Elasticsearch and Kibana Installation Document

Project Manager: Mr. Gholizade

Author: M.Yakhyan (SASSIZ)

What is Elasticsearch?

Elasticsearch is an open source project that reliably and securely take data from any source, in any format, and search, analyze, and visualize it in real time. Elasticsearch is a distributed, JSON-based search and analytics engine designed for horizontal scalability, maximum reliability, and easy management.

Install Elasticsearch

Download package for your platform. Extract .zip or tar.gz file cd into extracted folder.

Note: In this case we continue installation on Linux.

Note: Elasticsearch do not running with root user for security reason, so execute it with another user who in sudoer group.

cd bin directory and in your terminal run ./elasticsearch command, elasticsearch start running and listen on port 9300. You can communicate with that by curl on address 127.0.0.1:9300 like:

```
root@kali:/var/elasticsearch-5.6.4/bin#curl -XGET
'localhost:9200/_cluster/health?pretty'
```

And it return this reponse:

```
{
  "cluster_name" : "testcluster",
  "status" : "yellow",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "active_primary_shards" : 5,
  "active_shards" : 5,
```

```
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 5,
  "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": 50.0
}
```

Index API in Elasticsearch:

```
curl -XPUT 'localhost:9200/twitter/tweet/1?pretty' -H 'Content-Type:
application/json' -d'
{
    "user" : "kimchy",
    "post_date" : "2009-11-15T14:12:12",
    "message" : "trying out Elasticsearch"
}
'
```

List All Indexes:

```
curl 'localhost:9200/_cat/indices?v'
```

Install plugin in Elasticsearch:

Go to the Elasticsearch directory, cd to bin directory and run this command:

```
./elasticsearch-plugin install mapper-attachments
```

Note: mapper-attachments is plugin name.

Index File In Elasticsearch like pdf,doc,txt,html and etc

Elasticsearch is generally used to index data of types like string, number, date, etc. However, what if you wanted to index a file like a .pdf or a .doc directly and make it searchable? This is a real-time use case in applications like HCM, ERP, and e-commerce.

Elasticsearch caters to this need via a specialized data type called "attachment". For this you must install plugin mapper-attachments in Elasticsearch.

The first step is to install the Elasticsearch plugin: mapper-attachments, which enables ES to recognize the “attachment” data type. In turn, it uses Apache Tika for content extraction and supports several file types such as .pdf, .doc, .xls, .rtf, .html, .odt, etc.

The plugin can be installed by running the following on the command line:

```
$ES_HOME> sudo bin/elasticsearch-plugin install mapper-attachments
```

Once the plugin is installed, restart ES for the new plugin to be loaded into ES. Let's get started by creating a mapping under the index “company”:

```
curl -X POST "http://localhost:9200/company" -d '{
  "mappings":{
    "employee":{
      "properties":{
        "resume":{
          "type":"attachment"
        },
        "name":{
          "type":"string"
        }
      }
    }
```

```
  }
}'
```

As highlighted in the above mapping, “resume” is of the type “attachment”.

Now that the mapping has been created, let’s index a file under the “company” index and type as "employee”. The file must be base64 encoded. The encoded file content is put under “resume” and “name” is set to Mark:

```
Curl –x POST “http://localhost:9200/company/employee/1" -d '{

"resume":
"UWJveCBtYWtlcyBpdCBlYXN5IGZvciB1cyB0byBwcm92aXNpb24gYW4gR
Wxhc3RpY3NlYXJjaCBjbHVzdGVyIHdpdGhvdXQgd2FzdGluZyB0aW1lIG9uI
GFsbCB0aGUgZGV0YWlscyBvZiBjbHVzdGVyIGNvbmZpZ3VyYXRpb24u",

"name":"Mark"

}’
```

Now, let us search. Since “QBOX” is a word that is present in the file that was indexed, let us search for it.

```
url -X POST "http://localhost:9200/company/employee/_search" -d '{
  "query":{
    "query_string":{
      "query":"QBOX"
    }
  }
}'
Search Results:
{
 "took": 394,
 "timed_out": false,
```

```
"_shards": {
  "total": 5,
  "successful": 5,
  "failed": 0
},
"hits": {
  "total": 1,
  "max_score": 0.047945753,
  "hits": [
    {
      "_index": "company",
      "_type": "employee",
      "_id": "1",
      "_score": 0.047945753,
      "_source": {
        "resume":
"UWJveCBtYWtlcyBpdCBlYXN5IGZvciB1cyB0byBwcm92aXNpb24gYW4gRWxhc3RpY3NlYXJjaCBjbHVzdGVyIHdpdGhvdXQgd2FzdGluZyB0aW1lIG9uIGFsbCB0aGUgZGV0YWlscyBvZiBjbHVzdGVyIGNvbmZpZ3VyYXRpb24u",
        "name": "Mark"
      }
    }
  ]
}
}
```

Now that the search is successful, see how we can make indexing files more efficient.

Base64-encoding a file increases the content by 33%. Therefore, storing the Base64 content in the document makes it bulkier, consuming more space. There is a solution for this.

Install Kibana

Downloada Kibana compressed package for your platform and extract that.

For install Kibana first you must install x-pack plugin in ES by run this command:

```
sudo ./elasticsearch-plugin install x-pack
```

restart Elasticsearch and create default username and password by this command:

Create role:

```
curl -XPOST -u elastic 'localhost:9200/_xpack/security/role/events_admin' -H "Content-Type: application/json" -d '{"indices" : [{"names" : [ "events*" ],"privileges" : [ "all" ]},{"names" : [ ".kibana*" ],"privileges" : [ "manage", "read", "index","create" ]}]}'
```

Note: for run this command, ES require password, the default password is "**changeme**".

create user:

```
curl -XPOST -u elastic 'localhost:9200/_xpack/security/user/root' -H "Content-Type: application/json" -d '{"password" : "userpassword","full_name" : "John Doe","email" : "john.doe@anony.mous","roles" : [ "events_admin" ]}'
```

now go to Kibana folder and run it:

```
$kibana_home/bin/kibana
```

In Kibana folder go to config directory and Add credentials to the kibana.yml file:

```
elasticsearch.username: "root"
elasticsearch.password:  "userpassword"
```

now on your browser like chrome, firefox and etc enter this url:

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

and enter username and password for login.

Create Index in Kibana Console

```
PUT /shakespeare
{
 "mappings": {
  "doc": {
   "properties": {
    "speaker": {"type": "keyword"},
    "play_name": {"type": "keyword"},
    "line_id": {"type": "integer"},
    "speech_number": {"type": "integer"}
   }
  }
 }
}
```

Set value to index shakepeare:

```
PUT /shakespeare/doc/2
{
```

```
"speaker": "test ",
"play_name":"test playe name",
"line_id": 14,
"speech_number": 11
}

PUT /logstash-2015.05.18
{
  "mappings": {
    "log": {
      "properties": {
        "geo": {
          "properties": {
            "coordinates": {
              "type": "geo_point"
            }
          }
        }
      }
    }
  }
}
```

Now search query run in the console:

```
GET _search
```

```
{
  "query": {
    "query_string": {
      "query": "working"
    }
  }
}
```

And the response is:

```
{
  "took": 140,
  "timed_out": false,
  "_shards": {
    "total": 32,
    "successful": 32,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 0.3938048,
    "hits": [
      {
        "_index": "shakespeare",
        "_type": "doc",
```

```
      "_id": "1",
      "_score": 0.3938048,
      "_source": {
        "speaker": "test ",
        "play_name": "test play name",
        "line_id": 14,
        "speech_number": 11
      }
    }
  ]
 }
}
```


References:

https://www.elastic.co

https://qbox.io

https://www.tutorialspoint.com/elasticsearch/

https://stackoverflow.com/