



Basic concepts about Elasticsearch

Andrés Felipe Alba Hernández
OPOS team



What is elasticsearch?

It is a search server based on Lucene (library written in java)

JSON Schema

http web
interface

How do ES (elasticsearch) queries work?

A ES query examines one or many target values and scores each of the elements in results according to how close they match the focus of the query. Simpler design than traditional databases.

Elasticsearch details.

- A distributed real-time document store where every field is indexed and searchable
- A distributed search engine with real-time analytics
- Capable of scaling to hundreds of servers and petabytes of structured and unstructured data
- You can use Java API, it use the port 9300
- You can talk to elasticsearch through RESTful API using the port 9200 (other language as python)

Let say something else about ES

You save entire objects or documents and you indexes the contents. Then you index, search, sort, and filter documents—not rows of columnar data.

HOW DOES ELASTICSEARCH SAVE YOUR OBJETS?

user object



```
{
  "email":      "john@smith.com",
  "first_name": "John",
  "last_name":  "Smith",
  "info": {
    "bio":      "Eco-warrior and defender of the weak",
    "age":      25,
    "interests": [ "dolphins", "whales" ]
  },
  "join_date": "2014/05/01"
}
```



Caño Cristales, Meta, Colombia

Ipiales, Colombia



Parallel between traditional data base and elasticsearch

Relational DB

Databases

Tables

Rows

Columns

Elasticsearch

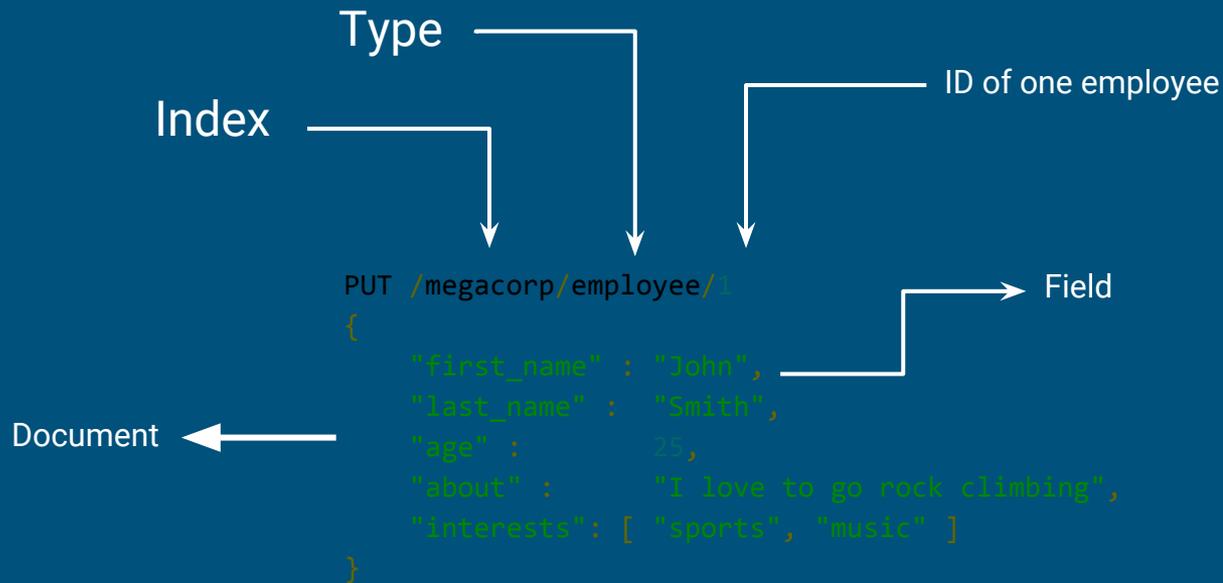
Indices

Types

Documents

Fields

Some examples



Queries

Command line query

→ `GET /megacorp/employee/_search?q=last_name:Smith`

query DSL (Domain-specific-language)

→

```
GET /megacorp/employee/_search
{
  "query" : {
    "match" : {
      "last_name" : "Smith"
    }
  }
}
```

The concepts of relevance

```
GET /megacorp/employee/_search
{
  "query" : {
    "match" : {
      "about" : "rock climbing"
    }
  }
}
```



```
{
  ...
  "hits": {
    "total":      2,
    "max_score": 0.16273327,
    "hits": [
      {
        ...
        "_score":      0.16273327,
        "_source": {
          "first_name": "John",
          "last_name":  "Smith",
          "age":        25,
          "about":      "I love to go rock climbing",
          "interests":  [ "sports", "music" ]
        }
      },
      {
        ...
        "_score":      0.016878016,
        "_source": {
          "first_name": "Jane",
          "last_name":  "Smith",
          "age":        32,
          "about":      "I like to collect rock
albums",
          "interests":  [ "music" ]
        }
      }
    ]
  }
}
```

Analytics

```
GET /megacorp/employee/_search
{
  "query": {
    "match": {
      "last_name": "smith"
    }
  },
  "aggs": {
    "all_interests": {
      "terms": {
        "field": "interests"
      }
    }
  }
}
```



```
...
  "all_interests": {
    "buckets": [
      {
        "key": "music",
        "doc_count": 2
      },
      {
        "key": "sports",
        "doc_count": 1
      }
    ]
  }
}
```

Analytics

```
GET /megacorp/employee/_search
{
  "aggs" : {
    "all_interests" : {
      "terms" : { "field"
: "interests" },
      "aggs" : {
        "avg_age" : {
          "avg" : {
            "field" : "age" }
        }
      }
    }
  }
}
```



```
...
"all_interests": {
  "buckets": [
    {
      "key": "music",
      "doc_count": 2,
      "avg_age": {
        "value": 28.5
      }
    },
    {
      "key": "forestry",
      "doc_count": 1,
      "avg_age": {
        "value": 35
      }
    },
    {
      "key": "sports",
      "doc_count": 1,
      "avg_age": {
        "value": 25
      }
    }
  ]
}
```



Santander, Colombia

Let's talk a little bit about the distributed nature

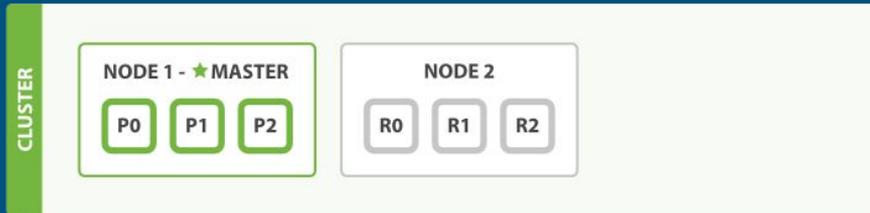
- A node is a running instance of Elasticsearch
- A cluster consists of one or more nodes with the same cluster.name that are working together to share their data and workload.

GET /_cluster/health



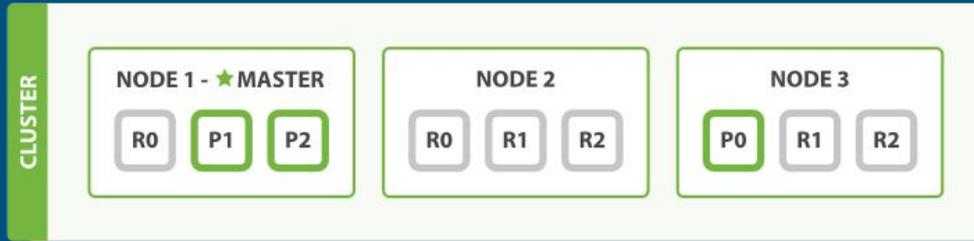
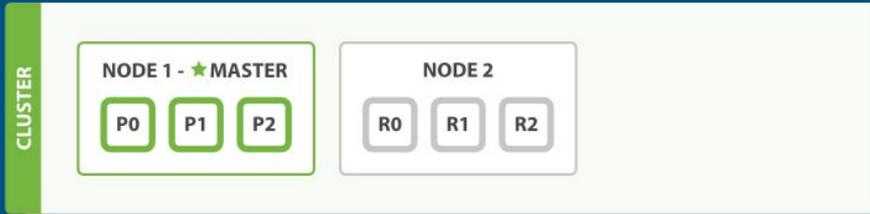
```
{
  "cluster_name":      "elasticsearch",
  "status":            "green",
  "timed_out":         false,
  "number_of_nodes":  1,
  "number_of_data_nodes": 1,
  "active_primary_shards": 0,
  "active_shards":    0,
  "relocating_shards": 0,
  "initializing_shards": 0,
  "unassigned_shards": 0
}
```

A little example how your clusters should look like

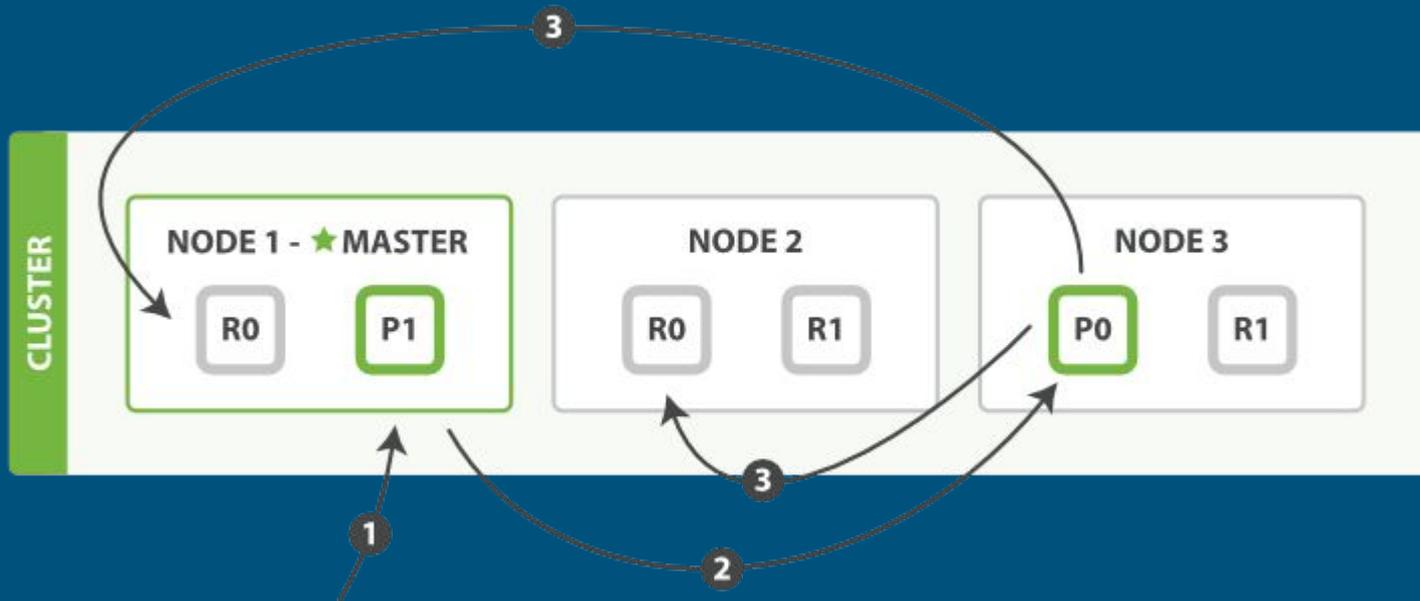


```
{
  "cluster_name": "elasticsearch",
  "status": "green",
  "timed_out": false,
  "number_of_nodes": 2,
  "number_of_data_nodes": 2,
  "active_primary_shards": 3,
  "active_shards": 6,
  "relocating_shards": 0,
  "initializing_shards": 0,
  "unassigned_shards": 0,
  "delayed_unassigned_shards": 0,
  "number_of_pending_tasks": 0,
  "number_of_in_flight_fetch": 0,
  "task_max_waiting_in_queue_millis": 0,
  "active_shards_percent_as_number":
  100
}
```

A little example how your clusters should look like



Creating, Indexing and Deleting a Document

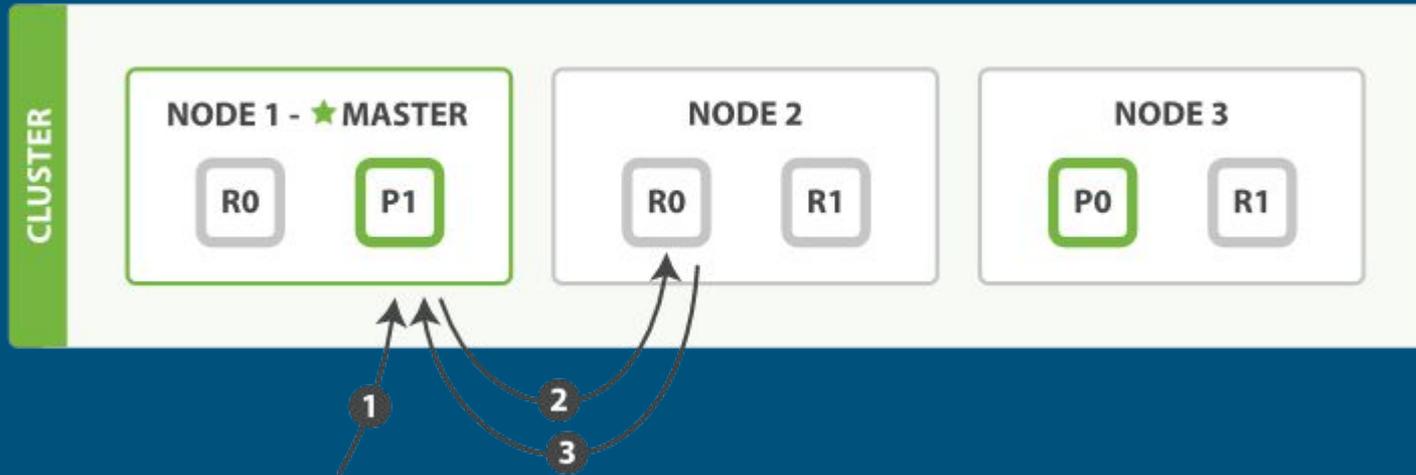


Consistency

By default, the primary shard requires a *quorum*, or majority, of shard copies (where a shard copy can be a primary or a replica shard) to be available before even attempting a write operation. This is to prevent writing data to the “wrong side” of a network partition. A quorum is defined as follows:

$$\text{int}((\text{primary} + \text{number_of_replicas}) / 2) + 1$$

Retrieving a Document





San Andres y Providencia
Colombia

Mapping

GET /gb/_mapping/tweet



```
{
  "gb": {
    "mappings": {
      "tweet": {
        "properties": {
          "date": {
            "type": "date",
            "format":
"strict_date_optional_time||epoch_millis"
          },
          "name": {
            "type": "string"
          },
          "tweet": {
            "type": "string"
          },
          "user_id": {
            "type": "long"
          }
        }
      }
    }
  }
}
```

Complex Core Field Types

- Multivalue Fields (must be the same datatype)
- Empty Fields (null value)
- Multilevel Objects (Inner objects)

```
{
  "followers": [
    { "age": 35, "name": "Mary White" },
    { "age": 26, "name": "Alex Jones" },
    { "age": 19, "name": "Lisa Smith" }
  ]
}
```

```
{
  "followers.age": [19, 26, 35],
  "followers.name": [alex, jones, lisa, smith, mary, white]
}
```

There are more things to understand

Mapping

Filters

Sorting and relevance

Aggregation

Thanks

Why do you think elasticsearch is useful?

Are you planning to use ES in any future projects?

Can you share your experiences in ES?

Questions ?????



Thanks again !!!

