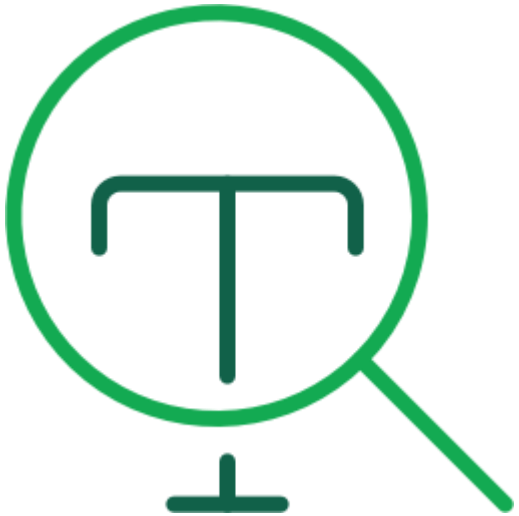
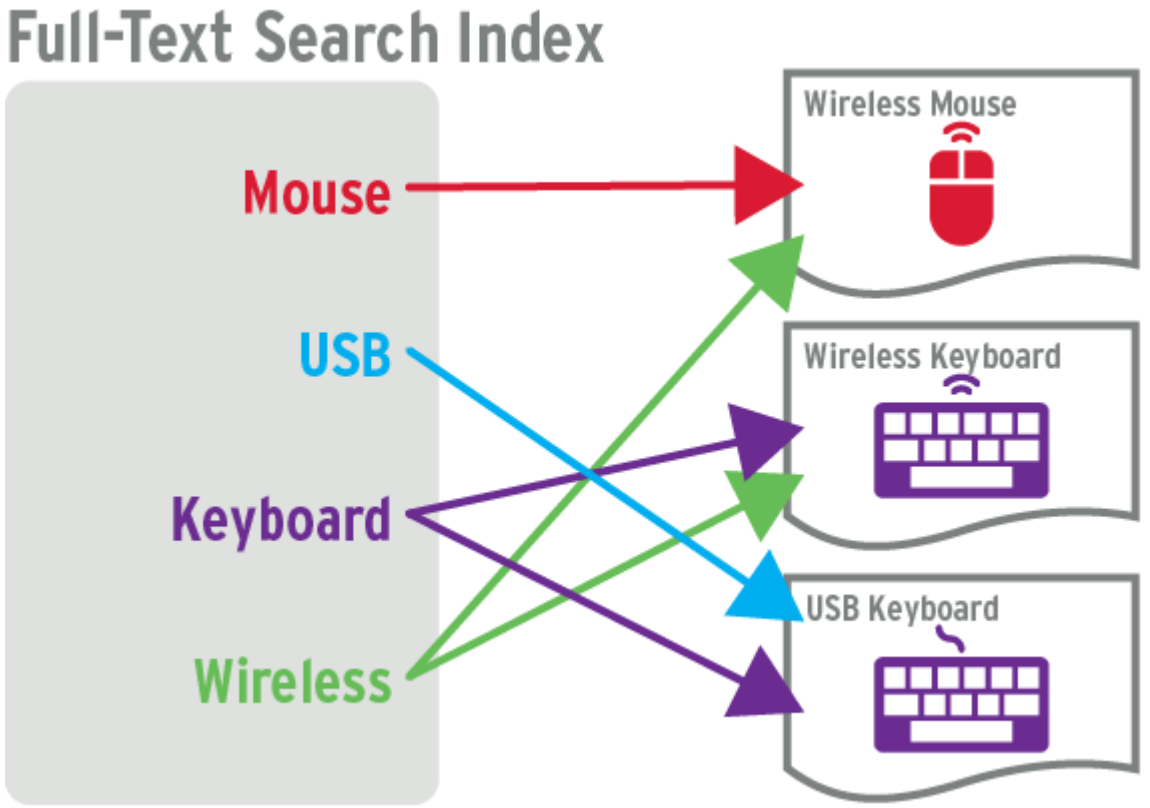
Full-text search



Apache Lucene

- Обратный индекс



Elasticsearch

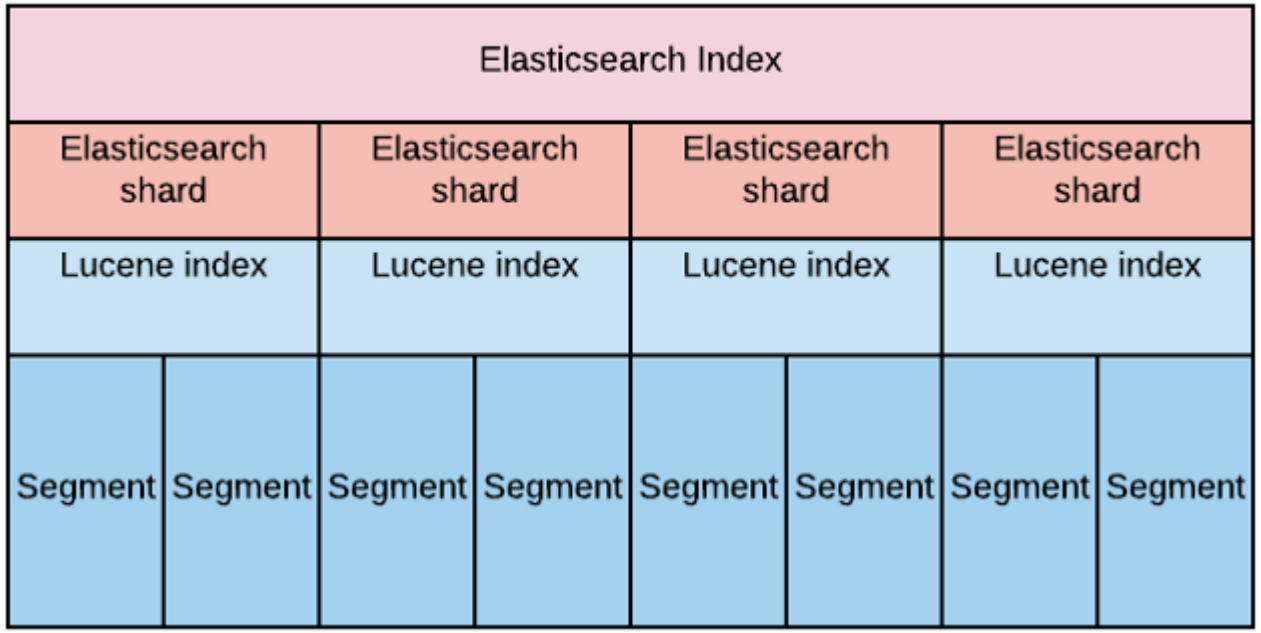
Keywords:

- Index
- Shards
- Analyzer
- Tokenizer (N-gram, Edge N-gram, standard)
- Mapping

download: https://www.elastic.co/downloads/elasticsearch

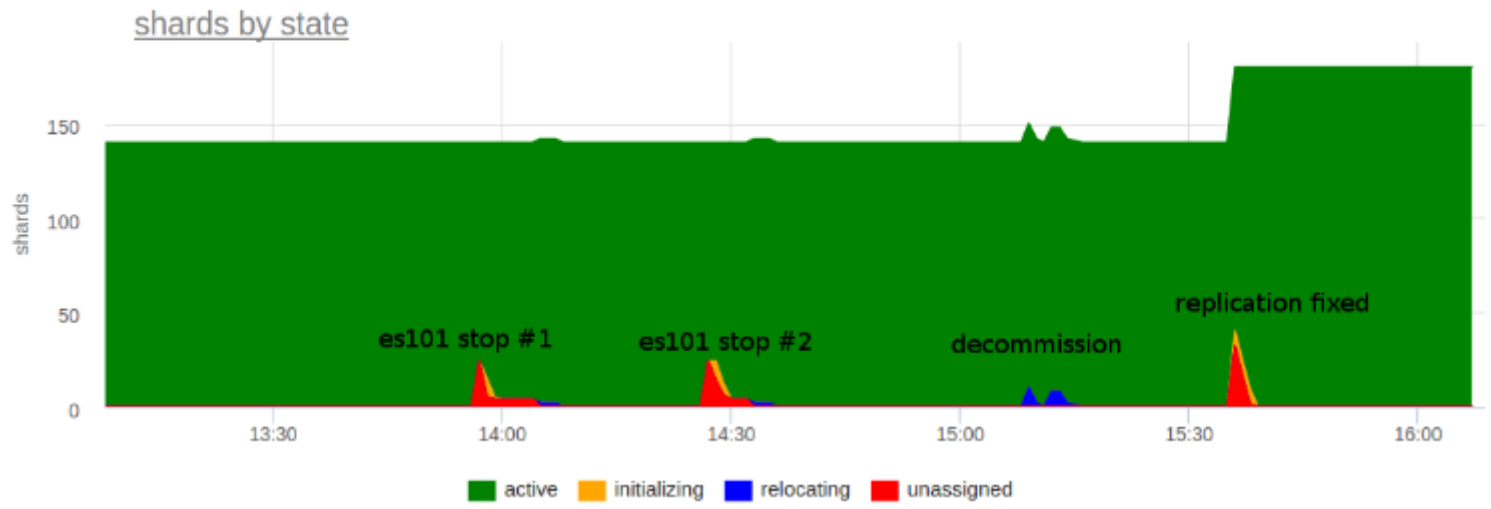
Elasticsearch

- Shards



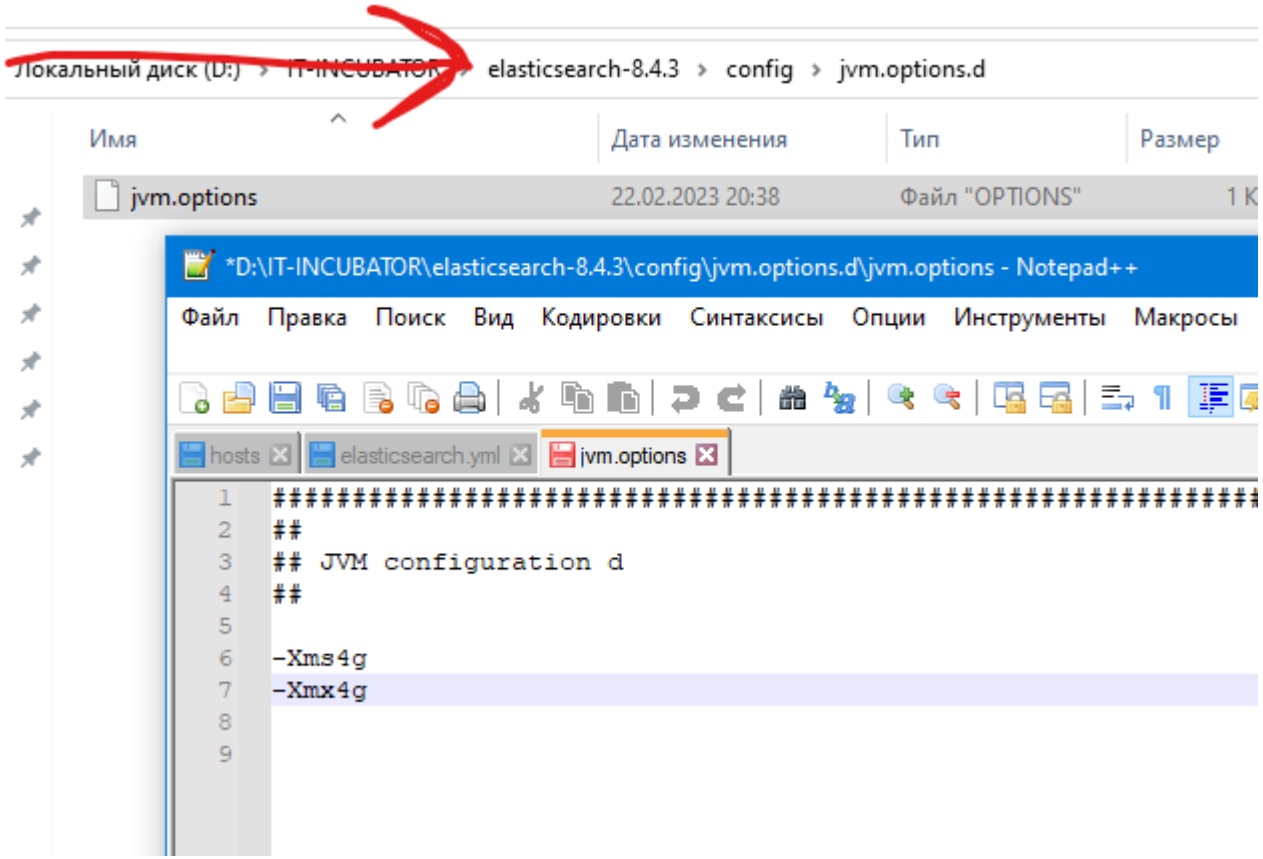
Elasticsearch

- Shards



JVM-env-settings

- max memory size JVM



Elastic-env-settings

- auth

кальный диск (D:) › IT-INCUBATOR › elasticsearch-8.4.3 › config

D:\IT-INCUBATOR\elasticsearch-8.4.3\config\elasticsearch.yml - Notepad++

Файл Правка Поиск Вид Кодировки Синтаксисы Опции Инструменты Макросы Запуск Плагины Вкладки ?

hosts | elasticsearch.yml | jvm.options

```
94  #
95  # ----------------------------------------------------------------------------
96
97  # Enable security features
98  xpack.security.enabled: false
99
100 xpack.security.enrollment.enabled: false
101
102 # Enable encryption for HTTP API client connections, such as Kibana, Logstash, and Agents
103 xpack.security.http.ssl:
104   enabled: false
105   keystore.path: certs/http.p12
106
107 # Enable encryption and mutual authentication between cluster nodes
108 xpack.security.transport.ssl:
109   enabled: false
110   verification_mode: certificate
111   keystore.path: certs/transport.p12
112   truststore.path: certs/transport.p12
113 # Create a new cluster with the current node only
114 # Additional nodes can still join the cluster later
115 cluster.initial_master_nodes: ["DESKTOP-MAIN"]
116
117 # Allow HTTP API connections from anywhere
118 # Connections are encrypted and require user authentication
119 http.host: 0.0.0.0
120
121 # Allow other nodes to join the cluster from anywhere
122 # Connections are encrypted and mutually authenticated
123 #transport.host: 0.0.0.0
124
125 #----------------------- END SECURITY AUTO CONFIGURATION -------------------------
126
127 #Custom properties
128 ingest.geoip.downloader.enabled: false
```

Elasticsearch

properties

edit

Type mappings, `object` fields and `nested` fields contain sub-fields, called `properties`. These properties may be of any data type, including `object` and `nested`. Properties can be added:

- explicitly by defining them when creating an index.
- explicitly by defining them when adding or updating a mapping type with the update mapping API.
- dynamically just by indexing documents containing new fields.

Below is an example of adding `properties` to a mapping type, an `object` field, and a `nested` field:

```
PUT my-index-000001
{
  "mappings": {
    "properties": { 1
      "manager": {
        "properties": { 2
          "age":  { "type": "integer" },
          "name": { "type": "text"  }
        }
      },
      "employees": {
        "type": "nested",
        "properties": { 3
          "age":  { "type": "integer" },
          "name": { "type": "text"  }
        }
      }
    }
  }
}
```

Elasticsearch

`term_vector`

Term vectors contain information about the terms produced by the analysis process, including:

- a list of terms.
- the position (or order) of each term.
- the start and end character offsets mapping the term to its origin in the original string.
- payloads (if they are available) — user-defined binary data associated with each term position.

These term vectors can be stored so that they can be retrieved for a particular document.

The `term_vector` setting accepts:

Elasticsearch

Tokenizer (N-gram, Edge N-gram, standard)

Elasticsearch

- Analyzer vs Search analyzer

Elasticsearch

`fields`

edit

It is often useful to index the same field in different ways for different purposes. This is the purpose of *multi-fields*. For instance, a `string` field could be mapped as a `text` field for full-text search, and as a `keyword` field for sorting or aggregations:

```
PUT my-index-000001
{
  "mappings": {
    "properties": {
      "city": {
        "type": "text",
        "fields": {
          "raw": { 1
            "type":  "keyword"
          }
        }
      }
    }
  }
}
```

Elasticsearch

Geo queries

Elasticsearch supports two types of geo data: `geo_point` fields which support lat/lon pairs, and `geo_shape` fields, which support points, lines, circles, polygons, multi-polygons, etc.

The queries in this group are:

`geo_bounding_box` **query**
: Finds documents with geoshapes or geopoints which intersect the specified rectangle.

`geo_distance` **query**
: Finds documents with geoshapes or geopoints within the specified distance of a central point.

`geo_grid` **query**
: Finds documents with:

- Geoshapes or geopoints which intersect the specified geohash
- Geoshapes or geopoints which intersect the specified map tile
- Geopoints which intersect the specified H3 bin

`geo_polygon` **query**
: Find documents with geoshapes or geopoints which intersect the specified polygon.

`geo_shape` **query**
: Finds documents with geoshapes or geopoints which are related to the specified geoshape. Possible spatial relationships to specify are: intersects, contained, within and disjoint.

Analyzer	Description
Standard	Uses the default analyzer for all Atlas Search indexes and queries.
Simple	Divides text into searchable terms wherever it finds a non-letter character.
Whitespace	Divides text into searchable terms wherever it finds a whitespace character.
Language	Provides a set of language-specific text analyzers.
Keyword	Indexes text fields as single terms.


```
{
  $text:
    {
      $search: <string>,
      $language: <string>,
      $caseSensitive: <boolean>,
      $diacriticSensitive: <boolean>
    }
}
```

- Tsvector
- Tsquery
- to_tsvector
- to_tsquery (*plainto_tsquery, phraseto_tsquery, websearch_to_tsquery*)
- ts_rank (ts_rank_cd)

Материалы

- https://lucene.apache.org/core/9_5_0/index.html
- https://www.elastic.co/guide/index.html
- https://www.mongodb.com/docs/manual/text-search/
- https://www.postgresql.org/docs/10/textsearch.html
- https://habr.com/ru/post/174457/
- https://habr.com/ru/company/likeastore/blog/223109/ переходим от монго к еластик
- https://albinjose.medium.com/full-text-search-in-mongodb-using-mongoose-b4154bc3a41b
- https://habr.com/ru/post/442170/ опыт полнотекстового поиска postgres
- https://codedzen.ru/category/uroki/elasticsearch/