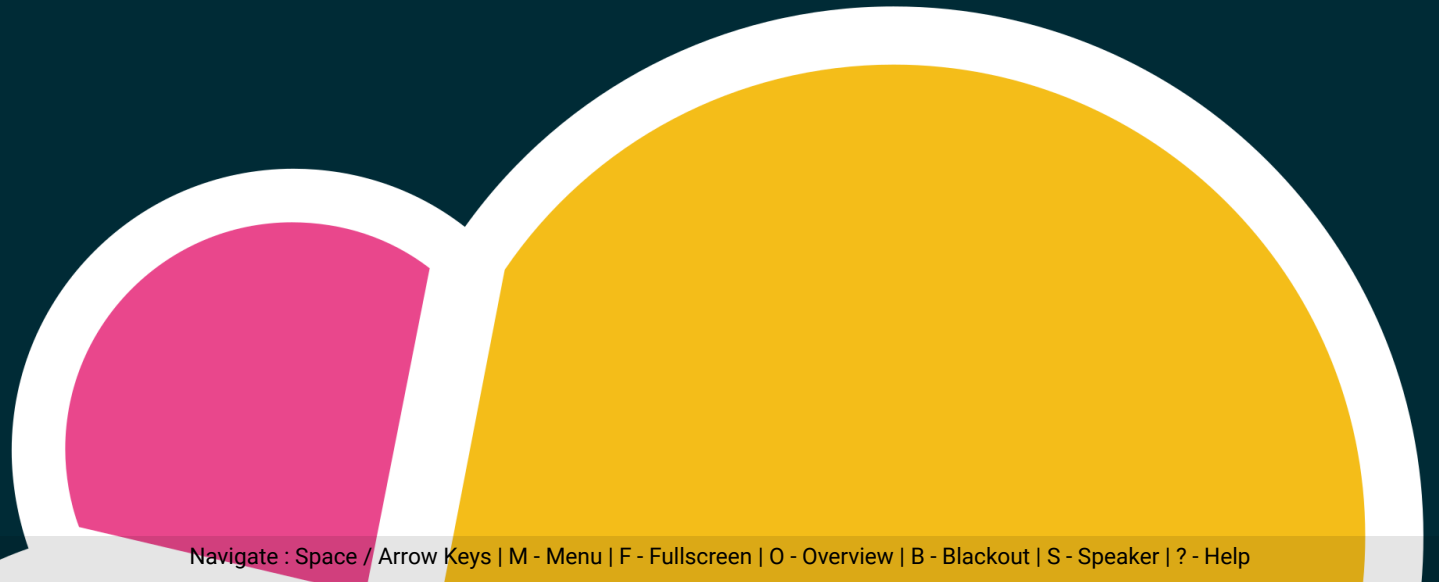
ELASTICSEARCH CRASH COURSE



A cura di Guglielmo Piacentini e Alfredo Serafini

CHE COS'È ELASTICSEARCH?

Un motore di ricerca e analisi full-text, open-source.

COME LO USO?

- Hai un negozio online e vuoi permettere ai tuoi clienti di ricercare nel catalogo, utilizzando dei suggerimenti e il completamento automatico. Il catalogo e i tuoi prodotti verranno salvati in Elastic.
- Vuoi analizzare file di log e statistiche per estrarne conoscenza. Una volta portati i dati su Elastic puoi aggregare e ricercare per estrarre informazioni di interesse.

- Hai una lista di indirizzi scritti nei modi più disparati, utilizzi Elastic come un dizionario di controllo per costruire una lista di indirizzi normalizzati.
- Hai un prodotto commerciale che si basa su analisi full-text/semantiche (i.e. Pupilla). Elastic è la spina dorsale su cui vengono sia salvati i documenti che analizzati.

COM'È FATTO?

Elasticsearch gira in cluster. Un cluster può essere formato da uno o più server. Ogni server nel cluster è un nodo. ES stocca i documenti in indici. Un indice può essere "scomposto" in shard.

ES si basa su Apache Lucene per l'indicizzazione dei documenti. Ogni shard è un indice Lucene.

LET'S GET OUR HANDS DIRTY.

DEMO

INSTALLAZIONE ES

Link

Creiamo un indice

```
PUT /customer?pretty
```

Chiediamo a ES la lista degli indici

```
GET /_cat/indices?v
```

La risposta:

```
health status index    uuid                   pri rep docs.
yellow open   customer 95SQ4TSUT7mWBT7VNHH67A   5   1
```

Inseriamo un documento semplice

```
PUT /customer/_doc/1?pretty
{
  "name": "John Doe"
}
```

La risposta:

```
{
  "_index" : "customer",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 1,
  "result" : "created"
  [...]
}
```

Ricerchiamo il documento appena inserito

```
GET /customer/_doc/1?pretty
```

La risposta:

```
{
  "_index" : "customer",
  "_type" : "_doc",
  "_id" : "1",
  "_version" : 1,
  "found" : true,
  "_source" : { "name": "John Doe" }
}
```

Eliminiamo l'indice creato

```
DELETE /customer?pretty
```

Richiamiamo la lista degli indici

```
GET /_cat/indices?v
```

La risposta sarà vuota

INSTALLAZIONE CEREBRO

Link

Cerebro è una GUI che permette di interagire con Elasticsearch.

REST API

Elastic espone delle API molto potenti che permettono di:

- Controllare lo stato di salute e le statistiche del cluster
- Amministrare il cluster, i nodi, gli indici, i dati e i metadati
- Performare CRUD (Create, Read, Update, and Delete) e operazioni di ricerca sugli indici
- Eseguire operazioni avanzate come paging, sorting, filtering, aggregazioni...

HANDS ON PT. II

Dataset accounts.json

Bulk insert (anche tramite Cerebro):

```
curl -H "Content-Type: application/json"
-XPOST "localhost:9200/bank/_doc/_bulk?pretty&refresh"
--data-binary "@accounts.json"
```

LA RICERCA TRAMITE URI

```
GET /bank/_search?q=*&sort=account_number:asc&pretty
```

QUERY DSL

La più semplice:

```
GET /bank/_search
{
  "query": { "match_all": {} }
}
```

Sorting:

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "sort": { "balance": { "order": "desc" } }
}
```

Match Query:

```
GET /bank/_search
{
  "query": { "match": { "address": "mill lane" } }
}
```

Filtri:

```
GET /bank/_search
{
  "query": {
    "bool": {
      "must": { "match_all": {} },
      "filter": {
        "range": {
          "balance": {
            "gte": 20000,
            "lte": 30000 }}}}}}
```

Aggregazioni:

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_state": {
      "terms": {
        "field": "state.keyword"
      }
    }
  }
}
```

Documentazione Search API

Documentazione Query DSL

USE CASE

Regione Veneto:

```
"settings": {
 "number_of_shards": 3,
 "number_of_replicas": 2,
 "analysis": {
  "analyzer": {
   "simple_rebuilt_comuni": {
    "tokenizer": "lowercase",
    "filter": [
     "lowercase",
     "asciifolding",
     "synonym_comuni"
    ]
   },
   "simple_rebuilt_province": {
```

Documenti contenuti:


```
{
    "index": {
        "_index": "luoghi-istat",
        "_type": "luogo"
    }
}
{
    "nome_comune": "Ala di Stura",
    "nome_provincia": "",
    "sigla_provincia": "TO",
    "nome_regione": "Piemonte",
    "citta_metropolitana": "Torino",
    "codice_comune": 1003,
    "codice_provincia": 1,
```


INDEX MAPPING

Elastic si fonda sul concetto di indice inverso:

- Un indice inverso consiste nella lista di tutte le parole non ripetute che appaiono in un documento e, per ognuna di queste, una lista dei documenti nelle quali appaiono.
- Questo permette una ricerca full-text particolarmente rapida.

- Analyzer

ES permette l'uso di diversi analyzer out of the box, oltre che la possibilità di costruirne di custom.

```
GET _analyze
{
  "analyzer" : "standard",
  "text" : "this is a test"
}
```

Altri casi di test

N.B.

C'è una differenza sostianziale nella analisi a livello indice e l'analisi durante la ricerca.

Term Query VS Full-Text Query

More here

- Tokenizer

I tokenizer ricevono uno stream di caratteri e li dividono in token individuali a seconda del tokenizer (di solito singole parole)

- Token Filter (Synonym)

I token filter accettano e modificano gli stream di token provenienti dai tokenizer. Nello specifico il filtro synonym permette la trasformazione dei token secondo delle tabelle di sinonimi determinate dall'utente.

USE CASE #2

CRAIM: Il KM si appoggia ad ES per le funzionalità di ricerca full-text.

Il mapping di un unico indice dei documenti sul KM di CRAIM vede più di 5300 righe di codice.

```
"push": {
                "_all": {
                    "enabled": false
                },
                "_source": {
                    "excludes": ["formattedtext"]
                },
                "dynamic_templates": [{
                    "suggestion": {
                        "mapping": {
                            "analyzer": "suggestion_analyze
                            "type": "completion"
                        },
                        "match": "*_suggest_*"
```


```
{
    "document_raw_storage": "yes",
    "CHANNEL_ID": "69737",
    "entitymap": "rO0ABXNyABFqYXZhLnV0aWwuSGFzaE1hcAUH2sHDF
    "named_entity_Comuni": "Force[:]Mattinata[:]Romana[:]Pa
    "insertDate": "2017-07-11T08:07:19.296Z",
    "checksum": "4f2c990e29132757db0d59a7eb0aee29",
    "named_entity_Location": "Italia[:]Paese[:]Londra[:]Bre
    "CONTACT_ID": "FILESYSTEM_VOICE_71b1db82-7271-4780-b074
    "CHANNEL_NAME": "CanalePVT",
    "mimetype": "text/plain",
    "ONTOLOGY_FILENAME": "CraimInt_6",
    "CHANNEL_TYPE": "PUSH",
    "parentURI": "http://almawave.it/ontologies/2017/02/07/
```


THE END