elasticsearch

workshop

elasticsearch - the company

- Founded in 2012

  by the people behind elasticsearch project

- Professional services

  Training (public and on site)

  Development support

  Production support subscription

- Commercial product

  Marvel (included with support)

Agenda

- Elasticsearch overview
- Workshop 0: getting started
- Workshop 1: let's index some documents
- Workshop 2: let's search them
- Workshop 3: let's pull some analytics
- Workshop 4: let's add a powerful live UI on top
- Workshop 5: snapshot and restore

fundamentals

a search engine

- Create indices from documents
- Search in indices

elasticsearch

- Cloud based search engine
- Based on Lucene
- Hide Lucene complexity by exposing all services

  HTTP / REST / JSON
- Works with all technologies
- Horizontal scaling, replication, fail over, load balancing
- Blazing fast!
- It’s a search engine! Not a search tool in a box!

think document!

- Change your mindset:

  Forget SQL!

  Index what you want to find

- A document

  A JSON object

  Core field types (string, numbers, booleans)

  Complex field types (arrays, objects)

  Additional field types (geo points, geo shapes)

organize your documents!

- Documents coordinates:

  index (hold setup)

  type (holds mapping)

  id (can be auto-generated)

```
{
    "name" : "elasticsearch",
    "website" : "http://www.elasticsearch.com",
    "category" : "software",
    "founded_year" : 2012,
    "overview" : "The company behind the elasticsearch open source project",
    "tags" : ["search", “datastore", "analytics", "realtime", "scalability"],
    "location" : {
        "city" : "Amsterdam",
        "country_code" : "NL",
        "geo" : {
            "lat" : 52.370176,
            "lon" : 4.895008
        }
    }
}
```

glossary

- Node

  a running elasticsearch instance (JVM process)

- Cluster

  a group of nodes

- Shard

  a part of an index

  a Lucene index under the hood

  primary: unique in the cluster

  replica: one or more copy of the primary

workshop 0

setup

setup (manually)

- get elasticsearch x.x.x

```
curl -OL -k download.elasticsearch.org/elasticsearch/elasticsearch/
elasticsearch-x.x.x.zip
```

- edit config/elasticsearch.yml

```
cluster.name: workshop
discovery.zen.ping.multicast.enabled: false
discovery.zen.ping.unicast.hosts: ["127.0.0.1"]
```

- install marvel plugin

```
bin/plugin -install elasticsearch/marvel/latest
```

setup (easier)

```
bin/plugin -install marvel -url file:../binaries/marvel-latest.zip
```

play with nodes

- start an elasticsearch node

```
bin/elasticsearch
```

- open marvel

```
open http://localhost:9200/_plugin/marvel/
```

workshop 1

we index persons

indexing a document

```
POST /person/person
{
   "name":"Anaelle Alessio",
   "dateOfBirth":"2009-09-05",
   "gender":"female",
   "marketing":{
      "shoes":1000,
      "fashion":1200,
      "music":800
   },
   "address":{
      "country":"England",
      "zipcode":"5226",
      "city":"Plymouth",
      "countrycode":"GB"
   }
}
```

```
{
   "_index": "person",
   "_type": "person",
   "_id": "zvb2udm2TSa8Zyp9LnD1nA",
   "_version": 1,
   "created": true
}
```

getting a document

```
GET /person/person/zvb2udm2TSa8Zyp9LnD1nA
```

```
{
   "_index": "person",
   "_type": "person",
   "_id": "zvb2udm2TSa8Zyp9LnD1nA",
   "_version": 1,
   "found": true,
   "_source": {
      "name": "Anaelle Alessio",
      "dateOfBirth": "2009-09-05",
      "gender": "female",
      "marketing": {
         "shoes": 1000,
         "fashion": 1200,
         "music": 800
      },
      "address": {
         "country": "England",
         "zipcode": "5226",
         "city": "Plymouth",
         "countrycode": "GB"
      }
   }
}
```

updating a document

```
PUT /person/person/zvb2udm2TSa8Zyp9LnD1nA
{
   "name":"Anaelle Alessio",
   "dateOfBirth":"2009-09-05",
   "gender":"female",
   "marketing":{
      "shoes":1001,
      "fashion":1200,
      "music":800
   },
   "address":{
      "country":"England",
      "zipcode":"5226",
      "city":"Plymouth",
      "countrycode":"GB"
   }
}
```

```
{
   "_index": "person",
   "_type": "person",
   "_id": "zvb2udm2TSa8Zyp9LnD1nA",
   "_version": 2,
   "created": false
}
```

deleting a document

```
DELETE /person/person/zvb2udm2TSa8Zyp9LnD1nA
```

```
{
   "found": true,
   "_index": "person",
   "_type": "person",
   "_id": "1",
   "_version": 3
}
```

workshop 1: index some persons

```
PUT /person/person/1
{
   "name":"Anaelle Alessio"
}
```

```
PUT /person/person/2
{
   "name":"Joe Smith"
}
```

```
PUT /person/person/1
{
   "name":"Anaelle Alessio",
   "dateOfBirth":"2009-09-05"
}
```

```
PUT /person/person/2
{
   "name":"Joe Smith",
   "gender":"male"
}
```

workshop 1: 500 000 persons

- use injector script

```
java -jar injector-x.x.x.jar 500000 10000 workshop
```

- see effect in marvel

```
open http://localhost:9200/_plugin/marvel/
```

- start more nodes

```
bin/elasticsearch
bin/elasticsearch
bin/elasticsearch
...
```

workshop 2

we search for persons

searching persons in Germany

```
GET /person/person/_search
{
    "query": {
        "term": {
           "address.country": {
              "value": "Germany"
           }
        }
    }
}
```

```
{
  "took" : 3,
  "hits" : {
    "total" : 0,
    "max_score" : null,
    "hits" : [ ]
  }
}
```

searching persons in germany

```
GET /person/person/_search
{
    "query": {
        "term": {
           "address.country": {
              "value": "germany"
           }
        }
    }
}
```

```
{
  "took" : 4,
  "hits" : {
    "total" : 30004,
    "max_score" : 2.100946,
    "hits" : [ {
      "_index" : "person",
      "_type" : "person",
      "_id" : "SUy7Py3zSvqhjQroJPVFCw",
      "_score" : 2.100946,
      "_source" : {"name":"Fadi Norah", "address":{"country":"Germany"}}
      }, { ... }
      ]
  }
}
```

searching persons in Germany

```
GET /person/person/_search
{
  "query": {
    "match": {
      "address.country": "Germany"
    }
  }
}
```

```
{
  "took" : 4,
  "hits" : {
    "total" : 30004,
    "max_score" : 2.100946,
    "hits" : [ {
      "_index" : "person",
      "_type" : "person",
      "_id" : "SUy7Py3zSvqhjQroJPVFCw",
      "_score" : 2.100946,
      "_source" : {"name":"Fadi Norah", "address":{"country":"Germany"}}
      }, { ... }
      ]
  }
}
```

searching for persons

```
GET /person/person/_search
{
    "query": {
        "bool": {
            "must": [ {
                "match": {
                      "address.country": "Germany"
                   }
               }, {
                   "range": {
                      "dateOfBirth": {
                         "from": "1970",
                         "to": "1971"
                      }
                   }
               }
            ]
        }
    }
}
```

workshop 2: reinject with mapping

- delete old data

```
DELETE /person
```

- use injector script

```
java -jar injector-x.x.x.jar 500000 10000 workshop
```

- get mapping

```
GET /person/person/_mapping
```

workshop 2: search again

```
GET /person/person/_search
{
  "query": {
    "term": {
      "address.country": "Germany"
    }
  }
}
```

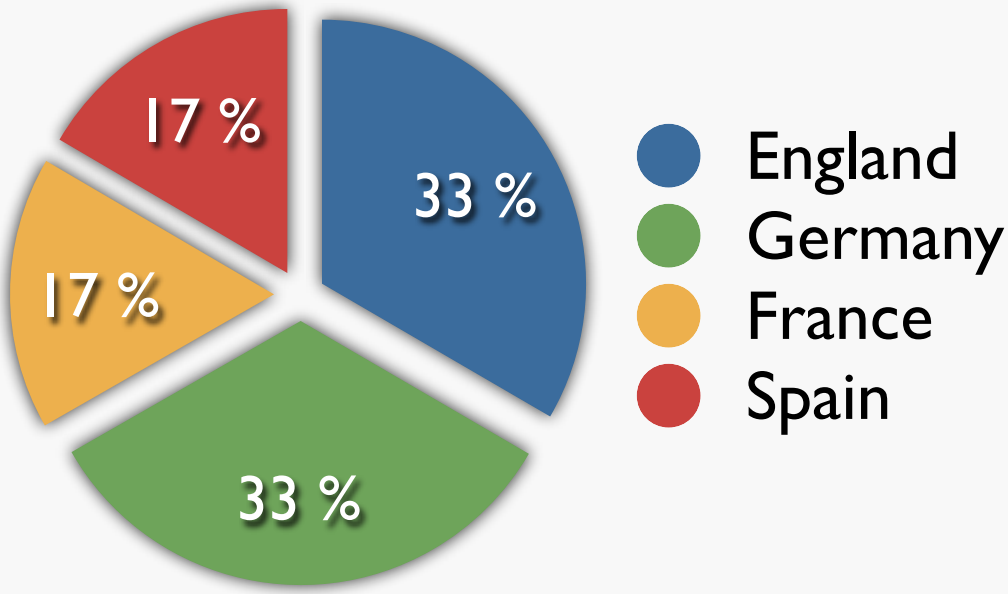
```
{
  "took" : 4,
  "hits" : {
    "total" : 30004,
    "max_score" : 2.100946,
    "hits" : [ {
      "_index" : "person",
      "_type" : "person",
      "_id" : "SUy7Py3zSvqhjQroJPVFCw",
      "_score" : 2.100946,
      "_source" : {"name":"Fadi Norah", "address":{"country":"Germany"}}
      }, { ... }
      ]
  }
}
```

workshop 3

make sense of your data: aggs!

break by country

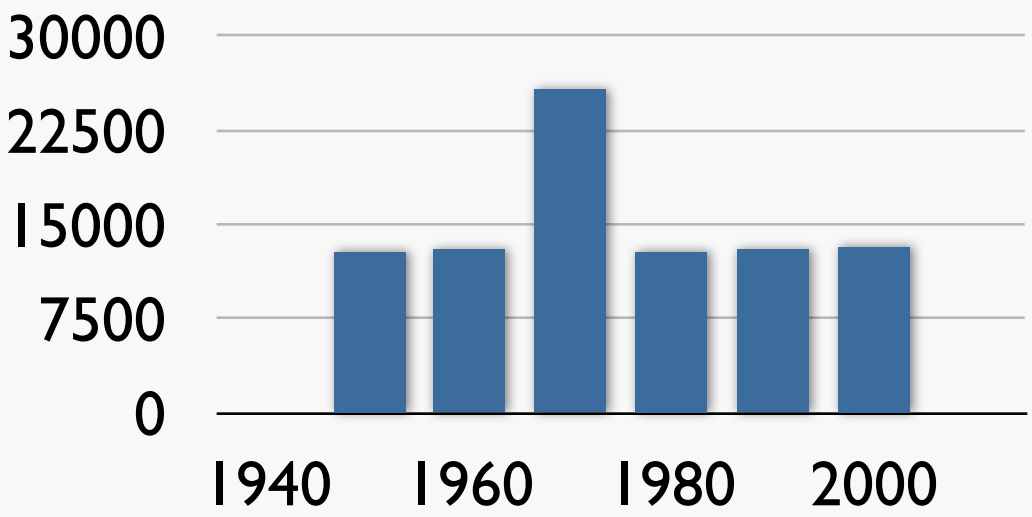
```
GET /person/person/_search?search_type=count
{
    "aggs": {
       "by_country": {
          "terms": {
             "field": "address.country"
          }
       }
    }
}
```




```
{ ..., "aggregations" : {
    "by_country" : {
     "buckets" : [ {
        "key" : "England",
        "doc_count" : 30051
      }, {
        "key" : "Germany",
        "doc_count" : 30004
      }, {
        "key" : "France",
        "doc_count" : 15034
      }, {
        "key" : "Spain",
        "doc_count" : 14912
      } ]}}}
```

date of birth histogram

```
GET /person/person/_search?search_type=count
{
    "aggs": {
       "by_date": {
          "date_histogram": {
             "field": "dateOfBirth",
             "interval": "year",
             "format": "yyyy"
          }
       }
    }
}
```




```
{ ..., "aggregations": {
      "by_date": {
        "buckets": [
            {
               "key_as_string": "1960",
               "key": -946080000000,
               "doc_count": 39
            },
            {
               "key_as_string": "1961",
               "key": -630720000000,
               "doc_count": 12677
            },
            {
               "key_as_string": "1962",
               "key": -315360000000,
               "doc_count": 12936
            }, ...
        ]
      }
   }}
```

searching for persons with aggs

```
GET /person/person/_search?search_type=count
{
  "query": { "bool": { "must": [
      { "match": { "address.country": "Germany"} },
      { "range": { "dateOfBirth": { "from": "1970", "to": "1971" }}}
  ]}},
  "aggs": {
    "by_date": {
      "date_histogram": {
        "field": "dateOfBirth", "interval": "month", "format": "yyyy-MM"
      },
      "aggs": {
        "by_gender": {
          "terms": {
            "field": "gender"
          },
          "aggs": {
            "children": {
              "stats": {
                "field": "children"
              }
            }
          }
        }
      }
    }
  }
}
```

workshop 4

click & play!

setup

- get kibana

```
bin/plugin -install elasticsearch/kibana
# or
curl -OL -k http://download.elasticsearch.org/kibana/kibana/kibana-latest.zip
```

- open kibana

```
open http://localhost:9200/_plugin/kibana/
```

- build your dashboard as you need!

workshop 5

snapshot and restore

backup

- create repository

```
PUT /_snapshot/main_backup
{
    "type" : "fs",
    "settings" : {
        "location" : "/tmp/es-backup"
    }
}
```

- backup

```
PUT /_snapshot/main_backup/snap1?wait_for_completion=true
```

- show all backups

```
GET /_snapshot/main_backup/_all
```

restore

- create repository (if needed)

```
PUT /_snapshot/main_backup
{
    "type" : "fs",
    "settings" : {
        "location" : "/tmp/es-backup"
    }
}
```

- restore

```
POST /_snapshot/main_backup/snap1/_restore?wait_for_completion=true
{
    "indices":"+person"
}
```

```
POST /_snapshot/main_backup/snap1/_restore
{
    "indices":"+person",
    "rename_pattern": "person",
    "rename_replacement": "new_person"
}
```

thank you!

@elasticsearch

http://elasticsearch.com/support