

- How Elasticsearch & Kibana work
- Installation & setup
- Index creation, data import, queries

# Elasticsearch + Kibana: Full Documentation

## 1. Introduction to Elasticsearch and Kibana

### ♦ What is Elasticsearch?

Elasticsearch is a search and analytics engine that allows you to store, search, and analyze large volumes of data efficiently. It is a NoSQL database that stores data in JSON format and uses REST APIs for querying.

### ♦ How does Elasticsearch work?

1. Data is stored in indexes (like tables in SQL databases).
2. Each document is a JSON object (similar to rows in SQL).
3. Queries are executed via REST API (using `curl` or Kibana Dev Tools).

### ♦ What is Kibana?

Kibana is a visual interface for working with Elasticsearch.

With Kibana, you can:

- View and explore data (**Discover**)
- Build visualizations (**Visualize**)
- Perform complex queries (**Dev Tools**)

## ♦ 2. Installing Elasticsearch and Kibana

### 2.1 Installing Elasticsearch

Download and extract Elasticsearch:

```
wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.17.7-linux-x86_64.tar.gz  
tar -xzf elasticsearch-7.17.7-linux-x86_64.tar.gz
```

Start Elasticsearch:

```
cd elasticsearch-7.17.7  
./bin/elasticsearch
```

Verify if Elasticsearch is running:

```
curl -XGET "http://localhost:9200/?pretty"
```

### 2.2 Installing Kibana

Download and extract Kibana:

```
wget https://artifacts.elastic.co/downloads/kibana/kibana-7.17.7-linux-x86_64.tar.gz  
tar -xzf kibana-7.17.7-linux-x86_64.tar.gz
```

Start Kibana:

```
cd kibana-7.17.7-linux-x86_64  
./bin/kibana
```

Open Kibana in a browser:

```
http://localhost:5601
```

## ◆ 3. Working with Elasticsearch: CRUD Operations

### 3.1 Create an index (**movies2**)

Create a new index:

```
curl -XPUT "http://localhost:9200/movies2" -H "Content-Type: application/json" --data-binary @mapping.json
```

Check if the index is created:

```
curl -XGET "http://localhost:9200/_cat/indices?v"
```

Check its structure (mapping):

```
curl -XGET "http://localhost:9200/movies2/_mapping?pretty"
```

### ✓ 3.2 Adding and Deleting Documents

✚ Add a document to **movies2**:

```
curl -XPOST "http://localhost:9200/movies2/_doc/1" -H "Content-Type: application/json" -d '{  
  "title": "Inception",  
  "year": 2010,  
  "genres": ["Sci-Fi", "Action"],  
  "rating": 8.8  
'
```

Retrieve a document by ID:

```
curl -XGET "http://localhost:9200/movies2/\_doc/1?pretty"
```

Delete a document:

```
curl -XDELETE "http://localhost:9200/movies2/_doc/1"
```

### 3.3 Deleting and Recreating an Index

Delete the **movies2** index (if it exists):

```
curl -XDELETE "http://localhost:9200/movies2"
```

Recreate with correct **mapping.json**: 

```
curl -XPUT "http://localhost:9200/movies2" -H "Content-Type: application/json" --data-binary @mapping.json
```

There are some GET functions that I executed in DevTools, but I also used the Elasticsearch library, and I completed everything in all the TPs

```
GET movies2/_search
{
  "size": 0,
  "aggs": {
    "average_rating": {
      "avg": { "field": "fields.rating" }
    }
  }
}
```

```
2* {
  "took": 6,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 4849,
      "relation": "eq"
    },
    "max_score": null,
    "hits": [ ]
  },
  "aggregations": {
    "average_rating": {
      "value": 6.387107691895831
    }
  }
}
```

History Settings Help

```
125 GET movies2/_search
126 {
127   "size": 0,
128   "query": {
129     "match": {
130       "fields.directors": "George Lucas"
131     }
132   },
133   "aggs": {
134     "average_rating": {
135       "avg": { "field": "fields.rating" }
136     },
137     "average_rank": {
138       "avg": { "field": "fields.rank" }
139     }
140   }
141 }
142 }
```

```
2* {
  "took": 12,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 57,
      "relation": "eq"
    },
    "max_score": null,
    "hits": [ ]
  },
  "aggregations": {
    "average_rank": {
      "value": 2580.9824561403507
    },
    "average_rating": {
      "value": 6.916666631345996
    }
  }
}
```

```
1 #! Elasticsearch built-in security features are not enabled. Without authentication, your cluster could be accessible to anyone. See https
2 //www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable security.
3 health status index uuid pri rep docs.count docs.deleted store.size pri.store.size
4 green open .geoip_databases pKvX873ySCuq14ne6yi-Uw 1 0 39 0 36.7mb 36.7mb
5 yellow open movies 90G7Ga48RGuIQoJO_wokw 1 1 9550 0 9.1mb 9.1mb
6 green open .apm-custom-link QeXE1XqCRNuW6F3he1C7tw 1 0 0 0 226b 226b
7 green open .kibana_task_manager_7.17.7_001 6kh9Kw52TOG8rbkzCvs86w 1 0 17 61 349.2kb 349.2kb
8 green open .apm-agent-configuration lOurqh-hTS2Djpfz4Qc0JA 1 0 0 0 226b 226b
9 green open .async-search cVLEnfi8T1Gi9XIRel8fVug 1 0 0 0 3.5kb 3.5kb
10 green open .kibana_7.17.7_001 dr42Nm3NQjmETGIw5f_MoQ 1 0 447 2 4.9mb 4.9mb
11 yellow open movies2 RvTmuo73TnuqeAu9sfDnSw 1 1 4849 0 3.4mb 3.4mb
12 green open .tasks RR_rjgAlTTWtryMCT_5x1Q 1 0 17 1 71.2kb 71.2kb
```

Dev Tools

ConsoleSearch ProfilerGrok DebuggerPainless LabBETA

HistorySettingsHelp

```
143 GET movies2/_search
144 {
145   "size": 0,
146   "aggs": {
147     "movies_per_year": {
148       "terms": {
149         "field": "fields.year",
150         "size": 10
151       }
152     }
153   }
154 }
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
```

```
17   "hits": [ ]
18 }
19
20 "aggregations": {
21   "movies_per_year": {
22     "doc_count_error_upper_bound": 0,
23     "sum_other_doc_count": 2192,
24     "buckets": [
25       {
26         "key": 2013,
27         "doc_count": 448
28       },
29       {
30         "key": 2012,
31         "doc_count": 404
32       },
33       {
34         "key": 2011,
35         "doc_count": 308
36       },
37       {
38         "key": 2009,
39         "doc_count": 253
40       },
41       {
42         "key": 2010,
43         "doc_count": 249
44       },
45       {
46         "key": 2008,
47         "doc_count": 207
48       },
49       {
50         "key": 2006,
51         "doc_count": 204
52       },
53       {
54         "key": 2007,
55         "doc_count": 200
56       },
57       {
58         "key": 2005,
59         "doc_count": 170
60       },
61       {
62         "key": 2014,
63         "doc_count": 152
64       }
65     ]
66   }
67 }
```

```
GET movies2/_search
{
  "size": 0,
  "aggs": {
    "average_rating": {
      "avg": {"field": "fields.rating"}
    }
  }
}
```

```
1 #! Elasticsearch built-in security features at
2 //www.elastic.co/guide/en/elasticsearch/ref
3
4 {
5   "took": 12,
6   "timed_out": false,
7   "shards": {
8     "total": 1,
9     "successful": 1,
10    "skipped": 0,
11    "failed": 0
12  },
13  "hits": {
14    "total": {
15      "value": 4849,
16      "relation": "eq"
17    },
18    "max_score": null,
19    "hits": [ ]
20  },
21  "aggregations": {
22    "average_rating": {
23      "value": 6.387107691895831
24    }
25  }
26 }
```

```
170 GET movies2/_search
171 {
172   "size": 0,
173   "query": {
174     "match": {"fields.directors": "George Lucas"}
175   },
176   "aggs": {
177     "average_rating": {"avg": {"field": "fields.rating"}},
178     "average_rank": {"avg": {"field": "fields.rank"}}
179   }
180 }
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
```

```
1 #! Elasticsearch built-in security feature
2 ://www.elastic.co/guide/en/elasticsearch
3
4 {
5   "took": 5,
6   "timed_out": false,
7   "shards": {
8     "total": 1,
9     "successful": 1,
10    "skipped": 0,
11    "failed": 0
12  },
13  "hits": {
14    "total": {
15      "value": 57,
16      "relation": "eq"
17    },
18    "max_score": null,
19    "hits": [ ]
20  },
21  "aggregations": {
22    "average_rank": {
23      "value": 2580.9824561403507
24    },
25    "average_rating": {
26      "value": 6.916666631345396
27    }
28  }
29 }
```

Click to send request

```
181 GET movies2/_search
182 {
183   "size": 0,
184   "aggs": {
185     "by_year": {
186       "terms": {"field": "fields.year", "size": 10},
187       "aggs": {
188         "average_rating": {"avg": {"field": "fields.rating"}}
189       }
190     }
191   }
192 }
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
```

```
13 "aggregations": {
14   "by_year": {
15     "doc_count_error_upper_bound": 0,
16     "sum_other_doc_count": 2192,
17     "buckets": [
18       {
19         "key": 2013,
20         "doc_count": 448,
21         "average_rating": {
22           "value": 5.962700002789497
23         }
24       },
25       {
26         "key": 2012,
27         "doc_count": 404,
28         "average_rating": {
29           "value": 5.961786593160322
30         }
31       },
32       {
33         "key": 2011,
34         "doc_count": 308,
35         "average_rating": {
36           "value": 6.114285714440531
37         }
38       },
39       {
40         "key": 2009,
41         "doc_count": 253,
42         "average_rating": {
43           "value": 6.268774692240921
44         }
45       },
46       {
47         "key": 2010,
48         "doc_count": 249,
49         "average_rating": {
50           "value": 6.239759046868627
51         }
52       },
53       {
54         "key": 2008,
55         "doc_count": 207,
56         "average_rating": {
57           "value": 6.230917865527425
58         }
59       },
60       {
61         "key": 2006,
62         "doc_count": 204,
63         "average_rating": {
64           "value": 6.230917865527425
65         }
66       }
67     ]
68   }
69 }
```