Creating an elasticsearch index

Jay Chow

Software Engineer, Cisco eCommerce, IT

Cisco, San Jose

24th April 2020

Agenda

elastic

1. What is elasticsearch?
2. Benefits of elasticsearch
3. Interacting with elasticsearch
4. Creating an index
5. Inserting Data
6. Retrieving Data
7. Reference

What is elasticsearch?

- Elasticsearch is a search server based on lucene and provides a distributable full text search engine that’s accessible through a REST interface

- Schema-less and uses JSON, open source and built in Java, can run elasticsearch on any platform

- Document-based store, can be used as alternative to other document stores like MongoDB

- Elasticsearch compromises of nodes and clusters which are the center of the architecture

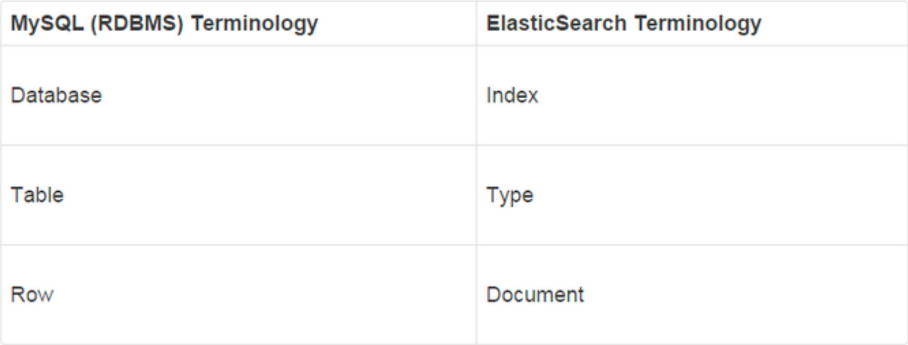
- node is a server that stores part of the data
- cluster is a collection of nodes
- nodes support searching new data, indexing and manipulating

Benefits of elasticsearch

1) Fast and Scalable in terms of read performance

2) REST APIs are used to interact with elasticsearch:

REST API Format : http://host:port/[index]/[type]/[_action/id]



MySQL (RDBMS) Terminology	ElasticSearch Terminology
Database	Index
Table	Type
Row	Document

A table comparing terminology of MySQL with ElasticSearch

Interacting with elasticsearch

- Once elasticsearch has started, any REST API client such as postman can be used
- ElasticSearch lets you use HTTP methods such as GETs, POSTs, DELETEs, and PUTs along with a JSON payload
- To get a list of all available indices in elasticsearch, use the following URL :

http://localhost:9200/_cat/indices

- To get the status of an index (say, company), use the following URL:

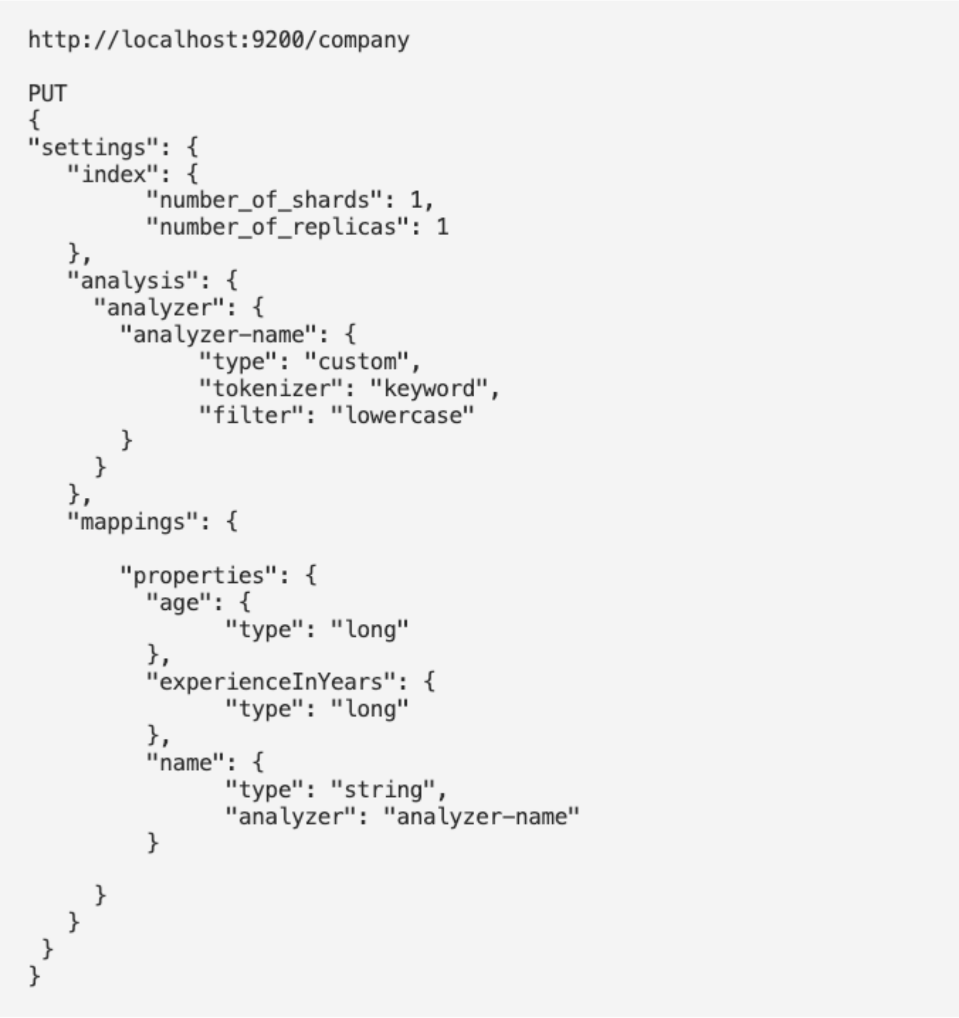
http://localhost:9200/company?pretty

- The first part (*localhost*) is denotes the **host** (server) where your ElasticSearch is hosted, and the default **port** of 9200:

http://localhost:9200/company/employee/_search

The second part (*company*) is **index** , followed by the (*employee*) **type** name, followed by (*_search*) **action**.

Creating an index



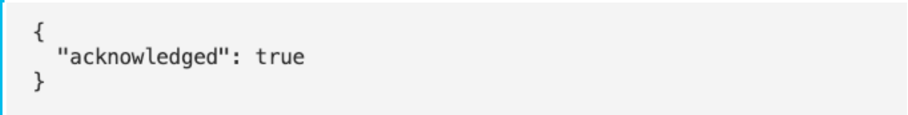
```
http://localhost:9200/company

PUT
{
"settings": {
   "index": {
         "number_of_shards": 1,
         "number_of_replicas": 1
   },
   "analysis": {
     "analyzer": {
       "analyzer-name": {
             "type": "custom",
             "tokenizer": "keyword",
             "filter": "lowercase"
       }
     }
   },
   "mappings": {

       "properties": {
         "age": {
               "type": "long"
         },
         "experienceInYears": {
               "type": "long"
         },
         "name": {
               "type": "string",
               "analyzer": "analyzer-name"
         }

     }
   }
 }
}
```

Creating an index (response received)

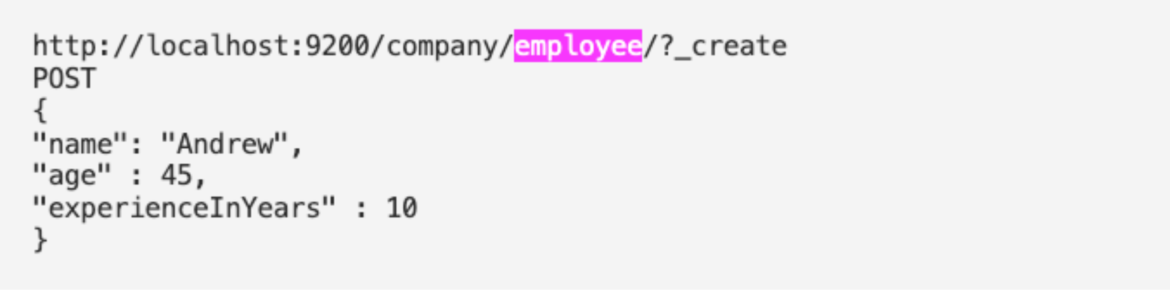


```
{
  "acknowledged": true
}
```

Previous command creates an index called Company with type named employee having fields called age, experiencedInYears and name

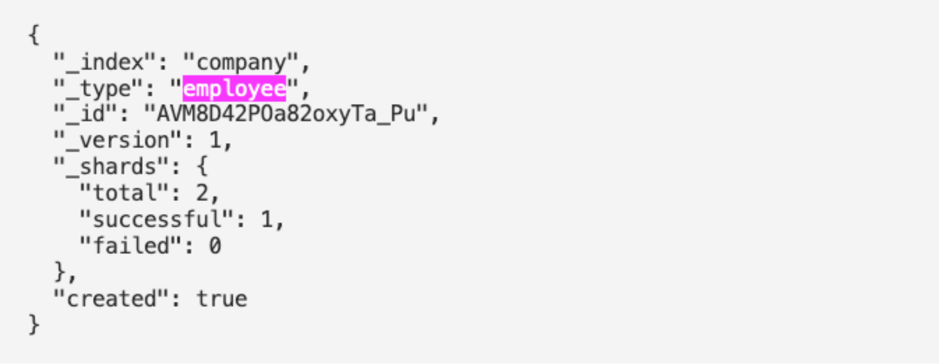
Note: type named employee was not in previous PUT payload

Inserting Data



```
http://localhost:9200/company/employee/?_create
POST
{
"name": "Andrew",
"age" : 45,
"experienceInYears" : 10
}
```

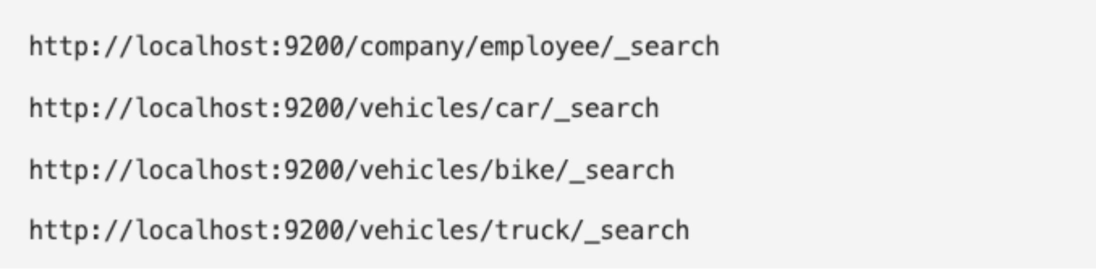
Response:



```
{
  "_index": "company",
  "_type": "employee",
  "_id": "AVM8D42P0a82oxyTa_Pu",
  "_version": 1,
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "created": true
}
```

Retrieving Data

- To read **all** records of a type within an index, use:



```
http://localhost:9200/company/employee/_search

http://localhost:9200/vehicles/car/_search

http://localhost:9200/vehicles/bike/_search

http://localhost:9200/vehicles/truck/_search
```

Note: under vehicles index, we have types such as company, car, bike and truck

- Also using RESTful APIs, we could check the status of the elasticsearch server, perform create, read, update, delete and search operations against indexes and Perform operations like paging, sorting, filtering, scripting, faceting and aggregations

Reference

https://medium.com/@ashish_fagna/getting-started-with-elasticsearch-creating-indices-inserting-values-and-retrieving-data-e3122e9b12c6