elastic | The Search AI Company

Visualizing Real-Time Flight Traffic with dump1090 and Elasticsearch

Justin Castilla

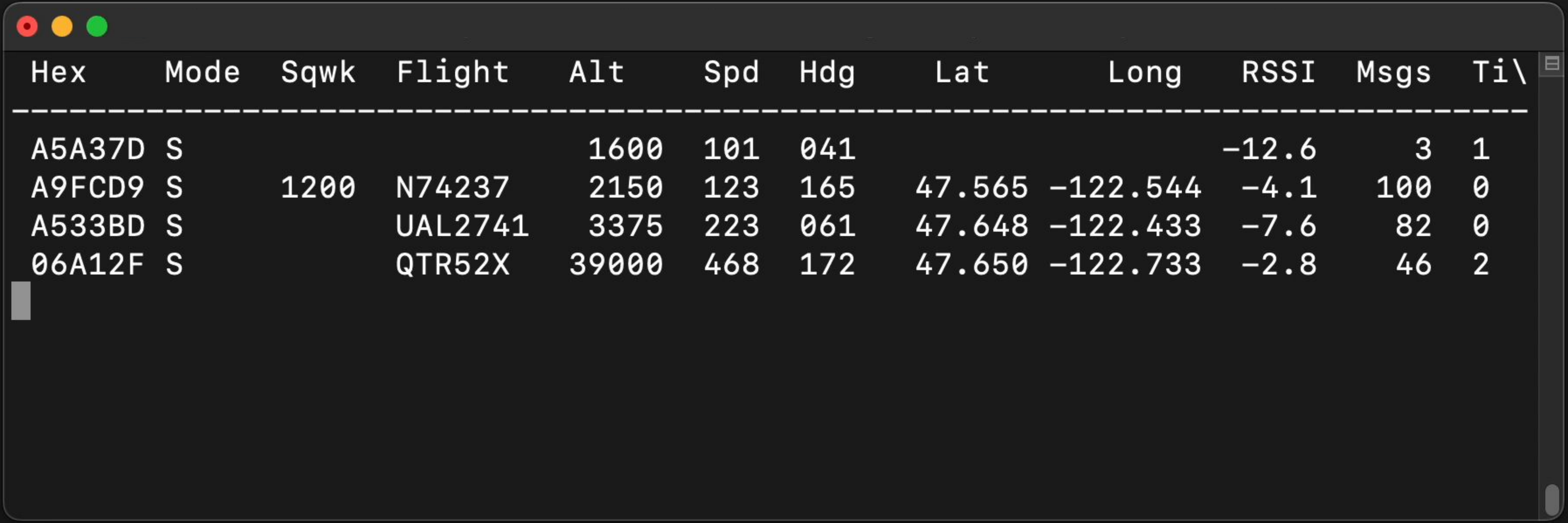
Tuesday May 13, 2025

What if you could track flights... yourself?

The Hardware: Realtek Software Defined Radio

RTL SDR BLOG
RTL-SDR.COM
☑ RTL2832U ☑ R828D ☑ TCXO ☑ BIAS-T ☑ HF
QUICKSTART: RTL-SDR.COM/QSG
MODEL: V4

The Data Source: dump1090



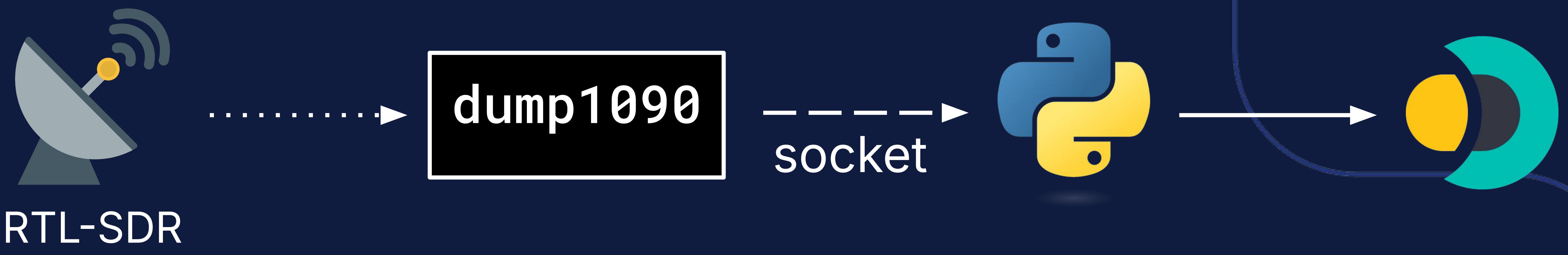
```
 Hex    Mode  Sqwk  Flight   Alt    Spd  Hdg    Lat      Long   RSSI  Msgs  Ti\
-------------------------------------------------------------------------------
 A5A37D S                     1600  101  041                   -12.6     3  1
 A9FCD9 S     1200  N74237    2150  123  165   47.565 -122.544  -4.1   100  0
 A533BD S           UAL2741   3375  223  061   47.648 -122.433  -7.6    82  0
 06A12F S           QTR52X   39000  468  172   47.650 -122.733  -2.8    46  2
```

What I Wanted to Build

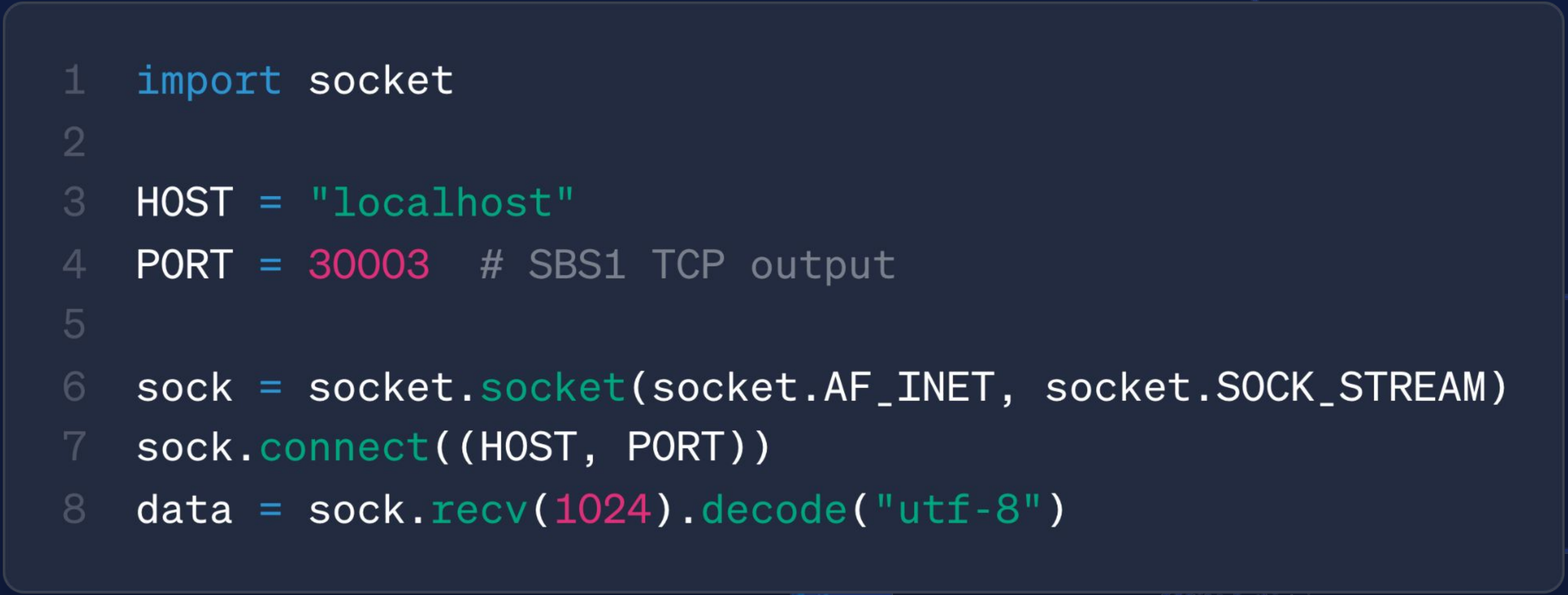
- Real-time flight dashboard
- Searchable, historical flight data
- Map of planes in Kibana using the Geospatial type

elastic

The Stack

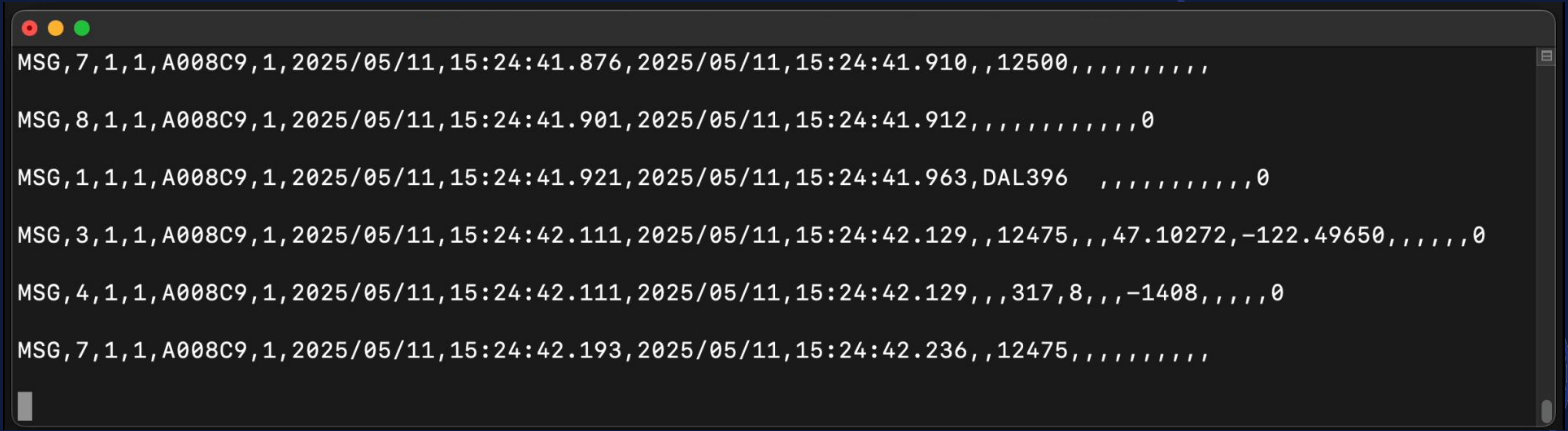


Connecting to dump1090



```python
import socket

HOST = "localhost"
PORT = 30003  # SBS1 TCP output

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((HOST, PORT))
data = sock.recv(1024).decode("utf-8")
```

Parsing Flight Messages

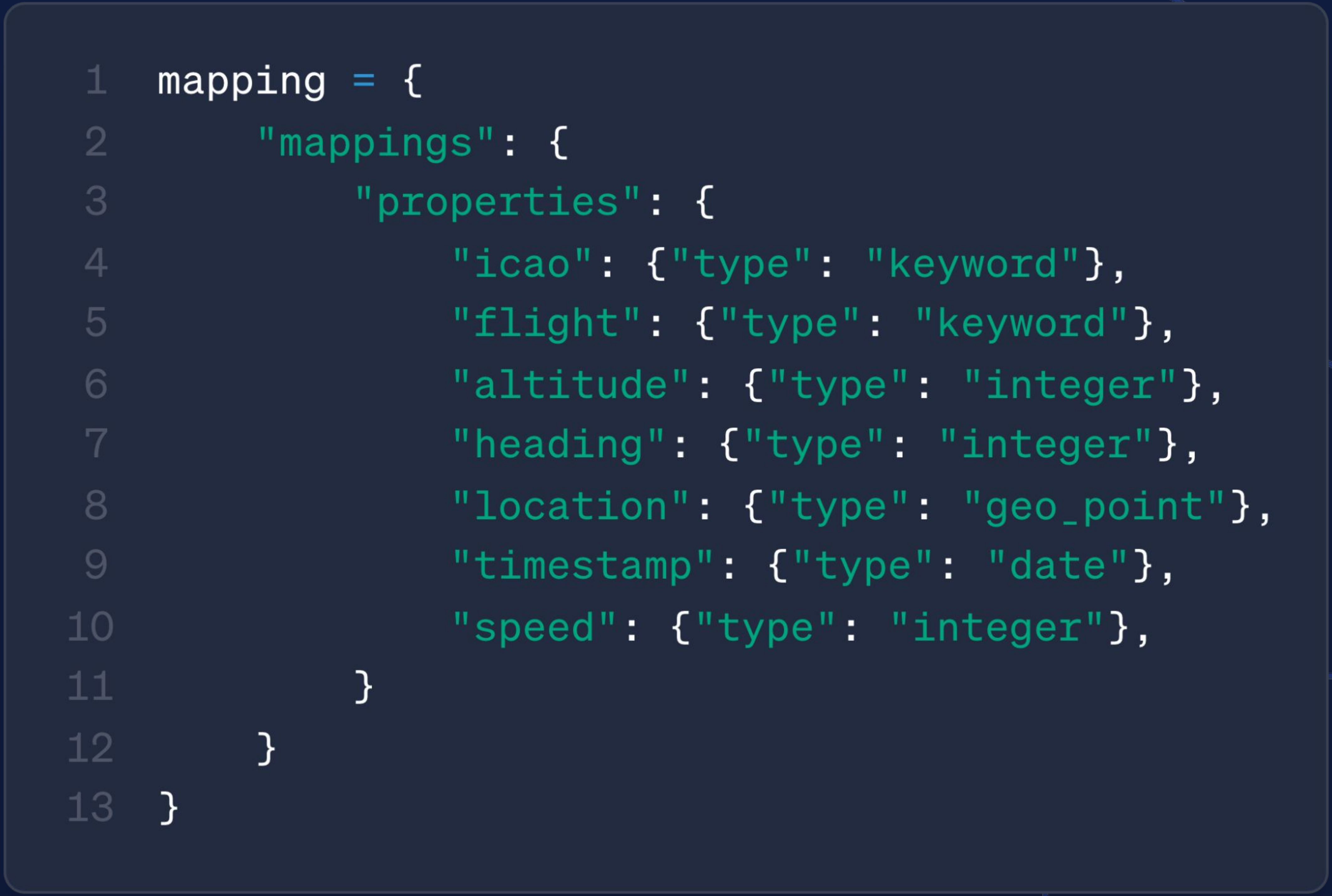


```
MSG,7,1,1,A008C9,1,2025/05/11,15:24:41.876,2025/05/11,15:24:41.910,,12500,,,,,,,,,,

MSG,8,1,1,A008C9,1,2025/05/11,15:24:41.901,2025/05/11,15:24:41.912,,,,,,,,,,,,0

MSG,1,1,1,A008C9,1,2025/05/11,15:24:41.921,2025/05/11,15:24:41.963,DAL396  ,,,,,,,,,,,0

MSG,3,1,1,A008C9,1,2025/05/11,15:24:42.111,2025/05/11,15:24:42.129,,12475,,,47.10272,-122.49650,,,,,,0

MSG,4,1,1,A008C9,1,2025/05/11,15:24:42.111,2025/05/11,15:24:42.129,,,317,8,,,-1408,,,,,0

MSG,7,1,1,A008C9,1,2025/05/11,15:24:42.193,2025/05/11,15:24:42.236,,12475,,,,,,,,,,
```

Why is the data partial?

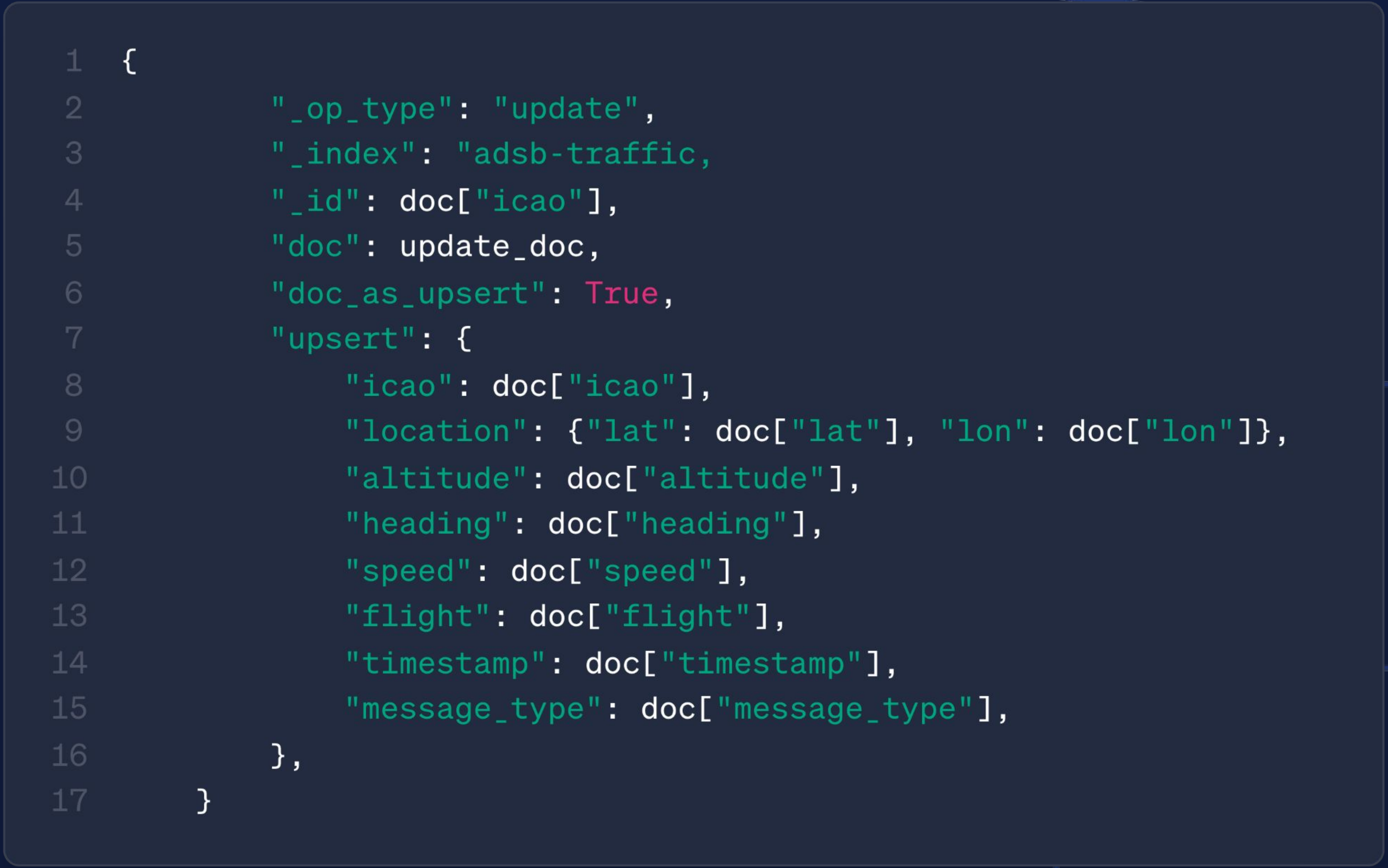
- Planes release small amounts of information instead of large payloads to reduce overall radio noise
- Each message is sent out 1 - 5 seconds based on what it is

Defining the Mapping



```
mapping = {
    "mappings": {
        "properties": {
            "icao": {"type": "keyword"},
            "flight": {"type": "keyword"},
            "altitude": {"type": "integer"},
            "heading": {"type": "integer"},
            "location": {"type": "geo_point"},
            "timestamp": {"type": "date"},
            "speed": {"type": "integer"},
        }
    }
}
```

Building a bigger picture



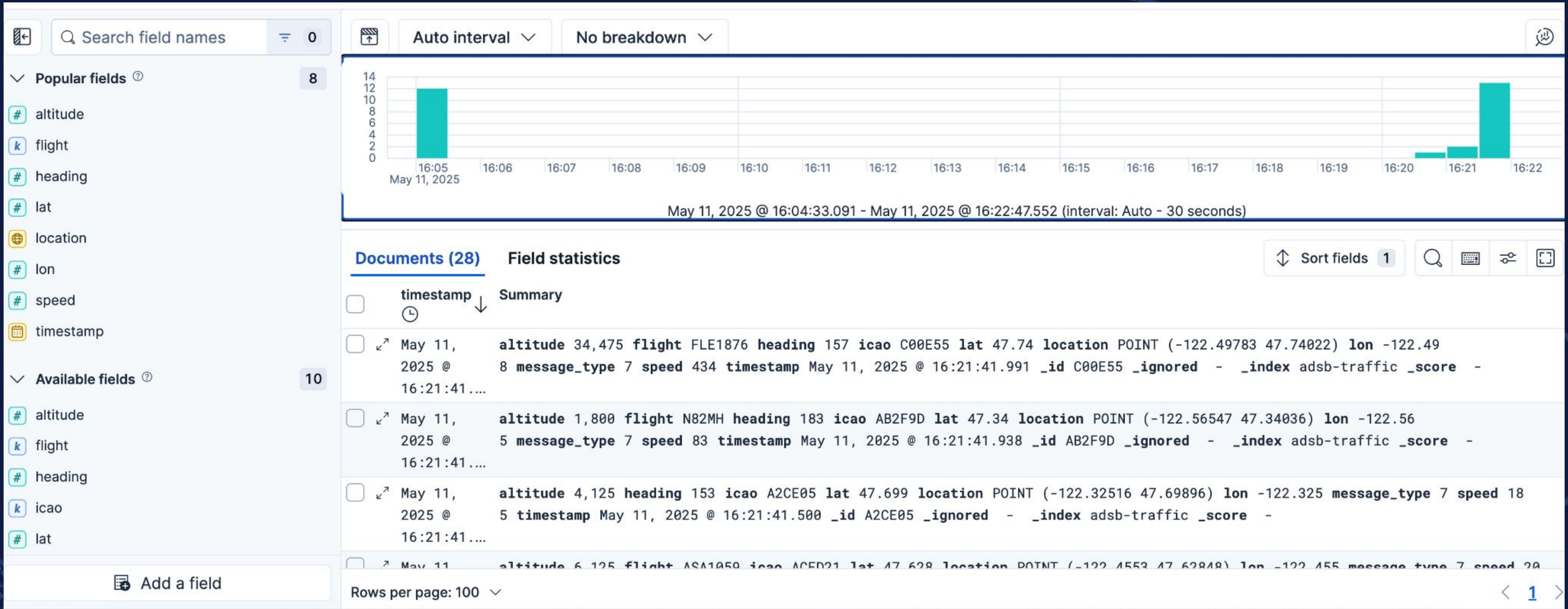
```
{
        "_op_type": "update",
        "_index": "adsb-traffic,
        "_id": doc["icao"],
        "doc": update_doc,
        "doc_as_upsert": True,
        "upsert": {
            "icao": doc["icao"],
            "location": {"lat": doc["lat"], "lon": doc["lon"]},
            "altitude": doc["altitude"],
            "heading": doc["heading"],
            "speed": doc["speed"],
            "flight": doc["flight"],
            "timestamp": doc["timestamp"],
            "message_type": doc["message_type"],
        },
    }
```

Connecting it all

- Read from dump1090
- Format lines into dictionaries
- Set up bulk action upsert
- Send bulk request
- Repeat

elastic

Creating the Kibana Map



Popular fields 8

- altitude
- flight
- heading
- lat
- location
- lon
- speed
- timestamp

Available fields 10

- altitude
- flight
- heading
- icao
- lat

May 11, 2025 @ 16:04:33.091 - May 11, 2025 @ 16:22:47.552 (interval: Auto - 30 seconds)

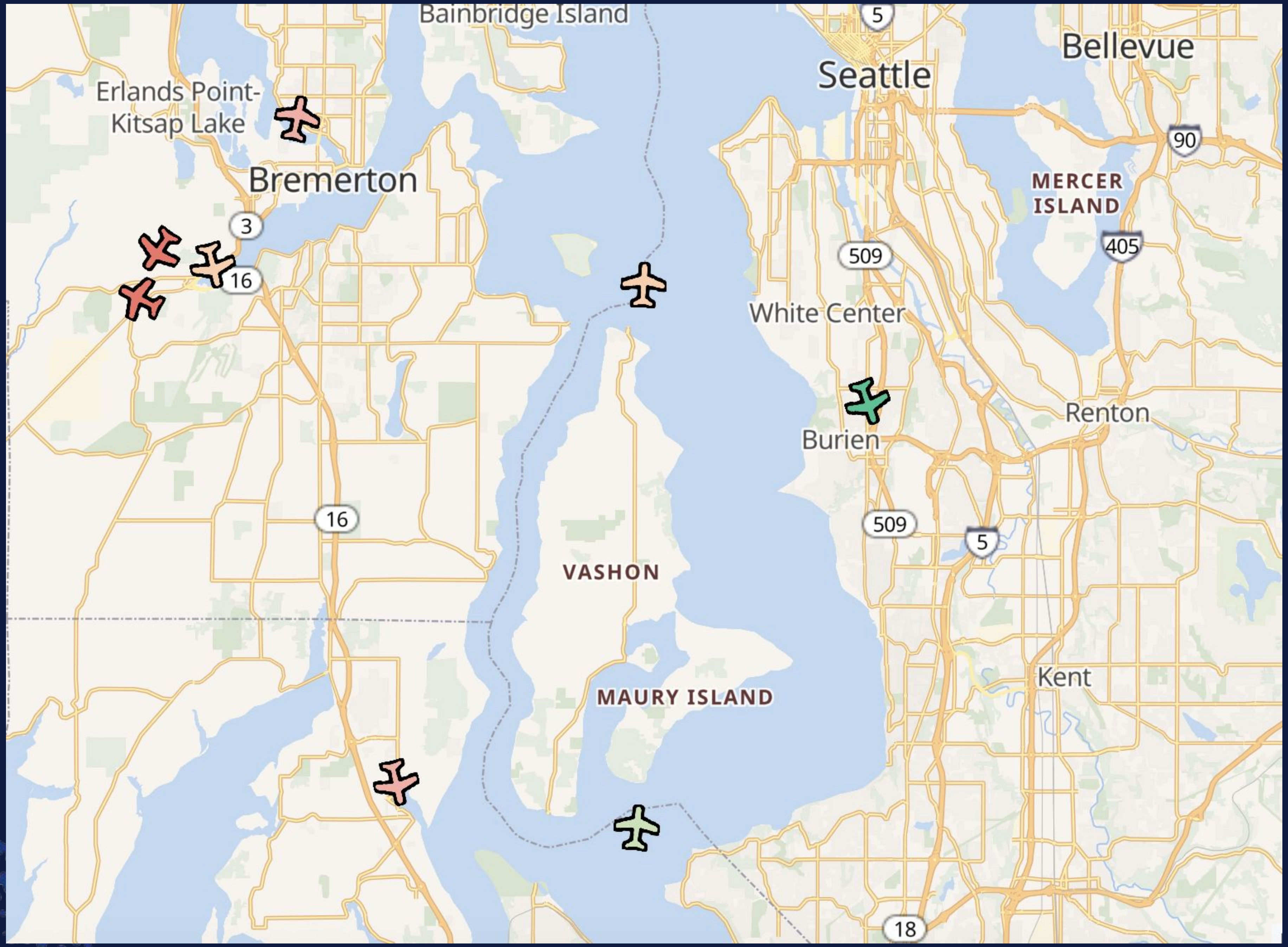
Documents (28) | **Field statistics**

timestamp	Summary
May 11, 2025 @ 16:21:41....	**altitude** 34,475 **flight** FLE1876 **heading** 157 **icao** C00E55 **lat** 47.74 **location** POINT (-122.49783 47.74022) **lon** -122.498 **message_type** 7 **speed** 434 **timestamp** May 11, 2025 @ 16:21:41.991 **_id** C00E55 **_ignored** - **_index** adsb-traffic **_score** -
May 11, 2025 @ 16:21:41....	**altitude** 1,800 **flight** N82MH **heading** 183 **icao** AB2F9D **lat** 47.34 **location** POINT (-122.56547 47.34036) **lon** -122.565 **message_type** 7 **speed** 83 **timestamp** May 11, 2025 @ 16:21:41.938 **_id** AB2F9D **_ignored** - **_index** adsb-traffic **_score** -
May 11, 2025 @ 16:21:41....	**altitude** 4,125 **heading** 153 **icao** A2CE05 **lat** 47.699 **location** POINT (-122.32516 47.69896) **lon** -122.325 **message_type** 7 **speed** 185 **timestamp** May 11, 2025 @ 16:21:41.500 **_id** A2CE05 **_ignored** - **_index** adsb-traffic **_score** -
May 11	**altitude** 6,125 **flight** ASA1059 **icao** ACED21 **lat** 47.628 **location** POINT (-122.4553 47.62848) **lon** -122.455 **message_type** 7 **speed** 20

Rows per page: 100

1

Creating the Kibana Map



So many tasty fields!

- Geopoint - place on map
- Elevation - color gradient
- Heading - icon direction
- 20 minutes from NOW

What I learned from this project (planes):

- Cessna Skyhawk 172 is THE most common plane
- Flight patterns are consistent
- Busy times
- Military doesn't always transpond
- Flight APIs are overpriced

elastic

What I learned from this project (Elastic):

- `doc_as_upsert` is key to building documents over time from partial fragments

- Personal opinion: bulk actions client should be native or built out more, at least for Python

Dashboard enrichment

Future work to be done:

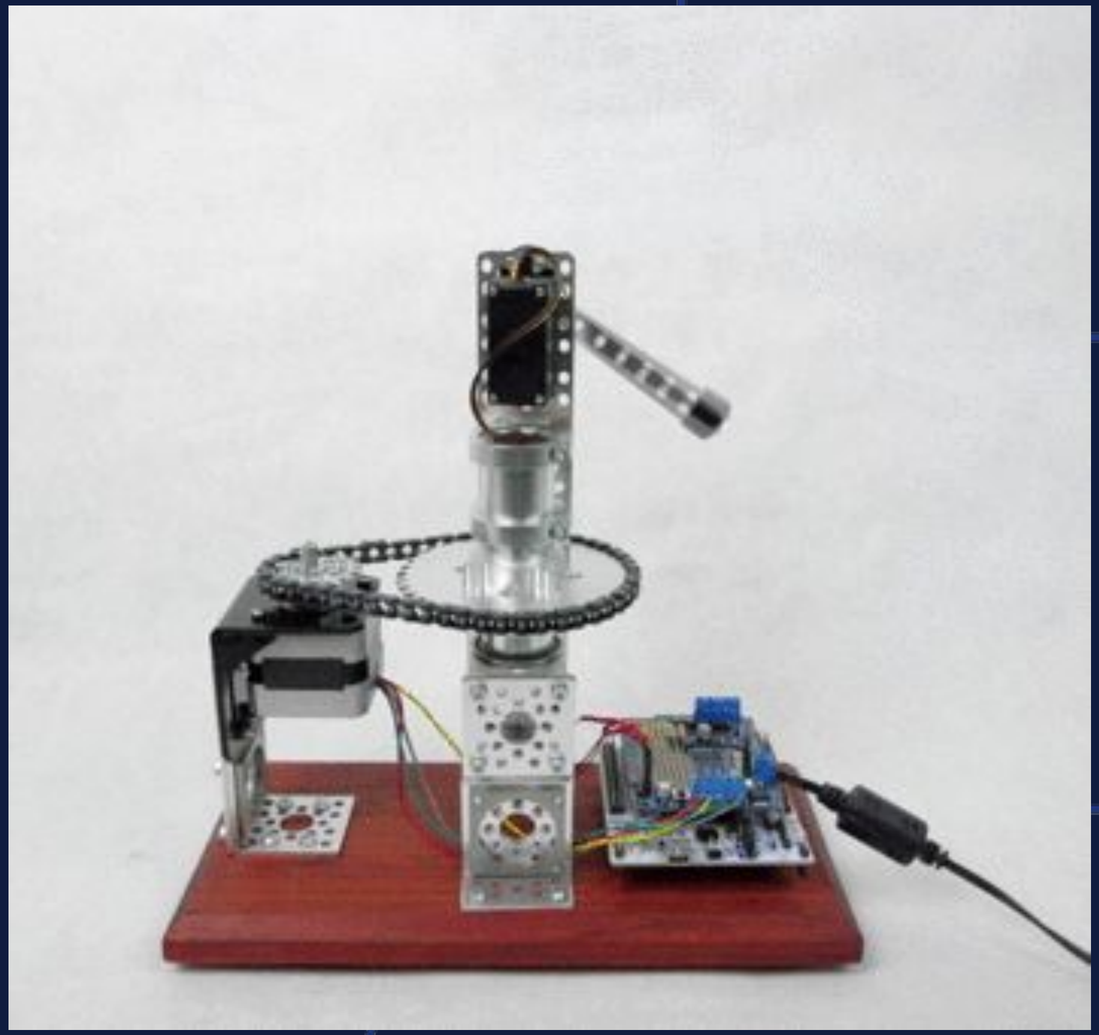
- Notifications for new or flagged aircraft
- Squawk notifications (7500 is bad)
- Visual flight path based on previous geopoint
- Bigger antenna

Challenges

- Elasticsearch document TTL (time-to-live)
  - Used to have native TTL support, but it was deprecated in version 5.0 and removed in 6.0.
  - Index Lifecycle Management, but that’s brutish and coarse
  - Call a function to trim old timestamped documents?

Further Applications

- Marine Traffic is very similar
- Send and receive data packets and store in Elastic
- Organize CB communications
- Literally point at an object in the sky



Thank you!

- Repository is available with BOM and links!
- We can share our data live!
- Planes are cool!