Если вы видите что-то необычное, просто сообщите мне.

Работа с elasticsearch

Создание индекса

```
curl -u ПОЛЬЗВАТЕЛЬ:ПАРОЛЬПОЛЬЗОВАТЕЛЯ -X GET "IPАДРЕС:5002/samples/"
```

Добавление

```
curl -u ПОЛЬЗВАТЕЛЬ:ПАРОЛЬПОЛЬЗОВАТЕЛЯ -XPOST --header 'Content-Type: application/json' IPАДРЕС:5002/sample/\_doc -d '{
   "school" : "asdasaTEST", "@timestamp" : "'"$(date +%Y-%m-%dT%H:%M:%S)"'"
}'
```

Измнение

```
curl -u ПОЛЬЗВАТЕЛЬ:ПАРОЛЬПОЛЬЗОВАТЕЛЯ -XPUT --header 'Content-Type: application/json' IPАДРЕС:5002/samples/\\\_doc/4 -d '{

   "school" : "asaTEST", "@timestamp" : "'"$(date +%Y-%m-%dT%H:%M:%S)"'"

}'
```

> Вот несколько распространненых примеров команд ElasticSearch используя `curl` ElasticSearch часто сложен. Тут мы постараемся сделать его легче.

Удаление индексов.

Ниже индекс назван *sample*

```
curl -X DELETE 'http://localhost:9200/samples'
```

Показать все индексы

```
curl -X GET 'http://localhost:9200/\_cat/indices?v'
```

Показать все докуменыт в индексах

```
curl -X GET 'http://localhost:9200/sample/\_search'
```

Запрос используя параметры URL.

Тут мы используем Lucene запрос формат для написания: *q=school:Harvard*

```
curl -X GET http://localhost:9200/samples/\_search?q=school:Harvard
```

Запрос с JSON aka DSL для запросов в Elasticsearch.

Вы можете использовать параметры для URL. Но вы можете так же использоват JSON, как показано в следующем примере. JSON будет легче для чтения и отладки, когда у вас сложный запрос, чем один длинный запрос в виде URL.

```
curl -XGET --header 'Content-Type: application/json' http://localhost:9200/samples/\_search -d '{
    "query" : {
      "match" : { "school": "Harvard" }
   }
}'
```

Показать список индексов.

Все поля индексов. Выведет все поля и их типы в каждом индексе.

```
curl -X GET http://localhost:9200/samples
```

Добавить данные

```
curl -XPUT --header 'Content-Type: application/json' http://localhost:9200/samples/\_doc/1 -d '{
   "school" : "Harvard"
}'
```

Обновление документа.

Вот как добавить поле к существующему документу. Для начала создадим его, затем обновим.

Копирование

```
curl -XPUT --header 'Content-Type: application/json' http://localhost:9200/samples/\_doc/2 -d '
{
    "school": "Clemson"
}'
```

```
curl -XPOST --header 'Content-Type: application/json' http://localhost:9200/samples/\_doc/2/\_update -d '{
"doc" : {
             "students": 50000}
}'
```

Бэкап для индкса.

```
curl -XPOST --header 'Content-Type: application/json' http://localhost:9200/\_reindex -d '{
  "source": {
    "index": "samples"
  },
  "dest": {
    "index": "samples\_backup"
  }
}'
```

Объем загруженных данных в формате JSON:

```
export pwd="elastic:"
```

```
curl --user $pwd  -H 'Content-Type: application/x-ndjson' -XPOST 'https://58571402f5464923883e7be42a037917.eu-central-1.aws.cloud.es.io:9243/0/\_bulk?pretty' --data-binary @<file>
```

Показать здоровье кластера

```
curl --user $pwd  -H 'Content-Type: application/json' -XGET https://58571402f5464923883e7be42a037917.eu-central-1.aws.cloud.es.io:9243/\_cluster/health?pretty
```

Сбор

Для nginx веб сервера это произведет подсчет пользователей по городам.

```
curl -XGET --user $pwd --header 'Content-Type: application/json'
 https://58571402f5464923883e7be42a037917.eu-central-1.aws.cloud.es.io:9243/logstash/\_search?pretty -d '{
      "aggs": {
          "cityName": {
                "terms": {
                 "field": "geoip.city\_name.keyword",
                          "size": 50


      }
   }
  }
}'
```

Это расширит на код ответа количества городов в nginx логах веб сервера.

```
curl -XGET --user $pwd --header 'Content-Type: application/json'
 https://58571402f5464923883e7be42a037917.eu-central-1.aws.cloud.es.io:9243/logstash/\_search?pretty -d '{
      "aggs": {
        "city": {
            "terms": {
                  "field": "geoip.city\_name.keyword"
            },
      "aggs": {
        "responses": {
```

```
            "terms": {
                "field": "response"
            }
        }
      }
    },
    "responses": {
            "terms": {
                "field": "response"
            }
     }
  }
}'
```

Использование ElasticSearch с базовой авторизацией.

Если у вас включена безопасность в ElasticSearch, тогда вам необходимо предоставить пользователя и пароль, как показано ниже для всех команд-запросов:

```
curl -X GET 'http://localhost:9200/\_cat/indices?v' -u elastic:(password)
```

Красивый вывод.

Добавьте ?pretty=true к любому поиску чтобы вывести причесаный JSON:

```
curl -X GET 'http://localhost:9200/(index)/\_search'?pretty=true
```

Запрос на получение только определенных полей.

Вернет только определенные поля поместив их в массив *_source*

```
GET filebeat-7.6.2-2020.05.05-000001/\_search
 {
   "\_source": \["suricata.eve.timestamp","source.geo.region\_name","event.created"\],
   "query":      {
      "match" : { "source.geo.country\_iso\_code": "GR" }
   }
}
```

Запрос по дате.

В случае когда поле типа дата вы можете использовать математику дат:

```
GET filebeat-7.6.2-2020.05.05-000001/\_search
 {
   "query": {
      "range" : {
         "event.created": {
            "gte" : "now-7d/d"
         }
      }
}
}
```

Revision #3

Created 13 October 2022 14:00:26 by gasick

Updated 2 November 2022 07:25:12 by gasick