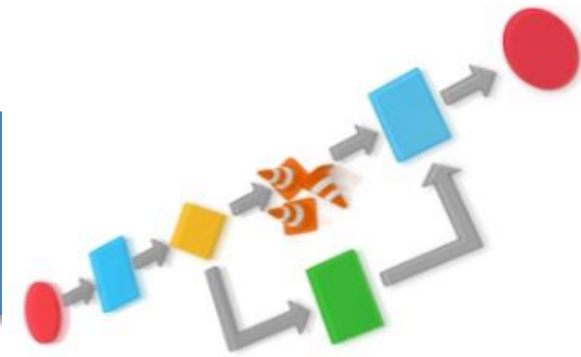


# Elasticsearch

Jiří Šebek

*b6b36nss*



```
public final void onSensorChanged(SensorEvent event)
{
    m_flightIntensity = event.values[0];
    m_etAmblight.setText("" + m_flightIntensity + " lx");
}

... resume()
... light, ... NORMAL);
```

# Obsah

- Definice
- Instalace
- Implementace a pouziti v projektu

# Obsah

- Elasticsearch = fulltextového vyhledávače vycházejícího z Apache Lucene.
- Disponuje RESTful rozhraním a nabízí vysokou dostupnost, rychlost a škálovatelnost.
- Java based technology
- Vhodny pro **Big data**
- Vhodny na naslednou analyzu dat
- Podobny system jako napriklad **Hadoop**

# Vlastnosti

- Data v reálném čase
- Škálovatelnost
- Vysoká dostupnost
- Fulltextové vyhledávání
- RESTful API

# Způsob uložení dat

- Elasticsearch je bezschémovou databází.
- Není proto třeba definovat strukturu databáze,
- protože ta se vytvoří sama na základě vložených dat.

# Elasticsearch vs Relational Database

Relational DB	Elasticsearch
DataBase	Index
Table	Type
Row/Record	Document
Column Name	Field

# Elasticsearch vs Relational Database

Kdy je lepší použít elasticsearch než relační db

- Relevance search

- Analýza logů (elastic + kibana 😊)**

- Hledání synonym

- Fulltext search

- Hledání slov, která jsou špatně napsána (hledání na základě patternů)

- .....

# Relevance based search

```
{
  "_index": "test",
  "_type": "product",
  "_id": "AV0iKK_ZJJfvpLB9dSHI",
  "_score": 0.51623213,      =====> Relevance Score calculated by ES
  "_source": {
    "id": 2,
    "name": "Red Shirt"
  }
}
```



# Hledání slov, která jsou špatně napsána (hledání na základě patternů)

## Query

```
• {  
  "query": {  
    "match": {  
      "name": {  
        "query": "shrt",  
        "fuzziness": 2,  
        "prefix_length": 0  
      }  
    }  
  }  
}
```

## Result

```
{  
  "_index": "test",  
  "_type": "product",  
  "_id": "AV0iKKplJJfvpLB9dSHk",  
  "_score": 0.21576157,  
  "_source": {  
    "id": 1,  
    "name": "Shirt"  
  }  
}
```

Source  
Text

The two <em>lazy</em> dogs were slower than the less  
lazy <em>dog</em>, Rover.

html\_strip  
Char Filter

The two lazy dogs were slower than the less lazy dog,  
Rover.

standard  
Tokenizer

The<sup>1</sup> two<sup>2</sup> lazy<sup>3</sup> dogs<sup>4</sup> than<sup>5</sup> the<sup>6</sup>  
less<sup>7</sup> lazy<sup>8</sup> dog<sup>9</sup> Rover<sup>1</sup>

lowercase  
Token Filter

the<sup>1</sup> two<sup>2</sup> lazy<sup>3</sup> dogs<sup>4</sup> than<sup>5</sup> the<sup>6</sup>  
less<sup>7</sup> lazy<sup>8</sup> dog<sup>9</sup> rover<sup>10</sup>

stop  
Token Filter

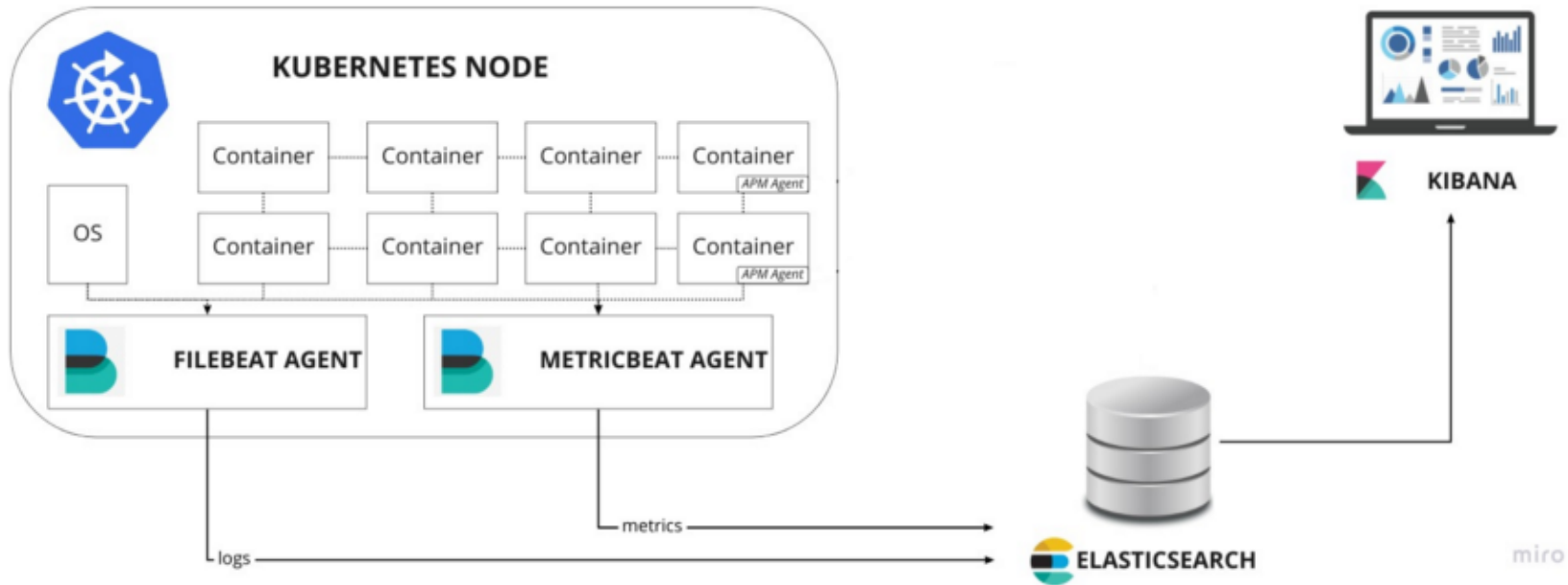
two<sup>2</sup> lazy<sup>3</sup> dogs<sup>4</sup> than<sup>5</sup>  
less<sup>7</sup> lazy<sup>8</sup> dog<sup>9</sup> rover<sup>10</sup>

# ELK

•Elastic Stack (ELK) = elastic + logstash + kibana



# ELK



# ELK

 D Home



## Add Data to Kibana

Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.



### APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

[Add APM](#)



### Logging

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

[Add log data](#)



### Metrics

Collect metrics from the operating system and services running on your servers.

[Add metric data](#)



### SIEM

Centralize security events for interactive investigation in ready-to-go visualizations.

[Add security events](#)

### Add sample data

[Load a data set and a Kibana dashboard](#)

### Upload data from log file

[Import a CSV, NDJSON, or log file](#)

### Use Elasticsearch data

[Connect to your Elasticsearch index](#)

## Visualize and Explore Data



### APM

Automatically collect in-depth performance metrics and errors from inside your applications.



### Canvas

Showcase your data in a pixel-perfect way.



### Dashboard

Display and share a collection of visualizations



### Discover

Interactively explore your data by querying and

## Manage and Administer the Elastic Stack



### Console

Skip cURL and use this JSON interface to work with your data directly.



### Index Patterns

Manage the index patterns that help retrieve your data from Elasticsearch.



### Monitoring

Track the real-time health and performance of your



### Rollups

Summarize and store historical data in a smaller

# Příklady

## .# Vytvoření indexu

```
$ curl -XPUT 'http://localhost:9200/twitter/'
```

## .# Přidání dokumentu

```
$ curl -XPUT 'http://localhost:9200/twitter/tweet/1' -d '{  
  "tweet" : {  
    "user" : "kimchy",  
    "post_date" : "2009-11-15T14:12:12",  
    "message" : "trying out Elastic Search"  
  }  
}'
```

## .# Získání dokumentu dle jeho id

```
$ curl -XGET 'http://localhost:9200/twitter/tweet/1'
```

## .# Vyhledání dokumentu

```
$ curl -XGET 'http://localhost:9200/twitter/tweet/_search?q=user:kimchy'
```

# Instalace/nastaveni

- stahnout z <https://www.elastic.co/downloads/past-releases>
- zvolit Elasticsearch v drop down a vybrat spravnou verzi (5.5.0 pro jdk8) a kliknout na download
- Na dalsi strance stahnout jako Zip pro windows. Tar pro linux
- Otevrit Zip a rozbalit obsah. Ve slozce bin najdete elasticsearch.bat. Spuste. *Spoustejte bat soubor pres cmd ! Uvidite chybove hlasky.*
- POZOR : pokud se objevi hlaska typu :

***elasticsearch could not reserve enough space for object heap***

•\config\jvm.options

•-Xms512m

-Xmx512m

# Instalace/nastaveni

- Otevře v prohlizeci: `http://localhost:9200/`.
- Port 9300 je také povolen jak cluster node
- Port 9200 je pro REST komunikaci
- Otestujte pomocí curl příkazu z command line :
- `curl http://localhost:9200/`



# Test

•V insomnii/postmanovi si vytvorte request :

•<http://localhost:9200/users/employee/1>

–POST metoda

–JSON body :

```
{
  "userId" : "1",
  "name" : "Rajesh",
  "userSettings" : {
    "gender" : "male",
    "occupation" : "CA",
    "hobby" : "chess"
  }
}
```

# Test

- V insomnii/postmanovi si vytvorte request :
- <http://localhost:9200/users/employee/1>
- GET metoda

# Test

- Zkusme search :
- curl -XGET '[http://localhost:9200/users/employee/\\_search](http://localhost:9200/users/employee/_search)'
- Pozor windows nema rad ' '.. proto nahradme za « »
- curl -XGET « http://localhost:9200/users/employee/\_search »

# Test

- Zkusme search s query:
- `curl -XGET "http://localhost:9200/users/employee/_search" -H "Content-Type: application/json" -d "{}"`
- `curl -XGET 'http://localhost:9200/users/employee/_search'`
- `-H 'Content-Type: application/json' -d`
- `'{"query": { "match": { "name" : "Rajesh" } } }'`

# Springboot

## Pom.xml

```
<dependency>
```

```
<groupId>org.elasticsearch</groupId>
```

```
<artifactId>elasticsearch</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.elasticsearch.client</groupId>
```

```
<artifactId>transport</artifactId>
```

```
<version>5.0.0</version>
```

```
</dependency>
```

# Springboot

## •application.properties

# Local Elasticsearch config

elasticsearch.host=localhost

elasticsearch.port=9200

# App config

server.port=8102

spring.application.name=BootElastic

# Springboot

## •Domain object

```
•public class User {  
  
• private String userId;  
  
• private String name;  
  
• private Date creationDate = new Date();  
  
• private Map<String, String> userSettings = new HashMap<>();  
  
• -- getter/setter methods  
  
•//nebo pouzijeme Lombok  
  
•}
```

# Springboot

## .config

@Configuration

```
public class config{
    @Value("${elasticsearch.host:localhost}")
    public String host;
    @Value("${elasticsearch.port:9300}")
    public int port;
    public String getHost() {
return host;
}
public int getPort() {
return port;
}
}
```

@Bean

```
public Client client(){
    TransportClient client = null;
    try{
        System.out.println("host:"+ host+"port:"+port);
        client = new PreBuiltTransportClient(Settings.EMPTY)
            .addTransportAddress(new InetSocketAddress(InetAddress.getByName(host), port));
    } catch (UnknownHostException e) {
        e.printStackTrace();
    }
    return client;
}
}
```



# Springboot

## .Controller

### •@Autowired

- Client client;
- @PostMapping("/create")
- public String create(@RequestBody User user) throws IOException {
- IndexResponse response = client.prepareIndex("users", "employee", user.getUserId())
- .setSource(jsonBuilder().startObject().field("name", user.getName())
- .field("userSettings", user.getUserSettings()).endObject()
- ).get();
- System.out.println("response id:"+response.getId());
- return response.getResult().toString();
- }

# Springboot

## .Controller

- `@GetMapping("/view/{id}")`
- `public Map<String, Object> view(@PathVariable final String id) {`
- `GetResponse getResponse = client.prepareGet("users", "employee", id).get();`
- `return getResponse.getSource();`
- `}`

# Springboot

## .Controller

- `@GetMapping("/view/name/{field}")`
- `public Map<String, Object> searchByName(@PathVariable final String field) {`
- `Map<String, Object> map = null;`
- `SearchResponse response = client.prepareSearch("users").setTypes("employee")`  
`.setSearchType(SearchType.QUERY_AND_FETCH).setQuery(QueryBuilders.matchQuery("name",`  
`field)).get();`
- `List<SearchHit> searchHits = Arrays.asList(response.getHits().getHits());`
- `map = searchHits.get(0).getSource();`
- `return map;`
- `}`

# Springboot

## .Controller

- `@GetMapping("/update/{id}")`
- `public String update(@PathVariable final String id) throws IOException {`
- `UpdateRequest updateRequest = new UpdateRequest();`
- `updateRequest.index("users").type("employee").id(id)`
- `.doc(jsonBuilder().startObject().field("name", "Rajesh").endObject());`
- `try {`
- `UpdateResponse updateResponse = client.update(updateRequest).get();`
- `System.out.println(updateResponse.status());`
- `return updateResponse.status().toString();`
- `} catch (InterruptedException | ExecutionException e) { System.out.println(e); }`
- `return "Exception";`

# Springboot

•Otestujte :

•<http://localhost:8102/rest/users/create> ... POST

–JSON body :

```
{  
  "userId":"1",  
  "name": "Sumit",  
  "userSettings": {  
    "gender" : "male",  
    "occupation" : "CA",  
    "hobby" : "chess"  
  }  
}
```

# Springboot

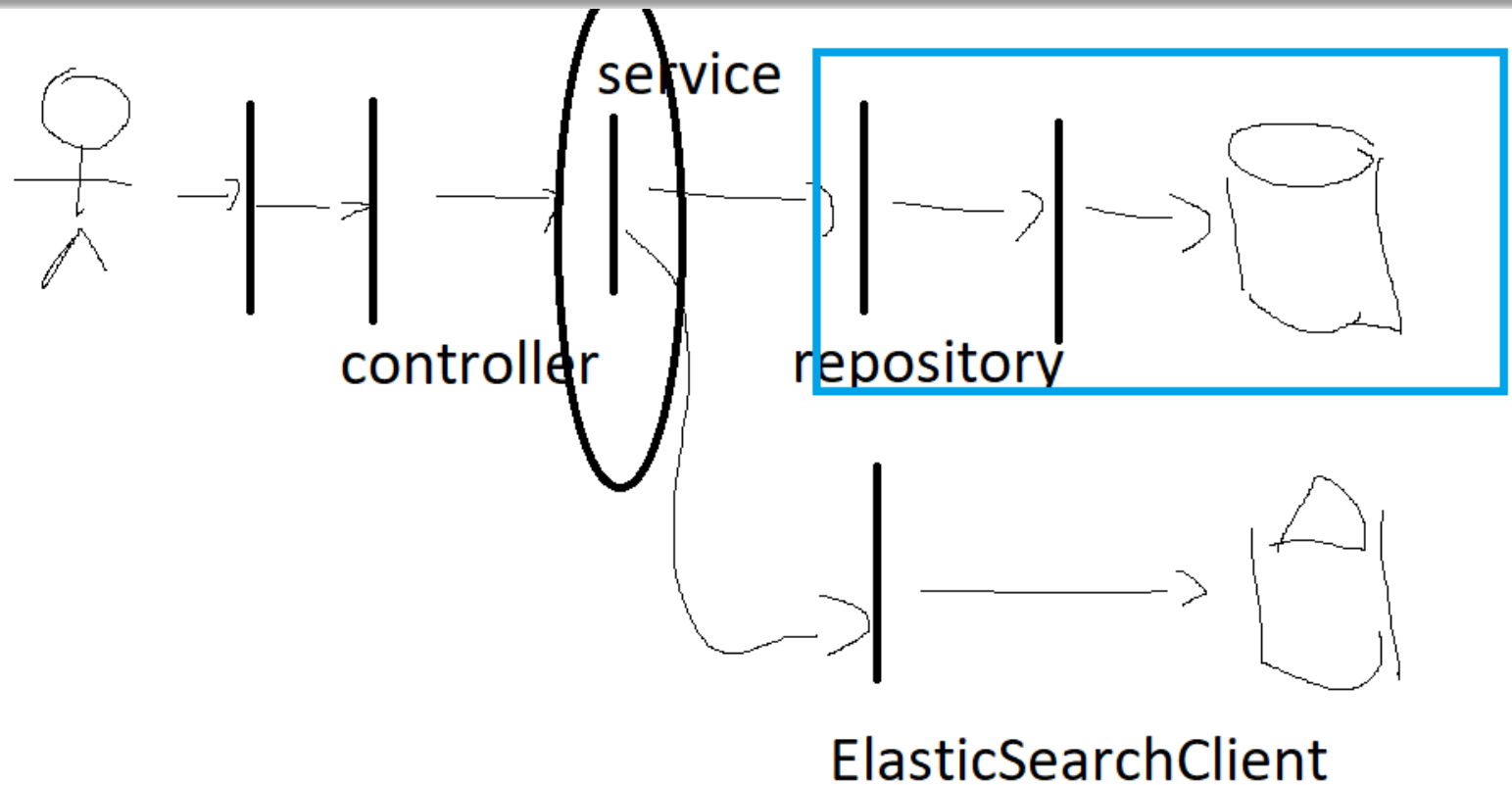
•Otestujte :

•<http://localhost:8102/rest/users/view/1> ... GET

•Otestujte :

•Podobne delete i update

# Springboot kam spravne dat implementaci?



# Kibana

.Demo :

<https://www.elastic.co/demos>

.Overview.. Prometheus

**.Logy z namespaces**

.Filtry

•..