



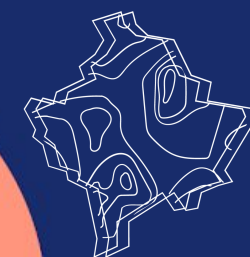
# Web mapping with Elasticsearch

Jorge Sanz | [jorge.sanz@elastic.co](mailto:jorge.sanz@elastic.co)

Craig Taverner | [craig.taverner@elastic.co](mailto:craig.taverner@elastic.co)

---

FOSS4G 2023 - June - Prizren, Kosovo



**FOSS4G**  
Prizren, 2023



# Forward Looking Statements and Non-GAAP Disclaimer



This presentation and the accompanying oral presentation contain forward-looking statements that involve substantial risk and uncertainties, which include, but are not limited to, our expected financial results for the fiscal quarter ending July 31, 2020 and the fiscal year ending April 30, 2021, our expectations regarding the impact of the COVID-19 pandemic, our customer base, potential market and growth opportunities, and our go-to-market strategy. These forward-looking statements are subject to the safe harbor provisions under the Private Securities Litigation Reform Act of 1995. In some cases, you can identify forward-looking statements because they contain words such as “may,” “will,” “should,” “would,” “expects,” “plans,” “anticipates,” “could,” “intends,” “target,” “projects,” “contemplates,” “believes,” “estimates,” “predicts,” “potential” or “continue” or the negative of these words or other similar terms or expressions that concern our expectations, strategy, plans or intentions. Our expectations and beliefs in light of currently available information regarding these matters may not materialize. Actual outcomes and results may differ materially from those contemplated by these forward-looking statements due to uncertainties, risks, and changes in circumstances, including but not limited to those related to: the impact of COVID-19 on our business, operations, hiring and financial results, and on businesses of our customers and partners, including the effect of governmental lockdowns, restrictions and new regulations; our future financial performance, including our expectations regarding our revenue, cost of revenue, gross profit or gross margin, operating expenses (which include changes in sales and marketing, research and development and general and administrative expenses), and our ability to achieve and maintain future profitability; our ability to continue to deliver and improve our offerings and successfully develop new offerings, including security-related product offerings and SaaS offerings; customer acceptance and purchase of our existing offerings and new offerings, including the expansion and adoption of our SaaS offerings; our ability to maintain and expand our user and customer base; the impact of foreign currency exchange rate and interest rate fluctuations on our results; our international expansion strategy; our operating results and cash flows; our beliefs and objectives for future operations; the sufficiency of our capital resources; our ability to successfully execute our go-to-market strategy and expand in our existing markets and into new markets; and general market, political, economic and business conditions (including developments and volatility arising from the COVID-19 pandemic).

Any additional or unforeseen effect from the COVID-19 pandemic may exacerbate these risks. Additional risks and uncertainties that could cause actual outcomes and results to differ materially are included in our filings with the Securities and Exchange Commission (the “SEC”), including the quarterly report on Form 10-Q for the quarter ended January 31, 2020 and any subsequent reports filed with the SEC. SEC filings are available on the Investor Relations section of Elastic’s website at [ir.elastic.co](http://ir.elastic.co) and the SEC’s website at [www.sec.gov](http://www.sec.gov). Elastic assumes no obligation to, and does not currently intend to, update any such forward-looking statements, except as required by law.

In addition to GAAP financial information, this presentation and the accompanying oral presentation include certain non-GAAP financial measures. See the Appendix for a reconciliation of all historical non-GAAP financial measures to their nearest GAAP equivalent.

# Agenda

What are we covering in the next two hours:

- Introductions: Elastic, Elasticsearch, and Kibana (for geo)
- Web mapping with Elasticsearch
  - Rendering documents and aggregated data
  - Searching and drilling down with dates and any text
  - Geospatial filtering

What are we not covering today:

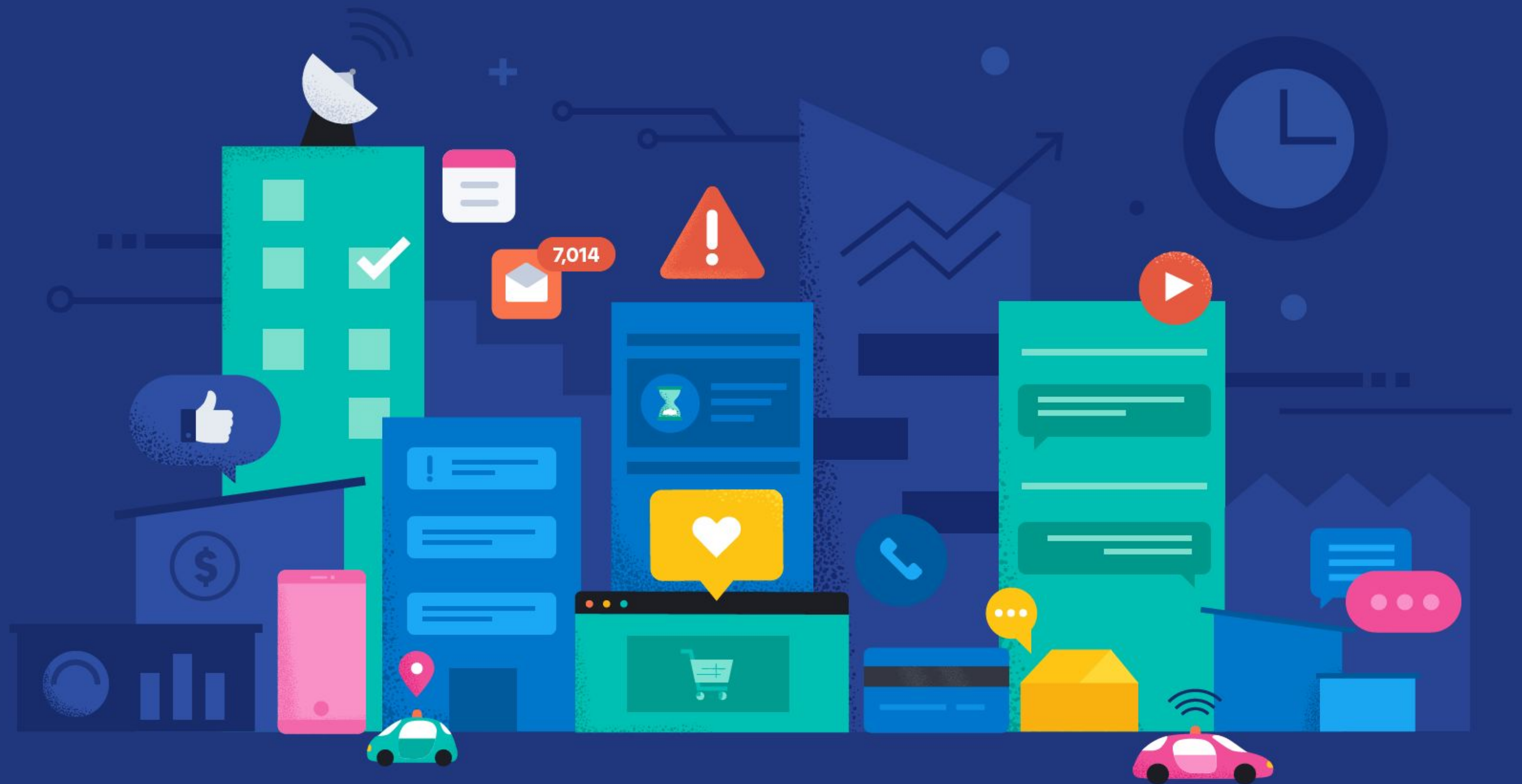
- Advanced web processing or UI (TypeScript, React, and such)
- Generative AI

# Elastic, a search company

Search. Observe. Protect



Today we live in an *always on world*



# A world characterized by **real challenges**



Content is becoming  
**harder** to find



Enterprise IT is becoming  
more **complex**



Cyber threats are becoming  
more **sophisticated**

**A world characterized by endless data**

**480EB**

1 EB = 1000 PB = 1,000,000 TB

Data produced  
daily by 2025

# Meet Elastic

**Elastic** helps the world's leading organizations **accelerate results that matter** by putting data to work with the **power of search**.





# Elastic at a glance

NYSE: ESTC



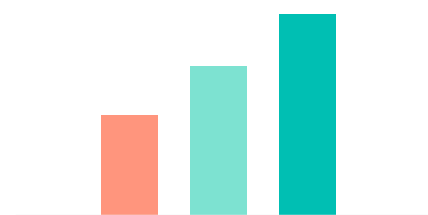
**Founded  
in 2012**



**3000+**  
employees



**50+**  
countries with  
employees



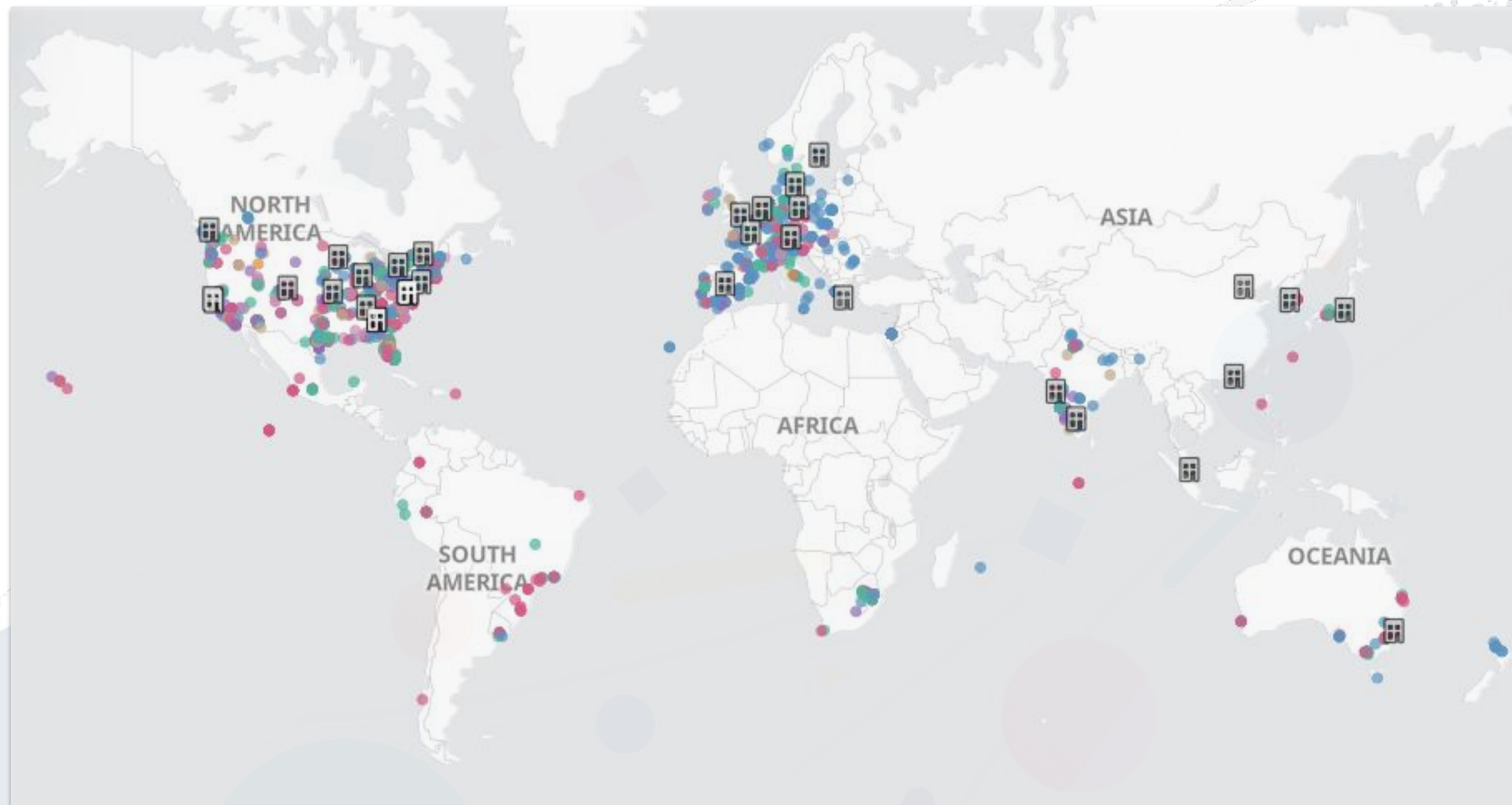
**17,900+**  
subscriptions




**54%**  
of Fortune 500  
companies trust Elastic

# Elastic at a glance

NYSE: ESTC



# The Elastic Search Platform *is for everyone*

| TECHNOLOGY  | FINANCE   | TELCO  | CONSUMER  | HEALTHCARE  | PUBLIC SECTOR   | AUTOMOTIVE /<br>TRANSPORTATION  | RETAIL  |
|---|---|--|---|---|---|---|---|
|    |    |    |    |    |    |    |    |
|    |    |    |    |    |    |    |    |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|   |   |  |   |   |   |   |  |



# Community

<https://github.com/elastic>

<https://ela.st/slack>

<https://discuss.elastic.co>

The screenshot shows the Elastic community forum interface. At the top, the Elastic logo is on the left, and navigation icons (YouTube, Discord, Slack, Search, Menu, Profile) are on the right. Below the header, there are filters for 'all categories' and 'all tags', followed by tabs for 'Categories', 'Latest', 'New (154)', 'Unread (21)', and 'Top'. A '+ New Topic' button is in the top right. The main content area is divided into two columns. The left column lists categories: 'Announcements' (1 topic, 1 unread), 'Elastic Stack' (322 topics, 19 unread), 'Elastic Enterprise Search' (5 topics, 2 new), and 'Elastic Observability' (23 topics, 14 new). The right column shows a 'Latest' feed of topics, including 'Notes on Using These Forums', 'Logstash pipeline graceful shutdown: потеря in-memory данных?', 'Collapse within top hit aggregation results', 'Drilldown is not working with Visualization', 'Do not show results on page load', 'Custom transactions in checkout process', 'How to view sql queries in APM', and 'Installation seems to hang'.

| Category  | Topics                             | Latest  |
|---|------------------------------------|---|
| <b>Announcements</b><br>Release and security announcements and other bits about all of our Elastic products that we think will be useful to everyone.<br>■ Security Announcements ■ Community Ecosystem 1 unread  | 1 / week<br>1 unread<br>1 new      | <b>Notes on Using These Forums</b> 2<br>■ Meta Elastic Apr 2017   |
| <b>Elastic Stack</b><br>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 topic under the relevant product category - Elasticsearch, Kibana, Beats, Logstash.<br>■ Elasticsearch 4 unread 59 new ■ Kibana 14 unread 32 new<br>■ Beats 20 new ■ Logstash 1 unread 18 new | 322 / week<br>19 unread<br>129 new | <b>Logstash pipeline graceful shutdown: потеря in-memory данных?</b> 0<br>■ Вопросы на русском языке 4m |
| <b>Elastic Enterprise Search</b><br>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.<br>■ App Search 2 new ■ Site Search ■ Workplace Search   | 5 / week<br>2 new                  | <b>Collapse within top hit aggregation results</b> 0<br>■ Elasticsearch 5m                              |
| <b>Elastic Observability</b><br>Bring your logs, infrastructure and availability metrics, and APM traces together at scale in a   | 23 / week<br>14 new                | <b>Drilldown is not working with Visualization</b> 2<br>■ Kibana 9m                                     |
|   |                                    | <b>Do not show results on page load</b> 0<br>■ App Search 20m   |
|   |                                    | <b>Custom transactions in checkout process</b> 0<br>■ Elasticsearch 21m                                 |
|   |                                    | <b>How to view sql queries in APM</b> 6<br>■ APM dotnet 23m   |
|   |                                    | <b>Installation seems to hang</b> 3   |

# The Elastic Search Platform

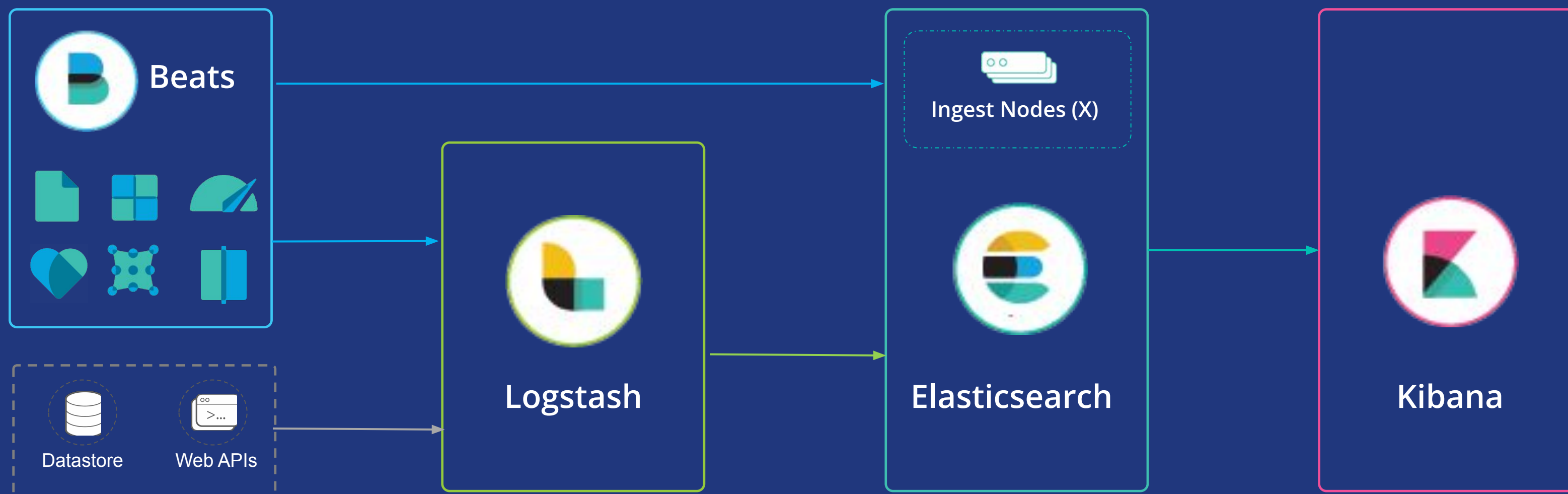


# Elasticsearch and Kibana intro



# Elastic Stack

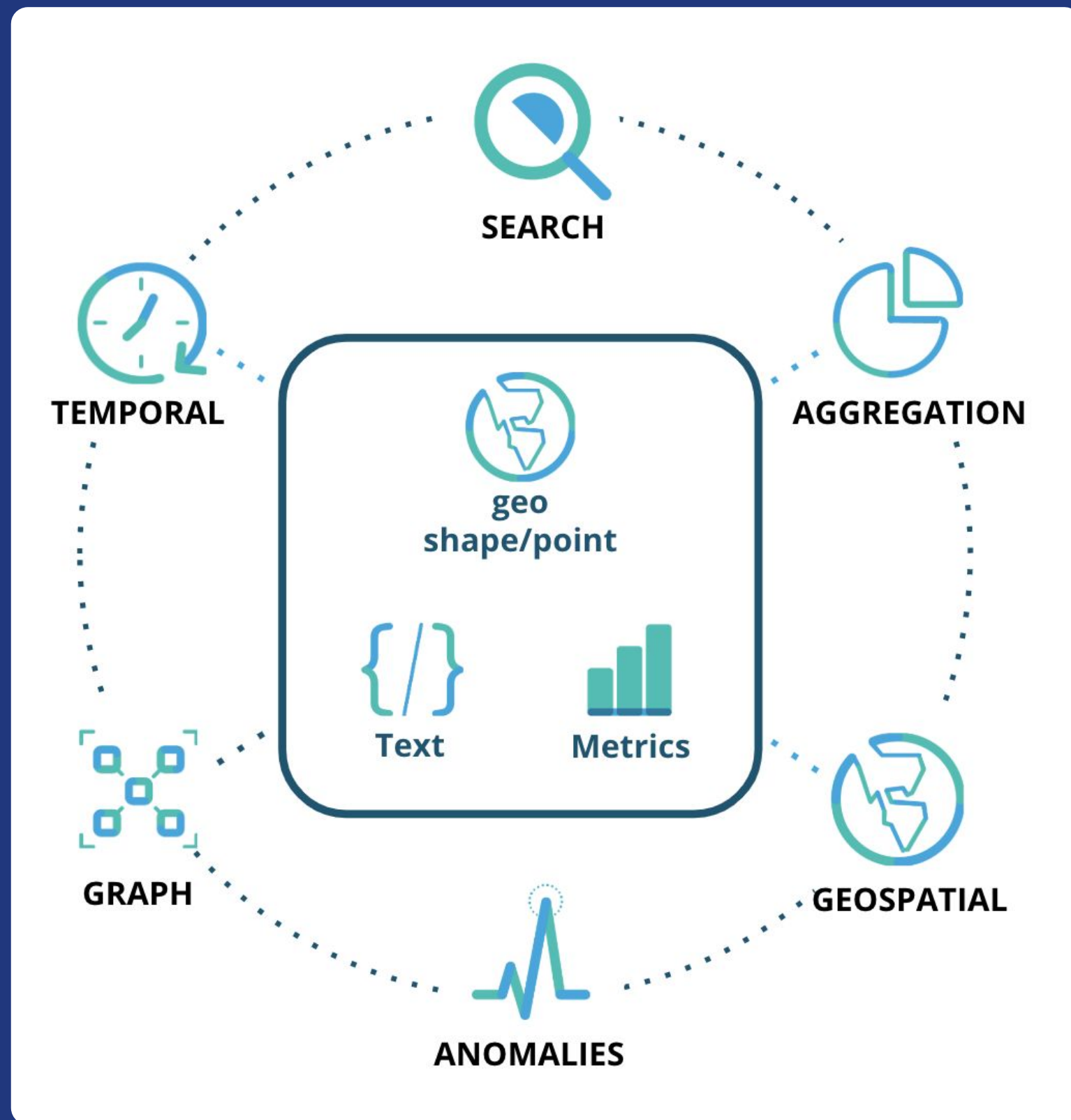
Ingest, Store, Search, Visualise



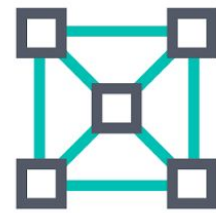


# Elasticsearch

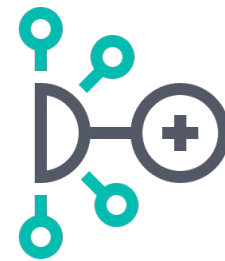
All data is welcome



# Elasticsearch components



Cluster



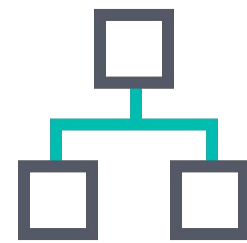
Node



Shard



Index



Mapping





Document



Field



# Communicating with Elasticsearch





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

```
~  
→ curl -s --user "${ELASTIC_USER}:${ELASTIC_PASSWORD}" "${ELASTIC_HOST}/" | jq  
{  
  "took": 1, "timed_out": false, "_source": true, "hits": {  
    "total": 20, "max_score": 1, "hits": [  
      {  
        "_type": "location",  
        "_id": "122.71_10.668",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 122.71,  
            "lat": 10.668,  
            "country": "Philippines",  
            "name": "East Valencia"  
          },  
          "unlocodename": "East Valencia",  
          "name": "East Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "30.27718_-23.87011",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 30.27718,  
            "lat": -23.87011,  
            "country": "South Africa",  
            "name": "Valencia Estate"  
          },  
          "unlocodename": "Valencia Estate",  
          "name": "Valencia Estate"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-87.45963_19.69255",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -87.45963,  
            "lat": 19.69255,  
            "country": "Mexico",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-81.41667_8.06667",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -81.41667,  
            "lat": 8.06667,  
            "country": "Panama",  
            "name": "La Valencia"  
          },  
          "unlocodename": "La Valencia",  
          "name": "La Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-61.19993_10.64988",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -61.19993,  
            "lat": 10.64988,  
            "country": "Trinidad And Tobago",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-103.4128_26.29734",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -103.4128,  
            "lat": 26.29734,  
            "country": "Mexico",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-97.90795_21.584",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -97.90795,  
            "lat": 21.584,  
            "country": "Mexico",  
            "name": "La Valencia"  
          },  
          "unlocodename": "La Valencia",  
          "name": "La Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-97.55388_18.65448",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -97.55388,  
            "lat": 18.65448,  
            "country": "Mexico",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-75.11332_9.13451",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -75.11332,  
            "lat": 9.13451,  
            "country": "Colombia",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-76.6136_2.44189",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -76.6136,  
            "lat": 2.44189,  
            "country": "Colombia",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-78.4_-0.36667",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -78.4,  
            "lat": -0.36667,  
            "country": "Ecuador",  
            "name": "Hacienda Valencia"  
          },  
          "unlocodename": "Hacienda Valencia",  
          "name": "Hacienda Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-61.1668_10.68233",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -61.1668,  
            "lat": 10.68233,  
            "country": "Trinidad And Tobago",  
            "name": "Ward of Valencia"  
          },  
          "unlocodename": "Ward of Valencia",  
          "name": "Ward of Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-102.35591_29.33355",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -102.35591,  
            "lat": 29.33355,  
            "country": "Mexico",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "125.0_7.95",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 125.0,  
            "lat": 7.95,  
            "country": "Philippines",  
            "name": "City of Valencia"  
          },  
          "unlocodename": "City of Valencia",  
          "name": "City of Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "-109.80707_29.09612",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": -109.80707,  
            "lat": 29.09612,  
            "country": "Mexico",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "124.19428_13.58267",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 124.19428,  
            "lat": 13.58267,  
            "country": "Philippines",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "123.62489_10.14994",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 123.62489,  
            "lat": 10.14994,  
            "country": "Philippines",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "123.39093_9.7588",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 123.39093,  
            "lat": 9.7588,  
            "country": "Philippines",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "121.0378_14.6104",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 121.0378,  
            "lat": 14.6104,  
            "country": "Philippines",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      },  
      {  
        "_type": "location",  
        "_id": "121.6537_14.065",  
        "score": 1,  
        "source": {  
          "location": {  
            "lon": 121.6537,  
            "lat": 14.065,  
            "country": "Philippines",  
            "name": "Valencia"  
          },  
          "unlocodename": "Valencia",  
          "name": "Valencia"  
        }  
      }  
    ]  
  }  
}
```

curl -s --user "\${ELASTIC\_USER}:\${ELASTIC\_PASSWORD}" 0,07s user 0,00s system 11% cpu 0,635 total  
jq -c ".hits.hits[].\_source | { g: .location, c: .UNLOCODENAME, n: .Name}" 0,02s user 0,00s system 11% cpu 0,635 total



# 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 and circle
  - Inserted with GeoJSON or WKT notation
- `point` , `shape` 
  - Supports any **cartesian** geometry type
  - Inserted with GeoJSON or WKT notation

# API for Vector tiles

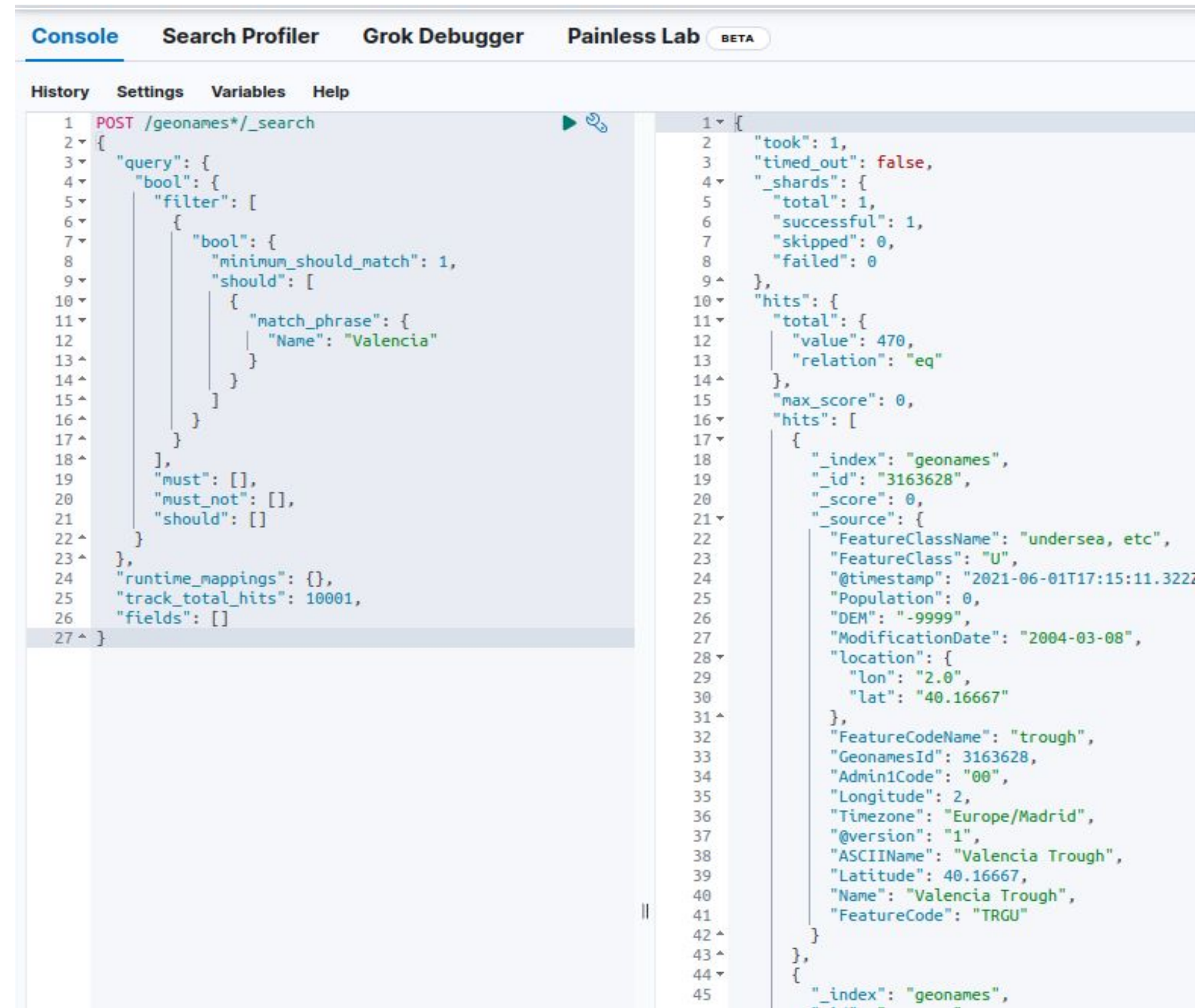
Integrate in to your own system

## Elasticsearch search API

- Modern, **REST**-based API
- **JSON** is the default output format

## Elasticsearch Vector Tiles API

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



The screenshot displays the Elasticsearch DevTools interface. The top navigation bar includes 'Console', 'Search Profiler', 'Grok Debugger', and 'Painless Lab BETA'. Below this, the 'History' tab is active, showing a list of queries. The selected query is a POST request to `/geonames*/_search`. The query body is a JSON object with a 'query' section containing a 'bool' filter. This filter has a 'filter' array with a 'bool' object that specifies 'minimum\_should\_match': 1 and a 'should' array containing a 'match\_phrase' query for 'Name': 'Valencia'. The response body is a JSON object with fields like 'took', 'timed\_out', '\_shards', 'total', 'successful', 'skipped', 'failed', 'hits', 'max\_score', and a single hit object. The hit object contains detailed information about a 'Valencia Trough' feature, including its ID, source, feature class, timestamp, population, DEM, modification date, location (longitude and latitude), feature code name, Geonames ID, admin code, longitude, timezone, version, ASCII name, latitude, name, and feature code.

```
1 POST /geonames*/_search
2 {
3   "query": {
4     "bool": {
5       "filter": [
6         {
7           "bool": {
8             "minimum_should_match": 1,
9             "should": [
10              {
11                "match_phrase": {
12                  "Name": "Valencia"
13                }
14              }
15            ]
16          }
17        }
18      ],
19      "must": [],
20      "must_not": [],
21      "should": []
22    }
23  },
24  "runtime_mappings": {},
25  "track_total_hits": 10001,
26  "fields": []
27 }
```

```
1 {
2   "took": 1,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": {
12      "value": 470,
13      "relation": "eq"
14    },
15    "max_score": 0,
16    "hits": [
17      {
18        "_index": "geonames",
19        "_id": "3163628",
20        "_score": 0,
21        "_source": {
22          "FeatureClassName": "undersea, etc",
23          "FeatureClass": "U",
24          "@timestamp": "2021-06-01T17:15:11.322Z",
25          "Population": 0,
26          "DEM": "-9999",
27          "ModificationDate": "2004-03-08",
28          "location": {
29            "lon": "2.0",
30            "lat": "40.16667"
31          },
32          "FeatureCodeName": "trough",
33          "GeonamesId": 3163628,
34          "Admin1Code": "00",
35          "Longitude": 2,
36          "Timezone": "Europe/Madrid",
37          "@version": "1",
38          "ASCIIName": "Valencia Trough",
39          "Latitude": 40.16667,
40          "Name": "Valencia Trough",
41          "FeatureCode": "TRGU"
42        }
43      },
44      {
45        "_index": "geonames",
```



# Search

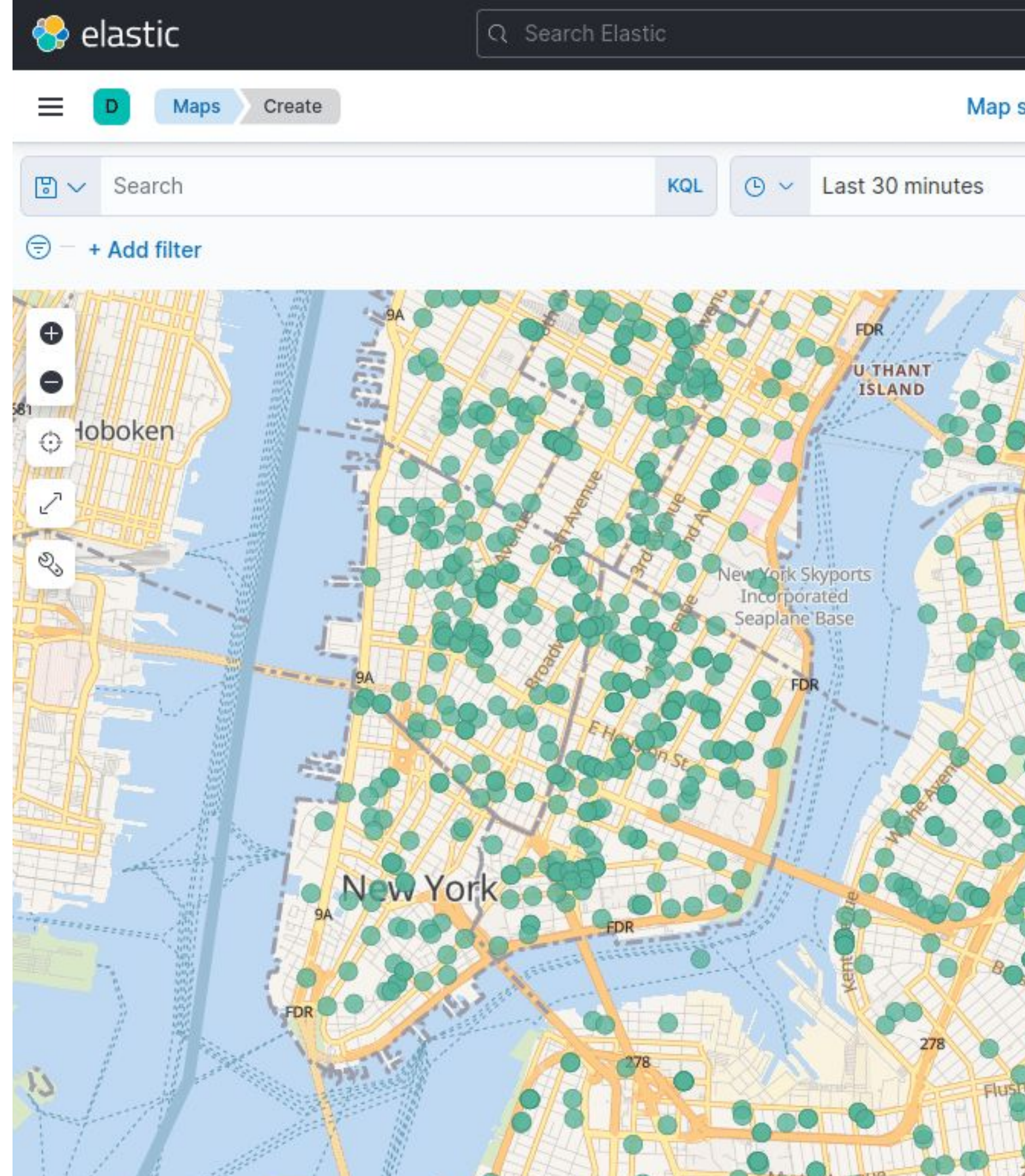
Filter documents with geospatial relationships

## Geo Filters

- Bounding box
- Point and radius
- Polygon
- An indexed geo\_shape

Plus every other Elasticsearch filter

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





# Aggregate

## Geo Bucket

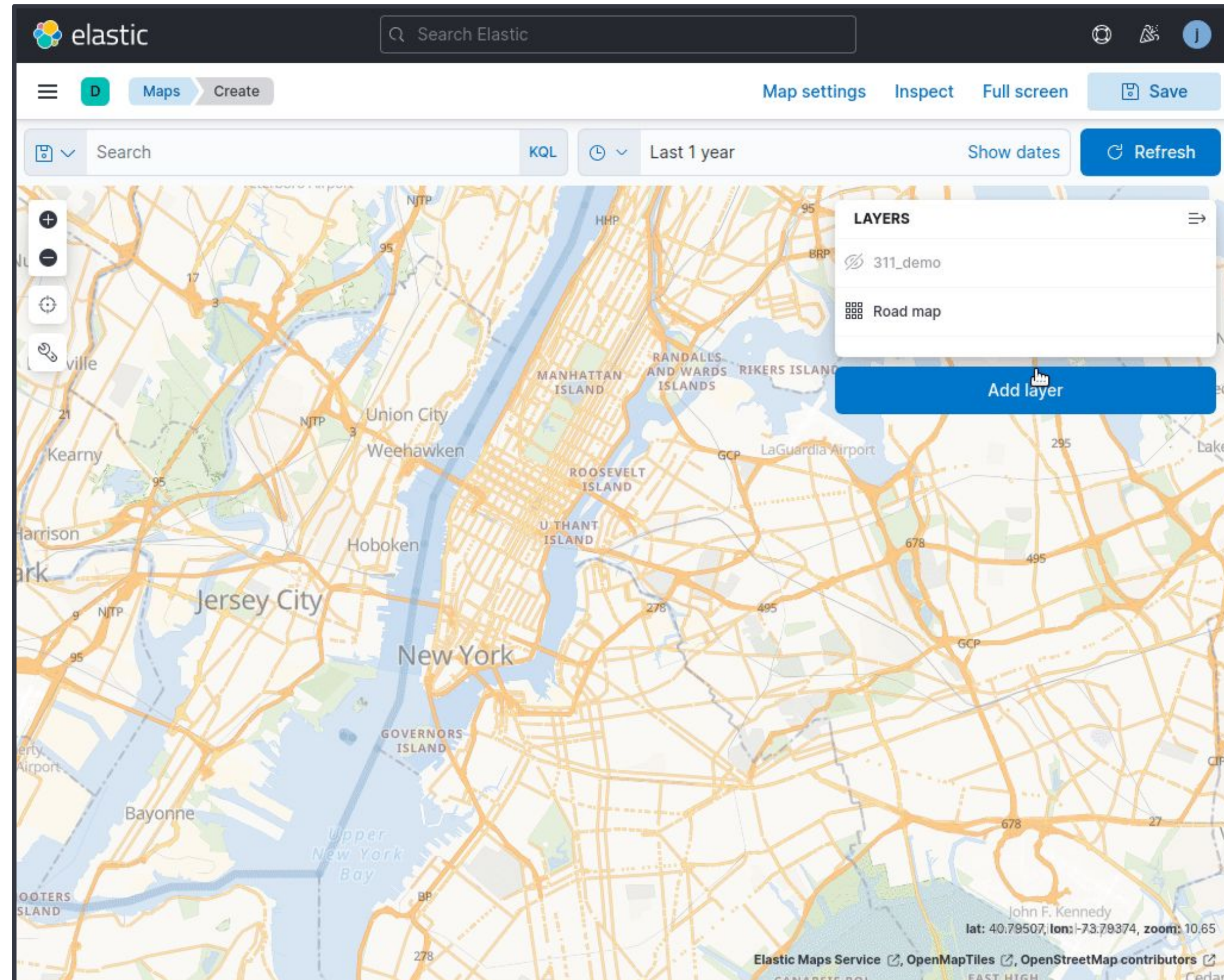
- Distance (rings) 📖
- Hash 📖
- Geotile 📖
- Hex Grid 📖

## Geo Metric

- Centroid 📖
- Bounds 📖
- Geoline 📖

## Aggregate non-geo using geo filters

- Huge range of aggregations 📖

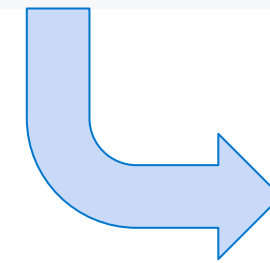




# Ingest

```
PUT airports
{
  "mappings": {
    "properties": {
      "coords": {
        "type": "geo_point"
      },
      "abbrev": {
        "type": "keyword"
      },
      "name": {
        "type": "text"
      },
      "type": {
        "type": "keyword"
      }
    }
  }
}
```

```
POST airports/_doc
{
  "coords": [75.9570722, 30.8503599],
  "name": "Sahnewal",
  "abbrev": "LUH",
  "type": "small"
}
```



```
{
  "_index": "airports",
  "_id": "CyQ-kIcB0vAUcSbABKjz",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 18,
  "_primary_term": 1
}
```



```
POST _bulk
{ "index" : { "_index" : "airports", "_id" : "1" } }
{"coords":[75.9570722,30.8503599],"name":"Sahnewal","abbrev":"LUH","type":"small"}
{ "index" : { "_index" : "airports", "_id" : "2" } }
{"coords":[75.9330598,17.6254152],"name":"Solapur","abbrev":"SSE","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "3" } }
{"coords":[85.323597,23.3177246],"name":"Birsa Munda","abbrev":"IXR","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "4" } }
{"coords":[48.7471065,31.3431586],"name":"Ahwaz","abbrev":"AWZ","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "5" } }
{"coords":[78.2172187,26.2854877],"name":"Gwalior","abbrev":"GWL","type":"mid and military"}
{ "index" : { "_index" : "airports", "_id" : "6" } }
{"coords":[42.9710963,14.7552534],"name":"Hodeidah Int'l","abbrev":"HOD","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "7" } }
{"coords":[75.8092915,22.7277492],"name":"Devi Ahilyabai Holkar Int'l","abbrev":"IDR","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "8" } }
{"coords":[73.8105675,19.9660206],"name":"Gandhinagar","abbrev":"ISK","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "9" } }
{"coords":[76.8017261,30.6707249],"name":"Chandigarh Int'l","abbrev":"IXC","type":"major and military"}
{ "index" : { "_index" : "airports", "_id" : "10" } }
{"coords":[75.3958433,19.867297],"name":"Aurangabad","abbrev":"IXU","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "11" } }
{"coords":[72.9878191,31.3627435],"name":"Faisalabad Int'l","abbrev":"LYP","type":"mid and military"}
{ "index" : { "_index" : "airports", "_id" : "12" } }
{"coords":[73.3163595,54.9576483],"name":"Omsk Tsentralny","abbrev":"OMS","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "13" } }
{"coords":[82.6671525,55.0095847],"name":"Novosibirsk Tolmachev","abbrev":"OVB","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "14" } }
{"coords":[35.3018729,47.8732636],"name":"Zaporozhye Int'l","abbrev":"OZH","type":"mid and military"}
{ "index" : { "_index" : "airports", "_id" : "15" } }
{"coords":[101.4465693,0.4646009],"name":"Simpang Tiga","abbrev":"PKU","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "16" } }
{"coords":[145.2439803,14.1717713],"name":"Rota Int'l","abbrev":"ROP","type":"mid"}
{ "index" : { "_index" : "airports", "_id" : "17" } }
{"coords":[73.4084965,61.3401672],"name":"Surgut","abbrev":"SGC","type":"mid"}
```

```
{
  "took": 37,
  "errors": false,
  "items": [
    {
      "index": {
        "_index": "airports",
        "_id": "1",
        "_version": 2,
        "result": "updated",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "_seq_no": 19,
        "_primary_term": 1,
        "status": 200
      }
    },
    {
      "index": {
        "_index": "airports",
        "_id": "2",
        "_version": 2,
        "result": "updated",
        "_shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "_seq_no": 20,
        "_primary_term": 1,
        "status": 200
      }
    },
    {
      "index": {
        "_index": "airports",
```



# More ways to add data

In addition to adding [integrations](#), you can try our sample data or upload your own data.

Sample data

Upload file

## airports.csv

### Import data

Simple

Advanced

Index name

index name

☒

 Create data view

Data view name

Combined fields

+ Add combined field

Index settings

1 {

2 "number\_of\_shards": 1

3 }

Mappings

1 {

2 "properties": {

3 "column1": {

4 "type": "keyword"

5 },

6 "column10": {

7 "type": "keyword"

8 },

9 "column11": {

10 "type": "keyword"

11 },

12 "column12": {

13 "type": "keyword"

14 },

15 "column13": {

16 "type": "text"

17 },

18 "column14": {

19 "type": "keyword"

Ingest pipeline

1 {

2 "description": "Ingest pipeline created by text structure finder",

3 "processors": [

4 {

5 "csv": {

6 "field": "message",

7 "target\_fields": [

8 "column1",

9 "column2",

10 "column3",

11 "column4",

12 "column5",

13 "column6",

14 "column7",

15 "column8",

16 "column9",

17 "column10",

18 "column11"

Import



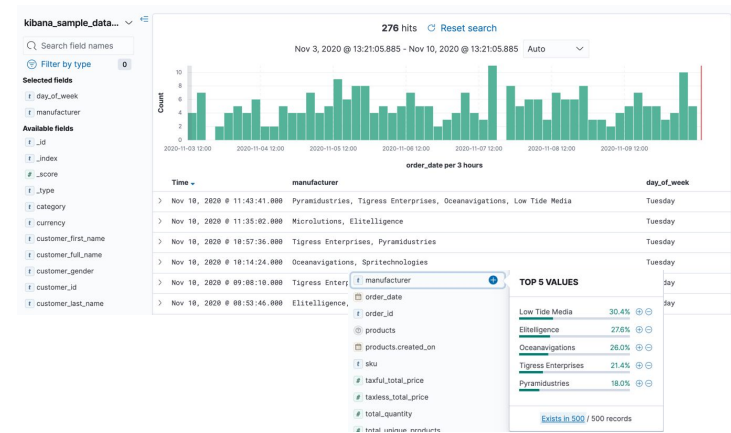


# Kibana

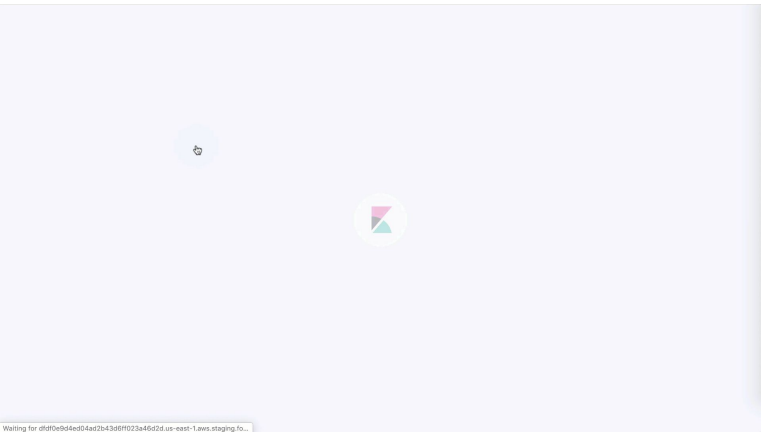
Some basic concepts about Kibana



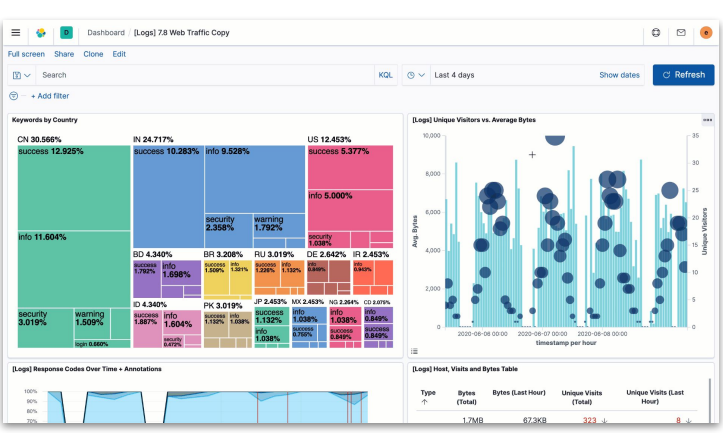
# Data Analysis with Kibana



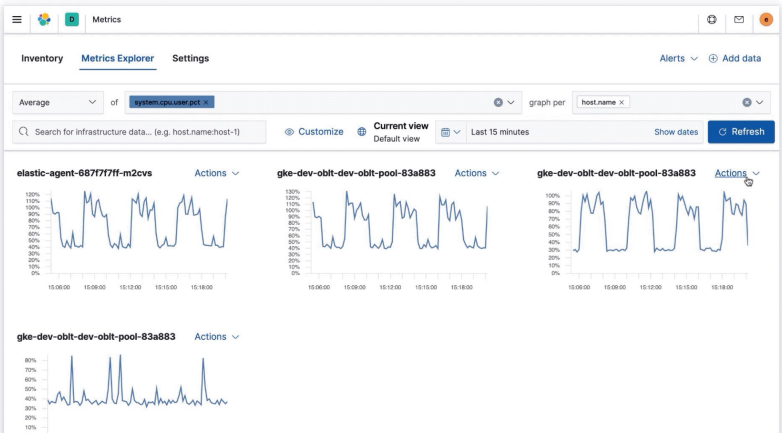
Discover



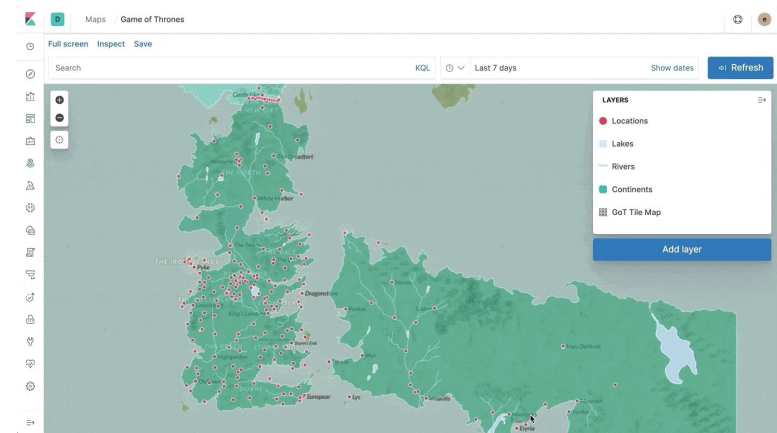
Lens



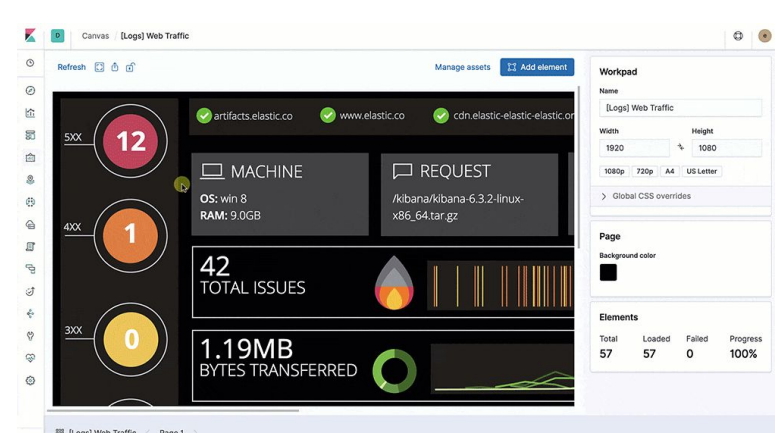
Dashboards & DrillDown



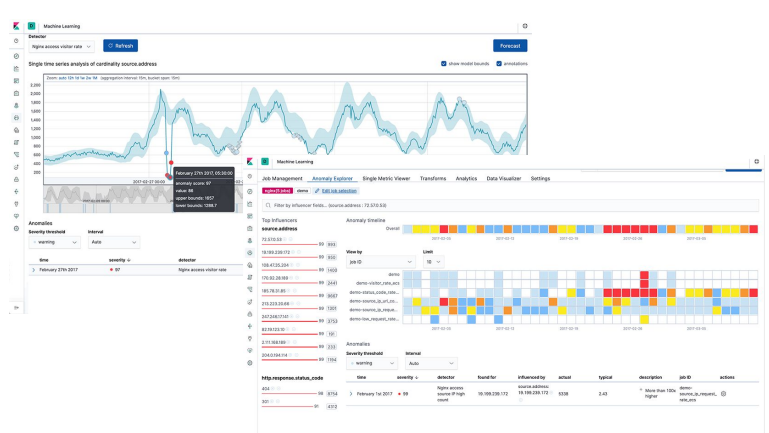
Alerting & Actions



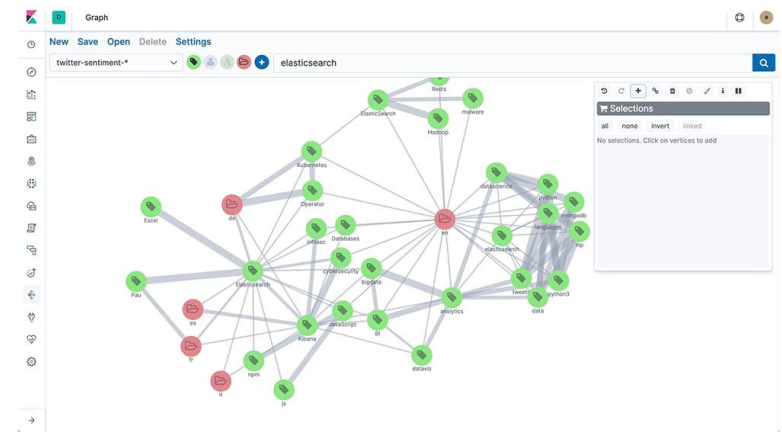
Maps



Canvas



Machine Learning



Graph

and much more ...

# Who uses Kibana?

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





# Developer Tools

## Console

Allows to run Elasticsearch queries with autocomplete, code formatting, history, etc.

## Search profiler

Shows statistics about query performance.

## Grok debugger

Helps creating grok expressions for Logstash.

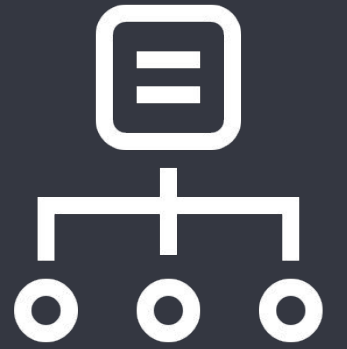
## Painless lab

An environment to test painless scripts.

The screenshot displays the Elastic Developer Tools interface. At the top, the Elastic logo and a search bar are visible. Below the navigation bar, the 'Console' tab is active, showing a REST client interface. The left pane contains a REST client request: a GET request to `flight_tracking*/_search` with a query body. The right pane shows the JSON response, which includes search statistics and two hits. The first hit is for a flight with ID `flcStIAB4XM30LHftGv_` and the second for `jFcTtIAB4XM30LHfr4QC`. The interface also shows tabs for 'Search Profiler', 'Grok Debugger', and 'Painless Lab' (marked as BETA). A status bar at the top right indicates '200 - OK' and '1240 ms'.

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

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



# Data Views

- Logic component that **gathers** indices using a name **pattern**
  - `my_application_logs_*`
- Defines field **formatters**: number, currency, image, URL, ...
- Defines **temporal field** for filtering (optional)
- **Runtime fields** for query time computations



# Discover

- Quick **exploration** tool
- **Time range** and automatic **refresh\***
- **Search bar** using Kibana Query Language or Lucene\*
- **Filters\***
- Table view with custom **columns**
- Field **statistics**
- **Inspect** tool: statistics, complete query and response
- **Save** your search to be used later on dashboards

\* shared UI with other Kibana applications



New Tab

← → ↺ 🔍

Discover ✓

Options New Open Share Alerts Inspect Save

GHCN Observations

🔍 element : "PRCP" and value >= 40

📅 ⌵ Refresh

🔍 Search field names

0

Popular fields 3

element id value

Available fields 10


@timestamp date element id m\_flag message obs\_time q\_flag s\_flag value

Empty fields 0

Meta fields 3

4,371,380 hits

Break down by Select field



Jan 1, 1900 @ 00:00:00.000 - Jan 1, 1911 @ 00:00:00.000 (interval: Auto - 30 days)

Documents

Field statistics

1 field sorted

|   | ↓ date 🕒                   | Document  |
|---|----------------------------|---|
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.508 date Jan 1, 1911 @ 00:00:00.000 id USC00508503 obs_time 1600 s_flg |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.541 date Jan 1, 1911 @ 00:00:00.000 id USC00510190 s_flag 6 value 325  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.549 date Jan 1, 1911 @ 00:00:00.000 id USC00510840 s_flag 6 value 406  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.549 date Jan 1, 1911 @ 00:00:00.000 id USC00510905 s_flag 6 value 97 _ |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.549 date Jan 1, 1911 @ 00:00:00.000 id USC00510999 s_flag 6 value 310  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.555 date Jan 1, 1911 @ 00:00:00.000 id USC00511460 s_flag 6 value 127  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.556 date Jan 1, 1911 @ 00:00:00.000 id USC00511484 s_flag 6 value 64 _ |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.641 date Jan 1, 1911 @ 00:00:00.000 id USC00512121 s_flag 6 value 264  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.641 date Jan 1, 1911 @ 00:00:00.000 id USC00512156 s_flag 6 value 109  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.631 date Jan 1, 1911 @ 00:00:00.000 id USC00511850 s_flag 6 value 254  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.631 date Jan 1, 1911 @ 00:00:00.000 id USC00511864 s_flag 6 value 165  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.641 date Jan 1, 1911 @ 00:00:00.000 id USC00511930 s_flag 6 value 246  |
| 🔗 | Jan 1, 1911 @ 00:00:00.000 | element PRCP @timestamp Oct 3, 2022 @ 23:08:05.667 date Jan 1, 1911 @ 00:00:00.000 id USC00512182 s_flg 6 value 58 _  |

stic



# Lens

## Your data in front of you

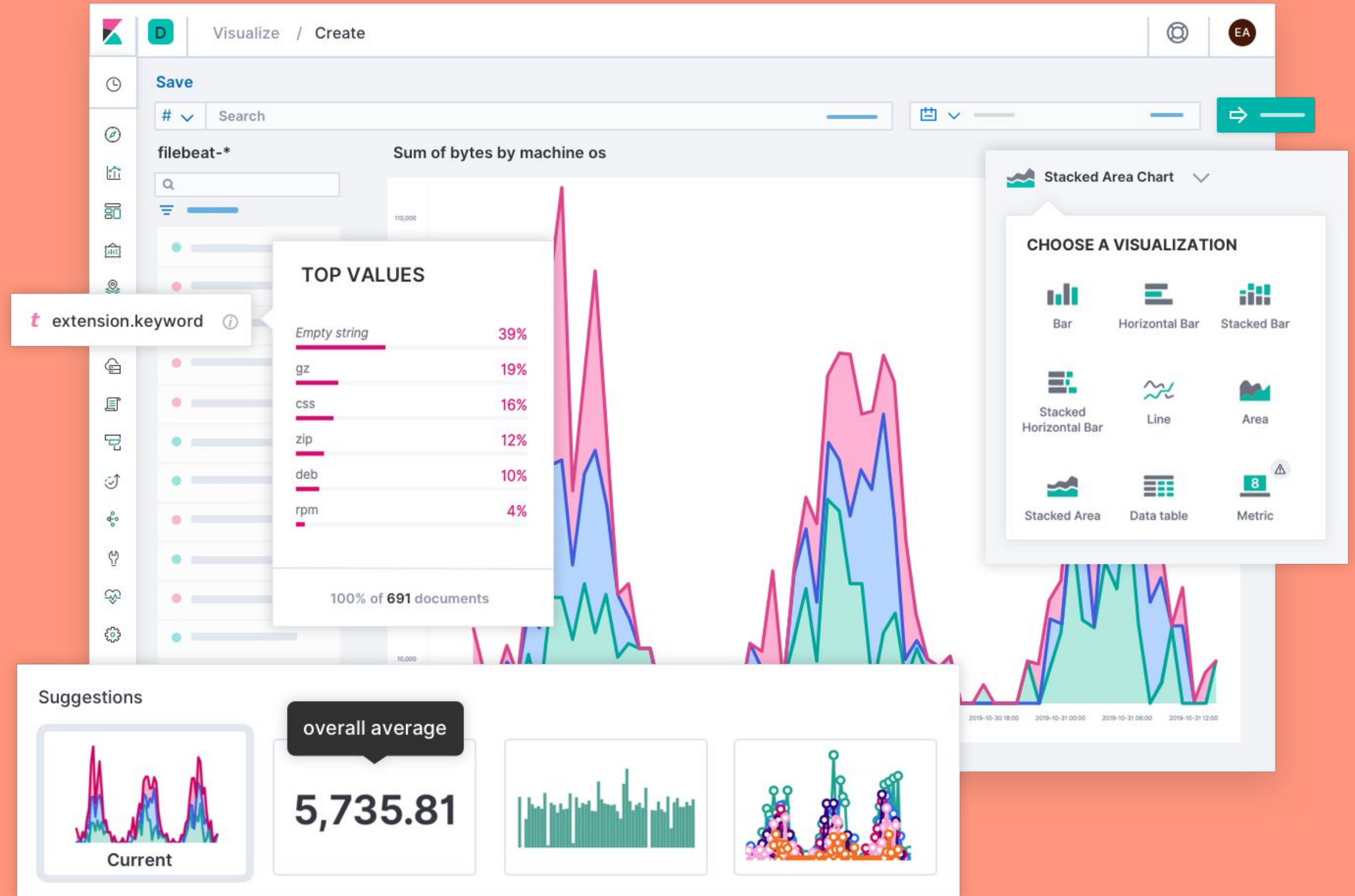
Explore your fields with a single click

## 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

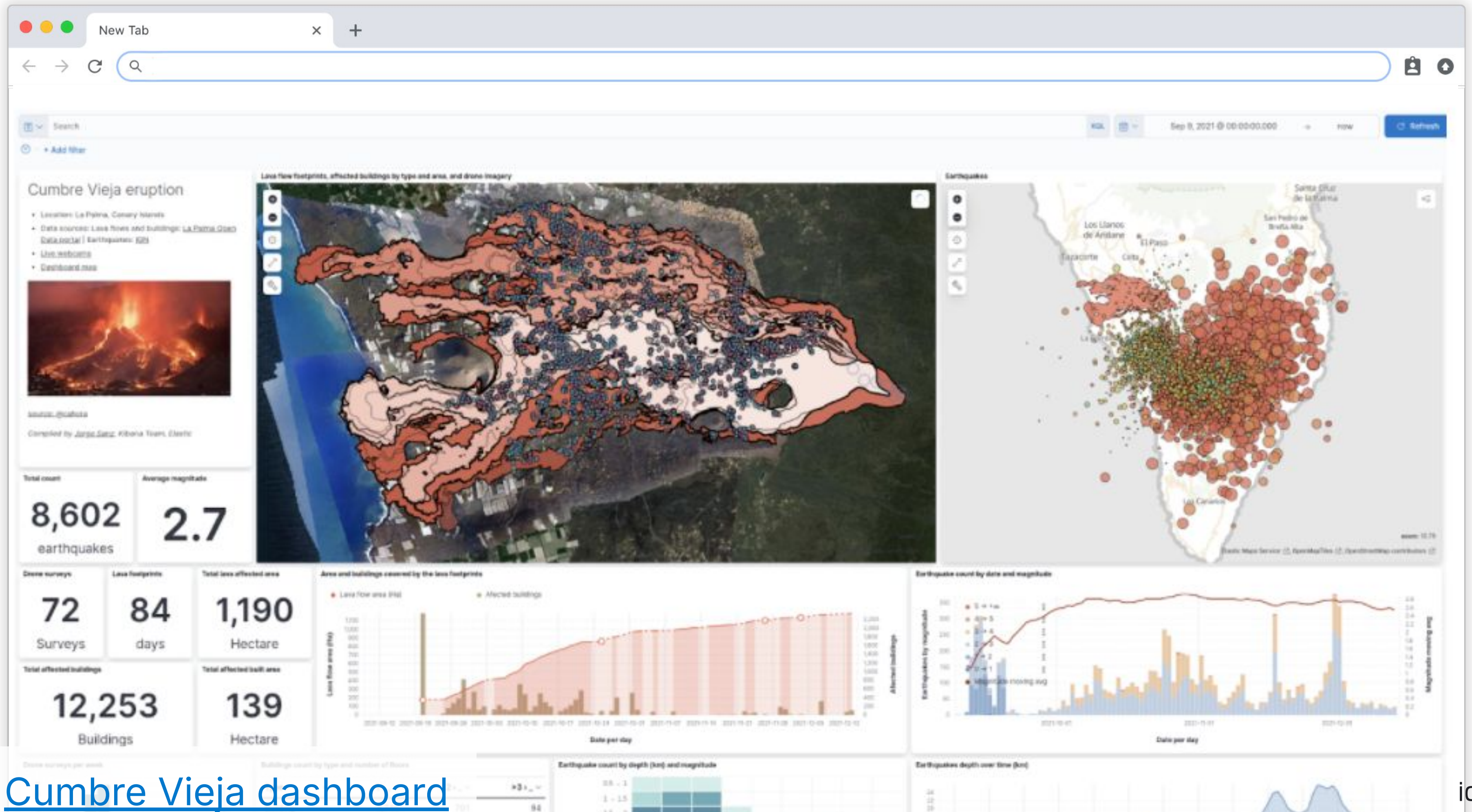




# Dashboards

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





Cumbre Vieja dashboard



# Elastic Maps

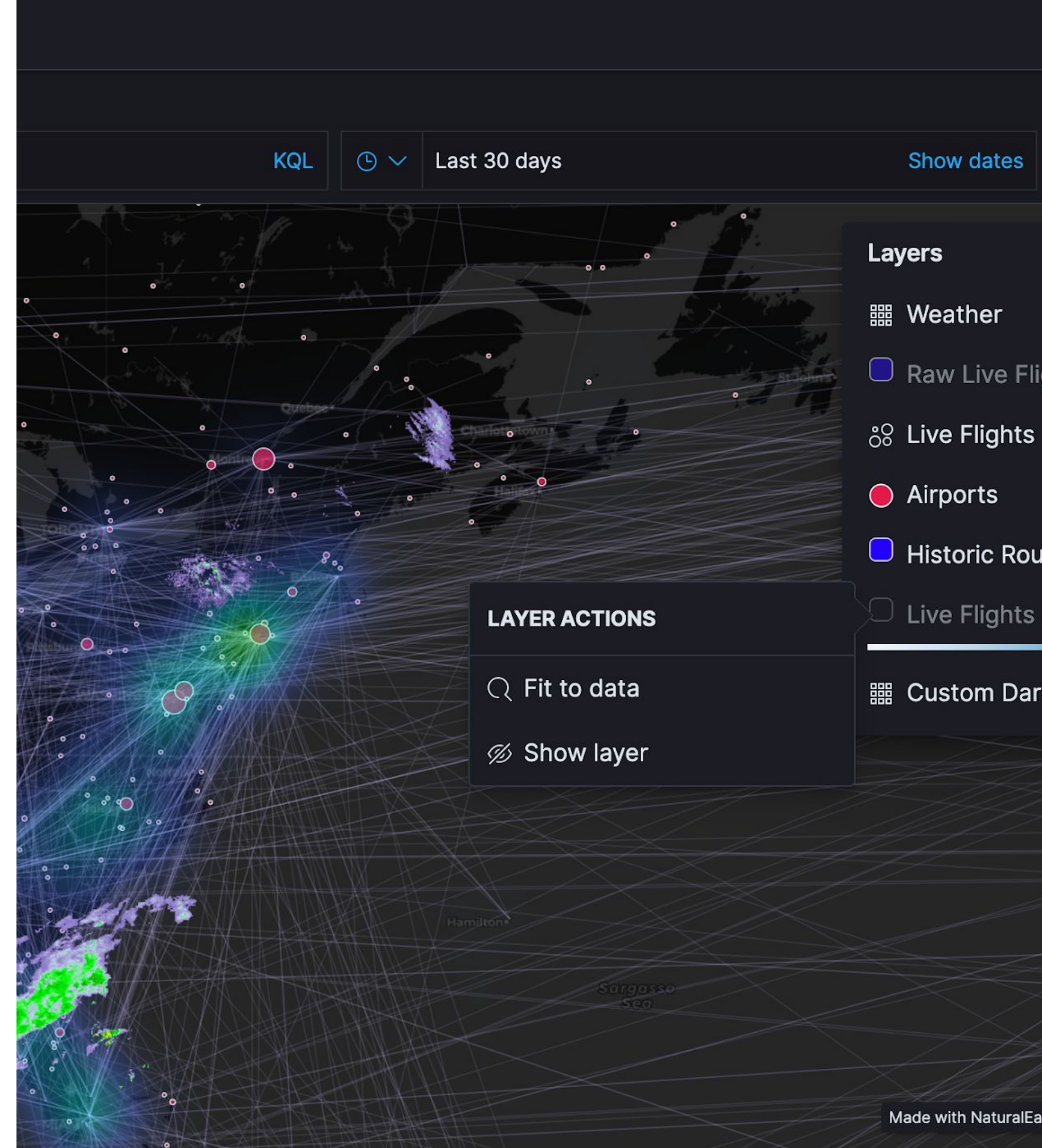
Kibana approach to Geographical  
Information Systems

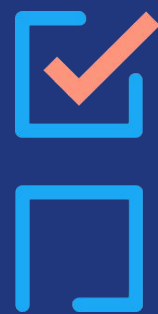


# Elastic Maps

OOTB 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 or Canvas workpads
- Embedded in other Kibana solution applications





# Quick web mapping intro





tangram js  
deck gl

Leaflet

OpenLayers

MapLibre

mapbox

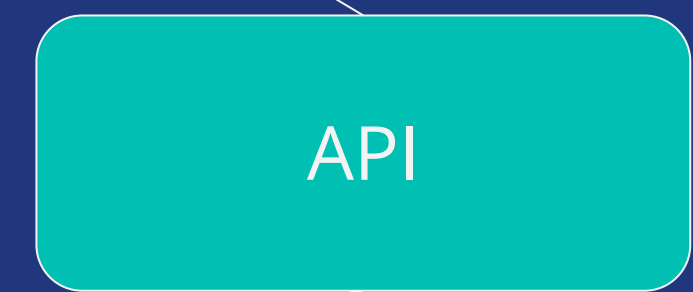
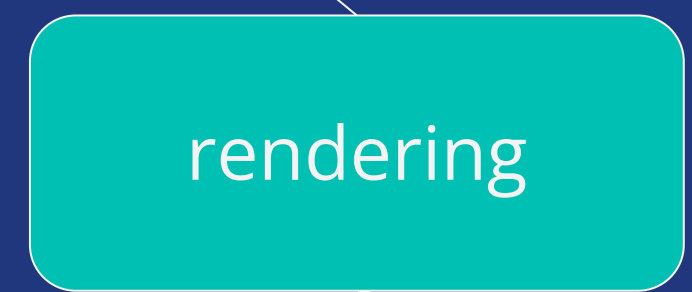
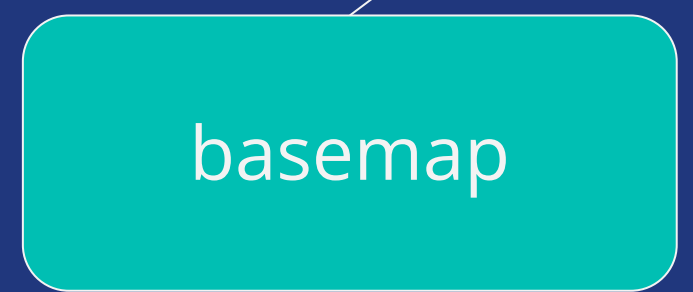


elastic



mvt  
png  
...

json  
xml  
...



GeoServer



maptiler

CARTO



mapbox

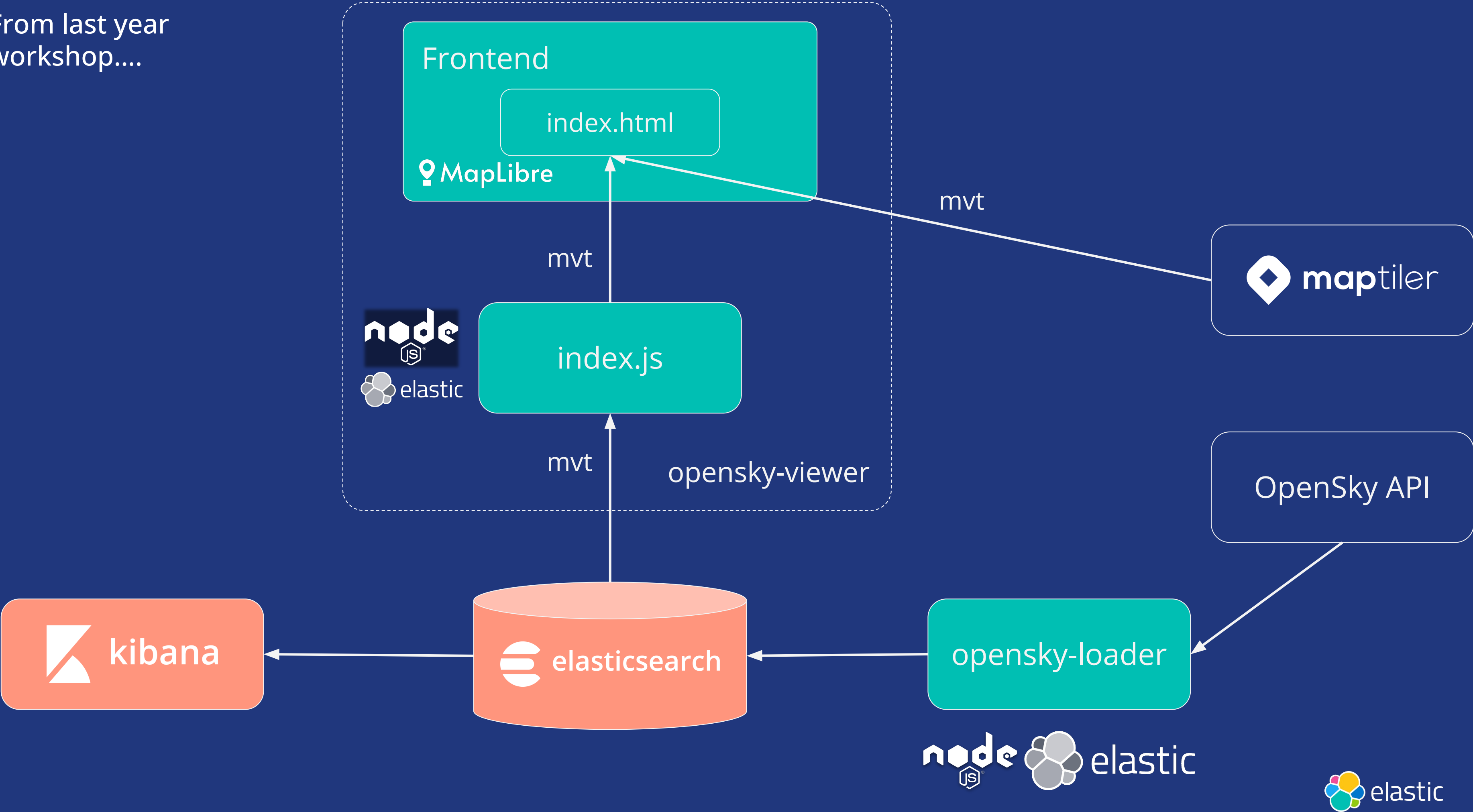


here




elastic

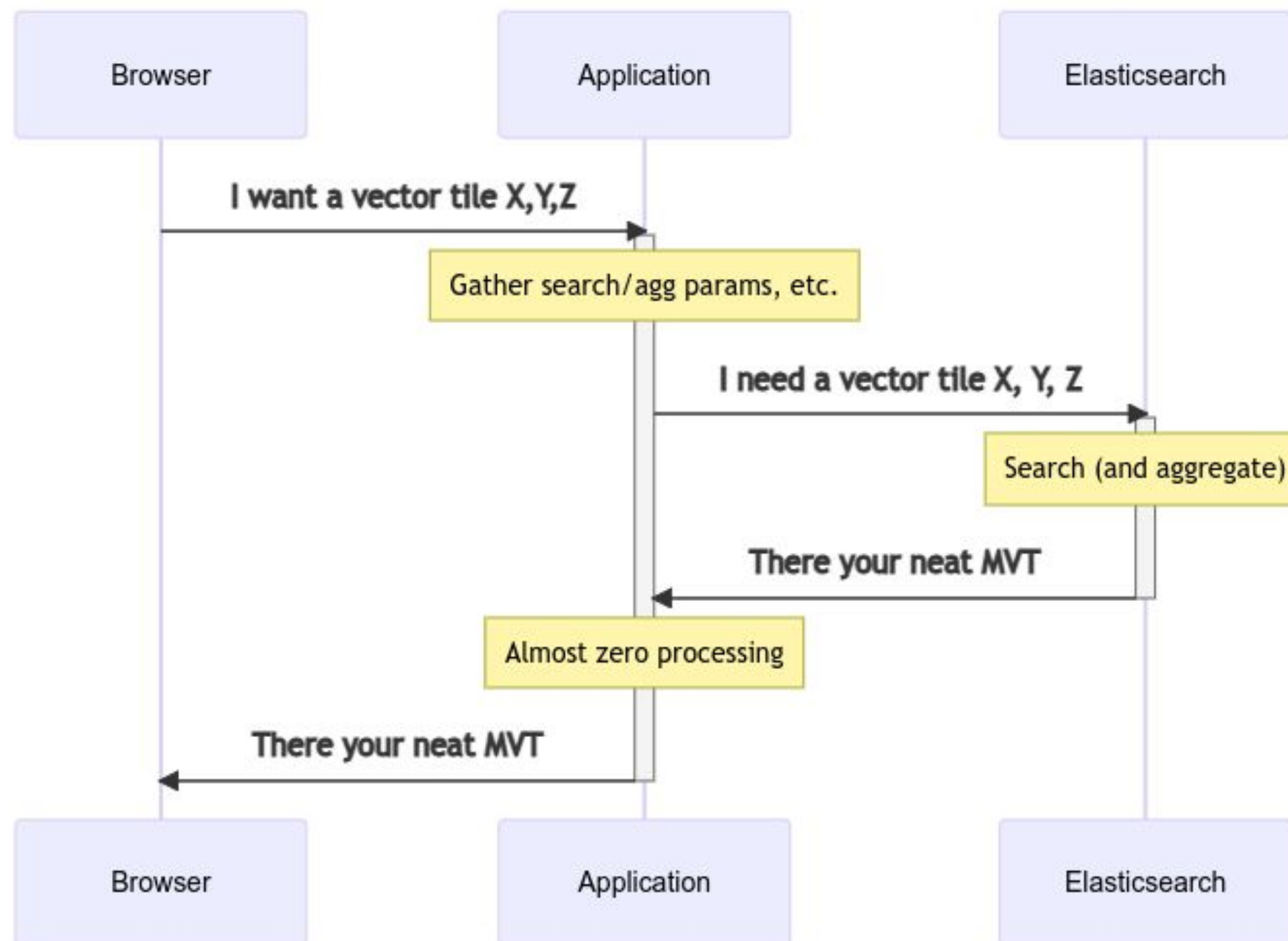
From last year workshop....





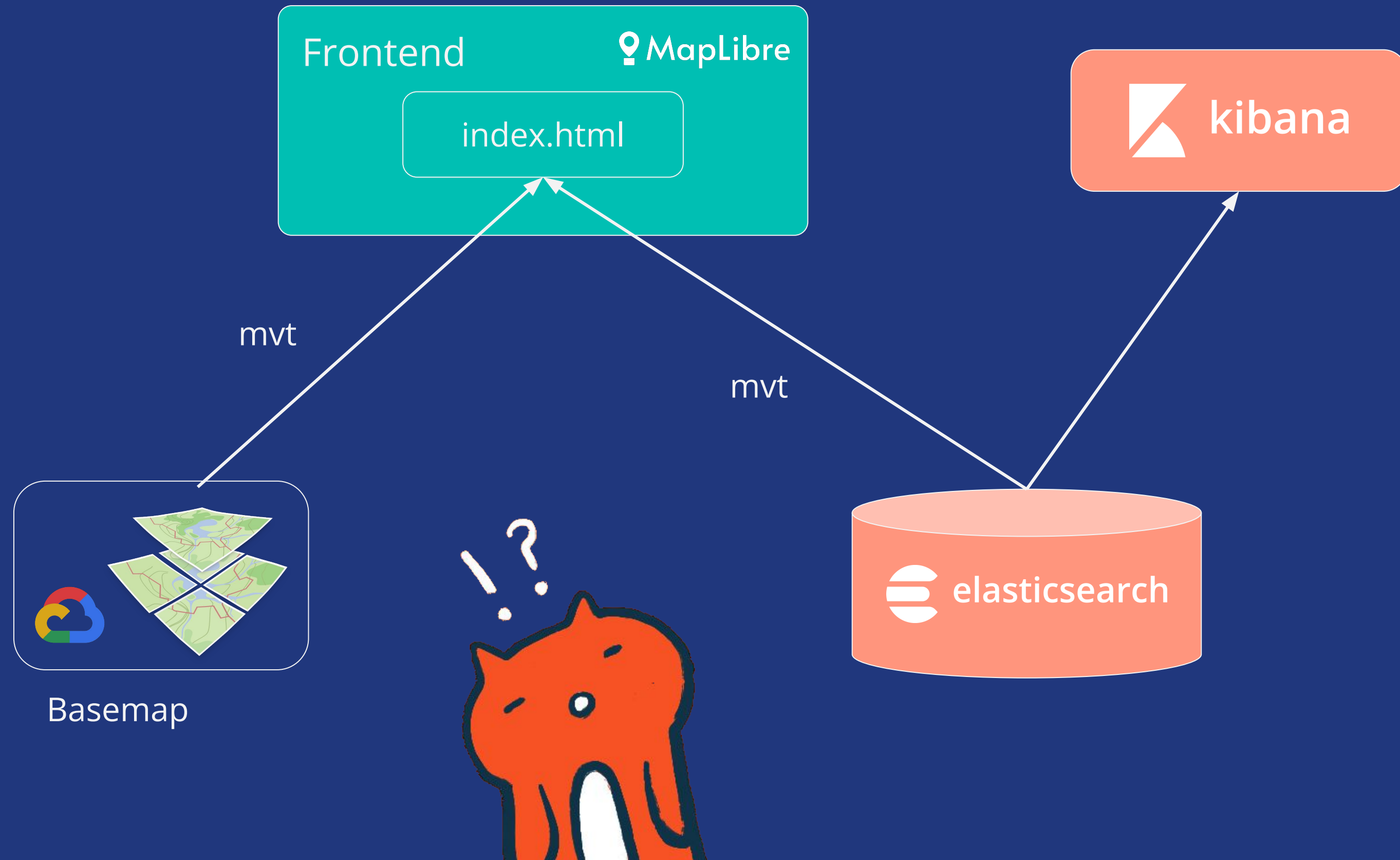
# Modern approach

1. New `_mvt` endpoint 
2. Elasticsearch outputs mapbox **vector tiles** in *protobuf* format
3. Can render up to **10.000** documents per tile
4. Geometries are **simplified**
5. `meta` layer with **details**
6. Optional **label** positions





Today

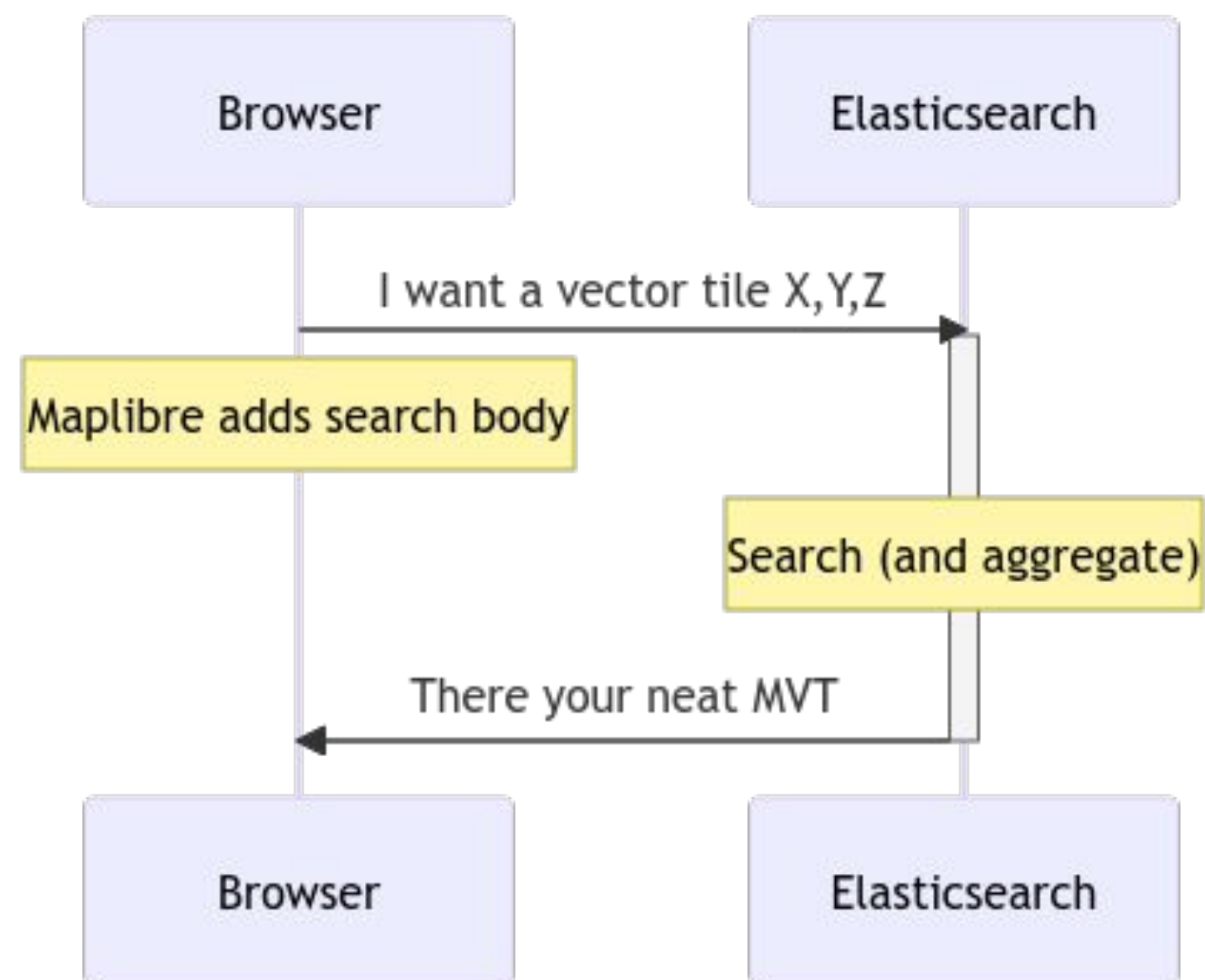




# YOLO

## approach

Remove the middleware by  
instructing Maplibre how to  
make valid Elasticsearch  
requests





# Simplifying our setup thanks to:

## **Changing a basemaps provider by a PMTiles file**

Use planetiler and pmtiles to conveniently generate a basemap file suited for this project and removing the need for a basemaps provider.

## **Connecting to Elasticsearch with API keys**

Setting up our cluster to allow well scoped API key requests.

## **Leveraging MapLibre transformRequest feature**

With this feature, we can include a payload in our vector tile requests to perform arbitrarily complex search and aggregation queries.







# Replacing a maps provider by our own vector tiles



- Use [planetiler](#) to quickly and easily generate vector tiles in the [PMTiles format](#)
  - All **Kosovo** is generated in around **49 seconds** (laptop)
  - **Denmark** in less than **4 minutes** (laptop)
  - Whole **planet** in less than **3 hours** (medium sized server)

```
$ java -jar planetiler.jar openmaptiles \
  --download --keep_unzipped=true --area=kosovo \
  --output=data/kosovo.pmtiles
```

- For this workshop we will use a file that combines:
  - A **full planet** for zooms **1 to 6**
  - **New York City** for zooms **> 6**

# Replacing a maps provider by our own vector tiles

- Adapt the OSM Bright style to consume this file

```
8     ],
9     "zoom": 1,
10    "bearing": 0,
11    "pitch": 0,
12    "sources": {
13      "openmaptiles": {
14        "type": "vector",
15        "url": "pmtiles://https://storage.googleapis.com/jsanz-bucket/planet/planet-nyc.pmtiles"
16      }
17    },
```

- Add the JavaScript `pmtiles` library and enable the protocol

```
/* initialize pmtiles support */
let protocol = new pmtiles.Protocol();
maplibregl.addProtocol("pmtiles",protocol.tile);
```

# Exposing Elasticsearch to the Internet securely

Elasticsearch can be accessed anonymously 📖

```
xpack.security.authc:  
  anonymous:  
    username: anonymous_user ❶  
    roles: role1, role2 ❷  
    authz_exception: true ❸
```

Instead, we will use an API key 📖 to read dedicated indices

```
POST /_security/api_key  
{  
  "name": "workshop-api-key",  
  "expiration": "5d",  
  "role_descriptors": {  
    "workshop": {  
      "index": [  
        {  
          "names": [  
            "geonames",  
            "311",  
            "nyc_boroughs"  
          ],  
          "privileges": [  
            "read",  
            "view_index_metadata"  
          ],  
          "field_security": {  
            "grant": [  
              "**"  
            ]  
          }  
        }  
      ]  
    }  
  }  
}
```



# Exposing Elasticsearch to the Internet securely

CORS is disabled by default 📖

```
http.cors:
  enabled : true
  allow-origin: "*"
  allow-methods: OPTIONS, HEAD, GET, POST
  allow-headers: "X-Requested-With, Content-Type, Content-Length, Authorization, Accept, User-Agent"
```

Our cluster is ready to accept API key requests 🎉

```
$ ELASTIC_HOST="https://your-cluster-url"
$ ELASTIC_APIKEY="your-encoded-name-and-api-key-here"
$ curl -H "Authorization: ApiKey ${ELASTIC_APIKEY}" \
  "${ELASTIC_HOST}/geonames/_count?pretty=true"
```



# How to make Elasticsearch queries from Maplibre?

## Vector Tile servers understand GET

In general templates for querying vector tile servers contain all parameters in the URL like

```
http://myserver/{z}/{x}/{y}.[pbf|png]?search={query}
```

## Elasticsearch `_mvt` and `_search` endpoints really need a payload

Some parameters are allowed in the URL but most of the extensive capabilities for searching are only available as payloads on POST requests

## Maplibre allows to "hack" the requests for tiles

The `transformRequest` map creation option allows arbitrary changes to each HTTP request, even changing the method.

# How to make Elasticsearch queries from Maplibre?

```
const map = new maplibregl.Map({
  container: 'map',
  style: './assets/style-ad.json',
  center: [1.523, 42.505],
  minZoom: 10,
  zoom: 15,
  hash: true,
  transformRequest: function (url, resourceType) {
    /* This function enriches the HTTP request to include
    the ES search body, change to a POST request, and include
    the Content-Type header */
    if (resourceType === 'Tile' && url.startsWith(ES_HOST)) {
      // Get which layer are we working with
      const body = getBody(
        url.includes(LAYERS[0])
          ? LAYERS[0]
          : LAYERS[1]
      );

      return {
        url: url,
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
          'Authorization': `ApiKey ${ES_APIKEY}`
        },
        body: JSON.stringify(body)
      };
    }
  }
});
```





# Laboratory

Let's draw some maps, finally!

# Set up

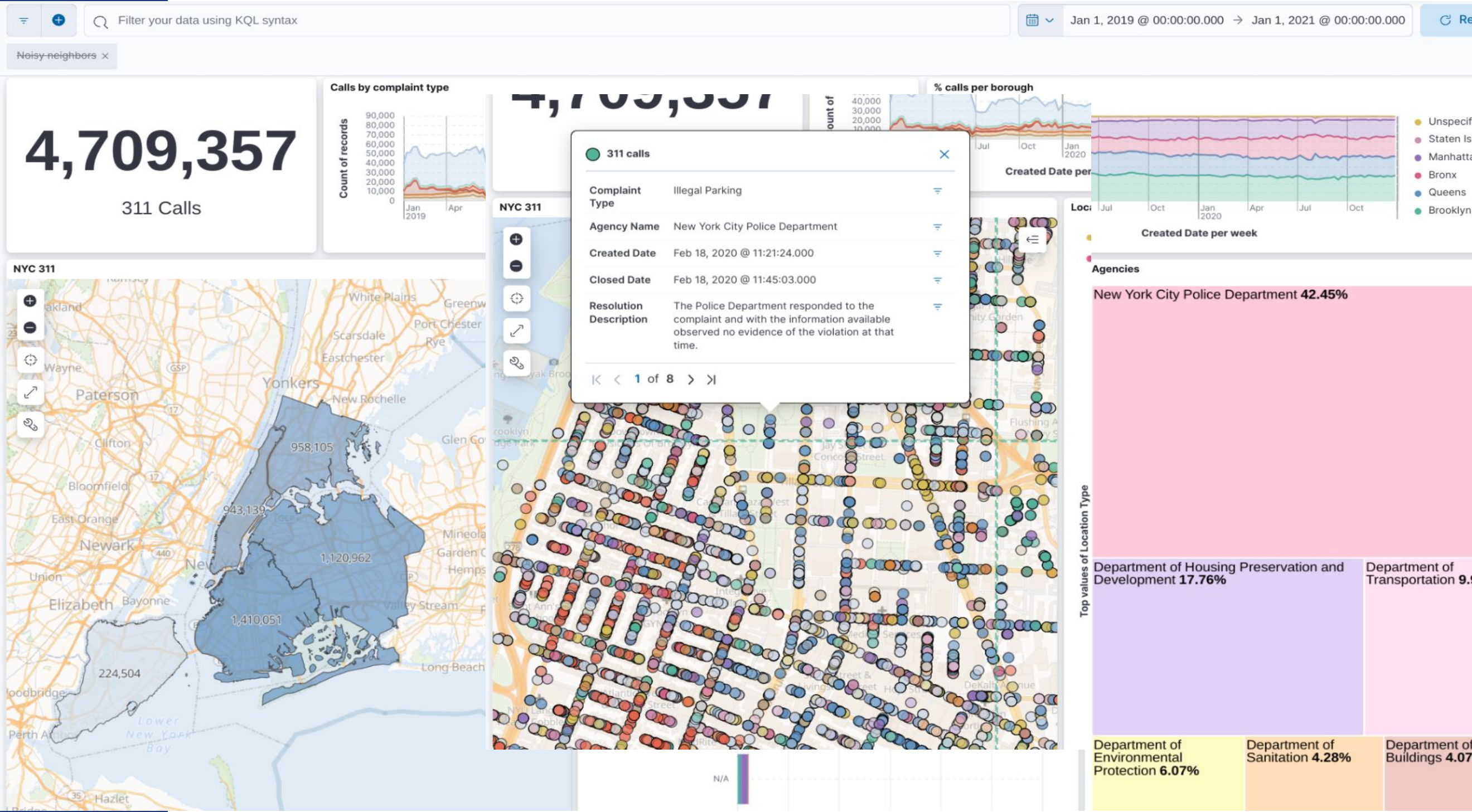
Get the **code** we will review together

- Easiest way, hit the "Remix" button at this glitch project:
  - <https://glitch.com/edit/#!/webmapping-elasticsearch-2023>
- If you are quick with git(hub) and node:
  - Clone or download the repo [jsanz/elastic-workshop](#)
  - Run the project inside [lab/vector-tile-viewer](#)
  - Play with HTML documents inside [pages](#) folder

Open [Kibana](#) with the read-only credentials shared on the session notes and check:

- Dev Console
- Discover
- Maps
- Dashboards

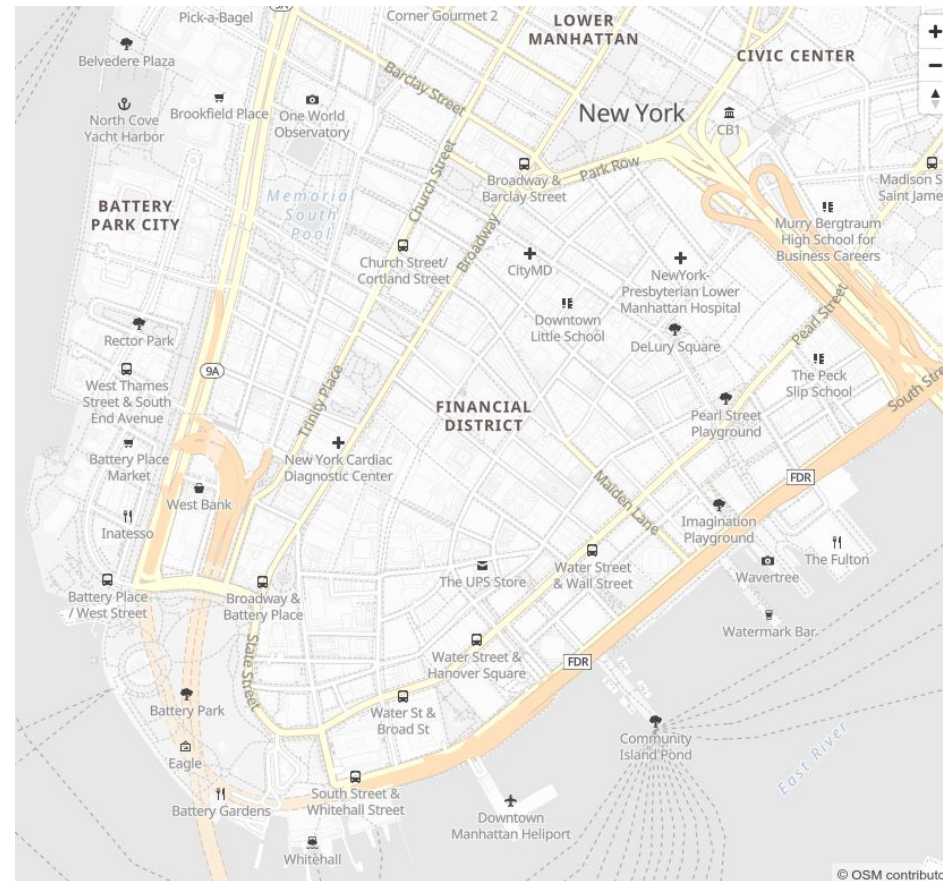
# Data: NYC 311





# Just a basemap

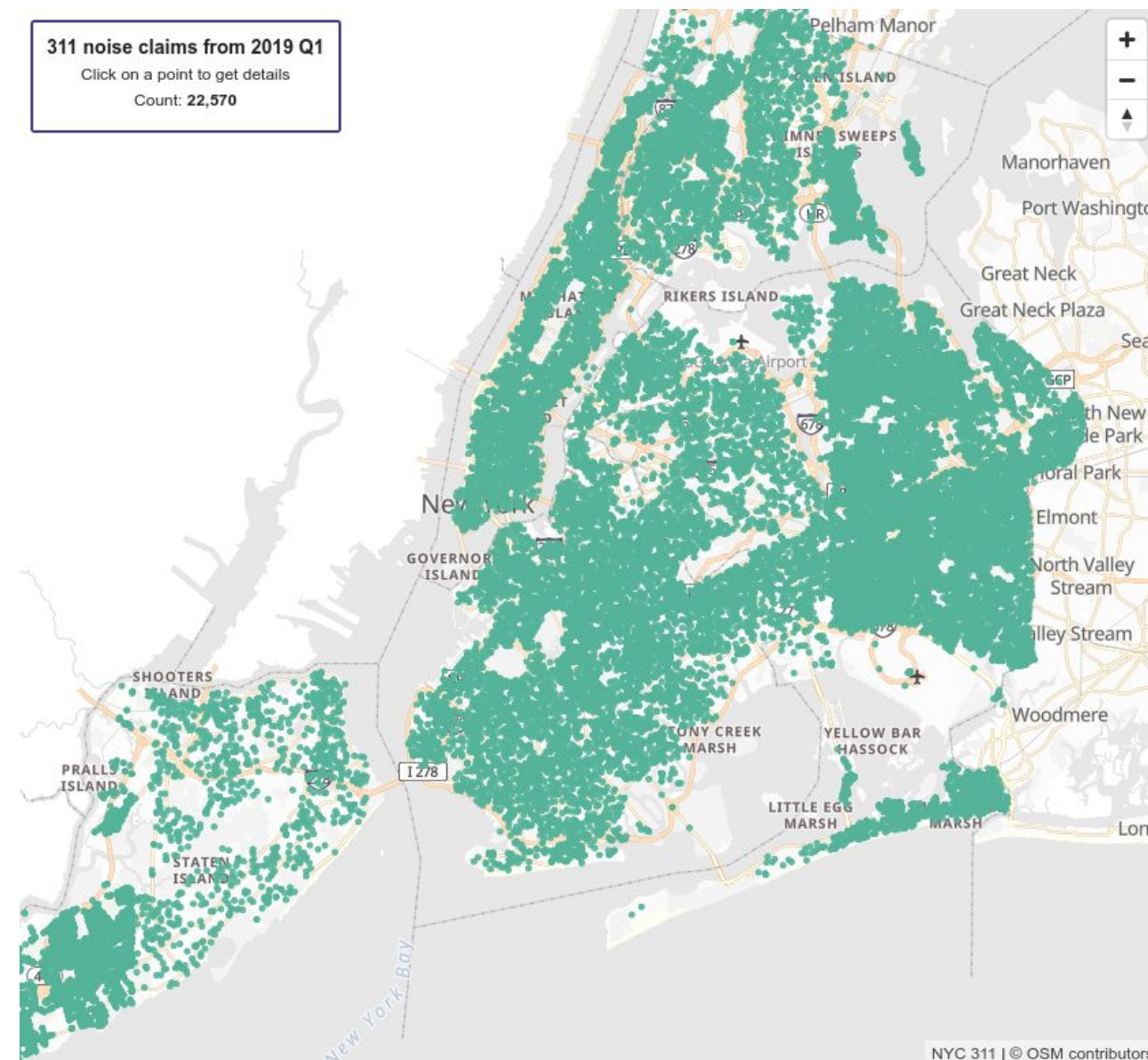
- Create a map with the OSM bright style



- Extra: consider updating the style to point to other tilesets available in the same folder:  
[catalonia.pmtiles](#), [denmark.pmtiles](#),  
[kosovo.pmtiles](#), [andorra.pmtiles](#)

# First documents from Elasticsearch

- Define a query and a new vector layer
- Count geometries from the rendered features

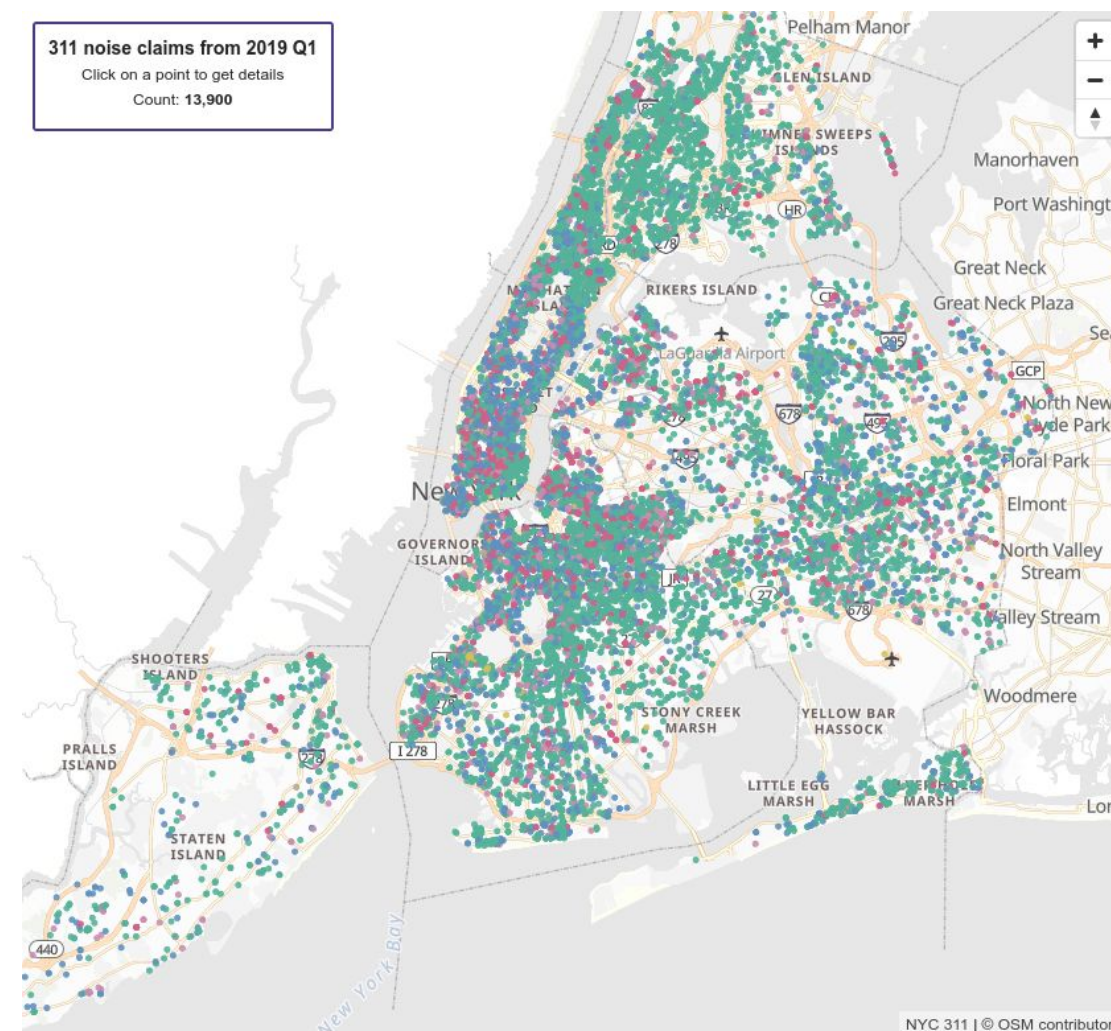


- Extra:
  - consider checking with Discover the data to select another time frame
  - Render geonames dataset



# Documents from Elasticsearch themed by complaint type

- Extend the query to filter by terms
- Thematic mapping with Maplibre styling

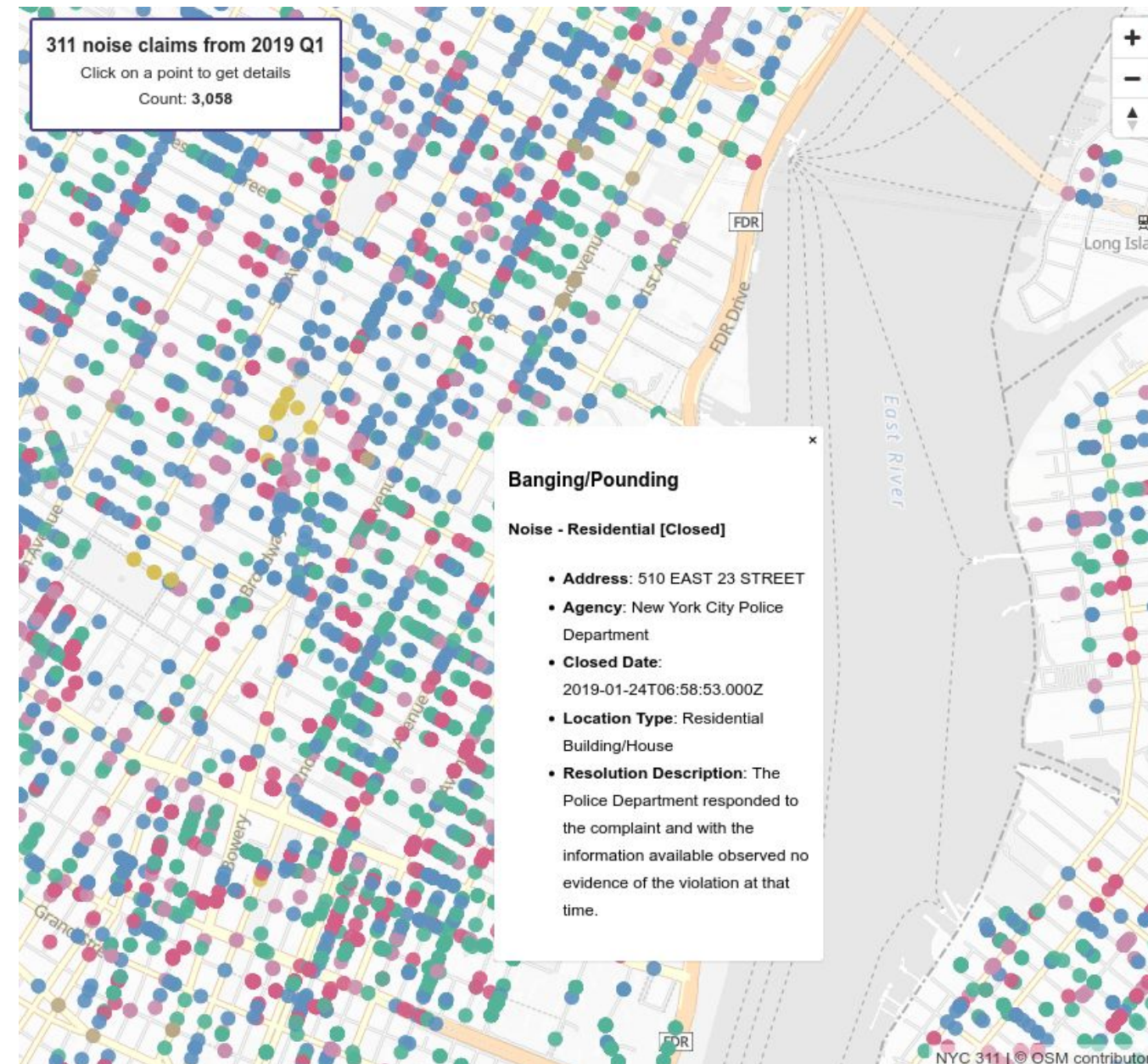


- Extra:
  - Render by another field, change size/opacity
  - Use other conditions to filter your data (Maps, Discover and their *Inspect tool* will be your friends for this)



# Add a popup with details of the complaint

- Add more fields to the vector tiles responses
- Include a basic popup implementation

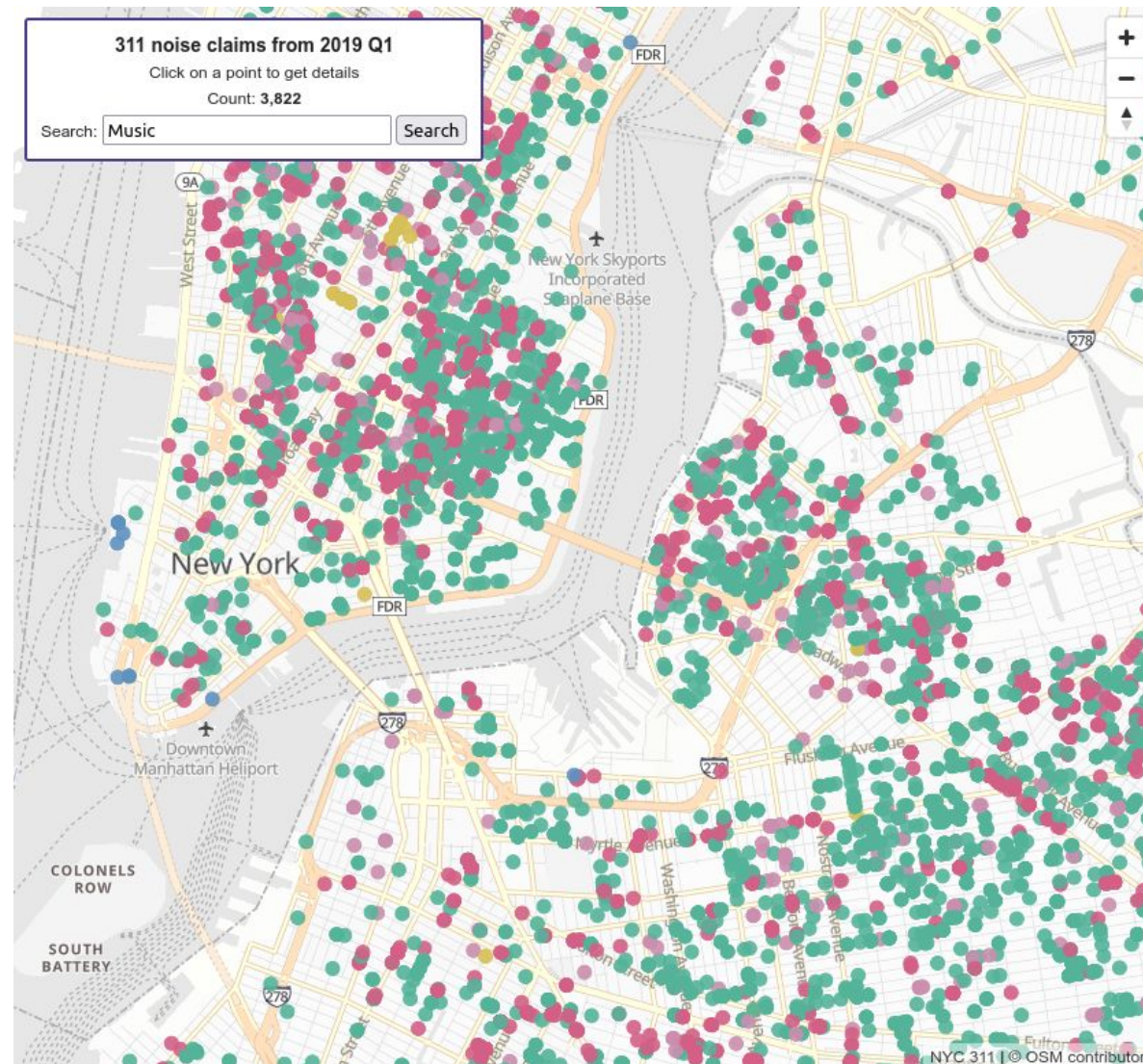


- Extra:
  - Play with the pop up template
  - Tooltip on hover?



# Search and filter documents from Elasticsearch

- Add a simple form with a text input
- Update the ES body to include the search query
- Reload the layer

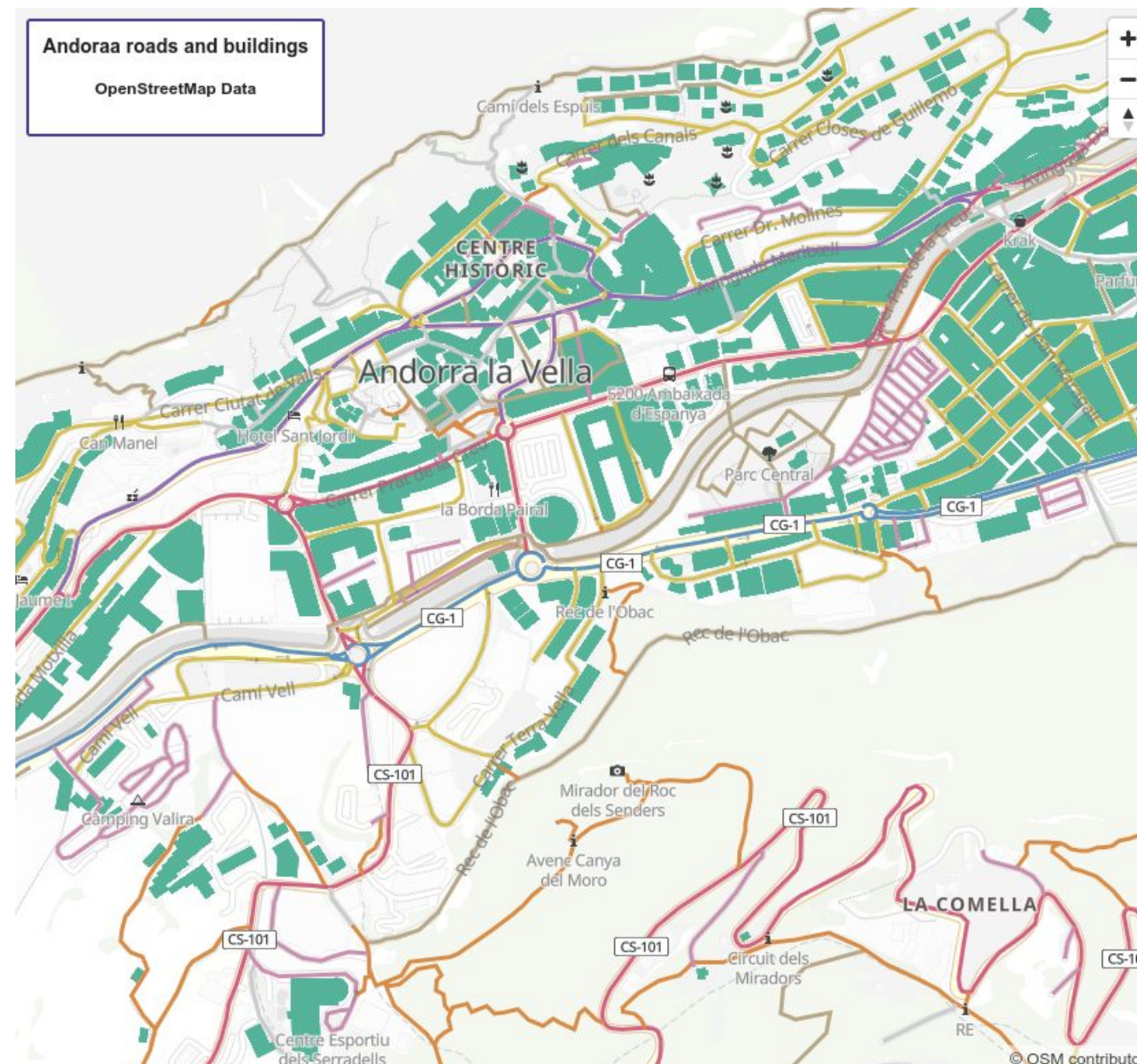


- Extra:
  - Refine the search to query only a single field
  - Add more fields to extend the query (date filter, agency selector, etc.)



# Render other geometry types

- Query two different aliases filtering data from the same index
- Adapt the code and styling for two layers

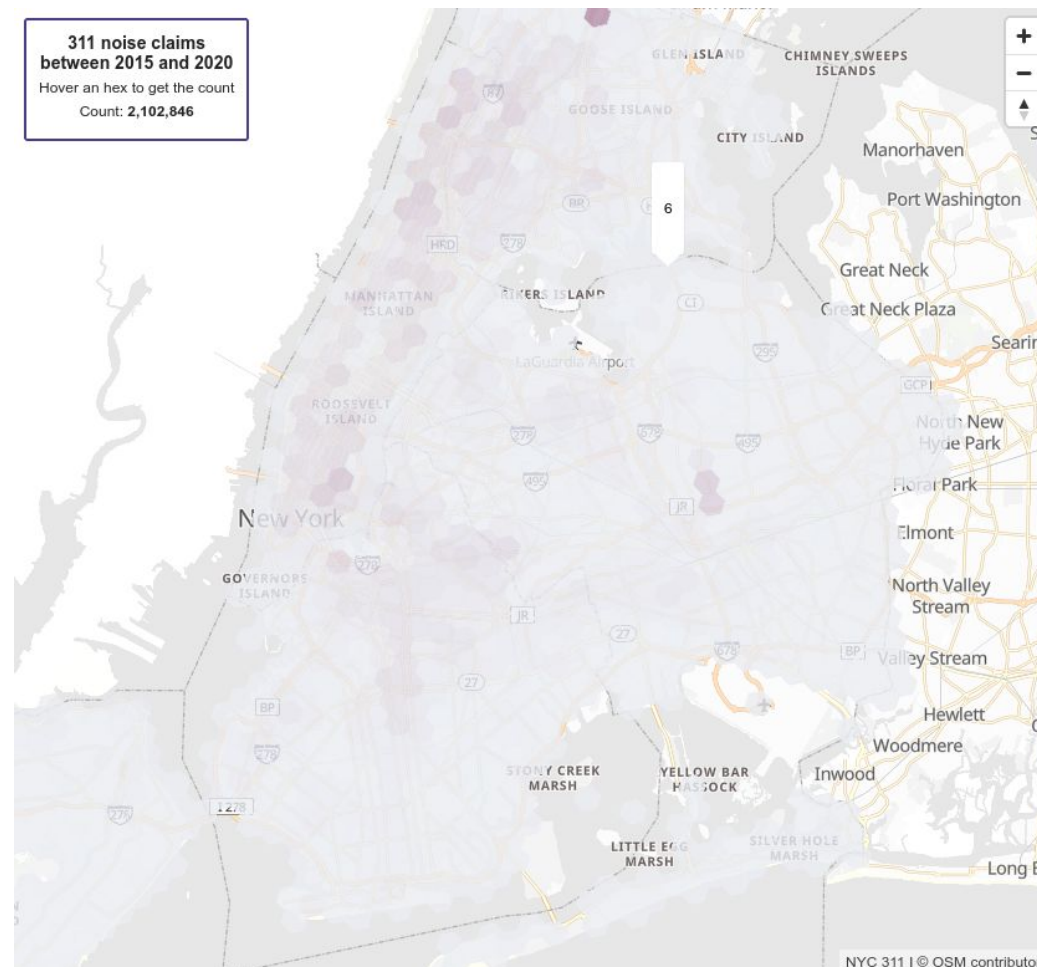


- Extra:
  - Check the different values for the highways



# Render aggregated data into H3 hexagons

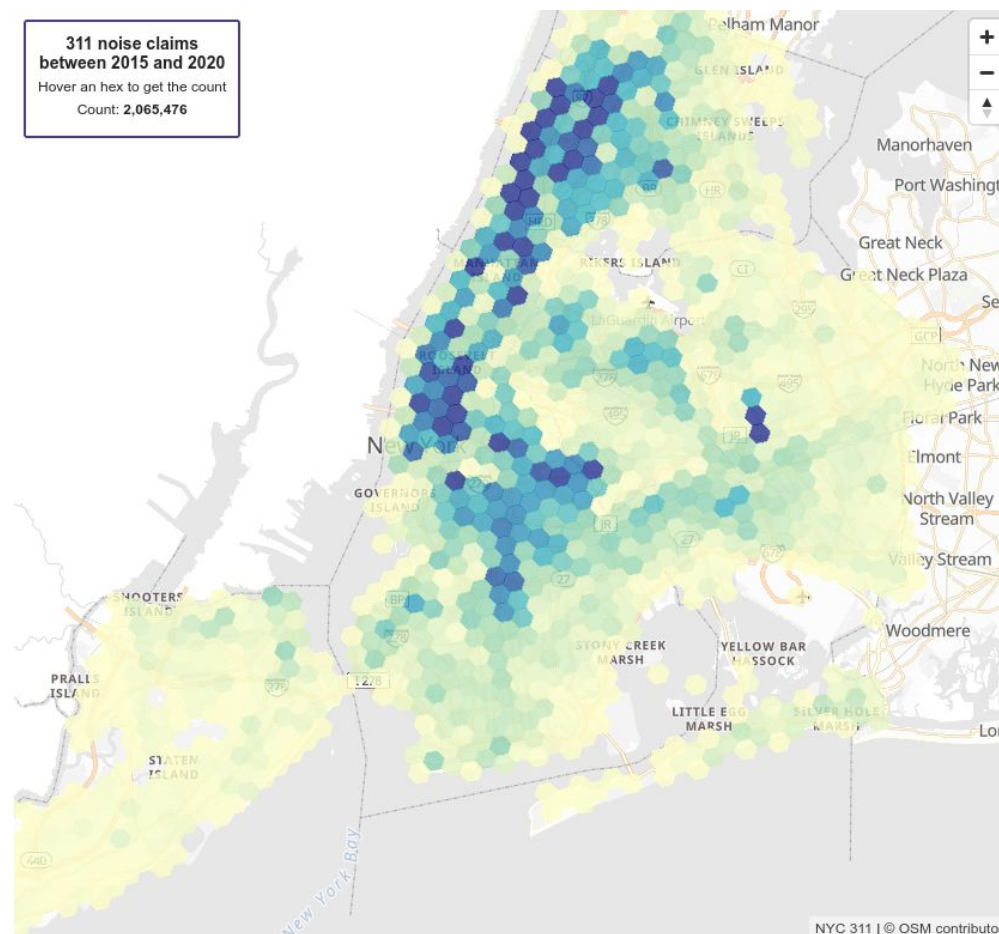
- Update the query with `grid_agg`, `grid_type` properties
- Pop up now works as a tooltip showing the count
- New problem: the legend is fixed but our counts heavily depend on the zoom level!



- Extra:
  - Do you notice an outlier?
  - Play with the `grid_precision` query property

# Adapt the legend to the zoom level using basic stats

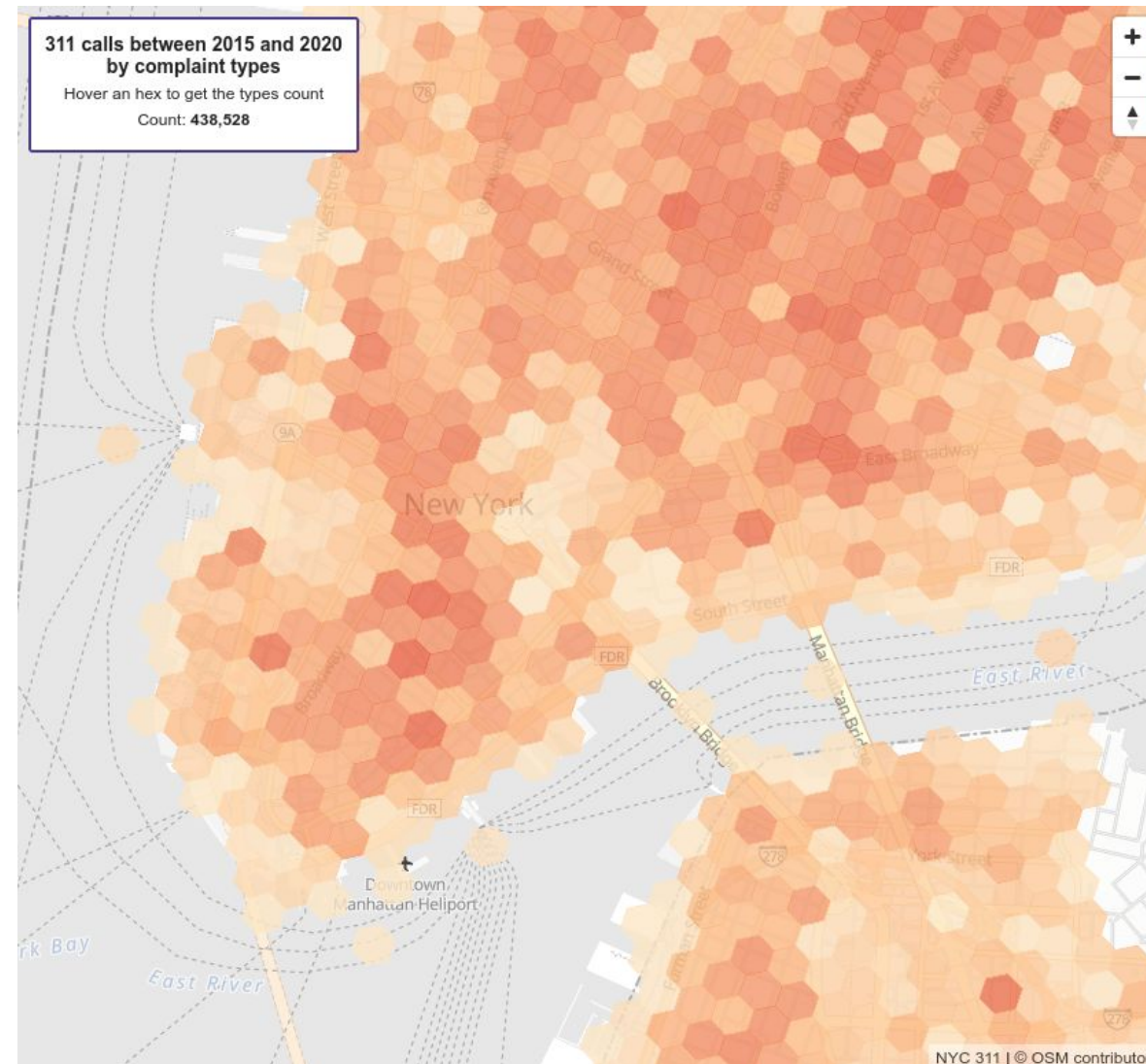
- Add a listener to the `zoomend` event to update the style based in the new maximum for the aggregation count and reload the layer.
- Include a "must\_not → geo\_distance" filter to remove those picky new yorkers



- Extra:
  - Use Kibana Maps to identify that outlier
  - Find other filter types to remove it

# Add a new metric: cardinality

- Include a new aggregation to compute how many different complaint types exist per hexagon.
- Increase the number of steps in the legend for better display

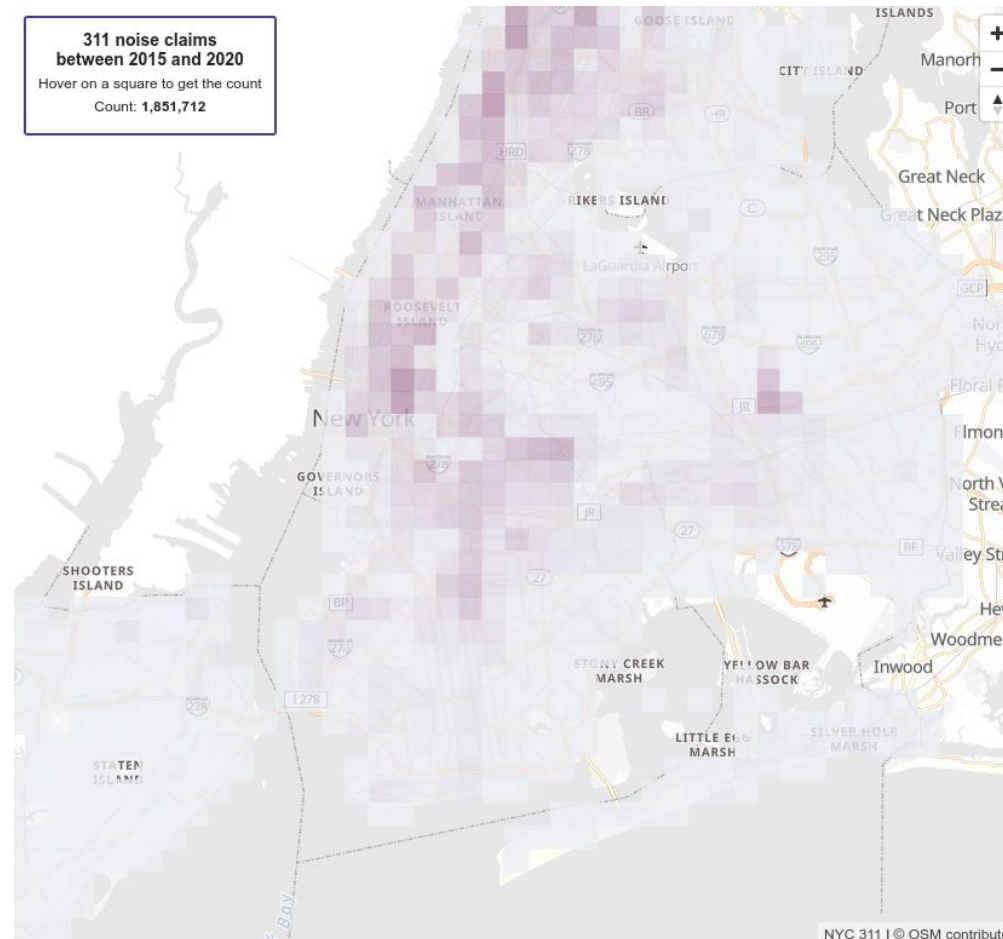


- Extra:
  - Find other interesting metrics available with Kibana Maps.



# Aggregate using mercator tiles

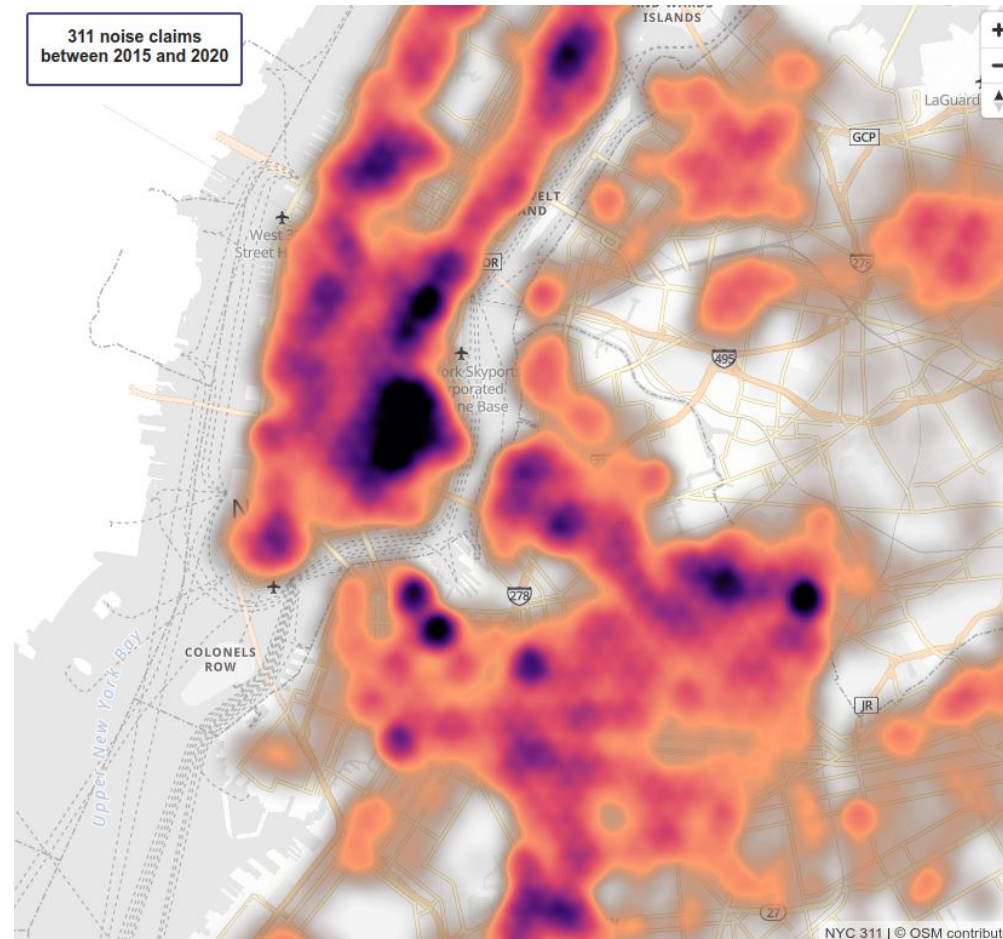
- Change the grid\_agg type to geotile (square mercator tiles).
- This aggregation type is much faster



- Extra:
  - Play with the max and min zoom levels of the application
  - Play with different color schemes and get help from Kibana Maps (<https://colorbrewer2.org> is a classic)

# Aggregate using a heat map

- A heatmap is in reality a grid with a custom styling
- Increase the grid\_precision to 7



- Extra:
  - Play changing the grid\_agg between geohez and geotile
  - Play with different color schemes and heatmap parameters
  - Figure out why it crashes on Firefox 🥲

# Closing

## Some final remarks

- Everything we covered in this session is available in the **free Basic license** (cloud or self-hosted).
- This setup is OK for **controlled** environments
- A **proper backend** to interact with Elasticsearch is the recommended approach
- Elasticsearch offers a **wide variety** of capabilities
  - `geo_line` aggregation and `geo_shape` aggregation (hex and tile)
  - Time Series, data streams, ingest pipelines, transforms, Cross Cluster Search, Cross Cluster Replication, ...
  - Including Artificial Intelligence and large language models!





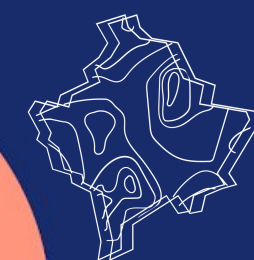
# Thanks!

Web mapping with Elasticsearch

---

Jorge Sanz | [jorge.sanz@elastic.co](mailto:jorge.sanz@elastic.co)

Craig Taverner | [craig.taverner@elastic.co](mailto:craig.taverner@elastic.co)



**FLOSS4G**  
Prizren, 2023