elastic search

- Real time distributed data engine
- supports full-text search
- completely document based instead of tables | schemas
- used for: Single Page App Projects

¿Why?

query

- perform and combine many types of search like structured, unstruct., geo, metrics, etc.
- can ask query "anyway you want"

analyze

- understand billions of log lines easily
- provides aggregation which help you zoom out to explore trends and patterns of the data

So, for example, we could search the files where the place is "Dubai" or based on the geography.

- scalability
- really fast
- multilingual
- document oriented
- autocompletition & instantsearch
- schema free

Basic Concepts

- **near real time:** sligth time from doc. indexing → doc. searchable

- **cluster:**
  - collection of 1+ nodes that togetner hold the entire data.
  - It provides indexing and search capabilities across all nodes.
  - Identified by an unique name

- **node:** single server, part of the cluster
  stores data
  partiapates in cluster indexing and search capabilities

- **Index:** collection of documents identified by a name. → similar characteristics

- **Type:** category of an index

- **Document:** JSON. can be indexed.

- **shards:** subdivision of an index.
  fully-functional independant "index" that can be hosted on any node within the cluster

- **replicas:** replicas of the shards

• API Conventions

multiple indices

different notations

- comma separated
- wildcard not. (a, t, -)
- URL query string param (_)

date math support in index name

format

<static_name { date_math_expr { date_format | time_zone }}>

- static_name: static text part of name
- date_math_expr: computes date dynamically
- date_format: optional date format
- time_zone: optional time zone

common options

Pretty Result	Time Units
Human Readable Output	Byte Size Units
Date Math	Unit-less quantities
Response Filtering	Distance Units
Flat Settings	Fuzziness
Parameter	Enabling Stack Traces
No Values	Request Body In Query String

URL based Access Control

- can also use proxy
- can specify an index in the URL and on individual requests within the request body such as:
  - multi-search
  - bulk
  - multi-get

• API's

- Document API → perform operation
- search API → search across indexes
- aggregation API → aggregation
- Index API → op. at index level
- cluster API → op. at cluster level

↳ Document API

* kibana

Single Document API

- Index API
- Get API
- Update API
- Delete API

Multi-Document API

- Multi Get API
- Bulk API
- Delete By Query API
- Update By Query API
- Reindex API

Operations :

- create — PUT
- update — PUT
- read — GET
- delete — DELETE

} CRUD

↳ Search API

Using a search API, you can execute a search query and get back search hits that match the query

Multi Index — You can search for the documents present in all the indices or in some specific indices

Multi - Type — You can search all the documents in an index across all types or in some specified type

URI Search — Various parameters can be passed in a search operation using Uniform Resource Identifier:

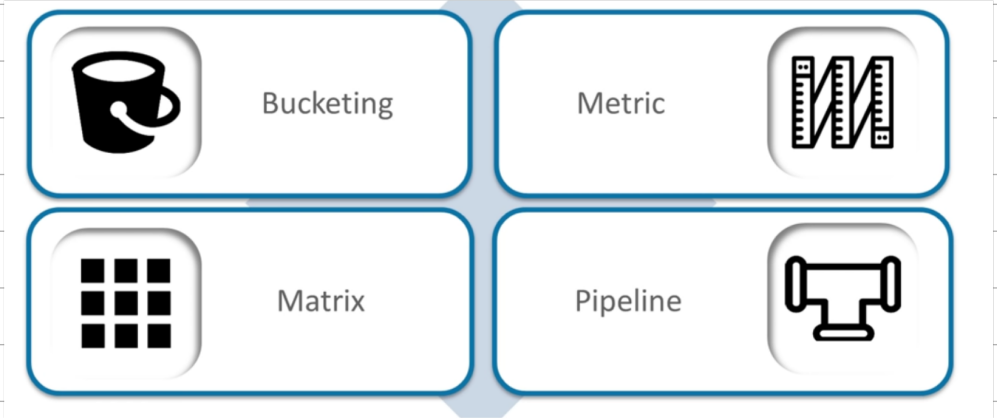
- q
- lenient
- fields
- sort
- timeout
- terminate_after
- from
- size

↳ Aggregation API

- collects data which is selected by the search query

- aggregators → build complex summaries of data

TYPES



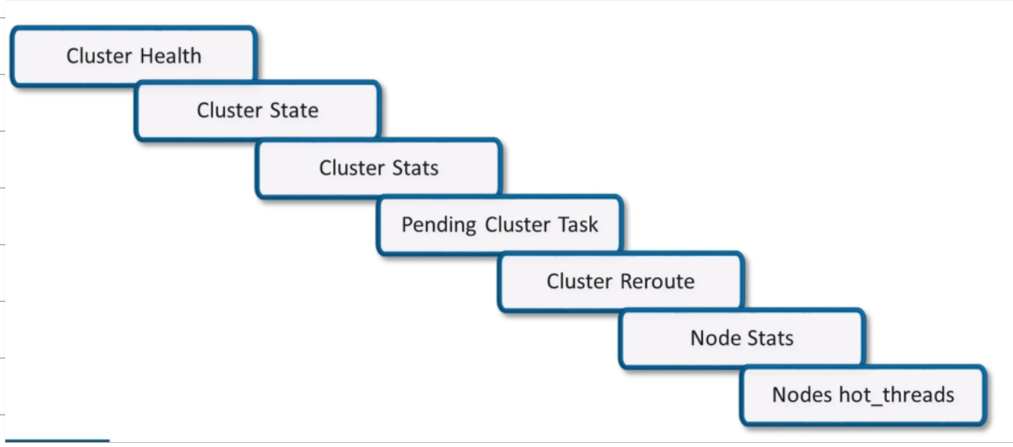
↳ Index API

manages settings, aliases, mappings, templates

- Create Index
- Delete Index
- Get Index
- Index Exits
- Open/ Close Index API
- Index Aliases
- Index Settings
- Analyze
- Index Template
- Index Stats
- Flush
- Refresh

↳ Cluster API

getting info about clusters and nodes and changing them



Query DSL

domain specific language

- AST of queries
  two types of clause:

  1. Leaf query clause: looks for a particular value in a particular field

  2. Compound query: combines leaf q.c. with others.

Mapping

- Defining how a document and its fields are stored and indexed

- **Types:**
  - meta-fields
  - fields or properties

- **Field data types:**
  - core data
  - complex data
  - geo data
  - specialized data

Dynamic Mapping

Mapping Parameters

- ✓ Analyzer
- ✓ Boost
- ✓ Coerce
- ✓ copy_to
- ✓ doc_values
- ✓ Dynamic
- ✓ Enabled
- ✓ Fielddata
- ✓ Geohash
- ✓ geohash_precision
- ✓ geohash_prefix
- ✓ Format
- ✓ ignore_above
- ✓ ignore_malformed
- ✓ include_in_all
- ✓ index_options
- ✓ lat_lon
- ✓ Index
- ✓ Fields
- ✓ Norms
- ✓ null_value
- ✓ position_increment_gap
- ✓ Properties
- ✓ search_analyser
- ✓ Similarity
- ✓ Store
- ✓ term_vector

Analysis

how data is configured

↳ during search the index's content is analysed by analysis module

1. Analyzers
2. Tokenizer
3. Token filter
4. character filter

• Modules

static setting ↓

have to be configured before hand
(elasticsearch.yml)

dynamic setting ↓

can be set on live

Types: