Elasticsearch

Matúš Raček
Dominik Pilár
Jakub Havrila
Adam Vaňko

Contents

What is Elasticsearch?

- ➔ Apache Lucene-based search server
- ➔ open source full-text search and analytics engine
- ➔ accessible from RESTful web service
- ➔ built on Java programming language
- ➔ capable of scaling to hundreds of servers and petabytes of structured and unstructured data

Why Elasticsearch ?

- ➔ easy to scale
- ➔ everything in one JSON call away
- ➔ unleashed power of Lucene under the hood
- ➔ excellent QueryDSL
- ➔ configurable and extensible

Why Elasticsearch ? II

- ➔ document oriented
- ➔ schema free
- ➔ conflict management
- ➔ active community
- ➔ used by many big organizations like Wikipedia, The Guardian, StackOverflow, GitHub etc

Key Concepts

Node

- single running instance of Elasticsearch
- single physical or virtual server

Cluster

- collection of one or more nodes
- cluster - collective indexing and search capabilities across all the nodes for entire data

Index

- collection of different types of documents

Key Concepts II

Document

- collection of fields in a specific manner defined in JSON format
- belongs to a type and resides inside an index
- associated with a unique identifier called the UID

Shard

- indexes are horizontally subdivided into shards
- simply define the number of shards when creating index

Key Concepts III

Replicas

- increasing the availability
- improves the performance of searching => parallel search operations in these replicas
- shards

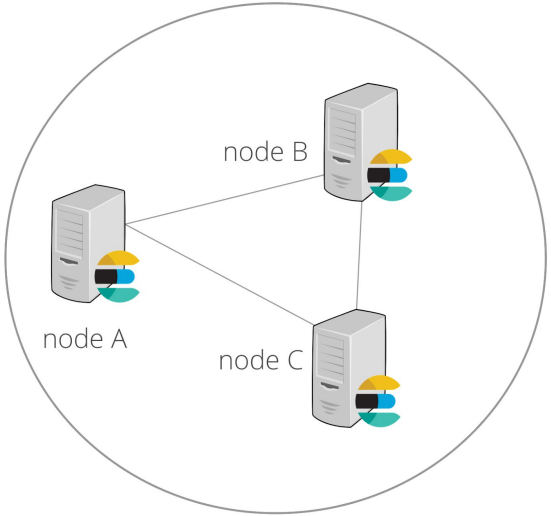
Comparison with RDBMS

Elasticsearch	RDBMS
Cluster	Cluster
Index	Database
Type	Table
Document	Row
Field	Column

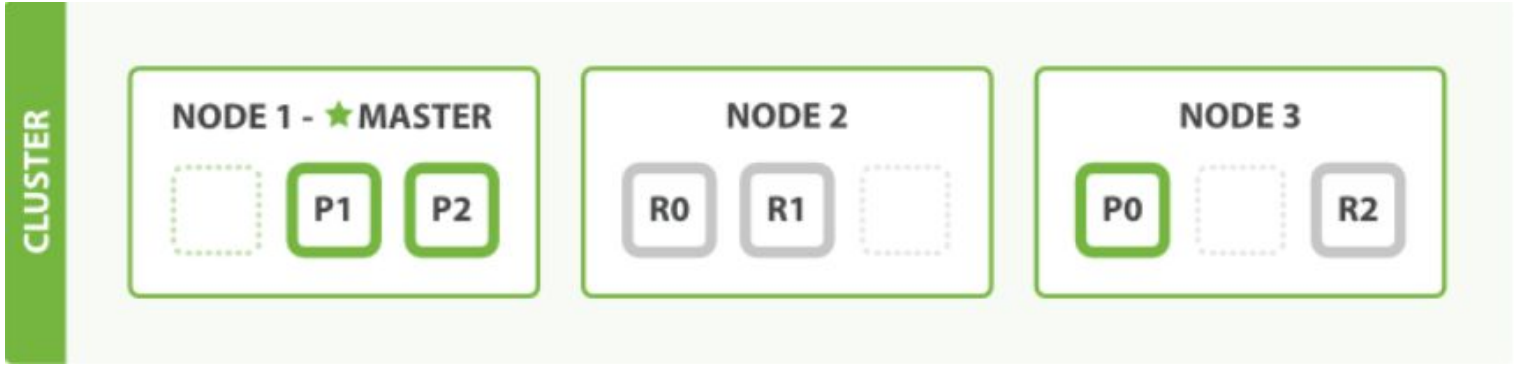
Features

Easy to scale

- ➔ horizontal scalability
- ➔ more nodes in cluster
- ➔ concept of replicas



Elasticsearch Cluster



RESTful API

➔ Elasticsearch is API driven
➔ uses JSON, which is both machine and human readable

APIs:

- Cluster API
- Index APIs
- Search APIs
- Document APIs
- ...

Java API Client

```
Client client = new PreBuiltTransportClient(
  Settings.builder().put("client.transport.sniff", true)
                    .put("cluster.name","elasticsearch").build())
  .addTransportAddress(new InetSocketTransportAddress(InetAddress.getByName("127.0.0.1"), 9300));
```

- Java API
- Python API
- JavaScript API
- Perl API
- Ruby API
- Go API
- .NET API
- PHP API

```
public void givenJsonString_whenJavaObject_thenIndexDocument() {
    String jsonObject = "{\"age\":10,\"dateOfBirth\":1471466076564,"
      +"\"fullName\":\"John Doe\"}";
    IndexResponse response = client.prepareIndex("people", "Doe")
      .setSource(jsonObject, XContentType.JSON).get();

    String id = response.getId();
    String index = response.getIndex();
    String type = response.getType();
    long version = response.getVersion();

    assertEquals(Result.CREATED, response.getResult());
    assertEquals(0, version);
    assertEquals("people", index);
    assertEquals("Doe", type);
}
```

Schema free

- ➔ documents can be indexed without explicitly providing a schema
- ➔ generate dynamically during indexing

```
{
    "name": {
        "first": "John"
    }
}
```

```
{
  "my_type" : {
    "properties" : {
      "name" : {
        "properties" : {
          "first" : {
            "type" : "string"
          }
        }
      }
    }
  }
}
```

Excellent Query DSL

- ➔ REST API exposes a very complex and capable query DSL
- ➔ filtered queries helps leverage caching and thus speed up common queries
- ➔ filtering by exact values vs searching on analyzed text

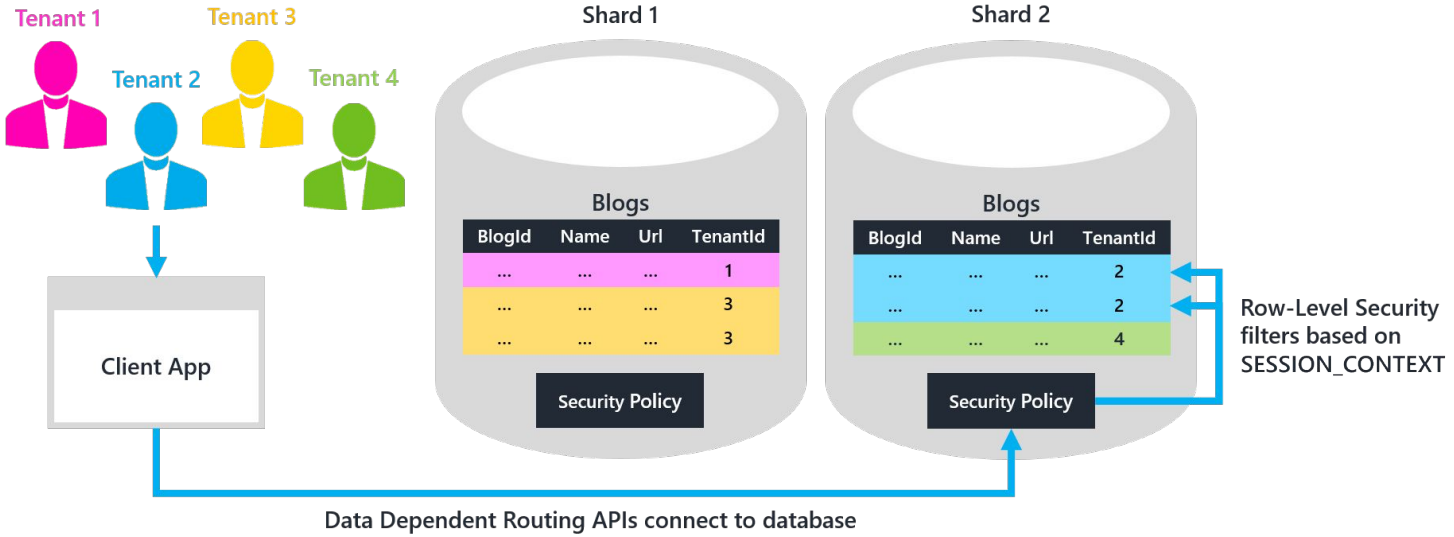
```
{
  QUERY_CLAUSE: {
    ARGUMENT: VALUE,
    ARGUMENT: VALUE,...
  }
}


{
  QUERY_CLAUSE: {
    FIELD_NAME: {
      ARGUMENT: VALUE,
      ARGUMENT: VALUE,...
    }
  }
}
```

```
{
  "range": {
    "born" : {
        "gte": "01/01/2012",
        "lte": "2013",
        "format": "dd/MM/yyyy||yyyy"
    }
  }
}
```

```
{
    "query":{
        "match" : {
            "rating":"4.5"
        }
    }
}
```

Multi-tenancy

- ➔ you can host multiple indexes on one ES installation, node or cluster. Each index can have multiple “types”.
- ➔ filtered views of an index
- ➔ indices can be queried independently or as a group



Conflict management

- ➔ optimistic version control can be used where needed to ensure that data is never lost due to conflicting changes from multiple processes

Per-operation persistence

- ➔ data safety first
- ➔ data changes are recorded in logs to minimize the chance of any loss

Active community

- ➔ community is very helpful and supporting
- ➔ books currently written by community members and blogs posts around the net sharing experiences and knowledge

Advantages

- ➔ Lots of search options
- ➔ Document-oriented
- ➔ Speed
- ➔ Scalability
- ➔ RESTful API

Disadvantages

- ➔ Does not have multi-language support in terms of handling request and response data (only JSON)
- ➔ Has a problem of Split brain situations - in rare cases
- ➔ Streaming TB’s of data every day, you will find that it either chokes or loses data

Installation and running

Installation

- ➔ Hosted on Elastic Cloud
  - ◆ GCP and AWS
  - ◆ 30 day free trial version
- ➔ Install Elasticsearch yourself
  - ◆ many possibilities (tar.gz, zip, deb, msi, rpm, brew)
  - ◆ Docker
- ➔ Our is running on OpenStack

The elastic stack

- Elasticsearch
  - Elasticsearch is a distributed, JSON-based search and analytics engine.
- Kibana
  - Kibana is the window into the Elastic Stack. Explore your data and manage the stack.
- Beats
  - Beats is a platform for lightweight shippers that send data from edge machines.
- Logstash
  - Logstash is a dynamic data collection pipeline with an extensible plugin ecosystem.

Data loading (json)

- Elasticsearch index API
  - Adds a JSON document to the specified index and makes it searchable. If the document already exists, updates the document and increments its version.
  - curl -X POST "localhost:9200/twitter/_doc/?pretty" -H 'Content-Type: application/json' -d' { "user" : "kimchy", "post_date" : "2009-11-15T14:12:12", "message" : "trying out Elasticsearch" } '

- Use Elasticsearch bulk API for big data sets
  - curl -H 'Content-Type: application/x-ndjson' -XPOST 'localhost:9200/shakespeare/_bulk?pretty' --data-binary @shakespeare.json

Logstash (other formats)

- ➔ open source, created by Elastic group
- ➔ tool which convert data into json and send to Elasticsearch
- ➔ server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to your favorite "stash."
- ➔ Logstash example https://gist.github.com/RacekM/54f3e3371210c426b52337c84a951ec9#file-logstash-example

Demonstration

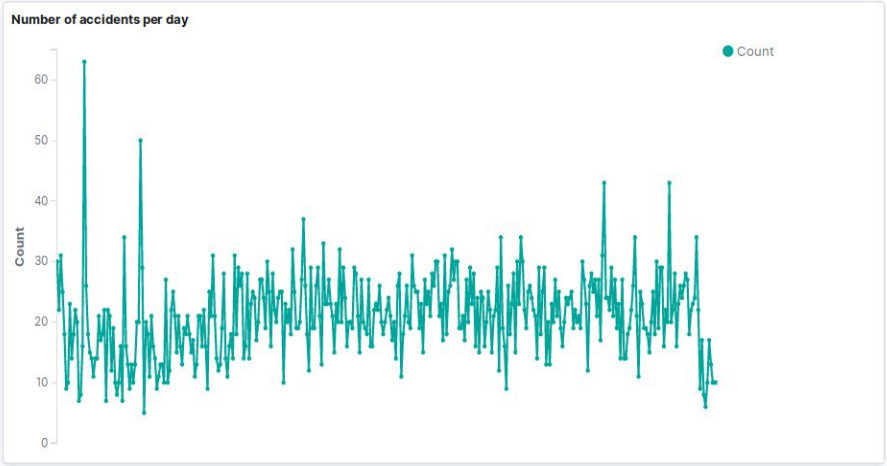
Environment and data description

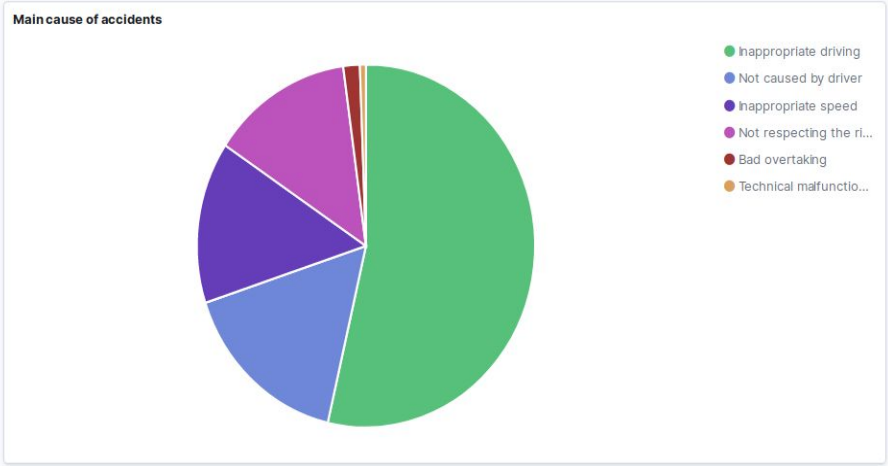
- ➔ Running and installed in a VM running on OpenStack
- ➔ Elasticsearch use port 9300 and 9200 for HTTP interface
- ➔ Kibana interface uses 5601
- ➔ Data downloaded from DATA.BRNO
- ➔ Dataset description:
  - ◆ Car accidents in 2018

Live demo queries

- **Search document -** GET data-brno/_search {"query": {"match": { Alcohol": "No" }}}
- **Delete document -** DELETE data-brno/_doc/<_id>
- **Create document -** PUT data-brno/_doc/<_id> {"Day": "Thursday"}
- **Update document -** POST data-brno/_update/123 {"doc" : {"Day" : "Friday"}}
  KQL/Lucene - “title”: “text”

Kibana visualisations





Happy New Year and Merry Christmas