

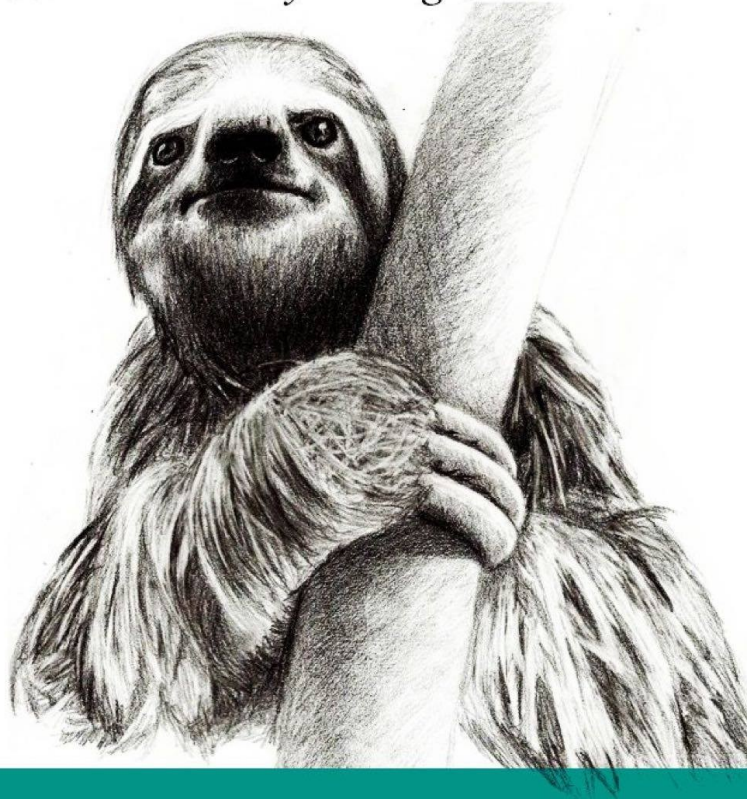
Nie szukaj, znajdź

Elasticsearch nie tylko dla Wielkodanowców

Ja

- Łukasz i inne
- developer
- autofirma
- elastycznie
- gram w nogę

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- demo 1 : tekst
- demo 2 : agregacje
- pytania i odpowiedzi

Motywacja

- Po co nam dane
 - Gdzie są
 - Kto ich szuka
 - Dane a informacje
- Co gdy brak informacji
 - Support team jest męczony
 - Leczymy skutki
 - Nie widzisz

Historie sukcesu

- Instalacja i uruchomienie
 - Docker
 - REST jest łatwo konsumowalny
- Reakcje są nagrodą
 - Namierzenie botów
 - Błędy po wdrożeniu
 - Automatyczne alerty – zdążyć przed ticketem
 - Wąskie gardła

Nasz „krajobraz”



Oni już to mają

- Github
 - kod
- Stackoverflow
 - pytania i odpowiedzi
- Symantec
 - zdarzenia od klientów
- ...
- twój zespół?

Agenda

- motywacja
- **pierwsze uruchomienie**
- troszkę teorii
- demo 1 : tekst
- demo 2 : agregacje
- pytania i odpowiedzi

Opcje

- On premises – hostuj sobie sam
 - instalaki (Winda i Linuch)
 - Docker-owy obrazek
- Hostingi
 - AWS
 - Bonsai.io
 - usługa albo Linuxbox

zostań kompozytorem

```
luk@luk-UX410UAK: ~/prj/nieszukaj
version: '2'
services:
  kibana:
    image: kibana
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch
    environment:
      - ELASTICSEARCH_URL=http://elasticsearch:9200
      - elasticsearch.username=
      - elasticsearch.password=
  elasticsearch:
    image: luk/polski_elasticsearch
    ports:
      - 9200:9200
    volumes:
      - /var/rss_esdata:/usr/share/elasticsearch/data
    environment:
      - bootstrap.memory_lock=true
      - xpack.security.enabled=false
      - xpack.monitoring.enabled=false
      - xpack.graph.enabled=false
      - "ES_JAVA_OPTS=-Xms4g -Xmx4g"
      - network.host=0.0.0.0
    cap_add:
      - IPC_LOCK
    ulimits:
      memlock:
        soft: -1
        hard: -1
      mem_limit: 8g
(END)
```

minidemo: Pierwsze uruchomienie

- cel
 - pokazać że skomplikowany silnik wyszukiwania da się uruchomić na Twoim kompie
 - pokazać że to żyje
- kamienie milowe
 - Linux + Docker
 - Health
 - Any REST
 - curl
 - Kibana devtools

Agenda

- motywacja
- pierwsze uruchomienie
- **troszkę teorii**
- demo 1 : tekst
- demo 2 : agregacje
- pytania i odpowiedzi

Pokaż kotku co masz w środku

- Elasticsearch to
 - Search engine
 - Biblioteka Lucene
 - RESTowe narzędzie
- Pojęcia
 - index > type > document
 - cluster > node > shard
 - replica

Podstawowe operacje

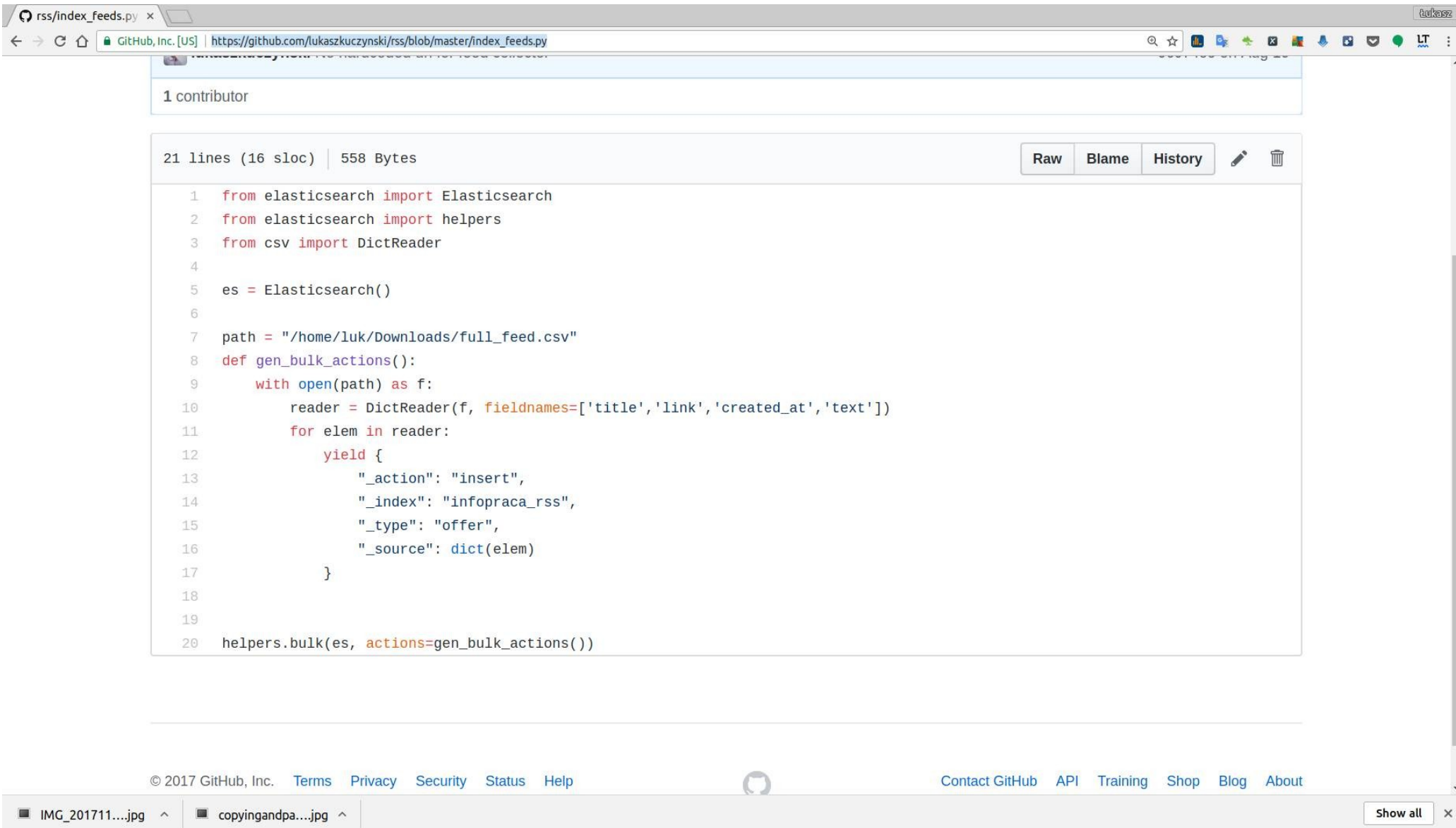
- Cel
 - Zindeksujemy dokument i zobaczymy co automatycznie stworzył nam silnik ES
- Kamienie milowe:
 - Tworzę indeks
 - Wrzucam dokument
 - Indeks ma mapowanie
 - Szukam i znajduję



ES to .. bogactwo API

- REST
- Language-specific
 - Java
 - official ES library
 - Spring
 - SpringData Repository
 - Python
 - pokażmy kod

w Pythonie jest łatwo




1 contributor

21 lines (16 sloc) | 558 Bytes

Raw Blame History

```
1 from elasticsearch import Elasticsearch
2 from elasticsearch import helpers
3 from csv import DictReader
4
5 es = Elasticsearch()
6
7 path = "/home/luk/Downloads/full_feed.csv"
8 def gen_bulk_actions():
9     with open(path) as f:
10         reader = DictReader(f, fieldnames=['title', 'link', 'created_at', 'text'])
11         for elem in reader:
12             yield {
13                 "_action": "insert",
14                 "_index": "infopraca_rss",
15                 "_type": "offer",
16                 "_source": dict(elem)
17             }
18
19
20 helpers.bulk(es, actions=gen_bulk_actions())
```

© 2017 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

IMG_201711....jpg ^ copyingandpa....jpg ^ Show all x

Elasticsearch to „baza”

- Nie zawsze jako podstawa
 - Świetny cache
 - Aplikacje zorientowane na search
- Security
 - X-Pack to dodatek
- Transakcje
 - Wersjonowanie dokumentów

Twój Devops doceni!




- Powtarzalne
 - łatwo daje się *zdockeryzować*
- Skalowalne
 - replica
 - status zielony gdy 1 replica
- Chmurowe i znane
 - sporo providerów – PaaS
 - community



Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- **demo 1 : tekst**
- demo 2 : agregacje
- pytania i odpowiedzi



Demo #1 : twitter

- Cel: o czym piszą fani Java
- Zindeksowana historia
 - pokaz 
- Live: L z ELK
 - konfiguracja
 - output
 - konsola
 - ES 
- Query to nie Select
 - Search Lite vs DSL 

Relevantna informacja

- Robimy search
 - Search text
- URI : index/typ/_search
- Search
 - Score
 - TF-IDF
 - Vector Space Model
- **Jak** a nie **Czy** pasuje

Filtr czy Query

- Filtr zwraca rezultaty
 - constant_score 
- Zalety
 - Szybkość
 - Cache
- Filter + Query
 - bool 

Analiza jest z pudełka

- text vs keyword
- analiza
 - char filter (np. HTML strip)
 - tokenizer (np. whitespace)
 - token filter (np. stopwords)
- search korzysta z tego
 - index time
 - fraza szukana
- dopasowanie
 - wbudowane
 - stwórz sobie sam

in [Custom Analyzers](#).

Built-in Analyzers



However, Elasticsearch also ships with prepackaged analyzers that you can use directly. We list the most important ones next and, to demonstrate the difference in behavior, we show what terms each would produce from this string:

```
"Set the shape to semi-transparent by calling set_trans(5)"
```

Standard analyzer

The standard analyzer is the default analyzer that Elasticsearch uses. It is the best general choice for analyzing text that may be in any language. It splits the text on *word boundaries*, as defined by the [Unicode Consortium](#), and removes most punctuation. Finally, it lowercases all terms. It would produce

```
set, the, shape, to, semi, transparent, by, calling, set_trans, 5
```

Simple analyzer

The simple analyzer splits the text on anything that isn't a letter, and lowercases the terms. It would produce

```
set, the, shape, to, semi, transparent, by, calling, set, trans
```

Whitespace analyzer

The whitespace analyzer splits the text on whitespace. It doesn't lowercase. It would produce

```
Set, the, shape, to, semi-transparent, by, calling, set_trans(5)
```

Language analyzers

- + [Life Inside a Cluster](#)
- + [Data In, Data Out](#)
- + [Distributed Document Store](#)
- + [Searching—The Basic Tools](#)
- [Mapping and Analysis](#)
 - [Exact Values Versus Full Text](#)
 - [Inverted Index](#)
 - [Analysis and Analyzers](#)**
 - [Mapping](#)
 - [Complex Core Field Types](#)

- + [Full-Body Search](#)
- + [Sorting and Relevance](#)
- + [Distributed Search Execution](#)
- + [Index Management](#)
- + [Inside a Shard](#)

- + [Search in Depth](#)
- + [Dealing with Human Language](#)
- + [Aggregations](#)
- + [Geolocation](#)
- + [Modeling Your Data](#)
- + [Administration, Monitoring, and Deployment](#)

Docs

```
"analysis": {
  "filter": {
    "english_stop": {
      "type": "stop",
      "stopwords": "_english_" ❶
    },
    "english_keywords": {
      "type": "keyword_marker",
      "keywords": [] ❷
    },
    "english_stemmer": {
      "type": "stemmer",
      "language": "english"
    },
    "english_possessive_stemmer": {
      "type": "stemmer",
      "language": "possessive_english"
    }
  },
  "analyzer": {
    "english": {
      "tokenizer": "standard",
      "filter": [
        "english_possessive_stemmer",
        "lowercase",
        "english_stop",
        "english_keywords",
        "english_stemmer"
      ]
    }
  }
}
```



Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- demo 1 : tekst
- **demo 2 : agregacje**
- pytania i odpowiedzi



Nie-pełnotekstowo

- Mapowanie
 - Long, inne
- Agregacje
 - Pojęcia
 - Bucket, Metric
- Tekst
 - Popularne frazy
- Liczby
 - Średnia, Kumulacja



Demo #2, trzęsienia ziemi

- Zdarzenia
 - Miejsce, ile ofiar, gdzie
- Technicznie
 - Logstash
 - Kibana
- Pokazujemy
 - Kiedy ludzie ginęli 
 - Geo punkty 


Tworzenie jest proste

- Kibana
 - search
 - wizualizacja
- Elasticsearch
 - bucket 
 - bucket + metric 

Mapowanie

- Nie ma DDL
- Auto
 - Coś powstało 
- Możesz zdefiniować
 - Put mapping 

Agregacje można zawężyć

- Filtr
 - constant_score
- Query
 - Zapytanie
 - Domyślnie : match_all
- Przykład
 - Dla konkretnego kraju 

Dashboard = agregacje + czas

- dashboard to zbiór wizualizacji
- przypadki użycia
 - nieustanny monitoring logów
 - trzymaj rękę na pulsie social-media
- events dashboard
 - przefiltrujmy go razem



Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- demo 1 : tekst
- demo 2 : agregacje
- **pytania i odpowiedzi**

QA

opcjonalne

- Jest Stempel
- Jest też chiński
- Filebeat jest na topie w ELK
- typo = Fuzzy
- sugestie = Tokenizuj mądrze

in touch

- Twitter : @panlukasz
- blog : lukcreates.pl

Nie szukaj, znajdź

Elasticsearch nie tylko dla Wielkodarowców

Dzisiejsza prezentacja jest przeznaczona zarówno dla osób technicznych jak i nietechnicznych.

Pokażę dzisiaj zarówno bezpośrednie zastosowania (tzw. przypadki użycia) Elasticsearcha jak i radość z pisania kodu :)

Dla każdego coś dobrego

Chciałbym żebyś po tej prezentacji uwierzył że silniki wyszukiwania to przyszłość analizy danych a jest to dostępne dla każdego.

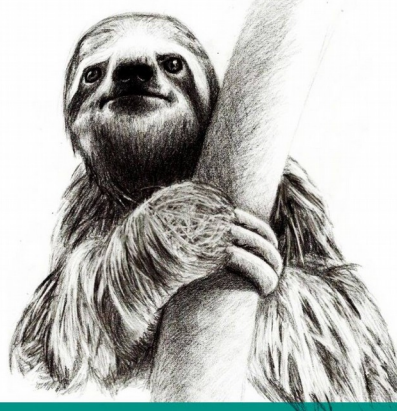
Doradca, jak zyskać wirtualnego doradcę!

Ja

- Łukasz i inne
- developer
- autofirma
- elastycznie
- gram w nogę

Ja jestem skromnym programistą, z zamiłowania. 7 lat programuję w różnych językach. Ostatnio najczęściej w Pythonie choć moje stanowisko to Programista Javy.

Cutting corners to meet arbitrary management deadlines



Essential

Copying and Pasting from Stack Overflow

O'REILLY®

The Practical Developer
@ThePracticalDev

Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- demo 1 : tekst
- demo 2 : agregacje
- pytania i odpowiedzi

dlaczego od uruchomienia
bo najbardziej męczy mnie poznawanie teorii bez
praktyki
jestem za jak najszybszym (agile) dostarczaniem
produktu dla SIEBIE
a potem stopniowym usprawnianiem

Motywacja

- Po co nam dane
 - Gdzie są
 - Kto ich szuka
 - Dane a informacje
- Co gdy brak informacji
 - Support team jest męczony
 - Leczymy skutki
 - Nie widzisz

BigData to brzmi szumnie ale rzeczywiście jesteśmy otoczeni danymi

coraz mniejsze urządzenia potrafią generować dane żeby je zrozumieć trzeba je jakoś przerobić

gdzie

różne źródła to komplikacja, wiele serwerów, wiele baz/systemów plików

kompetencje ludzi którzy ich szukają mają znaczenie

Dane pochodzą z wielu źródeł
dane to nie informacje

Wizualizacja jest istotna

Jesteśmy stworzeni w taki sposób że obrazy trafiają do nas mocniej niż tekst – pokaż dane

Historie sukcesu

- Instalacja i uruchomienie
 - Docker
 - REST jest łatwo konsumowalny
- Reakcje są nagrodą
 - Namierzenie botów
 - Błędy po wdrożeniu
 - Automatyczne alerty – zdążyć przed ticketem
 - Wąskie gardła

Docker w topologii korporacyjnej

Rest to świetny sposób komunikacji ES ze światem –
łatwo się integruje (vs SOLR)

sukces historie

zobaczyliśmy na własne oczy która część systemu
działa najwolniej (wąskie gardło)

Nasz „krajobraz”



Oni już to mają

- Github
 - kod
- Stackoverflow
 - pytania i odpowiedzi
- Symantec
 - zdarzenia od klientów
- ...
- twój zespół?

Agenda

- motywacja
- **pierwsze uruchomienie**
- troszkę teorii
- demo 1 : tekst
- demo 2 : agregacje
- pytania i odpowiedzi

Opcje

- On premises – hostuj sobie sam
 - instalki (Winda i Linuch)
 - Docker-owy obrazek
- Hostingi
 - AWS
 - Bonsai.io
 - usługa albo Linuxbox

zostań kompozytorem

```
luk@luk-UX410UAK:~/jazz/nieszukaj
version: '2'
services:

  kibana:
    image: kibana
    ports:
      - 5601:5601
    depends_on:
      - elasticsearch
    environment:
      - ELASTICSEARCH_URL=http://elasticsearch:9200
      - elasticsearch.username=
      - elasticsearch.password=

  elasticsearch:
    image: luk/polski_elasticsearch
    ports:
      - 9200:9200
    volumes:
      - /var/rss_esdata:/usr/share/elasticsearch/data
    environment:
      - bootstrap.memory_lock=true
      - xpack.security.enabled=false
      - xpack.monitoring.enabled=false
      - xpack.graph.enabled=false
      - "ES_JAVA_OPTS=-Xms4g -Xmx4g"
      - network.host=0.0.0.0
    cap_add:
      - IPC_LOCK
    ulimits:
      memlock:
        soft: -1
        hard: -1
      mem_limit: 8g

(END)
```


minidemo: Pierwsze uruchomienie

- cel
 - pokazać że skomplikowany silnik wyszukiwania da się uruchomić na Twoim kompie
 - pokazać że to żyje
- karmienie milowe
 - Linux + Docker
 - Health
 - Any REST
 - curl
 - Kibana devtools

10

pierwsze uruchomienie

1. pokazujemy compose
2. curl na 9200
3. idziemy do przeglądarki
4. otwieramy devtools

Agenda

- motywacja
- pierwsze uruchomienie
- **troszkę teorii**
- demo 1 : tekst
- demo 2 : agregacje
- pytania i odpowiedzi

Pokaż kotku co masz w środku

- Elasticsearch to
 - Search engine
 - Biblioteka Lucene
 - RESTowe narzędzie
- Pojęcia
 - index > type > document
 - cluster > node > shard
 - replica

Pojęcia

Niech analogia z bazy relacyjnej będzie dziś z nami

Indeks – baza danych

Typ – jak tabela

Dokument to krotka

Document to json

Node to serwer

Cluster zbiera node razem

Shard to podstawowa jednostka składowania
(Lucene)

Główne i replica

Podstawowe operacje

- Cel
 - Zindeksujemy dokument i zobaczymy co automatycznie stworzył nam silnik ES
- Kamienie milowe:
 - Tworzę indeks
 - Wrzucam dokument
 - Indeks ma mapowanie
 - Szukam i znajduję



wrzucam blogpost

first_interaction.txt

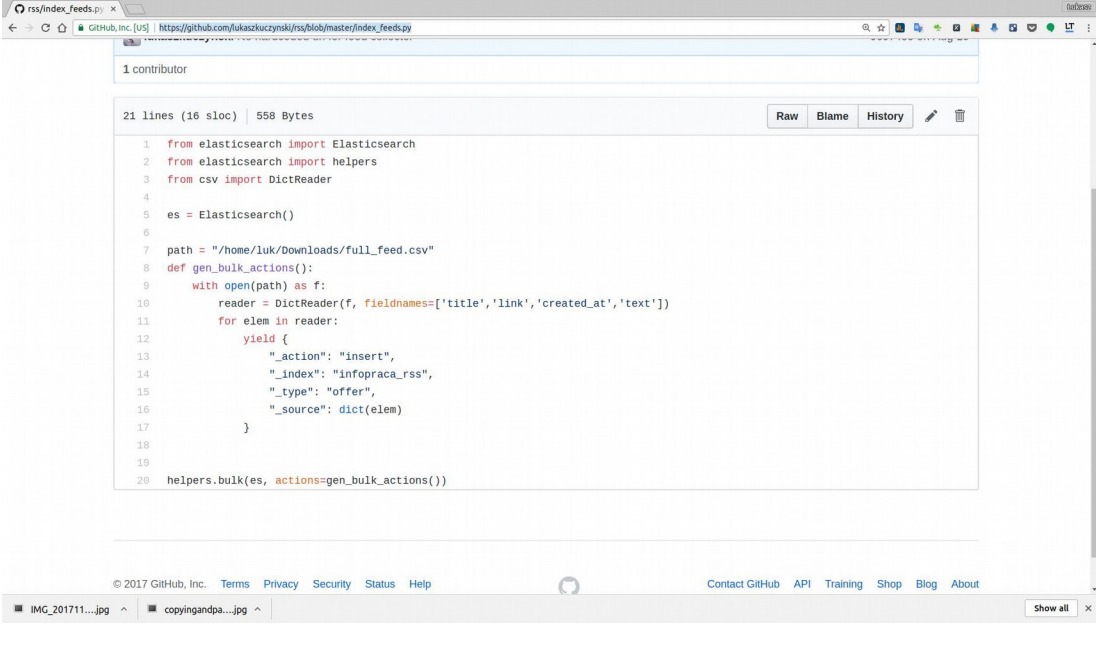
ES to .. bogactwo API

- REST
- Language-specific
 - Java
 - official ES library
 - Spring
 - SpringData Repository
 - Python
 - pokażmy kod

Przykłady

W dev_tools (kiedyś sense wtyczka do przeglądarek)
używaliśmy API REST

w Pythonie jest łatwo



The screenshot shows a web browser displaying a GitHub repository page for 'rss/index_feeds.py'. The page indicates 1 contributor and 21 lines of code (16 sloc) totaling 558 Bytes. The code is a Python script that uses the Elasticsearch Python client and the 'helpers' module to bulk index data from a CSV file into an Elasticsearch index. The script imports 'Elasticsearch' and 'helpers' from 'elasticsearch', and 'DictReader' from 'csv'. It initializes an 'Elasticsearch' object, sets a local path to a CSV file, and defines a generator function 'gen_bulk_actions()' that reads the CSV file and yields Elasticsearch bulk actions. Finally, it calls 'helpers.bulk()' to perform the bulk indexing.

```
1 from elasticsearch import Elasticsearch
2 from elasticsearch import helpers
3 from csv import DictReader
4
5 es = Elasticsearch()
6
7 path = "/home/luk/Downloads/full_feed.csv"
8 def gen_bulk_actions():
9     with open(path) as f:
10         reader = DictReader(f, fieldnames=['title', 'link', 'created_at', 'text'])
11         for elem in reader:
12             yield {
13                 "_action": "insert",
14                 "_index": "infopraca_rss",
15                 "_type": "offer",
16                 "_source": dict(elem)
17             }
18
19
20 helpers.bulk(es, actions=gen_bulk_actions())
```

SDK python
Pokazujemy kod

Python helpers
Piękno generatorów (obsługiwanie kolekcji),
domyślnie tworzone chunks do wydajnego
indeksowania danych (bulk API)

Elasticsearch to „baza”

- Nie zawsze jako podstawa
 - Świetny cache
 - Aplikacje zorientowane na search
- Security
 - X-Pack to dodatek
- Transakcje
 - Wersjonowanie dokumentów

Primary datasource – nunu!

niewiele jest aplikacji które potrzebują bezpiecznego zapisu, ostatnio słyszałem o tym że są całe ruchy społeczne nakłaniające do użycia w fintech (dla finansjery)

Xpack to bogaty dodatek, posiada wiele rozszerzeń niebędących częścią core ES

Transakcje nie są od ręki,
wersja dokumentów sprawdzanie po zapisie
dokument który stworzyliśmy krok temu też
miał wersję, teraz jak zrobimy PUT 2x to będzie
podniesiona

Twój Devops doceni!

- Powtarzalne
 - łatwo daje się *zdockeryzować*
- Skalowalne
 - replica
 - status zielony gdy 1 replica
- Chmurowe i znane
 - sporo providerów - PaaS
 - community



Skalowalne

pokaż status/ indices yellow.txt




Chmurowe

jak coś kiedyś pisałeś to wiesz jak ważne jest community

Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- **demo 1 : tekst**
- demo 2 : agregacje
- pytania i odpowiedzi

Demo #1 : twitter

- Cel: o czym piszą fani Java
- Zindeksowana historia
 - pokaz 
- Live: L z ELK
 - konfiguracja
 - output
 - konsola
 - ES 
- Query to nie Select
 - Search Lite vs DSL 

historia wizualizacja [tutaj](#)

Logstash nappełnił dane
odpalamy

Query to nie select

Lite vs DSL
search_dsl_lite.txt

pole _all wspomnieć

Relevantna informacja

- Robimy search
 - Search text
- URI : index/typ/_search
- Search
 - Score
 - TF-IDF
 - Vector Space Model
- **Jak** a nie **Czy** pasuje

search

pandas,javascript,spring

score

TF - jak często token występuje w dokumencie
(więcej > większy score)

IDF - jak często wyrażenie występuje w
każdym dokumencie

normalizacja długości pola (krótsze lepiej)

VSM – dokumenty zamieniane na wektory na
podstawie score

filtr vs search

filtr: chcę kawę a nie herbatę

zapytanie punktowane: jadę na wakacje i ważę w
głowie czynniki- odległość od morza, basen,
jedzenie, cena itp...

Filtr czy Query

- Filtr zwraca rezultaty
 - constant_score ⚡
- Zalety
 - Szybkość
 - Cache
- Filter + Query
 - bool ⚡

filter.txt

zalety
nie zależy nam na wyniku
cachowanie z pudełka

filter_query.txt

Analiza jest z pudełka

- text vs keyword
- analiza
 - char filter (np. HTML strip)
 - tokenizer (np. whitespace)
 - token filter (np. stopwords)
- search korzysta z tego
 - index time
 - fraza szukana
- dopasowanie
 - wbudowane
 - stwórz sobie sam

blog_analyze

analizator.txt

Docs

in Custom Analyzers.

Built-in Analyzers

However, Elasticsearch also ships with prepackaged analyzers that you can use directly. We list the most important ones next and, to demonstrate the difference in behavior, we show what terms each would produce from this string:

"Set the shape to semi-transparent by calling set_trans(5)"

Standard analyzer

The standard analyzer is the default analyzer that Elasticsearch uses. It is the best general choice for analyzing text that may be in any language. It splits the text on *word boundaries*, as defined by the [Unicode Consortium](#), and removes most punctuation. Finally, it lowercases all terms. It would produce

set, the, shape, to, semi, transparent, by, calling, set_trans, 5

Simple analyzer

The simple analyzer splits the text on anything that isn't a letter, and lowercases the terms. It would produce

set, the, shape, to, semi, transparent, by, calling, set, trans

Whitespace analyzer

The whitespace analyzer splits the text on whitespace. It doesn't lowercase. It would produce

Set, the, shape, to, semi-transparent, by, calling, set_trans(5)

Language analyzers

edit

- + Life Inside a Cluster
- + Data In, Data Out
- + Distributed Document Store
- + Searching—The Basic Tools
 - Mapping and Analysis
 - Exact Values Versus Full Text
 - Inverted Index
 - Analysis and Analyzers**
 - Mapping
 - Complex Core Field Types
- + Full-Body Search
- + Sorting and Relevance
- + Distributed Search Execution
- + Index Management
- + Inside a Shard

+ Search in Depth

+ Dealing with Human Language

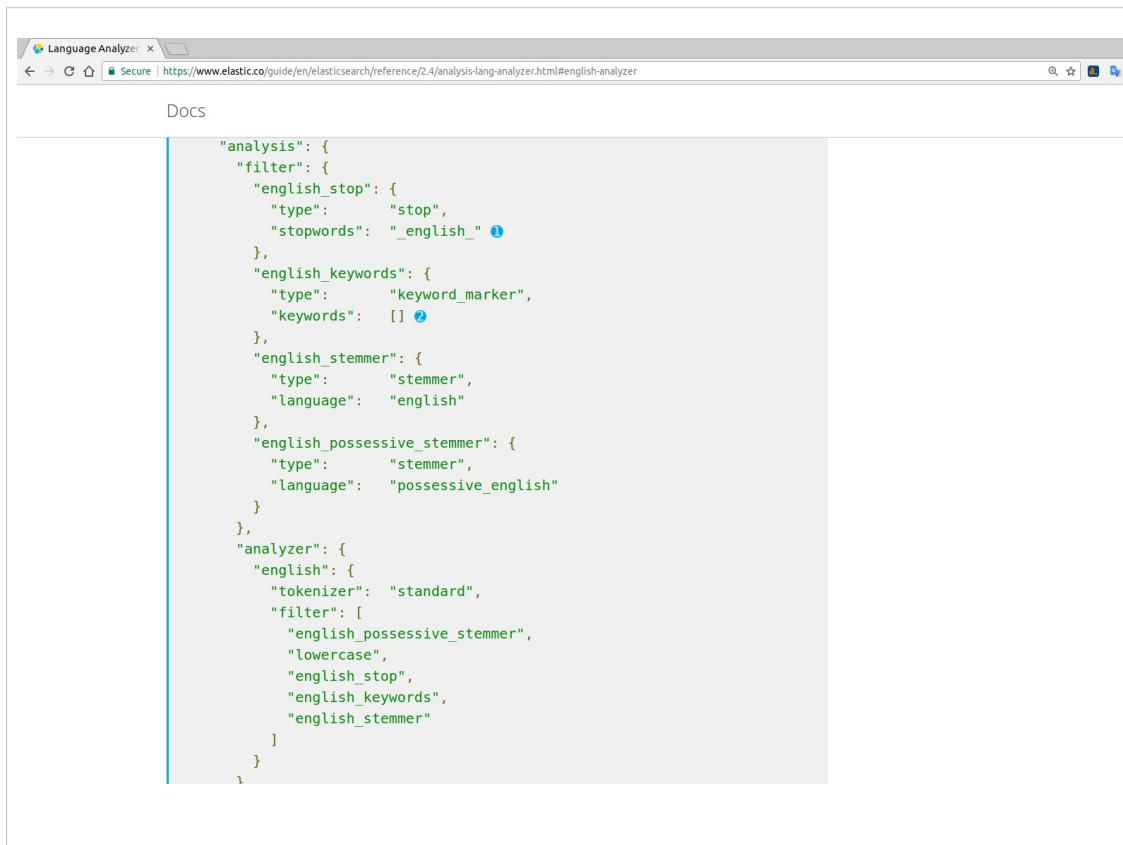
+ Aggregations

+ Geolocation

+ Modeling Your Data

+ Administration, Monitoring, and Deployment

1 analizator



2 analizator

Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- demo 1 : tekst
- **demo 2 : agregacje**
- pytania i odpowiedzi

Nie-pełnotekstowo

- Mapowanie
 - Long, inne
- Agregacje
 - Pojęcia
 - Bucket, Metric
- Tekst
 - Popularne frazy
- Liczby
 - Średnia, Kumulacja

przed mapowaniem ES widzi json jako tekst
już dzisiaj widzieliśmy 1 agregację
datehistogram

elasticsearch to także liczby

pojęcia podstawowe

porównanie HAVING MAX z SQL?

bucket to zbiór, sposób podziału

metric to operacja na tym, np count (podstawowa)

date histogram który widzieliśmy to bucket typu



czas(1min) count() pokaż na wykresie to!

tekst

popularne frazy

liczby: wymień kilka ..

Demo #2, trzęsienia ziemi

- Zdarzenia
 - Miejsce, ile ofiar, gdzie
- Technicznie
 - Logstash
 - Kibana
- Pokazujemy
 - Kiedy ludzie ginęli 
 - Geo punkty 

pokazujemy 2 wizualizacje

Tworzenie jest proste

- Kibana
 - search
 - wizualizacja
- Elasticsearch
 - bucket ⚡
 - bucket + metric ⚡

histogram

bucket.txt

bucket_metric.txt

smaczne api – łatwo konsumować te JSONy
to właśnie Kibana pokazuje

Mapowanie


- Nie ma DDL
- Auto
 - Coś powstało ⚡
- Możesz zdefiniować
 - Put mapping ⚡

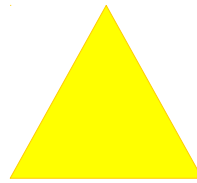
czy zauważyłeś że nie powstało żadne DDL do tej pory? Elasticsearch sam zgaduje nazwy typów i tworzy mapowanie
funkcja którą można wyłączyć ale na potrzeby developerskie jest po prostu wygodna

mapowanie plik tworzymy dla bloga a potem korzystamy z radości działań na liczbach które dopiero co zindeksowaliśmy


mapowanie powstało
zdefiniujemy mapowanie
teraz można range, avg agregacje

Agregacje można zawężyć

- Filtr
 - constant_score
- Query
 - Zapytanie
 - Domyślnie : match_all
- Przykład
 - Dla konkretnego kraju 



Dashboard = agregacje + czas

- dashboard to zbiór wizualizacji
- przypadki użycia
 - nieustanny monitoring logów
 - trzymaj rękę na pulsie social-media
- events dashboard
 - przefiltrujmy go razem 

Agenda

- motywacja
- pierwsze uruchomienie
- troszkę teorii
- demo 1 : tekst
- demo 2 : agregacje
- **pytania i odpowiedzi**

QA

opcjonalne

- Jest Stempel
- Jest też chiński
- Filebeat jest na topie w ELK
- typo = Fuzzy
- sugestie = Tokenizuj mądrze

in touch

- Twitter : @panlukasz
- blog : lukcreates.pl