Elasticsearch AWBGI Practica 2

Michal Ormos

Universidad de Valladolid, Spain

Abstract

This short paper introduce my work with elasticsearch, from installation to short sample of code fragments of program which demonstrate work with elasticsearch in Python.

1. Introduction

I found given data files interesting, but I rather decide to use one cool and easy to access online database of API data. Accessible from https://swapi.co/.

The version of elasticsearch I was using is 6.2.4 and I also used kibana for early understanding and control of the system, version 6.2.4.

My testing operating system was elementary OS 0.4.1 Loki, Built on "Ubuntu 16.04.3 LTS", Linux 4.13.0-41-generic, GTK+ 3.18.9.

Instalation of Elasticsearch:

- Download elasticsearch from their original websites Elastichsearch 6.2.4 link
- On your linux Ubuntu machine do *sudo apt-get update*
- Copy all files needed to your desired directory and run *./bin/elasticsearch*
- If youre running Elasticsearch on Windows, simply run *bin/elasticsearch.bat* instead.

- You may have some problems with running the elasticsearch on your desired distribution of Linux, Windows. But it may different from version to version so I leave it for reader to find out how to run his elasticsearch.

- You may want to do some more configuration in /config/elasticsearch.yml file.

- To check if your elasticsearch work well, just type http://localhost:9200 in your browser, if you didn't change localhost port in config files. And you will see response similar as:


```
{
  "name" : "x5wB8Wb",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "MBqlynO8SSalflYHrQNkqw",
  "version" : {
    "number" : "6.2.4",
    "build_hash" : "ccec39f",
    "build_date" : "2018-04-12T20:37:28.497551Z",
    "build_snapshot" : false,
    "lucene_version" : "7.2.1",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```


- In same manner, if you want, you can install and run Kibana.

2. Examples you can find in script

The srcipt name is *main.py*. Is made in 'automated' way. So the user/teacher can observe the work of script.It has included *sleep(x)* statements for better understanding. It was implemented in language *Python* with help of libraries *Elasticsearch*, *json*, *request*, *time*. IT WAS NOT TESTED ON WINDOWS.

How to run the script:

- Version with sleep statements:

```
python3 –O main.py
```

- Version without sleep statements:

```
python3 main.py
```

- with any problems compiling or running the program, is recommended to downgrade to python version 2, so run will look like *python -O main.py*.

Brief illustration of script queries:

- 1. Connect to our cluster

```
es = Elasticsearch(
        [{'host': 'localhost', 'port': 9200}])
```

- 2. Index some test data, get that data, delete that data

```
es.index(index='test-index', doc_type='test', id=1, \
        body={'test': 'test'})

res = es.get(index='test-index', doc_type='test', id=1)

delete = es.delete(index='test-index', doc_type='test', id=1)
```

- 3. Index some more complicated test data

```
es.index(index='sw', doc_type='people', id=1, body={
        "name": "Luke Skywalker",
        "height": "172",
        "mass": "77",
        "hair_color": "blond",
        "birth_year": "19BBY",
        "gender": "male",
})
```

- 4. Lets add all SW people

```
r = requests.get('http://localhost:9200')
i = 1
while r.status_code == 200:
    r = requests.get('http://swapi.co/api/people/'+ str(i))
    es.index(index='sw', doc_type='people', id=i, \
        body=json.loads(r.content))
    i=i+1
print("We added %d characters from API" %i)
```

- 5. Who is under id 5?

```
res = es.get(index='sw', doc_type='people', id=5)
```

- 6. Lets add more SW people

```
r = requests.get('http://localhost:9200')
i = 18
while r.status_code == 200:
    r = requests.get('http://swapi.co/api/people/'+ str(i))
    es.index(index='sw', doc_type='people', id=i, \
        body=json.loads(r.content))
    i=i+1
print("We added %d more characters from API" %i)
```

- 7. Do we have some Darth Vader here? Let's search him.

```
res = es.search(index="sw", body={"query": {"match" : \
        { "name" : "Darth" }}})
```

- 8. Let's see of some name start's with sting 'Lu'

```
res = es.search(index="sw", body={"query": {"prefix" : \
        { "name" : "lu" }}})
```

- 9. Let's find who is born in year 19

```
res = es.search(index="sw", body={"query": {"prefix" : \
        { "birth_year" : "19" }}})
```

- 10. Let's try som fuzzy query

```
res = es.search(index="sw", body={"query": {"fuzzy" : \
        { "name" : {"value" : "jaba", "max_expansions":5}}}})
```

3. Conclusion

We introduce, implemented and demonstrate use of Eleasticsearch. This was just a simple overview on how to set up your Elasticsearch server and start working with some data using Python. We can see it is very powerful and fast search engine which can take care of many queries with easy and speed.