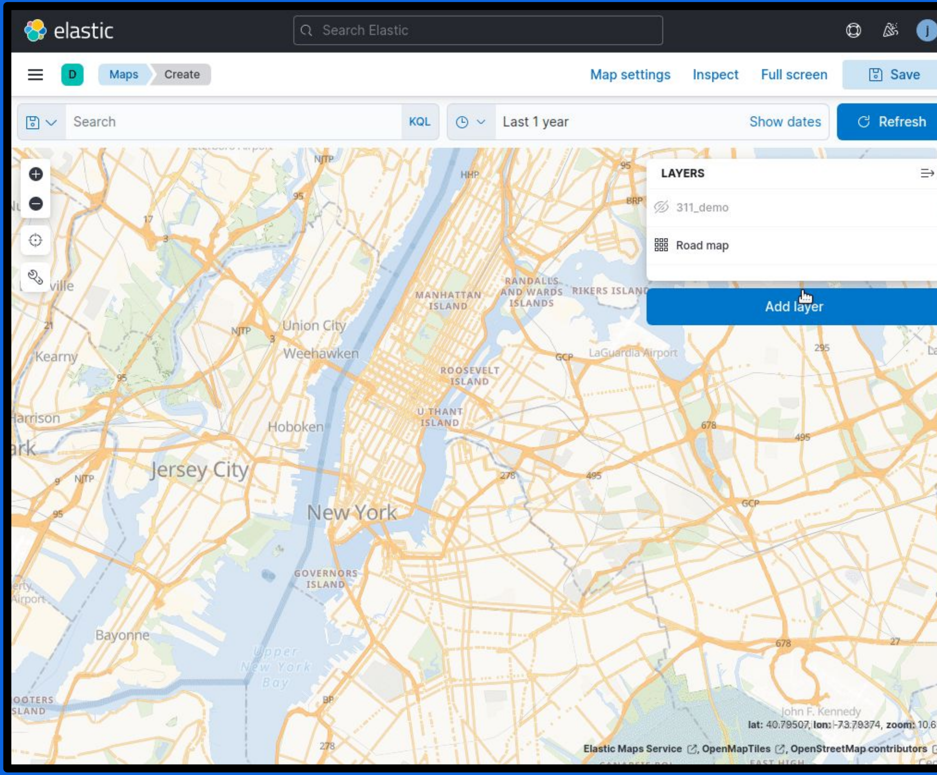
Binning (bucket agg)

- Geo-Distance (rings) 📖
- Geohash grid 📖
- Geotile grid 📖
- Geohex grid 📖 🛒

Derived geometries (metric agg)

- Geo-centroid 📖
- Geo-bounds 📖
- Geo-line 📖 🛒

Non-geo aggregations: Huge range of bucket and metric aggregations 📖



Introducing ES|QL

What is ES|QL?

What is ES|QL?

Declarative

Piped

Tabular

Distributed

Vectorized

```
FROM airports
| EVAL distance = ST_DISTANCE(
                    location,
                    TO_GEOPOINT("POINT(12.565 55.673)"))
| WHERE distance < 1000000
    AND scalerank < 6
    AND distance > 10000
| SORT distance ASC
| KEEP distance, abbrev, name, location, country, city
```

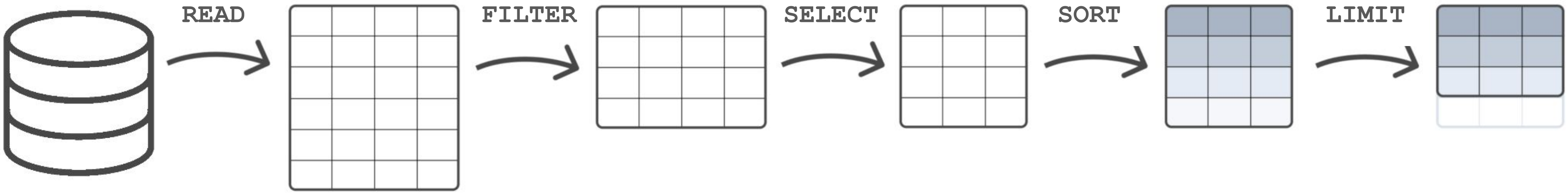
Declarative, Piped, Tabular

```
1  FROM airports
2  | WHERE scalerank < 6
3  | KEEP abbrev, name, location, country, city
4  | SORT abbrev ASC
5  | LIMIT 3
```

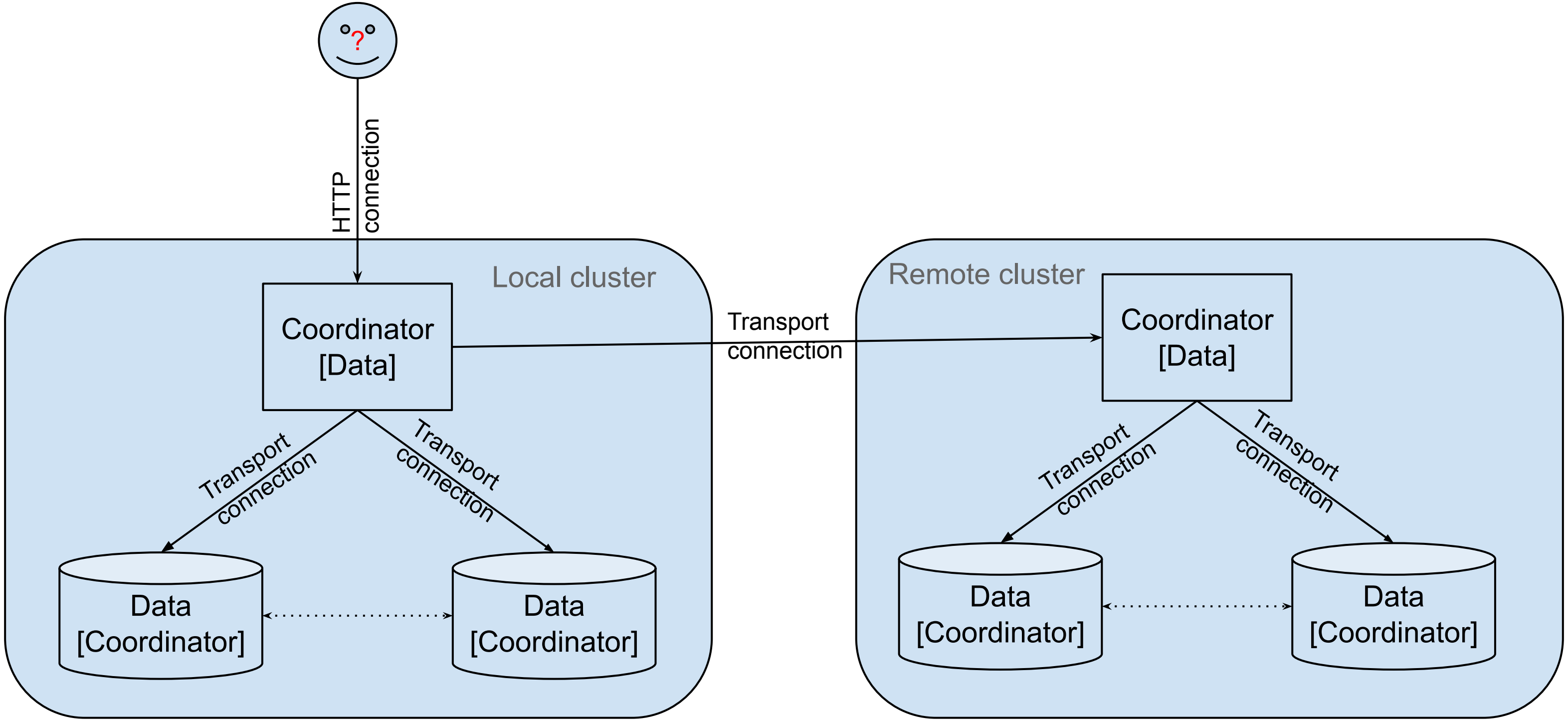
ES|QL

```
3  SELECT abbrev, name, location, country, city
1  FROM airports
2  WHERE scalerank < 6
4  ORDER BY abbrev ASC
5  LIMIT 3
```

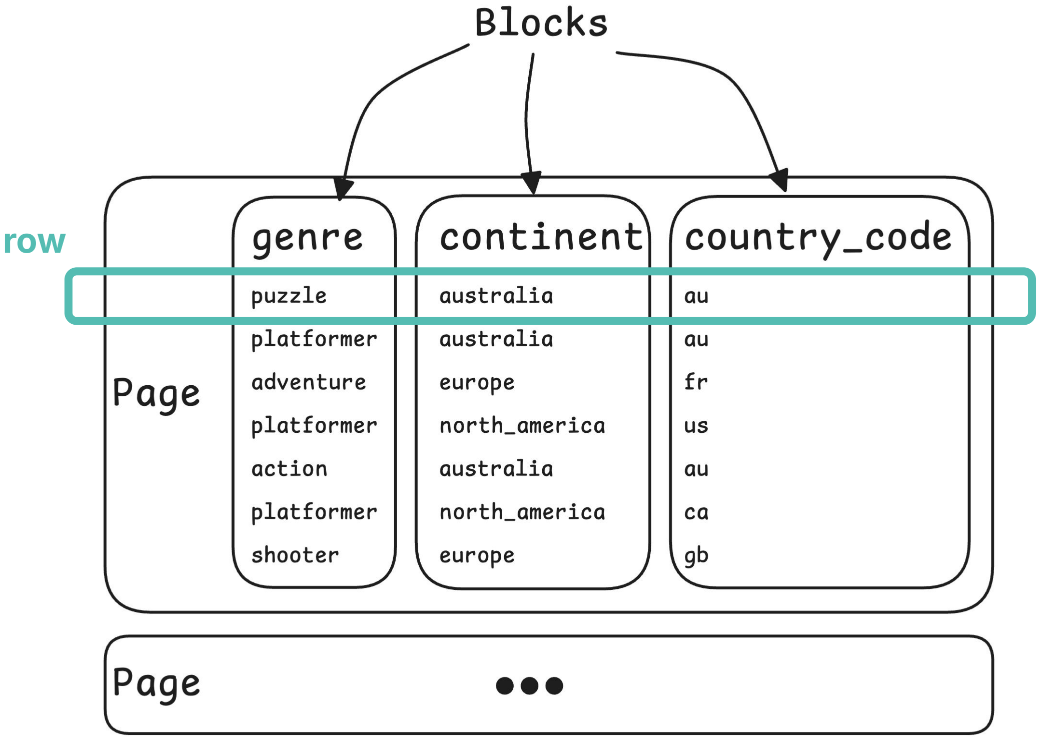
SQL



Distributed



Vectorized



genre	continent	country_code
puzzle	australia	au
platformer	australia	au
adventure	europe	fr
platformer	north_america	us
action	australia	au
platformer	north_america	ca
shooter	europe	gb

How many languages are there in Elastic?

9 Languages of Elastic

Existing challenging

Think about the challenges that you have with the following:

- Query DSL
- Runtime fields
- Mapping (schema definition)
- Aggregation, sub-aggregation
- …

ES|QL

Elasticsearch Query Language (ES|QL) provides a powerful way to **filter, transform, and analyze** data stored in Elasticsearch.

It is designed to be **easy to learn** and use, by end users, SRE teams, application developers, subject matter experts, and administrators.

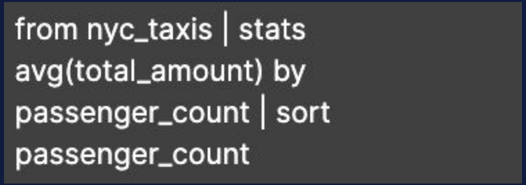
Keywords: speed, simplicity, and efficiency

Distributed & Dedicated Query Engine

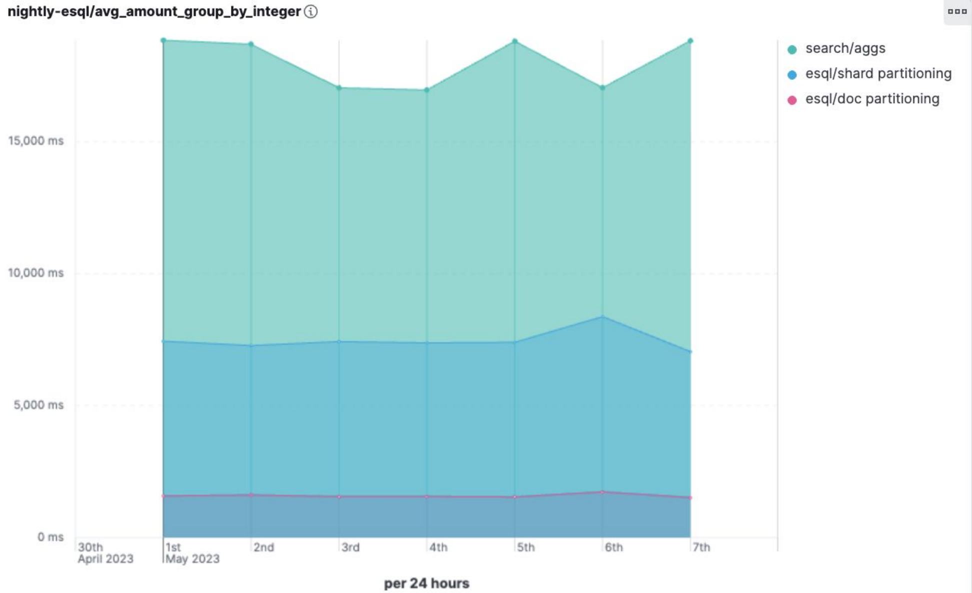
`_query`

- No transpilation or translation
- Queries are parsed and optimized for distributed execution
- It operates in blocks, instead of one row at a time
- It takes advantage of specialization and multi-threading
- Benchmarking has shown ES|QL can outperform DSL in many instances

ESQL Performance Status



```
from nyc_taxis | stats
avg(total_amount) by
passenger_count | sort
passenger_count
```



nightly-esql/avg_amount_group_by_integer

There is a performance dashboard to follow along with performance benchmarking:

Link to dashboard

ESQL is **faster** than Elasticsearch aggregations in some cases, even without many optimizations

Key Benefits:

ES|QL License: Basic

Tech Preview: 8.11-8.13

GA: 8.14+

Greater query speed

Simplify Elasticsearch

New transformative search engine

Key Benefits:

ES|QL License: Basic

Tech Preview: 8.11-8.13

GA: 8.14+

Fast Time to Insights

Reduce the friction of bringing data into Elasticsearch

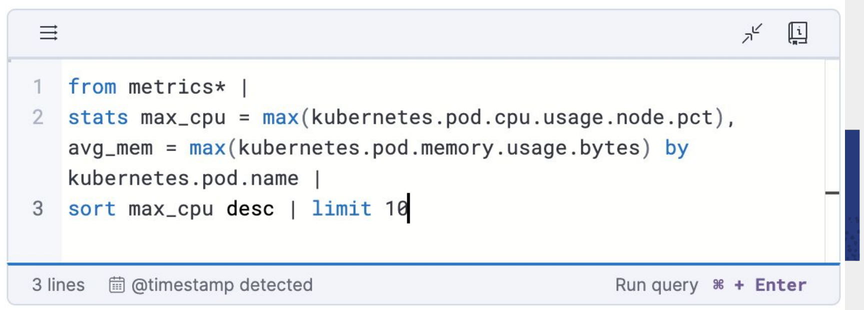
Improved Alerting

Observability

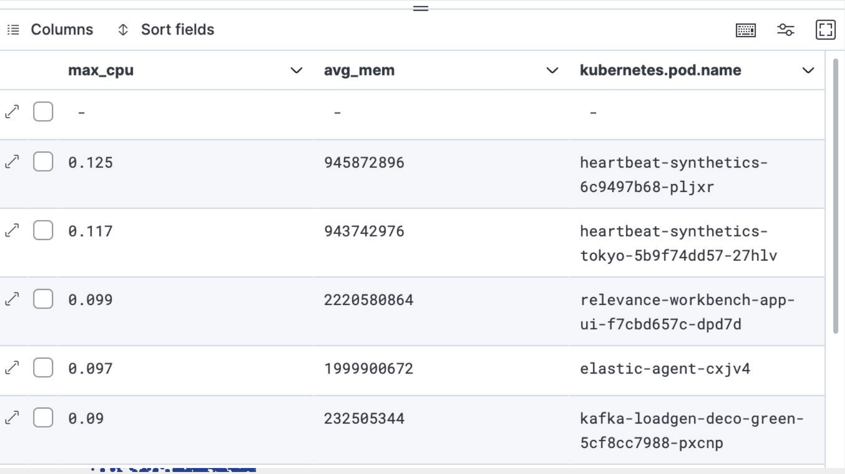
Using ES|QL greatly simplifies the analyzing of metrics, logs, and traces from a single query, quickly identifying performance issues all from a single search box

Define fields on the fly, enrich data with lookups, and concurrent query processing, for speed and efficiency.

Integrating ES|QL with Elastic ML and AiOps improves detection accuracy along with aggregated value thresholds.



```
from metrics* |
stats max_cpu = max(kubernetes.pod.cpu.usage.node.pct),
avg_mem = max(kubernetes.pod.memory.usage.bytes) by
kubernetes.pod.name |
sort max_cpu desc | limit 10
```



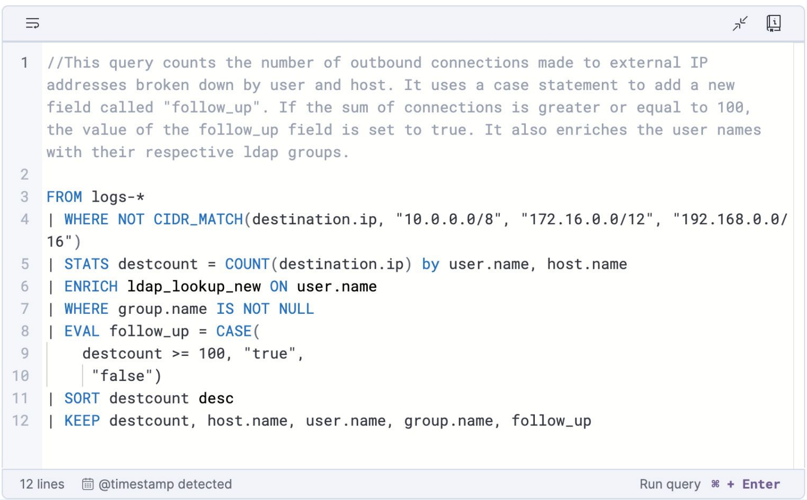
max_cpu	avg_mem	kubernetes.pod.name
-	-	-
0.125	945872896	heartbeat-synthetics-6c9497b68-pljxr
0.117	943742976	heartbeat-synthetics-tokyo-5b9f74dd57-27hlv
0.099	2220580864	relevance-workbench-app-ui-f7cbd657c-dpd7d
0.097	1999900672	elastic-agent-cxjv4
0.09	232505344	kafka-loadgen-deco-green-5cf8cc7988-pxcnp

Security

ES|QL enhances SecOps by streamlining workflows and investigations providing a singular place to find what you are looking for

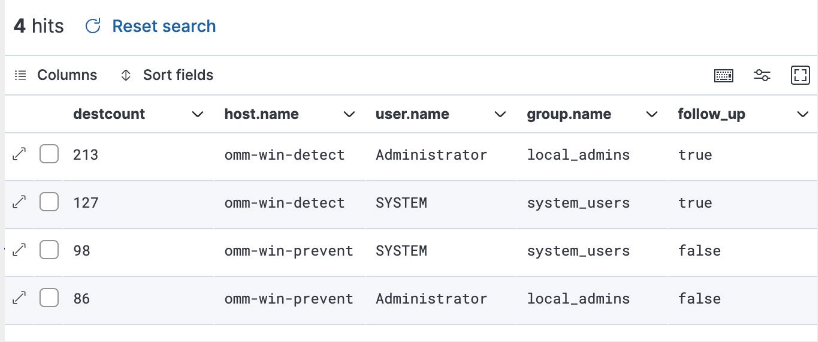
Pull in critical context for investigations with ES|QL lookups. Enrich data and defining fields on the fly for valuable insights for accelerated action

ES|QL reduces alarm fatigue and ensures more accurate alerts by incorporating aggregated values in detection rules



```
//This query counts the number of outbound connections made to external IP addresses broken down by user and host. It uses a case statement to add a new field called "follow_up". If the sum of connections is greater or equal to 100, the value of the follow_up field is set to true. It also enriches the user names with their respective ldap groups.

FROM logs-*
| WHERE NOT CIDR_MATCH(destination.ip, "10.0.0.0/8", "172.16.0.0/12", "192.168.0.0/16")
| STATS destcount = COUNT(destination.ip) by user.name, host.name
| ENRICH ldap_lookup_new ON user.name
| WHERE group.name IS NOT NULL
| EVAL follow_up = CASE(
    destcount >= 100, "true",
     "false")
| SORT destcount desc
| KEEP destcount, host.name, user.name, group.name, follow_up
```



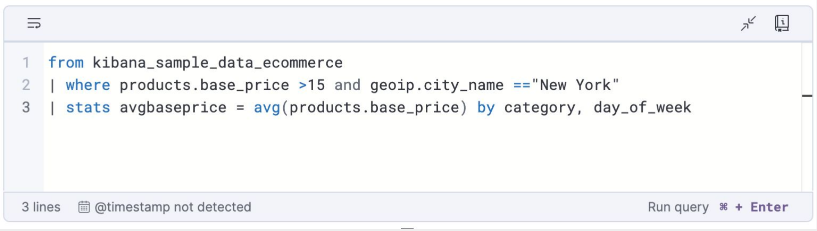
destcount	host.name	user.name	group.name	follow_up
213	omm-win-detect	Administrator	local_admins	true
127	omm-win-detect	SYSTEM	system_users	true
98	omm-win-prevent	SYSTEM	system_users	false
86	omm-win-prevent	Administrator	local_admins	false

Security

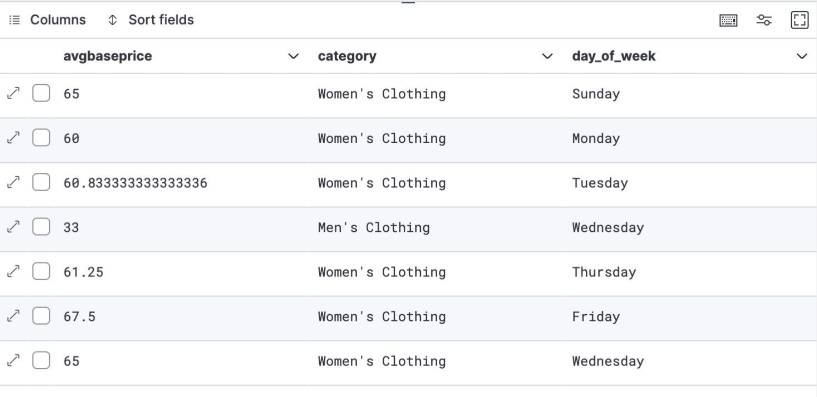
Developers will benefit from a simplified coding and querying experience with ES|QL. Saving time and reducing cost with these efficiencies.

ES|QL delivers a simple way of understanding more about your data. What does it contain, how should I organize it, and how to troubleshoot when issues arise. Saving time and reducing cost.

ES|QL streamlines tasks into one query which can be concurrently processed for even faster performance. Lower TCO, more for less.



```
from kibana_sample_data_ecommerce
| where products.base_price >15 and geoip.city_name =="New York"
| stats avgbaseprice = avg(products.base_price) by category, day_of_week
```

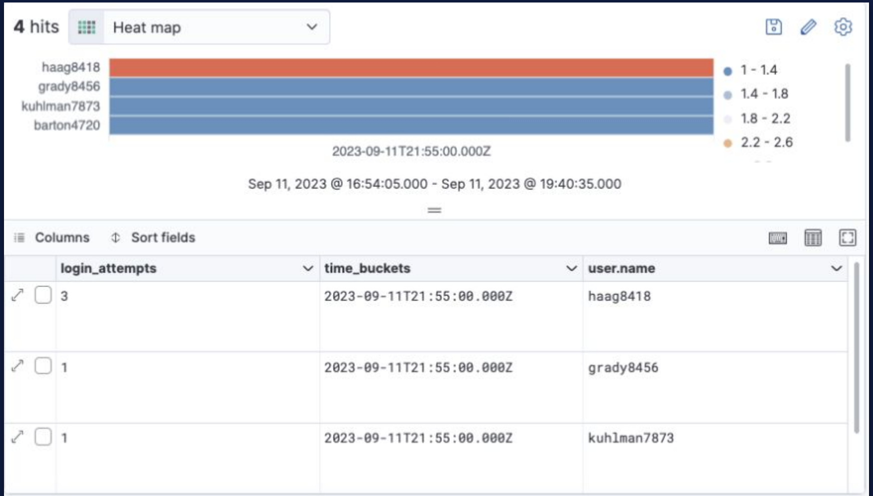


avgbaseprice	category	day_of_week
65	Women's Clothing	Sunday
60	Women's Clothing	Monday
60.833333333333336	Women's Clothing	Tuesday
33	Men's Clothing	Wednesday
61.25	Women's Clothing	Thursday
67.5	Women's Clothing	Friday
65	Women's Clothing	Wednesday

Understanding ES|QL Syntax

An ES|QL query

```
FROM apache-logs
| WHERE url.original == ‘/login’
| EVAL time_buckets = auto_bucket (@timestamp,
50,”2023-09-11T21:54:05.000Z”,”2023-09-12T00:40
:35.000Z”)
| STATS login_attempts = count(user.name) by
time_buckets, user.name
| SORT login_attempts desc
```



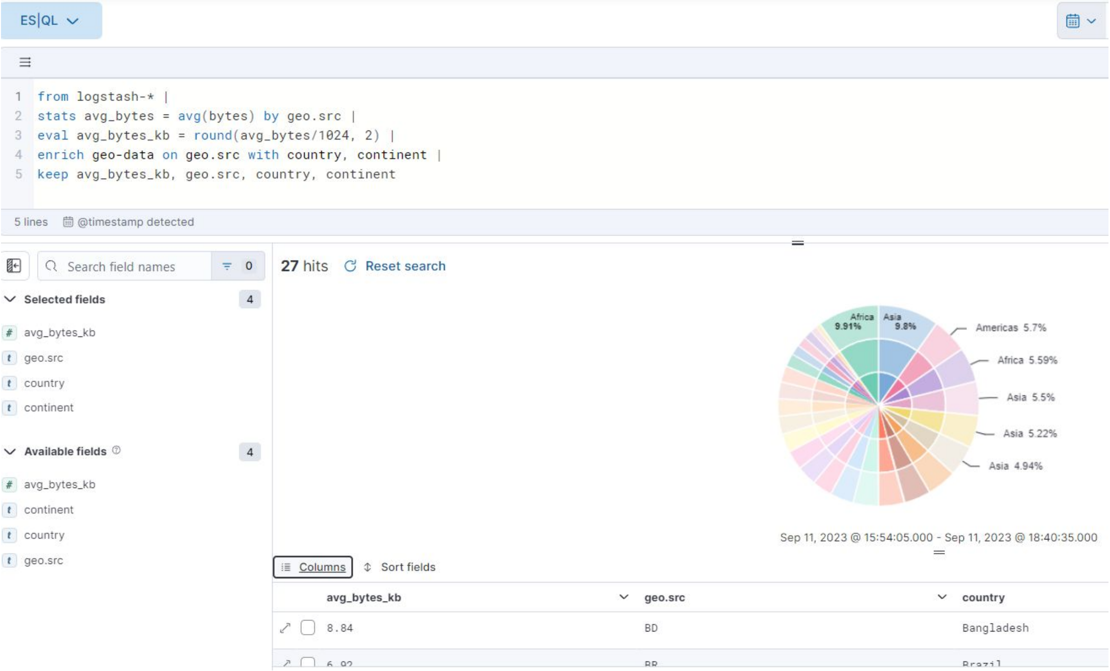
login_attempts	time_buckets	user.name
3	2023-09-11T21:55:00.000Z	haag8418
1	2023-09-11T21:55:00.000Z	grady8456
1	2023-09-11T21:55:00.000Z	kuhlman7873

Expressive, Powerful, Composable, Extensible, Fast

Unified User Experience

ES|QL UX

Data Exploration, Transformation and Visualization all in one



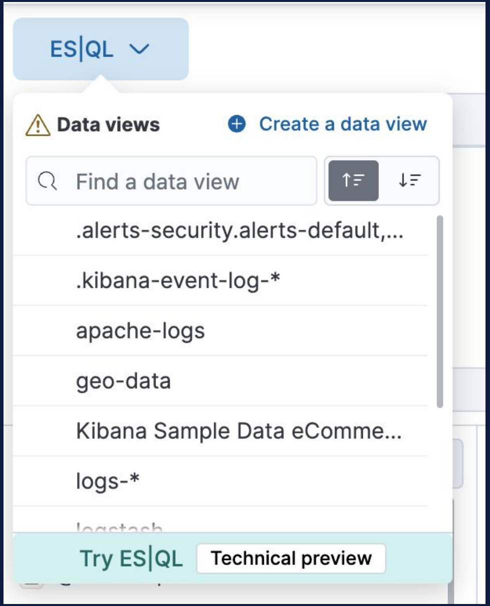
```
from logstash-* |
stats avg_bytes = avg(bytes) by geo.src |
eval avg_bytes_kb = round(avg_bytes/1024, 2) |
enrich geo-data on geo.src with country, continent |
keep avg_bytes_kb, geo.src, country, continent
```

5 lines @timestamp detected

avg_bytes_kb	geo.src	country
8.84	BD	Bangladesh
6.92	BR	Brazil

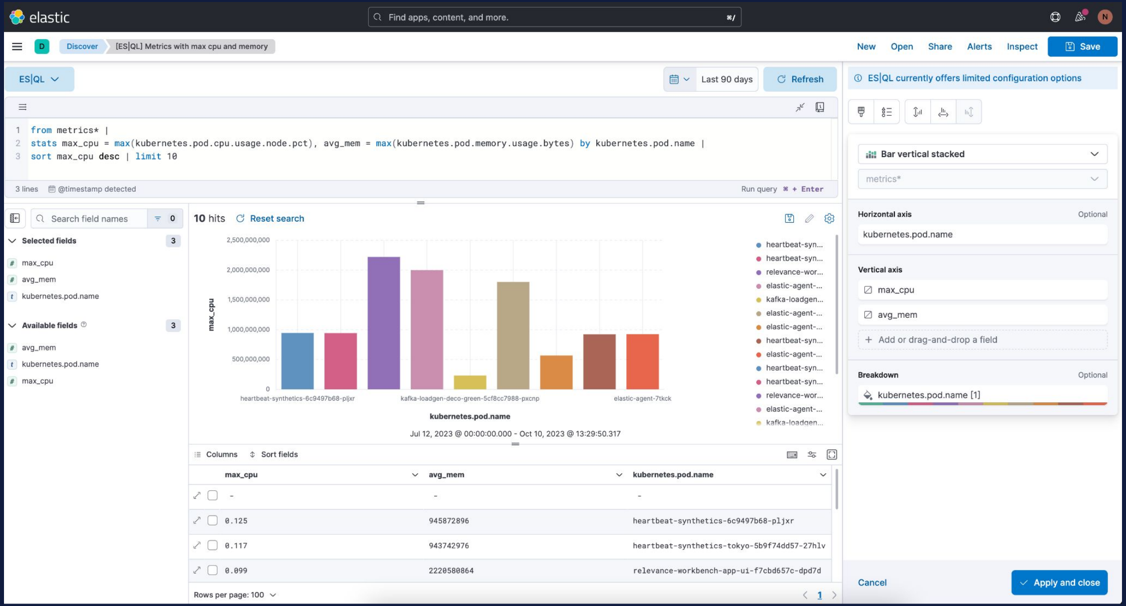
elastic

ES|QL in Discover



- .alerts-security.alerts-default,...
- .kibana-event-log-*
- apache-logs
- geo-data
- Kibana Sample Data eComme...
- logs-*
- logstash

ES|QL is under the data view picker in Discover



```
from metrics* |
stats max_cpu = max(kubernetes.pod.cpu.usage.node.pct), avg_mem = max(kubernetes.pod.memory.usage.bytes) by kubernetes.pod.name |
sort max_cpu desc | limit 10
```

Jul 12, 2023 @ 00:00:00.000 - Oct 10, 2023 @ 13:29:50.317

max_cpu	avg_mem	kubernetes.pod.name
-	-	-
0.125	945872896	heartbeat-synthetics-6c9497b68-pljxr
0.117	943742976	heartbeat-synthetics-tokyo-5b9f74dd57-27hlv
0.099	2220580864	relevance-workbench-app-ui-f7cbd657c-dpd7d

Bar vertical stacked

metrics*

Horizontal axis (Optional): kubernetes.pod.name

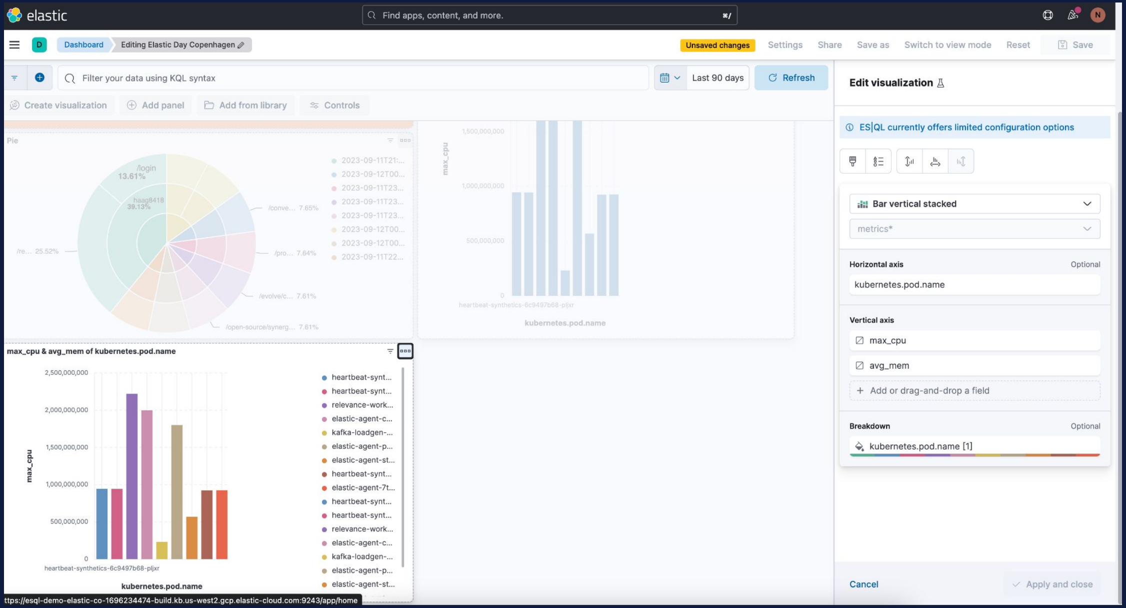
Vertical axis: max_cpu, avg_mem

+ Add or drag-and-drop a field

Breakdown (Optional): kubernetes.pod.name [1]

The ES|QL experience in Discover includes Lens visualizations and in-line editing.

ES|QL in Dashboard



ES|QL currently offers limited configuration options

Bar vertical stacked

metrics*

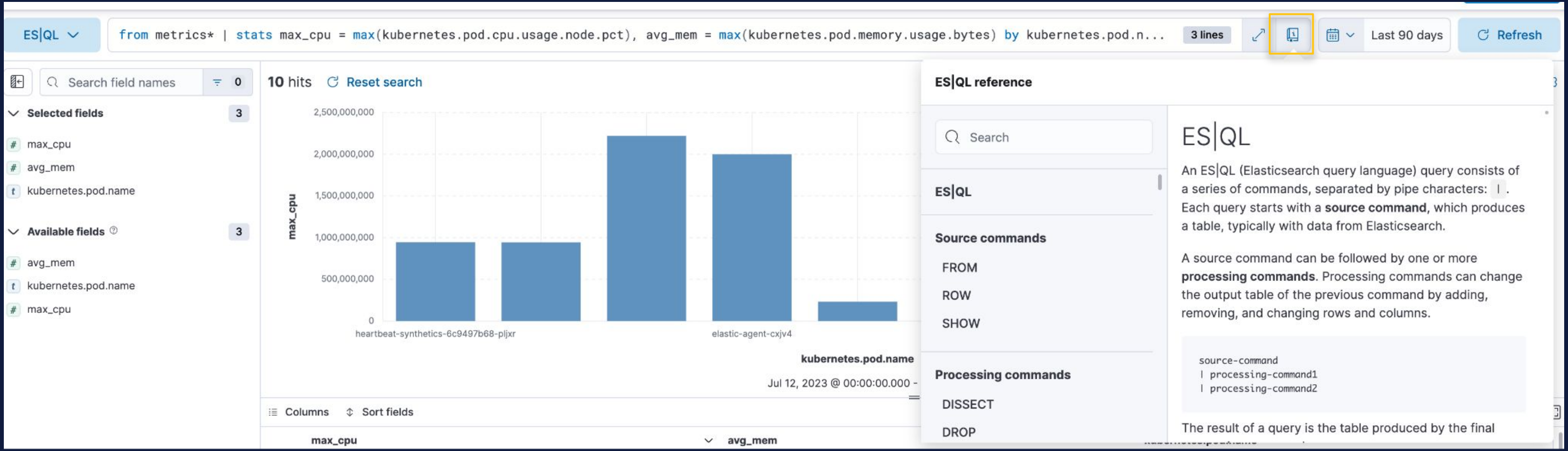
Horizontal axis (Optional): kubernetes.pod.name

Vertical axis: max_cpu, avg_mem

Breakdown (Optional): kubernetes.pod.name [1]

Save ES|QL charts from Discover and use them on Dashboards.
ES|QL charts also have in-line editing in Dashboard

In Product ES|QL Documentation



```
from metrics* | stats max_cpu = max(kubernetes.pod.cpu.usage.node.pct), avg_mem = max(kubernetes.pod.memory.usage.bytes) by kubernetes.pod.n...
```

ES|QL

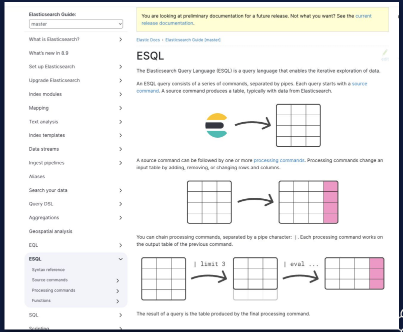
An ES|QL (Elasticsearch query language) query consists of a series of commands, separated by pipe characters: `|`. Each query starts with a **source command**, which produces a table, typically with data from Elasticsearch.

A source command can be followed by one or more **processing commands**. Processing commands can change the output table of the previous command by adding, removing, and changing rows and columns.

```
source-command
| processing-command1
| processing-command2
```

The result of a query is the table produced by the final

- In-line documentation right at your fingertips!
- Full documentation page: **ES|QL**



You are looking at preliminary documentation for a future release. Not what you want? See the current release documentation.

ESQL

The Elasticsearch Query Language (ESQL) is a query language that enables the iterative exploration of data.

An ESQL query consists of a series of commands, separated by pipes. Each query starts with a source command. A source command produces a table, typically with data from Elasticsearch.

A source command can be followed by one or more processing commands. Processing commands change an input table by adding, removing, or changing rows and columns.

You can chain processing commands, separated by a pipe character: `|`. Each processing command works on the output table of the previous command.

The result of a query is the table produced by the final processing command.

ES|QL Workshop

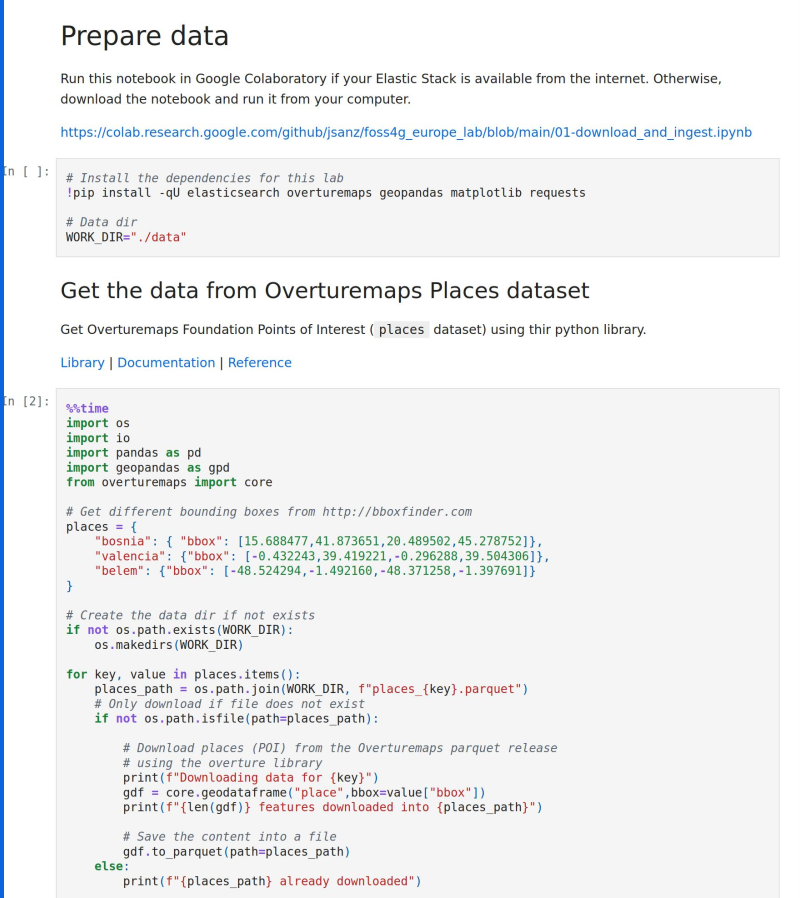
01-download_and_ingest.ipynb

Details on how to download and ingest in Elasticsearch datasets for this workshop:

- Overture Maps Foundation places dataset from parquet files
- Natural Earth countries zipped shapefile
- OSM, Geonames, and GHCD snapshots

To achieve these tasks:

- How to use the OvertureMaps Python API
- How to read parquet files into (Geo)Pandas Dataframes
- How to define Elasticsearch index mappings and bulk uploading efficiently large datasets
- Use the Kibana API to create Data Views
- Some troubleshooting for a misbehaving geospatial feature
- How to restore snapshots from read-only HTTP repositories



Prepare data

Run this notebook in Google Colaboratory if your Elastic Stack is available from the internet. Otherwise, download the notebook and run it from your computer.

https://colab.research.google.com/github/jsanz/foss4g_europe_lab/blob/main/01-download_and_ingest.ipynb

In [ ]:

```
# Install the dependencies for this lab
!pip install -qU elasticsearch overturemaps geopandas matplotlib requests

# Data dir
WORK_DIR="./data"
```

Get the data from Overturemaps Places dataset

Get Overturemaps Foundation Points of Interest (`places` dataset) using thir python library.

Library | Documentation | Reference

In [2]:

```
%%time
import os
import io
import pandas as pd
import geopandas as gpd
from overturemaps import core

# Get different bounding boxes from http://bboxfinder.com
places = {
    "bosnia": { "bbox": [15.688477,41.873651,20.489502,45.278752]},
    "valencia": {"bbox": [-0.432243,39.419221,-0.296288,39.504306]},
    "belem": {"bbox": [-48.524294,-1.492160,-48.371258,-1.397691]}
}

# Create the data dir if not exists
if not os.path.exists(WORK_DIR):
    os.makedirs(WORK_DIR)

for key, value in places.items():
    places_path = os.path.join(WORK_DIR, f"places_{key}.parquet")
    # Only download if file does not exist
    if not os.path.isfile(path=places_path):

        # Download places (POI) from the Overturemaps parquet release
        # using the overture library
        print(f"Downloading data for {key}")
        gdf = core.geodataframe("place",bbox=value["bbox"])
        print(f"{len(gdf)} features downloaded into {places_path}")

        # Save the content into a file
        gdf.to_parquet(path=places_path)
    else:
        print(f"{places_path} already downloaded")
```

elastic

Lab datasets

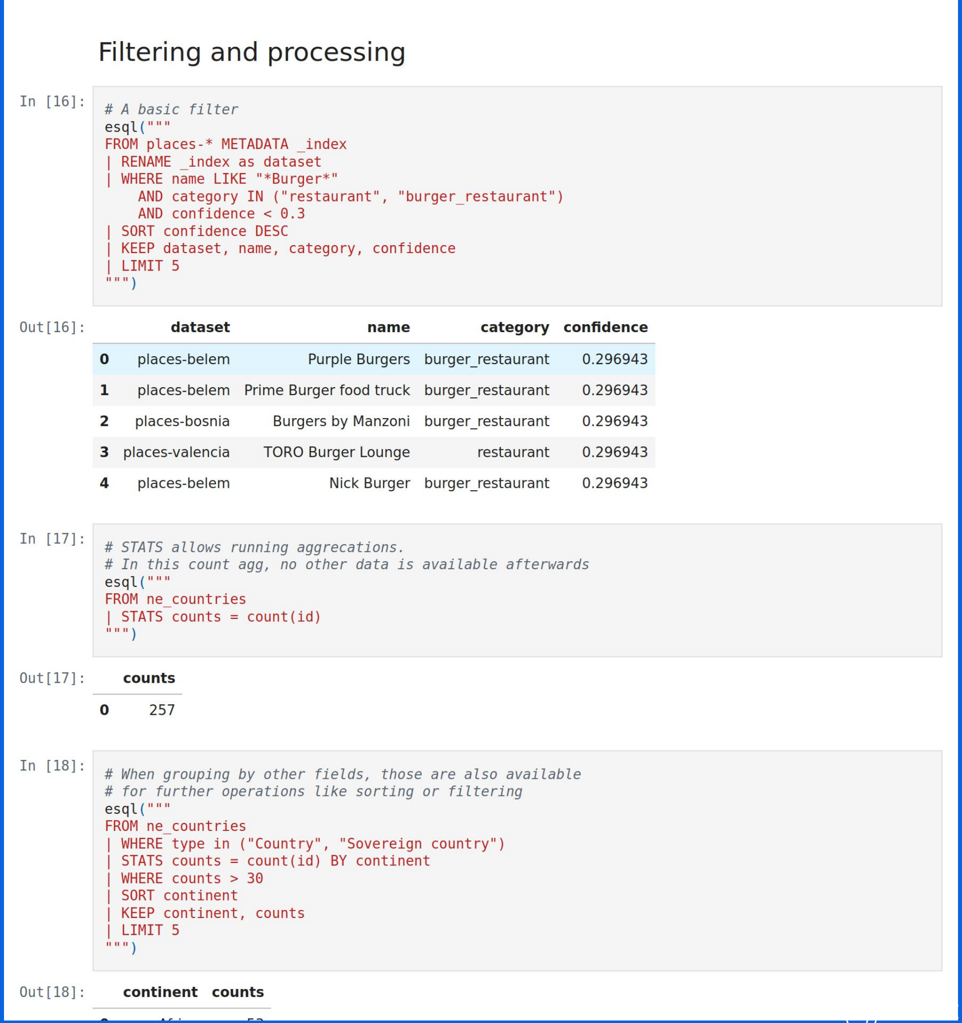
```
GET _cat/indices?v&h=index,docs.count,dataset.size&s=index
```

```
index                                                              docs.count dataset.size
-------------------------------------------------------------------------------------------
.ds-kibana_sample_data_logs-2025.07.11-000001                           14074          9mb
airports                                                                  891       97.9kb
flight_tracking_2025-07-10                                            2047259      376.4mb
geonames                                                             11968314        1.9gb
ghcnd_daily_observations                                             29075053          4gb
kibana_sample_data_ecommerce                                             4675        4.3mb
kibana_sample_data_flights                                              13014        5.9mb
ne_countries                                                              257       35.1mb
osm_andorra                                                            284619         55mb
osm_estonia                                                          12787609        2.8gb
osm_italy_centro                                                     43002709        8.4gb
osm_spain_valencia                                                   12355000        2.4gb
osm_usa_arizona                                                      31160000        5.1gb
places-auckland                                                         43678       17.6mb
places-belem                                                            27736       10.6mb
places-bosnia                                                          166644       60.4mb
places-capetown                                                         82148       32.6mb
places-seoul                                                           121128         46mb
places-valencia                                                         36193       14.8mb
places-victoria                                                         17475        7.7mb
```

02-esql.ipynb

With a helper function that takes a ES|QL query and return a (Geo)Dataframe, go through the different aspects of the language to learn its syntax:

- Source commands
- Controlling the output
- Processing commands
    - Filtering
    - Aggregations
    - Joins



Filtering and processing

In [16]:

```
# A basic filter
esql("""
FROM places-* METADATA _index
| RENAME _index as dataset
| WHERE name LIKE "*Burger*"
    AND category IN ("restaurant", "burger_restaurant")
    AND confidence < 0.3
| SORT confidence DESC
| KEEP dataset, name, category, confidence
| LIMIT 5
""")
```

Out[16]:

	dataset	name	category	confidence
0	places-belem	Purple Burgers	burger_restaurant	0.296943
1	places-belem	Prime Burger food truck	burger_restaurant	0.296943
2	places-bosnia	Burgers by Manzoni	burger_restaurant	0.296943
3	places-valencia	TORO Burger Lounge	restaurant	0.296943
4	places-belem	Nick Burger	burger_restaurant	0.296943

In [17]:

```
# STATS allows running aggrecations.
# In this count agg, no other data is available afterwards
esql("""
FROM ne_countries
| STATS counts = count(id)
""")
```

Out[17]:

	counts
0	257

In [18]:

```
# When grouping by other fields, those are also available
# for further operations like sorting or filtering
esql("""
FROM ne_countries
| WHERE type in ("Country", "Sovereign country")
| STATS counts = count(id) BY continent
| WHERE counts > 30
| SORT continent
| KEEP continent, counts
| LIMIT 5
""")
```

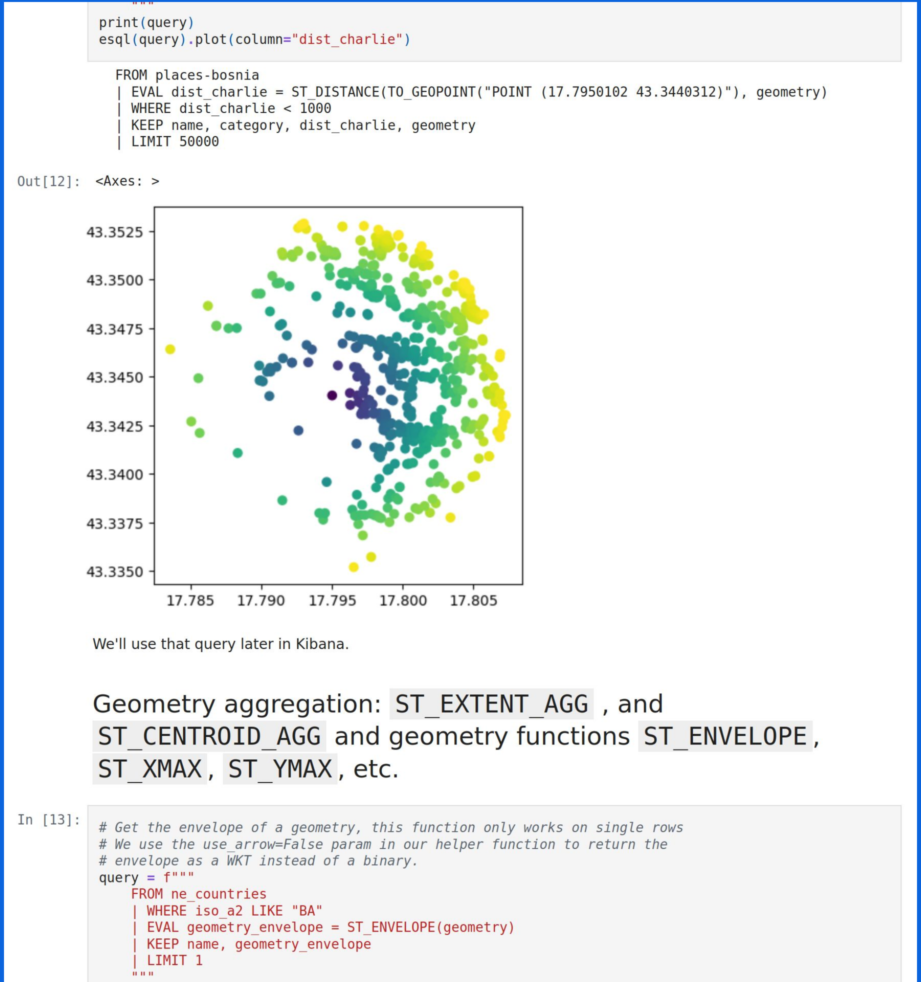
Out[18]:

	continent	counts

03-geospatial_esql.ipynb

Focusing on the current geospatial features in ES|QL:

- Type conversions
- Distance computations
- Geometry aggregations
- Geometry functions



```
print(query)
esql(query).plot(column="dist_charlie")
```

```
FROM places-bosnia
| EVAL dist_charlie = ST_DISTANCE(TO_GEOPOINT("POINT (17.7950102 43.3440312)"), geometry)
| WHERE dist_charlie < 1000
| KEEP name, category, dist_charlie, geometry
| LIMIT 50000
```

Out[12]: `<Axes: >`

We'll use that query later in Kibana.

Geometry aggregation: `ST_EXTENT_AGG` , and `ST_CENTROID_AGG` and geometry functions `ST_ENVELOPE`, `ST_XMAX`, `ST_YMAX`, etc.

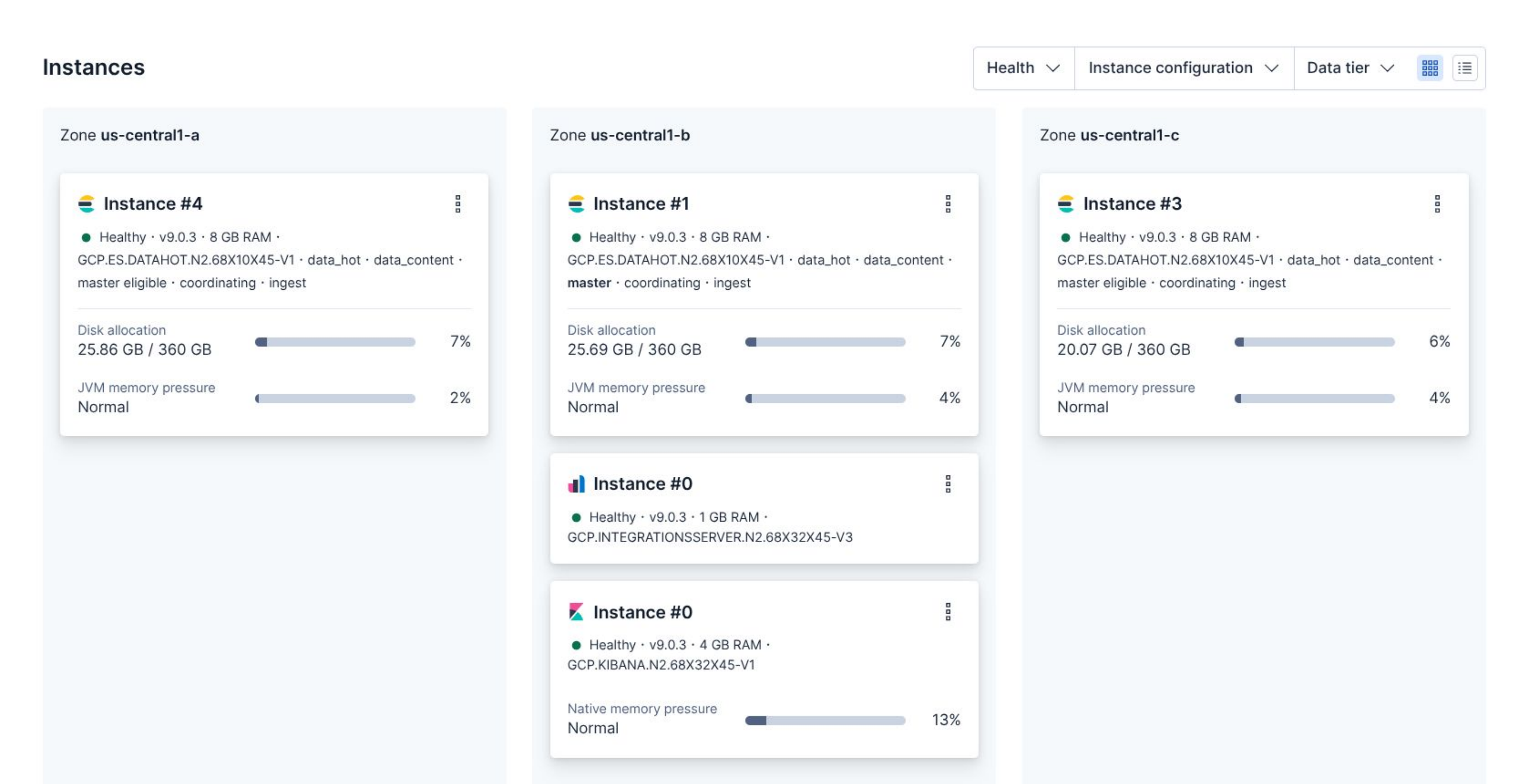
In [13]:

```
# Get the envelope of a geometry, this function only works on single rows
# We use the use_arrow=False param in our helper function to return the
# envelope as a WKT instead of a binary.
query = f"""
    FROM ne_countries
    | WHERE iso_a2 LIKE "BA"
    | EVAL geometry_envelope = ST_ENVELOPE(geometry)
    | KEEP name, geometry_envelope
    | LIMIT 1
    """
```

Kibana

Home for all Elastic graphic applications

Please, now it is a good time to connect to your Kibana instance if running locally, or to the Elastic Cloud Kibana instance provided in the workshop notes.



Instances

Zone **us-central1-a**

Instance #4

● Healthy · v9.0.3 · 8 GB RAM · GCP.ES.DATAHOT.N2.68X10X45-V1 · data_hot · data_content · master eligible · coordinating · ingest

Disk allocation
25.86 GB / 360 GB — 7%

JVM memory pressure
Normal — 2%

Zone **us-central1-b**

Instance #1

● Healthy · v9.0.3 · 8 GB RAM · GCP.ES.DATAHOT.N2.68X10X45-V1 · data_hot · data_content · **master** · coordinating · ingest

Disk allocation
25.69 GB / 360 GB — 7%

JVM memory pressure
Normal — 4%

Instance #0

● Healthy · v9.0.3 · 1 GB RAM · GCP.INTEGRATIONSSERVER.N2.68X32X45-V3

Instance #0

● Healthy · v9.0.3 · 4 GB RAM · GCP.KIBANA.N2.68X32X45-V1

Native memory pressure
Normal — 13%

Zone **us-central1-c**

Instance #3

● Healthy · v9.0.3 · 8 GB RAM · GCP.ES.DATAHOT.N2.68X10X45-V1 · data_hot · data_content · master eligible · coordinating · ingest

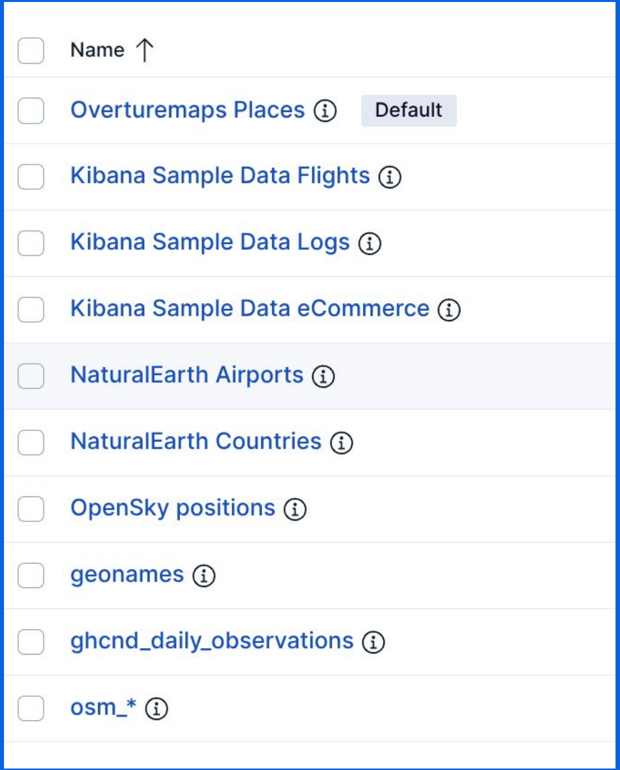
Disk allocation
20.07 GB / 360 GB — 6%

JVM memory pressure
Normal — 4%

Lab datasets

Datasets to experiment with

- Kibana sample datasets
  - `kibana_sample_data_commerce`
  - `Kibana_sample_data_flights`
  - `kibana_sample_data_logs`
- Natural Earth airports and countries
  - `airports`
  - `ne_countries`
- Open Sky positions: `flight_tracking_2025-07-10`
- Overturemaps Places: `places*`
- Geonames gazetter: `geonames`
- GHCD daily observations: `ghcd`
- OSM data: `osm*`

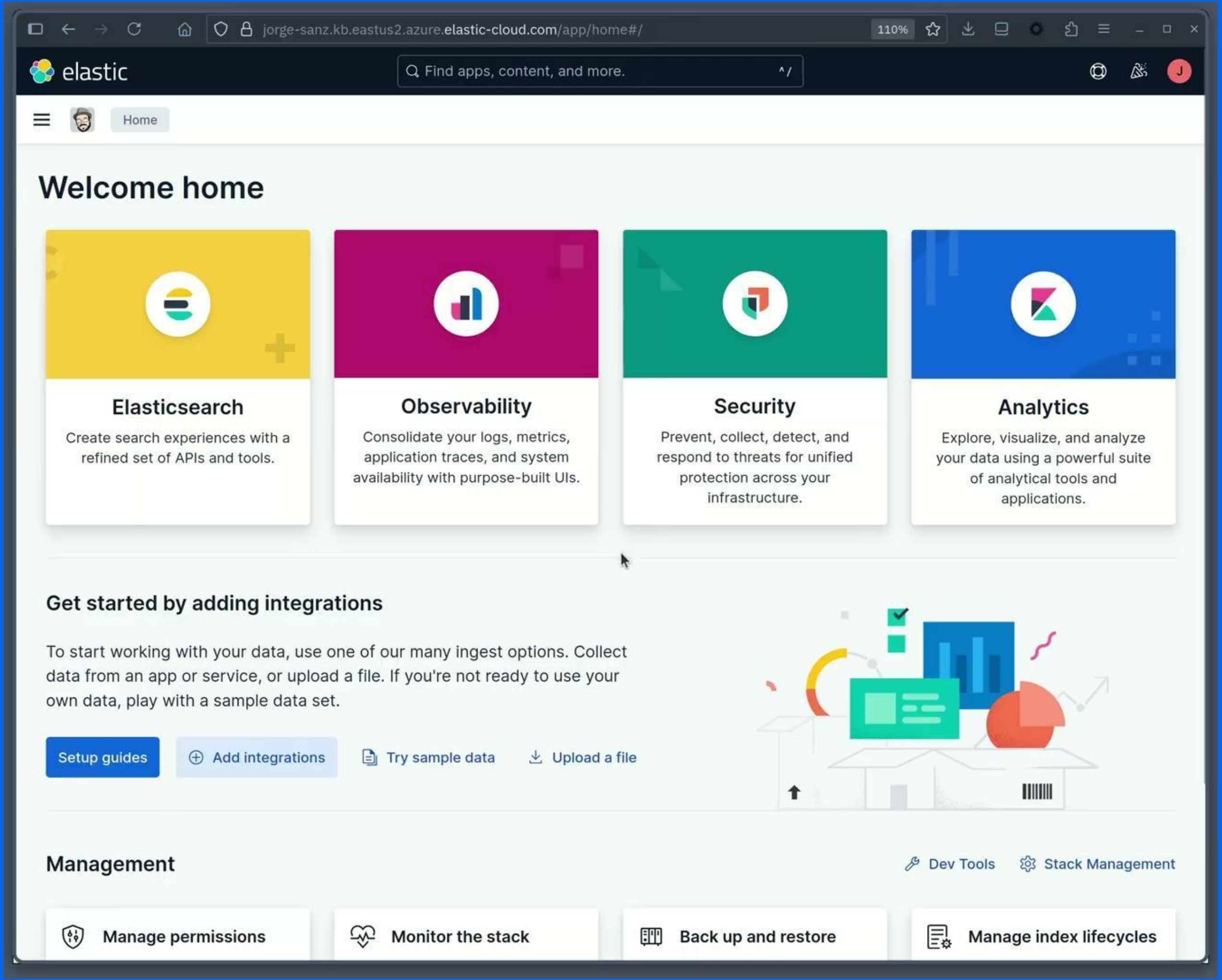


□	Name ↑
□	Overturemaps Places ⓘ Default
□	Kibana Sample Data Flights ⓘ
□	Kibana Sample Data Logs ⓘ
□	Kibana Sample Data eCommerce ⓘ
□	NaturalEarth Airports ⓘ
□	NaturalEarth Countries ⓘ
□	OpenSky positions ⓘ
□	geonames ⓘ
□	ghcnd_daily_observations ⓘ
□	osm_* ⓘ

Lab datasets

```
GET _cat/indices?v&h=index,docs.count,dataset.size&s=index
```

```
index                                                              docs.count dataset.size
------------------------------------------------------------------------------------------
.ds-kibana_sample_data_logs-2025.07.11-000001                           14074          9mb
airports                                                                  891       97.9kb
flight_tracking_2025-07-10                                            2047259      376.4mb
geonames                                                             11968314        1.9gb
ghcnd_daily_observations                                             29075053          4gb
kibana_sample_data_ecommerce                                             4675        4.3mb
kibana_sample_data_flights                                              13014        5.9mb
ne_countries                                                              257       35.1mb
osm_andorra                                                            284619         55mb
osm_estonia                                                          12787609        2.8gb
osm_italy_centro                                                     43002709        8.4gb
osm_spain_valencia                                                   12355000        2.4gb
osm_usa_arizona                                                      31160000        5.1gb
places-auckland                                                         43678       17.6mb
places-belem                                                            27736       10.6mb
places-bosnia                                                          166644       60.4mb
places-capetown                                                         82148       32.6mb
places-seoul                                                           121128         46mb
places-valencia                                                         36193       14.8mb
places-victoria                                                         17475        7.7mb
```



jorge-sanz.kb.eastus2.azure.elastic-cloud.com/app/home#/

Find apps, content, and more.

Home

Welcome home

Elasticsearch
Create search experiences with a refined set of APIs and tools.

Observability
Consolidate your logs, metrics, application traces, and system availability with purpose-built UIs.

Security
Prevent, collect, detect, and respond to threats for unified protection across your infrastructure.

Analytics
Explore, visualize, and analyze your data using a powerful suite of analytical tools and applications.

Get started by adding integrations

To start working with your data, use one of our many ingest options. Collect data from an app or service, or upload a file. If you're not ready to use your own data, play with a sample data set.

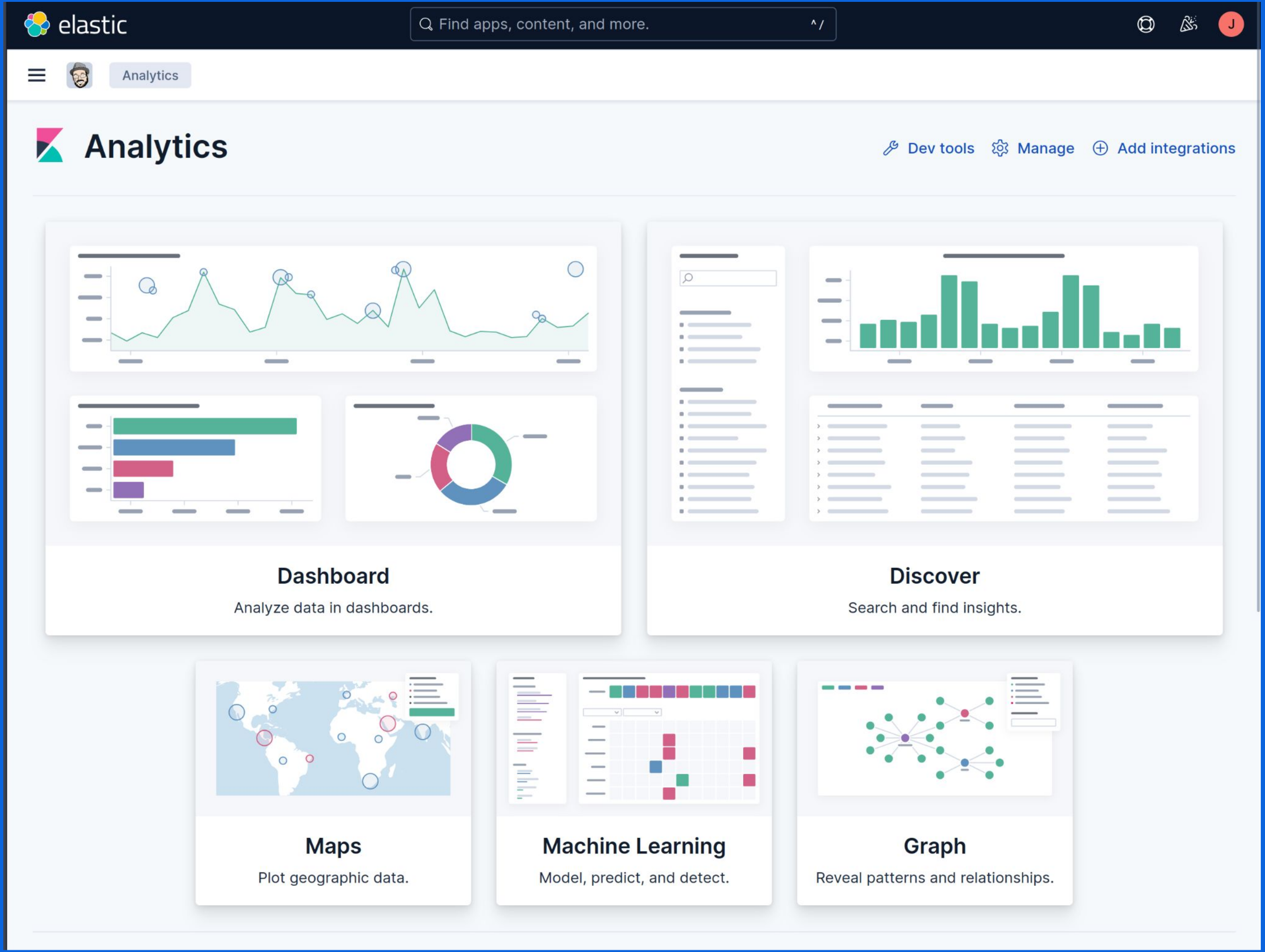
Setup guides | Add integrations | Try sample data | Upload a file

Management

Dev Tools | Stack Management

Manage permissions | Monitor the stack | Back up and restore | Manage index lifecycles

ANALYTICS



elastic

Find apps, content, and more.

Analytics

Analytics

Dev tools Manage Add integrations

Dashboard

Analyze data in dashboards.

Discover

Search and find insights.

Maps

Plot geographic data.

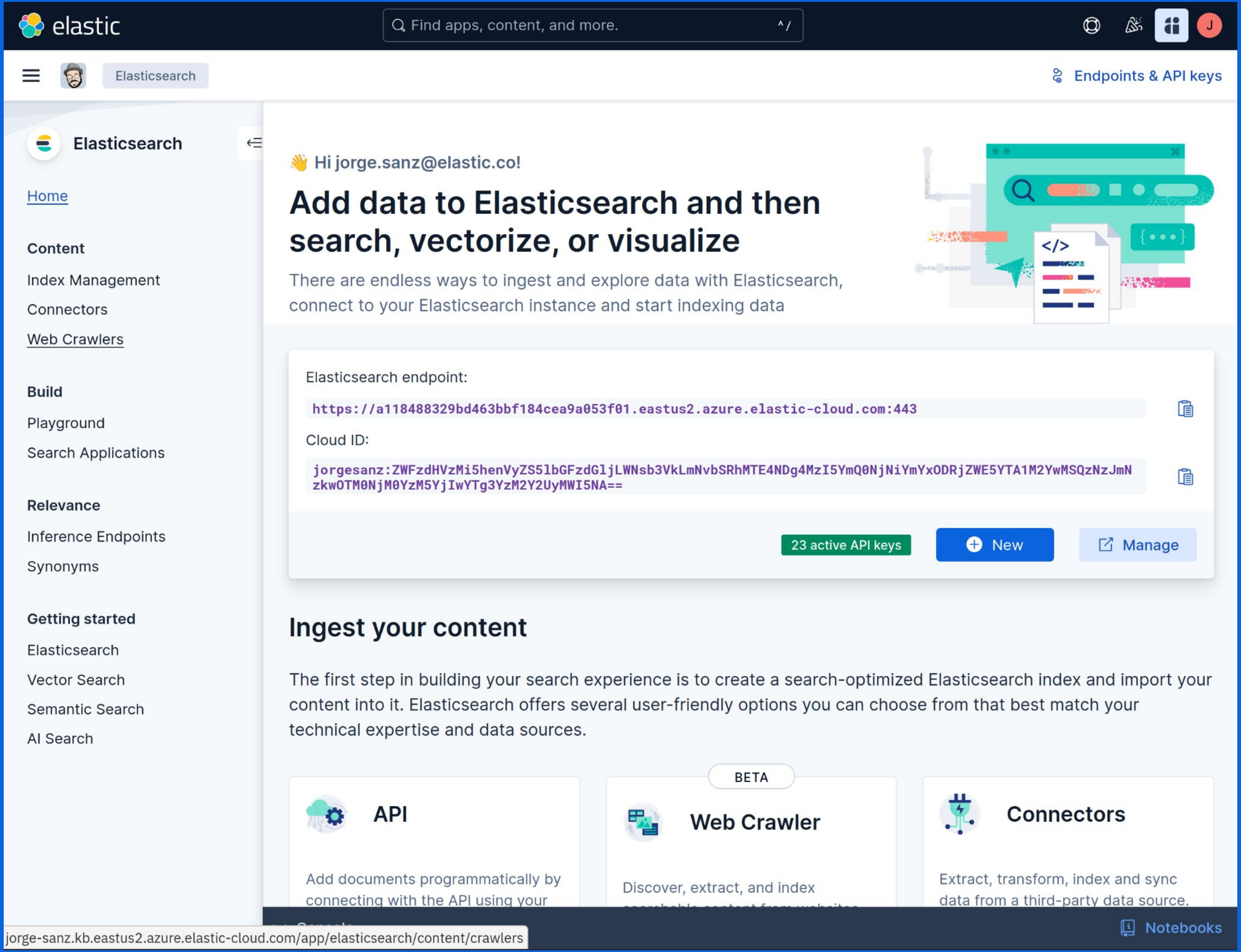
Machine Learning

Model, predict, and detect.

Graph

Reveal patterns and relationships.

Search platform



👋 Hi jorge.sanz@elastic.co!

Add data to Elasticsearch and then search, vectorize, or visualize

There are endless ways to ingest and explore data with Elasticsearch, connect to your Elasticsearch instance and start indexing data

Elasticsearch endpoint:

```
https://a118488329bd463bbf184cea9a053f01.eastus2.azure.elastic-cloud.com:443
```

Cloud ID:

```
jorgesanz:ZWFzdHVzMi5henVyZS5lbGFzdGljLWNsb3VkLmNvbSRhMTE4NDg4MzI5YmQ0NjNiYmYxODRjZWE5YTA1M2YwMSQzNzJmN
zkwOTM0NjM0YzM5YjIwYTg3YzM2Y2UyMWI5NA==
```

Ingest your content

The first step in building your search experience is to create a search-optimized Elasticsearch index and import your content into it. Elasticsearch offers several user-friendly options you can choose from that best match your technical expertise and data sources.

API

Add documents programmatically by connecting with the API using your

BETA

Web Crawler

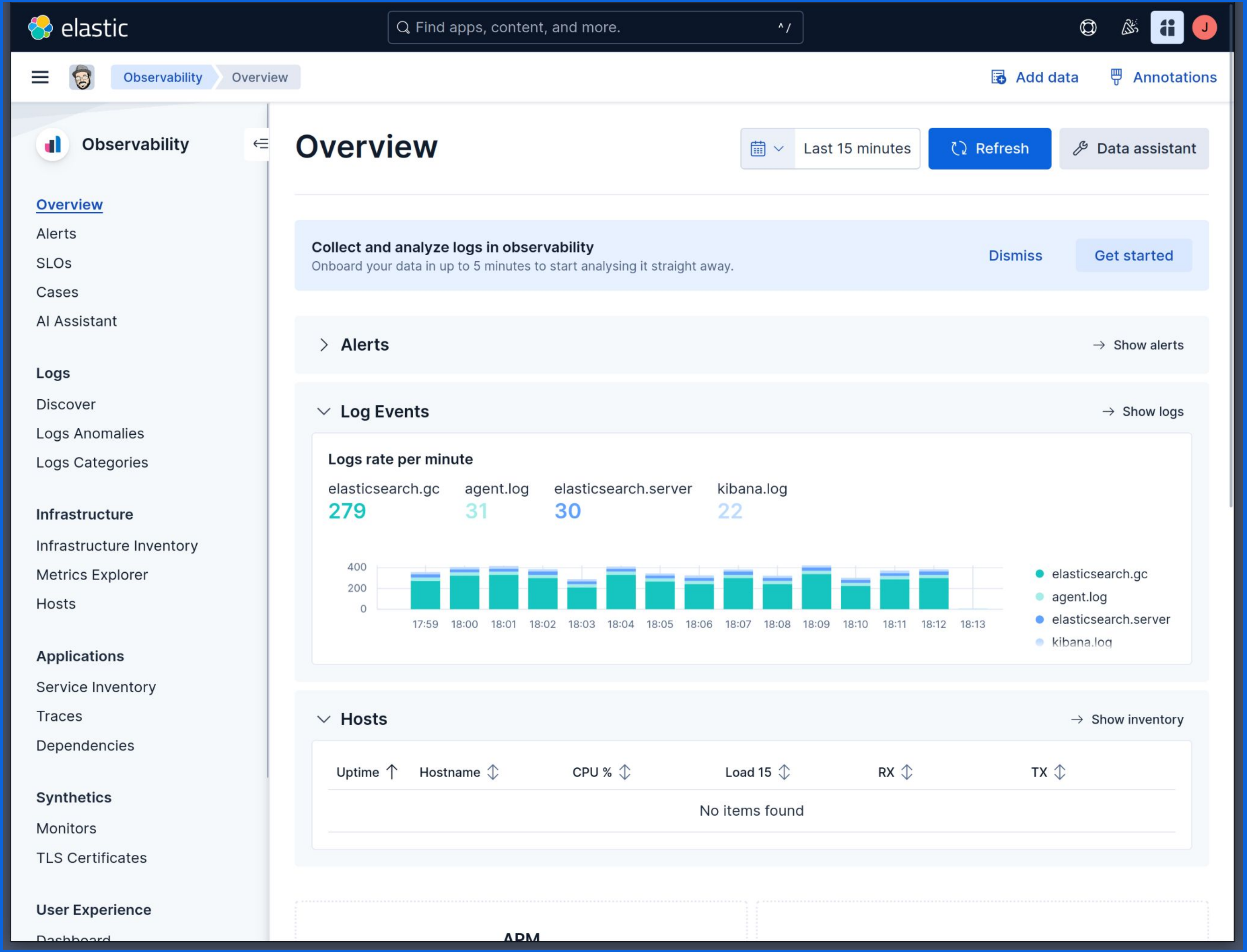
Discover, extract, and index

Connectors

Extract, transform, index and sync data from a third-party data source.

Notebooks
jorge-sanz.kb.eastus2.azure.elastic-cloud.com/app/elasticsearch/content/crawlers

Observability



elastic

Find apps, content, and more.

Observability > Overview

Add data   Annotations

Observability

- Overview
- Alerts
- SLOs
- Cases
- AI Assistant

Logs

- Discover
- Logs Anomalies
- Logs Categories

Infrastructure

- Infrastructure Inventory
- Metrics Explorer
- Hosts

Applications

- Service Inventory
- Traces
- Dependencies

Synthetics

- Monitors
- TLS Certificates

User Experience

- Dashboard

Overview

Last 15 minutes   Refresh   Data assistant

Collect and analyze logs in observability
Onboard your data in up to 5 minutes to start analysing it straight away.

Dismiss   Get started

Alerts   → Show alerts

Log Events   → Show logs

Logs rate per minute

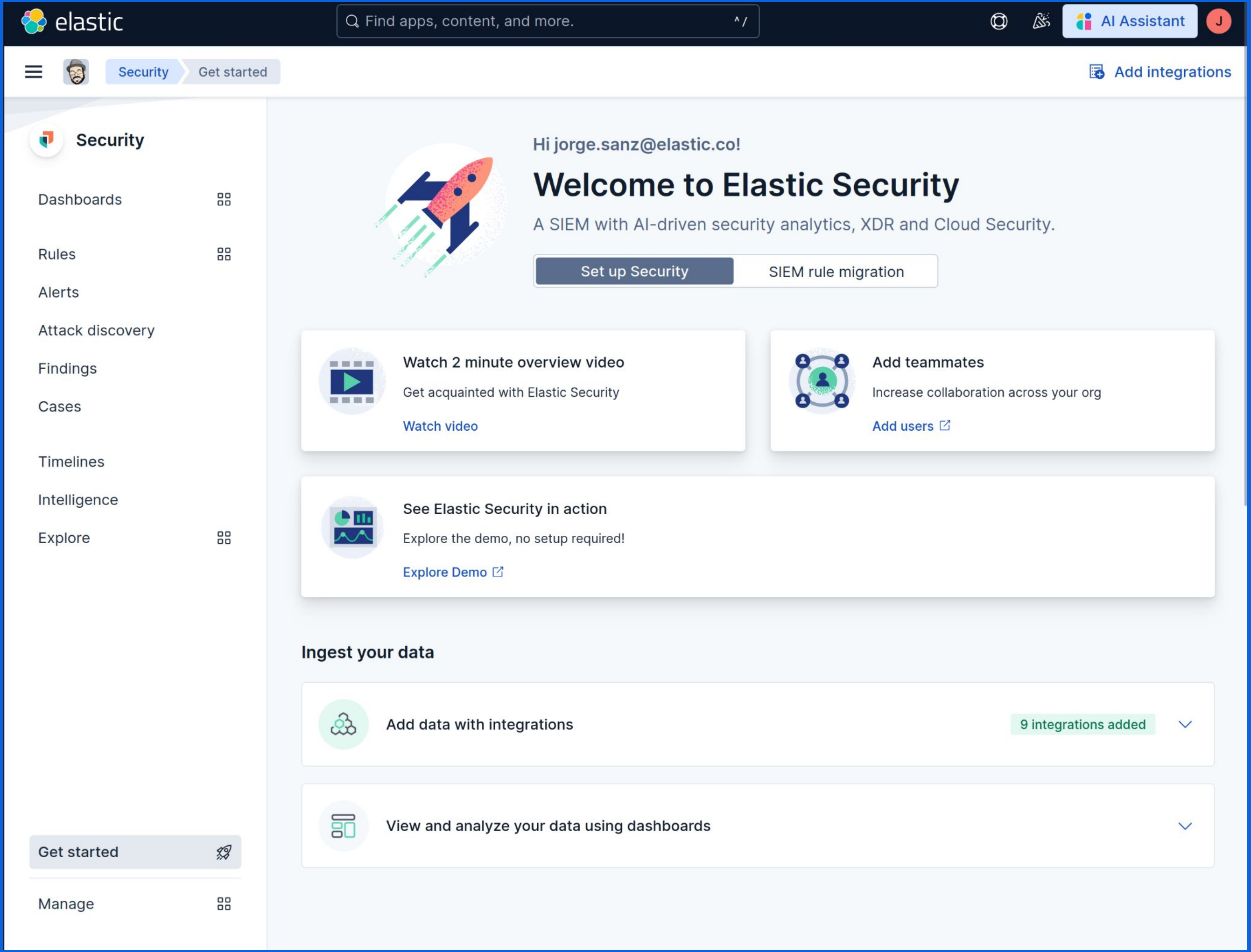
elasticsearch.gc	agent.log	elasticsearch.server	kibana.log
279	31	30	22

Hosts   → Show inventory

Uptime ↑	Hostname	CPU %	Load 15	RX	TX
No items found					

APM

Security



elastic

Find apps, content, and more.

AI Assistant

Security > Get started

Add integrations

Security

- Dashboards
- Rules
- Alerts
- Attack discovery
- Findings
- Cases
- Timelines
- Intelligence
- Explore
- Get started
- Manage

Hi jorge.sanz@elastic.co!

Welcome to Elastic Security

A SIEM with AI-driven security analytics, XDR and Cloud Security.

Set up Security | SIEM rule migration

Watch 2 minute overview video

Get acquainted with Elastic Security

Watch video

Add teammates

Increase collaboration across your org

Add users

See Elastic Security in action

Explore the demo, no setup required!

Explore Demo

Ingest your data

Add data with integrations — 9 integrations added

View and analyze your data using dashboards

Who uses Kibana?

- Log/metrics and security analysts
- Data service providers
- Business analysts
- Data scientists
- Anyone trying to make sense of data

elastic

Kibana Analytics

Kibana analytics

Applications that power data analysis in Kibana.

Machine learning features are not part of the Basic offering and not covered today.

Data Views

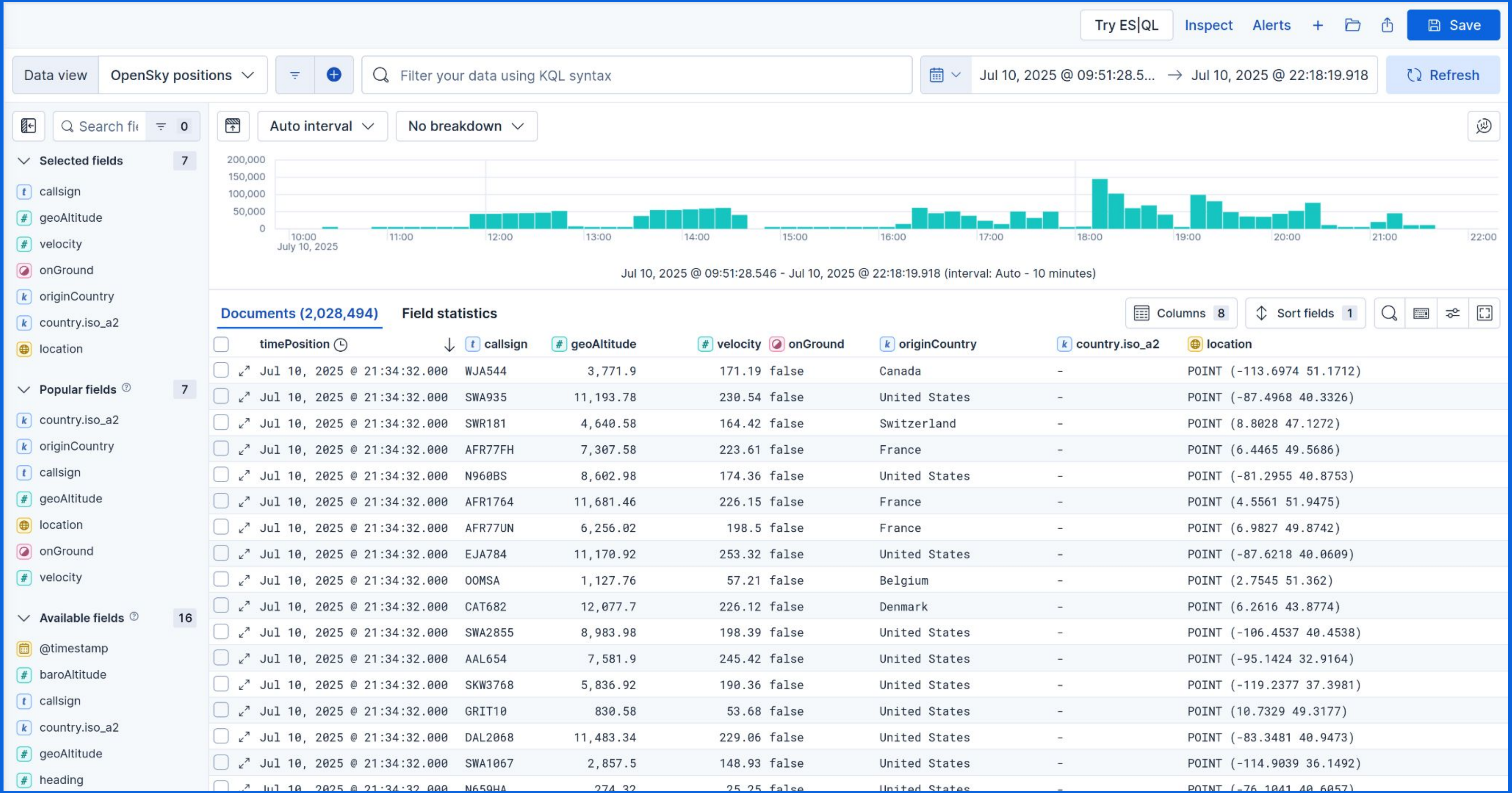
Discover

Dashboards

Visualizations: Lens & ES|QL

Maps

Discover



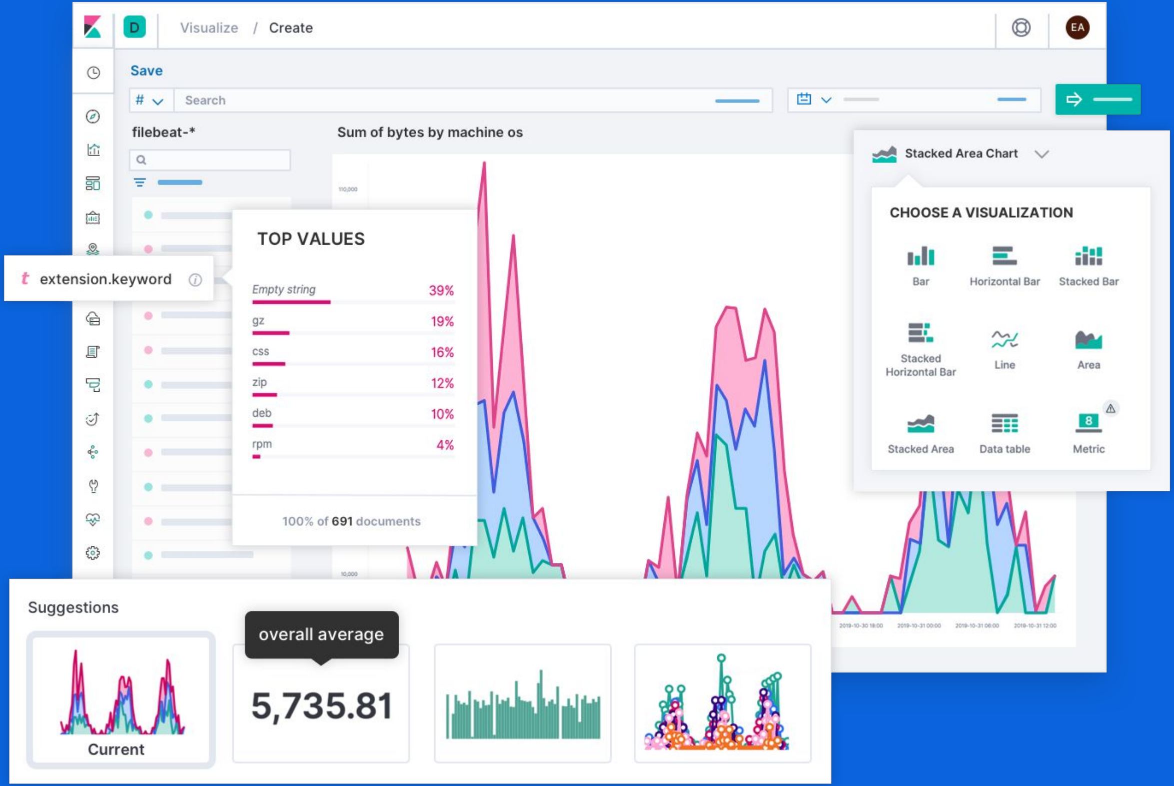
Jul 10, 2025 @ 09:51:28.546 - Jul 10, 2025 @ 22:18:19.918 (interval: Auto - 10 minutes)

timePosition	callsign	geoAltitude	velocity	onGround	originCountry	country.iso_a2	location
Jul 10, 2025 @ 21:34:32.000	WJA544	3,771.9	171.19	false	Canada	-	POINT (-113.6974 51.1712)
Jul 10, 2025 @ 21:34:32.000	SWA935	11,193.78	230.54	false	United States	-	POINT (-87.4968 40.3326)
Jul 10, 2025 @ 21:34:32.000	SWR181	4,640.58	164.42	false	Switzerland	-	POINT (8.8028 47.1272)
Jul 10, 2025 @ 21:34:32.000	AFR77FH	7,307.58	223.61	false	France	-	POINT (6.4465 49.5686)
Jul 10, 2025 @ 21:34:32.000	N960BS	8,602.98	174.36	false	United States	-	POINT (-81.2955 40.8753)
Jul 10, 2025 @ 21:34:32.000	AFR1764	11,681.46	226.15	false	France	-	POINT (4.5561 51.9475)
Jul 10, 2025 @ 21:34:32.000	AFR77UN	6,256.02	198.5	false	France	-	POINT (6.9827 49.8742)
Jul 10, 2025 @ 21:34:32.000	EJA784	11,170.92	253.32	false	United States	-	POINT (-87.6218 40.0609)
Jul 10, 2025 @ 21:34:32.000	OOMSA	1,127.76	57.21	false	Belgium	-	POINT (2.7545 51.362)
Jul 10, 2025 @ 21:34:32.000	CAT682	12,077.7	226.12	false	Denmark	-	POINT (6.2616 43.8774)
Jul 10, 2025 @ 21:34:32.000	SWA2855	8,983.98	198.39	false	United States	-	POINT (-106.4537 40.4538)
Jul 10, 2025 @ 21:34:32.000	AAL654	7,581.9	245.42	false	United States	-	POINT (-95.1424 32.9164)
Jul 10, 2025 @ 21:34:32.000	SKW3768	5,836.92	190.36	false	United States	-	POINT (-119.2377 37.3981)
Jul 10, 2025 @ 21:34:32.000	GRIT10	830.58	53.68	false	United States	-	POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000	DAL2068	11,483.34	229.06	false	United States	-	POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000	SWA1067	2,857.5	148.93	false	United States	-	POINT (-114.9039 36.1492)
Jul 10, 2025 @ 21:34:32.000	N659HA	274.32	25.25	false	United States	-	POINT (-76.1041 40.6057)

Lens

Your data in front of you

- Explore your fields with a single click
- Drag and drop
- Go from nothing to visual insights with a single mouse gesture.
- Smart suggestions
- Let Lens help guide your analysis with useful chart suggestions



TOP VALUES

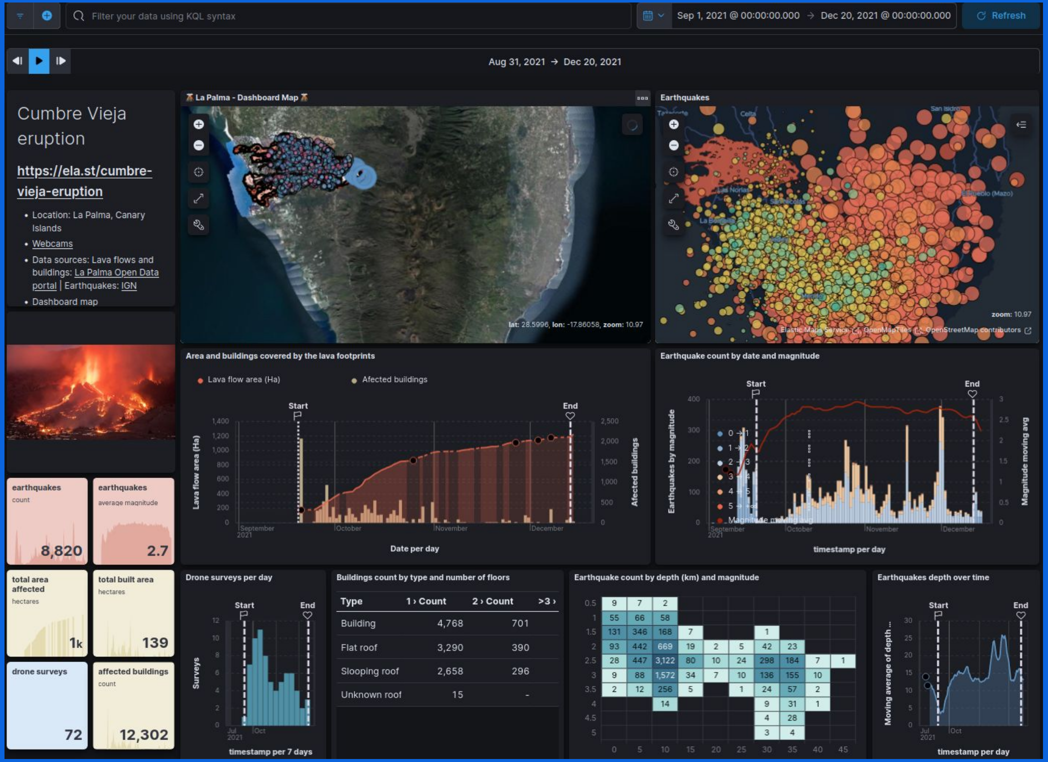
Empty string	39%
gz	19%
css	16%
zip	12%
deb	10%
rpm	4%

100% of 691 documents

Dashboards

All your information in a single place

- Combine multiple visualizations: **panels**
- Time Range + Search Bar + Filters
- Panels can use filters to perform **drill downs**
- Panels can have custom **time ranges and filters**
- **Share**
- **Export** to PDF or PNG 🛒



Filter your data using KQL syntax

Sep 1, 2021 @ 00:00:00.000 → Dec 20, 2021 @ 00:00:00.000

Refresh

Aug 31, 2021 → Dec 20, 2021

Cumbre Vieja eruption

https://ela.st/cumbre-vieja-eruption

- Location: La Palma, Canary Islands
- Webcams
- Data sources: Lava flows and buildings: La Palma Open Data portal | Earthquakes: IGN
- Dashboard map

earthquakes
count
8,820

earthquakes
average magnitude
2.7

total area affected
hectares
1k

total built area
hectares
139

drone surveys
72

affected buildings
count
12,302

Buildings count by type and number of floors

Type	1 › Count	2 › Count	>3 ›
Building	4,768	701	
Flat roof	3,290	390	
Slooping roof	2,658	296	
Unknown roof	15	-	

Earthquake count by depth (km) and magnitude

	0	5	10	15	20	25	30	35	40	45
0.5	9	7	2							
1	55	66	58							
1.5	131	346	168	7			1			
2	93	442	669	19	2	5	42	23		
2.5	28	447	3,122	80	10	24	298	184	7	1
3	9	88	1,572	34	7	10	136	155	10	
3.5	2	12	256	5		1	24	57	2	
4			14				9	31	1	
4.5							4	28		
5							3	4		

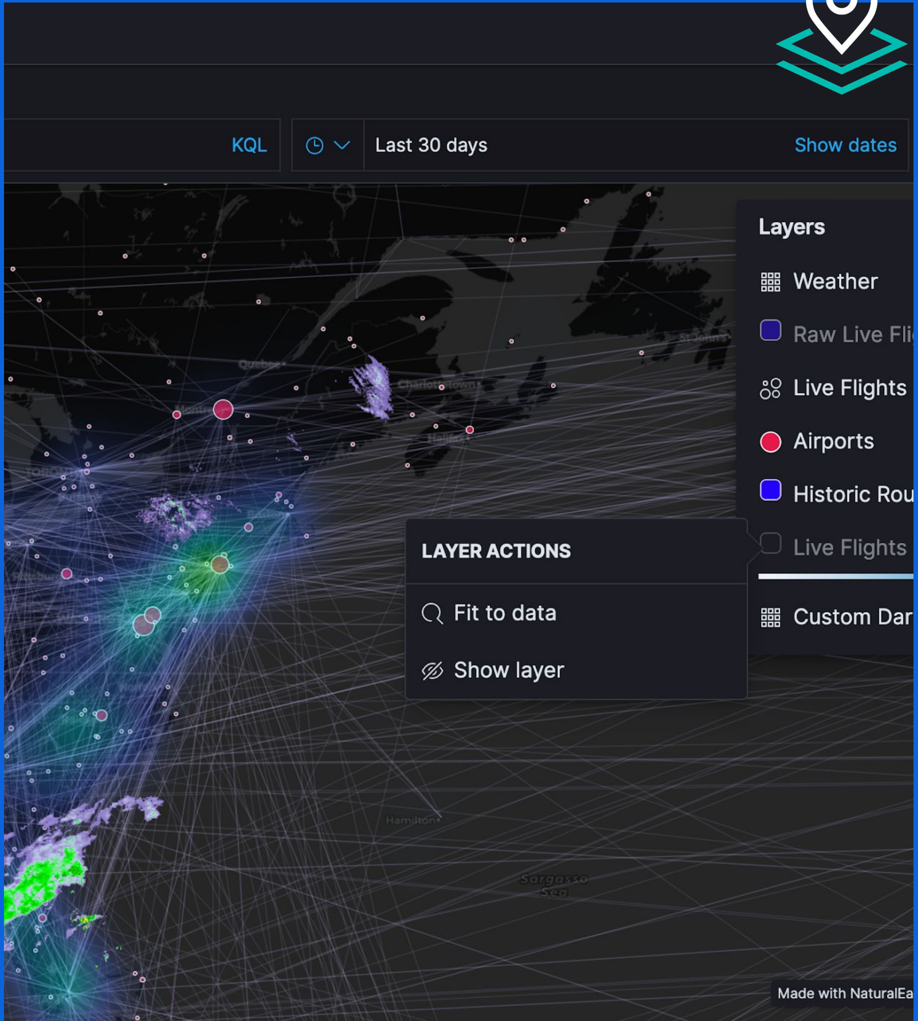
Elastic Maps

elastic

Elastic Maps

Geo Analytics interface within Kibana

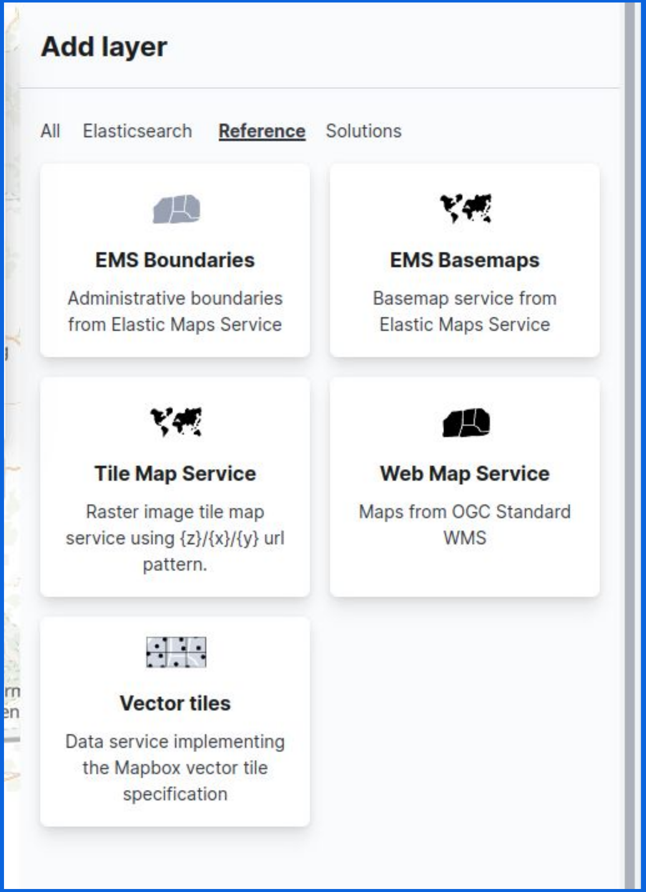
- **Friendly** user experience
- **Aggregations**: heat map, clustering, grids, geoline
- Data driven **styling**
- **Tools** for drawing, filtering, measuring
- Add layers from **external** tile servers
- Used alone or in dashboards
- **Embedded** in other apps



Reference data

Data that provides context

- Elastic provides **basemaps** (OSM + OpenMapTiles) and **boundaries** (OSM + Natural Earth + Wikidata)
- Third party basemaps providers
  - **WMS**
  - **Tiles Maps Service**
  - **Vector Tiles**



Add layer

EMS Boundaries

Administrative boundaries from Elastic Maps Service

EMS Basemaps

Basemap service from Elastic Maps Service

Tile Map Service

Raster image tile map service using {z}/{x}/{y} url pattern.

Web Map Service

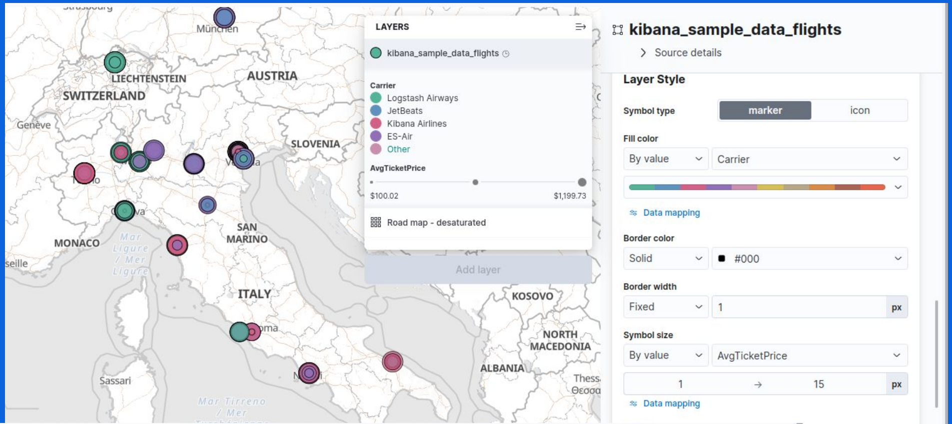
Maps from OGC Standard WMS

Vector tiles

Data service implementing the Mapbox vector tile specification

Data Driven Styling

- Quantitative:
  - Size
  - Widths
  - Color ramp
  - Label text
- Qualitative
  - Color palette
  - Label text



LAYERS

kibana_sample_data_flights

Carrier
- Logstash Airways
- JetBeats
- Kibana Airlines
- ES-Air
- Other

AvgTicketPrice

$100.02 — $1,199.73

Road map - desaturated

Add layer

kibana_sample_data_flights

Source details

Layer Style

Symbol type: marker | icon

Fill color: By value — Carrier

Data mapping

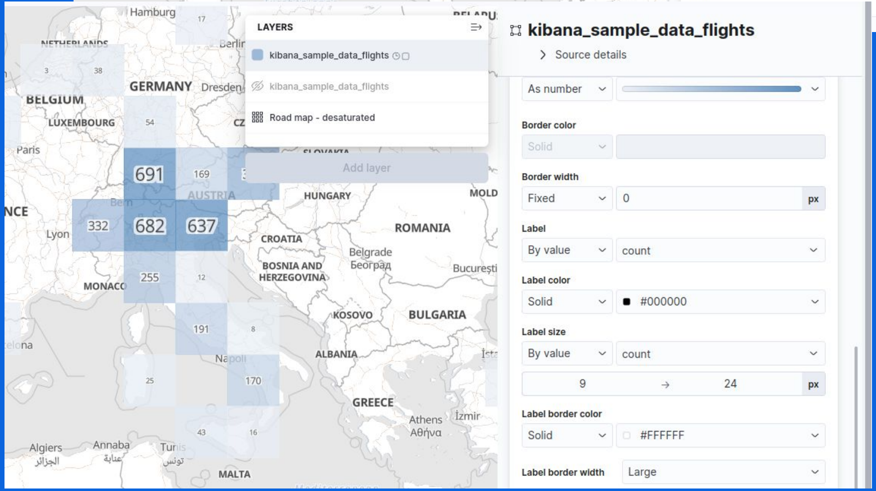
Border color: Solid — #000

Border width: Fixed — 1 px

Symbol size: By value — AvgTicketPrice

1 → 15 px

Data mapping



LAYERS

kibana_sample_data_flights

kibana_sample_data_flights

Road map - desaturated

Add layer

kibana_sample_data_flights

Source details

As number

Border color: Solid

Border width: Fixed — 0 px

Label: By value — count

Label color: Solid — #000000

Label size: By value — count

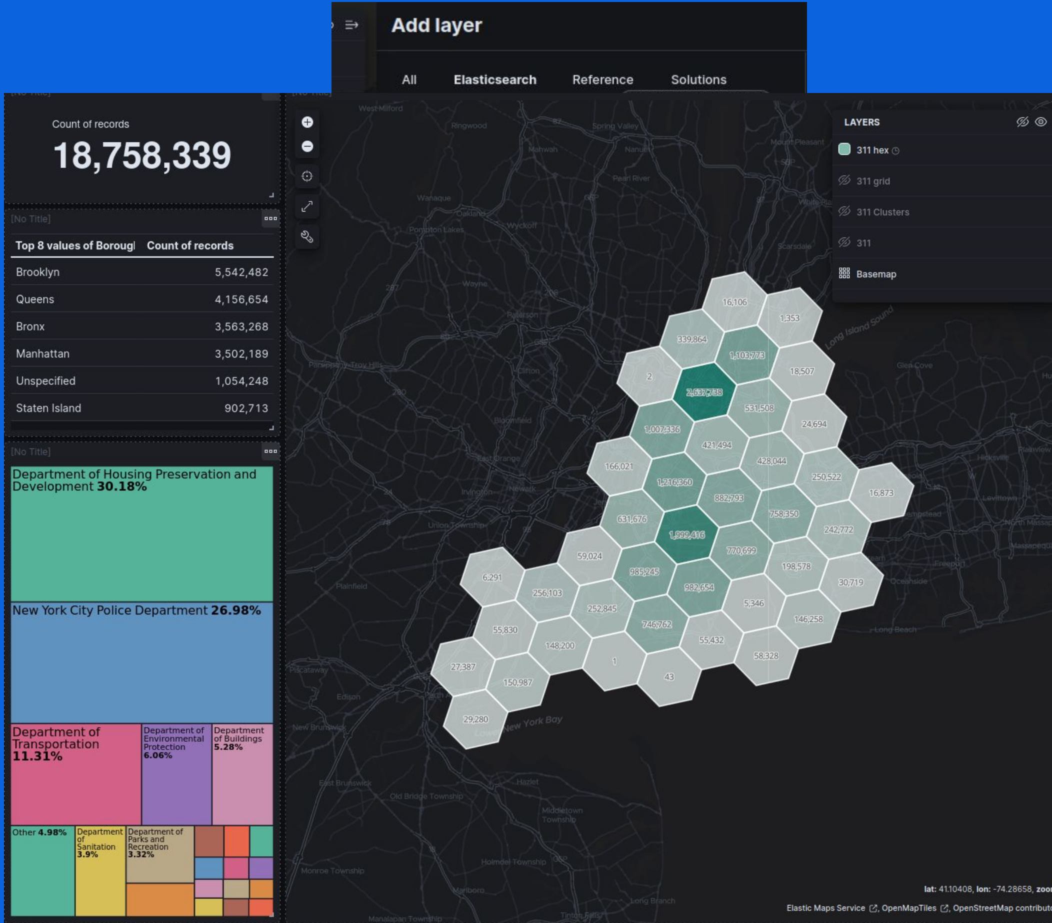
9 → 24 px

Label border color: Solid — #FFFFFF

Label border width: Large

Big Data Rendering

- Heatmap
- Clusters
- Tile aggregation
- Hexagon aggregation 🛒



Add layer

All Elasticsearch Reference Solutions

Count of records

18,758,339

[No Title]

Top 8 values of Borougl	Count of records
Brooklyn	5,542,482
Queens	4,156,654
Bronx	3,563,268
Manhattan	3,502,189
Unspecified	1,054,248
Staten Island	902,713

[No Title]

Department of Housing Preservation and Development **30.18%**

New York City Police Department **26.98%**

Department of Transportation **11.31%**

Department of Environmental Protection **6.06%**

Department of Buildings **5.28%**

Other **4.98%**

Department of Sanitation **3.9%**

Department of Parks and Recreation **3.32%**

LAYERS

- 🟩 311 hex
- 311 grid
- 311 Clusters
- 311
- Basemap

lat: 41.10408, lon: -74.28658, zoo

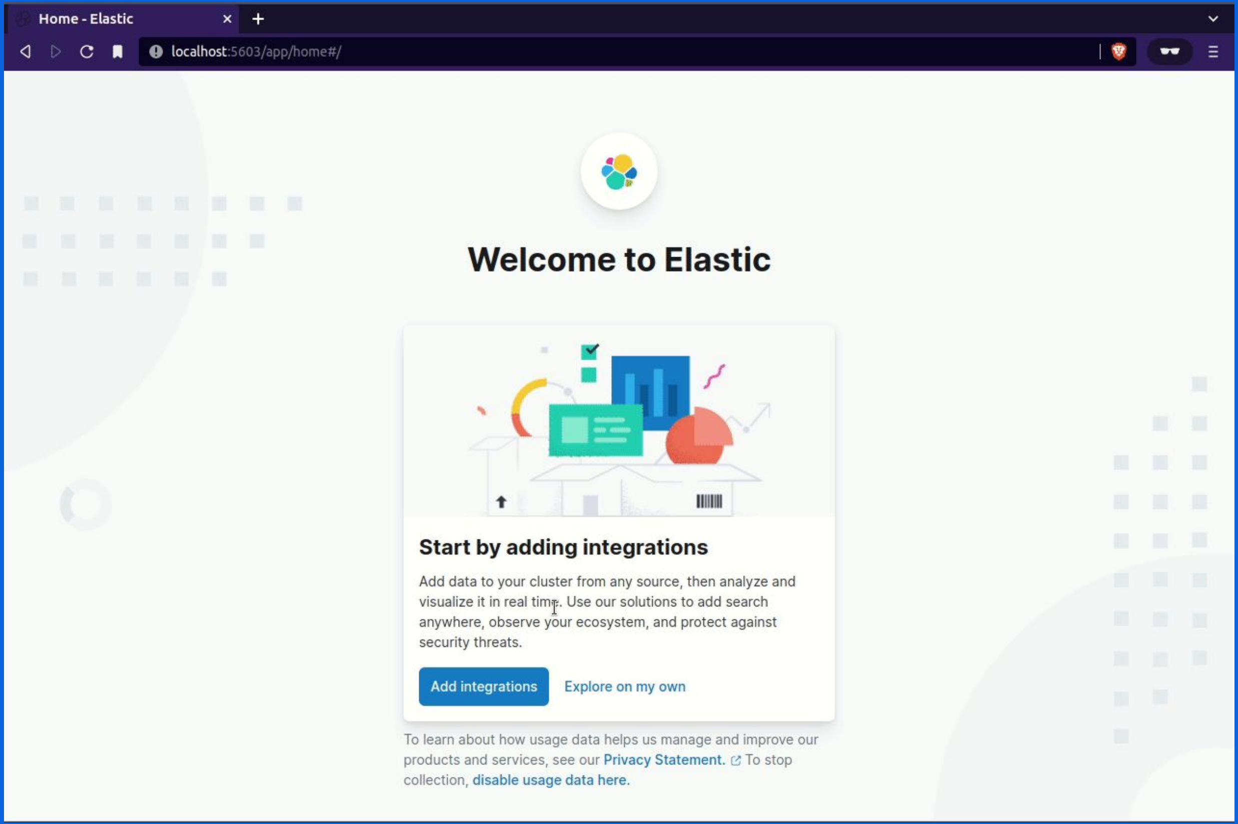
Elastic Maps Service, OpenMapTiles, OpenStreetMap contribut

Data Views

Data views

Abstracting index patterns

- A data view is an index pattern (like `places-*`) with some extra metadata
- An optional (but very common) field that defines the time for the document
- Custom formatter for dates, URLs, images, etc.
- Create new computed fields (runtime fields)



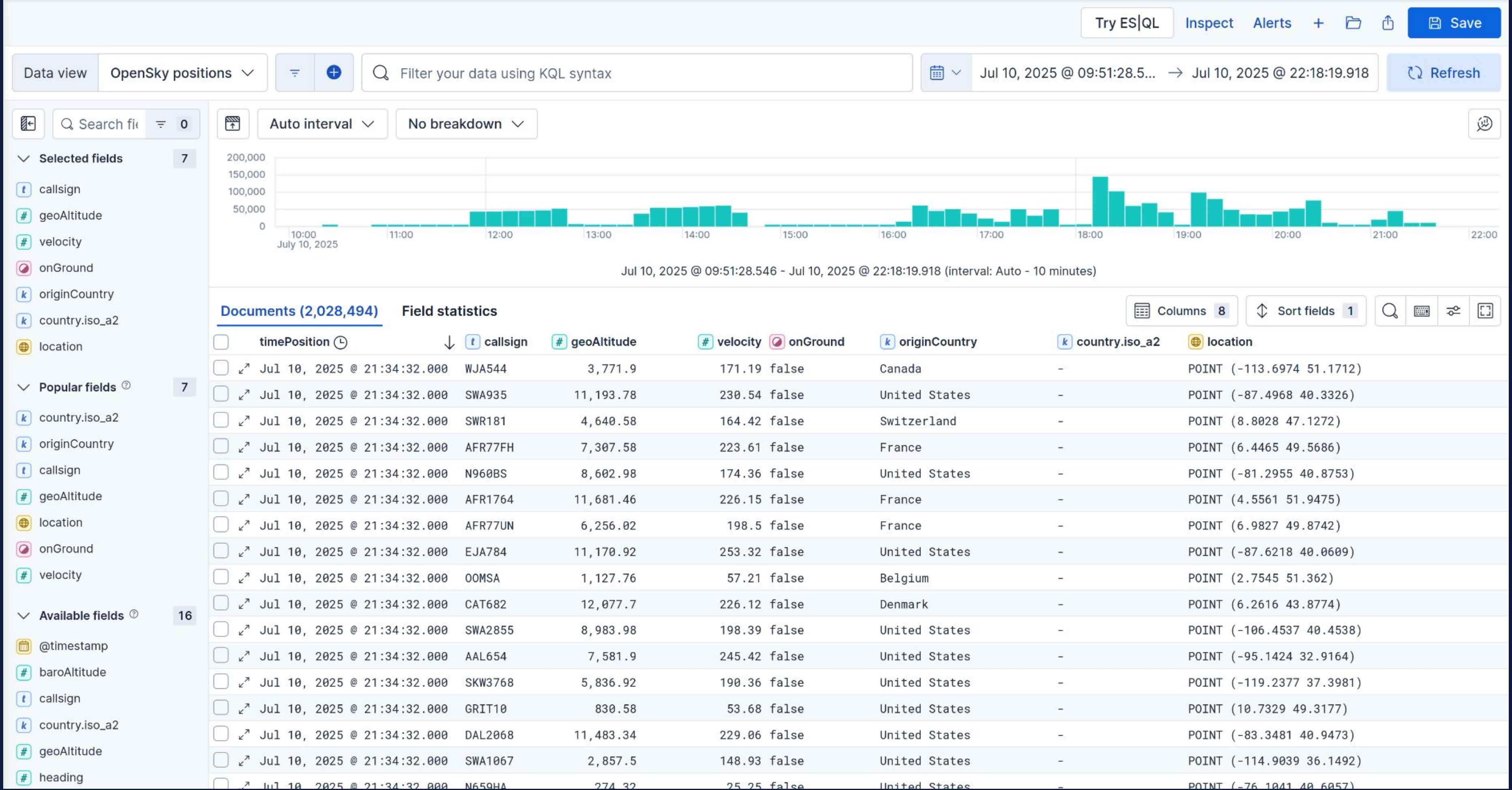
Welcome to Elastic

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.

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



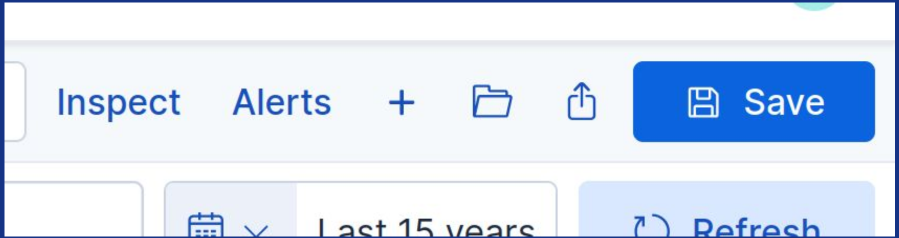
Jul 10, 2025 @ 09:51:28.546 - Jul 10, 2025 @ 22:18:19.918 (interval: Auto - 10 minutes)

timePosition	callsign	geoAltitude	velocity	onGround	originCountry	country.iso_a2	location
Jul 10, 2025 @ 21:34:32.000	WJA544	3,771.9	171.19	false	Canada	-	POINT (-113.6974 51.1712)
Jul 10, 2025 @ 21:34:32.000	SWA935	11,193.78	230.54	false	United States	-	POINT (-87.4968 40.3326)
Jul 10, 2025 @ 21:34:32.000	SWR181	4,640.58	164.42	false	Switzerland	-	POINT (8.8028 47.1272)
Jul 10, 2025 @ 21:34:32.000	AFR77FH	7,307.58	223.61	false	France	-	POINT (6.4465 49.5686)
Jul 10, 2025 @ 21:34:32.000	N960BS	8,602.98	174.36	false	United States	-	POINT (-81.2955 40.8753)
Jul 10, 2025 @ 21:34:32.000	AFR1764	11,681.46	226.15	false	France	-	POINT (4.5561 51.9475)
Jul 10, 2025 @ 21:34:32.000	AFR77UN	6,256.02	198.5	false	France	-	POINT (6.9827 49.8742)
Jul 10, 2025 @ 21:34:32.000	EJA784	11,170.92	253.32	false	United States	-	POINT (-87.6218 40.0609)
Jul 10, 2025 @ 21:34:32.000	OOMSA	1,127.76	57.21	false	Belgium	-	POINT (2.7545 51.362)
Jul 10, 2025 @ 21:34:32.000	CAT682	12,077.7	226.12	false	Denmark	-	POINT (6.2616 43.8774)
Jul 10, 2025 @ 21:34:32.000	SWA2855	8,983.98	198.39	false	United States	-	POINT (-106.4537 40.4538)
Jul 10, 2025 @ 21:34:32.000	AAL654	7,581.9	245.42	false	United States	-	POINT (-95.1424 32.9164)
Jul 10, 2025 @ 21:34:32.000	SKW3768	5,836.92	190.36	false	United States	-	POINT (-119.2377 37.3981)
Jul 10, 2025 @ 21:34:32.000	GRIT10	830.58	53.68	false	United States	-	POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000	DAL2068	11,483.34	229.06	false	United States	-	POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000	SWA1067	2,857.5	148.93	false	United States	-	POINT (-114.9039 36.1492)
Jul 10, 2025 @ 21:34:32.000	N659HA	274.32	25.25	false	United States	-	POINT (-76.1041 40.6057)

Search sessions

Persist your common search settings

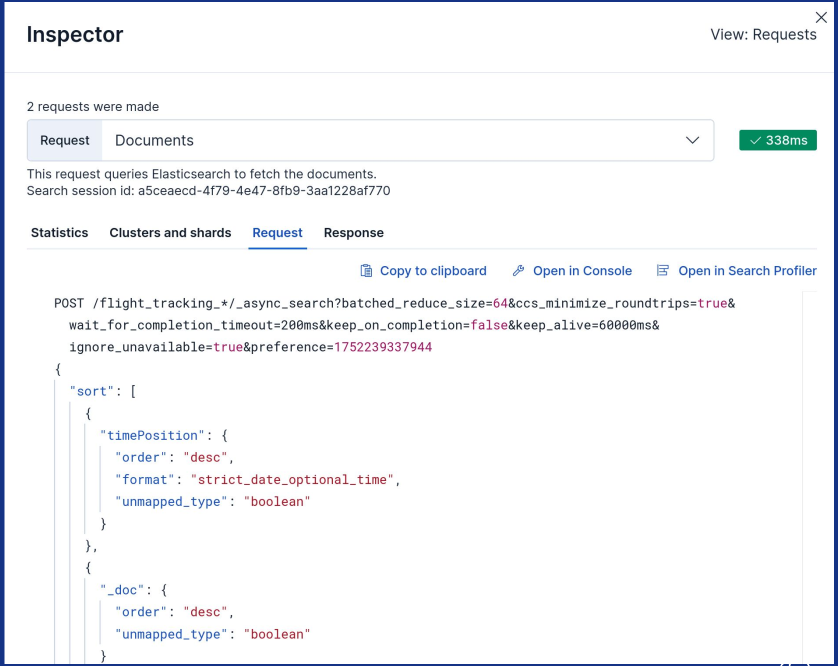
- Save and restore
- Can be added to dashboards
- Can be exported as links or CSV



Inspector

Get details of your queries to Elasticsearch

- Metadata about the query execution
- Request to Elasticsearch
- Response details

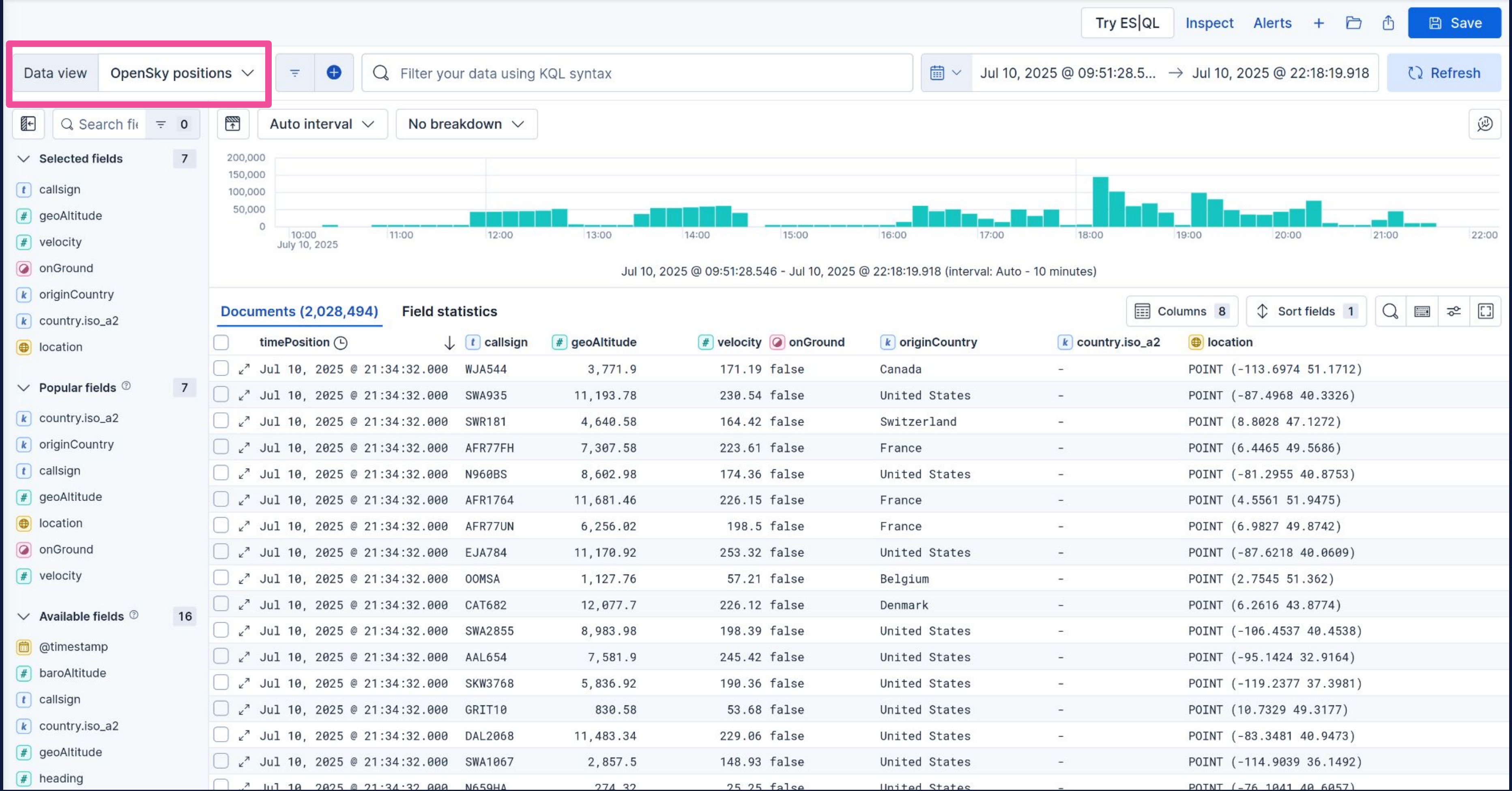


2 requests were made

Request: Documents ✓ 338ms

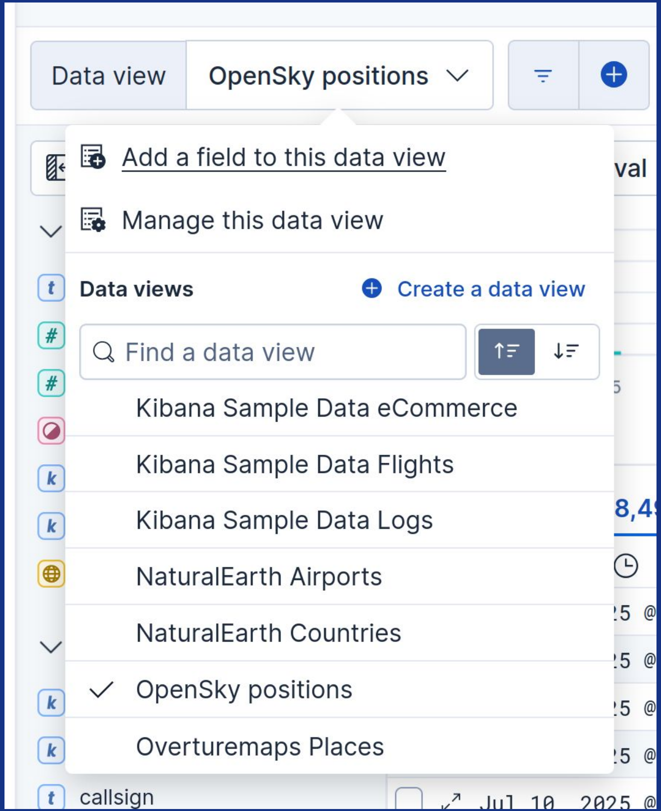
This request queries Elasticsearch to fetch the documents.
Search session id: a5ceaecd-4f79-4e47-8fb9-3aa1228af770

```
POST /flight_tracking_*/_async_search?batched_reduce_size=64&ccs_minimize_roundtrips=true&
  wait_for_completion_timeout=200ms&keep_on_completion=false&keep_alive=60000ms&
  ignore_unavailable=true&preference=1752239337944
{
  "sort": [
    {
      "timePosition": {
        "order": "desc",
        "format": "strict_date_optional_time",
        "unmapped_type": "boolean"
      }
    },
    {
      "_doc": {
        "order": "desc",
        "unmapped_type": "boolean"
      }
```



timePosition	callsign	geoAltitude	velocity	onGround	originCountry	country.iso_a2	location
Jul 10, 2025 @ 21:34:32.000	WJA544	3,771.9	171.19	false	Canada	-	POINT (-113.6974 51.1712)
Jul 10, 2025 @ 21:34:32.000	SWA935	11,193.78	230.54	false	United States	-	POINT (-87.4968 40.3326)
Jul 10, 2025 @ 21:34:32.000	SWR181	4,640.58	164.42	false	Switzerland	-	POINT (8.8028 47.1272)
Jul 10, 2025 @ 21:34:32.000	AFR77FH	7,307.58	223.61	false	France	-	POINT (6.4465 49.5686)
Jul 10, 2025 @ 21:34:32.000	N960BS	8,602.98	174.36	false	United States	-	POINT (-81.2955 40.8753)
Jul 10, 2025 @ 21:34:32.000	AFR1764	11,681.46	226.15	false	France	-	POINT (4.5561 51.9475)
Jul 10, 2025 @ 21:34:32.000	AFR77UN	6,256.02	198.5	false	France	-	POINT (6.9827 49.8742)
Jul 10, 2025 @ 21:34:32.000	EJA784	11,170.92	253.32	false	United States	-	POINT (-87.6218 40.0609)
Jul 10, 2025 @ 21:34:32.000	OOMSA	1,127.76	57.21	false	Belgium	-	POINT (2.7545 51.362)
Jul 10, 2025 @ 21:34:32.000	CAT682	12,077.7	226.12	false	Denmark	-	POINT (6.2616 43.8774)
Jul 10, 2025 @ 21:34:32.000	SWA2855	8,983.98	198.39	false	United States	-	POINT (-106.4537 40.4538)
Jul 10, 2025 @ 21:34:32.000	AAL654	7,581.9	245.42	false	United States	-	POINT (-95.1424 32.9164)
Jul 10, 2025 @ 21:34:32.000	SKW3768	5,836.92	190.36	false	United States	-	POINT (-119.2377 37.3981)
Jul 10, 2025 @ 21:34:32.000	GRIT10	830.58	53.68	false	United States	-	POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000	DAL2068	11,483.34	229.06	false	United States	-	POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000	SWA1067	2,857.5	148.93	false	United States	-	POINT (-114.9039 36.1492)
Jul 10, 2025 @ 21:34:32.000	N659HA	274.32	25.25	false	United States	-	POINT (-76.1041 40.6057)

Data View selector



Data view | OpenSky positions

Add a field to this data view

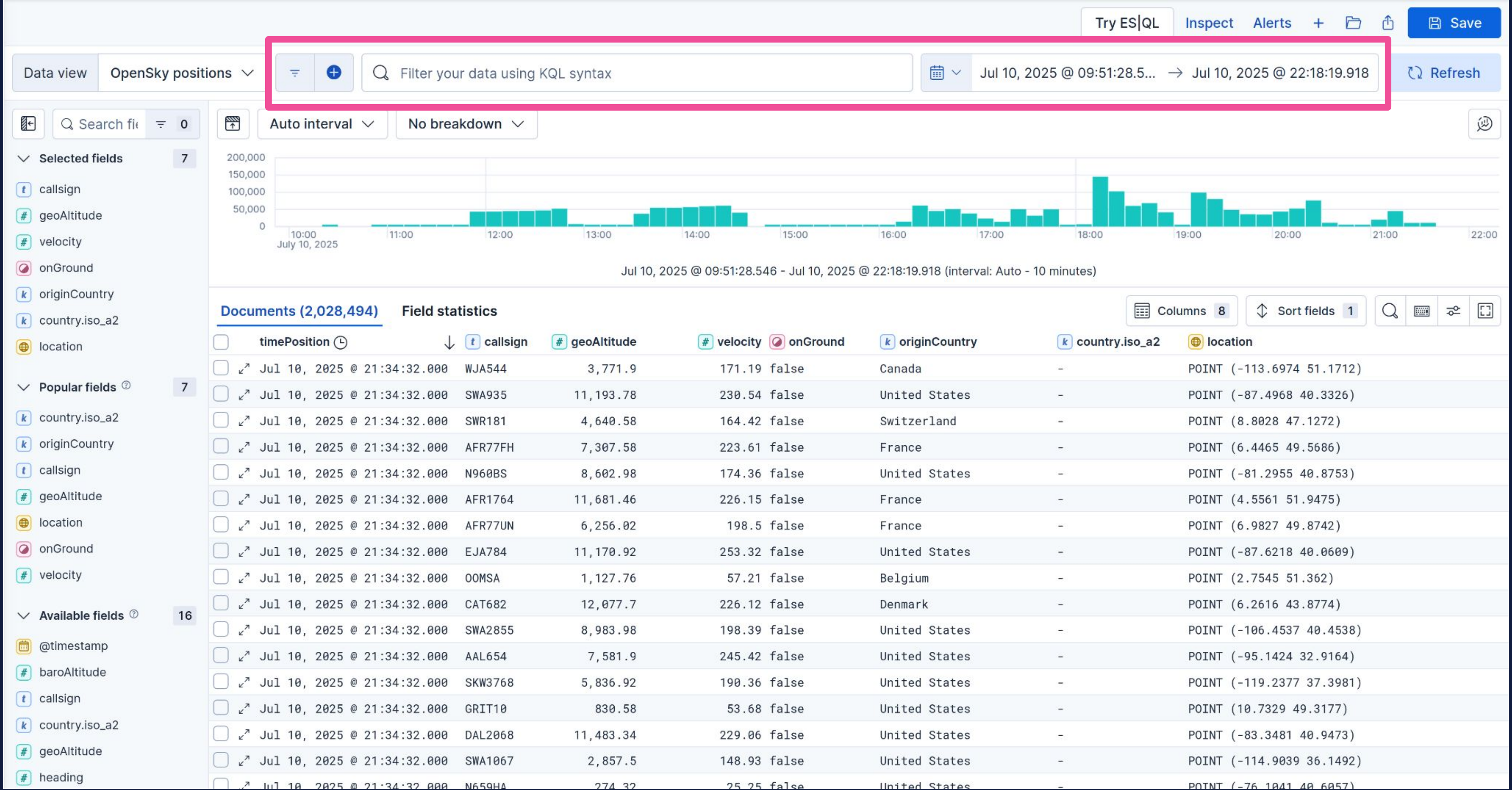
Manage this data view

Data views

Create a data view

Find a data view

- Kibana Sample Data eCommerce
- Kibana Sample Data Flights
- Kibana Sample Data Logs
- NaturalEarth Airports
- NaturalEarth Countries
- ✓ OpenSky positions
- Overturemaps Places



Jul 10, 2025 @ 09:51:28.546 - Jul 10, 2025 @ 22:18:19.918 (interval: Auto - 10 minutes)

Selected fields 7
- callsign
- geoAltitude
- velocity
- onGround
- originCountry
- country.iso_a2
- location

Popular fields 7
- country.iso_a2
- originCountry
- callsign
- geoAltitude
- location
- onGround
- velocity

Available fields 16
- @timestamp
- baroAltitude
- callsign
- country.iso_a2
- geoAltitude
- heading

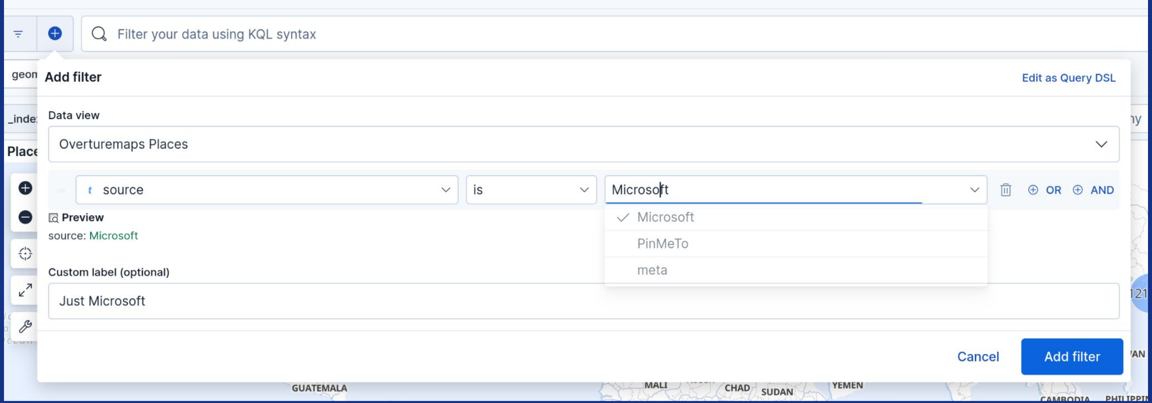
Documents (2,028,494) | Field statistics | Columns 8 | Sort fields 1

timePosition	callsign	geoAltitude	velocity	onGround	originCountry	country.iso_a2	location
Jul 10, 2025 @ 21:34:32.000	WJA544	3,771.9	171.19	false	Canada	-	POINT (-113.6974 51.1712)
Jul 10, 2025 @ 21:34:32.000	SWA935	11,193.78	230.54	false	United States	-	POINT (-87.4968 40.3326)
Jul 10, 2025 @ 21:34:32.000	SWR181	4,640.58	164.42	false	Switzerland	-	POINT (8.8028 47.1272)
Jul 10, 2025 @ 21:34:32.000	AFR77FH	7,307.58	223.61	false	France	-	POINT (6.4465 49.5686)
Jul 10, 2025 @ 21:34:32.000	N960BS	8,602.98	174.36	false	United States	-	POINT (-81.2955 40.8753)
Jul 10, 2025 @ 21:34:32.000	AFR1764	11,681.46	226.15	false	France	-	POINT (4.5561 51.9475)
Jul 10, 2025 @ 21:34:32.000	AFR77UN	6,256.02	198.5	false	France	-	POINT (6.9827 49.8742)
Jul 10, 2025 @ 21:34:32.000	EJA784	11,170.92	253.32	false	United States	-	POINT (-87.6218 40.0609)
Jul 10, 2025 @ 21:34:32.000	OOMSA	1,127.76	57.21	false	Belgium	-	POINT (2.7545 51.362)
Jul 10, 2025 @ 21:34:32.000	CAT682	12,077.7	226.12	false	Denmark	-	POINT (6.2616 43.8774)
Jul 10, 2025 @ 21:34:32.000	SWA2855	8,983.98	198.39	false	United States	-	POINT (-106.4537 40.4538)
Jul 10, 2025 @ 21:34:32.000	AAL654	7,581.9	245.42	false	United States	-	POINT (-95.1424 32.9164)
Jul 10, 2025 @ 21:34:32.000	SKW3768	5,836.92	190.36	false	United States	-	POINT (-119.2377 37.3981)
Jul 10, 2025 @ 21:34:32.000	GRIT10	830.58	53.68	false	United States	-	POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000	DAL2068	11,483.34	229.06	false	United States	-	POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000	SWA1067	2,857.5	148.93	false	United States	-	POINT (-114.9039 36.1492)
Jul 10, 2025 @ 21:34:32.000	N659HA	274.32	25.25	false	United States	-	POINT (-76.1041 40.6057)

Filters

Versatile pills for filtering

- Easy filter creation, but DSL also available
- Custom label
- Transferred across dashboards and applications
- Also available on view mode



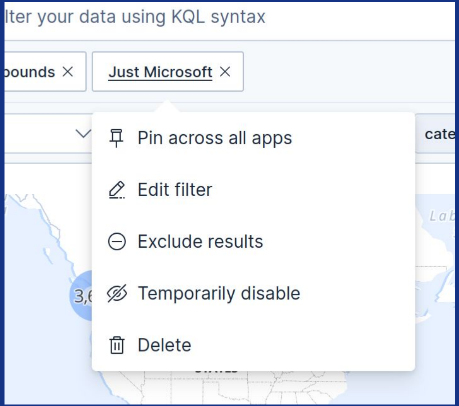
Add filter

Data view: Overturemaps Places

source is Microsoft

Preview
source: Microsoft

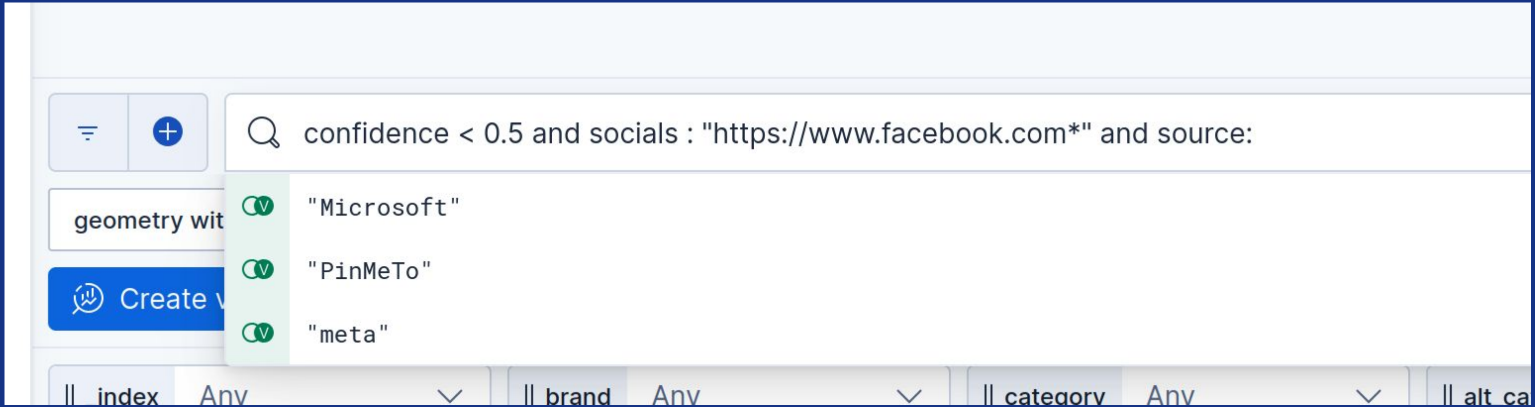
Custom label (optional): Just Microsoft



Query bar

Advanced ad-hoc queries

- Kibana Query Language
- Autocomplete for fields and values
- Can be saved in the dashboard definition and used in view mode

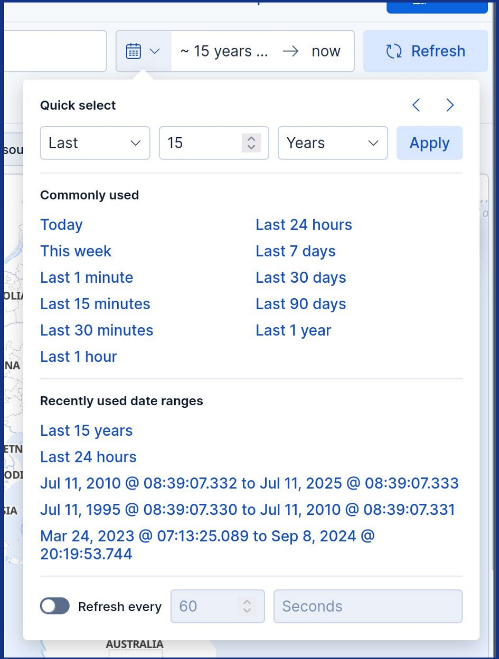


confidence < 0.5 and socials : "https://www.facebook.com*" and source:

```
"Microsoft"
"PinMeTo"
"meta"
```

Time picker

- Flexible time range selector with quick, absolute and relative selections.
- Auto-refresh



~ 15 years ... → now

Refresh

Quick select

Last | 15 | Years | Apply

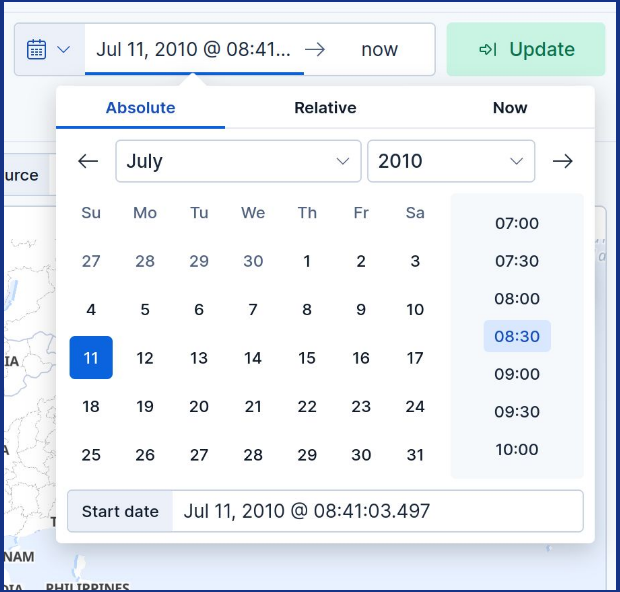
Commonly used

Today
This week
Last 1 minute
Last 15 minutes
Last 30 minutes
Last 1 hour
Last 24 hours
Last 7 days
Last 30 days
Last 90 days
Last 1 year

Recently used date ranges

Last 15 years
Last 24 hours
Jul 11, 2010 @ 08:39:07.332 to Jul 11, 2025 @ 08:39:07.333
Jul 11, 1995 @ 08:39:07.330 to Jul 11, 2010 @ 08:39:07.331
Mar 24, 2023 @ 07:13:25.089 to Sep 8, 2024 @ 20:19:53.744

Refresh every 60 Seconds



Jul 11, 2010 @ 08:41... → now

Update

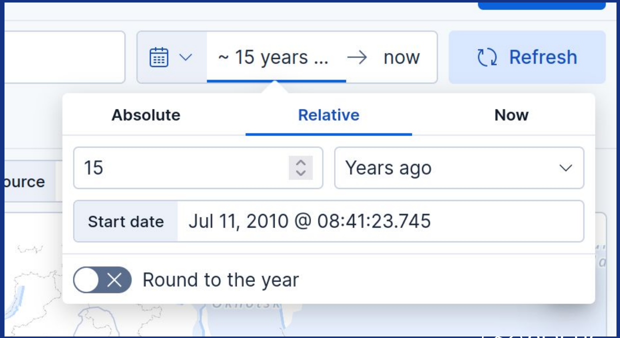
Absolute | Relative | Now

← July | 2010 →

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

07:00
07:30
08:00
08:30
09:00
09:30
10:00

Start date: Jul 11, 2010 @ 08:41:03.497



~ 15 years ... → now

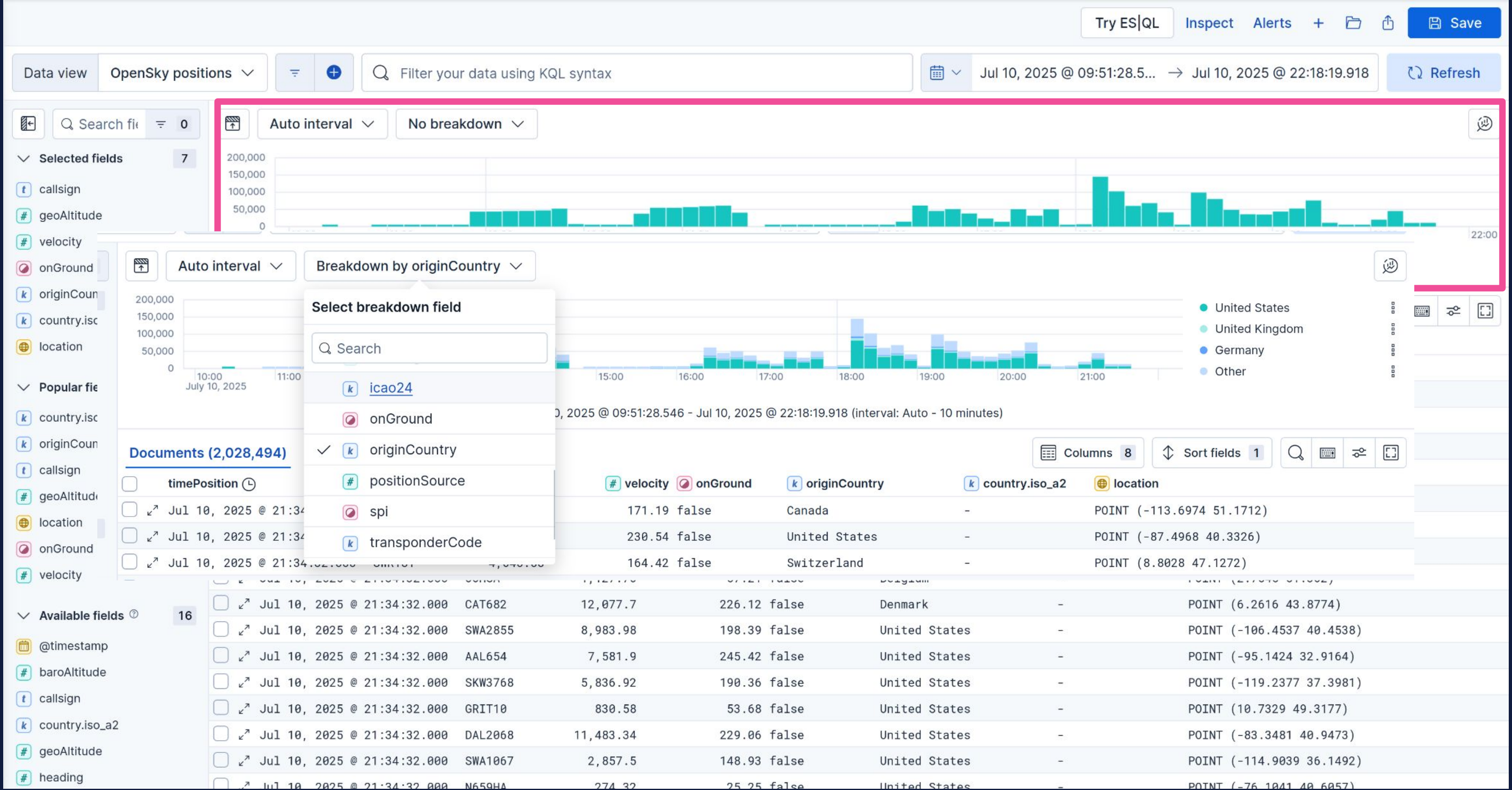
Refresh

Absolute | Relative | Now

15 | Years ago

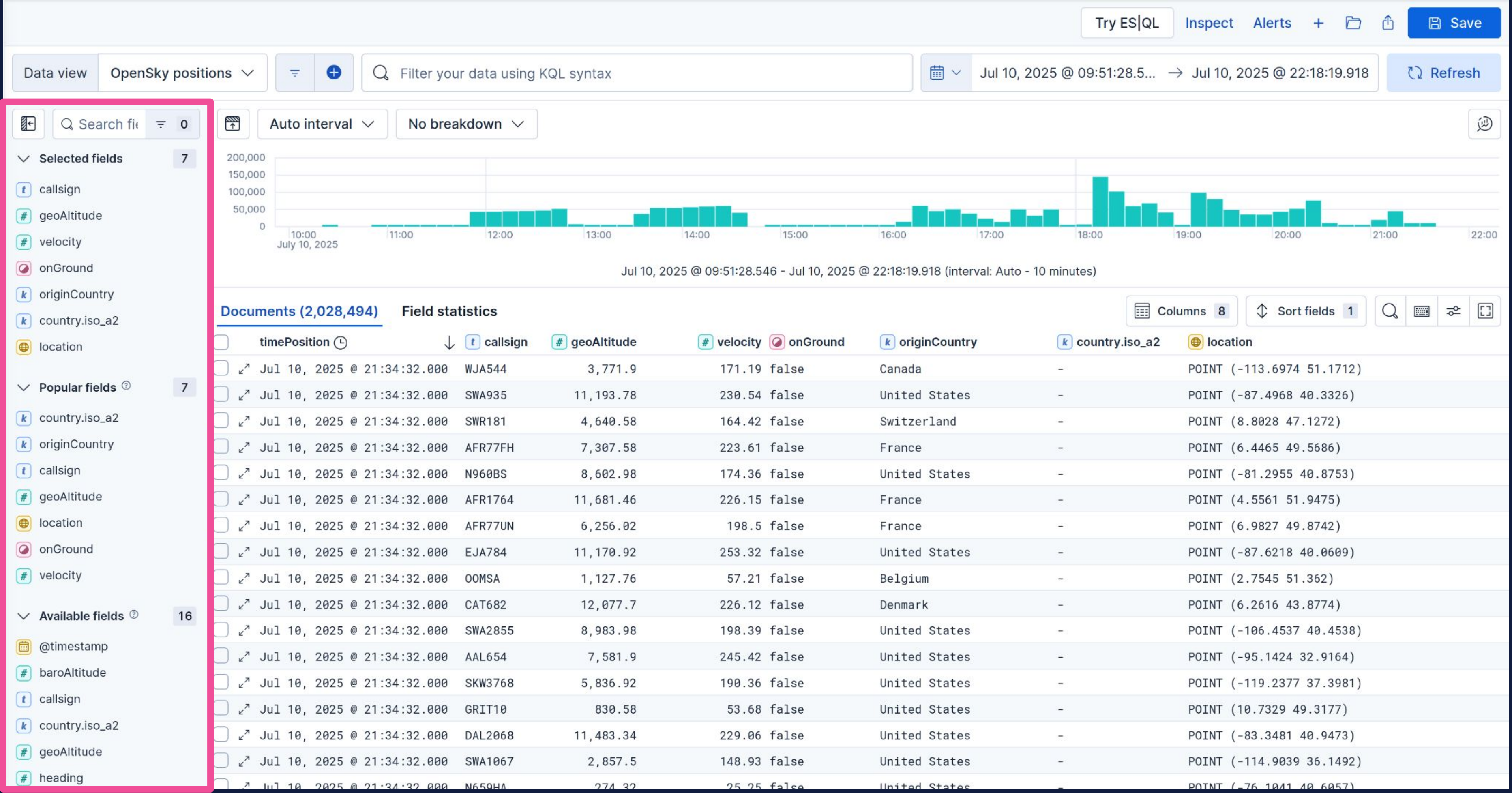
Start date: Jul 11, 2010 @ 08:41:23.745

Round to the year



0, 2025 @ 09:51:28.546 - Jul 10, 2025 @ 22:18:19.918 (interval: Auto - 10 minutes)

timePosition			velocity	onGround	originCountry	country.iso_a2	location
Jul 10, 2025 @ 21:34			171.19	false	Canada	-	POINT (-113.6974 51.1712)
Jul 10, 2025 @ 21:34			230.54	false	United States	-	POINT (-87.4968 40.3326)
Jul 10, 2025 @ 21:34			164.42	false	Switzerland	-	POINT (8.8028 47.1272)
Jul 10, 2025 @ 21:34:32.000	CAT682	12,077.7	226.12	false	Denmark	-	POINT (6.2616 43.8774)
Jul 10, 2025 @ 21:34:32.000	SWA2855	8,983.98	198.39	false	United States	-	POINT (-106.4537 40.4538)
Jul 10, 2025 @ 21:34:32.000	AAL654	7,581.9	245.42	false	United States	-	POINT (-95.1424 32.9164)
Jul 10, 2025 @ 21:34:32.000	SKW3768	5,836.92	190.36	false	United States	-	POINT (-119.2377 37.3981)
Jul 10, 2025 @ 21:34:32.000	GRIT10	830.58	53.68	false	United States	-	POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000	DAL2068	11,483.34	229.06	false	United States	-	POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000	SWA1067	2,857.5	148.93	false	United States	-	POINT (-114.9039 36.1492)
Jul 10, 2025 @ 21:34:32.000	N659HA	274.32	25.25	false	United States	-	POINT (-76.1041 40.6057)



Jul 10, 2025 @ 09:51:28.546 - Jul 10, 2025 @ 22:18:19.918 (interval: Auto - 10 minutes)

Selected fields 7

- callsign
- geoAltitude
- velocity
- onGround
- originCountry
- country.iso_a2
- location

Popular fields 7

- country.iso_a2
- originCountry
- callsign
- geoAltitude
- location
- onGround
- velocity

Available fields 16

- @timestamp
- baroAltitude
- callsign
- country.iso_a2
- geoAltitude
- heading

timePosition	callsign	geoAltitude	velocity	onGround	originCountry	country.iso_a2	location
Jul 10, 2025 @ 21:34:32.000	WJA544	3,771.9	171.19	false	Canada	-	POINT (-113.6974 51.1712)
Jul 10, 2025 @ 21:34:32.000	SWA935	11,193.78	230.54	false	United States	-	POINT (-87.4968 40.3326)
Jul 10, 2025 @ 21:34:32.000	SWR181	4,640.58	164.42	false	Switzerland	-	POINT (8.8028 47.1272)
Jul 10, 2025 @ 21:34:32.000	AFR77FH	7,307.58	223.61	false	France	-	POINT (6.4465 49.5686)
Jul 10, 2025 @ 21:34:32.000	N960BS	8,602.98	174.36	false	United States	-	POINT (-81.2955 40.8753)
Jul 10, 2025 @ 21:34:32.000	AFR1764	11,681.46	226.15	false	France	-	POINT (4.5561 51.9475)
Jul 10, 2025 @ 21:34:32.000	AFR77UN	6,256.02	198.5	false	France	-	POINT (6.9827 49.8742)
Jul 10, 2025 @ 21:34:32.000	EJA784	11,170.92	253.32	false	United States	-	POINT (-87.6218 40.0609)
Jul 10, 2025 @ 21:34:32.000	OOMSA	1,127.76	57.21	false	Belgium	-	POINT (2.7545 51.362)
Jul 10, 2025 @ 21:34:32.000	CAT682	12,077.7	226.12	false	Denmark	-	POINT (6.2616 43.8774)
Jul 10, 2025 @ 21:34:32.000	SWA2855	8,983.98	198.39	false	United States	-	POINT (-106.4537 40.4538)
Jul 10, 2025 @ 21:34:32.000	AAL654	7,581.9	245.42	false	United States	-	POINT (-95.1424 32.9164)
Jul 10, 2025 @ 21:34:32.000	SKW3768	5,836.92	190.36	false	United States	-	POINT (-119.2377 37.3981)
Jul 10, 2025 @ 21:34:32.000	GRIT10	830.58	53.68	false	United States	-	POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000	DAL2068	11,483.34	229.06	false	United States	-	POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000	SWA1067	2,857.5	148.93	false	United States	-	POINT (-114.9039 36.1492)
Jul 10, 2025 @ 21:34:32.000	N659HA	274.32	25.25	false	United States	-	POINT (-76.1041 40.6057)

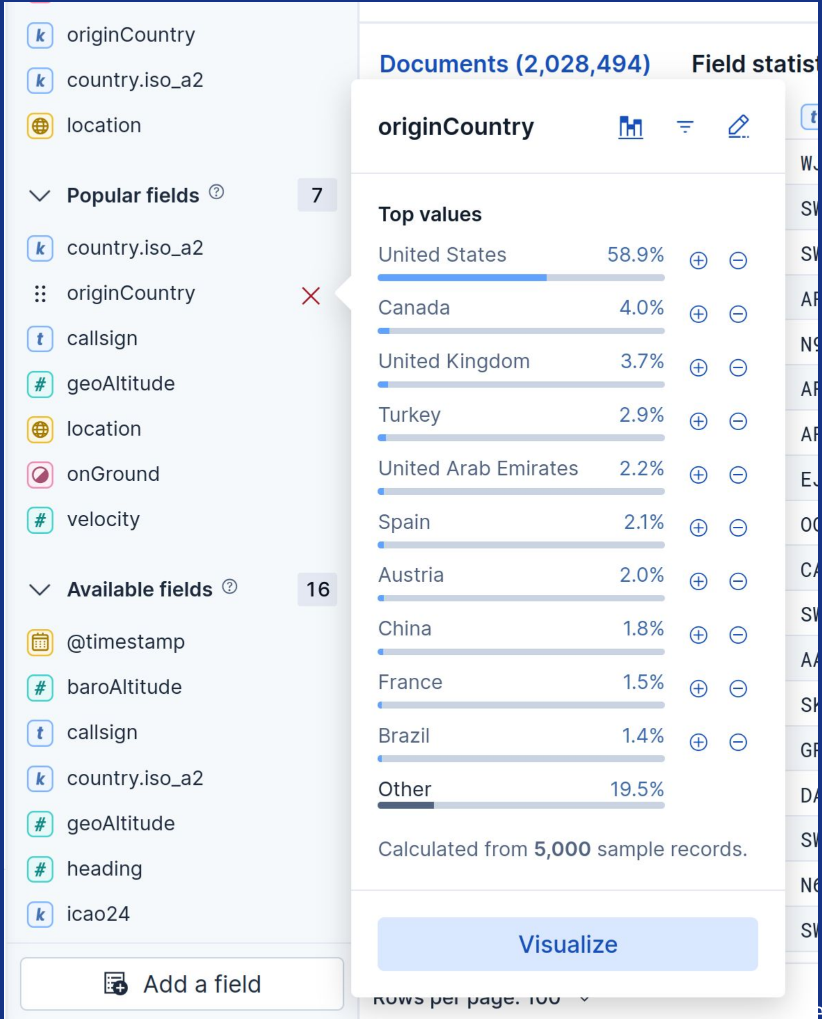
Selecting fields

Click on a field to reveal basic information

- Filter in/out for any of the top values
- Visualize: jump to chart authoring for this field

In the quick actions

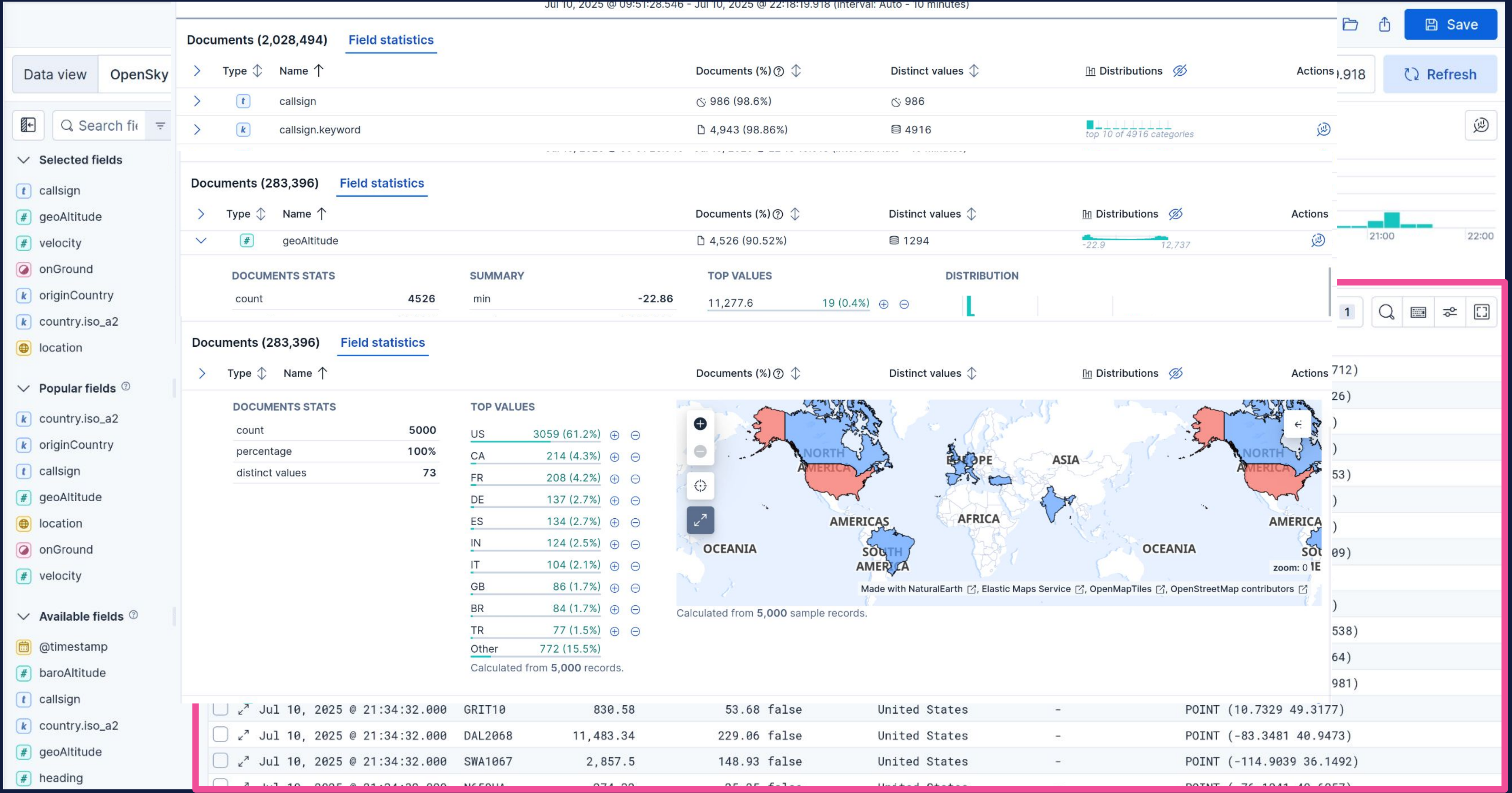
- Create a filter for this field to be present
- Add the field to the histogram breakdown
- Edit the Data View field



Top values

United States	58.9%
Canada	4.0%
United Kingdom	3.7%
Turkey	2.9%
United Arab Emirates	2.2%
Spain	2.1%
Austria	2.0%
China	1.8%
France	1.5%
Brazil	1.4%
Other	19.5%

Calculated from **5,000** sample records.



Jul 10, 2025 @ 09:51:28.546 - Jul 10, 2025 @ 22:18:19.918 (interval: Auto - 10 minutes)

Selected fields
- callsign
- geoAltitude
- velocity
- onGround
- originCountry
- country.iso_a2
- location

Popular fields
- country.iso_a2
- originCountry
- callsign
- geoAltitude
- location
- onGround
- velocity

Available fields
- @timestamp
- baroAltitude
- callsign
- country.iso_a2
- geoAltitude
- heading

Documents (2,028,494) | Field statistics

Type	Name	Documents (%)	Distinct values	Distributions	Actions
t	callsign	986 (98.6%)	986		
k	callsign.keyword	4,943 (98.86%)	4916	top 10 of 4916 categories	

Documents (283,396) | Field statistics

Type	Name	Documents (%)	Distinct values	Distributions	Actions
#	geoAltitude	4,526 (90.52%)	1294	-22.9 – 12,737	

DOCUMENTS STATS

count	4526

SUMMARY

min	-22.86

TOP VALUES

11,277.6	19 (0.4%)

DISTRIBUTION

Documents (283,396) | Field statistics

Type	Name	Documents (%)	Distinct values	Distributions	Actions

DOCUMENTS STATS

count	5000
percentage	100%
distinct values	73

TOP VALUES

US	3059 (61.2%)
CA	214 (4.3%)
FR	208 (4.2%)
DE	137 (2.7%)
ES	134 (2.7%)
IN	124 (2.5%)
IT	104 (2.1%)
GB	86 (1.7%)
BR	84 (1.7%)
TR	77 (1.5%)
Other	772 (15.5%)

Calculated from **5,000** records.

Made with NaturalEarth, Elastic Maps Service, OpenMapTiles, OpenStreetMap contributors

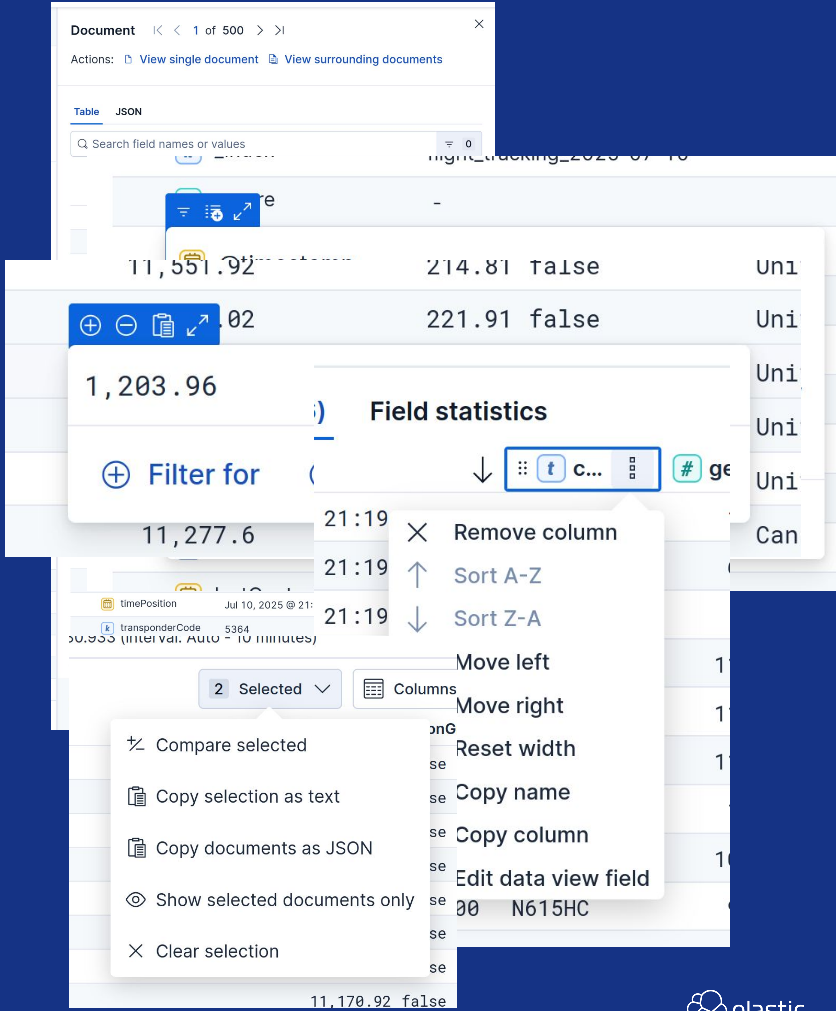
Calculated from **5,000** sample records.

```
Jul 10, 2025 @ 21:34:32.000  GRIT10       830.58   53.68 false  United States  -  POINT (10.7329 49.3177)
Jul 10, 2025 @ 21:34:32.000  DAL2068   11,483.34  229.06 false  United States  -  POINT (-83.3481 40.9473)
Jul 10, 2025 @ 21:34:32.000  SWA1067     2,857.5  148.93 false  United States  -  POINT (-114.9039 36.1492)
```

Documents table

View everything about each indexed document

- Toggle document viewer (table or raw JSON)
  - Quick actions on field names
- Click on any value to filter in/out
- Click on any column header to sort/shift
- Select documents to compare or copy them

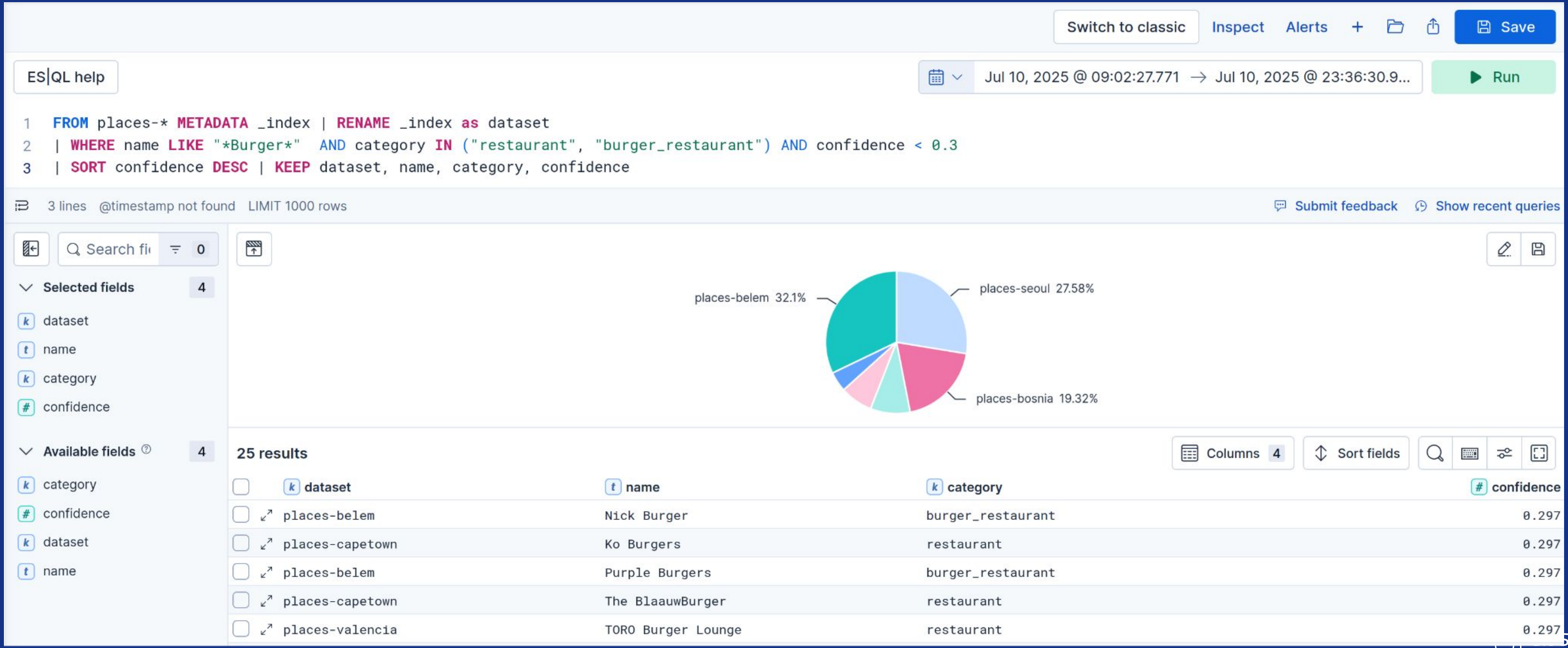


elastic

Discover & ES|QL

Discover & ES|QL

Rich editor for ES|QL replacing Data Views for data exploration and manipulation



```
FROM places-* METADATA _index | RENAME _index as dataset
| WHERE name LIKE "*Burger*"  AND category IN ("restaurant", "burger_restaurant") AND confidence < 0.3
| SORT confidence DESC | KEEP dataset, name, category, confidence
```

3 lines @timestamp not found LIMIT 1000 rows

Selected fields 4

- dataset
- name
- category
- confidence

Available fields 4

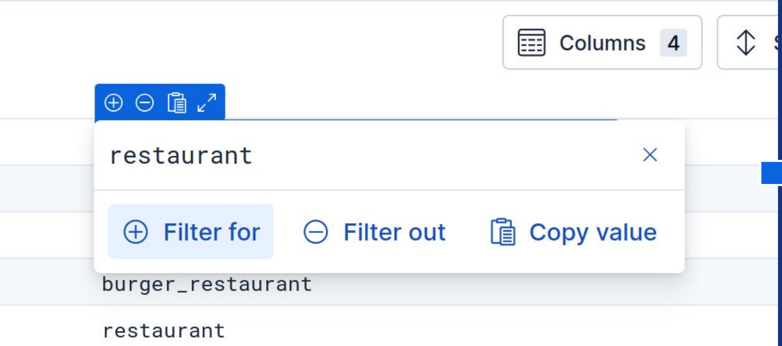
- category
- confidence
- dataset
- name

25 results

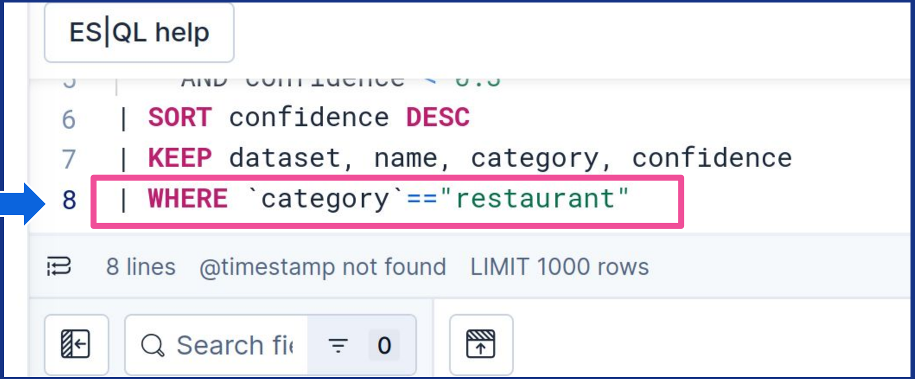
dataset	name	category	confidence
places-belem	Nick Burger	burger_restaurant	0.297
places-capetown	Ko Burgers	restaurant	0.297
places-belem	Purple Burgers	burger_restaurant	0.297
places-capetown	The BlaauwBurger	restaurant	0.297
places-valencia	TORO Burger Lounge	restaurant	0.297

Discover & ES|QL

Interactions in fields and values are translated into new query piped commands



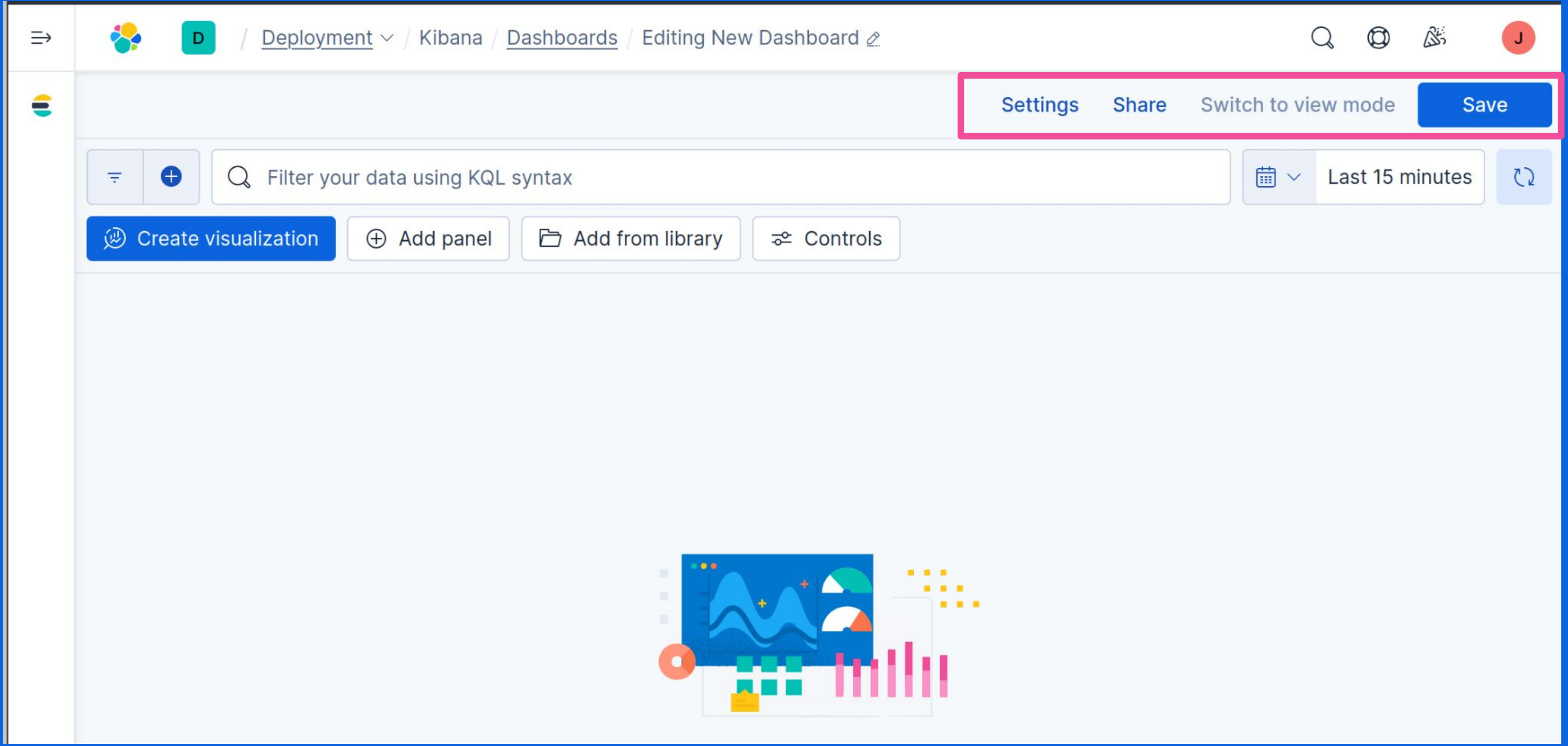
```
burger_restaurant
restaurant
```



```
    AND confidence < 0.5
  | SORT confidence DESC
  | KEEP dataset, name, category, confidence
  | WHERE `category`=="restaurant"
```

8 lines @timestamp not found LIMIT 1000 rows

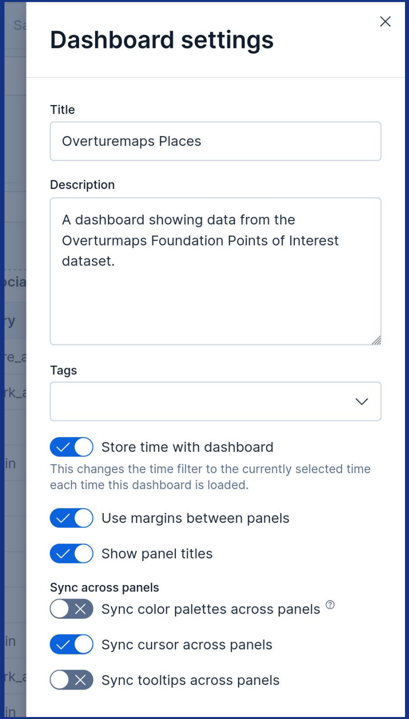
Dashboards



D
Deployment
Kibana
Dashboards
Editing New Dashboard
J
Settings
Share
Switch to view mode
Save
Filter your data using KQL syntax
Last 15 minutes
Create visualization
Add panel
Add from library
Controls

Settings

Metadata and general appearance



Dashboard settings

Title

Overturemaps Places

Description

A dashboard showing data from the Overturmaps Foundation Points of Interest dataset.

Tags

☑ Store time with dashboard

This changes the time filter to the currently selected time each time this dashboard is loaded.

☑ Use margins between panels

☑ Show panel titles

Sync across panels

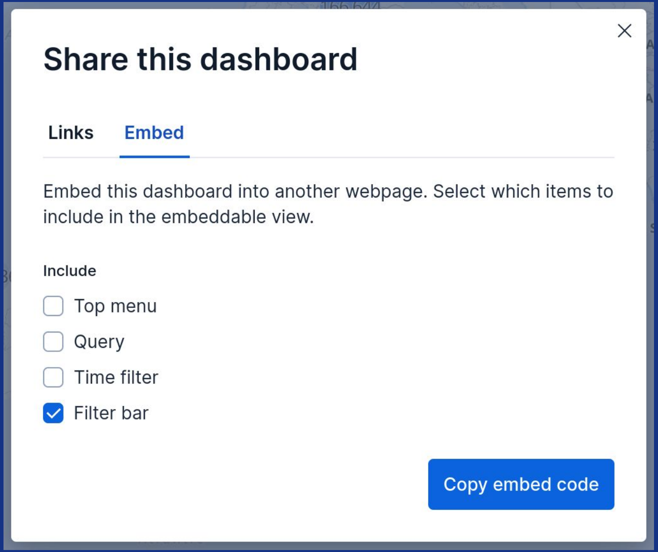
☐ Sync color palettes across panels

☑ Sync cursor across panels

☐ Sync tooltips across panels

Share

Generate links to your dashboard
or get the embed code (iframe)



Share this dashboard

Links | Embed

Embed this dashboard into another webpage. Select which items to include in the embeddable view.

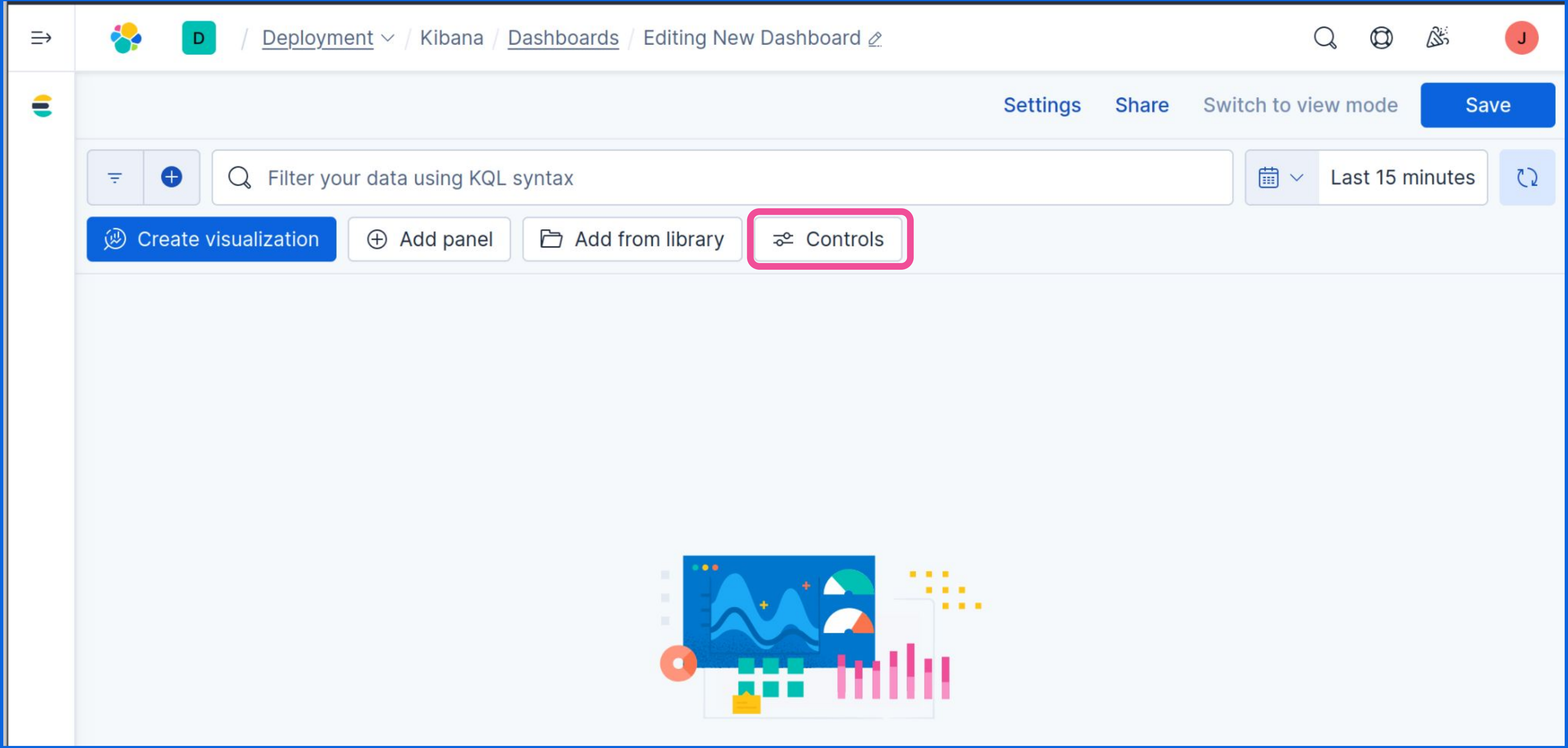
Include

☐ Top menu

☐ Query

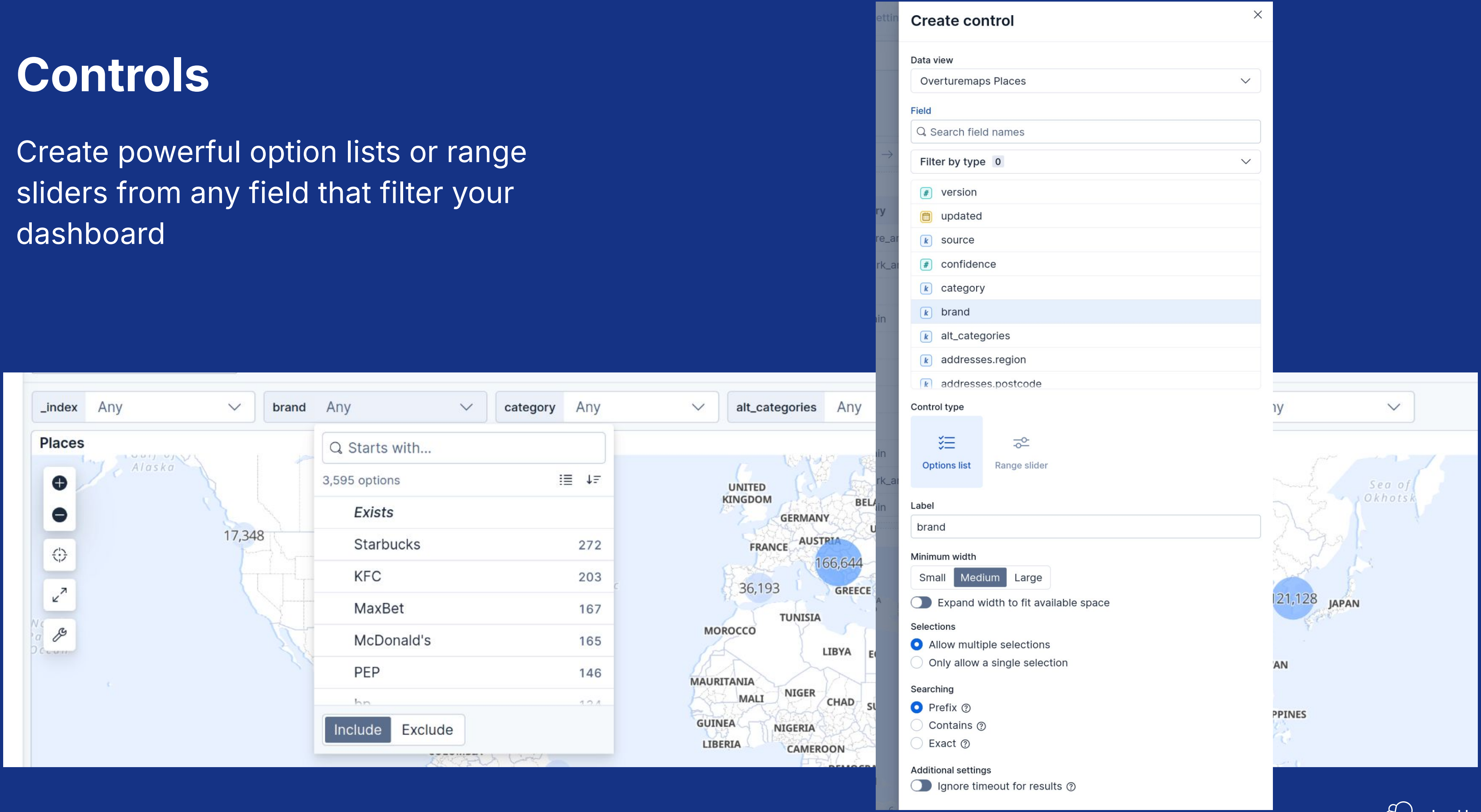
☐ Time filter

☑ Filter bar


D
Deployment
Kibana
Dashboards
Editing New Dashboard
J
Settings
Share
Switch to view mode
Save
Filter your data using KQL syntax
Last 15 minutes
Create visualization
Add panel
Add from library
Controls

Controls

Create powerful option lists or range sliders from any field that filter your dashboard



_index Any

brand Any

category Any

alt_categories Any

Places

Starts with...

3,595 options

Exists	
Starbucks	272
KFC	203
MaxBet	167
McDonald's	165
PEP	146

Include Exclude

Create control

Data view: Overturemaps Places

Field: Search field names

Filter by type 0

- version
- updated
- source
- confidence
- category
- brand
- alt_categories
- addresses.region
- addresses.postcode

Control type: Options list | Range slider

Label: brand

Minimum width: Small | **Medium** | Large

Expand width to fit available space

Selections

- ● Allow multiple selections
- ○ Only allow a single selection

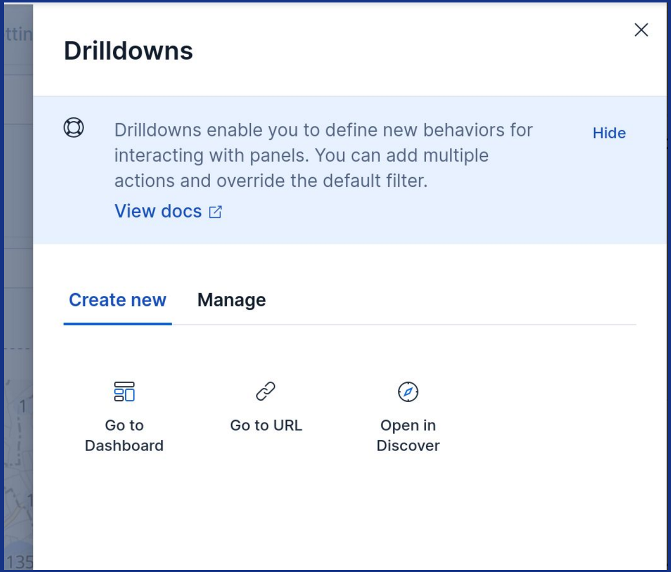
Searching

- ● Prefix
- ○ Contains
- ○ Exact

Additional settings

Ignore timeout for results

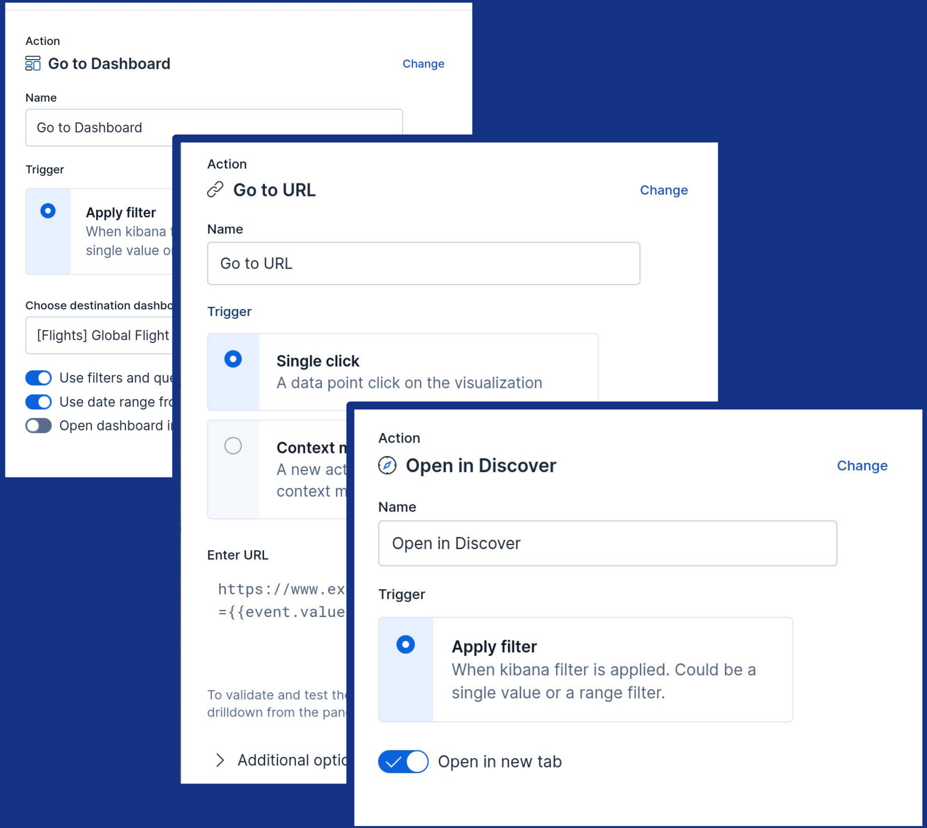
Drilldowns



Drilldowns

Drilldowns enable you to define new behaviors for interacting with panels. You can add multiple actions and override the default filter.

View docs



Action
Go to Dashboard
Name: Go to Dashboard
Trigger
● Apply filter
When kibana f
single value o
Choose destination dashbo
[Flights] Global Flight
Use filters and que
Use date range fro
Open dashboard i

Action
Go to URL
Name: Go to URL
Trigger
● Single click
A data point click on the visualization
○ Context m
A new act
context m
Enter URL

```
https://www.ex
={{event.value
```

To validate and test th
drilldown from the pan

Additional optio

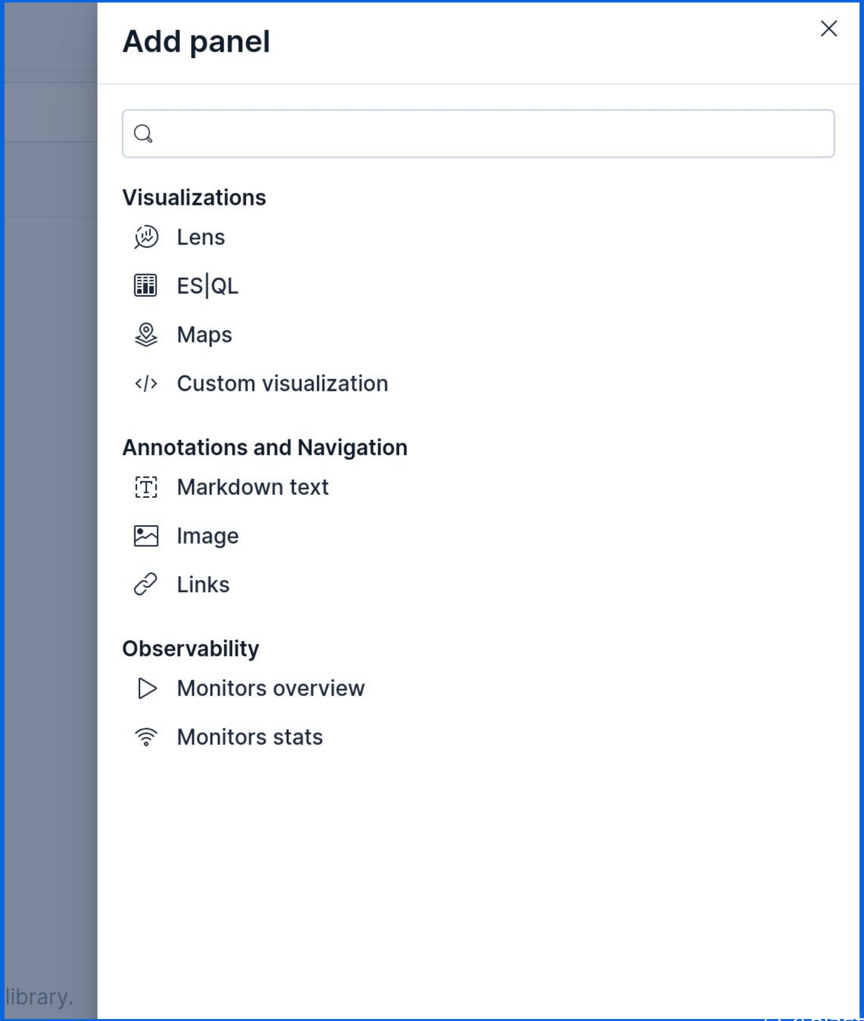
Action
Open in Discover
Name: Open in Discover
Trigger
● Apply filter
When kibana filter is applied. Could be a single value or a range filter.
Open in new tab

elastic

Dashboard panels

Visualizations

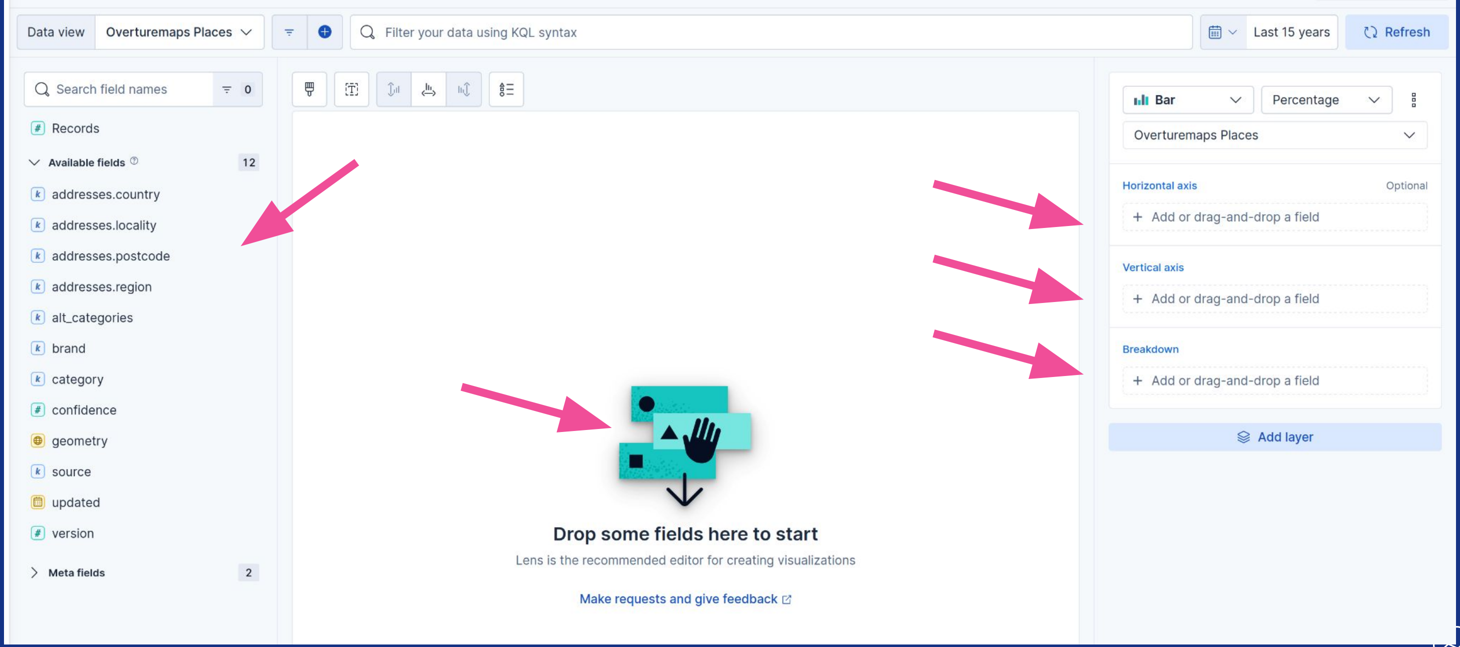
- Lens: drag & drop visualization builder
- ES|QL: create visualizations from queries
- Maps: geospatial visualizations
- Custom visualization: use Vega JSON specifications to create advanced visualizations



Lens

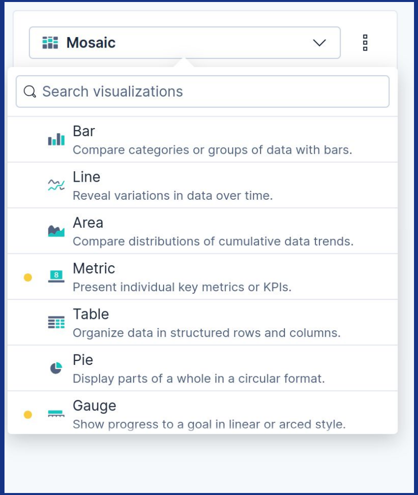
Lens

Drag & drop fields into the main area, axis, and breakdown selectors



Lens

Broad selection of chart types: table, area/bar/line chart, metrics, treemap, waffle, gauge



Bar
Compare categories or groups of data with bars.

Line
Reveal variations in data over time.

Area
Compare distributions of cumulative data trends.

Metric
Present individual key metrics or KPIs.

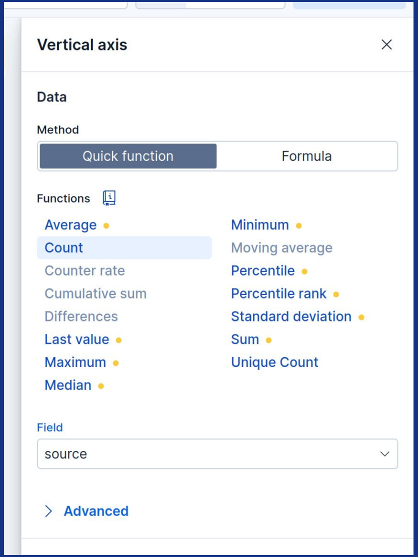
Table
Organize data in structured rows and columns.

Pie
Display parts of a whole in a circular format.

Gauge
Show progress to a goal in linear or arced style.

Lens

Easy metric aggregation selection & custom formula with in-product help



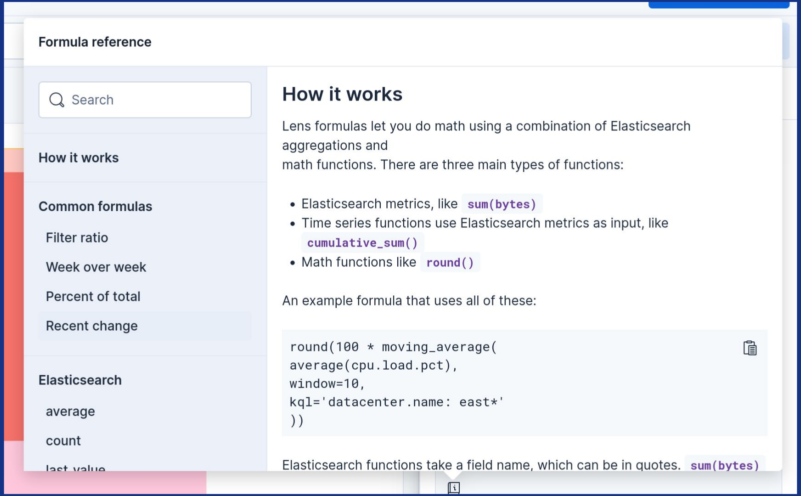
Vertical axis

Data

Method

Functions

Field: source



Formula reference

How it works

Lens formulas let you do math using a combination of Elasticsearch aggregations and
math functions. There are three main types of functions:

- Elasticsearch metrics, like `sum(bytes)`
- Time series functions use Elasticsearch metrics as input, like `cumulative_sum()`
- Math functions like `round()`

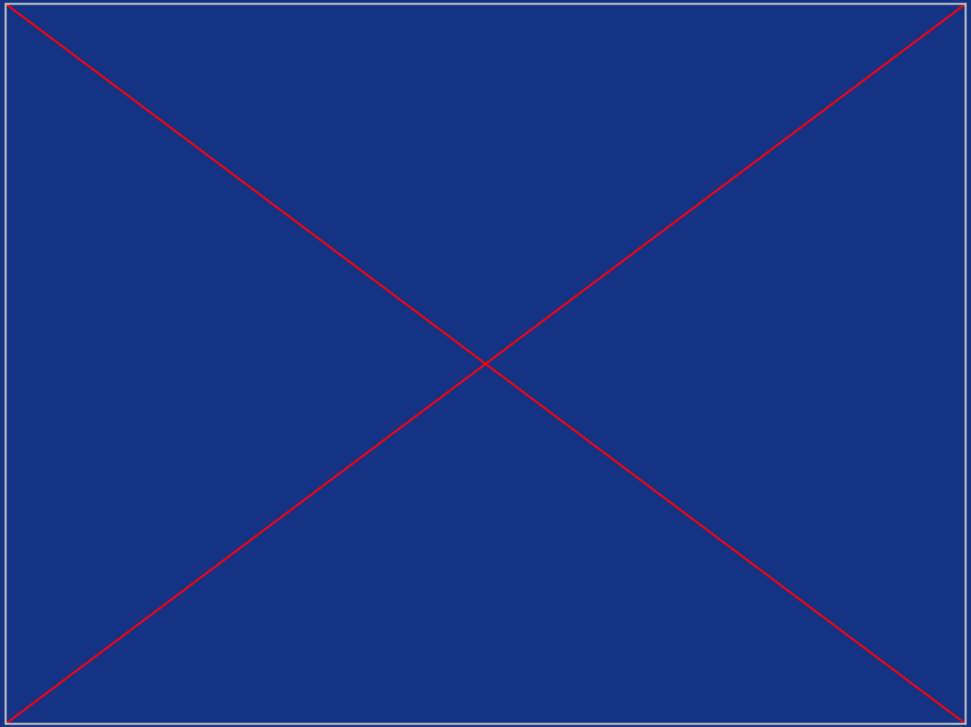
An example formula that uses all of these:

```
round(100 * moving_average(
average(cpu.load.pct),
window=10,
kql='datacenter.name: east*'
))
```

Elasticsearch functions take a field name, which can be in quotes. `sum(bytes)`

Lens

Lens visualizations can create filter pills interactively when brushing or clicking on chart elements

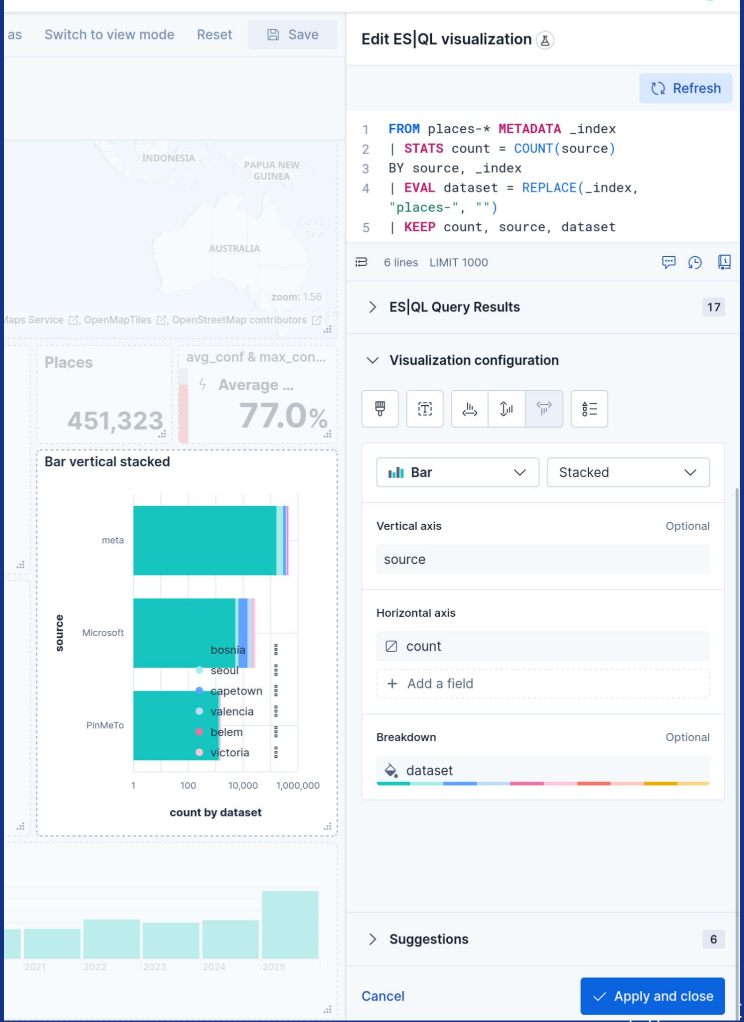


ES|QL visualizations

ES|QL visualizations

From queries to charts

- Create chart without leaving the dashboard
- Complete ES|QL editor with autocomplete, error highlighting, etc.
- Review query results
- Define the visualization with a lens-like interface
    - Chart type, visualization settings, axis, etc
    - Vertical and horizontal axis metrics
    - Optional breakdown
- In future releases:
    - Use variables in the query to create controls that allow interactive visualizations.



```
FROM places-* METADATA _index
| STATS count = COUNT(source)
BY source, _index
| EVAL dataset = REPLACE(_index,
"places-", "")
| KEEP count, source, dataset
```

Bar — Stacked

Vertical axis (Optional): source

Horizontal axis: count

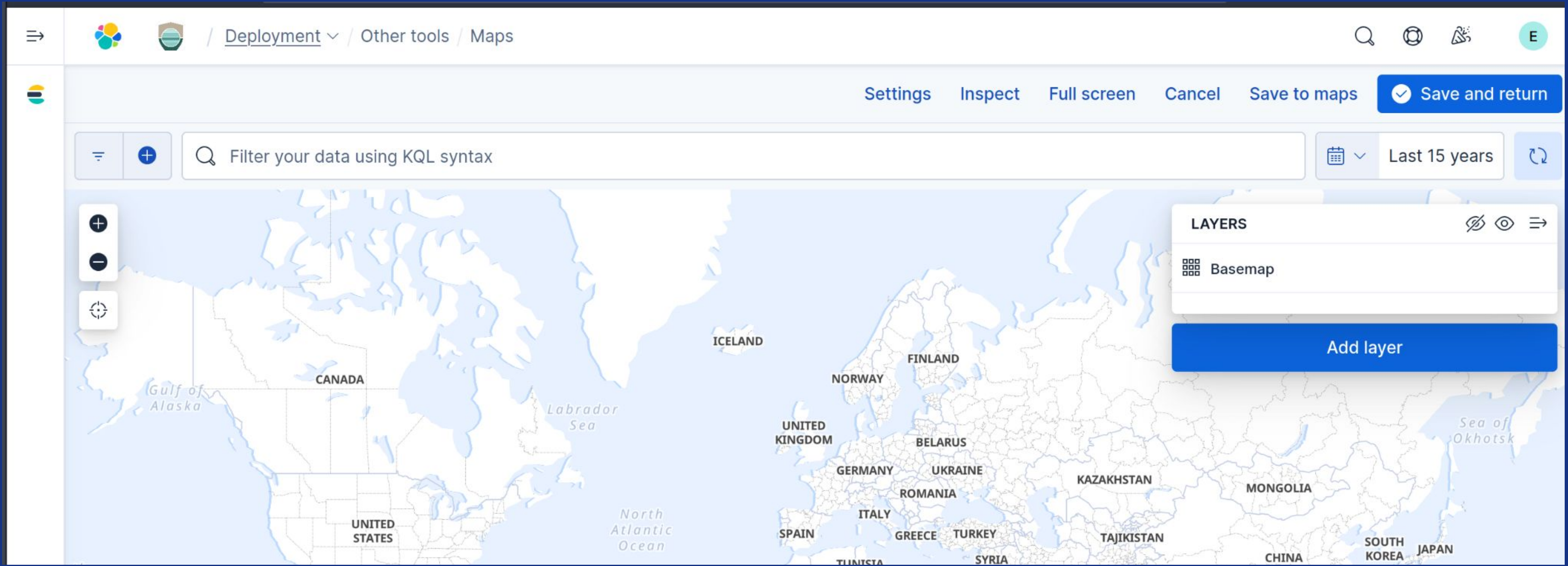
Breakdown (Optional): dataset

Elastic Maps

elastic

Interface

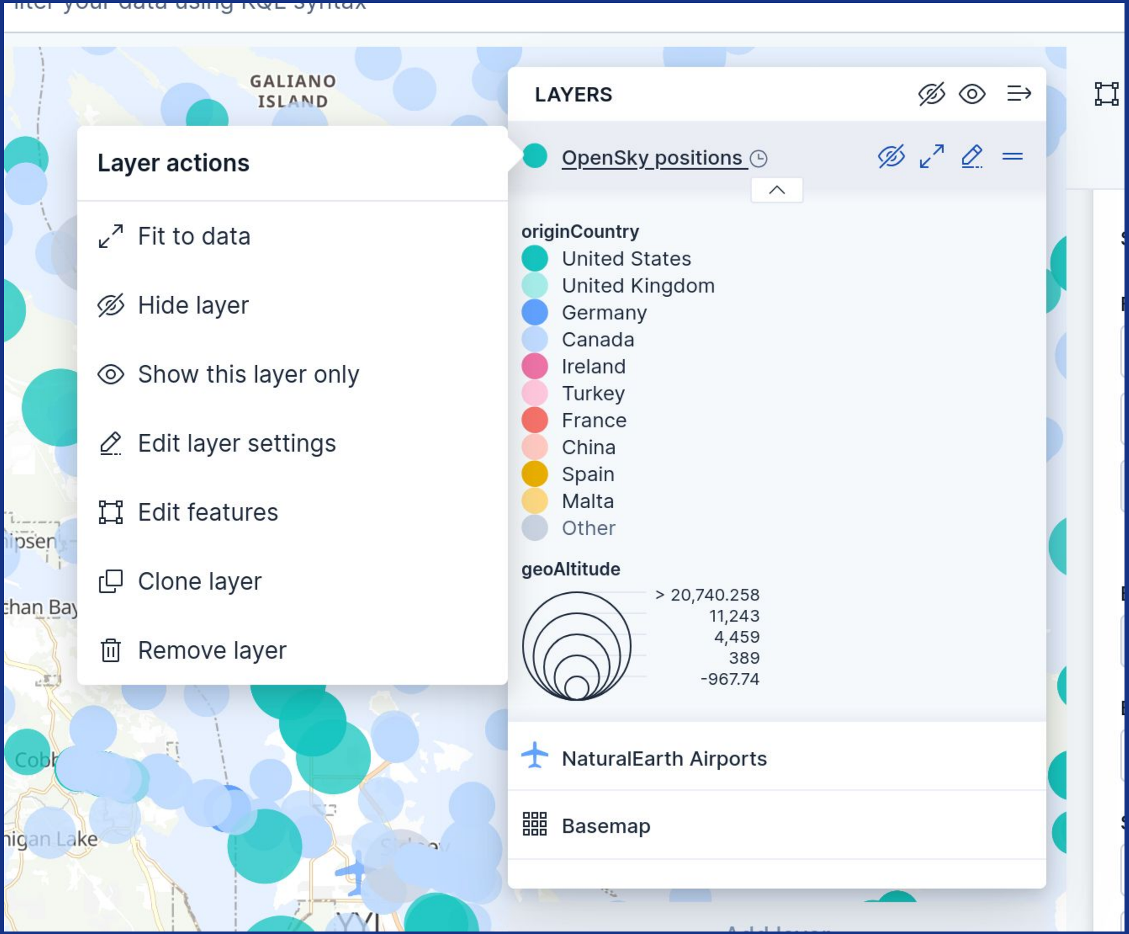
Same elements as in Lens, Dashboards, etc.



Interface

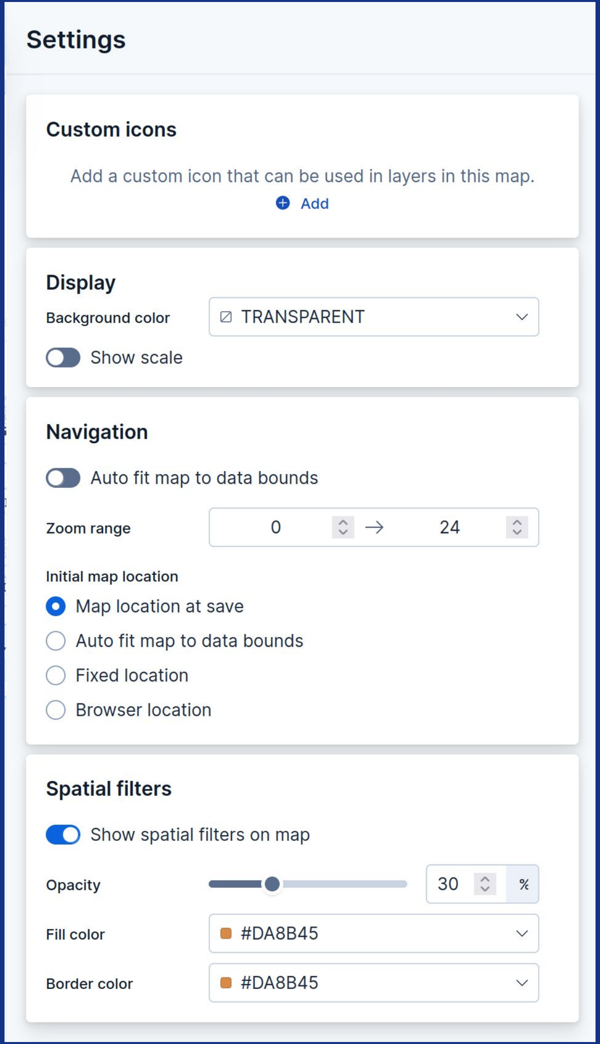
Familiar layers user interface

- Quick actions by the name
- Layers can be reordered, grouped, cloned
- Legend shows all data driven properties
- Actions depend on layer type



Settings

- Upload custom icons for point symbols
- Change background color if basemap is displayed
- Navigation defaults
- Spatial filters settings



Settings

Custom icons

Add a custom icon that can be used in layers in this map.

Display

Background color: TRANSPARENT

○ Show scale

Navigation

○ Auto fit map to data bounds

Zoom range: 0 → 24

Initial map location

● Map location at save

○ Auto fit map to data bounds

○ Fixed location

○ Browser location

Spatial filters

● Show spatial filters on map

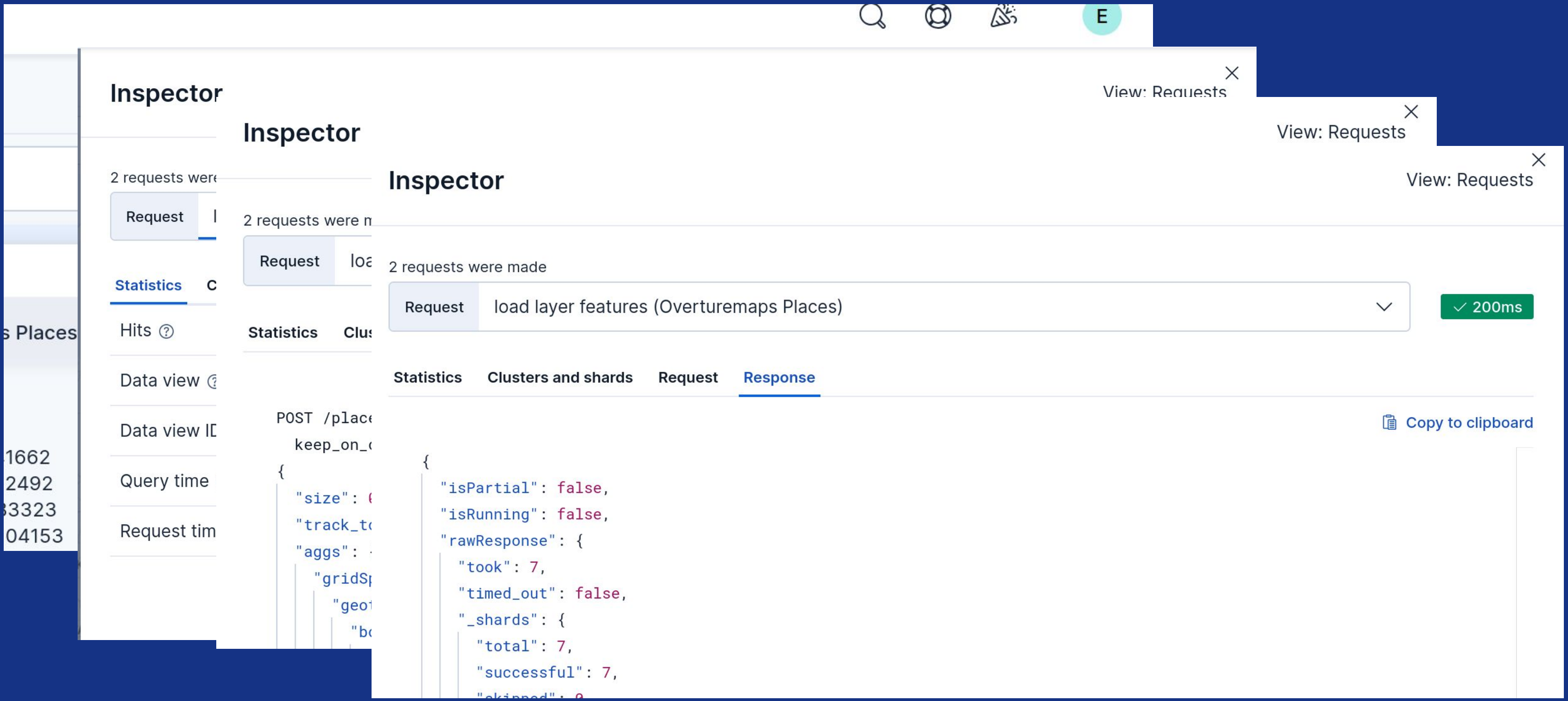
Opacity: 30 %

Fill color: 🟧 #DA8B45

Border color: 🟧 #DA8B45

Layer inspect

See the queries to Elasticsearch in detail



Inspector

View: Requests

2 requests were

Request

Statistics C

Hits

Data view

Data view ID

Query time

Request tim

s Places

1662
2492
33323
04153

Inspector

View: Requests

2 requests were m

Request loa

Statistics Clus

```
POST /place
  keep_on_c
{
  "size": (
  "track_tc
  "aggs": {
    "gridSp
      "geot
        "bc
```

Inspector

View: Requests

2 requests were made

Request load layer features (Overturemaps Places) ✓ 200ms

Statistics | Clusters and shards | Request | Response

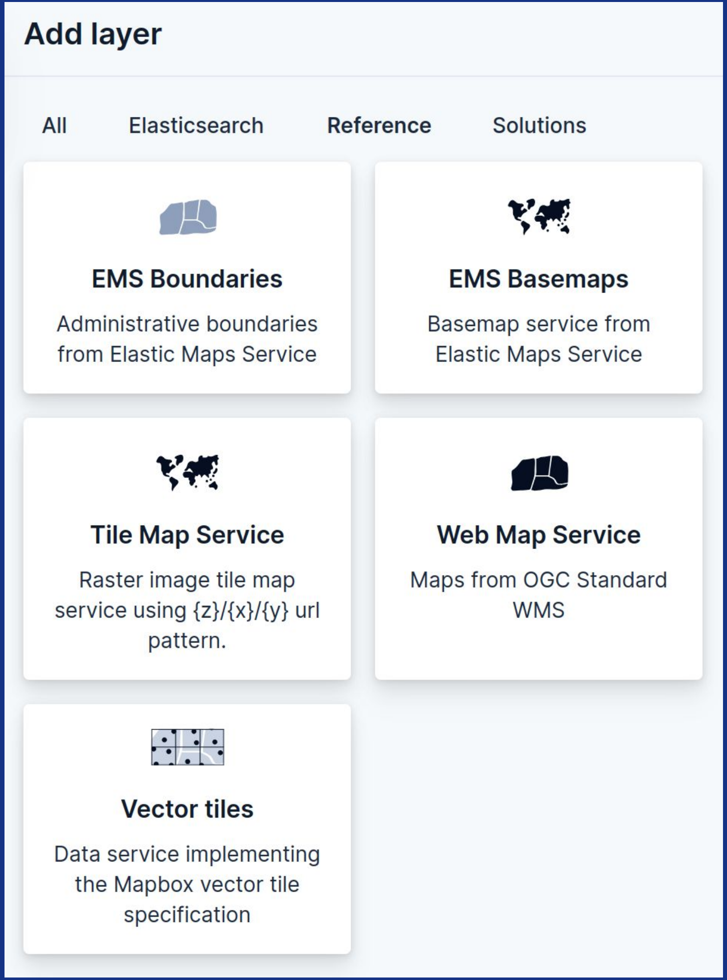
Copy to clipboard

```
{
  "isPartial": false,
  "isRunning": false,
  "rawResponse": {
    "took": 7,
    "timed_out": false,
    "_shards": {
      "total": 7,
      "successful": 7,
      "skipped": 0
```

Reference layers

Data outside from Elasticsearch

- EMS Basemaps
  - Default basemap provided by Elastic
- EMS Boundaries
  - Administrative boundaries ready to join with Elasticsearch data
- Web Map Service and Tile Map Service
  - Custom basemaps (imagery, official cartography, etc.)
- Vector Tiles
  - Vector data to style manually



Add layer

EMS Boundaries

Administrative boundaries from Elastic Maps Service

EMS Basemaps

Basemap service from Elastic Maps Service

Tile Map Service

Raster image tile map service using {z}/{x}/{y} url pattern.

Web Map Service

Maps from OGC Standard WMS

Vector tiles

Data service implementing the Mapbox vector tile specification

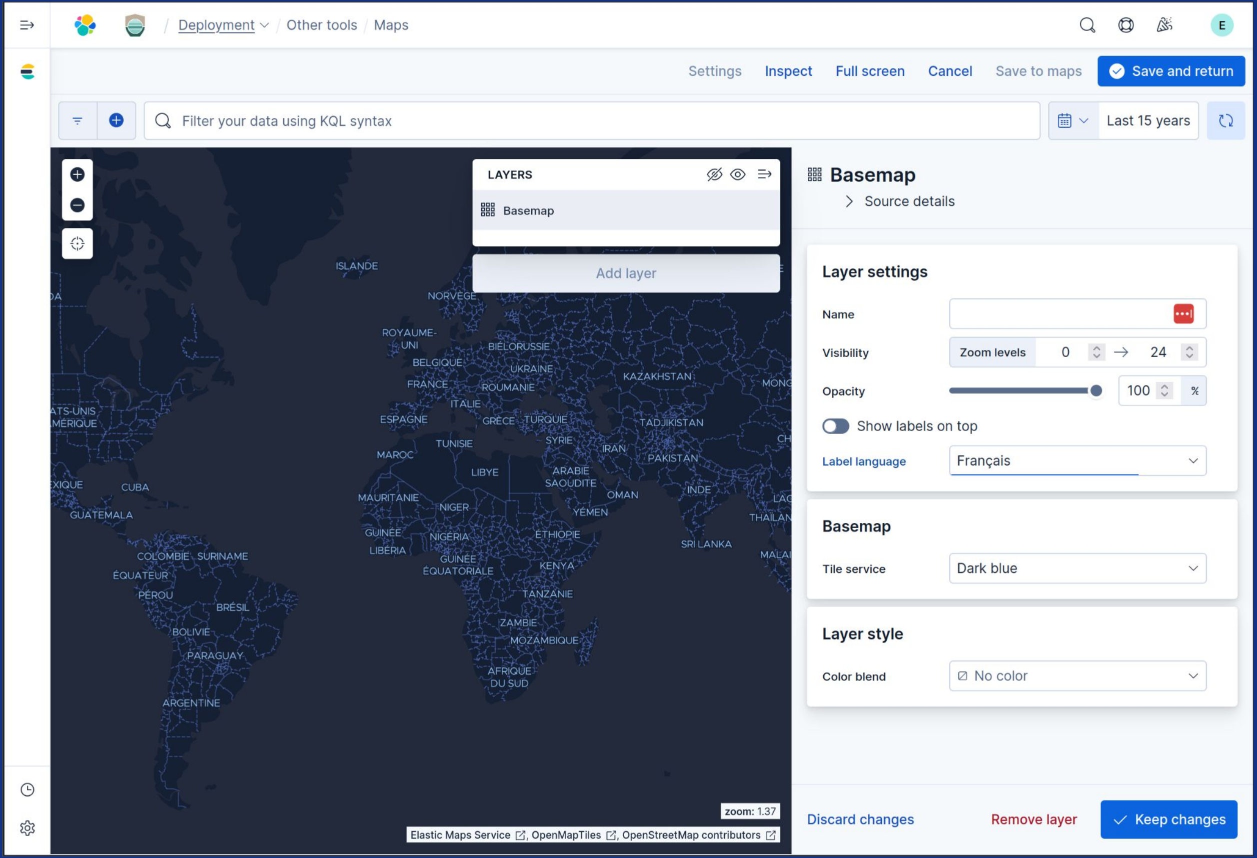
EMS Basemaps

Settings

- Labels language
- Labels on top
- Opacity
- Basemap style
- Colorize

In 9.1

- Globe mode



Basemap

Source details

Layer settings

Name

Visibility: Zoom levels 0 → 24

Opacity: 100 %

Show labels on top

Label language: Français

Basemap

Tile service: Dark blue

Layer style

Color blend: No color

Data layers

Loading Elasticsearch data in different ways

Documents: load individual index documents using vector tiles or JSON representation

ES|QL: craft queries that return geometries

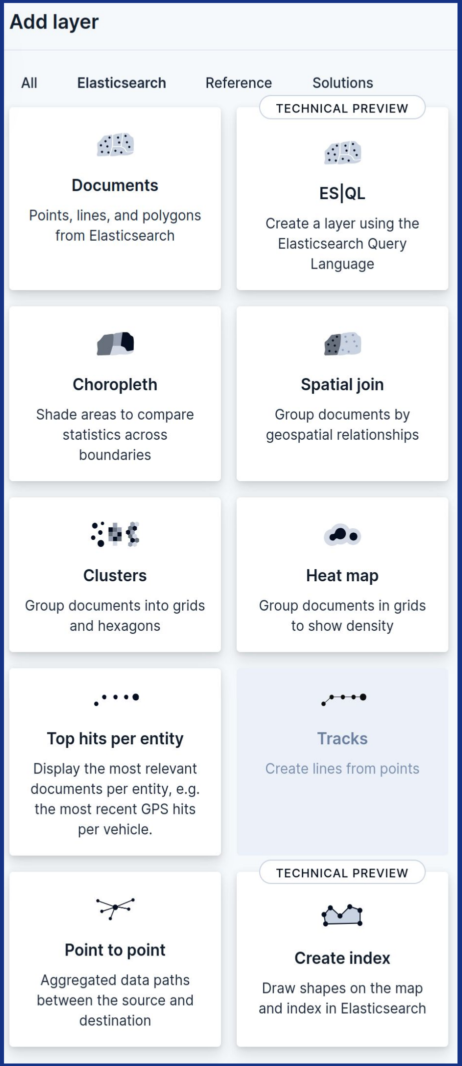
Spatial Join: basic support for client side spatial join

Clusters: aggregate into clusters, grids, and hexagons (non-free)

Heat map

Top hits per entity: display the n-latest documents of time series

Point to point: connect source and destination fields



Add layer

All | Elasticsearch | Reference | Solutions

Documents
Points, lines, and polygons from Elasticsearch

TECHNICAL PREVIEW
ES|QL
Create a layer using the Elasticsearch Query Language

Choropleth
Shade areas to compare statistics across boundaries

Spatial join
Group documents by geospatial relationships

Clusters
Group documents into grids and hexagons

Heat map
Group documents in grids to show density

Top hits per entity
Display the most relevant documents per entity, e.g. the most recent GPS hits per vehicle.

Tracks
Create lines from points

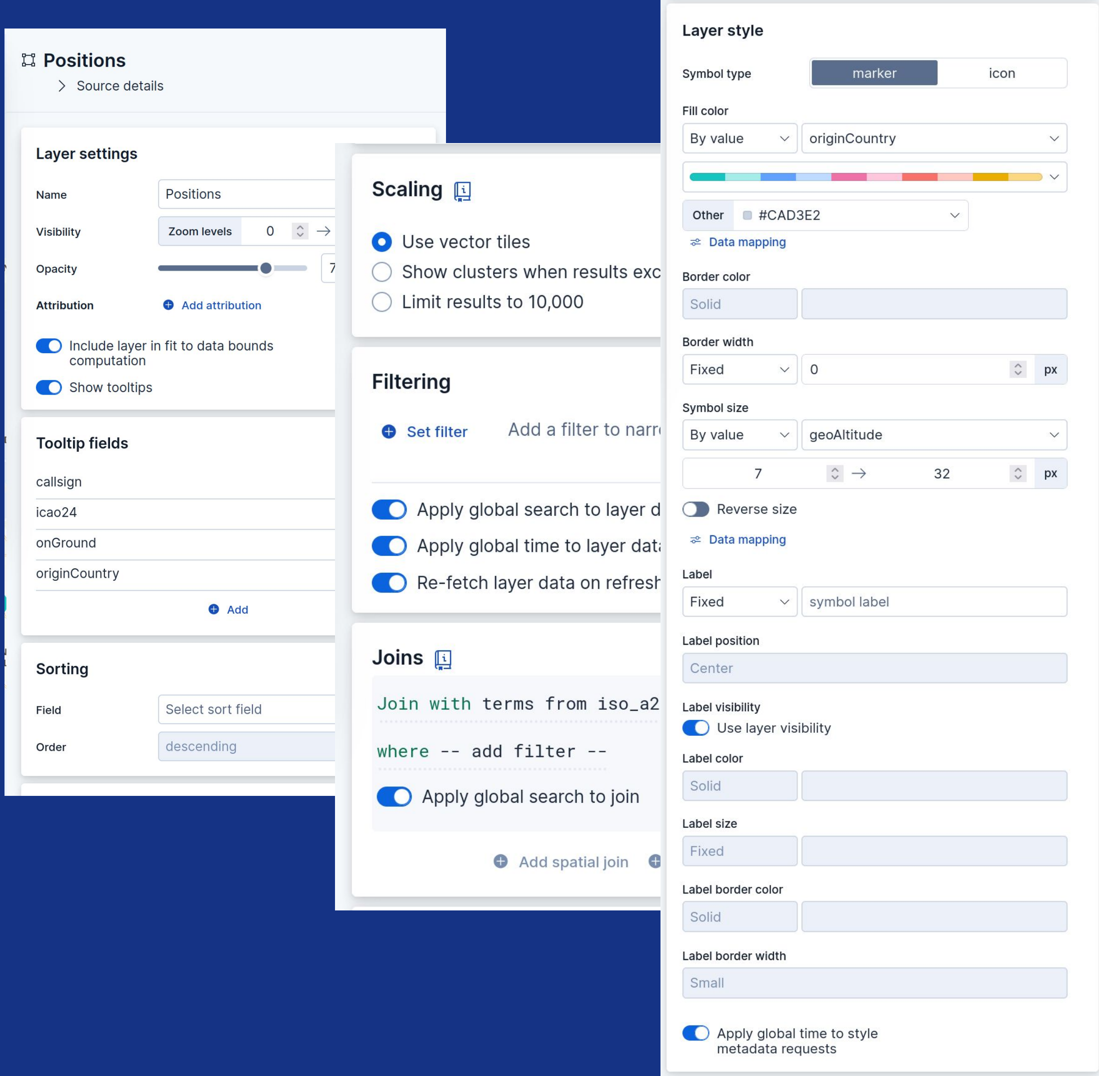
Point to point
Aggregated data paths between the source and destination

TECHNICAL PREVIEW
Create index
Draw shapes on the map and index in Elasticsearch

Documents

Render individual documents

- Zoom based visibility and Opacity
- Select fields for tooltips
- Sort by a field
- Scaling:
  - Vector tiles
  - First 10K documents
  - Automatically cluster > 10K
- Join with another index
- Styling
  - Symbol, sizes, colors, label



Positions

Source details

Layer settings

Name: Positions

Visibility: Zoom levels 0 →

Opacity: 7

Attribution: Add attribution

Include layer in fit to data bounds computation

Show tooltips

Tooltip fields

callsign

icao24

onGround

originCountry

Add

Sorting

Field: Select sort field

Order: descending

Scaling

● Use vector tiles

○ Show clusters when results exc

○ Limit results to 10,000

Filtering

Set filter — Add a filter to narr

Apply global search to layer d

Apply global time to layer dat

Re-fetch layer data on refresh

Joins

```
Join with terms from iso_a2
where -- add filter --
```

Apply global search to join

Add spatial join

Layer style

Symbol type: marker | icon

Fill color: By value — originCountry

Other: #CAD3E2

Data mapping

Border color: Solid

Border width: Fixed — 0 px

Symbol size: By value — geoAltitude

7 → 32 px

Reverse size

Data mapping

Label: Fixed — symbol label

Label position: Center

Label visibility: Use layer visibility

Label color: Solid

Label size: Fixed

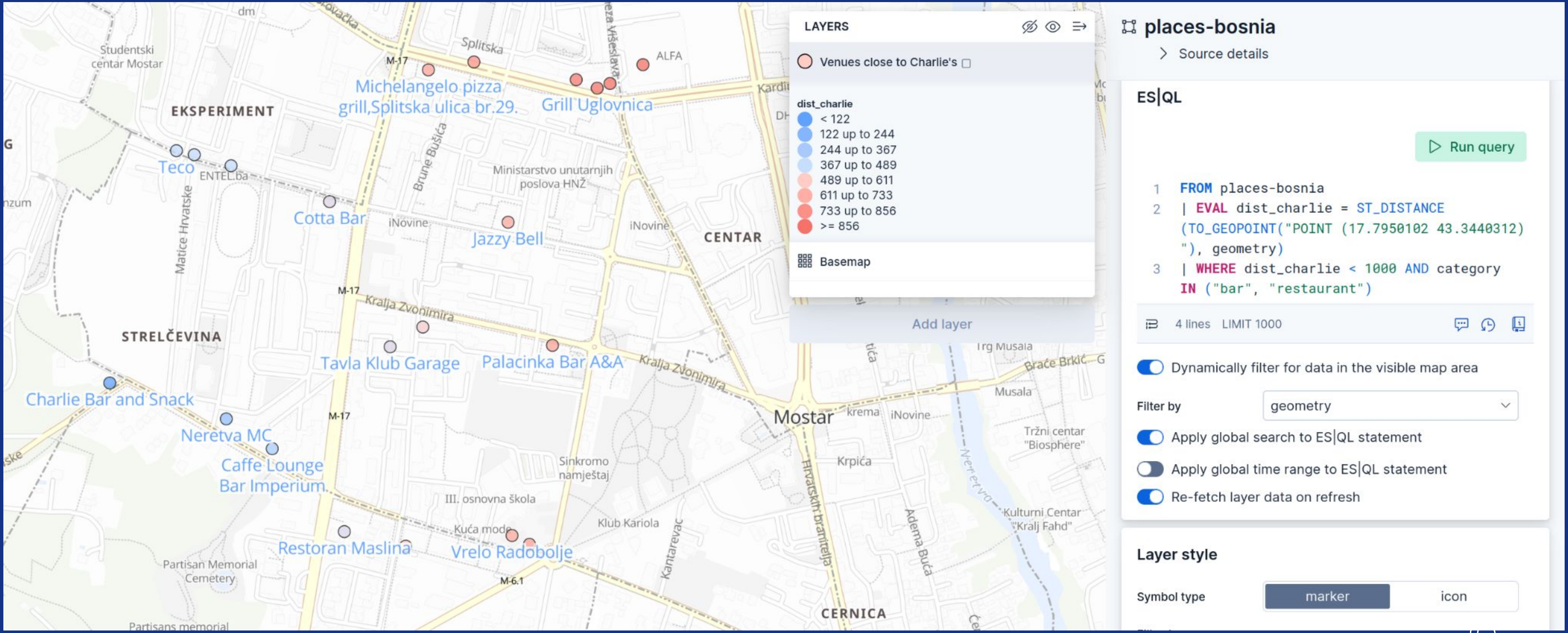
Label border color: Solid

Label border width: Small

Apply global time to style metadata requests

ES|QL

Similar to the documents layer type, but using a query as the source for the layer features



LAYERS

Venues close to Charlie's □

dist_charlie

- ● < 122
- ● 122 up to 244
- ● 244 up to 367
- ● 367 up to 489
- ● 489 up to 611
- ● 611 up to 733
- ● 733 up to 856
- ● >= 856

Basemap

Add layer

places-bosnia

Source details

ES|QL

▷ Run query

```
FROM places-bosnia
| EVAL dist_charlie = ST_DISTANCE
(TO_GEOPOINT("POINT (17.7950102 43.3440312)
"), geometry)
| WHERE dist_charlie < 1000 AND category
IN ("bar", "restaurant")
```

4 lines LIMIT 1000

☑ Dynamically filter for data in the visible map area

Filter by: geometry

☑ Apply global search to ES|QL statement

☐ Apply global time range to ES|QL statement

☑ Re-fetch layer data on refresh

Layer style

Symbol type: [✓ marker] icon

elastic

Clusters

Rendering big data

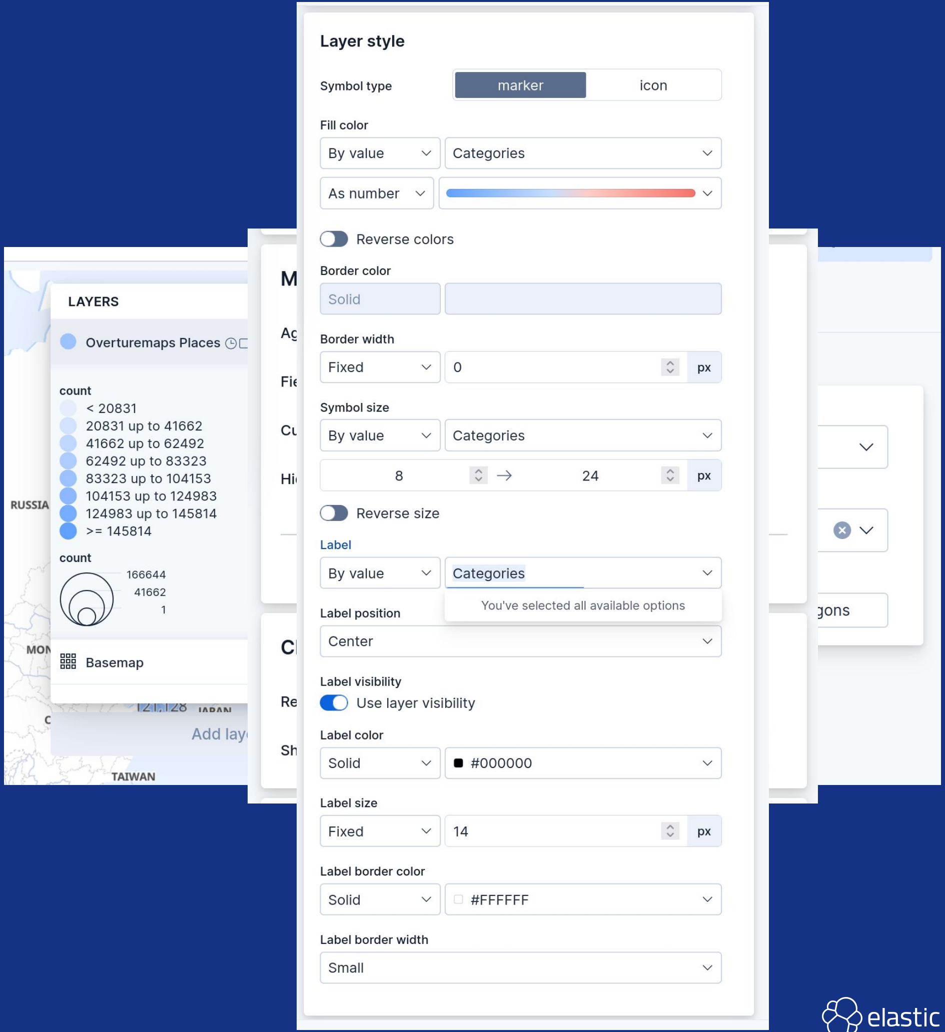
Aggregate into:

- Geotile: clusters or grid
- H3 grid 🛒

Layer settings

- Each metric defines an aggregation function on a field
  - To be used as labels, and data driven properties
- Spatial grid resolution
- Aggregation switch

Layer styling



LAYERS

Overturemaps Places

count

- < 20831
- 20831 up to 41662
- 41662 up to 62492
- 62492 up to 83323
- 83323 up to 104153
- 104153 up to 124983
- 124983 up to 145814
- >= 145814

count

166644
41662
1

Basemap

Add lay

Layer style

Symbol type: marker | icon

Fill color: By value | Categories

As number | color ramp

Reverse colors

Border color: Solid

Border width: Fixed | 0 px

Symbol size: By value | Categories

8 → 24 px

Reverse size

Label: By value | Categories

You've selected all available options

Label position: Center

Label visibility: Use layer visibility

Label color: Solid | ■ #000000

Label size: Fixed | 14 px

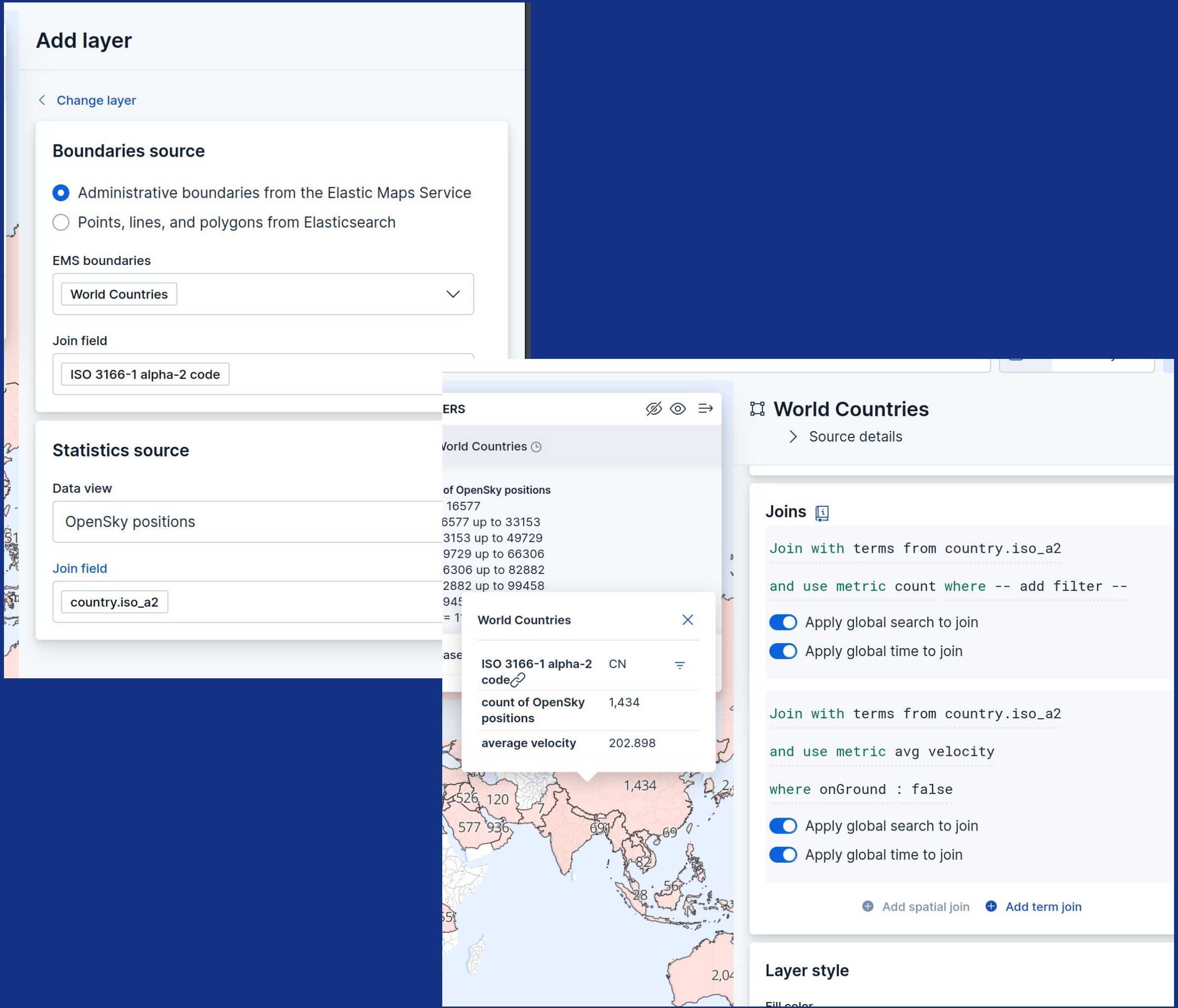
Label border color: Solid | □ #FFFFFF

Label border width: Small

Choropleth

Aggregate and join with reference data

- Use EMS boundaries or an index for the reference data
- Aggregate any metrics using a common field (ISO codes, usually)
  - Available for styling
  - Tooltips
- Apply filters
- Allow removing by threshold
  - Hide outliers



Add layer

< Change layer

Boundaries source

● Administrative boundaries from the Elastic Maps Service
○ Points, lines, and polygons from Elasticsearch

EMS boundaries: World Countries

Join field: ISO 3166-1 alpha-2 code

Statistics source

Data view: OpenSky positions

Join field: country.iso_a2

World Countries

ISO 3166-1 alpha-2 code	CN
count of OpenSky positions	1,434
average velocity	202.898

World Countries

> Source details

Joins

```
Join with terms from country.iso_a2
and use metric count where -- add filter --
```

Apply global search to join
Apply global time to join

```
Join with terms from country.iso_a2
and use metric avg velocity
where onGround : false
```

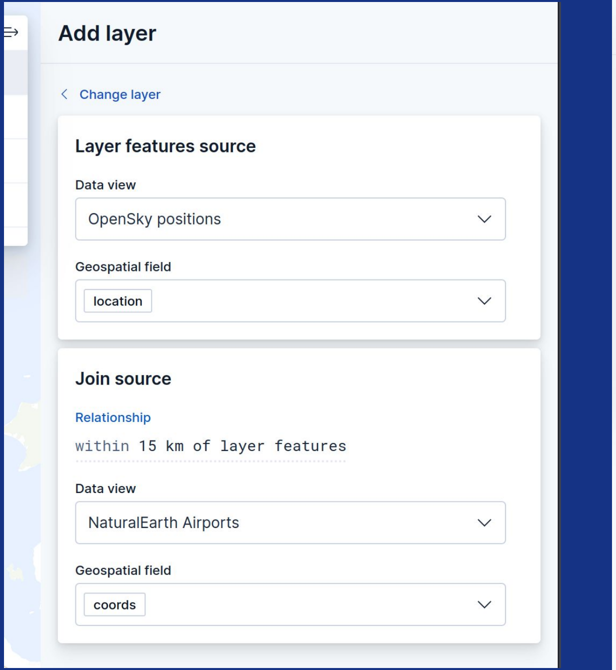
Apply global search to join
Apply global time to join

⊕ Add spatial join ⊕ Add term join

Layer style

Fill color

Other types



Add layer

Change layer

Layer features source

Data view: OpenSky positions

Geospatial field: location

Join source

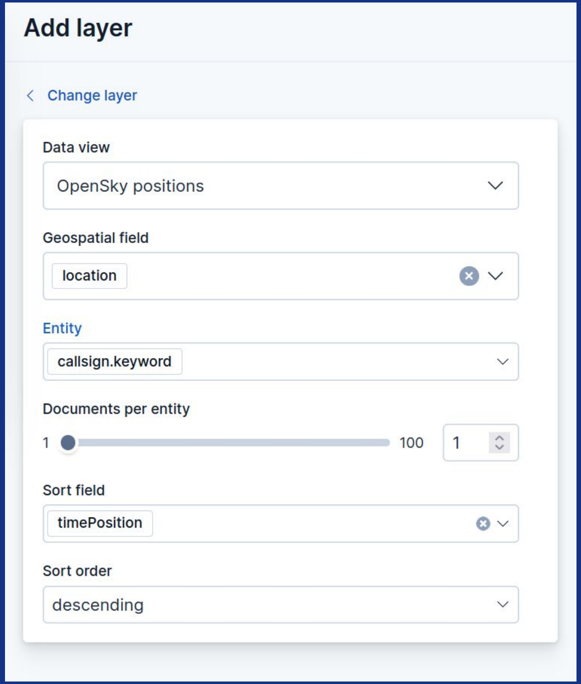
Relationship

```
within 15 km of layer features
```

Data view: NaturalEarth Airports

Geospatial field: coords

Spatial Join



Add layer

Change layer

Data view: OpenSky positions

Geospatial field: location

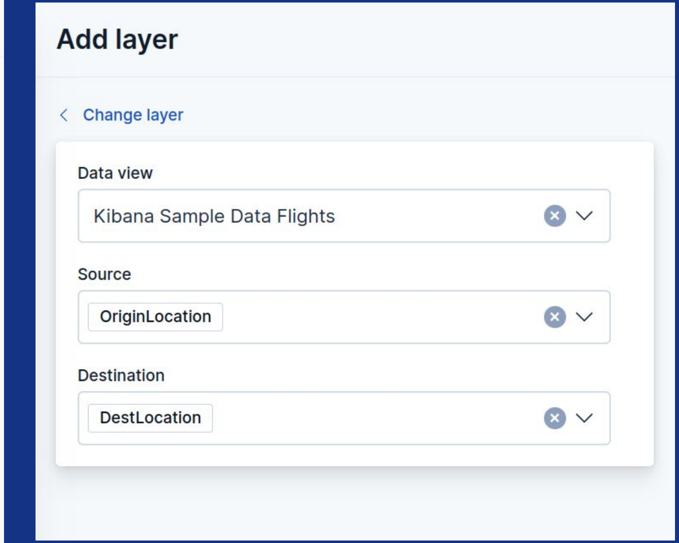
Entity: callsign.keyword

Documents per entity: 1 — 100, 1

Sort field: timePosition

Sort order: descending

Top hits per entity



Add layer

Change layer

Data view: Kibana Sample Data Flights

Source: OriginLocation

Destination: DestLocation

Point to point

elastic

Maps in dashboards: filters

Maps in dashboards: synchronized extents

My Rules of Thumb

Working With Kibana

- **Segregate** your data and visualizations whenever possible
- Let the Elastic Stack do the **heavy lifting** (when possible)
- **Saved searches** save you time and energy.
- When you are stuck, look at the **time picker** and the **filters**.

My Rules of Thumb

Design

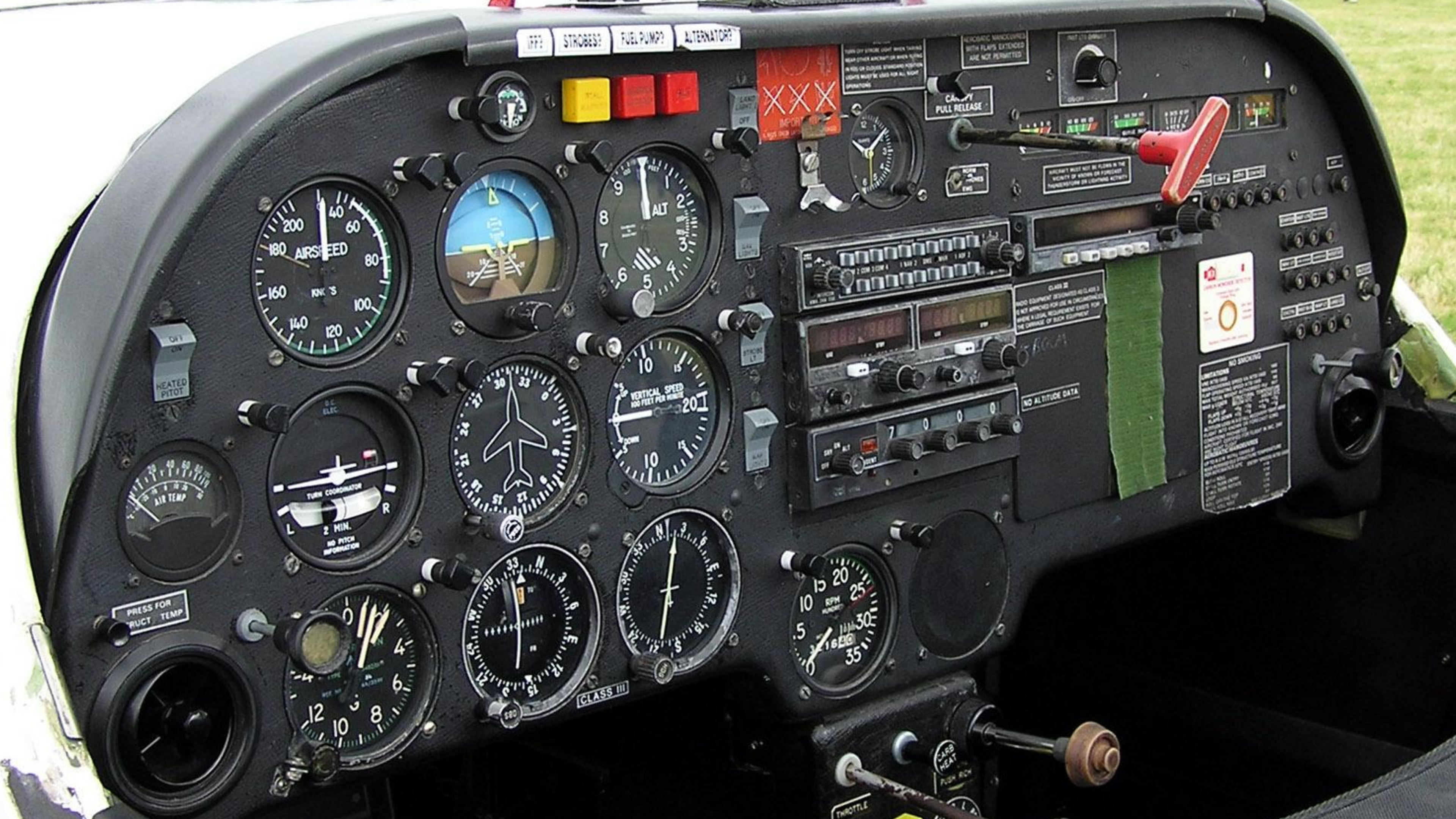
- Know your **audience**.
- **Lead** your audience in the correct direction(s).
- Don’t use **color** to communicate meaning (exclusively)
- Plan for **filtering** with Indicators and Trends

My Rules of Thumb

Design

- Differentiate between **executive**, **operational**, and **analytic** dashboards.
- Focused, **smaller** dashboards are better than a single monster.

LOWER CARGO FIRE PROTECTION
COMPARTMENT
DETECTOR TEST
SQUIB TEST
AFT CARGO
MAIN


IFF?
STROBES?
FUEL PUMP?
ALTERNATOR?
TURN OFF STROBE LIGHT WHEN TAXIING NEAR OTHER AIRCRAFT OR WHEN FLYING IN FOG OR CLOUDS. STANDARD POSITION LIGHTS MUST BE USED FOR ALL NIGHT OPERATIONS
AEROBATIC MANOEUVRES WITH FLAPS EXTENDED ARE NOT PERMITTED
CANOPY PULL RELEASE
AIRCRAFT MUST NOT BE FLOWN IN THE VICINITY OF KNOWN OR FORECAST THUNDERSTORM OR LIGHTNING ACTIVITY
LAND LIGHT OFF
AIRSPEED
KNOTS
ALT
VERTICAL SPEED 100 FEET PER MINUTE
UP
DOWN
OFF ON
HEATED PITOT
STROBE LT
D.C. ELEC.
TURN COORDINATOR
L
R
2 MIN.
NO PITCH INFORMATION
AIR TEMP
PRESS FOR STRUCT. TEMP
CLASS III
RADIO EQUIPMENT DESIGNATED AS CLASS 3 IS NOT APPROVED FOR USE IN CIRCUMSTANCES WHERE A LEGAL REQUIREMENT EXISTS FOR THE CARRIAGE OF SUCH EQUIPMENT
NO ALTITUDE DATA
NO SMOKING
LIMITATIONS
RPM HUNDREDS
CARB HEAT
PUSH RICH
THROTTLE

My Rules of Thumb

Layout

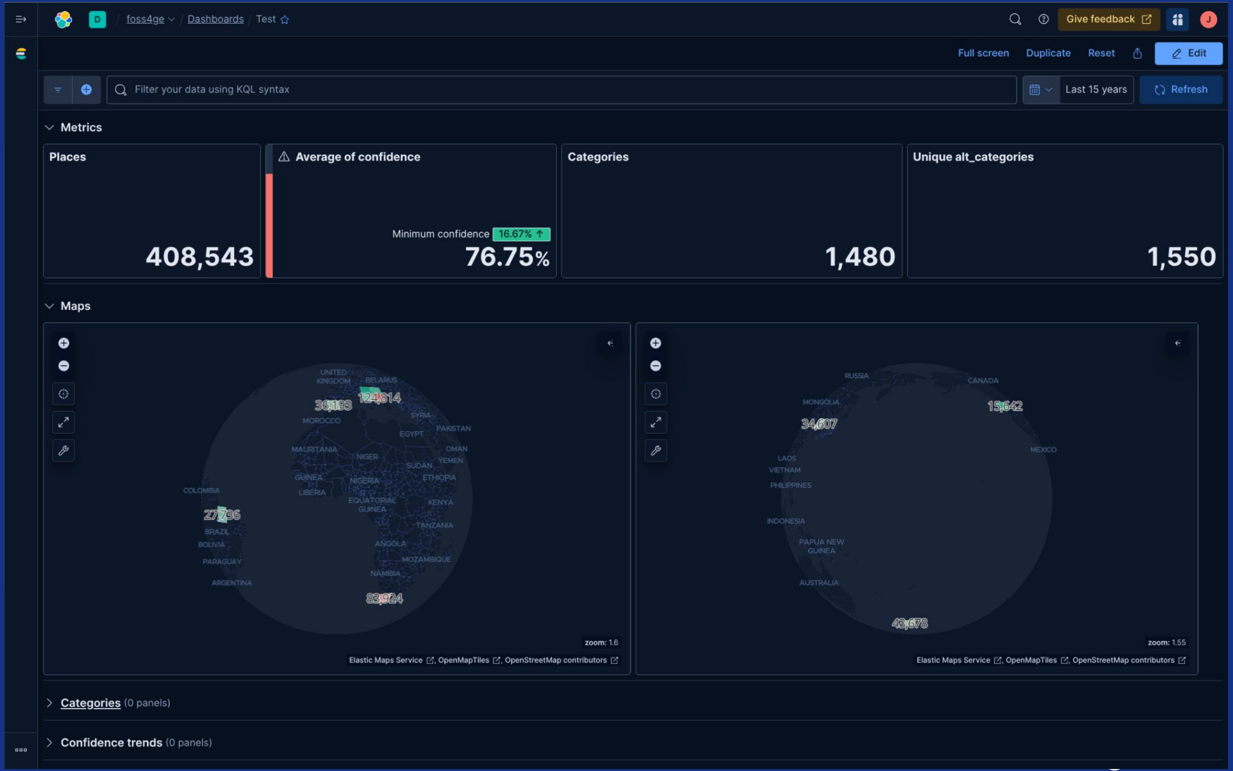
- Think about how your dashboard will be viewed.
- Ensure that your indicators go “above the fold”
- Be deliberate with columns (I rarely use more than 3)
- Horizontal space can be used for effective timeline comparisons

What's coming to 9.1?

What's new in 9.1

Already available in Elastic Serverless offering

- Improvements in ES|QL text search functions
- Maps Globe projection
- Collapsible panels on dashboards
- ES|QL controls and ?variables in queries
- Improvements in metric and table visualization types
- View chart configuration in read-only dashboards



Metrics

Places	Average of confidence	Categories	Unique alt_categories
408,543	Minimum confidence 16.67% ↑ 76.75%	1,480	1,550

Maps

Categories (0 panels)

Confidence trends (0 panels)

Questions?

Thank you!

FOSS4G Europe, July, 2025

Mostar, Bosnia-Herzegovina

https://ela.st/2025-foss4ge-workshop
https://ela.st/2025-foss4ge-workshop-notes

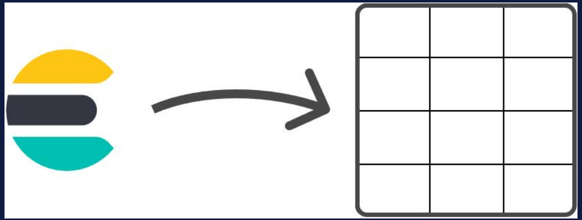
elastic | The Search AI Company

FOSS4G
EUROPE MOSTAR 2025

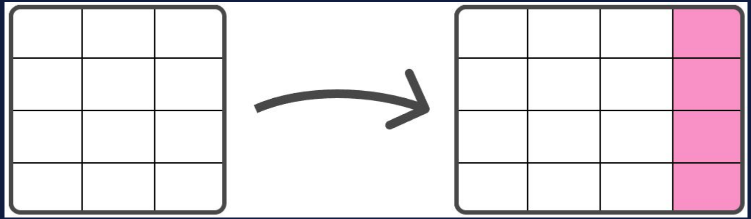
ES|QL supporting material

ES|QL

- An ES|QL query is comprised of a series of commands changed together by pipes
  - Source commands retrieve or generate data in the form of tables

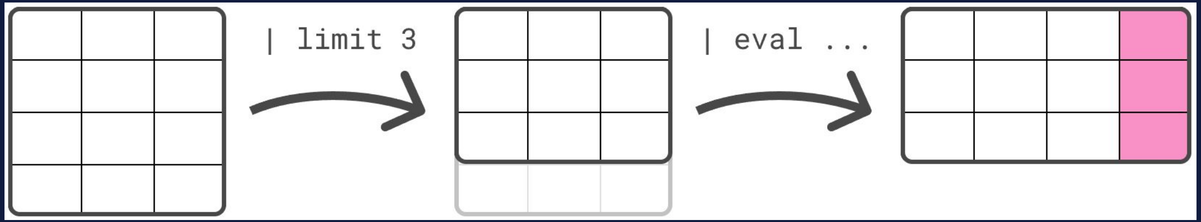


  - Processing commands take a table as input and produce a table as output

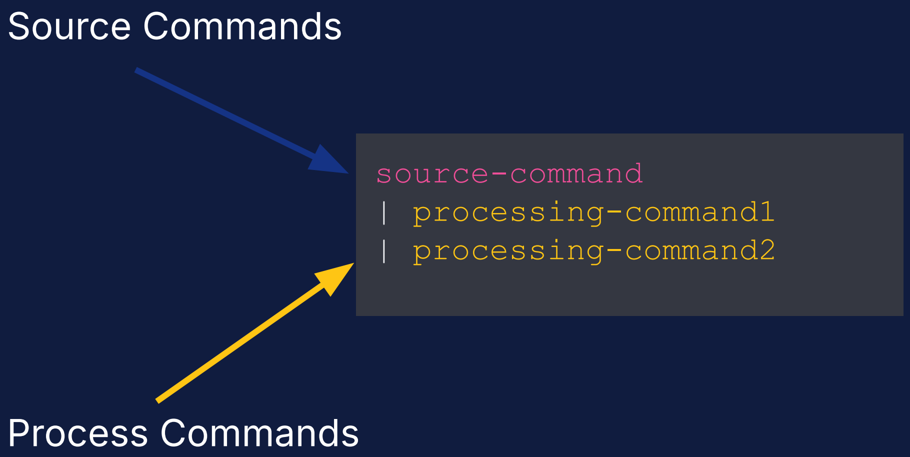


ES|QL

- You can chain processing commands, separated by a pipe character: | Each processing command works on the output table of the previous command.



ES|QL Syntax



Source Commands

```
source-command
| processing-command1
| processing-command2
```

Process Commands

OR

```
source-command | processing-command1 | processing-command2
```

ES|QL

Example command

```
POST /_query
{
  "query": """
    FROM library
    | EVAL year = DATE_TRUNC(1
YEARS, release_date)
    | STATS MAX(page_count) BY year
    | SORT year
    | LIMIT 5
  """
  }
```

```
{
  "columns": [
    { "name":
"MAX(page_count)", "type":
"integer"},
    { "name": "year"
, "type": "date"}
  ],
  "values": [
    [268,
"1932-01-01T00:00:00.000Z"],
    [224,
"1951-01-01T00:00:00.000Z"],
    [227,
"1953-01-01T00:00:00.000Z"],
    [335,
"1959-01-01T00:00:00.000Z"],
    [604,
"1965-01-01T00:00:00.000Z"]
  ]
  }
```

ES|QL

Example command

To return results formatted as text, CSV, or TSV, use the format parameter:

```
POST /_query?format=txt
{
 "query": """
  FROM library
  | EVAL year = DATE_TRUNC(1
YEARS, release_date)
  | STATS MAX(page_count) BY
year
  | SORT year
  | LIMIT 5
 """
 }
```

ES|QL - 8.15 Supported types

ES|QL currently supports the following field types:

- alias
- boolean
- date
- double (float, half_float, scaled_float are represented as double)
- ip
- keyword family including keyword, constant_keyword, and wildcard
- int (short and byte are represented as int)
- long
- null
- text
- unsigned_long
- version
- Spatial types
  - geo_point
  - geo_shape
  - point
  - shape

elastic

ES|QL - 8.15 Unsupported types

Field types

- `binary`
- `completion`
- `dense_vector`
- `double_range`
- `flattened`
- `float_range`
- `histogram`
- `integer_range`
- `ip_range`
- `long_range`
- `nested`
- `rank_feature`
- `rank_features`
- `search_as_you_type`

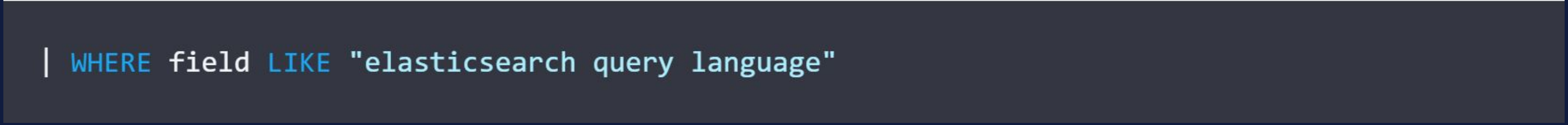
TSDB metrics

- `counter`
- `position`
- `aggregate_metric_double`

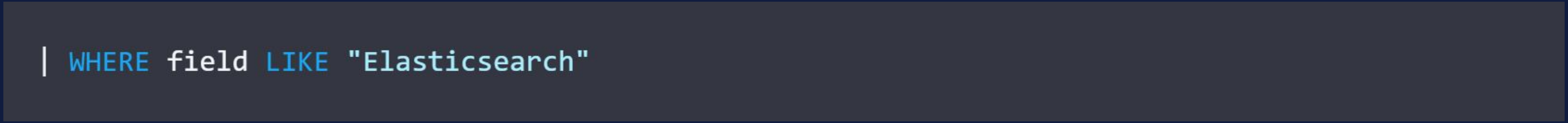
Date/time

- `date_nanos`
- `date_range`

ES|QL - 8.12+ Full-text search is not supported (for now)

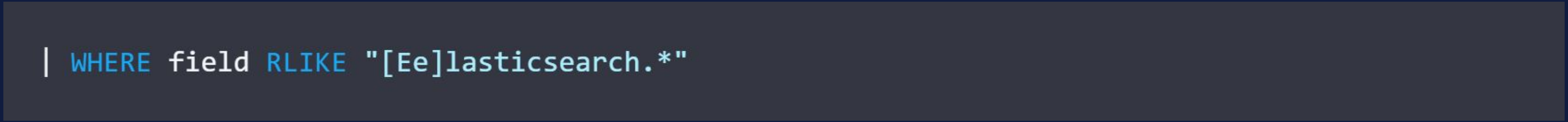


```
| WHERE field LIKE "elasticsearch query language"
```



```
| WHERE field LIKE "Elasticsearch"
```

ES|QL - 8.12 Full-text search is not supported (for now)



```
| WHERE field RLIKE "[Ee]lasticsearch.*"
```

`text` fields behave like `keyword` fields

ES|QL Source Commands

ES|QL Source Commands `From`

- `From` – Returns a table with up to 500 documents from a data stream, index, or alias. Each row in the resulting table represents a document. Each column corresponds to a field, and can be accessed by the name of that field.

```
from logs-*, metrics-*, kibana_sample_data_logs
```

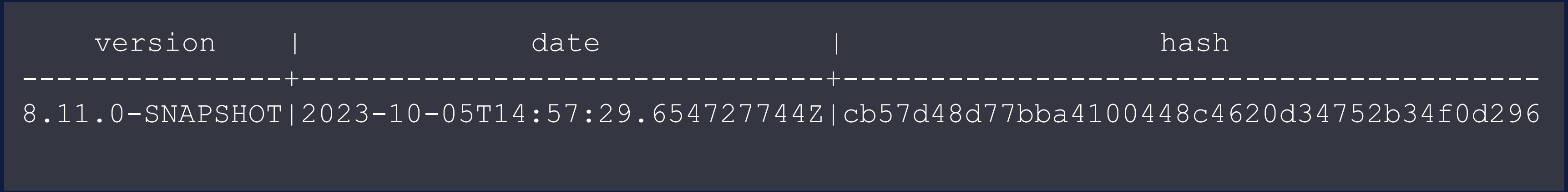
I like to think its calling the index name in `GET /logs/_search`

- ES|QL can access metadata fields. The currently supported ones are:
  - `_index:` the index to which the document belongs. The field is of the type keyword.
  - `_id:` the source document’s ID. The field is of the type `keyword`.
  - `_version:` the source document’s version. The field is of the type `long`.

```
from index [METADATA _index, _id]
```

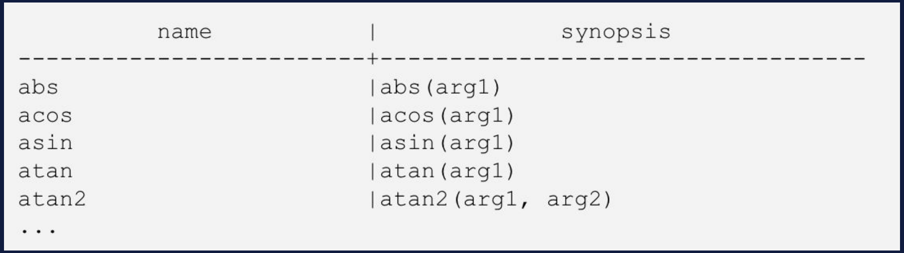
ES|QL Source Commands Show

- Show – returns information about the deployment and its capabilities:
  - SHOW INFO to return the deployment's version, build date and hash.



```
    version     |              date              |                  hash
----------------+--------------------------------+----------------------------------------
8.11.0-SNAPSHOT|2023-10-05T14:57:29.654727744Z|cb57d48d77bba4100448c4620d34752b34f0d296
```

  - SHOW FUNCTIONS to return a list of all supported functions and a synopsis of each function.



```
          name           |            synopsis
-------------------------+----------------------------------
abs                      |abs(arg1)
acos                     |acos(arg1)
asin                     |asin(arg1)
atan                     |atan(arg1)
atan2                    |atan2(arg1, arg2)
...
```

ES|QL Source Commands Row

- Row – produces a row with one or more columns with values that you specify.

```
POST /_query?format=txt
{
  "query": """
  ROW a = 1, b = "abc", c = null, d = [1,2,3]
  """
}
```

OR

```
row a = 1, b = "abc", c = [1,2,3]
```

Output

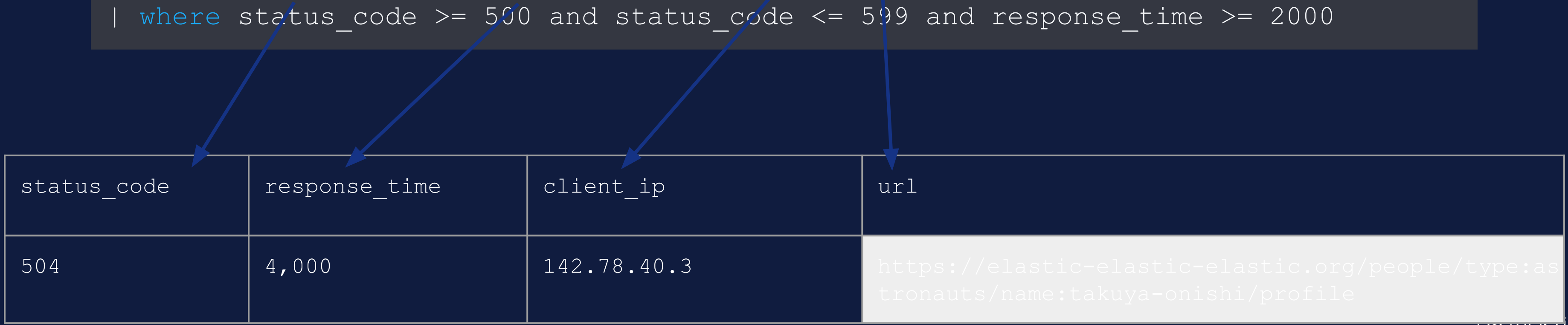
```
       a       |       b       |       c       |       d
---------------+---------------+---------------+---------------
-
1              |abc            |null           |[1, 2, 3]
```

ES|QL Process Commands keep

- ! Fundamental Process Command !
- keep command allows you to specify which fields/columns should be included in the output table and tier order

In the previous example you’ll notice we have a keep

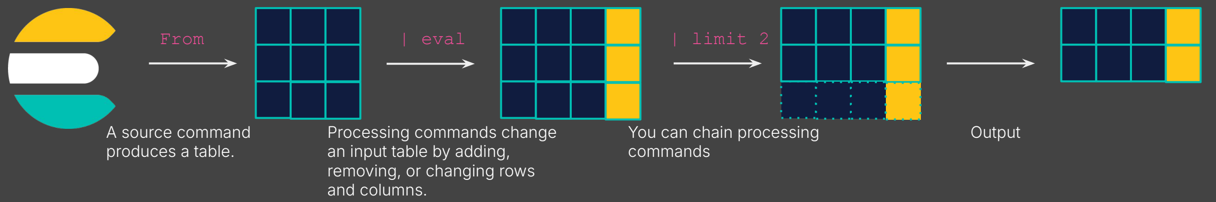
```
from apache-logs
| keep status_code, response_time, client_ip, url
| where status_code >= 500 and status_code <= 599 and response_time >= 2000
```



status_code	response_time	client_ip	url
504	4,000	142.78.40.3	https://elastic-elastic-elastic.org/people/type:astronauts/name:takuya-onishi/profile

ES|QL Quick Reference Guide

Version 1.3



From

| eval

| limit 2

A source command produces a table.

Processing commands change an input table by adding, removing, or changing rows and columns.

You can chain processing commands

Output

Source

- FROM
- ROW
- SHOW

Processing

- DISSECT
- DROP
- ENRICH
- EVAL
- GROK
- KEEP
- LIMIT
- MV_EXPAND
- RENAME
- SORT
- STATS ... BY
- WHERE

Operators

equality: ==
inequality: !=
less than: <
less than or equal: <=
larger than: >
larger than or equal: >=

- IS NULL
- IS NOT NULL
- CIDR_MATCH
- ENDS_WITH
- IN
- IS_FINITE
- IS_INFINITE
- IS_NAN
- LIKE
- RLIKE
- STARTS_WITH

Functions

Mathematical

- ABS
- ACOS
- ASIN
- ATAN
- ATAN2
- CEIL
- COS
- COSH
- E
- FLOOR
- LOG10
- PI
- POW
- ROUND
- SIN
- SINH
- SQRT
- TAN
- TANH
- TAU

String

- CONCAT
- LEFT
- LENGTH
- LTRIM
- REPLACE
- RIGHT
- RTRIM
- SPLIT
- SUBSTRING
- TRIM

Type conversion

- TO_BOOLEAN
- TO_DATETIME
- TO_DEGREES
- TO_DOUBLE
- TO_INTEGER
- TO_IP
- TO_LONG
- TO_RADIANS
- TO_STRING
- TO_UNSIGNED_LONG
- TO_VERSION

Date-time

- AUTO_BUCKET
- DATE_EXTRACT
- DATE_FORMAT
- DATE_PARSE
- DATE_TRUNC
- NOW

Multivalue

- MV_AVG
- MV_CONCAT
- MV_COUNT
- MV_DEDUPE
- MV_MAX
- MV_MEDIAN
- MV_MIN
- MV_SUM

Aggregate

- AVG
- COUNT
- COUNT_DISTINCT
- MAX
- MEDIAN
- MEDIAN_ABSOLUTE_DEVIATION
- MIN
- PERCENTILE
- SUM

Conditional

- CASE
- COALESCE
- GREATEST
- LEAST

Syntax Basics

```
source-command
| processing-command1
| processing-command2
```

```
FROM hosts
| WHERE CDIR_MATCH(ip, "127.0.0.2/32")
```

```
FROM employees
| KEEP first_name, last_name, height
| EVAL fullname = CONCAT(first_name," ",
last_name)
```

```
FROM employees
| WHERE first_name LIKE "?b*"
| KEEP first_name, last_name
| SORT first_name
```

```
FROM employees
| KEEP first_name, last_name, hire_date
| EVAL hired = DATE_FORMAT(hire_date,
"YYYY-MM-dd")
```

```
ROW words="foo;bar;baz;qux;quux;corge"
| EVAL word = SPLIT(word, ";")
```

Example

```
ROW a=[3, 5, 1, 6]
| EVAL avg_a = MV_AVG(a)
```

a:integer	avg_a:double
[3, 5, 1, 6]	3.75

```
ROW a=[1,2,3], b="b", j=["a","b"]
| MV_EXPAND a
```

a:integer	b:keyword	j:keyword
1	b	["a", "b"]
2	b	["a", "b"]
3	b	["a", "b"]

```
ROW language_code = "1"
| ENRICH languages_policy
```

language_code:keyword	language_name:keyword
1	English

```
ROW a = "1953-01-23T12:15:00Z - some text - 127.0.0.1;"
| DISSECT a "%{Y}-%{M}-%{D}T%{h}:%{m}:%{s}Z - %{msg} - %{ip};"
| KEEP Y, M, D, h, m, s, msg, ip
```

Y:keyword	M:keyword	D:keyword	h:keyword	m:keyword	s:keyword	msg:keyword	ip:keyword
1953	01	23	12	15	00	some text	127.0.0.1

```
FROM employees
| STATS count = COUNT(emp_no) BY languages
| SORT languages
```

count:long	languages:integer
15	1
19	2
17	3
18	4
21	5
10	null

Supported field types

- ✧ alias
- ✧ boolean
- ✧ Date
- ✧ ip
- ✧ long
- ✧ null
- ✧ text
- ✧ unsigned_long
- ✧ double (float, half_float, scaled_float are represented as double)
- ✧ keyword family including keyword, constant_keyword, and wildcard
- ✧ int (short and byte are represented as int)

version

https://ela.st/esqlquickreferenceguide

elastic

ES|QL Processing Commands

ES|QL Process Commands

Process Commands

- Processing commands take a table as input and produce a table as output
- You can chain processing commands, separated by a pipe character: |
- Each processing command works on the output table of the previous command.

dissect	drop	enrich	eval
grok	keep	limit	mv_expand
rename	sort	stats… by	where

ES|QL Process Commands `where`

- ! Fundamental Process Command !
- `where` uses conditions to filter rows from the input table that satisfy a given condition

Example:

You are analyzing server logs, and they contain fields like:

status_code (HTTP 200, 404, 500, etc.)	response_time
client_ip	url

You need to identify requests that resulted in HTTP status codes 500-599 and took longer than 2 seconds to respond. You can use the where command to apply both conditions

ES|QL Process Commands

where

```
from apache-logs
| keep status_code, response_time, client_ip, url
| where status_code >= 500 and status_code <= 599 and response_time >= 2000
```

status_code	response_time	client_ip	url
504	8,000	142.78.40.3	https://elastic-elastic-elastic.org/people/type:astronauts/name:takuya-onishi/profile
501	2,454	200.76.93.202	https://elastic-elastic-elastic.org/people/type:astronauts/name:ronald-grabe/profile
501	6,946	61.231.10.118	https://elastic-elastic-elastic.org/people/type:astronauts/name:andrei-borisenko/profile
504	8,000	69.139.73.154	https://elastic-elastic-elastic.org/people/type:astronauts/name:pham-tuan/profile

ES|QL Process Commands `sort`

- ! Fundamental Process Command !
- `sort` command orders the row of the output table based on the values of one or more field/columns. The default is `asc` but you can also use `desc`
- You can also sort multiple fields/columns

Example:

```
from apache-logs
| keep status code, response time, client ip, url
| where status_code >= 500 and status_code <= 599 and response_time >= 2000
| sort status_code desc
```

By default, `null` values are larger than other values so you can control placement of `nulls first` or `nulls last`.

ES|QL Process Commands `keep`

- ! Fundamental Process Command !
- `keep` command allows you to specify which fields/columns should be included in the output table and tier order

In the previous example you'll notice we have a keep

```
from apache-logs
| keep status_code, response_time, client_ip, url
| where status_code >= 500 and status_code <= 599 and response_time >= 2000
```

status_code	response_time	client_ip	url
504	4,000	142.78.40.3	https://elastic-elastic-elastic.org/people/type:astronauts/name:takuya-onishi/profile

ES|QL Process Commands `limit`

- ! Fundamental Process Command !
- `limit` command allows you to determine the maximum rows to be returned in the output table.

Example:

```
from apache-logs
| keep status code, response time, client ip, url
| where status_code >= 500 and status_code <= 599 and response_time >= 2000
| sort status_code desc
| limit 1000
```

You can also return the top three hosts based on their sum_bytes by host

```
from apache-logs
| keep status_code, response_time, client_ip, url
| where status_code >= 500 and status_code <= 599 and response_time >= 2000
| limit 1000
| stats sum_bytes = sum(bytes) by host
| limit 3
```

ES|QL Process Commands `eval`

- ! Fundamental Process Command !
- `eval` command allows you to calculate an expression and create a new field or column

Example:
You want to calculate the total price for each transaction after applying the discount from the logs of your ecommerce application.

Logs contain
`Item_price`
`quantity`
`discount_percent` (Discount applied to the total price, represented as a percentage)

```
from ecommerce-logs
| eval total_price_before_discount = item_price * quantity
| eval discount_amount = total_price_before_discount * discount_percent / 100
| eval total_price_after_discount = total_price_before_discount - discount_amount
```

ES|QL Process Commands eval

Input:

```
from ecommerce-logs
| eval total_price_before_discount = item_price * quantity
| eval discount_amount = total_price_before_discount * discount_percent / 100
| eval total_price_after_discount = total_price_before_discount - discount_amount
```

Output:

_time	item_price	quantity	total_price_before_discount	discount_amount	total_price_after_discount
2023-10-05 08:32:01	20	2	40	8	32
2023-10-05 09:14:15	50	1	50	10	40
2023-10-05 10:05:33	30	3	90	18	72

ES|QL Functions

- Functions can be used with `row`, `eval`, and `where` commands
- Too many to cover today, refer to Introduction to ES|QL class and documentation

ABS	CONCAT	GREATEST	MV_CONCAT	REPLACE	TAN	TO_RADIANS
ACOS	COS	IS_FINITE	MV_COUNT	RIGHT	TANH	TO_STRING
ASIN	COSH	IS_INFINITE	MV_DEDUPE	ROUND	TAU	TO_UNSIGNED_LONG
ATAN	DATE_EXTRACT	IS_NAN	MV_MAX	RTRIM	TO_BOOLEAN	TO_VERSION
ATAN2	DATE_FORMAT	LEAST	MV_MEDIAN	SIN	TO_DATETIME	TRIM
AUTO_BUCKET	DATE_PARSE	LEFT	MV_MIN	SINH	TO_DEGREES	
CASE	DATE_TRUNC	LENGTH	MV_SUM	SPLIT	TO_DOUBLE	
CEIL	E	LOG10	NOW	SQRT	TO_INTEGER	
CIDR_MATCH	ENDS_WITH	LTRIM	PI	STARTS_WITH	TO_IP	
COALESCE	FLOOR	MV_AVG	POW	SUBSTRING	TO_LONG	

ES|QL Functions   Operators

Create conditions from boolean expression that can be formed using

- Relational operators such as <,>,<=, = >, ==, and !=
- Boolean functions like `starts_with`
- Boolean expressions created with `eval`
- `like` to match strings using wildcards `?` and `*`
  - Example “`?*n`” matches `John`, `Ethan`, but not `Natalie`
- `rlike` to match strings using regular expressions
  - While computational expensive, `rlike` match patterns such as timestamps, and email address, etc.
  - `(?<![0-9.+-1)(?>![+-]?(?:(?:[0-9]+(?:\.[0-9]+)?)|(?:\.[0-9]+)))` matches decimal numbers
  - ES|QL uses a `grok` parser as shown previously
- `in` operator tests whether a literal or a field/column are members of a list of literals/values
- Boolean operators can be used in combination using, `and`, `or`, `not`

ES|QL Functions Numeric Functions

Example:

```
row x = sin(pi()/2, y = e(), z = round(3.5), w = floor(3.5)
```

Output:

x	y	z	w
1.0	2.718281828459045	4.0	3.0

ABS	COS	IS_INFINITE	ROUND
ACOS	COSH	IS_FINITE	SIN
ASIN	E	LOG10	SINH
ATAN	FLOOR	PI	TAN
ATAN2	IS_FINITE	POW	TANH

ES|QL Process Commands `drop`

- `drop` is similar to `keep` but excluding fields/columns in your query.
- This is useful if you want to return the majority of the fields/columns in your document without listing all of them in `keep`.
- You can also use wildcards to drop all columns that matches the patterns

```
from employees
| drop height*
```

ES|QL Process Commands `rename`

- `rename` is used to rename a field/column
- This is useful if you have different labeled columns but similar data.
- `Rename` is helpful for standardizing fields names, and improving clarity

Example:

You're analyzing network traffic logs which have been sourced from multiple logging systems. These logs contain fields that represent the same kind of data but are named differently because of the disparate systems. For instance, one system might log source IP addresses as `src_ip`, while another system might use `source_ip`.

```
from network-logs
| rename src_ip AS source_ip, dest_ip AS destination_ip
```

ES|QL Process Commands

Extracting data from structuring strings

- There are two processing commands that parse data from a string: `grok` and `dissect`

- `dissect` matches the string against a delimiter-based pattern, and extracts the specified keys as fields/columns.

- `grok` matches the string against patterns, based on regular expressions, and extracts the specified patterns as columns.

ES|QL Process Commands dissect

- The advantage of `dissect` over `grok` is its simplicity and speed because `dissect` does not use Regular Expressions
- To use `dissect` you need to describe the delimiter pattern embedded the substrings containing the data. You can refer to the [dissect processor documentation](#) for the syntax of dissect patterns.

Example

```
ROW a = "1953-01-23T12:15:00Z - some text - 127.0.0.1;"
| DISSECT a "%{Y}-%{M}-%{D}T%{h}:%{m}:%{s}Z - %{msg} - %{ip};"
| KEEP Y, M, D, h, m, s, msg, ip
```

Output

Y:keyword	M:keyword	D:keyword	h:keyword	m:keyword	s:keyword	msg:keyword	ip:keyword
1953	01	23	12	15	00	some text	127.0.0.1

elastic

ES|QL Process Commands `grok`

- The advantage of `grok` over `dissect` is its ability to match the string against a complex patterns and not just extract data found between delimiters
- `grok` will reject strings that do not follow the syntax given by regex while `dissect` will capture invalid strings
- To use `grok` you need to create a pattern using the named regular expressions that comes with `grok`

Example

```
ROW a = "1953-01-23T12:15:00Z 127.0.0.1 some.email@foo.com 42"
| GROK a "%{TIMESTAMP_ISO8601:date} %{IP:ip} %{EMAILADDRESS:email} %{NUMBER:num:int}"
| KEEP date, ip, email, num
```

Output

date:keyword	ip:keyword	email:keyword	num:integer
1953-01-23T12:15:00Z	127.0.0.1	some.email@foo.com	42

ES|QL Functions String Functions

Example:

```
row first_name = "Shay", last_name = "Banon", roles =
"Founder,CTO,Engineer", product = "Elasticsearch"
| eval full_name = concat(first_name, " ", last_name)
| eval roles = split(roles, ",")
| eval trim(product)
| keep full_name, roles, product
```

Output:

first_name	last_name	full_name	roles	product
Shay	Banon	Shay Banon	[Founder,CTO,Engineer]	Elasticsearch

CONCAT	SPLIT	SUBSTRING
LENGTH	STARTS_WITH	TRIM

ES|QL Functions Date Functions

Example:

```
row date_string = "2023-10-05"
| EVAL date1 = DATE_PARSE("yyyy-MM-dd", date_string)
| eval date2 = date_format("yyyy/MM/dd", date1)
| eval trunced_date1 = date_trunc(1 year, date1)
| eval year = date_extract("year", date1)
| keep date1, date2, trunced_date1, year
```

Output:

date1	date2	truncated_date1	year
2023-10-05T00:00:00.00Z	2023/10/5	2023-01-01T00:00:00.00Z	2023

DATE_EXTRACT	DATE_PARSE	NOW
DATE_FORMAT	DATE_TRUNC	

elastic

ES|QL Functions Conversion Functions

Example:

```
row long = [5013792, 2147483647, 501379200000]
| eval int = TO_INTEGER(long)
```

Output:

long:long	int:integer
[5013792, 2147483647, 501379200000]	[5013792, 2147483647]

TO_BOOLEAN	TO_DOUBLE	TO_LONG	TO_STRING
TO_DATETIME	TO_INTEGER	TO_RADIANS	TO_UNSIGNED_LONG
TO_DEGREES	TO_IP	TO_STRING	TO_VERSION

ES|QL
Aggregations

Aggregations are now Kibana Discover

elastic

ES|QL Aggregations stats…by

The `stats … by` processing command is used with aggregation functions.

- `stats … by` groups the rows of a table into buckets based on values of a given field/column or based on grouping generated by the `auto_buckets` function
- One or more column aggregation function can be applied to rows of each bucket

```
from kibana_sample_data_logs
| stats avg_memory = avg(memory)by machine.os
```

avg_memory	machine.os
126930	Win 8
214922.96296296295	ios
146498	Win 7
166642.35294117648	osx
228420	Win xp

elastic

ES|QL Aggregations `stats…by`

ES|QL supports the following aggregation functions

- `AVG`
- `COUNT`
- `COUNT_DISTINCT`
- `MAX`
- `MEDIAN`
- `MEDIAN_ABSOLUTE_DEVIATION`
- `MIN`
- `PERCENTILE`
- `SUM`

ES|QL Aggregations stats…by

ES|QL supports the following aggregation functions

- AVG
- COUNT
- COUNT_DISTINCT
- MAX
- MEDIAN
- MEDIAN_ABSOLUTE_DEVIATION
- MIN
- PERCENTILE
- SUM

Counts the number of values in a column
Duplicates are counted
`Count` single value column with no `nulls`

ES|QL Aggregations `stats…by`

ES|QL supports the following aggregation functions

- AVG
- COUNT
- COUNT_DISTINCT →
- MAX
- MEDIAN
- MEDIAN_ABSOLUTE_DEVIATION
- MIN
- PERCENTILE
- SUM

- Approximates the number of distinct values in a column
- Computing exact counts requires loading values into a set and returning its size which doesn't scale when working on high-cardinality sets and/or large values.
- This `count_distinct` function is based on the HyperLogLog++ algorithm, which counts based on the hashes of the values with some interesting properties:
- Configurable precision

ES|QL Aggregations `stats…by`

ES|QL supports the following aggregation functions

- `AVG`
- `COUNT`
- `COUNT_DISTINCT`
- `MAX`
- `MEDIAN`
- `MEDIAN_ABSOLUTE_DEVIATION` →
- `MIN`
- `PERCENTILE`
- `SUM`

- A measure of variability. Robust statistic that it is useful for describing data that may have outliers, or may not be normally distributed.
- It is calculated as the median of each data point’s deviation from the median of the entire sample. For a random variable X, the median absolute deviation is `median(|median(X) - Xi|)`.
- Like `PERCENTILE`, `MEDIAN_ABSOLUTE_DEVIATION` is usually approximate.
- MEDIAN_ABSOLUTE_DEVIATION is also non-deterministic. This means you can get slightly different results using the same data.

ES|QL Aggregations `stats…by`

ES|QL supports the following aggregation functions

- ○ `AVG`
- ○ `COUNT`
- ○ `COUNT_DISTINCT`
- ○ `MAX`
- ○ `MEDIAN`
- ○ `MEDIAN_ABSOLUTE_DEVIATION`
- ○ `MIN`
- ○ `PERCENTILE` →
- ○ `SUM`

- The value at which a certain percentage of observed values occur. If the 95th percentile is the value which is greater than 95% of the observed values and the 50th percentile is the `MEDIAN`.
- The algorithm used by the percentile metric is called `TDigest` (introduced by Ted Dunning in Computing Accurate Quantiles using T-Digests).
- `PERCENTILE` is also non-deterministic. This means you can get slightly different results using the same data.

elastic

ES|QL Aggregations `stats … by`

Used with `stats … by`, you can create a distance histogram

```
from kibana_sample_data_flights
| keep DistanceMiles, FlightDelayMin
| eval distance_ranges = auto_bucket(DistanceMiles, 20, 0, 20000)
| stats delay = avg(FlightDelayMin) by distance_ranges
| sort distance_ranges
```

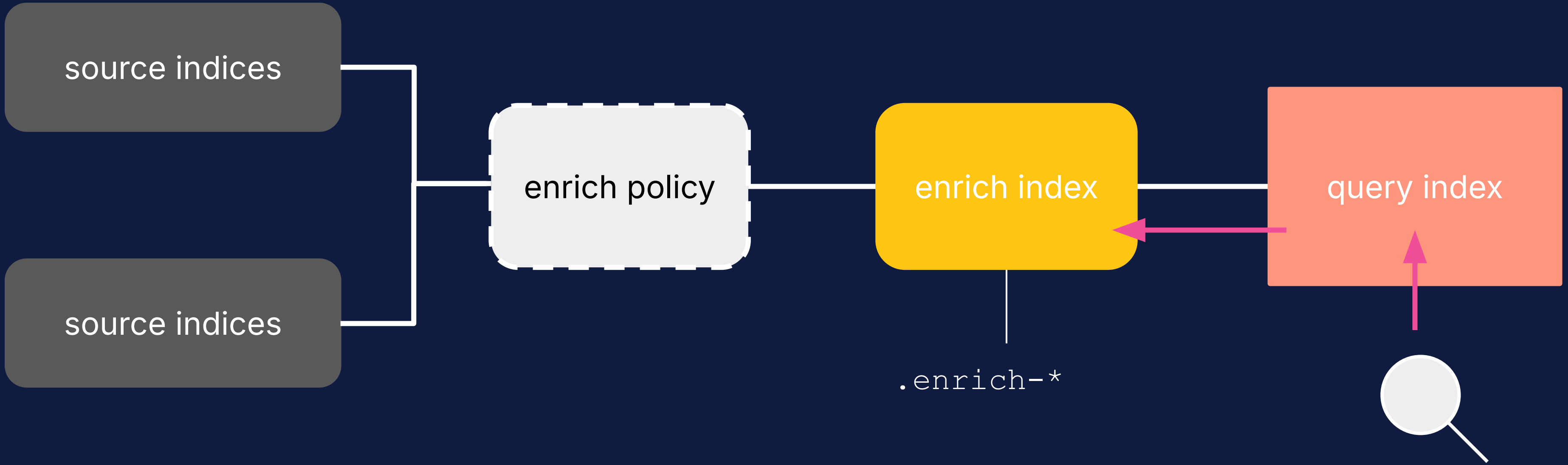
delay	distance_ranges
47.54573764110549	0
48.552223371251294	1000
49.66304347826087	2000
47.22154222766218	3000

ES|QL Enrich
-(lookups)

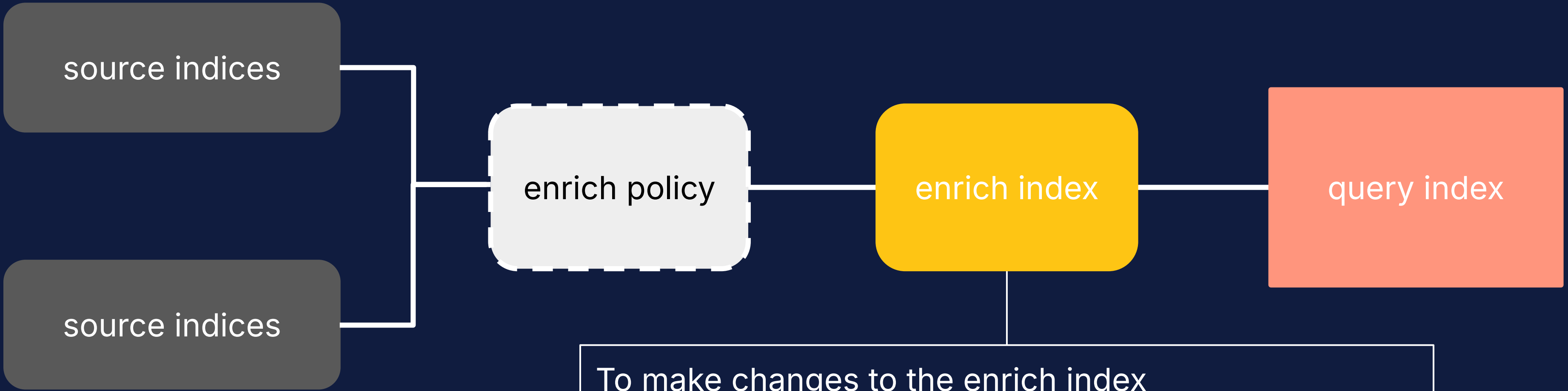
ES|QL Process Commands `enrich`

- `enrich` can add data from an Elasticsearch index to the output of a query
- Similar to ingest enrich, but it works at query time.
- You must first create an *enrich policy* in Elasticsearch. When executed, new index is created that will be used as a lookup table for the `enrich` process command.
- The enrich policy defines a match field (a key field) and a set of enrich fields.
- You can use enrich on remote clusters using Elastic's Cross Cluster Search

ES|QL `enrich` components



ES|QL `enrich` components



To make changes to the enrich index

- Delete the enrich policy
- Create a new policy with the updates
- If you are just making changes to the data but no the policy, you can re-execute the policy