Iskanje brezna velikih podatkov v realnem času

Big Data Analytics in Real-time

© 2025 Simon Šanca, simon.sanca@uib.no

University of Bergen / IT-Platform, Team Linux

OpenSearch specifikacija

- **Predstavitev:** O'Reilly Emerging Technology Conference - 15. marca 2005.
- **Prvotni cilj:**
  - Standardizacija objave rezultatov iskanja z uporabo opisnih datotek v XML,
  - standardizacija poizvedb ter odzivov v RSS ali Atom (query syntax).
- **Uporabnost:**
  - Hitro iskanje spletnih strani in enostavno deljenje rezultatov.
  - *A way for websites and search engines to publish search results in a standard and accessible format.*

OpenSearch je postal široko podprt v brskalnikih, kot so Firefox, Safari in Chrome, kar je takrat omogočilo dodajanje prilagojenih iskalnikov v iskalno vrstico brskalnika.

OpenSearch Project

OpenSearch is a community-driven, open-source search and analytics suite used by developers to ingest, search, visualize, and analyze data.

- veja Elasticsearch in Kibane, ki sta plačljiva.

v1.0 - julij 2021; Apache Licence, Version 2.0

Glavne komponente:

- **OpenSearch:** Shramba podatkov in iskalnik za hitro obdelavo poizvedb.
- **OpenSearch Dashboards:** Orodje za vizualizacijo podatkov in up. vmesnik.
- **OpenSearch Data Prepper:** Strežniški zbiralnik podatkov za pripravo podatkov.

Uporabniki lahko razširijo funkcionalnost OpenSearch z izbiro vtičnikov, ki izboljšajo iskanje, analitiko, opazovanje, varnost, strojno učenje in še več.

OpenSearch danes

Dokumentacija: https://opensearch.org/docs/latest/
Repozitorij: https://github.com/opensearch-project

Do avgusta 2024 je AWS poročal o "desetih tisočih" strankah, z več kot 700 milijoni prenosov in prispevki tisočev razvijalcev.

Septembra 2024 se je lastništvo preneslo z AWS na **OpenSearch Software Foundation** pod okriljem **Linux Foundation**.

Gre za porazdeljen iskalni in analitič sistem, ki temelji na vektorski podatkovni bazi.

Application Performance Monitoring, Log Analytics, Big Data Analytics, Time Series Analysis, Data Visualization in ostalo.

UNIVERSITAS BERGENSIS

UNIVERSITY OF BERGEN

Projekt DataOPS: **Log Analytics**; normalizacija / standardizacija zapisov (logs)


Time spent by service
frontend 64.19%
redis 4.82%
route 12.4%
customer 6.9%
driver 4.89%
mysql 6.8%
Span detail
0 ms
200 ms
400 ms
600 ms
800 ms
frontend
HTTP GET /dispatch
frontend
HTTP GET: /customer
frontend
HTTP GET
customer
HTTP GET /customer
mysql
SQL SELECT
frontend
/driver.DriverService/FindNearest
driver
/driver.DriverService/FindNearest
redis
FindDriverIDs
redis
GetDriver
2
redis


METRICS
METRIC
PERFORMANCE
RESOURCES
TIME
SAFETY
VALUE
INFORMATION
SEVEN
SCORECARD
UNDERSTANDABLE
ANALYSIS
EFFECTIVENESS
QUALITY
PROCESS
SCOPE
STEP
ACTIONS
OUTPUTS
MEASURES
STRATEGY
CRITERIA
COST
STATISTICAL
HIGH-VALUE
IMPROVEMENT
CONTROL
LEVELS
DATA
MEANINGFUL
RELEVANT
ACTIONABLE
MEASURABLE
TIME-PHASED


Logs

OpenSearch Dashboards

- Da vemo kaj se dogaja z našimi serverji.

Kako hranimo podatke v OpenSearch?

Dokument / Document

- Dokument je enota, ki shranjuje informacije (besedilo ali strukturirane podatke). V OpenSearch so dokumenti shranjeni v formatu JSON.
- En dokument pomeni en zapis (vrstico) v podatkovni bazi.
- Ko iščemo informacije, OpenSearch vrne dokumente, povezane z našim iskanjem.

```
{
  "ime": "Jurka",
  "ocena": 5.0,
  "leto_pridelave": 2020
  "kraj_pridelave": "Središče",
  "smo_spili": true
}
```

Indeks / Index

- Zbirka dokumentov / a collection of documents.
- V relacijski podatkovni bazi bi indeks predstavljal tabelo.
- Ko iščemo, vbistvu poizvedujemo po podatkih, ki jih hrani indeks.

In vino veritas:

Ime	Ocena	Leto pridelave	Kraj pridelave	Smo spili
Jurka	5.0	2020	Središče	Da
Refošk	4.6	2019	Koper	Ne
Modra frankinja	4.9	2021	Bizeljsko	Da

Primer indeksa v OpenSearch

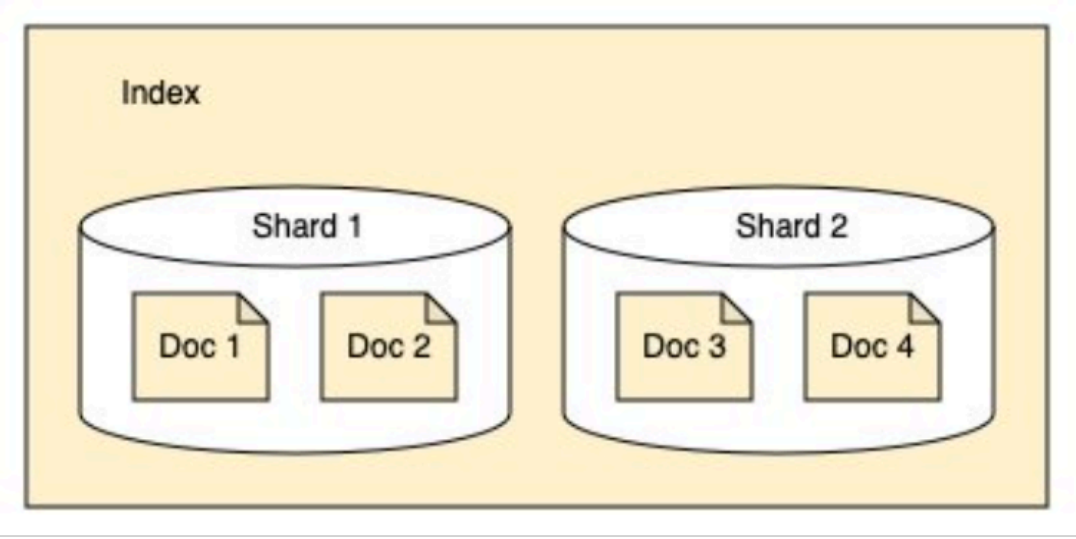
```
{
  "index": {
    "_index": "vina"
  },
  "data": [
    {
      "ime": "Jurka",
      "ocena": 5.0,
      "leto_pridelave": 2020,
      "kraj_pridelave": "Središče",
      "smo_spili": true
    },
    {
      "ime": "Refošk",
      "ocena": 4.6,
      "leto_pridelave": 2019,
      "kraj_pridelave": "Koper",
      "smo_spili": false
    },
    {
      "ime": "Modra frankinja",
      "ocena": 4.9,
      "leto_pridelave": 2021,
      "kraj_pridelave": "Bizeljsko",
      "smo_spili": true
    }
  ]
}
```

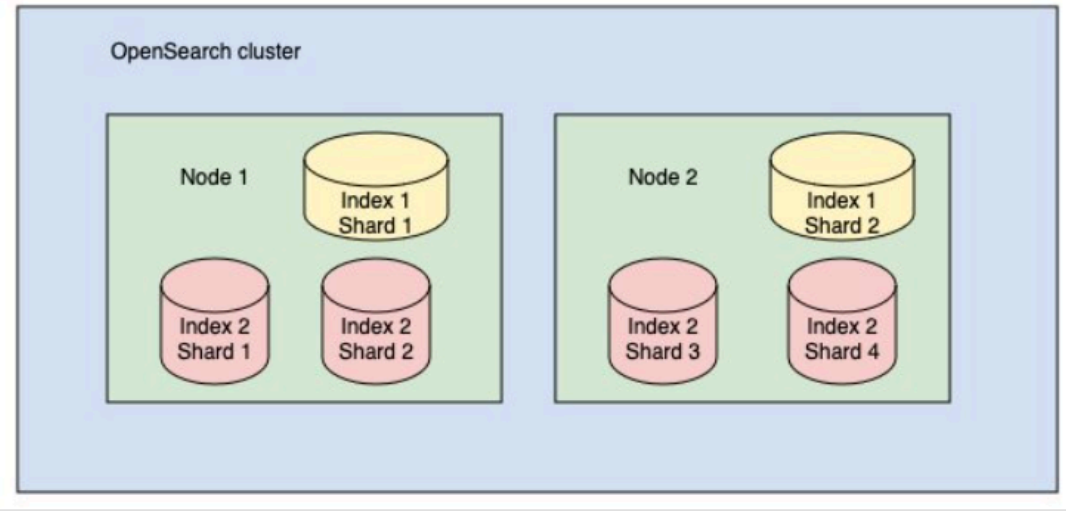
Gruče in vozli / Clusters and nodes

- OpenSearch je zasnovan kot porazdeljeni iskalnik; lahko deluje na enem ali več strežnikih (vozliščih), ki shranjujejo podatke in obdelujejo poizvedbe.
- Gruča (cluster) - je zbirka vozlišč / vozlov.
- V gruči z enim samim vozliščem mora ena naprava opraviti vse naloge: upravljati stanje gruče, indeksirati, predobdelati podatke pred indeksiranjem, ipd.
- Ker s podatki gruča raste, lahko naloge razdelimo na več vozlov in poskrbimo za nemoteno delovanje sistema na večih strežnikih.
- **Cluster manager node** - usklajuje operacije na ravni gruče, kot npr. ustvarjanje indeksa. Vozli med seboj komunicirajo; več vozlišč, hitre poizvedbe.

Razseki / Shards

OpenSearch razdeli indekse na razseke. Vsak razsek shranjuje podmnožico vseh dokumentov znotraj indeksa.



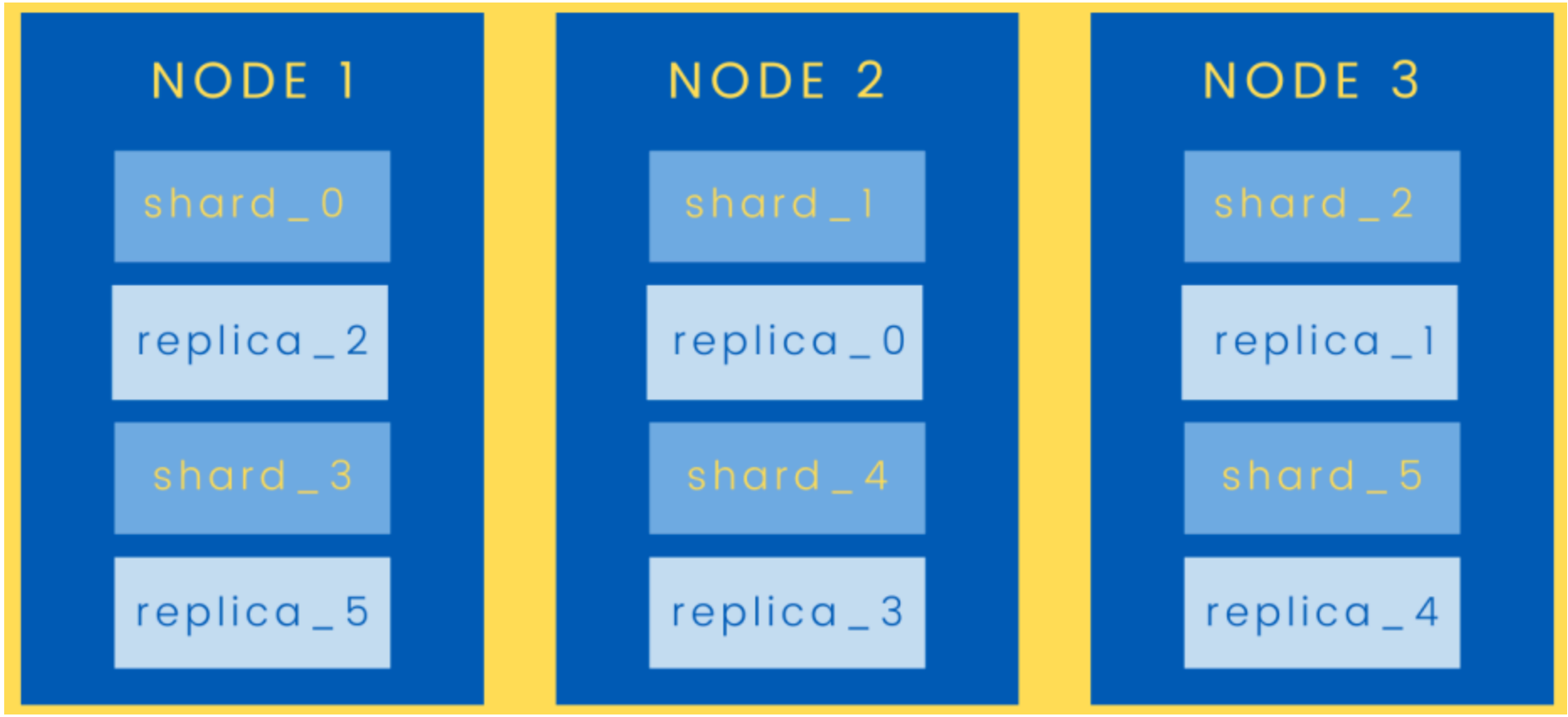


Razseki se uporabljajo za enakomerno porazdeljevanje po vozliščih v gruči.

Primer: Indeks velikosti 400 GB je morda prevelik, da bi ga eno samo vozlišče v clustru obvladalo, ampak, če ga razdelimo na 10 razsekov po 40 GB je vse lažje.

Replike / Replicas

Razsek je primarni (primary) ali replika (replica). OpenS. ustvari replika razsek za vsak primarni razsek. Če indeks razdelimo na 10 razsekov, OpenSearch ustvari 10 replik.



Indeksne predloge / Index templates

- Omogočajo preslikavo novih indeksov z vnaprej določenimi nastavitvami.

```
PUT _index_template/vina
{
  "index_patterns": ["vina*"],
  "template": {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 1
    },
    "mappings": {
      "properties": {
        "ime": { "type": "text" },
        "ocena": { "type": "float" },
        "leto_pridelave": { "type": "integer" },
        "kraj_pridelave": { "type": "keyword" },
        "smo_spili": { "type": "boolean" }
      }
    }
  }
```

OpenSearch in GEO

Geopoint

- točka s koordinatami: (lat, lon): `{ "point": { "lat": 40.71, "lon": 74.00 } }`

Geoshape

- OS razdeli poligon v trikotniško mrežo in shrani vsak trikotnik v BKD drevo.
- Formati: GeoJSON in WKT.
- Podatkovni tipi:
  - point, multipoint,
  - linestring, multilinestring,
  - polygon, multipolygon
  - geometrycollection, envelope

Cartesian field types

- Indeksirajo in iščejo 2D točke na ravnini, ne na sferi. So hitrejše za razvrščanje po razdalji kot geografski tipi. Natančnost: float z eno decimalko.
- Dva tipa:
  - xy_point
  - xy_shape

```
{ "point": { "x": 0.5, "y": 4.5 } }
```

```
{"location":{"type":"linestring","coordinates":[[0.5,4.5],[-1.5,2.3]]}}
```

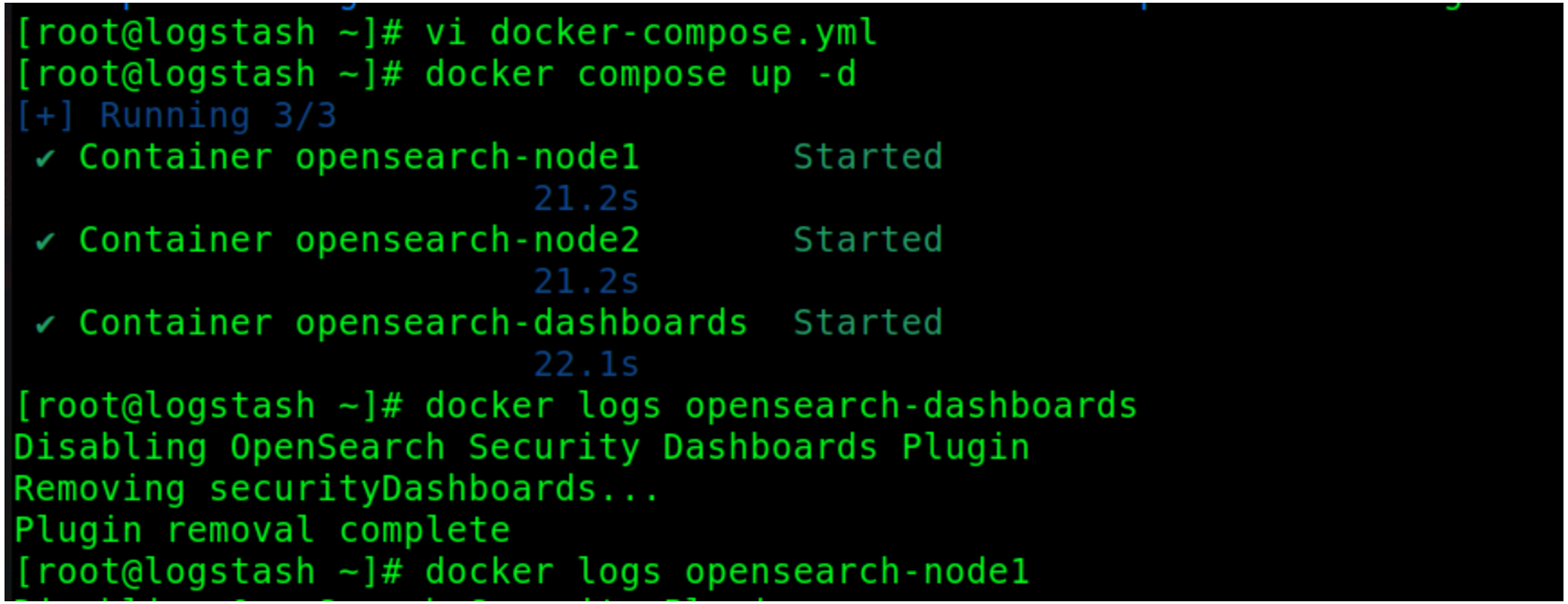
{} vinograd.json > {} _source > {} location

```json
{
    "_index": "slovenia_vineyards",
    "_id": "vinogradi_87_sredisce_06",
    "_version": 1,
    "_source": {
      "name": "Vinogradi ",
      "type": "Vineyard",
      "location": {
        "lat": 46.7716,
        "lon": 16.3060
      },
      "area": {
        "type": "polygon",
        "coordinates": [
        [
          [16.305551, 46.772072],
          [16.305920, 46.771487],
          [16.305974, 46.771508],
          [16.306275, 46.771103],
          [16.306505, 46.771201],
          [16.305829, 46.772140]
        ]
      ]
      },
      "region": "Goričko",
      "grape_varieties": ["Jurka", "Isabela", "Modra Frankinja"],
      "tags": ["vineyard", "wine", "organic", "Središče"],
      "elevation_m": 250,
      "annual_production_liters": 1000,
      "last_updated": "2025-04-08T07:00:00Z"
    }
  }
```

Namestitev / zagon

- Navodila na Githubu.
- https://opensearch.org/docs/latest/install-and-configure/install-opensearch/docker/
- YAML in Docker Compose
- `docker compose up -d`
- OOM exception, set: `sysctl -w vm.max_map_count=262144`
- OpenSearch dostopen na: http://localhost:5601/
- Uporabniško ime in geslo se lahko dodata v *docker-compose.yml* konfiguracijo.

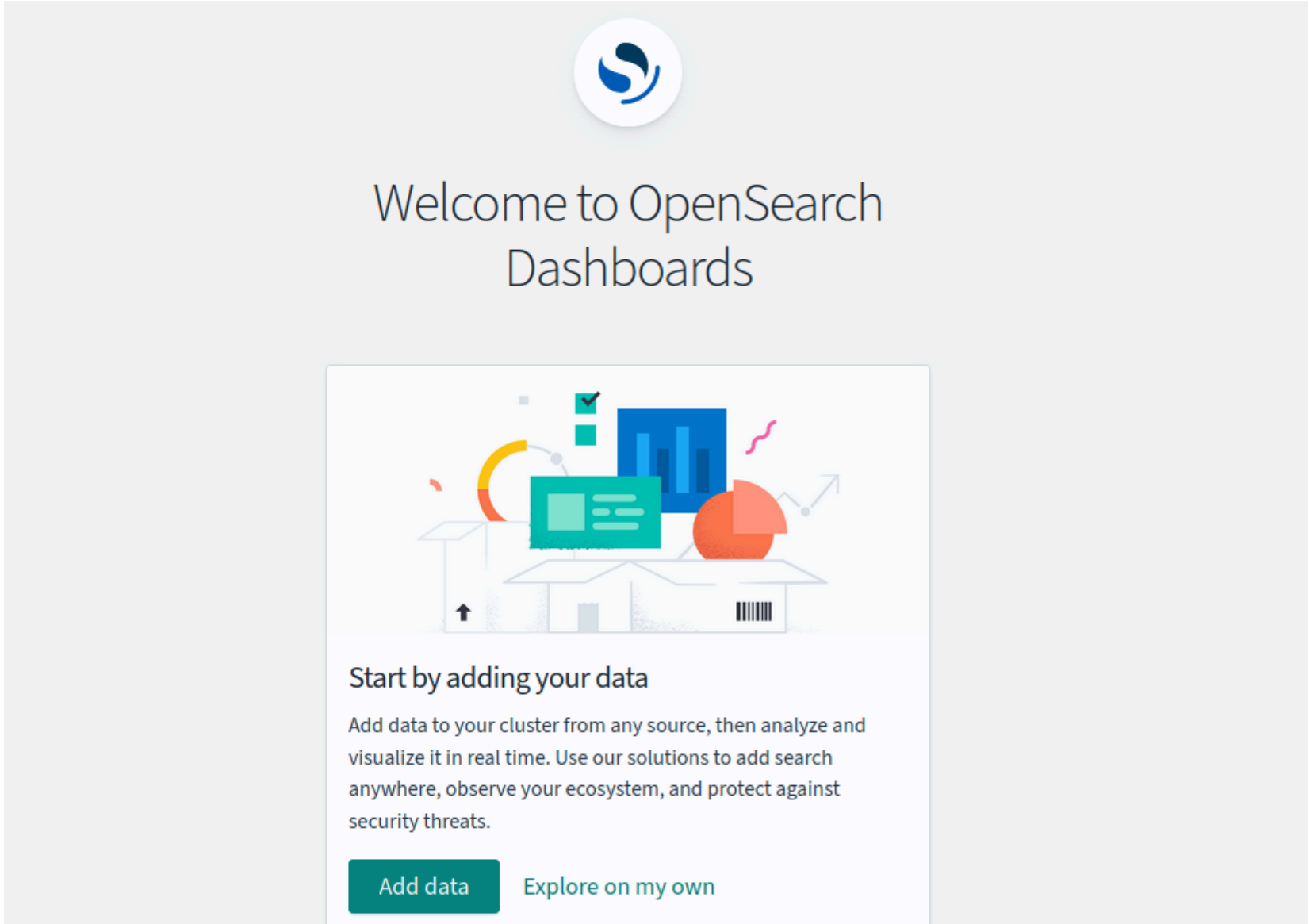
Our sample file creates two OpenSearch nodes and one OpenSearch Dashboards node with the Security plugin disabled.



Malo je treba potrpeti in pregledati log datoteke, in počakati da se vse naloži.

```
docker logs opensearch-dashboards
docker logs opensearch-node1
docker logs opensearch-node2
```

http://localhost:5601/



Preprost Geo primer

Primer geodeta ki z GNSS-jem meri na Goričkem pomeri prikazan vinograd, ker je alternativec ne uporablja Leice, ampak ruski Emlid. Meritve pošilja v pisarno v Mursko Soboto, kjer teče OpenSearch in ustvari preprost Dashboard.

Dober sosed Julius radovedno opazuje geodeta in ga po meritvah povabi pod senco murve na kosilo in na kozarec domače Jurke.

Zgodba bi se lahko začela tudi drugače:

- *Dolnji Slaveči, idilična vas na še bolj idiličnem Goričkem, kjer živijo daleč od tega ponorelega sveta prijazni, predsvem pa marljivi ljudje, ki se znajo skregati ...*

Workflow

1. Uvozimo podatke s curl ali ročno z DevTools v OpenSearch.
2. Preverimo če so podatki naloženi.

   `GET /meritve_sredisce/_search { "query": {  "match_all": {} } }`
3. Iskanje znotaj DevTools je možno z QueryDSL, npr.

```
GET /meritve_sredisce/_search
{
  "query": {
    "term": {
      "fix_quality": "2D"
    }
  }
}
```

4. Definiramo preslikavo podatkovih tipov z **Index Patterns**.
5. Usvarimo si svoj **Dashboard**.
6. **Discover** nam omogoča **iskanje po podatkih**.
7. **Anomaly detection** nam lahko najde anomalije; GNSS meritve s šumom.

Dodatne funkcionalnosti

- Vector search.
- Naučimo si modele strojnega učenja na lastnih podatkih.
- GenAI (RAG, LLMs).

Vprašanja?

Hvala ekipi OsGeo Slovenija <3

Keep it free and open

PS: Lepo je biti doma.

OpenSearchCon Europe 2025

- **Kdaj?** 30. april – 1. maj, Amsterdam
- https://events.linuxfoundation.org/opensearchcon-europe/