Artur Gramacki

Uniwersytet Zielonogórski

Instytut Sterowania i Systemów Informatycznych

2019

O systemie

- Historia

Version	Original release date	Latest version	Release date	Maintenance Status[9]
0.4	2010-02-08	0.4.0	2010-02-08	No longer supported
0.5	2010-03-05[10]	0.5.1	2010-03-09	No longer supported
0.6	2010-04-09[11]	0.6.0	2010-04-09	No longer supported
0.7	2010-05-14[12]	0.7.1	2010-05-17[13]	No longer supported
0.8	2010-05-27[14]	0.8.0	2010-05-27	No longer supported
0.9	2010-07-26[15]	0.9.0	2010-07-26	No longer supported
0.10	2010-08-27[16]	0.10.0	2010-08-27	No longer supported
5.4	2017-05-04[61]	5.4.3	2017-06-27[62]	Still supported
5.5	2017-07-06[63]	5.5.3	2017-07-06[64]	Still supported
5.6	2017-09-11[65]	5.6.8	2018-02-20[66]	Still supported
6.0	2017-11-14[67]	6.0.1	2017-12-06[68]	Still supported
6.1	2017-12-12[69]	6.1.3	2018-01-16[70]	Still supported
6.2	2018-02-06[71]	6.2.2	2018-02-20[1]	Latest

Legend: Old version | Older version, still supported | **Latest version** | Latest preview version

Elasticsearch 7.4.1

October 23, 2019

O systemie

- Baza danych (nierelacyjna), która do wyszukiwania wykorzystuje bibliotekę Apache Lucene 
- Lucene
- Projekt Open Source
  - Wysoce wydajna biblioteka napisana pierwotnie w czystej Javie do pełnotekstowego wyszukiwania i indeksowania tekstów (indexing and search technology). Daje również wsparcie dla zadań sprawdzania pisowni, podświetlania odszukanych tekstów, bardziej zaawansowanej analityki, tokenizacji tekstów itp.
  - Scalable, High-Performance Indexing
    - over 150GB/hour on modern hardware
    - small RAM requirements - only 1MB heap
    - incremental indexing as fast as batch indexing
    - index size roughly 20-30% the size of text indexed

O systemie

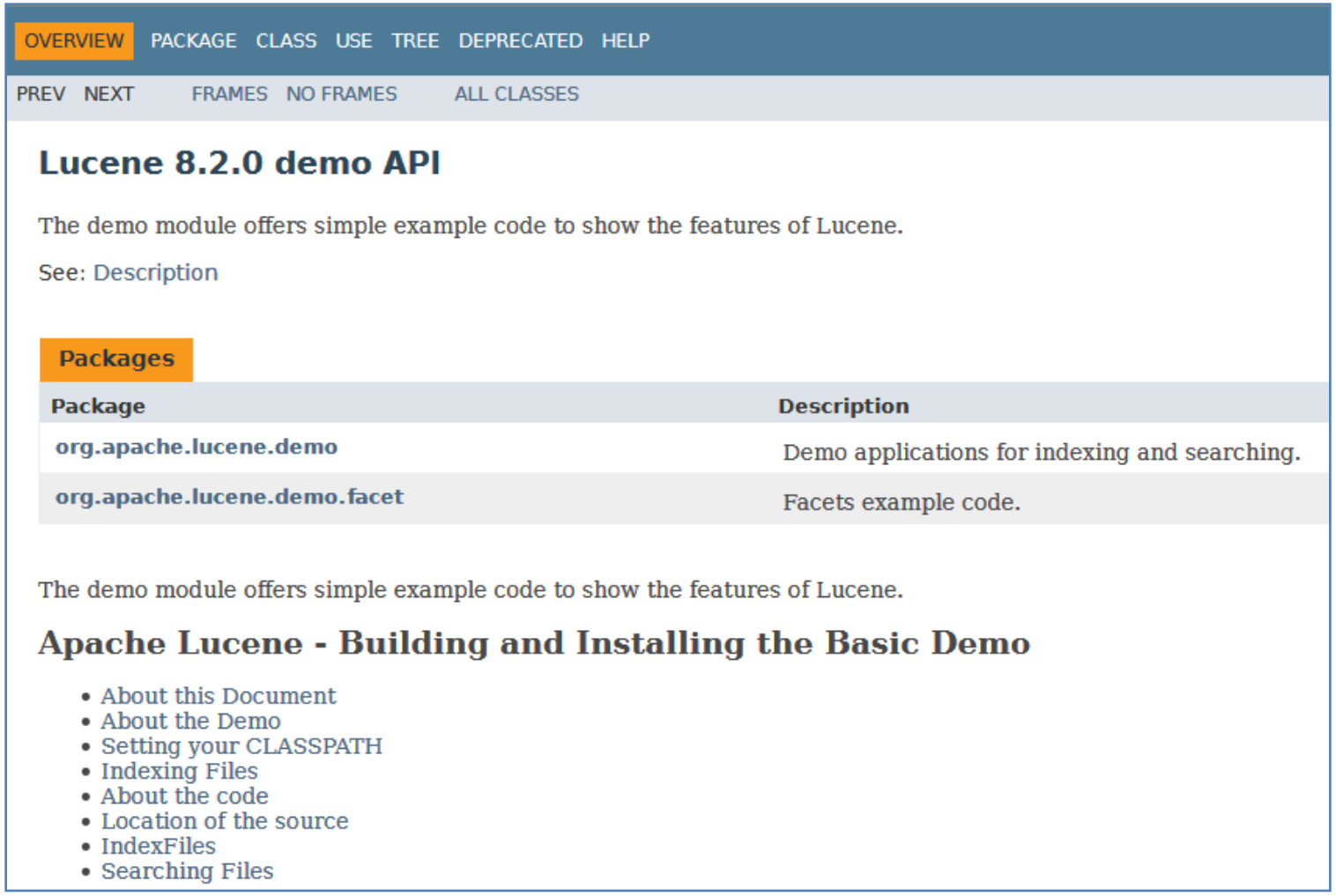
- Lucene
  - Powerful, Accurate and Efficient Search Algorithms
    - ranked searching - best results returned first
    - many powerful query types: phrase queries, wildcard queries, proximity queries, range queries and more
    - fielded searching (e.g. title, author, contents)
    - sorting by any field
    - multiple-index searching with merged results
    - allows simultaneous update and searching
    - flexible faceting, highlighting, joins and result grouping
    - fast, memory-efficient and typo-tolerant suggesters
    - pluggable ranking models, including the Vector Space Model and Okapi BM25
    - configurable storage engine (codecs)

O systemie

- Lucene
  - Cross-Platform Solution
    - available as Open Source software under the Apache License which lets you use Lucene in both commercial and Open Source programs
    - 100%-pure Java
    - implementations in other programming languages available that are index-compatible (Lucene has been ported to other programming languages including Object Pascal, Perl, C#, C++, Python, Ruby and PHP)

O systemie

- Lucene, demo API
  https://lucene.apache.org/core/8_2_0/demo/overview-summary.html#overview_description



OVERVIEW PACKAGE CLASS USE TREE DEPRECATED HELP

PREV NEXT FRAMES NO FRAMES ALL CLASSES

Lucene 8.2.0 demo API

The demo module offers simple example code to show the features of Lucene.

See: Description

Packages

Package	Description
org.apache.lucene.demo	Demo applications for indexing and searching.
org.apache.lucene.demo.facet	Facets example code.

The demo module offers simple example code to show the features of Lucene.

Apache Lucene - Building and Installing the Basic Demo

- About this Document
- About the Demo
- Setting your CLASSPATH
- Indexing Files
- About the code
- Location of the source
- IndexFiles
- Searching Files

O systemie

- Projekty oparte o Lucene (na podstawie https://en.wikipedia.org/wiki/Apache_Lucene)
  - Apache Nutch — provides web crawling and HTML parsing
  - Apache Solr — an enterprise search server
  - Compass — the predecessor to Elasticsearch
  - CrateDB — open source, distributed SQL database built on Lucene
  - DocFetcher — a multiplatform desktop search application
  - Elasticsearch — an enterprise search server
  - Kinosearch — a search engine written in Perl and C
  - Swiftype — an enterprise search startup based on Lucene
  - Twitter is using Lucene for its real time search
- Warto też zapoznać się z sekcją Company → Customers na stronie WWW projektu

O systemie

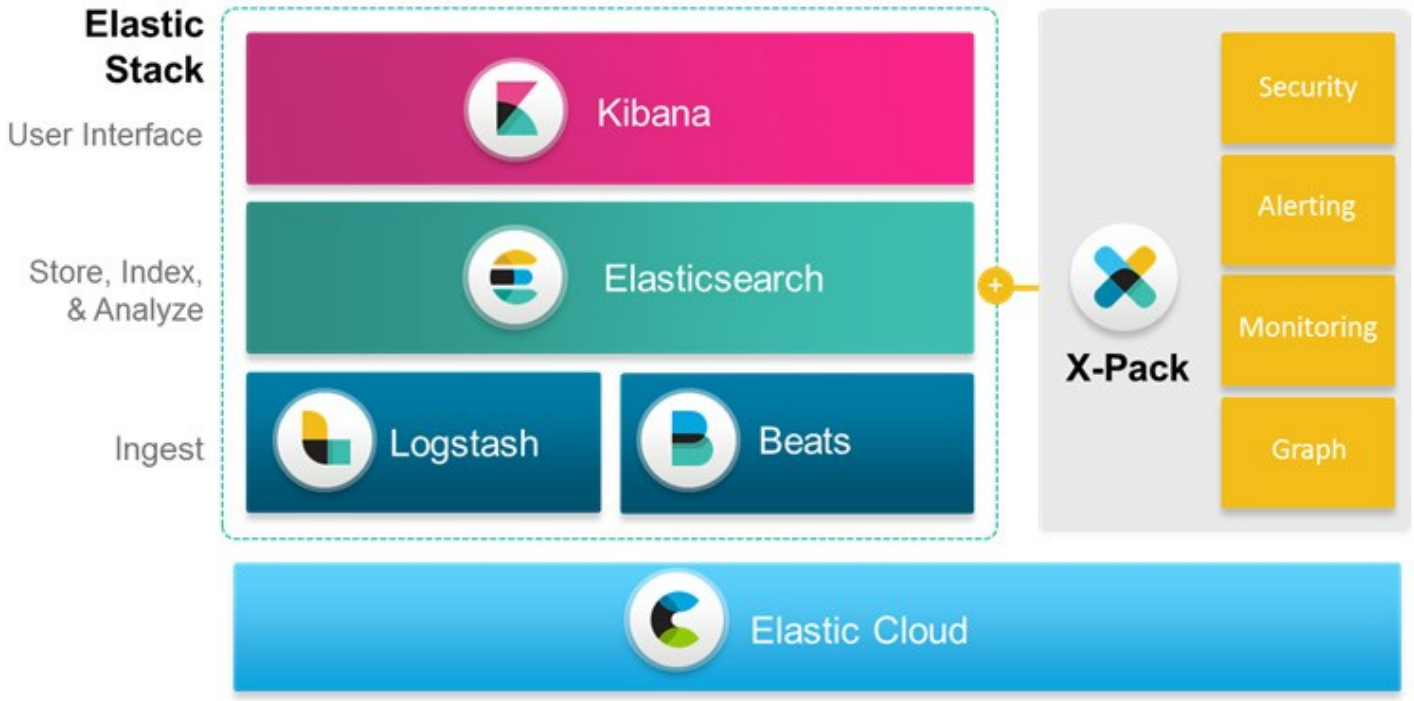
- Bardzo dobrze wpisuje się w nurt BigData
  - potrzeba obsługi dużych ilości danych
  - może działać w architekturze rozproszonej skierowanej na usługi (mikroserwisy)
  - REST-owe API
  - wspiera praktycznie wszystkie etapy analizy danych
    - zbieranie danych z różnych źródeł
    - wstępna obróbka danych
    - wyszukiwanie / przeszukiwanie danych
    - analizy
- Powstał rodzaj ekosystemu z kilkoma najważniejszymi projektami oraz bardzo wieloma rozszerzeniami, często tworzonych przez niezależnych dostawców
  - Elasticsearch
  - Kibana
  - Logstash
  - Beats
  - X-Pack

O systemie

- Nie jest monolitem, ale zbiorem współpracujących komponentów
- Jest doskonałym przykładem, jakie korzyści daje architektura mikroserwisów
- Komponenty mogą być wykorzystane osobno lub łącznie, co znakomicie zwiększa ilość potencjalnych zastosowań
  - dobrym przykładem jest **Logstash**. Powstał z myślą o zbieraniu, przetwarzaniu i dostarczaniu logów do Elasticsearch
- Near Realtime (NRT)
  - w praktyce oznacza to, że jest minimalne opóźnienie pomiędzy zaindeksowanie dokumentu a staniem się dokumentu przeszukiwalnym, jednak ten czas liczony jest maksymalnie w sekundach

O systemie

- Elastic Stack

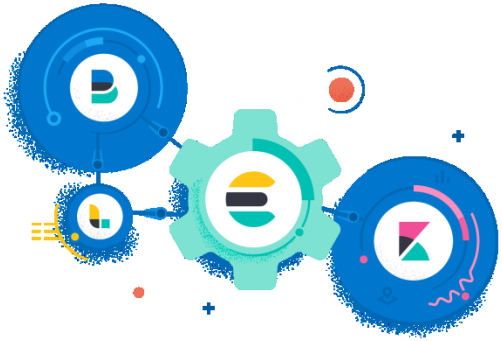


Bazy relacyjne a Elasticsearch – porównanie terminologii

Baza relacyjna	Elasticsearch
Database	Index
Table	Type
Row	Document
Column	Field
Primary Key	_id
Index	Analyze
Schema	Mapping
Physical Partition	Shard
SQL	Query DSL (Domain-Specific Language)
SELECT * FROM table	GET
UPDATE table SET	PUT
INSERT INTO table	PUT/POST
DELETE FROM table	DELETE

Komponenty – Elasticsearch

- Rozproszony system open-source do gromadzenia, przeszukiwania, analizowania wielkich ilości danych tekstowych
  - "almost real-time"
- Zbudowany w oparciu o wzorzec projektowy REST (Representational State Transfer; zmiana stanu poprzez reprezentacje) oraz format JSON
  - dzięki temu aplikacje klienckie mogą być tworzone praktyczne w każdej technologii (JAVA, C#, Pyton, JavaScript, PHP, Perl, Ruby itd.)
  - http://localhost:9200/
- Jest centralnym elementem Elastic Stack



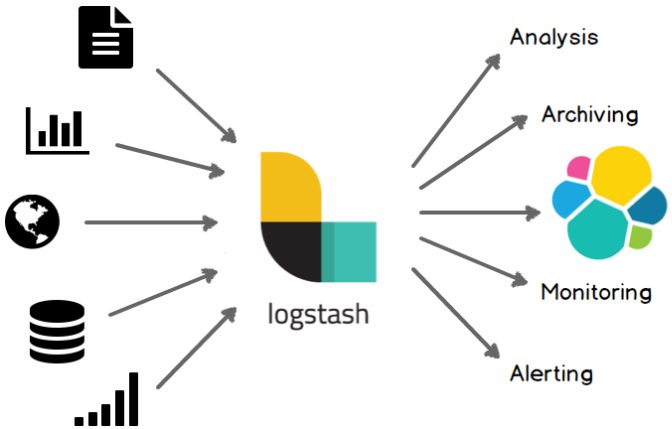
- Jest bardzo wydajny
  - przykładowo wszystkie kody zgromadzone w GitHub są zaindeksowane w Elasticsearch

Komponenty – Kibana

- Moduł open-source do analityki i wizualizacji danych przechowywanych w Elasticsearch
- Uruchamiany z poziomy przeglądarki (http://localhost:5601)
  - w zasadzie moduł bezkonfiguracyjny
- Bardzo rozbudowane możliwości graficznej prezentacji danych
  - w tym popularne obecnie tzw. dynamiczne kokpity informacyjne (ang. dynamic dashboards)
  - działają w trybie niemal rzeczywistym

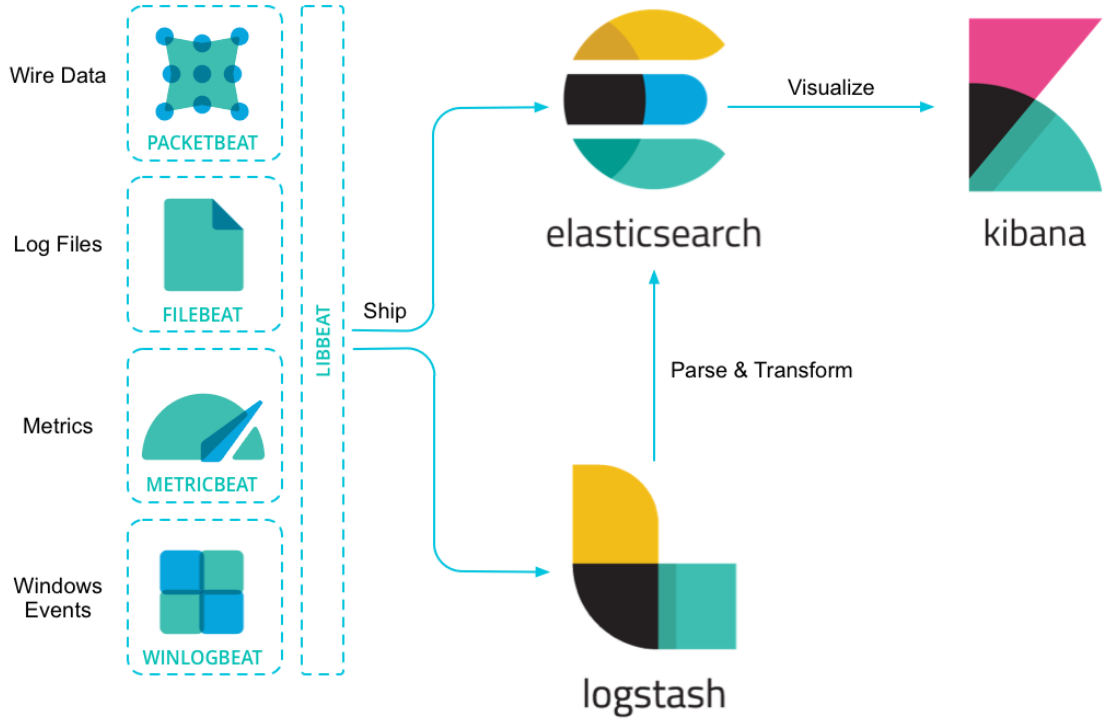
Komponenty – Logstash

- Moduł open-source umożliwiający łączenie wielu strumieni danych wejściowych, filtrowanie każdego strumienia i przekazywania ich do Elasticsearch
  - stash – chomikować, kryjówka, ukryte zapasy
  - collection engine with real-time pipelining capabilities
- Ma możliwość unifikacji danych z różnych źródeł
- Działanie Logstash opiera się na pliku konfiguracyjnym zawierającym opis tego co chcemy by Logstash robił



Komponenty – Beats

- Zbiór modułów wysyłających (ang. shippers) pobrane z różnych źródeł dane do Elasticsearch
  - pobrane dane mogą być wysyłane albo bezpośrednio do Elasticsearch albo do modułu "normującego" Logstash
  - przykładowe moduły: Packetbeat, Filebeat, Metricbeat, Winlogbeat

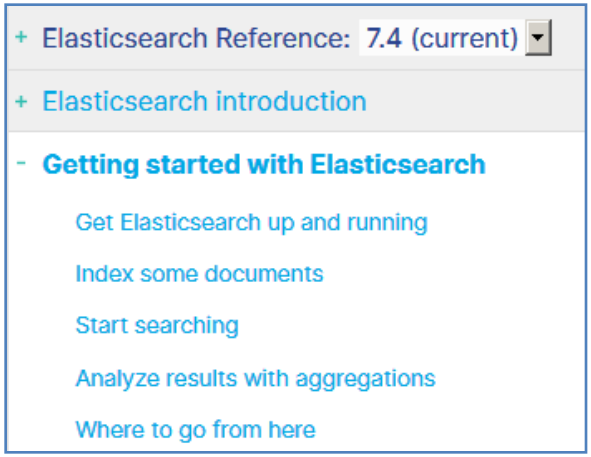


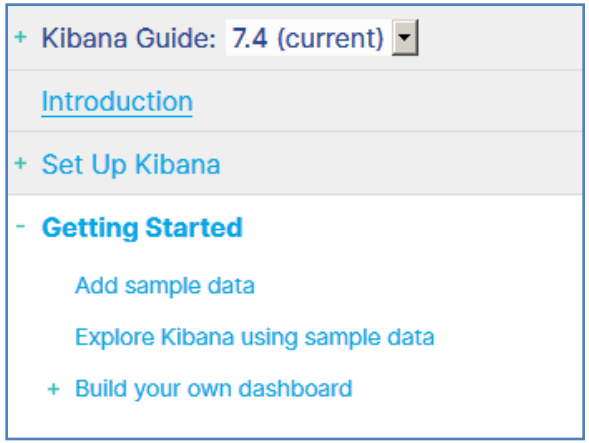
X-Pack

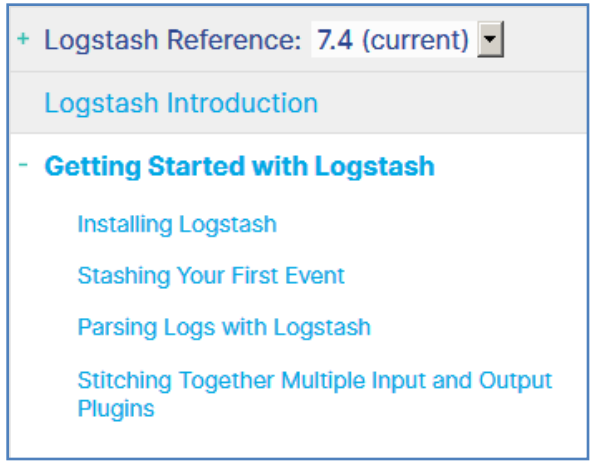
- Rozszerzenia (komponenty) do Elasticsearch implementujące dodatkową funkcjonalność
  - bezpieczeństwo, autoryzowany dostęp
  - monitorowanie, alerty, powiadomienia
  - raportowanie
  - grafika
  - Machine Learning
- Do wersji 5.x były oddzielne moduły, które trzeba było niezależnie instalować, konfigurować. Często też pojawiały się konflikty spowodowane niedopasowaniem wersji. X-Pack łączy to wszystko w jeden pakiet

Dokumentacja, Getting Started

- https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html
- https://www.elastic.co/guide/en/kibana/current/getting-started.html
- https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html

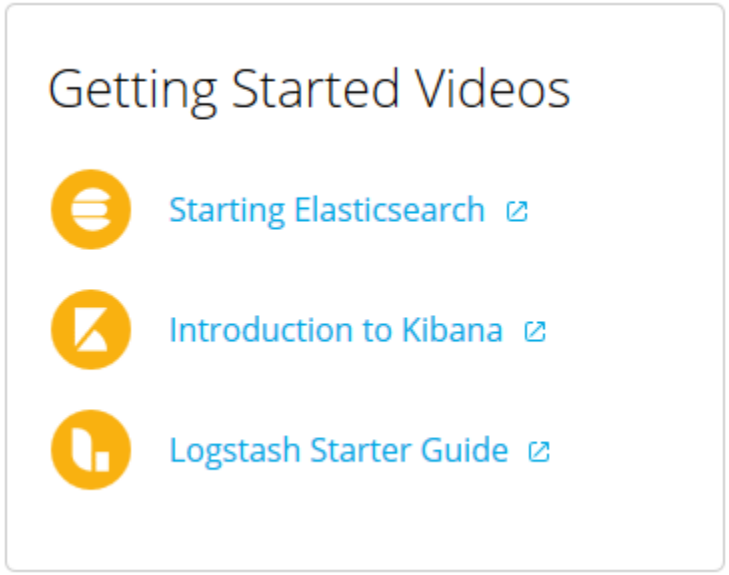






Dokumentacja, Getting Started

- Getting Started Videos



Budowa Elasticsearch

- **Node**
  - tą nazwą określamy pojedynczy serwer. Każdy serwer posiada swoją nazwę oraz port na którym działa, standardowy port to 9200. To tutaj następuje przetwarzanie oraz przeszukiwanie danych
- **Cluster**
  - jest zbiorem jednego lub więcej node-ów
- **Index**
  - to kolekcja dokumentów o zbliżonej charakterystyce. Nazwa indeksu jest kluczowym elementem, gdyż to na jej podstawie odwołujemy się do określonego indeksu i wyszukujemy dokumenty czy też dodajemy / usuwamy dokumenty. Odpowiednikiem indeksu w relacyjnej bazie danych jest tabela

Źródło: http://www.czterytygodnie.pl/wprowadzenie-do-elasticsearch

Budowa Elasticsearch

- **Document**
  - dokumenty są rekordami, które dodajemy tak, jak byśmy dodawali rekordy do tabeli w relacyjnej bazie danych. Przy czym dokumenty są zapisywane w formacie JSON (JavaScript Object Notation)

```
{
"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}
}
```

Budowa Elasticsearch

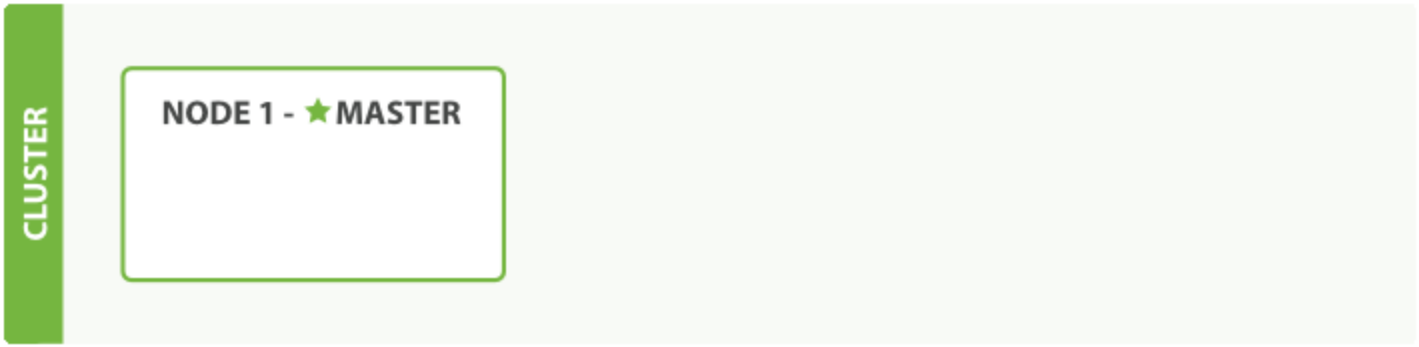
- **Shards (pol. odłamek, okruch, kawałek)**
  - mechanizm pozwalający dzielić indeks na mniejsze części zwane shard-ami
  - każdy shard stanowi samodzielny i niezależny indeks, który może być utworzony na dowolnym serwerze (node-ie)
  - poprawa wydajności, skalowanie horyzontalne, zrównoleglanie działania
  - mechanizm shard-ów jest całkowicie "przeźroczysty" dla użytkownika
- **Repliki (Replicas)**
  - Mechanizm pozwalający na wykonywanie kopii (replik) poszczególnych shard-ów
  - poprawa bezpieczeństwa ale też i wydajności (repliki mogą być przeszukiwane równolegle)

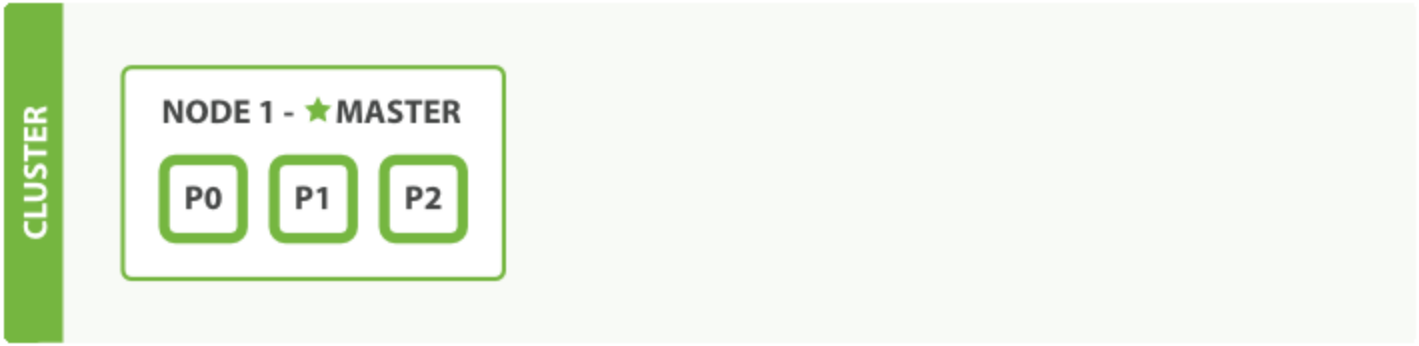
Budowa Elasticsearch

- Shards, Replicas, założenia, zasada działania
  - liczbę shard-ów oraz replik określa się w memencie definiowania indeksów
  - po zdefiniowaniu indeksu można zmieniać dynamicznie ilość replik
  - po zdefiniowaniu indeksu nie można już zmieniać ilości shard-ów
  - po zreplikowaniu, każdy indeksu będzie posiadał tzw. shard-y podstawowe (primary shards), czyli te, które były użyte do wykonania replik oraz shard-y zreplikowane (kopie shard-ów podstawowych)
  - domyślnie każdy indeks alokuje 5 shard-ów oraz powstaje 1 replika
    - oznacza to, że gdy posiadamy przynajmniej 2 węzły (nodes) w klastrze, to nasz indeks będzie posiadał 5 shard-ów podstawowych i 5 shard-ów zreplikowanych (czyli 1 komopletna replika). Czyli w sumie będzie 10 shard-ów na indeks
  - każdy shard może zwierać maksymalnie 2.147.483.519 (= Integer.MAX_VALUE - 128) dokumentów

Skalowanie horyzontalne - przykłady

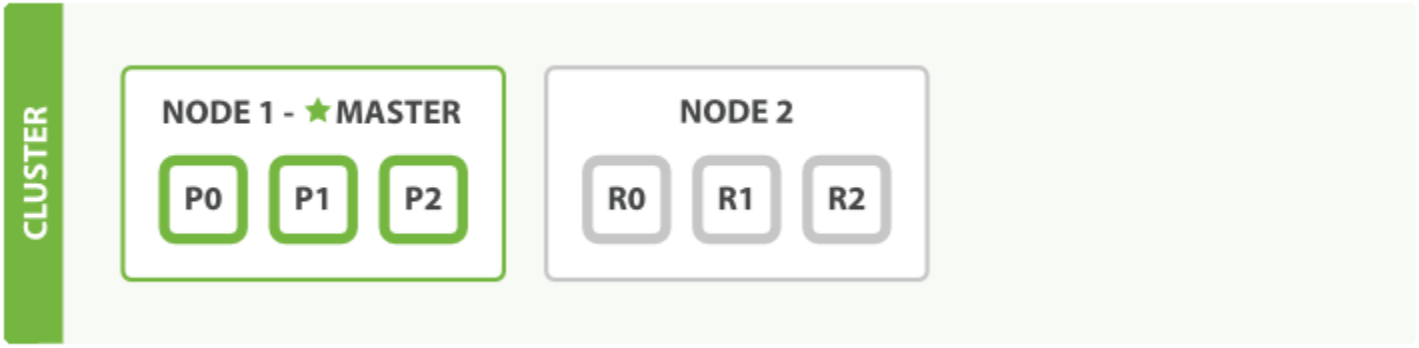
- Klaster tworzy 1 serwer (node), gdzie utworzono trzy podstawowe shardy (primary shards)



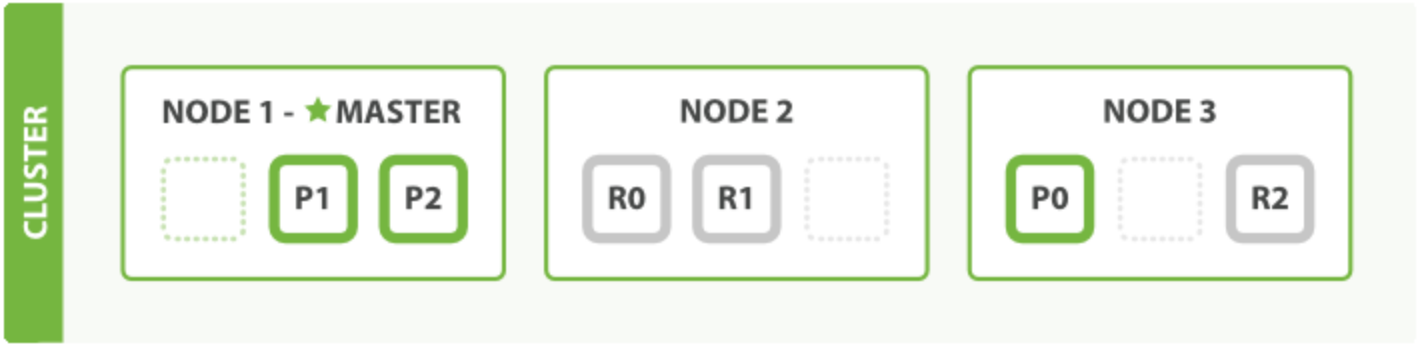


Skalowanie horyzontalne - przykłady

- Klaster tworzą 2 serwery (nodes). Trzy podstawowe shardy są

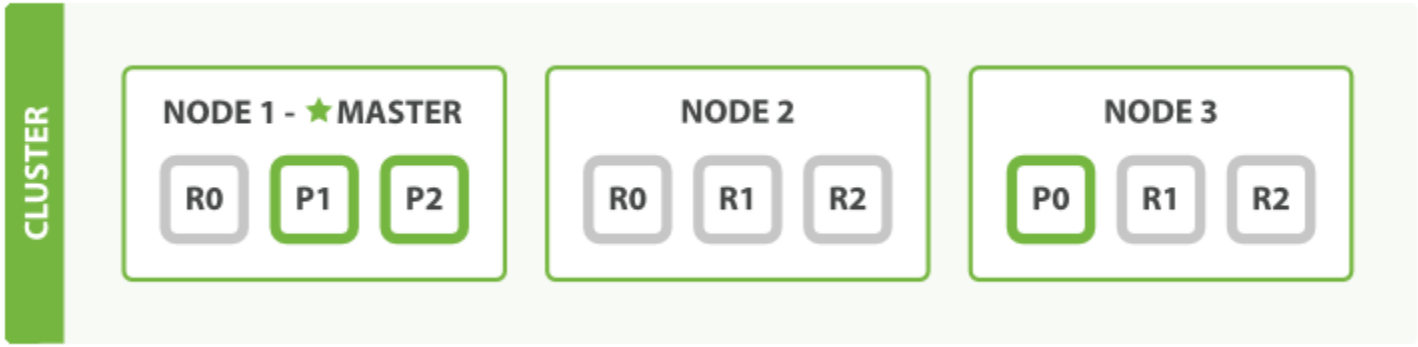


- Klaster tworzą 3 serwery (nodes). Po jednym shard-ie z dwóch pierwszych serwerów zostało realokowanych do serwera 3,

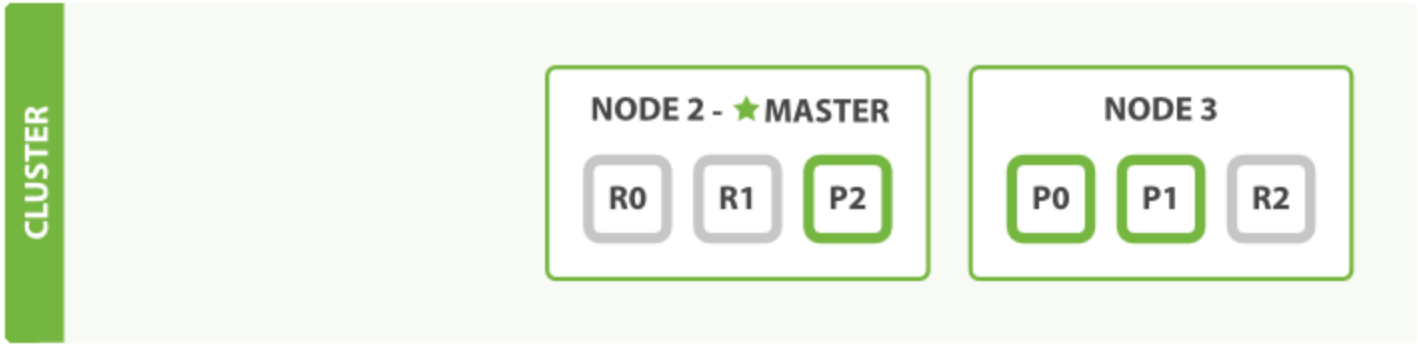


Skalowanie horyzontalne - przykłady

- Zwiększono parametr `number_of_replicas` do 2

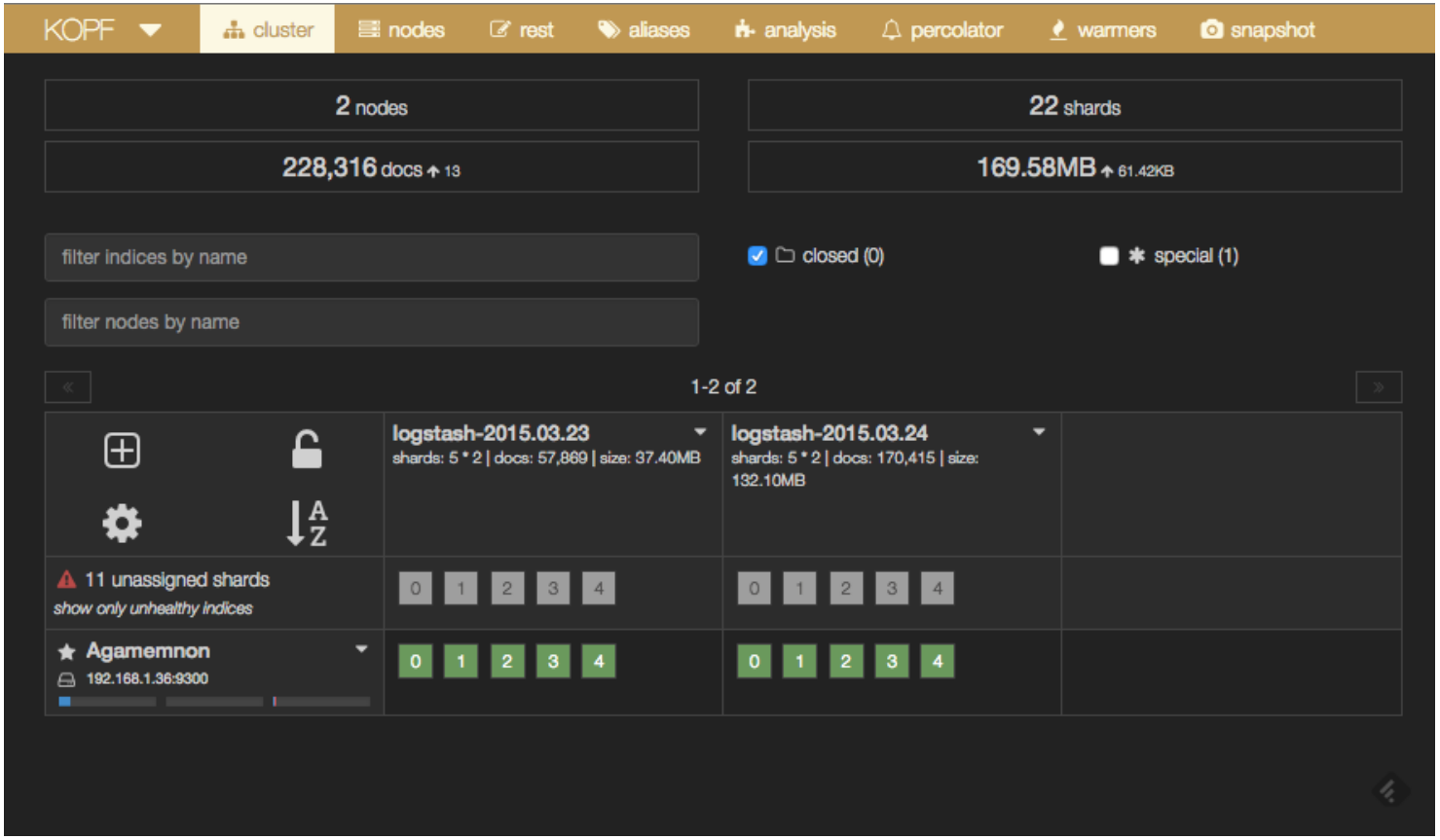


- Struktura klastra po awarii jednego serwera



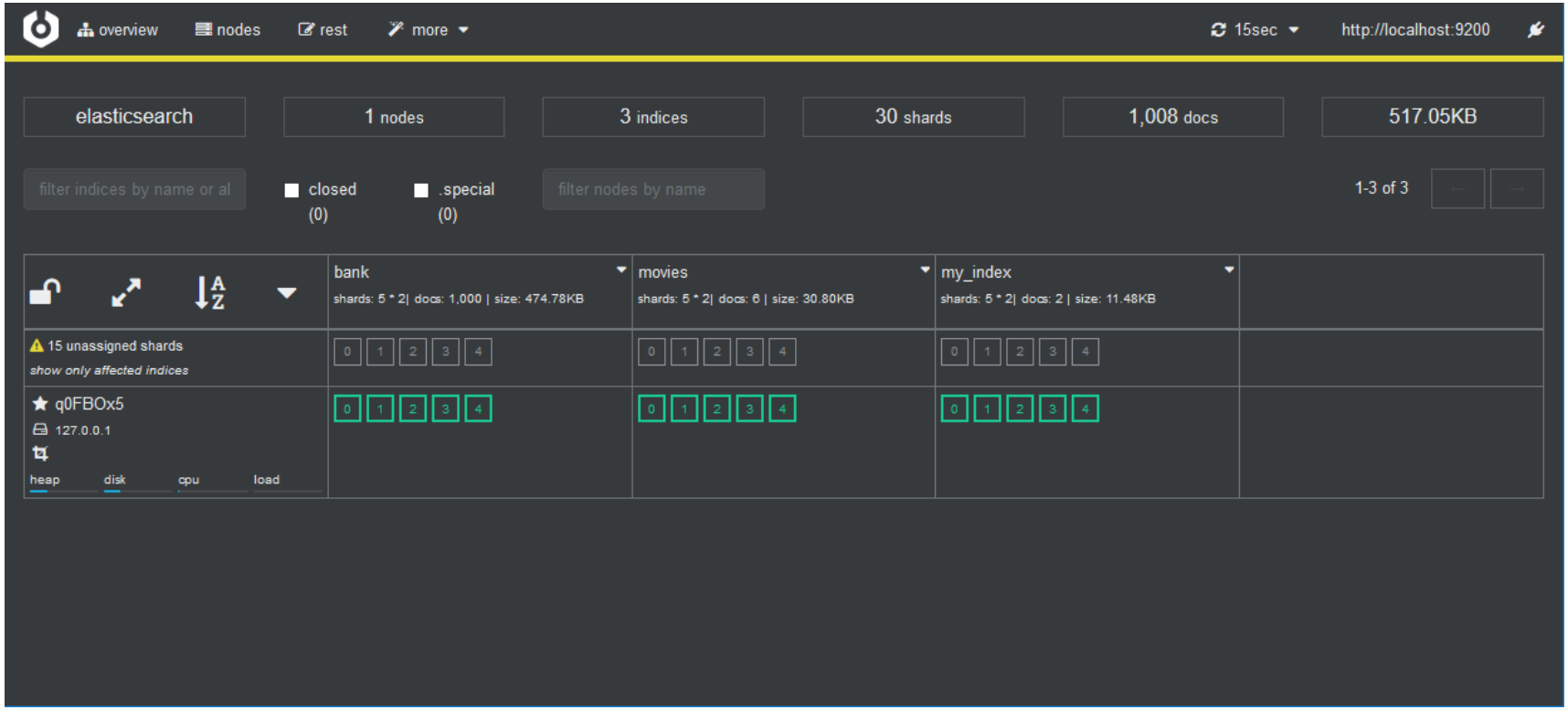
Skalowanie horyzontalne - oprogramowanie

- KOPF



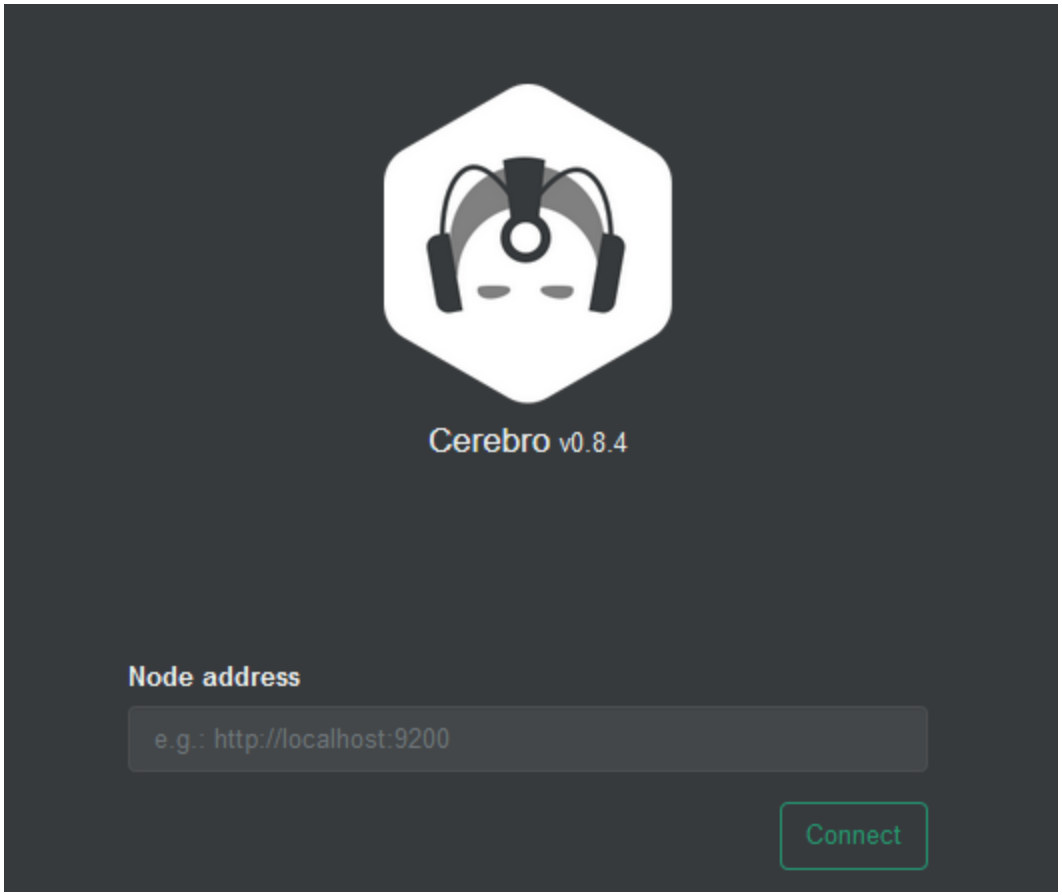
Skalowanie horyzontalne - oprogramowanie

- Uwaga: oprogramowanie dla Elasticsearch bardzo szybko się zmienia. Przykładowo KOPF (poprzedni slajd) nie jest już rozwijany i zastąpiło go oprogramowanie o nazwie **Cerebro**
  - https://github.com/lmenezes/cerebro



Cerebro

- http://localhost:9000



Instalacja

- Wymaga Javy w wersji przynajmniej 8
- ZIP, TAR, DEB, RPM, MSI, Doker
  - MSI wymaga przynajmniej .NET 4.5 framework
- ./bin/elasticsearch
- $ES_HOME/config/elasticsearch.yml
  - każde ustawienie parametru konfiguracyjnego w pliku można dokonać z linii poleceń, np.
    /bin/elasticsearch -d -Ecluster.name=my_cluster -Enode.name=node_1

Instalacja

Type	Description	Default Location	Setting
home	Elasticsearch home directory or `$ES_HOME`	Directory created by unpacking the archive	
bin	Binary scripts including `elasticsearch` to start a node and `elasticsearch-plugin` to install plugins	`$ES_HOME/bin`	
conf	Configuration files including `elasticsearch.yml`	`$ES_HOME/config`	`ES_PATH_CONF`
data	The location of the data files of each index / shard allocated on the node. Can hold multiple locations.	`$ES_HOME/data`	`path.data`
logs	Log files location.	`$ES_HOME/logs`	`path.logs`
plugins	Plugin files location. Each plugin will be contained in a subdirectory.	`$ES_HOME/plugins`	
repo	Shared file system repository locations. Can hold multiple locations. A file system repository can be placed in to any subdirectory of any directory specified here.	Not configured	`path.repo`
script	Location of script files.	`$ES_HOME/scripts`	`path.scripts`

Instalacja

- program cURL
  - sieciowa biblioteka programistyczna. Umożliwia wysyłanie zapytań HTTP, w tym pobieranie z serwerów stron i plików, a także wysyłanie treści formularzy
  - cURL obsługuje protokoły DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, POP3, POP3S, RTMP, RTSP, SCP, SFTP, SMTP, SMTPS, Telnet oraz TFTP. Wspiera także mechanizmy takie jak: certyfikaty SSL, HTTP POST, HTTP PUT, upload FTP, wysyłanie formularzy HTTP, serwery proxy, HTTP cookie, Uwierzytelnianie (użytkownik+hasło), wznawiania transferu plików, tunelowanie proxy HTTP oraz wiele innych
  - cURL jest dostępny dla następujących języków programowania i bibliotek: Ada95, Basic, C, C++, Ch, Cocoa, D, Dylan, Eiffel, Euphoria, Falcon, Ferite, Gambas, GTK+, Haskell, Java, Lisp, Lua, Mono, .NET, Object Pascal, OCaml, Pascal, Perl, PHP, Postgres, Python, R, Rexx, Ruby, RPG, Scheme, S-Lang, Smalltalk, SP-Forth, SPL, Tcl, Visual Basic, Visual FoxPro, Q, wxWidgets, XBLite.

Instalacja

- Program cURL
  - w Windows problemy z wykonywaniem kilkulinijkowych komend. To nie zadziała

```
curl -XGET localhost:9200/library/book/_search?pretty=true -d {
   "query" : {
      "query_string" : { "query" : "title:crime" }
   }
}
```

  - trzeba "kombinować", ale efekt jest taki sobie

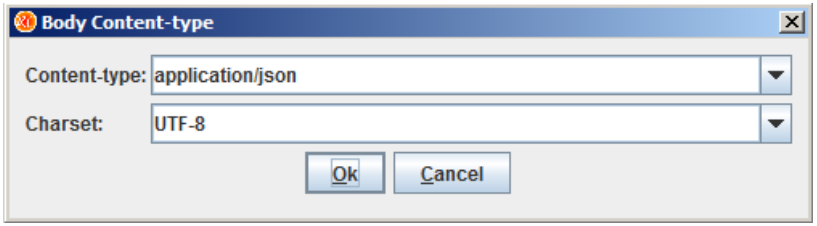
```
curl –XGET localhost:9200/library/book/_search?pretty=true -d "{
\"query\" : { \"query_string\" : { \"query\" : \"title:crime\" } } }"
```

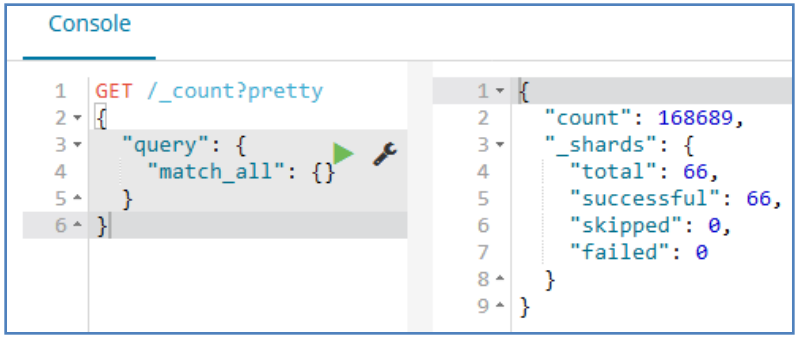
  - Wniosek: cURL pod Windows trzeba zastąpić jakimś innym programem / nakładką graficzną, np.
    - https://github.com/wiztools/rest-client
    - Fiddler
    - Postman

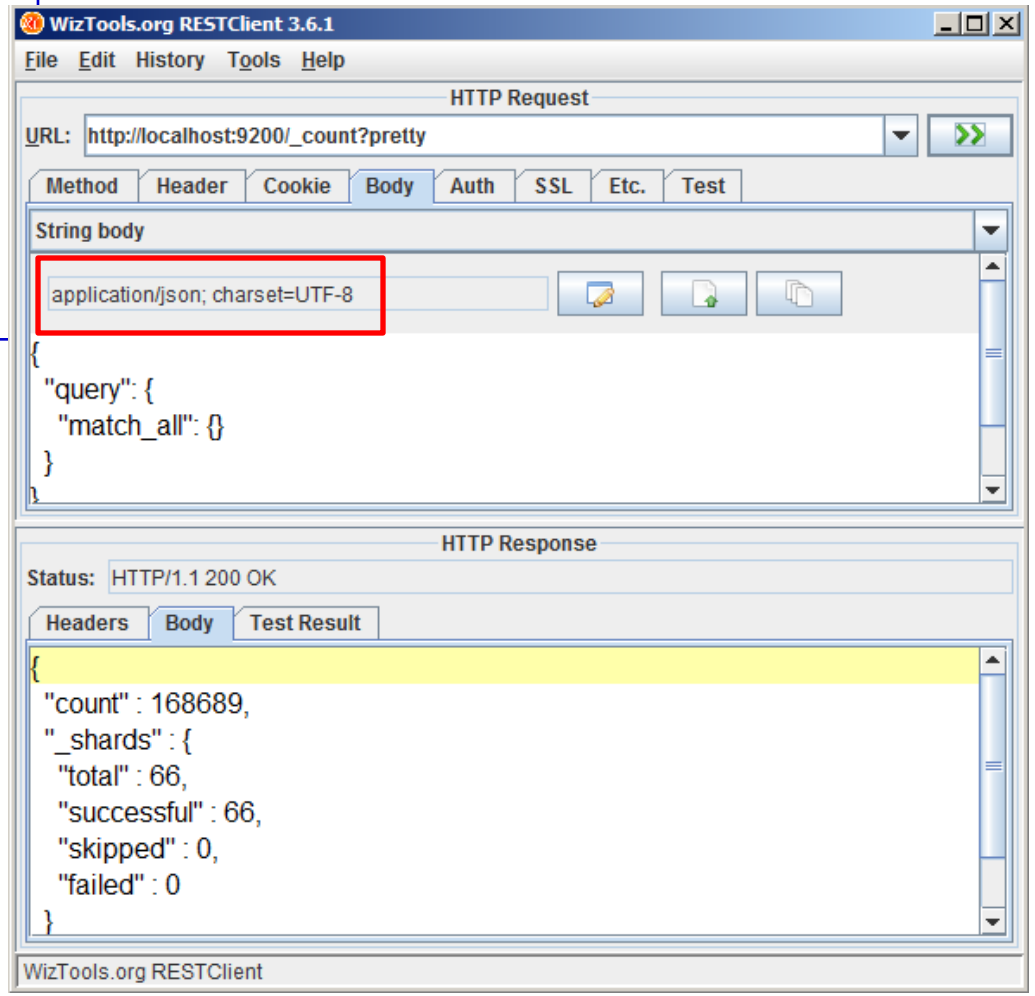
Instalacja

- Program WizTools.org REST Client

```
curl -XGET "http://localhost:9200/_count?pretty"
-H 'Content-Type: application/json' -d'
{
  "query": {
    "match_all": {}
  }
}'
```







Konfiguracja

- Ustawienia domyślne są bardzo dobre (przynajmniej na etapie zapoznawania się z programem)
- Pliki konfiguracyjne
  - elasticsearch.yml – konfiguracja lasticsearch
  - jvm.options – konfiguracja JVM
  - log4j2.properties – konfiguracja logów

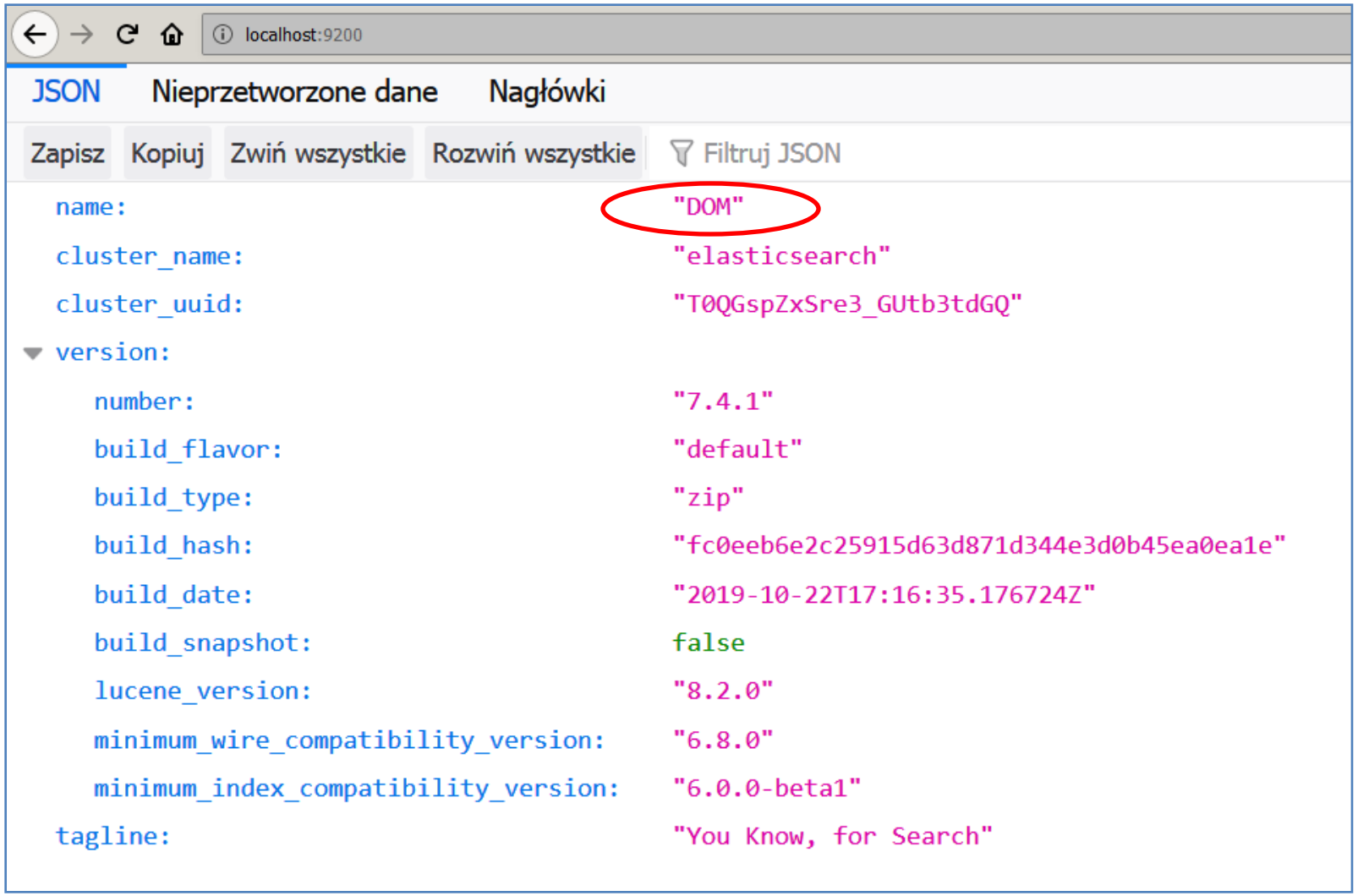
Pierwsze uruchomienie

```
c:\Program Files\Elastic\elasticsearch-7.4.1\bin>elasticsearch
OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a futur
[2019-10-31T00:39:44,064][INFO ][o.e.e.NodeEnvironment    ] [DOM] using [1] data paths, mounts [[(c:)]], net usable_space [93.8
58.9gb], types [NTFS]
[2019-10-31T00:39:44,082][INFO ][o.e.e.NodeEnvironment    ] [DOM] heap size [990.7mb], compressed ordinary object pointers [tru
[2019-10-31T00:39:44,099][INFO ][o.e.n.Node               ] [DOM] node name [DOM], node ID [2ukw8TNCQ9-feXZhw5KtxQ], cluster na
[2019-10-31T00:39:44,103][INFO ][o.e.n.Node               ] [DOM] version[7.4.1], pid[4944], build[default/zip/fc0eeb6e2c25915d
e/2019-10-22T17:16:35.176724Z], OS[Windows 7/6.1/amd64], JVM[AdoptOpenJDK/OpenJDK 64-Bit Server VM/13/13+33]
[2019-10-31T00:39:44,111][INFO ][o.e.n.Node               ] [DOM] JVM home [c:\Program Files\Elastic\elasticsearch-7.4.1\jdk]
[2019-10-31T00:39:44,116][INFO ][o.e.n.Node               ] [DOM] JVM arguments [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:C
action=75, -XX:+UseCMSInitiatingOccupancyOnly, -Des.networkaddress.cache.ttl=60, -Des.networkaddress.cache.negative.ttl=10, -XX
m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true,
imization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dio.netty.allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled
e.jmx=true, -Djava.io.tmpdir=C:\Users\ADMINI~1\AppData\Local\Temp\elasticsearch, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpP
=logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,pid,tags:filecount=32,filesize=64m, -Djava.lo
-Dio.netty.allocator.type=unpooled, -XX:MaxDirectMemorySize=536870912, -Delasticsearch, -Des.path.home=c:\Program Files\Elastic
Des.path.conf=c:\Program Files\Elastic\elasticsearch-7.4.1\config, -Des.distribution.flavor=default, -Des.distribution.type=zip
]
[2019-10-31T00:39:49,130][INFO ][o.e.p.PluginsService     ] [DOM] loaded module [aggs-matrix-stats]
[2019-10-31T00:39:49,134][INFO ][o.e.p.PluginsService     ] [DOM] loaded module [analysis-common]
[2019-10-31T00:39:49,142][INFO ][o.e.p.PluginsService     ] [DOM] loaded module [data-frame]
[2019-10-31T00:39:49,146][INFO ][o.e.p.PluginsService     ] [DOM] loaded module [flattened]
[2019-10-31T00:39:49,150][INFO ][o.e.p.PluginsService     ] [DOM] loaded module [frozen-indices]
```

Pierwsze uruchomienie

```
[2019-10-31T00:39:49,283][INFO ][o.e.p.PluginsService     ] [DOM] loaded module [x-pack-watcher]
[2019-10-31T00:39:49,287][INFO ][o.e.p.PluginsService     ] [DOM] no plugins loaded
[2019-10-31T00:39:57,428][INFO ][o.e.x.s.a.s.FileRolesStore] [DOM] parsed [0] roles from file [c:\Program Files\Elastic\elasticsearch-7.
es.yml]
[2019-10-31T00:39:58,617][INFO ][o.e.x.m.p.l.CppLogMessageHandler] [DOM] [controller/2936] [Main.cc@110] controller (64 bit): Version 7.
880bdacc5e8) Copyright (c) 2019 Elasticsearch BV
[2019-10-31T00:39:59,818][DEBUG][o.e.a.ActionModule       ] [DOM] Using REST wrapper from plugin org.elasticsearch.xpack.security.Securi
[2019-10-31T00:40:00,622][INFO ][o.e.d.DiscoveryModule    ] [DOM] using discovery type [zen] and seed hosts providers [settings]
[2019-10-31T00:40:02,575][INFO ][o.e.n.Node               ] [DOM] initialized
[2019-10-31T00:40:02,580][INFO ][o.e.n.Node               ] [DOM] starting ...
[2019-10-31T00:40:03,445][INFO ][o.e.t.TransportService   ] [DOM] publish_address {127.0.0.1:9300}, bound_addresses {127.0.0.1:9300}, {[
[2019-10-31T00:40:03,519][WARN ][o.e.b.BootstrapChecks    ] [DOM] the default discovery settings are unsuitable for production use; at l
iscovery.seed_hosts, discovery.seed_providers, cluster.initial_master_nodes] must be configured
[2019-10-31T00:40:03,537][INFO ][o.e.c.c.Coordinator      ] [DOM] cluster UUID [T0QGspZxSre3_GUtb3tdGQ]
[2019-10-31T00:40:03,558][INFO ][o.e.c.c.ClusterBootstrapService] [DOM] no discovery configuration found, will perform best-effort clust
ng after [3s] unless existing master is discovered
[2019-10-31T00:40:03,862][INFO ][o.e.c.s.MasterService    ] [DOM] elected-as-master ([1] nodes joined)[{DOM}{2ukw8TNCQ9-feXZhw5KtxQ}{oJh
dT2sw}{127.0.0.1}{127.0.0.1:9300}{dilm}{ml.machine_memory=6441984000, xpack.installed=true, ml.max_open_jobs=20} elect leader, _BECOME_M
FINISH_ELECTION_], term: 2, version: 18, reason: master node changed {previous [], current [{DOM}{2ukw8TNCQ9-feXZhw5KtxQ}{oJhAQl2VSsidgi
0.0.1}{127.0.0.1:9300}{dilm}{ml.machine_memory=6441984000, xpack.installed=true, ml.max_open_jobs=20}]}
[2019-10-31T00:40:04,051][INFO ][o.e.c.s.ClusterApplierService] [DOM] master node changed {previous [], current [{DOM}{2ukw8TNCQ9-feXZhw
VSsidgiCyMdT2sw}{127.0.0.1}{127.0.0.1:9300}{dilm}{ml.machine_memory=6441984000, xpack.installed=true, ml.max_open_jobs=20}]}, term: 2, v
ason: Publication{term=2, version=18}
[2019-10-31T00:40:04,397][INFO ][o.e.h.AbstractHttpServerTransport] [DOM] publish_address {127.0.0.1:9200}, bound_addresses {127.0.0.1:9
200}
[2019-10-31T00:40:04,404][INFO ][o.e.n.Node               ] [DOM] started
[2019-10-31T00:40:04,548][INFO ][o.e.l.LicenseService     ] [DOM] license [65c2ceca-7686-4b42-8f0e-908c5c57fbd0] mode [basic] - valid
[2019-10-31T00:40:04,553][INFO ][o.e.x.s.s.SecurityStatusChangeListener] [DOM] Active license is now [BASIC]; Security is disabled
[2019-10-31T00:40:04,580][INFO ][o.e.g.GatewayService     ] [DOM] recovered [0] indices into cluster_state
```

Pierwsze uruchomienie



Uruchomienie w Windows jako serwis

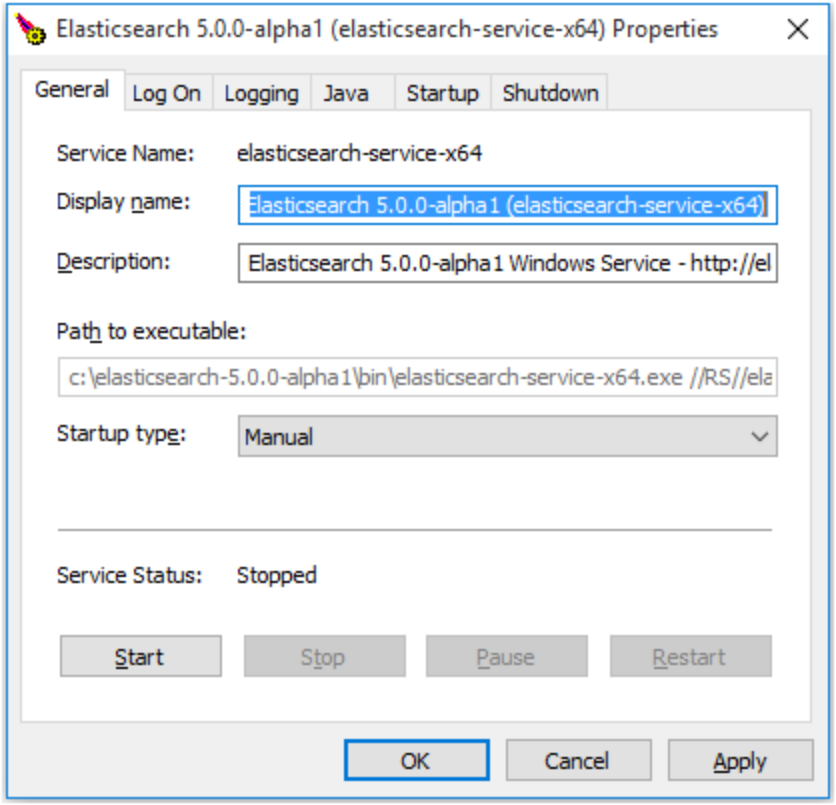
```
c:\elasticsearch-6.2.2\bin>elasticsearch-service.bat

Usage: elasticsearch-service.bat install|remove|start|stop|manager [SERVICE_ID]
```

`install`	Install Elasticsearch as a service
`remove`	Remove the installed Elasticsearch service (and stop the service if started)
`start`	Start the Elasticsearch service (if installed)
`stop`	Stop the Elasticsearch service (if started)
`manager`	Start a GUI for managing the installed service

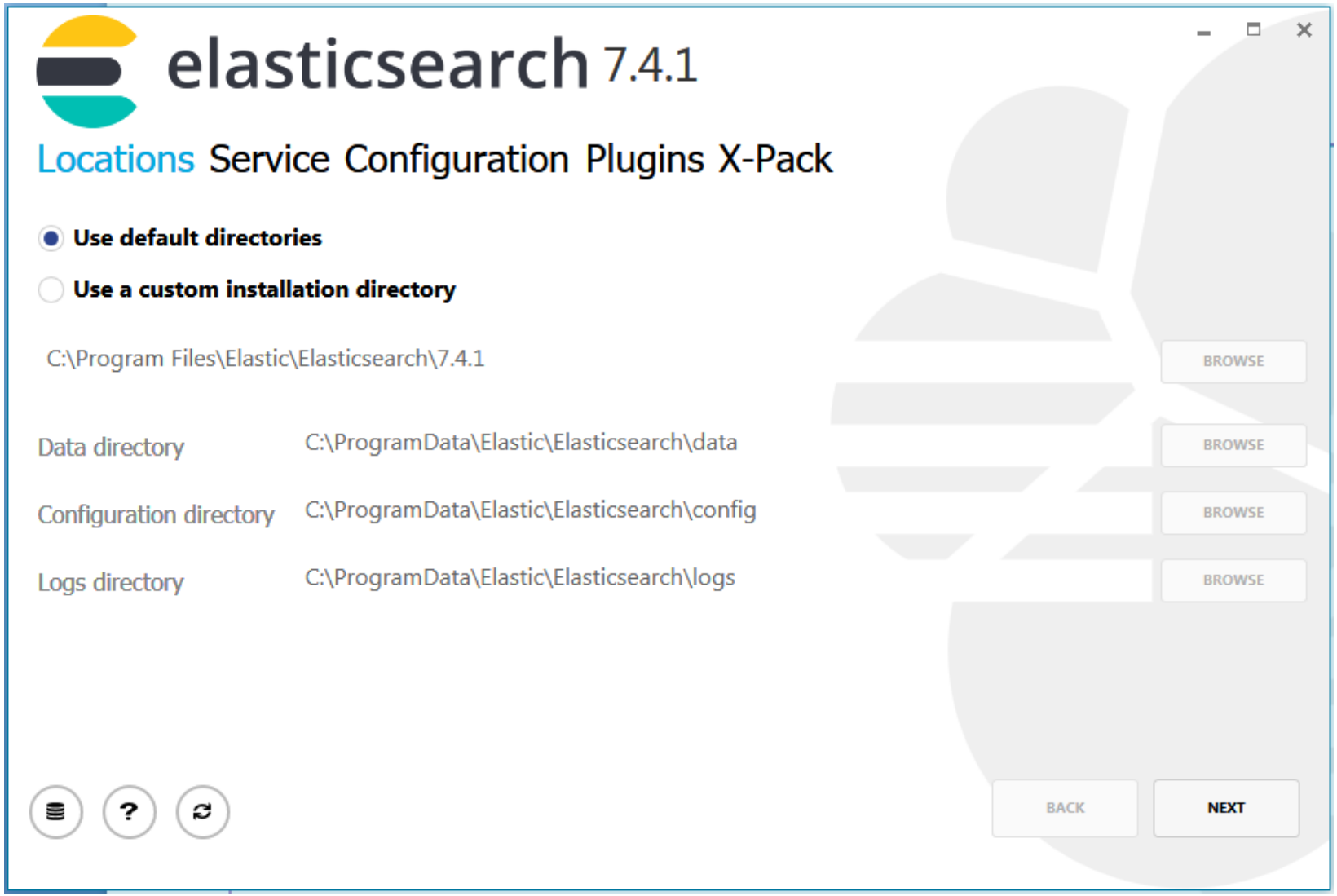
```
c:\elasticsearch-6.2.2\bin>elasticsearch-service.bat install
Installing service      :  "elasticsearch-service-x64"
Using JAVA_HOME (64-bit):  "c:\jvm\jdk1.8"
The service 'elasticsearch-service-x64' has been installed.
```

Konfiguracja usługi w Windows

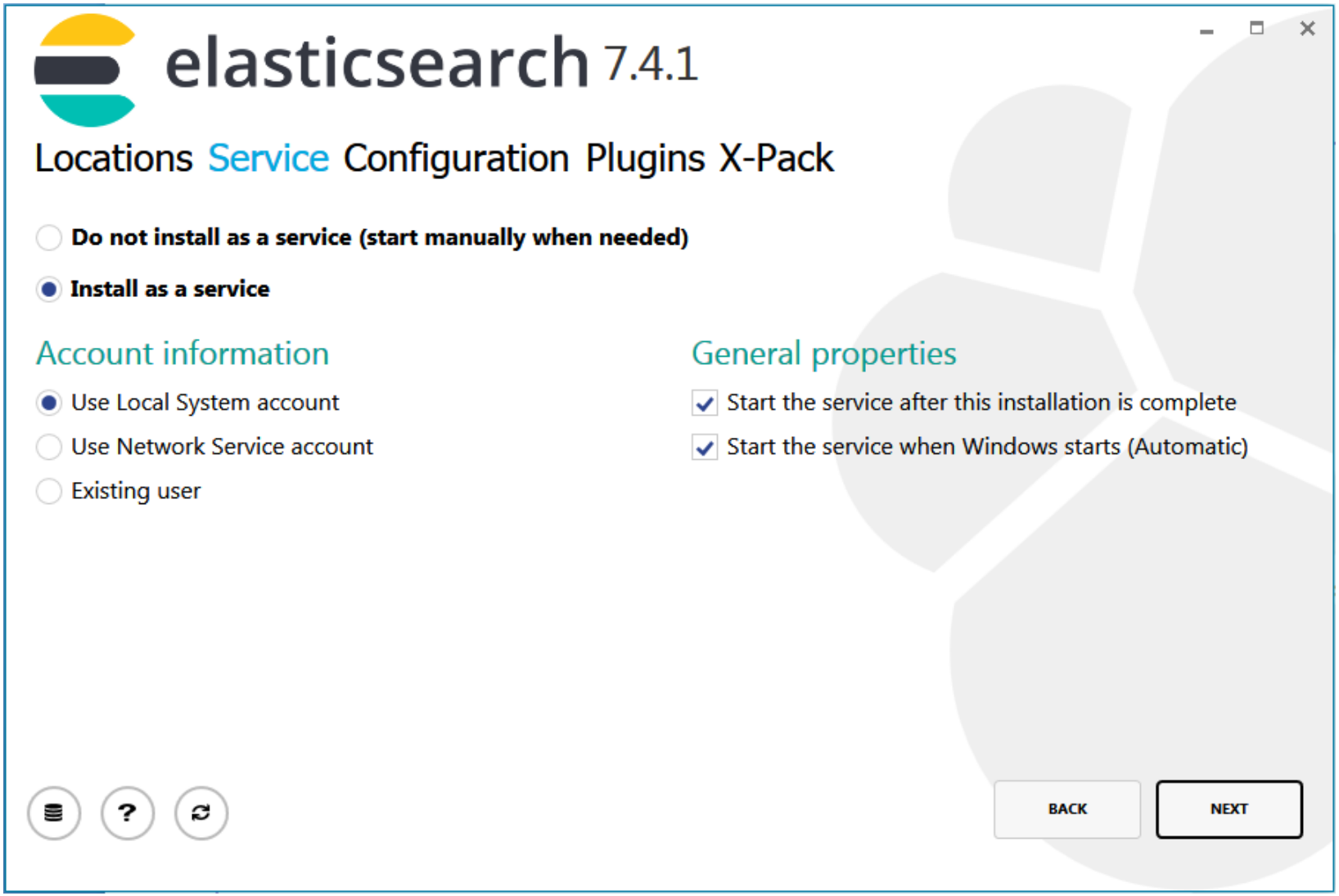
- elasticsearch-service-mgr.exe



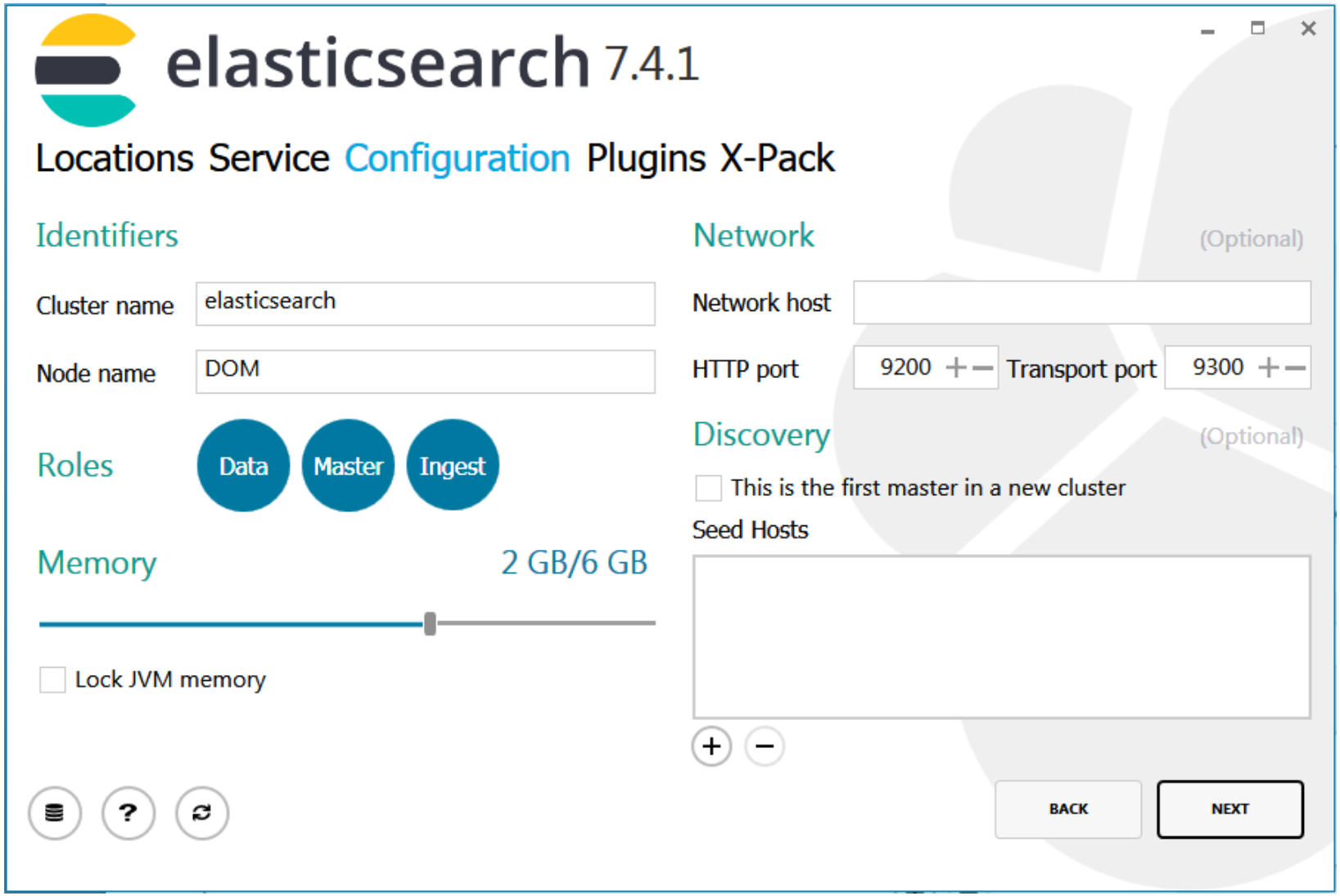
Instalacja MSI



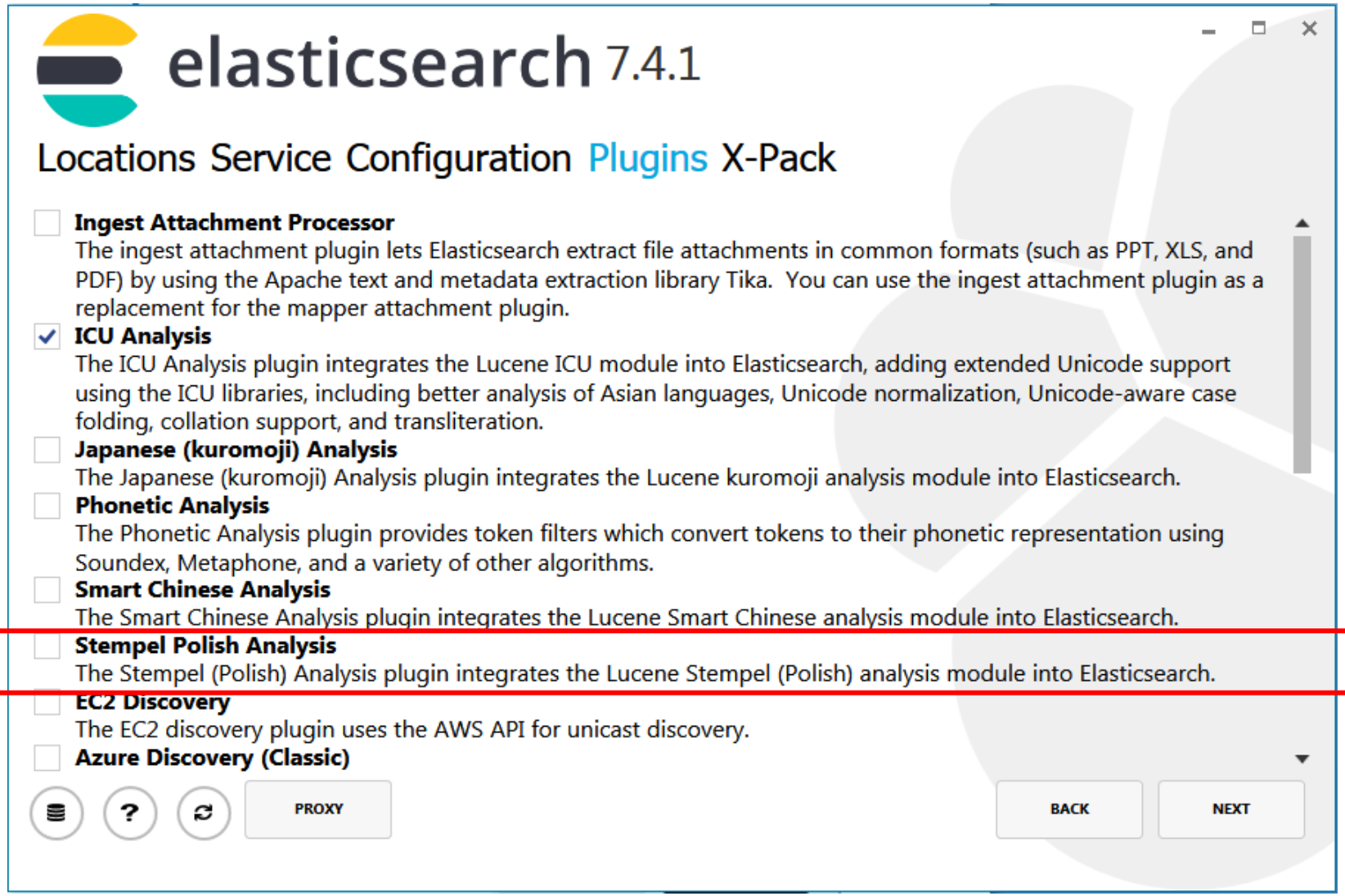
Instalacja MSI



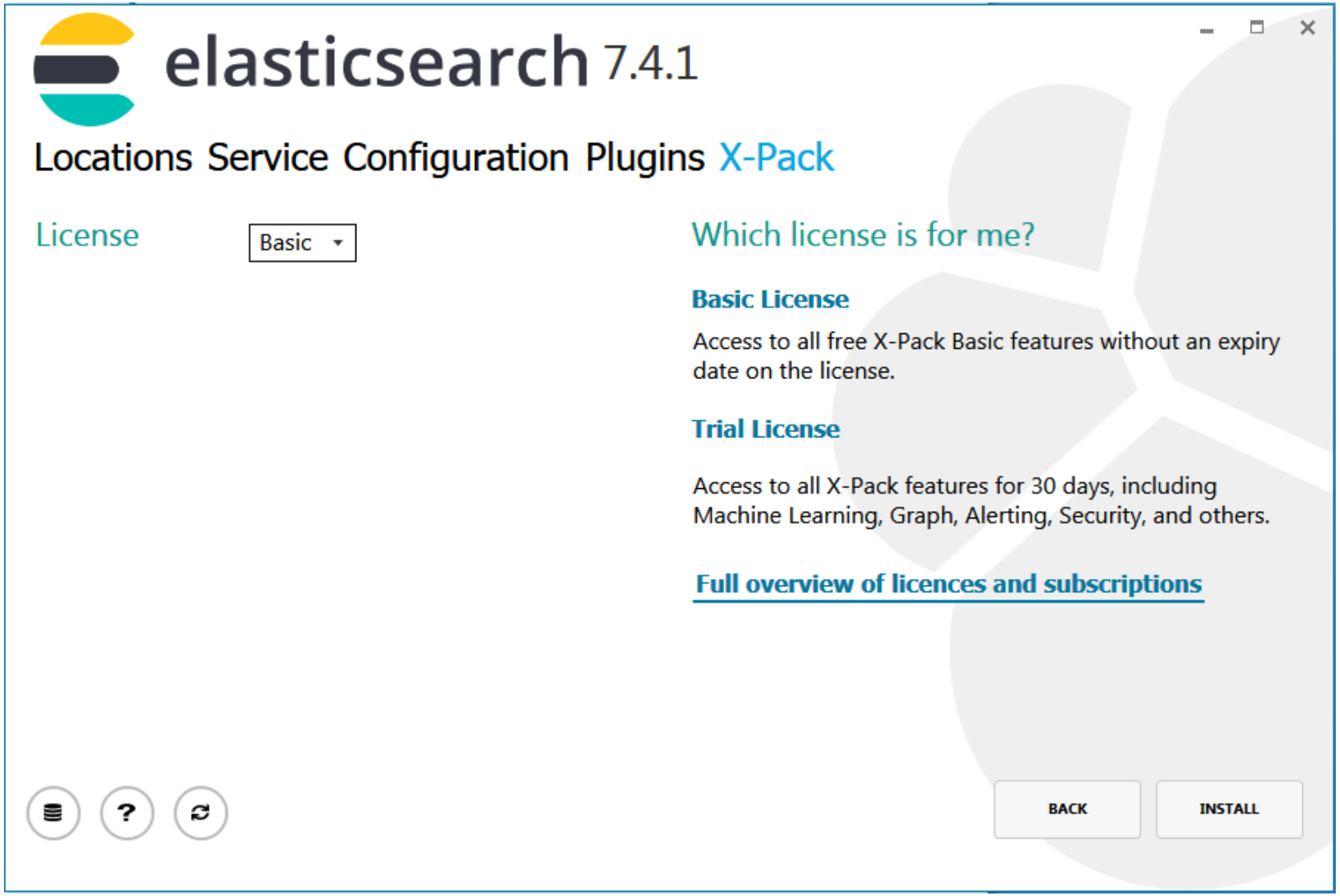
Instalacja MSI



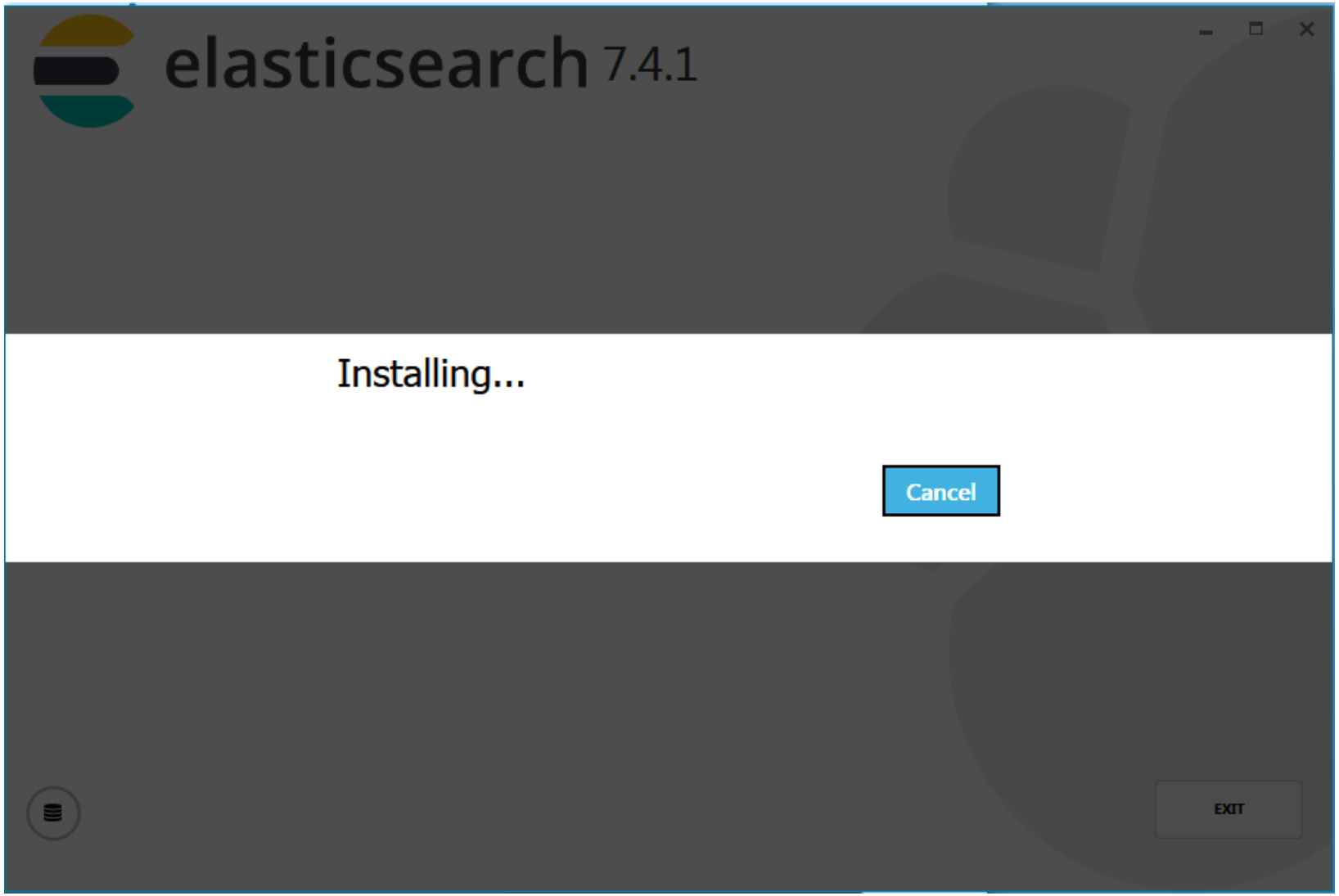
Instalacja MSI



Instalacja MSI



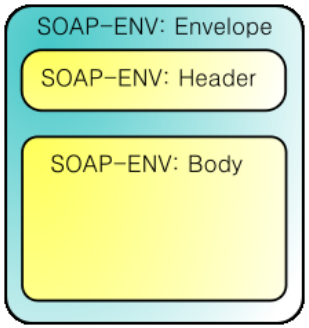
Instalacja MSI



REST (ang. Representational State Transfer)

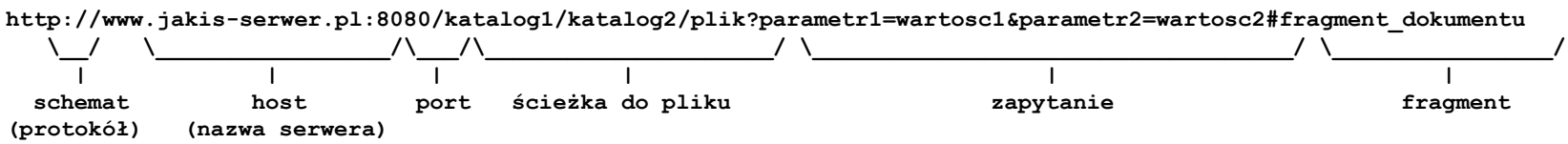
- REST to system komunikacji pomiędzy aplikacjami wykorzystujący protokół HTTP. To styl architektury oprogramowania wywiedziony z doświadczeń przy pisaniu specyfikacji protokołu HTTP dla systemów rozproszonych
- Używane są różne metody. Najczęściej są to:
  - GET: wykorzystywana do pobierania (listowania) danych
  - POST: wykorzystywana do utworzenia nowego zasobu
  - PUT: wykorzystywana do edycji zasobu (modyfikacja)
  - DELETE: wykorzystywana do usuwania zasobów
  - Inne mniej istotne: OPTIONS, HEAD, PATCH

- REST nie jest ustandaryzowanym systemem np. na poziomie odpowiedzi, jakie otrzymujemy od serwerów
- Konkurencyjny (ale starszy) standard to SOAP (ang. Simple Object Access Protocol)
  - SOAP jest ustandaryzowany, co znaczy, że to my musimy dostosować się do wytworzonych reguł, tworząc system komunikacji
  - SOAP wykorzystuje XML do kodowania wywołań i najczęściej protokołu HTTP do ich przenoszenia, możliwe jest jednak wykorzystanie innych protokołów do transportu danych



REST vs. RESTful

- API serwisów webowych, które stosują technikę REST zwykło się nazywać RESTful (ang. RESTful services)
- W serwisach RESTful dostęp do zasobów odbywa się za pośrednictwem identyfikatorów URI
  - ang. Uniform Resource Identifier
    pol. Ujednolicony Identyfikator Zasobów



- Bardzo prosty przykład aplikacji w PHP zbudowanej według założeń REST
  - https://phppot.com/php/php-restful-web-service/

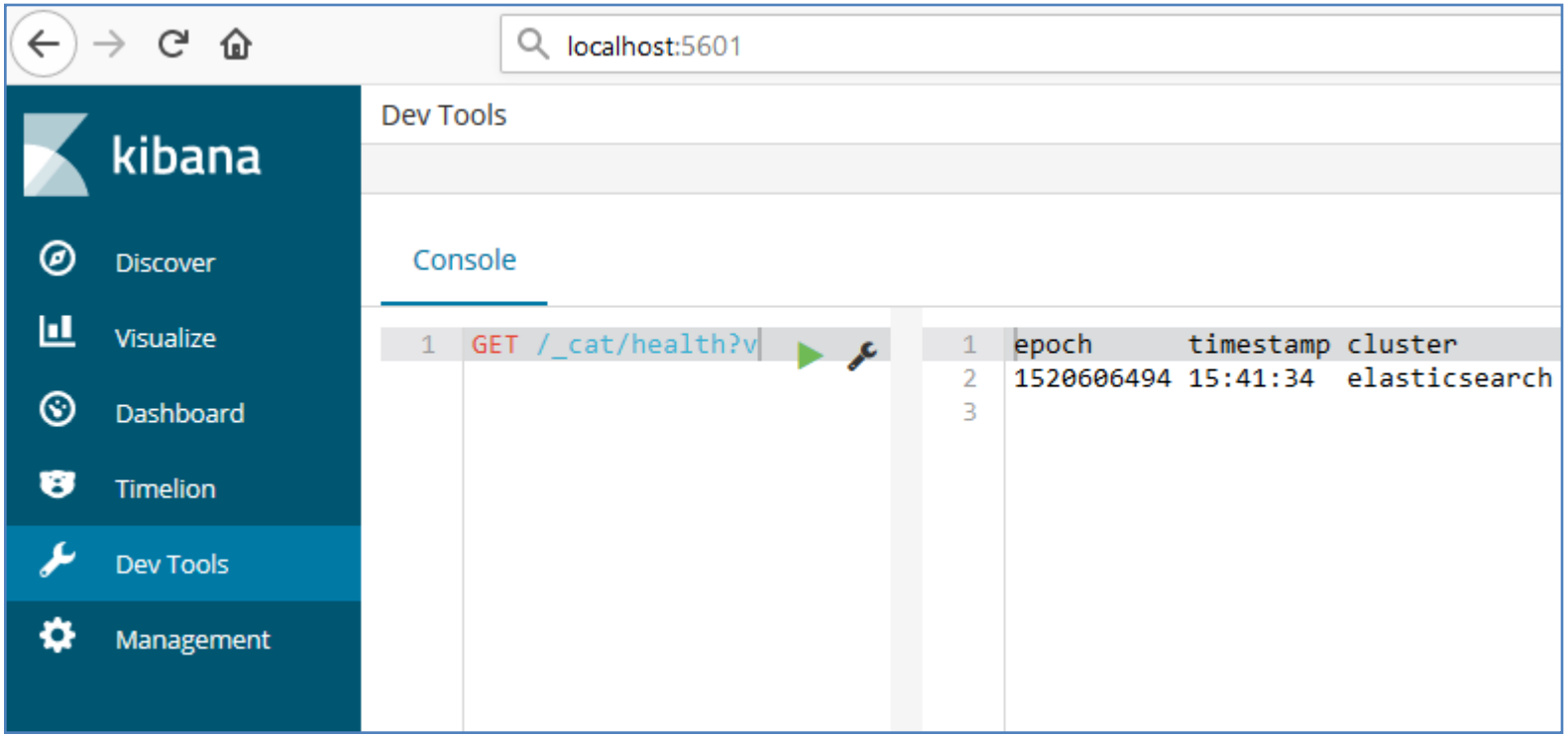
RESTful API with JSON over HTTP

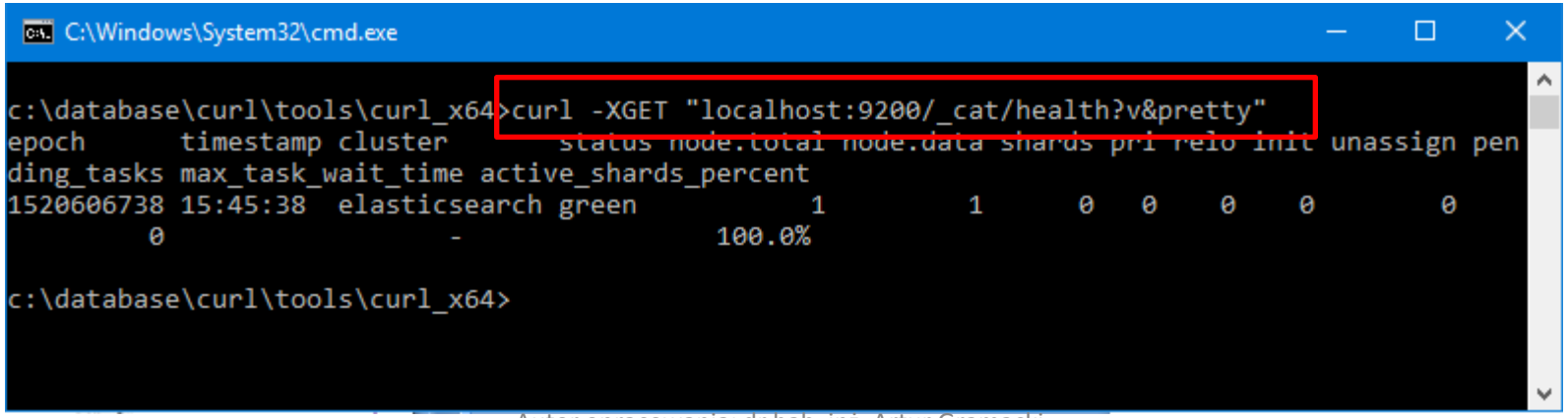
curl -X<VERB> '<PROTOCOL>://<HOST>/<PATH>?<QUERY_STRING>' -d '<BODY>'

```
curl -XGET 'http://localhost:9200/_count?pretty' -d '
{
  "query": {
    "match_all": {}
  }
}
'
```

```
GET /_count?pretty
{
  "query": {
    "match_all": {}
  }
}
```

Konsola w Kibana vs cURL



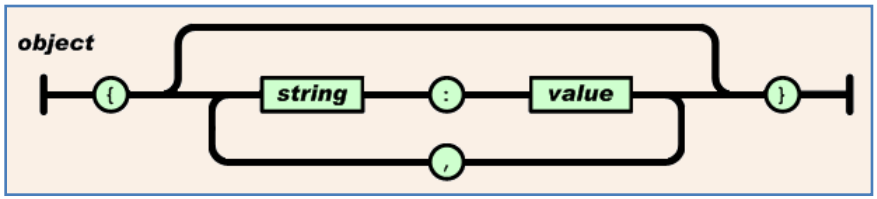


JSON

- JSON (JavaScript Object Notation)
- Lekki format wymiany danych komputerowych
  - tekstowy
  - pliki powinny być kodowane w UTF-8
- Podzbiór języka JavaScript
- Typ MIME dla formatu JSON to application/json
- Jeden z nieformalnych sposobów przekazywania danych do aplikacji opartych o AJAX
- Oparty o dwie struktury
  - zbiór par nazwa/wartość
  - uporządkowana lista wartości

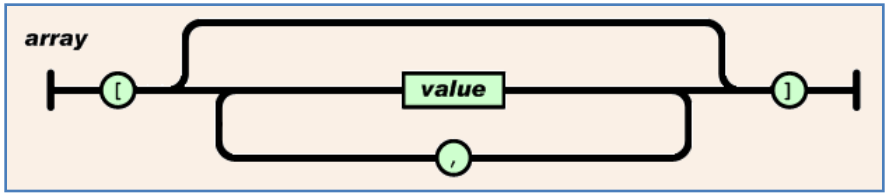
JSON

- **Obiekt** jest nieuporządkowanym zbiorem par nazwa/wartość



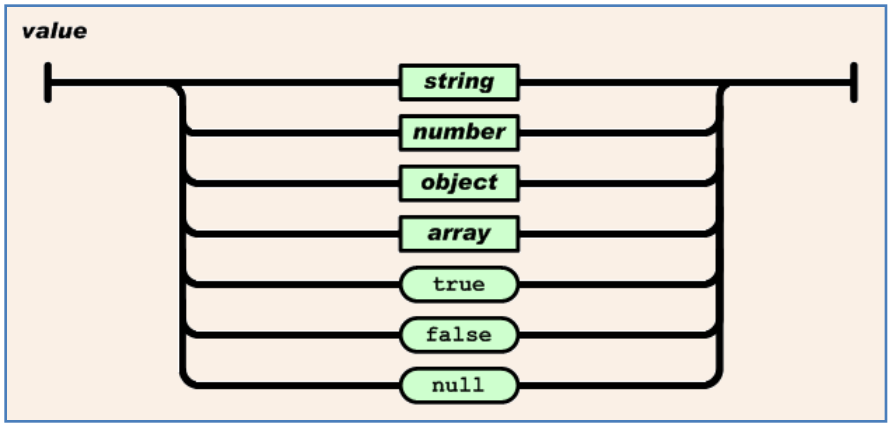
JSON

- **Tabela** jest uporządkowanym zbiorem wartości



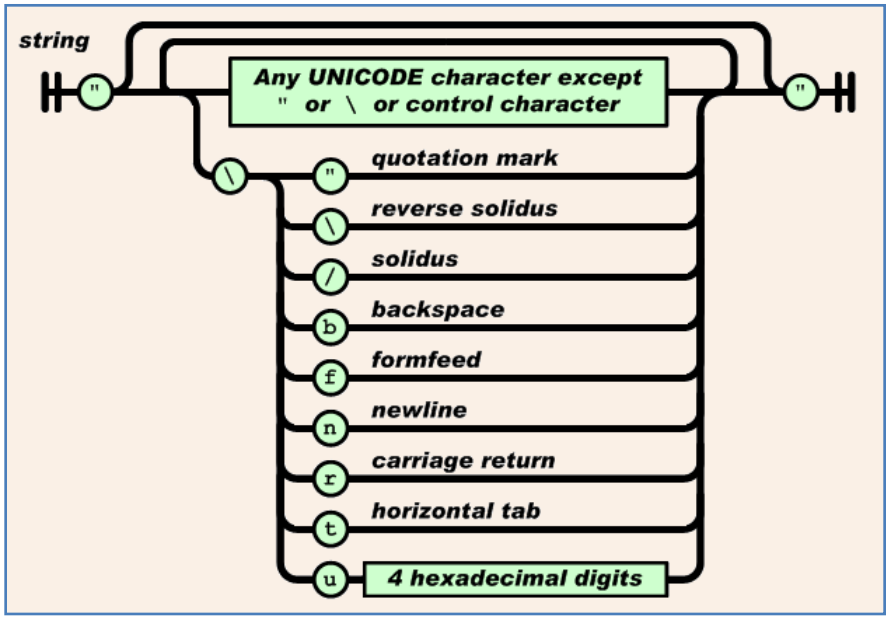
JSON

- **Wartość** to łańcuch znakowy



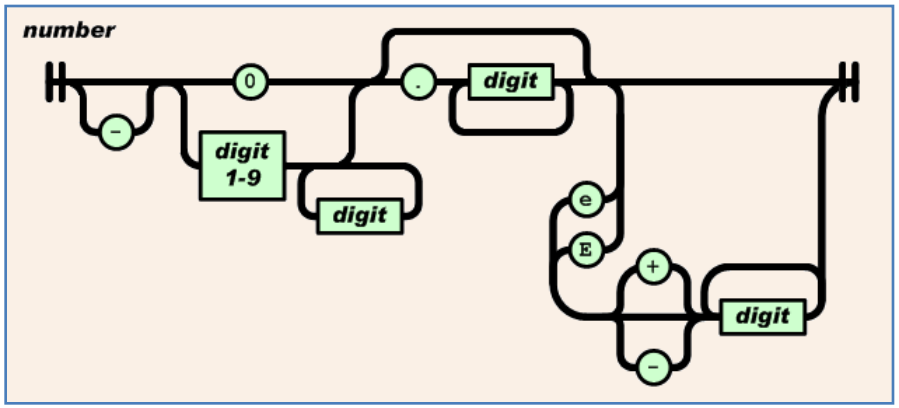
JSON

- **Łancuch znakowy** jest zbiorem zera lub większej ilości znaków Unicode



JSON

- **Liczby** zapisywane w formacie JSON są bardzo podobne do liczb w języku C lub Java, poza tym wyjątkiem, że nie używa się formatów ósemkowych i szesnastkowych

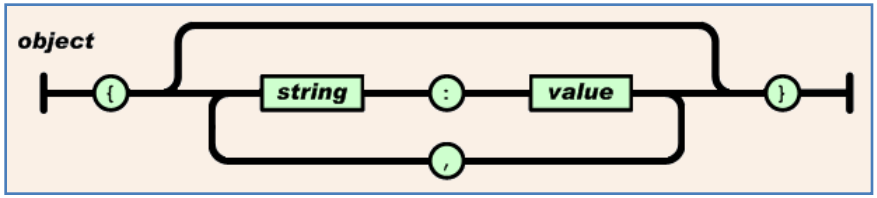


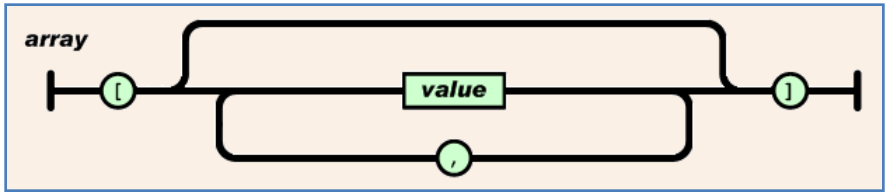
JSON

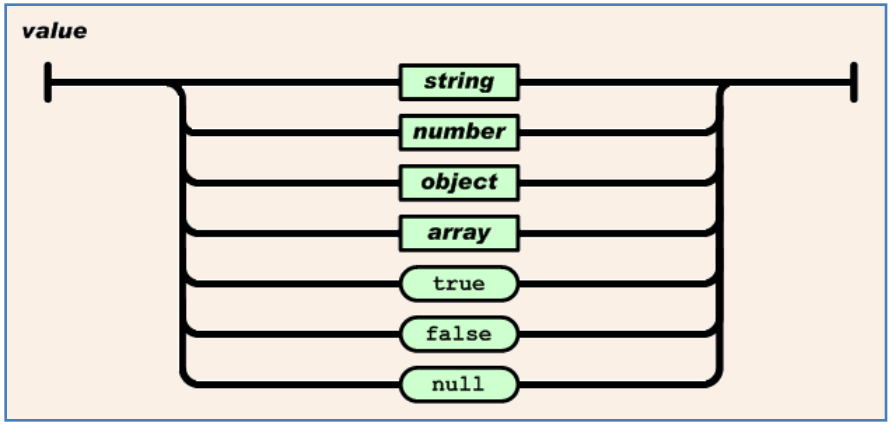
- **Obiekt** jest nieuporządkowanym zbiorem par nazwa/wartość.
- **Tabela** jest uporządkowanym zbiorem wartości
- **Wartość** to łańcuch znakowy
- **Łancuch znakowy** jest zbiorem zera lub większej ilości znaków Unicode
- **Liczby** zapisywane w formacie JSON są bardzo podobne do liczb w języku C lub Java, poza tym wyjątkiem, że nie używa się formatów ósemkowych i szesnastkowych

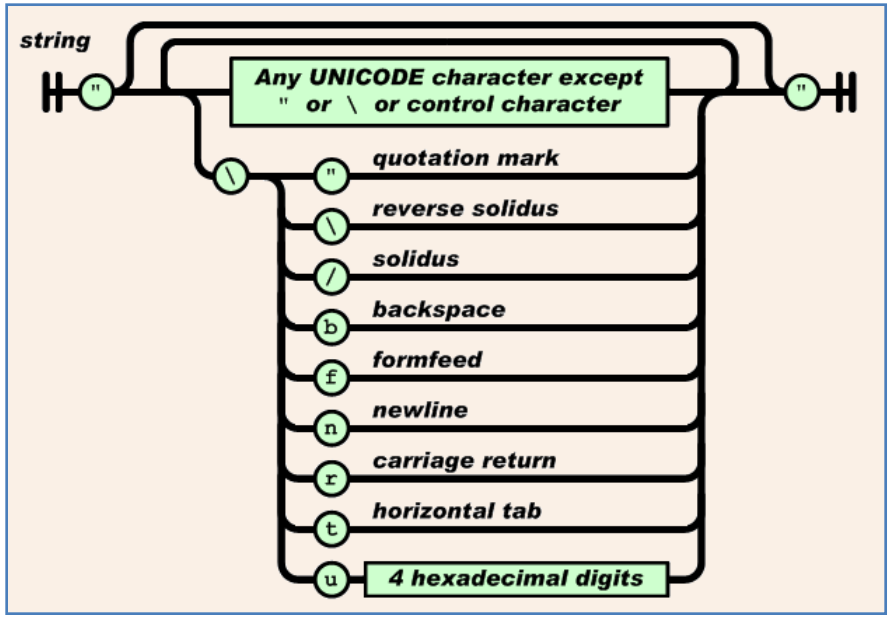
JSON

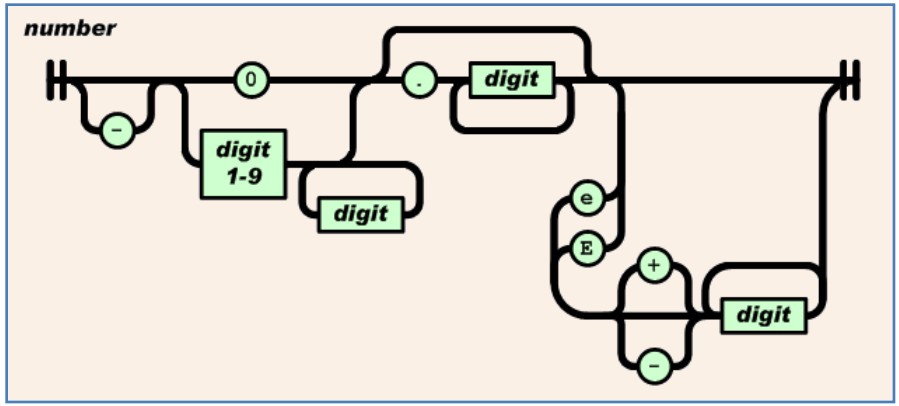
– https://www.json.org/json-pl.html











JSON vs XML

- Lżejszy, łatwiejszy do odczytu i zapisu
- Posiada typ tablicowy
- Pliki są bardziej "human readable"
- Nie ma wsparcia dla opisu sposobu wyświetlania
- Obsługuje skalarne typy danych
- Dzięki tablicom i obiektom można opisywać dane strukturalne

- Mniej prosty niż JSON
- Nie posiada typu tablicowego
- Pliki są mniej "human readable"
- Posiada wsparcie dla opisu sposobu wyświetlania (bo jest to język znaczników)
- Nie ma żadnego wsparcia dla opisu typów danych (informacje o typie przechowywane w XML Schema)

Oba formaty są:

- otwarte
- samodokumentujące
- hierarchiczne
- niezależne od języka programowania, który z nich korzysta
- łatwe do programowego parsowania
- mogą być źródłem danych w AJAX (korzystając z obiektu XMLHttpRequest)

JSON vs XML

JSON Example

```
{"employees":[
    { "firstName":"John", "lastName":"Doe" },
    { "firstName":"Anna", "lastName":"Smith" },
    { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

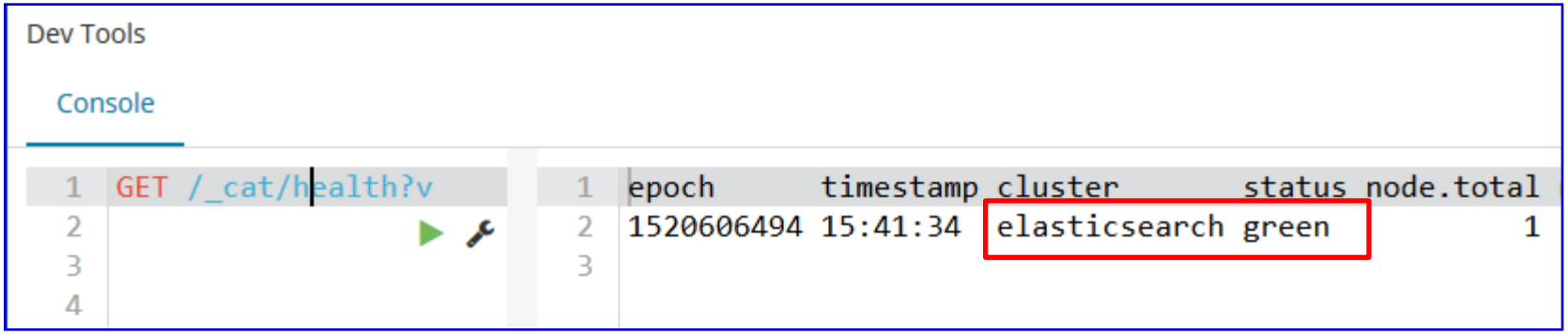
```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName> <lastName>Jones</lastName>
    </employee>
</employees>
```

Eksplorowanie klastra

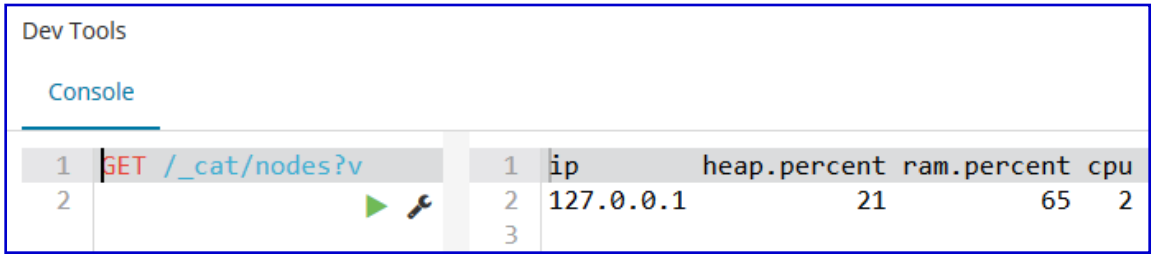
- Całość obsługi realizowana jest za pomocą REST API
- Podstawowe zadania
  - kontrola statusu klastra, węzłów, stanu indeksów, wyświetlanie różnych statystyk
  - administrowanie klastrem, węzłami, danymi w indeksach oraz różnymi rodzajami metadanych
  - operacje CRUD (Create, Read, Update, and Delete) oraz różne opcje przeszukiwania indeksów
  - wykonywanie różnych bardziej zaawansowanych operacji przeszukiwania (stronicowanie, sortowanie, filtrowanie, agregowanie, generowanie skryptów i wiele innych)

Eksplorowanie klastra

- „Kolorowe” stany klastra
  - Green - everything is good (cluster is fully functional)
  - Yellow - all data is available but some replicas are not yet allocated (cluster is fully functional)
  - Red - some data is not available for whatever reason (cluster is partially functional)

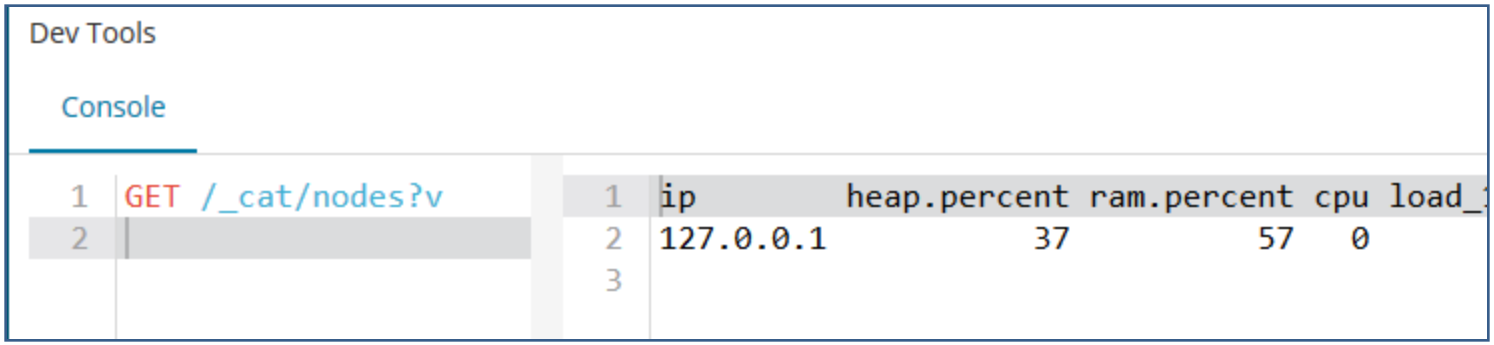


- Lista węzłów



Eksplorowanie klastra

- Lista węzłów



Przykładowe dane

- Podstawy
  - załadować przykładowe dane. Na stronie WWW projektu mamy pliki: shakespeare.json, accounts.zip, logs.jsonl.gz

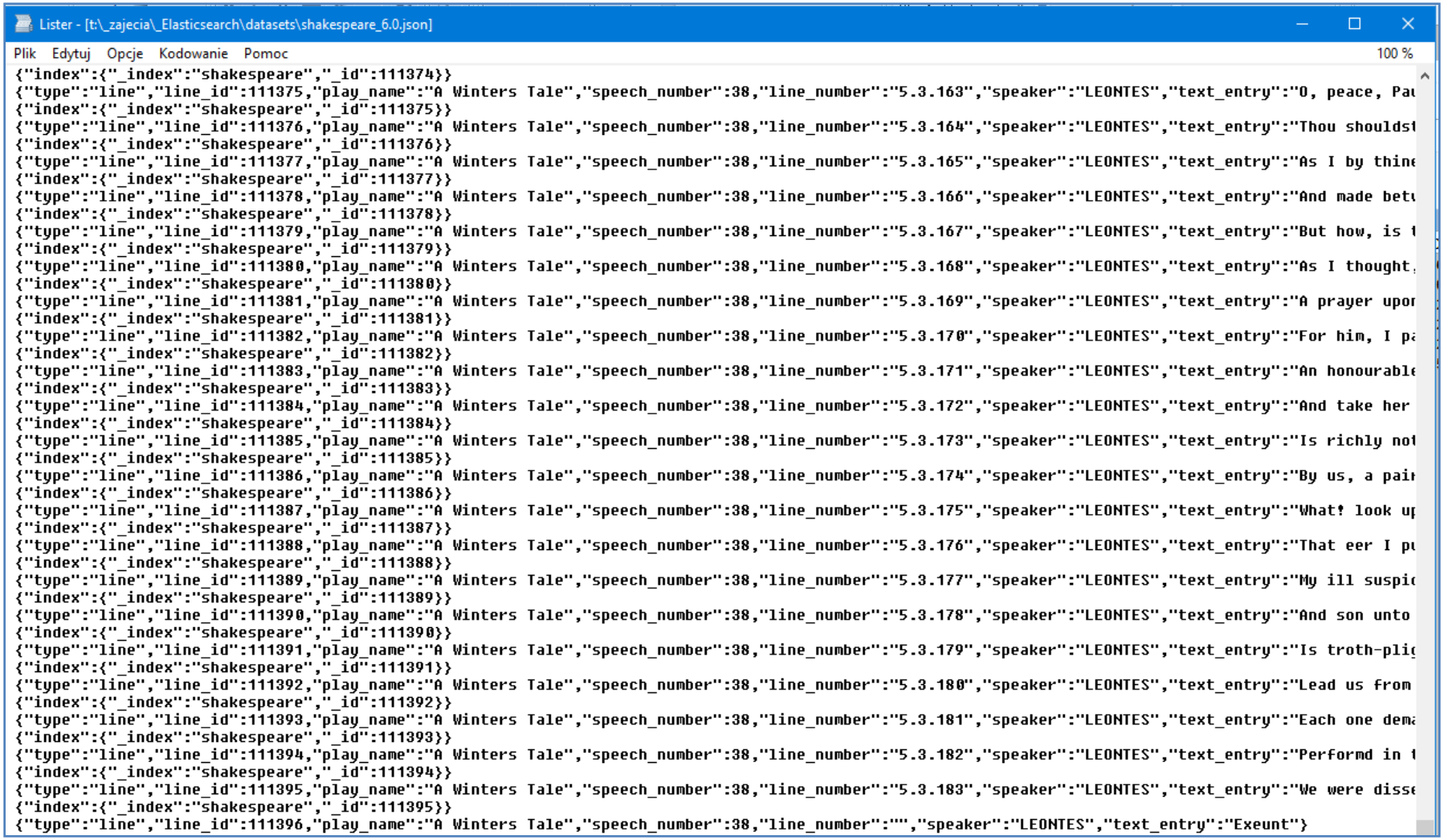
```
{
    "line_id": INT,
    "play_name": "String",
    "speech_number": INT,
    "line_number": "String",
    "speaker": "String",
    "text_entry": "String",
}
```

```
{
    "memory": INT,
    "geo.coordinates": "geo_point"
    "@timestamp": "date"
}
```

```
{
    "account_number": INT,
    "balance": INT,
    "firstname": "String",
    "lastname": "String",
    "age": INT,
    "gender": "M or F",
    "address": "String",
    "employer": "String",
    "email": "String",
    "city": "String",
    "state": "String"
}
```

Przykładowe dane

- Podstawy, plik [shakespeare.json]



Przykładowe dane

- Podstawy, plik shakespeare.json
  - utworzenie mapowania (ang. mappings) poszczególnych pół
  - mapowanie pozwala zdefiniować logiczną strukturę danych

```
{
  "index":{
    "_index":"shakespeare",
    "_id":3
  }
}{
  "type":"line",
  "line_id":4,
  "play_name":"Henry IV",
  "speech_number":1,
  "line_number":"1.1.1",
  "speaker":"KING HENRY IV",
  "text_entry":"So shaken as we are, so wan with care,"
}
```

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

```
{
    "line_id": INT,
    "play_name": "String",
    "speech_number": INT,
    "line_number": "String",
    "speaker": "String",
    "text_entry": "String",
}
```

Przykładowe dane

- Podstawy, shakespeare.json
  - załadowanie danych do Elasticsearch

```
c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -H "Content-Type: application/x-ndjson" -XPOST "localhost:
9200/shakespeare/doc/_bulk?pretty" --data-binary @shakespeare_6.0.json > log_shakespeare
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 62.7M  100 38.5M  100 24.1M  4473k  2801k  0:00:08  0:00:08 --:--:-- 3385k

c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -XGET "localhost:9200/_cat/indices?v"
health status index       uuid                   pri rep docs.count docs.deleted store.size pri.store.size
yellow open   shakespeare 34B6PBOTSIuxcPxpZ5WtdQ   5   1     111396            0     21.4mb         21.4mb
yellow open   bank        IFDCxNOzSxWPEiMD4RoFug   5   1       1000            0    475.1kb        475.1kb
yellow open   my_index    RFAOPhuIRvmzUJrYclsoqw   5   1          1            0      4.5kb          4.5kb
```

Dev Tools

Console

```
1  GET /_cat
   /indices?v
```

```
1  health status index       uuid                   pri rep docs.count docs.deleted store.size pri.store.size
2  yellow open   shakespeare 34B6PBOTSIuxcPxpZ5WtdQ   5   1     111396            0     21.3mb         21.3mb
3  yellow open   bank        IFDCxNOzSxWPEiMD4RoFug   5   1       1000            0    475.1kb        475.1kb
4  yellow open   my_index    RFAOPhuIRvmzUJrYclsoqw   5   1          1            0      4.5kb          4.5kb
5
```

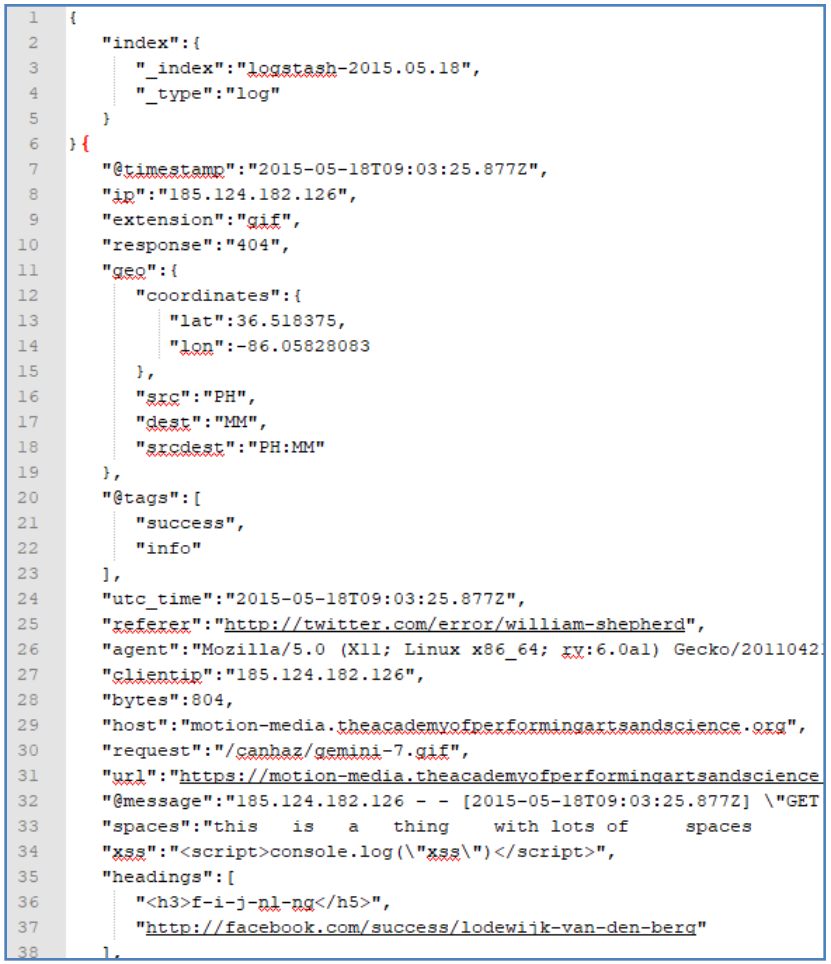
Przykładowe dane

- Podstawy, plik logs.jsonl


```
{"index":{"_index":"logstash-2015.05.18","_type":"log"}}
{"@timestamp":"2015-05-18T09:03:25.877Z","ip":"185.124.182.126","extension":"gif","response":"404","geo":{"coordinates":{"lat":36.518375,"lon":-
86.05828083},"src":"PH","dest":"MM","srcdest":"PH:MM"},"@tags":["success","info"],"utc_time":"2015-05-18T09:03:25.877Z","referer":
"http://twitter.com/error/william-shepherd","agent":"Mozilla/5.0 (X11; Linux x86_64; rv:6.0a1) Gecko/20110421 Firefox/6.0a1","clientip":
"185.124.182.126","bytes":804,"host":"motion-media.theacademyofperformingartsandscience.org","request":"/canhaz/gemini-7.gif","url":
"https://motion-media.theacademyofperformingartsandscience.org/canhaz/gemini-7.gif","@message":"185.124.182.126 - - [2015-05-18T09:03:25.877Z] \"GET
/canhaz/gemini-7.gif HTTP/1.1\" 404 804 \"-\" \"Mozilla/5.0 (X11; Linux x86_64; rv:6.0a1) Gecko/20110421 Firefox/6.0a1\"","spaces":"this   is   a
thing    with lots of     spaces       wwwwoooooo","xss":"<script>console.log(\"xss\")</script>","headings":["<h3>f-i-j-nl-ng</h5>",
"http://facebook.com/success/lodewijk-van-den-berg"],"links":["daniel-tani@facebook.com","http://nytimes.com/security/kathryn-sullivan",
"www.nytimes.com"],"relatedContent":[{"url":"http://www.laweekly.com/news/cbs-crew-rat-fink-2368032","og:type":"article","og:title":"CBS Crew Rat
Fink","og:description":"Near a couple of auto body shops (and a sharp new Space Invader mosaic that we&#039;ll post soon) near Temple and
Westmoreland is a CBS wall with a nice Rat ...","og:url":"http://www.laweekly.com/news/cbs-crew-rat-fink-2368032","article:published_time":
"2008-01-14T08:05:26-08:00","article:modified_time":"2014-10-28T14:59:52-07:00","article:section":"News","article:tag":"Mark Mauer","og:image":
"http://IMAGES1.laweekly.com/imager/cbs-crew-rat-fink/u/original/2430299/img_2049.jpg","og:image:height":"360","og:image:width":"480","og:site_name":
"LA Weekly","twitter:title":"CBS Crew Rat Fink","twitter:description":"Near a couple of auto body shops (and a sharp new Space Invader mosaic that
we&#039;ll post soon) near Temple and Westmoreland is a CBS wall with a nice Rat ...","twitter:card":"summary","twitter:image":
"http://IMAGES1.laweekly.com/imager/cbs-crew-rat-fink/u/original/2430299/img_2049.jpg","twitter:site":"@laweekly"},{"url":
"http://www.laweekly.com/news/push-and-retna-in-koreatown-2368043","og:type":"article","og:title":"Push and Retna in Koreatown","og:description":
"Yeah, I originally had this posted this morning as Push &amp;amp; Ayer - Sorry. It looked like a Retna piece, but I saw the Ayer in there and thought
that must ...","og:url":"http://www.laweekly.com/news/push-and-retna-in-koreatown-2368043","article:published_time":"2008-01-29T07:28:32-08:00",
"article:modified_time":"2014-10-28T14:59:54-07:00","article:section":"News","article:tag":"Shelley Leopold","og:image":
"http://IMAGES1.laweekly.com/imager/push-and-retna-in-koreatown/u/original/2430376/img_3671.jpg","og:image:height":"360","og:image:width":"480",
"og:site_name":"LA Weekly","twitter:title":"Push and Retna in Koreatown","twitter:description":"Yeah, I originally had this posted this morning as
Push &amp;amp; Ayer - Sorry. It looked like a Retna piece, but I saw the Ayer in there and thought that must ...","twitter:card":"summary","twitter:image"
:"http://IMAGES1.laweekly.com/imager/push-and-retna-in-koreatown/u/original/2430376/img_3671.jpg","twitter:site":"@laweekly"},{"url":
"http://www.laweekly.com/news/asylm-ruets-pdb-on-santa-monica-2368012","og:type":"article","og:title":"Asylm, Ruets, PDB on Santa Monica",
"og:description":"Not a new piece, but a well-hidden gem a little south of Santa Monica Blvd. in an alley off of Heliotrope or Edgemont. I&#039;ve
been sitting on this for a w...","og:url":"http://www.laweekly.com/news/asylm-ruets-pdb-on-santa-monica-2368012","article:published_time":
"2008-04-22T15:11:15-07:00","article:modified_time":"2014-10-28T14:59:48-07:00","article:section":"News","article:tag":"Culture and Lifestyle",
"og:image":"http://images1.laweekly.com/imager/asylm-ruets-pdb-on-santa-monica/u/original/2430137/img_5027.jpg","og:image:height":"360",
"og:image:width":"480","og:site_name":"LA Weekly","twitter:title":"Asylm, Ruets, PDB on Santa Monica","twitter:description":"Not a new piece, but a
well-hidden gem a little south of Santa Monica Blvd. in an alley off of Heliotrope or Edgemont. I&#039;ve been sitting on this for a w...",
"twitter:card":"summary","twitter:image":"http://images1.laweekly.com/imager/asylm-ruets-pdb-on-santa-monica/u/original/2430137/img_5027.jpg",
"twitter:site":"@laweekly"},{"url":"http://www.laweekly.com/news/laurence-tribe-tangles-with-cbs-and-la-city-hall-2396867","og:type":"article",
"og:title":"Laurence Tribe Tangles with CBS and L.A. City Hall","og:description":"The United States Court of Appeals for the Ninth Circuit&rsquo;s
Courtroom 3 - a miniature auditorium with comfortable, smoked salmon-colored seats - wa...","og:url":
"http://www.laweekly.com/news/laurence-tribe-tangles-with-cbs-and-la-city-hall-2396867","article:published_time":"2008-06-04T14:16:10-07:00",
"article:modified_time":"2014-11-26T14:43:59-08:00","article:section":"News","og:site_name":"LA Weekly","twitter:title":"Laurence Tribe Tangles with
CBS and L.A. City Hall","twitter:description":"The United States Court of Appeals for the Ninth Circuit&rsquo;s Courtroom 3 - a miniature auditorium
with comfortable, smoked salmon-colored seats - wa...","twitter:card":"summary","twitter:site":"@laweekly"}],"machine":{"os":"win xp","ram":3221225472
},"@version":"1"}
```

Przykładowe dane

- Podstawy, plik logs.jsonl, po zastosowaniu formatera
  - Np. https://jsonformatter.curiousconcept.com/



Przykładowe dane

- Podstawy, plik logs.jsonl

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

```
{
    "memory": INT,
    "geo.coordinates": "geo_point"
    "@timestamp": "date"
}
```

Przykładowe dane

{"index":{"_index":"logstash-2015.05.18","_type":"log"}}
{"@timestamp":"2015-05-18T09:03:25.877Z","ip":"185.124.182.126","extension":"gif","response":"404",
"geo":{"coordinates":{"lat":36.518375,"lon":-
86.05828083},"src":"PH","dest":"MM","srcdest":"PH:MM"},"@tags":["success","info"],"utc_time":"2015-05-
18T09:03:25.877Z","referer":"http://twitter.com/error/william-shepherd","agent":"Mozilla/5.0 (X11; Linux x86_64; rv:6.0a1) Gecko/20110421
Firefox/6.0a1","clientip":"185.124.182.126","bytes":804,"host":"motion-media.theacademyofperformingartsandscience.org","request":"/canhaz/gemini-
7.gif","url":"https://motion-media.theacademyofperformingartsandscience.org/canhaz/gemini-7.gif","@message":"185.124.182.126 - - [2015-05-
18T09:03:25.877Z] \"GET /canhaz/gemini-7.gif HTTP/1.1\" 404 804 \"-\" \"Mozilla/5.0 (X11; Linux x86_64; rv:6.0a1) Gecko/20110421
Firefox/6.0a1\"","spaces":"this is a thing with lots of spaces
wwwwooooooo","xss":"<script>console.log(\"xss\")</script>","headings":["<h3>f-i-j-nl-ng</h5>","http://facebook.com/success/lodewijk-van-den-
berg"],"links":["daniel-tani@facebook.com","http://nytimes.com/security/kathryn-
sullivan","www.nytimes.com"],"relatedContent":[{"url":"http://www.laweekly.com/news/cbs-crew-rat-fink-2368032","og:type":"article","og:title":"CBS
Crew Rat Fink","og:description":"Near a couple of auto body shops (and a sharp new Space Invader mosaic that we'll post soon) near Temple and
Westmoreland is a CBS wall with a nice Rat ...","og:url":"http://www.laweekly.com/news/cbs-crew-rat-fink-2368032","article:published_time":"2008-01-
14T08:05:26-08:00","article:modified_time":"2014-10-28T14:59:52-07:00","article:section":"News","article:tag":"Mark
Mauer","og:image":"http://IMAGES1.laweekly.com/imager/cbs-crew-rat-
fink/u/original/2430299/img_2049.jpg","og:image:height":"360","og:image:width":"480","og:site_name":"LA Weekly","twitter:title":"CBS Crew Rat
Fink","twitter:description":"Near a couple of auto body shops (and a sharp new Space Invader mosaic that we'll post soon) near Temple and
Westmoreland is a CBS wall with a nice Rat ...","twitter:card":"summary","twitter:image":"http://IMAGES1.laweekly.com/imager/cbs-crew-rat-
fink/u/original/2430299/img_2049.jpg","twitter:site":"@laweekly"},{"url":"http://www.laweekly.com/news/push-and-retna-in-koreatown-
2368043","og:type":"article","og:title":"Push and Retna in Koreatown","og:description":"Yeah, I originally had this posted this morning as Push
& Ayer - Sorry. It looked like a Retna piece, but I saw the Ayer in there and thought that must
...","og:url":"http://www.laweekly.com/news/push-and-retna-in-koreatown-2368043","article:published_time":"2008-01-29T07:28:32-
08:00","article:modified_time":"2014-10-28T14:59:54-07:00","article:section":"News","article:tag":"Shelley
Leopold","og:image":"http://IMAGES1.laweekly.com/imager/push-and-retna-in-
koreatown/u/original/2430376/img_3671.jpg","og:image:height":"360","og:image:width":"480","og:site_name":"LA Weekly","twitter:title":"Push and Retna
in Koreatown","twitter:description":"Yeah, I originally had this posted this morning as Push & Ayer - Sorry. It looked like a Retna piece, but I
saw the Ayer in there and thought that must ...","twitter:card":"summary","twitter:image":"http://IMAGES1.laweekly.com/imager/push-and-retna-in-
koreatown/u/original/2430376/img_3671.jpg","twitter:site":"@laweekly"},{"url":"http://www.laweekly.com/news/asylm-ruets-pdb-on-santa-monica-
2368012","og:type":"article","og:title":"Asylm, Ruets, PDB on Santa Monica","og:description":"Not a new piece, but a well-hidden gem a little south
of Santa Monica Blvd. in an alley off of Heliotrope or Edgemont. I've been sitting on this for a
w...","og:url":"http://www.laweekly.com/news/asylm-ruets-pdb-on-santa-monica-2368012","article:published_time":"2008-04-22T15:11:15-
07:00","article:modified_time":"2014-10-28T14:59:48-07:00","article:section":"News","article:tag":"Culture and
Lifestyle","og:image":"http://images1.laweekly.com/imager/asylm-ruets-pdb-on-santa-
monica/u/original/2430137/img_5027.jpg","og:image:height":"360","og:image:width":"480","og:site_name":"LA Weekly","twitter:title":"Asylm, Ruets, PDB
on Santa Monica","twitter:description":"Not a new piece, but a well-hidden gem a little south of Santa Monica Blvd. in an alley off of Heliotrope or
Edgemont. I've been sitting on this for a w...","twitter:card":"summary","twitter:image":"http://images1.laweekly.com/imager/asylm-ruets-pdb-
on-santa-monica/u/original/2430137/img_5027.jpg","twitter:site":"@laweekly"},{"url":"http://www.laweekly.com/news/laurence-tribe-tangles-with-cbs-
and-la-city-hall-2396867","og:type":"article","og:title":"Laurence Tribe Tangles with CBS and L.A. City Hall","og:description":"The United States
Court of Appeals for the Ninth Circuit’s Courtroom 3 - a miniature auditorium with comfortable, smoked salmon-colored seats -
wa...","og:url":"http://www.laweekly.com/news/laurence-tribe-tangles-with-cbs-and-la-city-hall-2396867","article:published_time":"2008-06-
04T14:16:10-07:00","article:modified_time":"2014-11-26T14:43:59-08:00","article:section":"News","og:site_name":"LA Weekly","twitter:title":"Laurence
Tribe Tangles with CBS and L.A. City Hall","twitter:description":"The United States Court of Appeals for the Ninth Circuit’s Courtroom 3 - a
miniature auditorium with comfortable, smoked salmon-colored seats -
wa...","twitter:card":"summary","twitter:site":"@laweekly"},"machine":{"os":"win xp","ram":3201225472},"@version":"1"}

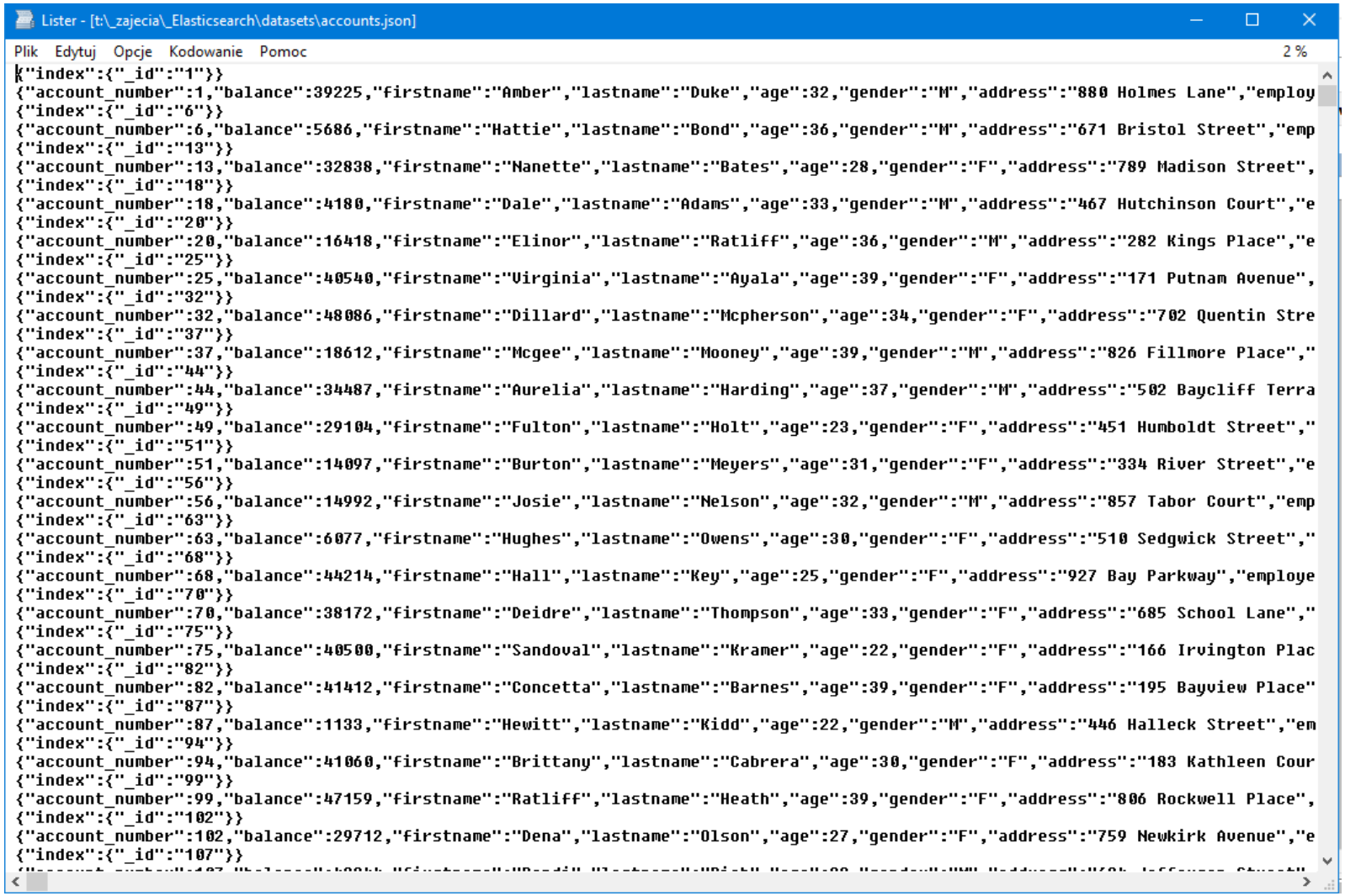
Przykładowe dane

- Podstawy, plik logs.jsonl
  - załadowanie danych do Elasticsearch

```
c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -H "Content-Type: application/x-ndjson" -XPOST "localhost:9200/_bulk
?pretty" --data-binary @logs.jsonl > logs.log
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 56.0M  100 5265k  100 50.8M   325k  3218k  0:00:16  0:00:16 --:--:--     0

c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -XGET "localhost:9200/_cat/indices?v"
health status index               uuid                   pri rep docs.count docs.deleted store.size pri.store.size
yellow open   logstash-2015.05.20 rYz-xvAvTxC-XOAdBiWZBg   5   1      19000            0       89mb           89mb
yellow open   bank                IFDCxNOzSxWPEiMD4RoFug   5   1       1000            0    475.1kb        475.1kb
yellow open   my_index            RFAOPhuIRvmzUJrYclsoqw   5   1          1            0      4.5kb          4.5kb
yellow open   logstash-2015.05.19 FW5rWPmuRdqI5R_VAT486Q   5   1      18496            0      92.7mb         92.7mb
yellow open   shakespeare         34B6PBOTSIuxcPxpZ5WtdQ   5   1     111396            0      43.1mb         43.1mb
yellow open   logstash-2015.05.18 JN95cZkTTPafzMQQ9cqE6A   5   1      18524            0       90mb           90mb
```

Przykładowe dane

- Podstawy, plik accounts.zip



Przykładowe dane

- Podstawy, plik accounts.zip
  - załadowanie danych do Elasticsearch

```
{
  "index":{
  "_id":"1"
  }
}
{
  "account_number": 1,
  "balance": 39225,
  "firstname": "Amber",
  "lastname": "Duke",
  "age": 32,
  "gender": "M",
  "address": "880 Holmes Lane",
  "employer": "Pyrami",
  "email": "amberduke@pyrami.com",
  "city": "Brogan",
  "state": "IL"
}
```

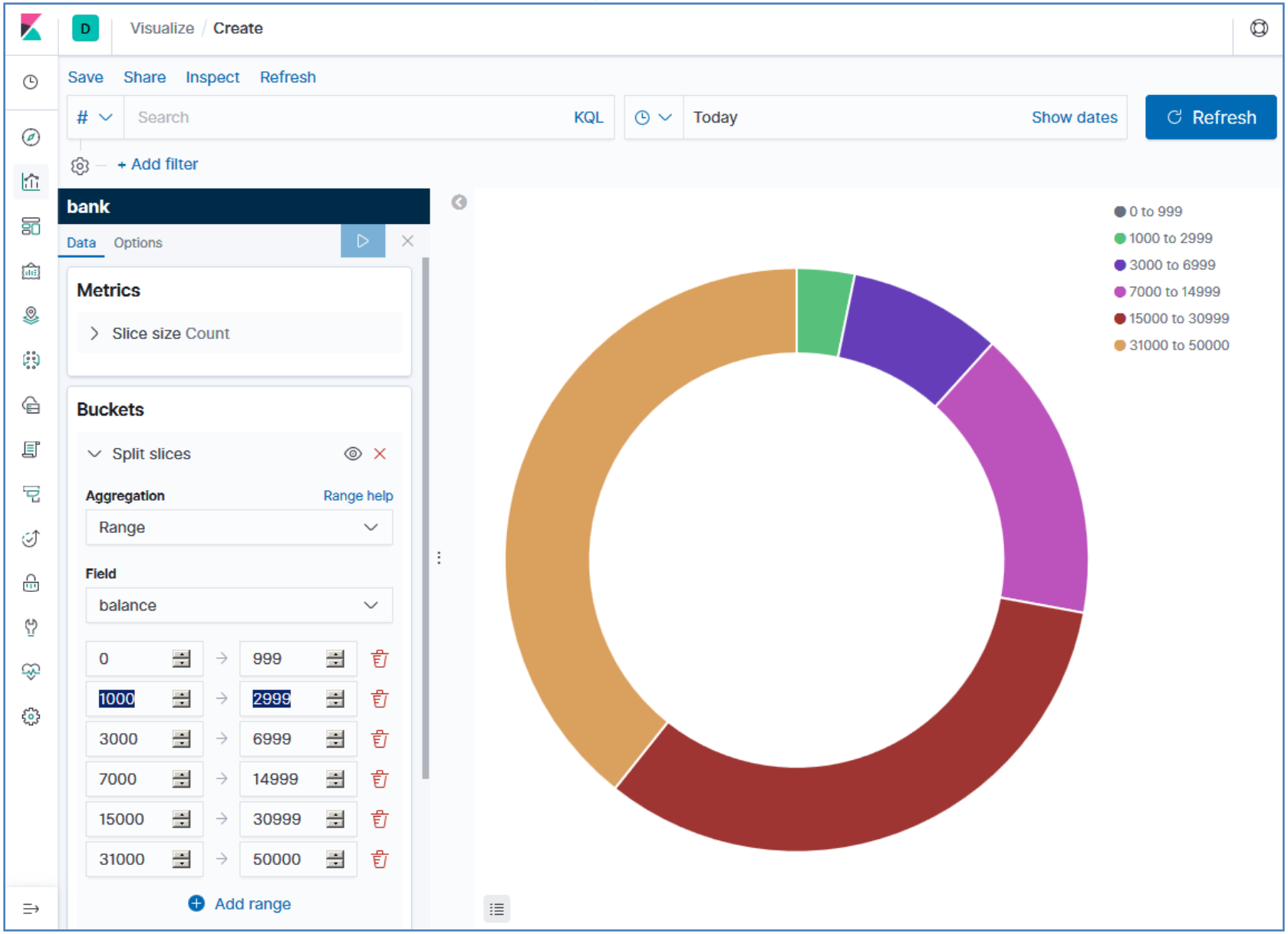
```
{
    "account_number": INT,
    "balance": INT,
    "firstname": "String",
    "lastname": "String",
    "age": INT,
    "gender": "M or F",
    "address": "String",
    "employer": "String",
    "email": "String",
    "city": "String",
    "state": "String"
}
```

```
c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -XGET "localhost:9200/_cat/indices?v"
health status index               uuid                   pri rep docs.count docs.deleted store.size pri.store.size
yellow open   logstash-2015.05.20 rYz-xvAvTxC-XOAdBiWZBg   5   1      19000            0       89mb           89mb
yellow open   bank                IFDCxNOzSxWPEiMD4RoFug   5   1       1000            0    947.7kb        947.7kb
```

Kibana, Discover

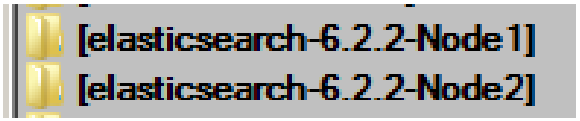
balance	account_number	address	city	age	email	employer	firstname	gender	lastname	state
48,086	32	702 Quentin Street	Vegu ita	34	dillardmcpherson@quailcom.com	Quailcom	Dillard	F	Mcpherson	IN
48,656	78	834 Amber Street	Dunbar	23	elvirapatterson@assistix.com	Assistix	Elvira	F	Patterson	TN
48,735	85	212 Irving Avenue	Kipp	20	wilcoxsellers@confrenzy.com	Confrenzy	Wilcox	M	Sellers	MT
49,671	97	512 Cumberland Walk	Frederic ktown	40	karentrujillo@tsunamia.com	Tsunamia	Karen	F	Trujillo	MO
48,868	8	699 Visitation Place	Wakulla	35	janburns@glasstep.com	Glasstep	Jan	M	Burns	AZ

Kibana, Visualize



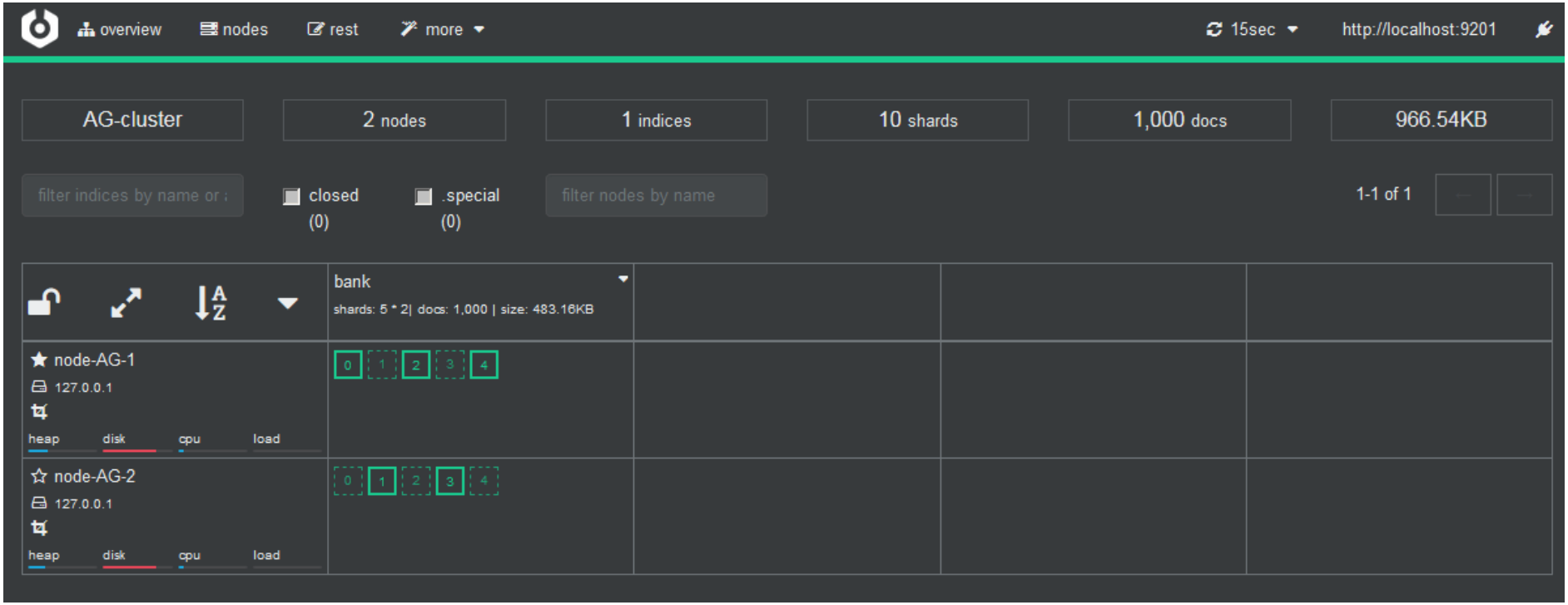
Jeden klaster, dwa node-y

- Node 1, plik elasticsearch.yml
  - cluster.name: AG-cluster
  - node.name: node-AG-1
  - http.port: 9201
- Node 2, plik elasticsearch.yml
  - cluster.name: AG-cluster
  - node.name: node-AG-2
  - http.port: 9202



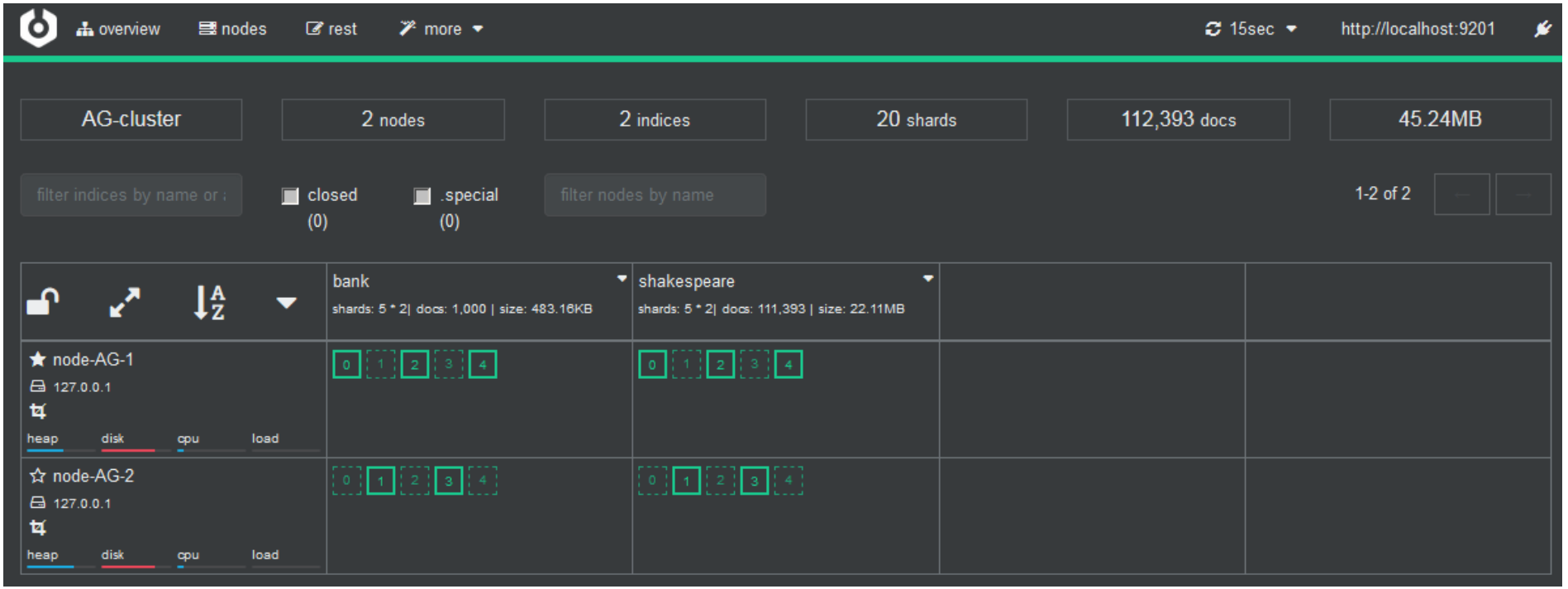
Jeden klaster, dwa node-y

- Stan po załadowaniu danych z pliku accounts.json



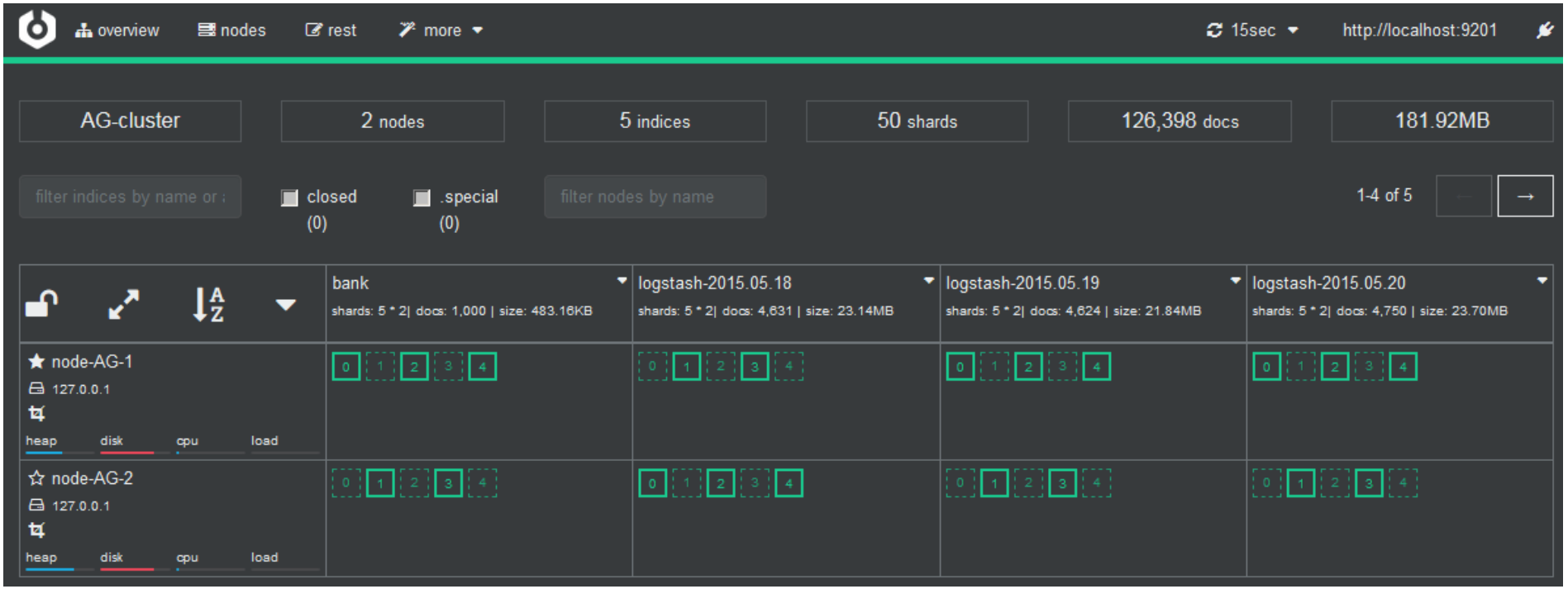
Jeden klaster, dwa node-y

- Stan po załadowaniu danych z pliku shakespeare_6.0.json



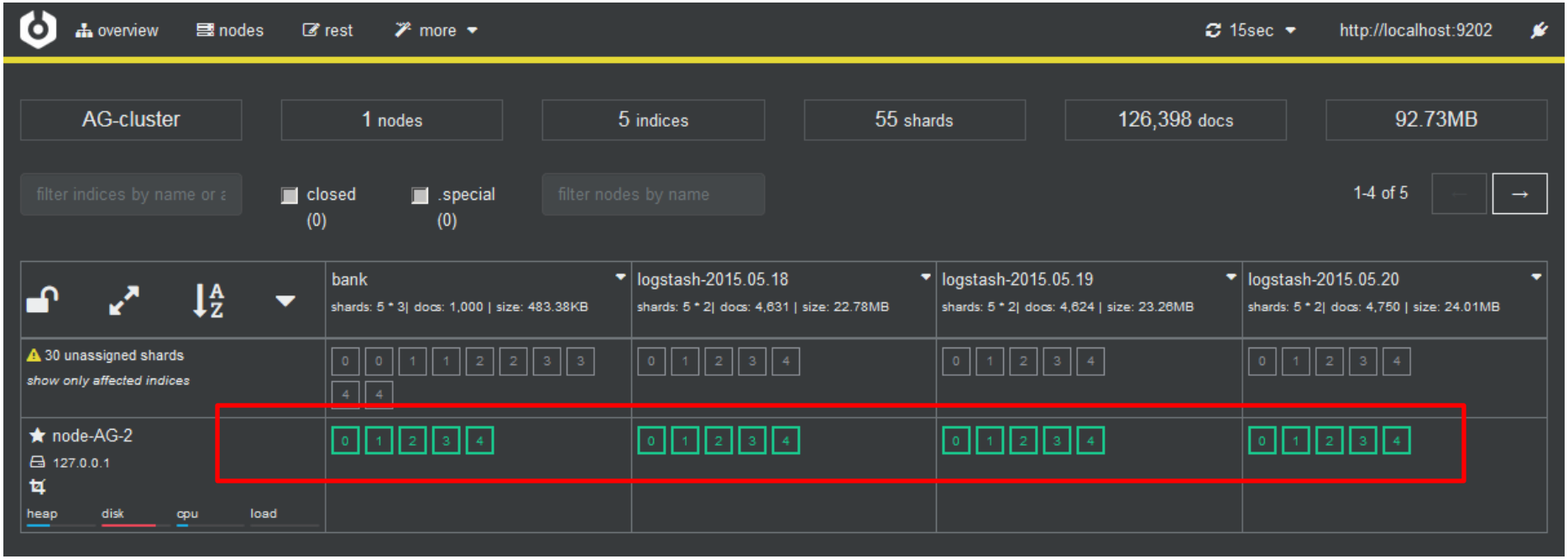
Jeden klaster, dwa node-y

- Stan po załadowaniu danych z pliku logs.jsonl



Jeden klaster, dwa node-y

- Wyłączono jeden z węzłów. Shard-y "zmigrowały" do węzła aktywnego



Jeden klaster, dwa node-y

- Tak ładowano trzy przykładowe pliki

```
c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -H "Content-Type: application/x-ndjson" -XPOST "localhost:9201/bank/
account/_bulk?pretty" --data-binary @accounts.json > account.log
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  586k  100  347k  100  239k  58918  40591  0:00:06  0:00:06 --:--:-- 27527

c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -H "Content-Type: application/x-ndjson" -XPOST "localhost:9202/shake
speare/doc/_bulk?pretty" --data-binary @shakespeare_6.0.json > shakespeare.log
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 62.6M  100 38.5M  100 24.1M  1245k   780k  0:00:31  0:00:31 --:--:-- 3703k

c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -H "Content-Type: application/x-ndjson" -XPOST "localhost:9201/_bulk
?pretty" --data-binary @logs.jsonl > logs.log
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 56.0M  100 5250k  100 50.8M   111k  1111k  0:00:46  0:00:46 --:--:--     0

c:\Program Files\Elastic\curl-7.46.0-win64\bin>
```

- Stan indeksów

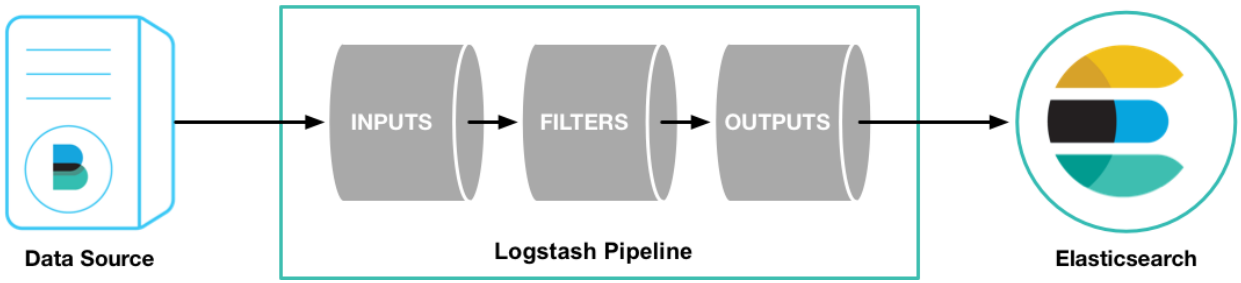
```
c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -X GET "localhost:9202/_cat/indices?v"
health status index               uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   logstash-2015.05.20 Ei7jGmzTSouhfKY1lWfNZQ   5   1       4750            0     47.1mb         23.7mb
green  open   logstash-2015.05.19 uIMAoeWOTGSgZ1JSNQnjDw   5   1       4624            0     44.3mb         21.8mb
green  open   logstash-2015.05.18 83MOZbNLSzGnjjxGdN4WWQ   5   1       4631            0     45.2mb         23.1mb
green  open   shakespeare         AqkXLUlnQRSpBfNtO5s87Q   5   1     111393            0     44.2mb         22.1mb
green  open   bank                yXN1fRsPTGqJ8CMpjwqcEg   5   1       1000            0    966.5kb        483.1kb
```

Logstash

- Konfiguracja i uruchomienie
  - rozpakować w dowolnym miejscu na dysku pliki z archiwum Logstash (zip, tar.gz, rpm, deb)
  - przygotować plik konfiguracyjny logstash.conf
  - uruchomić bin/logstash -f logstash.conf
    - gdy tylko chcemy sprawdzić poprawność działania: bin/logstash -f first-pipeline.conf --config.test_and_exit
    - gdy chcemy, aby zmiany w pliku konfiguracyjnym było od razu uwzględniane (nie trzeba robić restartu Logstash): bin/logstash -f logstash.conf --config.reload.automatic
    - wiele innych opcji uruchamiania, patrz dokumentacja (bin/logstash –h)

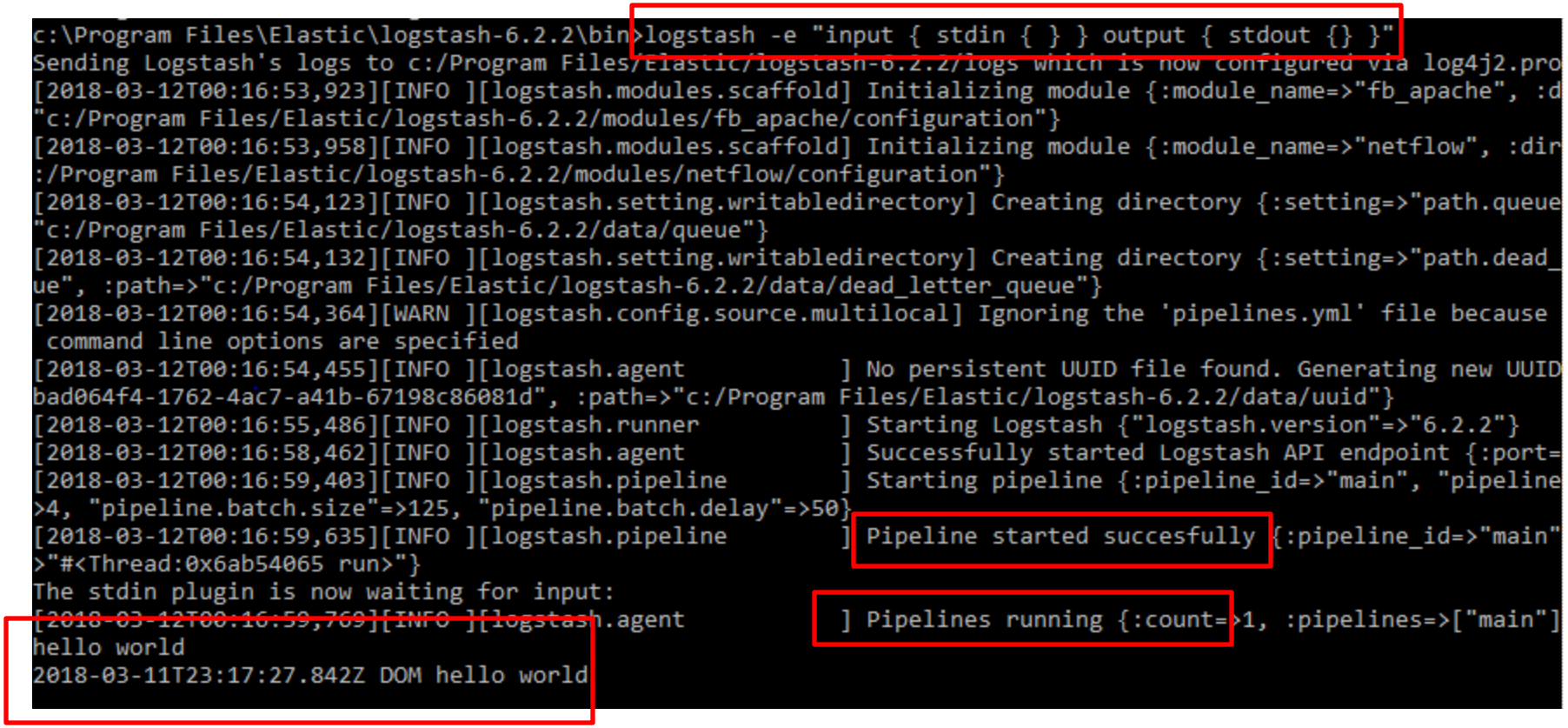
Logstash

- Prosty test działania
  - wymagane: INPUTS, OUTPUTS
  - opcjonalne: FILTERS
  - bin/logstash -e "input { stdin { } } output { stdout { } }"
    - przełącznik -e pozwala podać konfigurację wprost z linii poleceń
    - dla bardziej złożonych konfiguracji dużo lepiej pracować nie z opcją -e a z plikiem konfiguracyjnym



Logstash

- Prosty test działania
  - uwaga, dłuższy czas nic się nie dzieje, odczekać cierpliwie, nic się nie zawiesiło (oby ☺)



- W wersji 7.4.1 pojawi się top tak:

```
[2019-11-01T22:51:36,428][INFO ][logstash.agent
Hello World!
C:/Program Files/Elastic/logstash-7.4.1/vendor/bundl
 constant ::Fixnum is deprecated
{
          "host" => "DOM",
    "@timestamp" => 2019-11-01T21:53:06.866Z,
      "@version" => "1",
       "message" => "Hello World!\r"
}
```

- UWAGA, gdy błąd

  Error: Could not find or load main class Files\Elastic\logstash-7.4.1\logstash-core\lib\jars\animal-sniffer-annotations-1.14.jar;c:\Program…

  należy w pliku logstash.bat usunąć apostrofy o tutaj:

  %JAVA% %JAVA_OPTS% -cp "%CLASSPATH%" org.logstash.Logstash %*

Logstash

- Pobieranie danych z konsoli (INPUT) i wrzucanie ich do Elasticsearch (OUTPUT)
  - plik z konfiguracją ma taką postać

```
input {
    stdin { }
}

# na razie nie definiujemy filtra
filter {
}

output {
    elasticsearch {
        hosts => [ "localhost:9200" ]
    }
}
```

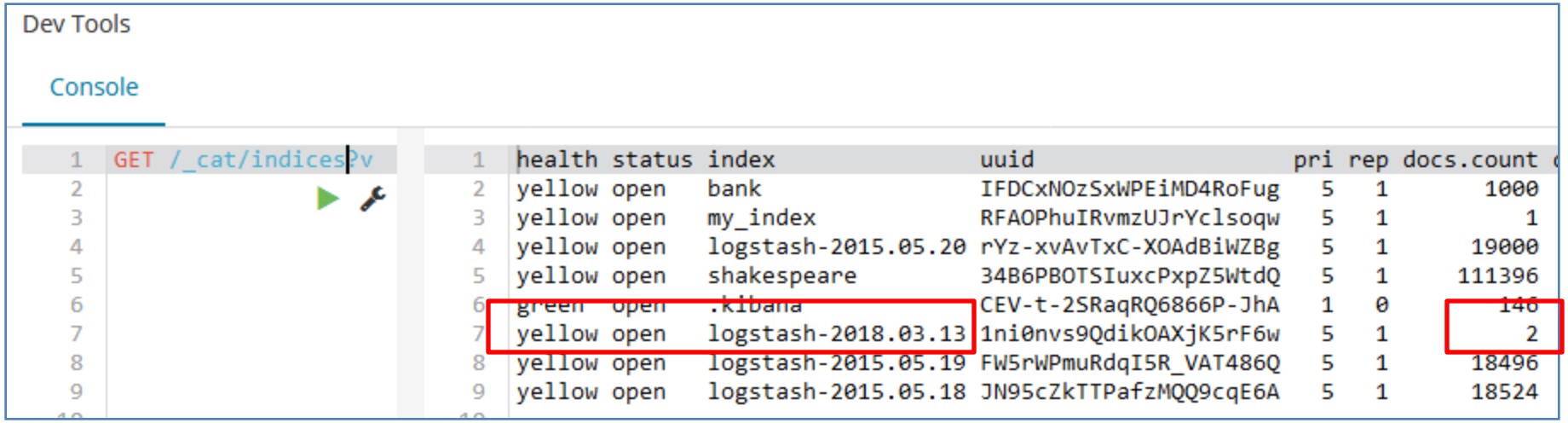
  - uruchomienie:
    bin\logstash -f pipeline-1.conf --config.reload.automatic

Logstash

```
c:\Program Files\Elastic\logstash-6.2.2>bin\logstash -f pipeline-1.conf --config.reload.automatic
Sending Logstash's logs to c:/Program Files/Elastic/logstash-6.2.2/logs which is now configured via log4j2.properties
[2018-03-13T13:15:47,458][INFO ][logstash.modules.scaffold] Initializing module {:module_name=>"fb_apache", :directory=>
"c:/Program Files/Elastic/logstash-6.2.2/modules/fb_apache/configuration"}
[2018-03-13T13:15:47,492][INFO ][logstash.modules.scaffold] Initializing module {:module_name=>"netflow", :directory=>"c
:/Program Files/Elastic/logstash-6.2.2/modules/netflow/configuration"}
[2018-03-13T13:15:47,912][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or
 command line options are specified
[2018-03-13T13:15:49,003][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"6.2.2"}
[2018-03-13T13:15:50,206][INFO ][logstash.agent           ] Successfully started Logstash API endpoint {:port=>9600}
[2018-03-13T13:15:54,090][INFO ][logstash.pipeline        ] Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=
>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50}
[2018-03-13T13:15:54,924][INFO ][logstash.outputs.elasticsearch] Elasticsearch pool URLs updated {:changes=>{:removed=>[
], :added=>[http://localhost:9200/]}}
[2018-03-13T13:15:54,941][INFO ][logstash.outputs.elasticsearch] Running health check to see if an Elasticsearch connect
ion is working {:healthcheck_url=>http://localhost:9200/, :path=>"/"}
[2018-03-13T13:15:55,284][WARN ][logstash.outputs.elasticsearch] Restored connection to ES instance {:url=>"http://local
host:9200/"}
[2018-03-13T13:15:55,378][INFO ][logstash.outputs.elasticsearch] ES Output version determined {:es_version=>nil}
[2018-03-13T13:15:55,384][WARN ][logstash.outputs.elasticsearch] Detected a 6.x and above cluster: the `type` event fiel
d won't be used to determine the document _type {:es_version=>6}
[2018-03-13T13:15:55,412][INFO ][logstash.outputs.elasticsearch] Using mapping template from {:path=>nil}
[2018-03-13T13:15:55,447][INFO ][logstash.outputs.elasticsearch] Attempting to install template {:manage_template=>{"tem
plate"=>"logstash-*", "version"=>60001, "settings"=>{"index.refresh_interval"=>"5s"}, "mappings"=>{"_default_"=>{"dynami
c_templates"=>[{"message_field"=>{"path_match"=>"message", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "
norms"=>false}}}, {"string_fields"=>{"match"=>"*", "match_mapping_type"=>"string", "mapping"=>{"type"=>"text", "norms"=>
false, "fields"=>{"keyword"=>{"type"=>"keyword", "ignore_above"=>256}}}}}], "properties"=>{"@timestamp"=>{"type"=>"date"
}, "@version"=>{"type"=>"keyword"}, "geoip"=>{"dynamic"=>true, "properties"=>{"ip"=>{"type"=>"ip"}, "location"=>{"type"=
>"geo_point"}, "latitude"=>{"type"=>"half_float"}, "longitude"=>{"type"=>"half_float"}}}}}}}}
[2018-03-13T13:15:55,521][INFO ][logstash.outputs.elasticsearch] New Elasticsearch output {:class=>"LogStash::Outputs::E
lasticSearch", :hosts=>["//localhost:9200"]}
[2018-03-13T13:15:55,720][INFO ][logstash.pipeline        ] Pipeline started succesfully {:pipeline_id=>"main", :thread=
>"#<Thread:0x21a6a5cf run>"}
The stdin plugin is now waiting for input:
[2018-03-13T13:15:55,871][INFO ][logstash.agent           ] Pipelines running {:count=>1, :pipelines=>["main"]}
POdajemy jakis tekst i naciskamy Enter
Druga linijka tekstu i ponownie Enter
```

Logstash

- Pobieranie danych z konsoli (INPUT) i wrzucanie ich do Elasticsearch (OUTPUT)



Logstash

- Pobieranie danych z pliku (INPUT) i wrzucanie ich do Elasticsearch (OUTPUT)

```
input {
    file {
        type => "TutorialLOGs"
        path => "c:/Program Files/Elastic/data/logstash-tutorial-dataset"
        start_position => "beginning"
    }
}

# na razie nie definiujemy filtra
filter {
}

output {
    elasticsearch {
        hosts => [ "localhost:9200" ]
    }

# dodatkowo wczytywane dane kierujemy też do konsoli
stdout {
        codec => rubydebug
    }
}
```

Logstash

```
health status index               uuid                   pri rep docs.count docs.deleted
yellow open   bank                IFDCxNOzSxWPEiMD4RoFug   5   1       1000            0
yellow open   my_index            RFAOPhuIRvmzUJrYclsoqw   5   1          1            0
yellow open   logstash-2015.05.20 rYz-xvAvTxC-XOAdBiWZBg   5   1      19000            0
yellow open   shakespeare         34B6PBOTSIuxcPxpZ5WtdQ   5   1     111396            0
green  open   .kibana             CEV-t-2SRaqRQ6866P-JhA   1   0        146            7
yellow open   logstash-2015.05.19 FW5rWPmuRdqI5R_VAT486Q   5   1      18496            0
yellow open   logstash-2018.03.13 ITqrtERvTa67Syd-jszc6w   5   1        100            0
yellow open   logstash-2015.05.18 JN95cZkTTPafzMQQ9cqE6A   5   1      18524            0
```

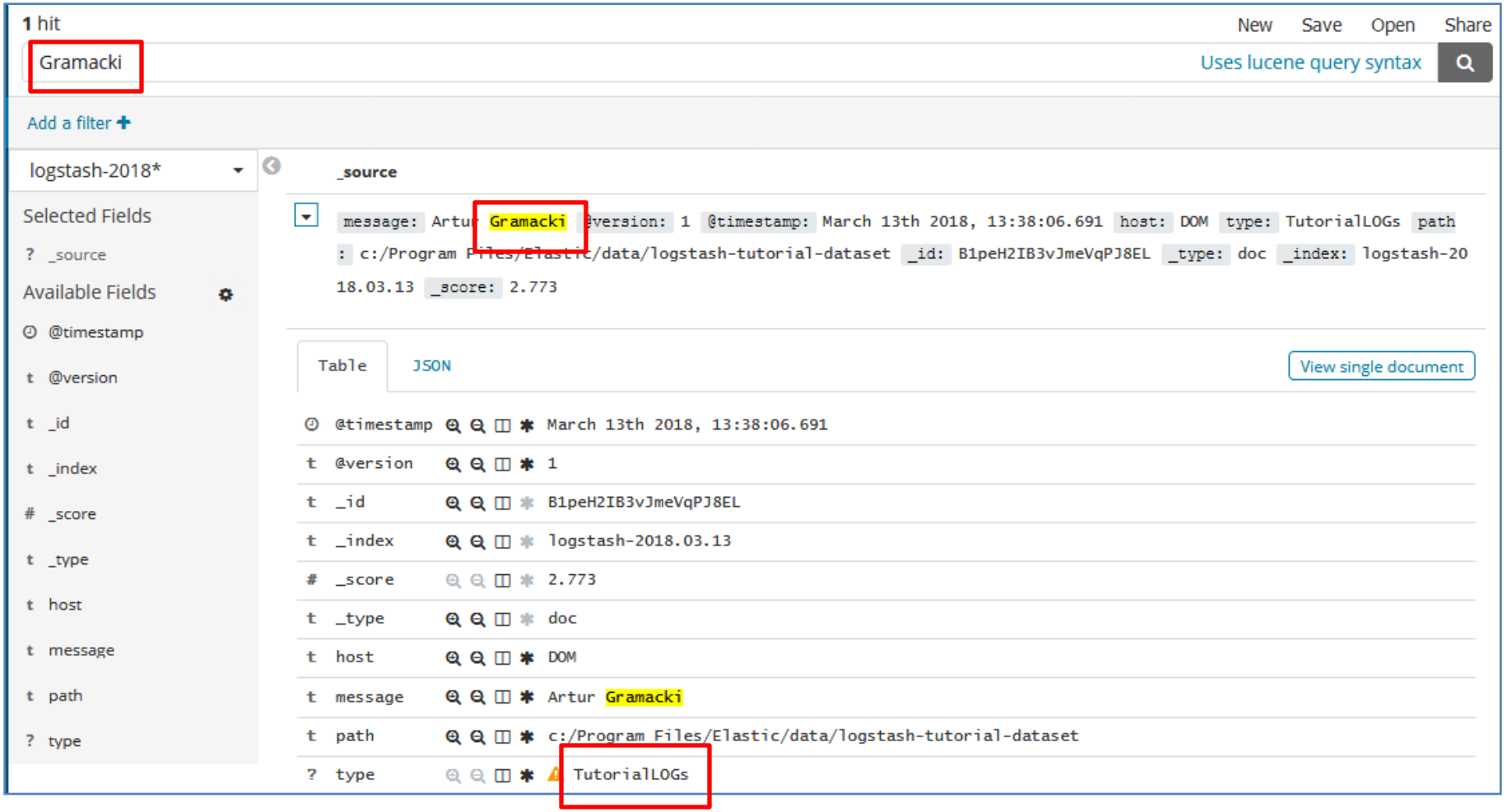
```
lasticSearch", :hosts=>["//localhost:9200"]}
[2018-03-13T13:34:42,216][INFO ][logstash.pipeline        ] Pipeline started succesfully {:pipeline_id=>"main", :thread=
>"#<Thread:0x250f1ef9 sleep>"}
[2018-03-13T13:34:42,407][INFO ][logstash.agent           ] Pipelines running {:count=>1, :pipelines=>["main"]}
{
      "@version" => "1",
       "message" => "83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] \"GET /presentations/logstash-monitorama-2013/plugin/
zoom-js/zoom.js HTTP/1.1\" 200 7697 \"http://semicomplete.com/presentations/logstash-monitorama-2013/\" \"Mozilla/5.0 (M
acintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36\"",
    "@timestamp" => 2018-03-13T12:34:43.041Z,
          "host" => "DOM",
          "type" => "TutorialLOGs",
          "path" => "c:/Program Files/Elastic/data/logstash-tutorial-dataset"
}
{
      "@version" => "1",
```

```
{
      "@version" => "1",
       "message" => "Artur Gramacki",
    "@timestamp" => 2018-03-13T12:38:06.691Z,
          "host" => "DOM",
          "type" => "TutorialLOGs",
          "path" => "c:/Program Files/Elastic/data/logstash-tutorial-dataset"
}
```

```
yellow open   logstash-2018.03.13 FmcKCt-rQzmE0jLee5ch8Q   5   1        101
```

Logstash

- Ostatnio dodany rekord widać w Kibanie



Logstash

- Pobieranie danych z pliku (INPUT) i wrzucanie ich do Elasticsearch (OUTPUT). Dodatkowo dane zostaną poddane parsowaniu (FILTER)
  - użyjemy filtru GROK (plugin do Logstash, instalowany domyślnie, nie trzeba nic samemu robić)
  - GROK poszukuje w odczytywanych danych określonych sekwencji i stara się je zinterpretować i podzielić na logiczne fragmenty. W pliku konfiguracyjnym dodajemy wpis jak niżej. Plugin będzie interpretował wpisy zgodnie ze standardem plików Apache.

```
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}"}
  }
}
```

```
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/images/kibana-search.png HTTP/1.1" 200 203023 "http://semicomplete.com/presentations/logstash-monitorama-2013/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.77 Safari/537.36"
```

Logstash

- Logi Apache

Information	Field Name
IP Address	`clientip`
User ID	`ident`
User Authentication	`auth`
timestamp	`timestamp`
HTTP Verb	`verb`
Request body	`request`
HTTP Version	`httpversion`
HTTP Status Code	`response`
Bytes served	`bytes`
Referrer URL	`referrer`
User agent	`agent`

Logstash

@timestamp		March 13th 2018, 13:34:43.098
@version		1
_id		uVpbH2IB3vJmeVqPDsBW
_index		logstash-2018.03.13
_score		1
_type		doc
host		DOM
message		87.169.99.232 - - [04/Jan/2015:05:23:41 +0000] "GET /presentations/puppet-at-loggly/puppet-at-loggly.pdf.html HTTP/1.1" 200 24747 "https://www.google.de/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"
path		c:/Program Files/Elastic/data/logstash-tutorial-dataset
type		TutorialLOGs

Logstash

- Pobieranie danych z pliku (INPUT) i wrzucanie ich do Elasticsearch (OUTPUT). Dodatkowo dane zostaną poddane parsowaniu (FILTER)
  - użyjemy dodatkowo filtru GEOIP (plugin do Logstash, instalowany domyślnie, nie trzeba nic samemu robić)
  - GEOIP będzie odczytywał numer IP i na jego podstawie będzie starał się powiązać go z lokalizacją geograficzną. W zapisanych w Elasticsearch logach zostaną dodane stosowne wpisy.

```
filter {
    grok {
        match => { "message" => "%{COMBINEDAPACHELOG}"}
    }
    geoip {
        source => "clientip"
    }
}
```

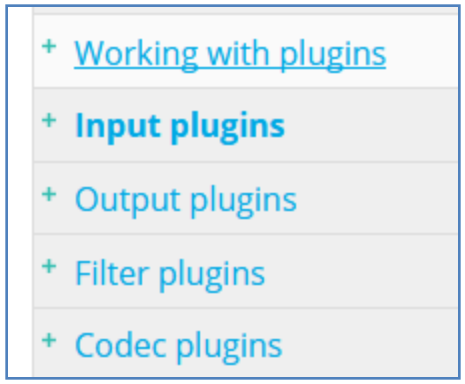
Logstash

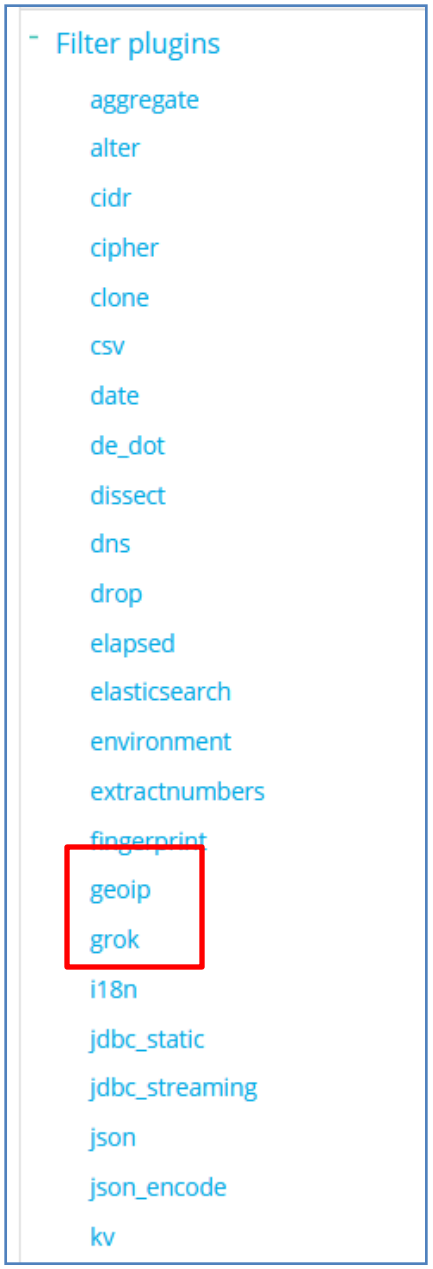
t	message	87.169.99.232 - - [04/Jan/2015:05:23:41 +0000] "GET /presentations/puppet-at-loggly/puppet-at-loggly.pdf.html HTTP/1.1" 200 24747 "https://www.google.de/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"

t	clientip	87.169.99.232
t	geoip.city_name	Berlin
t	geoip.continent_code	EU
t	geoip.country_code2	DE
t	geoip.country_code3	DE
t	geoip.country_name	Germany
	geoip.ip	87.169.99.232
#	geoip.latitude	52.521
	geoip.location	{ "lat": 52.5208, "lon": 13.2683 }
#	geoip.longitude	13.268
t	geoip.postal_code	14050
t	geoip.region_code	BE
t	geoip.region_name	Land Berlin
t	geoip.timezone	Europe/Berlin

Logstash

- Istnieje sporo innych plugin-ów do Logstash. My poznaliśmy działanie zaledwie dwóch z nich (GROK oraz GEOIP)
- W dokumentacji opisanych jest kilkadziesiąt innych plugin-ów
  - https://www.elastic.co/guide/en/logstash/current/input-plugins.html
- Sporo z nich jest w domyślnej instalacji Logstash, inne trzeba ręcznie zainstalować





Logstash

- Domyślne plugin-y (bin\logstash-plugin list --verbose)

```
logstash-codec-cef (5.0.2)
logstash-codec-collectd (3.0.8)
logstash-codec-dots (3.0.6)
logstash-codec-edn (3.0.6)
logstash-codec-edn_lines (3.0.6)
logstash-codec-es_bulk (3.0.6)
logstash-codec-fluent (3.1.5)
logstash-codec-graphite (3.0.5)
logstash-codec-json (3.0.5)
logstash-codec-json_lines (3.0.5)
logstash-codec-line (3.0.8)
logstash-codec-msgpack (3.0.7)
logstash-codec-multiline (3.0.9)
logstash-codec-netflow (3.11.0)
logstash-codec-plain (3.0.6)
logstash-codec-rubydebug (3.0.5)
```

```
logstash-output-cloudwatch (3.0.7)
logstash-output-csv (3.0.6)
logstash-output-elasticsearch (9.0.2)
logstash-output-email (4.1.0)
logstash-output-file (4.2.1)
logstash-output-graphite (3.1.4)
logstash-output-http (5.2.0)
logstash-output-kafka (7.0.8)
logstash-output-lumberjack (3.1.5)
logstash-output-nagios (3.0.5)
logstash-output-null (3.0.4)
logstash-output-pagerduty (3.0.6)
logstash-output-pipe (3.0.5)
logstash-output-rabbitmq (5.1.0)
logstash-output-redis (4.0.3)
logstash-output-s3 (4.0.13)
logstash-output-sns (4.0.6)
logstash-output-sqs (5.0.2)
logstash-output-stdout (3.1.3)
logstash-output-tcp (5.0.2)
logstash-output-udp (3.0.5)
logstash-output-webhdfs (3.0.5)
```

```
logstash-filter-aggregate (2.7.2)
logstash-filter-anonymize (3.0.6)
logstash-filter-cidr (3.1.2)
logstash-filter-clone (3.0.5)
logstash-filter-csv (3.0.8)
logstash-filter-date (3.1.9)
logstash-filter-de_dot (1.0.3)
logstash-filter-dissect (1.1.4)
logstash-filter-dns (3.0.7)
logstash-filter-drop (3.0.5)
logstash-filter-elasticsearch (3.3.0)
logstash-filter-fingerprint (3.1.2)
logstash-filter-geoip (5.0.3)
logstash-filter-grok (4.0.2)
logstash-filter-jdbc_static (1.0.0)
logstash-filter-jdbc_streaming (1.0.3)
logstash-filter-json (3.0.5)
logstash-filter-kv (4.0.3)
logstash-filter-metrics (4.0.5)
logstash-filter-mutate (3.2.0)
logstash-filter-ruby (3.1.3)
logstash-filter-sleep (3.0.6)
logstash-filter-split (3.1.6)
logstash-filter-syslog_pri (3.0.5)
logstash-filter-throttle (4.0.4)
logstash-filter-translate (3.0.4)
logstash-filter-truncate (1.0.4)
logstash-filter-urldecode (3.0.6)
logstash-filter-useragent (3.2.2)
logstash-filter-xml (4.0.5)
```

```
logstash-input-beats (5.0.6)
logstash-input-dead_letter_queue (1.1.2)
logstash-input-elasticsearch (4.2.0)
logstash-input-exec (3.1.5)
logstash-input-file (4.0.3)
logstash-input-ganglia (3.1.3)
logstash-input-gelf (3.1.0)
logstash-input-generator (3.0.5)
logstash-input-graphite (3.0.4)
logstash-input-heartbeat (3.0.5)
logstash-input-http (3.0.8)
logstash-input-http_poller (4.0.4)
logstash-input-imap (3.0.5)
logstash-input-jdbc (4.3.3)
logstash-input-kafka (8.0.4)
logstash-input-pipe (3.0.6)
logstash-input-rabbitmq (6.0.2)
logstash-input-redis (3.1.6)
logstash-input-s3 (3.2.0)
logstash-input-snmptrap (3.0.5)
logstash-input-sqs (3.0.6)
logstash-input-stdin (3.2.5)
logstash-input-syslog (3.2.4)
logstash-input-tcp (5.0.3)
logstash-input-twitter (3.0.7)
logstash-input-udp (3.2.1)
logstash-input-unix (3.0.6)
```

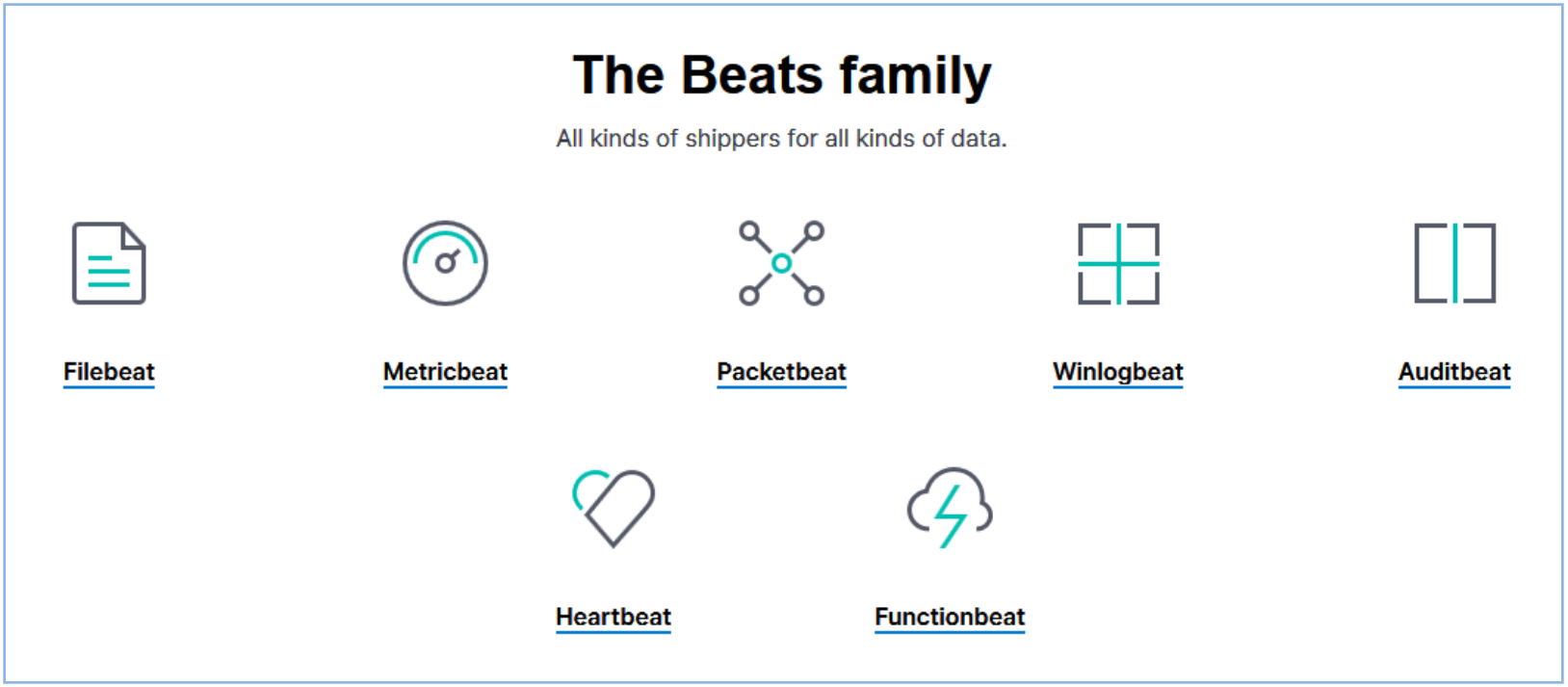
Beats (data shippers)

- Beats to współpracujące z produktami Elastic niezależne moduły (plugin-y), wspomagające akwizycję danych

- Zasady tworzenia opisane są w Beats Developer Guide
  - https://www.elastic.co/guide/en/beats/devguide/current/index.html

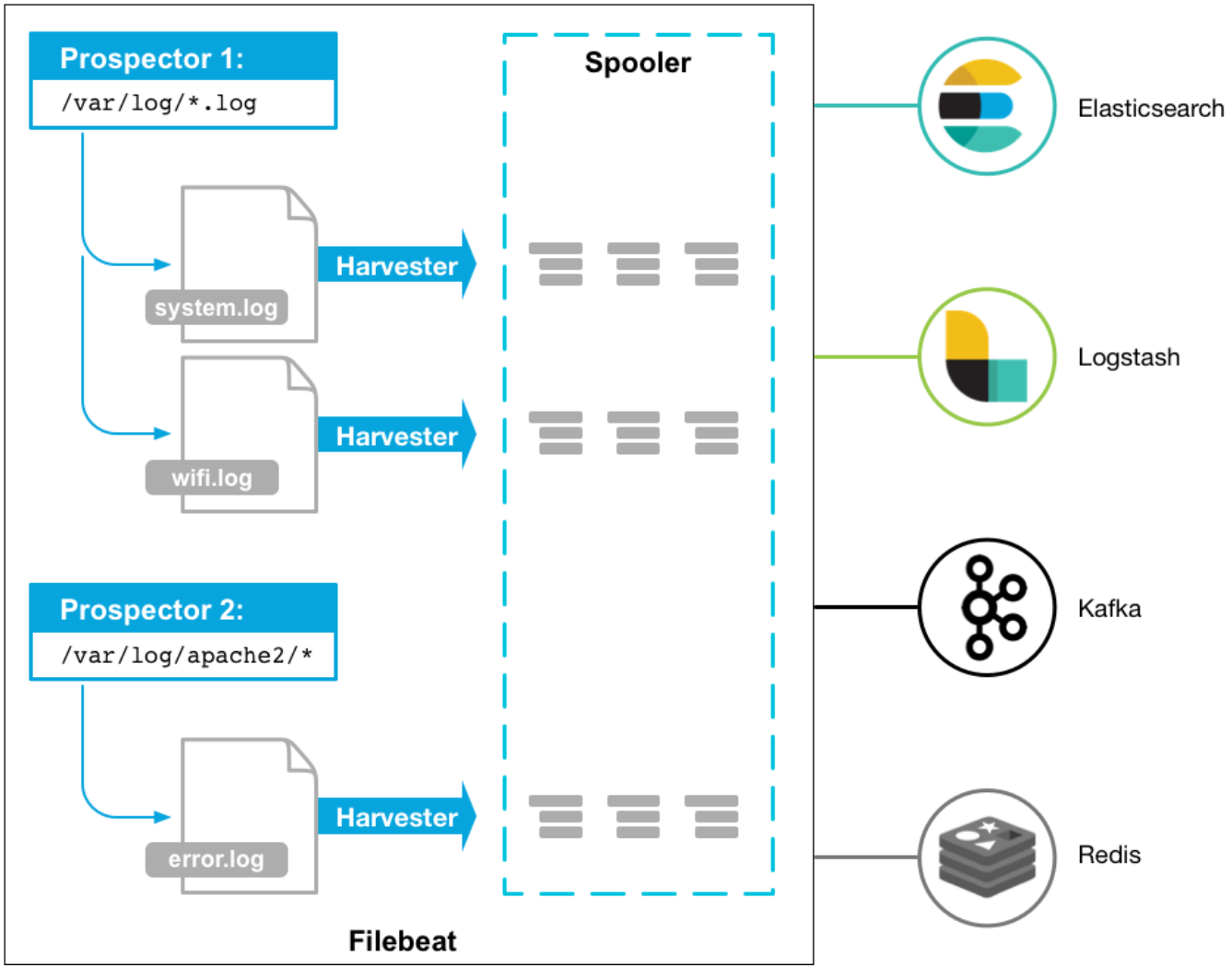
Beats (data shippers)

- Rodzina Beats



Filebeat

- Filebeat to „wysyłacz” logów (Lightweight Shipper for Logs)

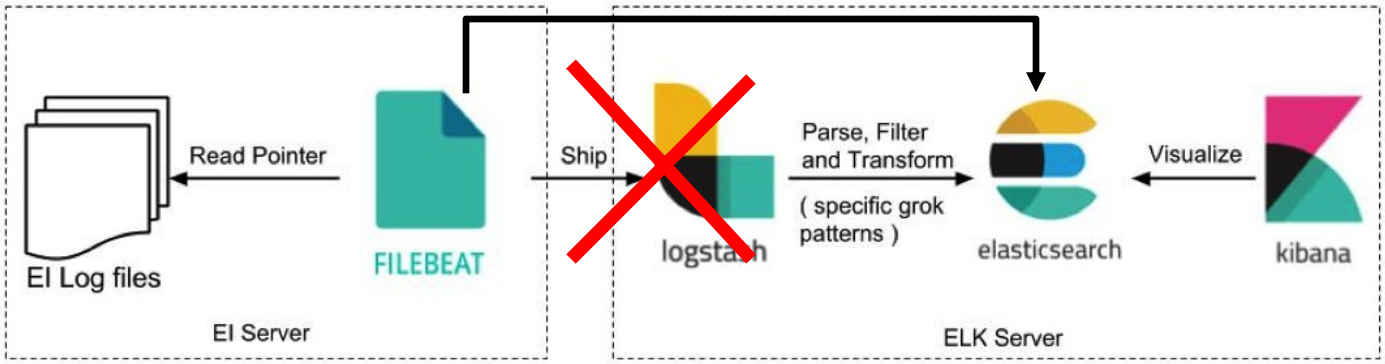


Filebeat

- Zakres ćwiczenia pierwszego:
  - wykorzystując Filebeat będziemy pobierali logi w stylu syslog (INPUT). Przykładowy plik z logami jest do pobrania ze strony https://download.elastic.co/demos/logstash/gettingstarted/logstash-tutorial.log.gz (plik: logstash-tutorial-dataset)

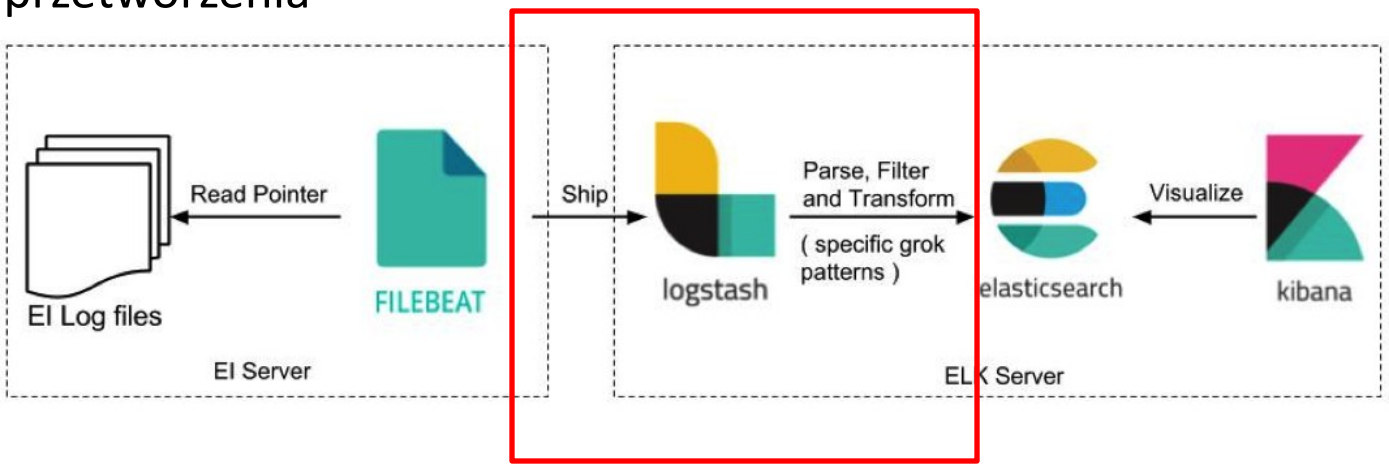
```
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/ima
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/ima
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plu
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plu
83.149.9.216 - - [04/Jan/2015:05:13:45 +0000] "GET /presentations/logstash-monitorama-2013/plu
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/ima
83.149.9.216 - - [04/Jan/2015:05:13:45 +0000] "GET /presentations/logstash-monitorama-2013/css
83.149.9.216 - - [04/Jan/2015:05:13:45 +0000] "GET /presentations/logstash-monitorama-2013/css
```

  - pobrane dane od razu będą przekazywane do Elasticsearch



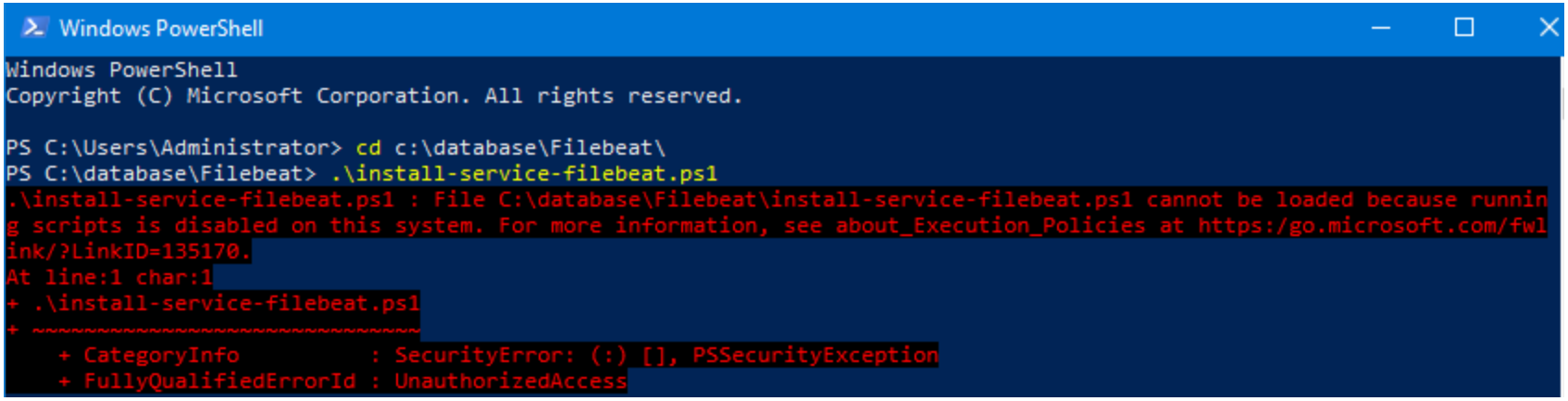
Filebeat

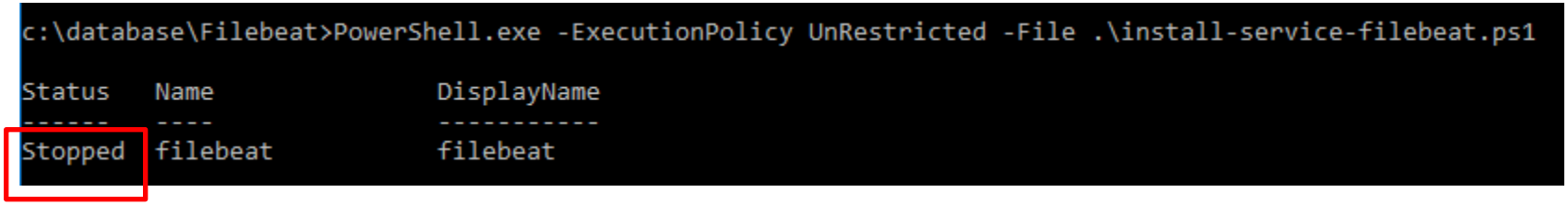
- Zakres ćwiczenia drugiego:
  - jak ćwiczenie pierwsze ale pobrane logi, zanim trafią finalnie do Elasticsearch, zostaną przekazane do Logstash celem wstępnego przetworzenia

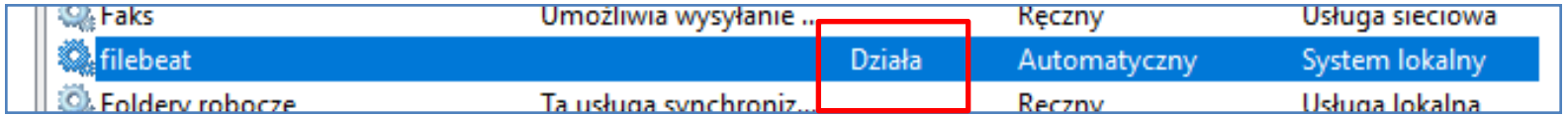


Filebeat – ćwiczenie 1

- Krok 1: pobranie Filebeat oraz instalacja go jak usługi Windows







Filebeat – ćwiczenie 1

- Krok 2: Konfiguracja Filebeat
  - Plik filebeat.yml
  - Plik filebeat.reference.yml (This file is a full configuration example documenting all non-deprecated options in comments. For a shorter configuration example, that contains only the most common options, please see filebeat.yml in the same directory)
  - krok 2.1: konfiguracja ścieżki do pliku (plików) z danymi źródłowymi (u nas są to logi Apache-a)

```
filebeat.prospectors:
- type: log
  enabled: true
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*
```

Filebeat – ćwiczenie 1

- Krok 2: Konfiguracja Filebeat
  - krok 2.2: wskazanie lokalizacji Elasticsearch-a (gdy Filebeat ma przekazywać logi bezpośrednio do Elasticsearch, z pominięciem przetwarzania przez Logstash)

```
output.elasticsearch:
  hosts: ["192.168.1.42:9200"]
```

  - krok 2.3: wskazanie lokalizacji Kibany (gdy planujemy skorzystać z dostarczonych przez Filebeat dedykowanych dashboardów)

```
setup.kibana:
  host: "localhost:5601"
```

Filebeat – ćwiczenie 1

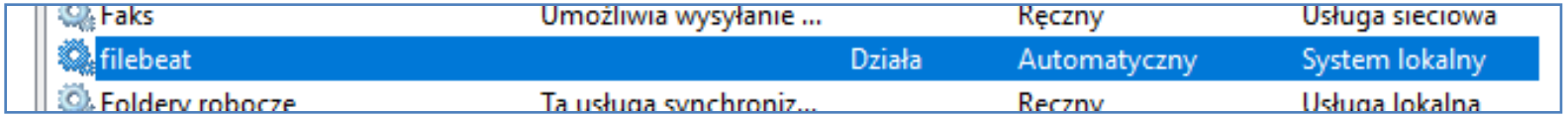
- Zawartość pliku filebeat.yml jest teraz taka:

```
#========================== Filebeat prospectors =============================
filebeat.prospectors:
- type: log
  # Change to true to enable this prospector configuration.
  enabled: true
  paths:
    - c:\Program Files\Elastic\data\*

#============================== Filebeat modules ===============================
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false

#==================== Elasticsearch template setting ==========================
setup.template.settings:
  index.number_of_shards: 3

#============================== Dashboards =====================================
setup.dashboards.enabled: true

#============================== Kibana =====================================
setup.kibana:
  host: "localhost:5601"

#-------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  hosts: ["localhost:9200"]
```

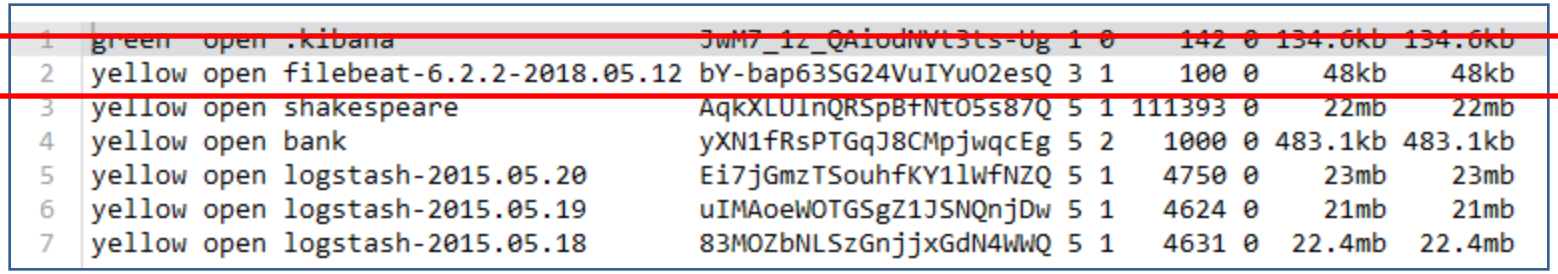
Filebeat – ćwiczenie 1

- Krok 3: uruchomienie Filebeat (uruchomienie usługi Windows). Można to oczywiście wykonać też ręcznie

```
PS > Start-Service filebeat
```

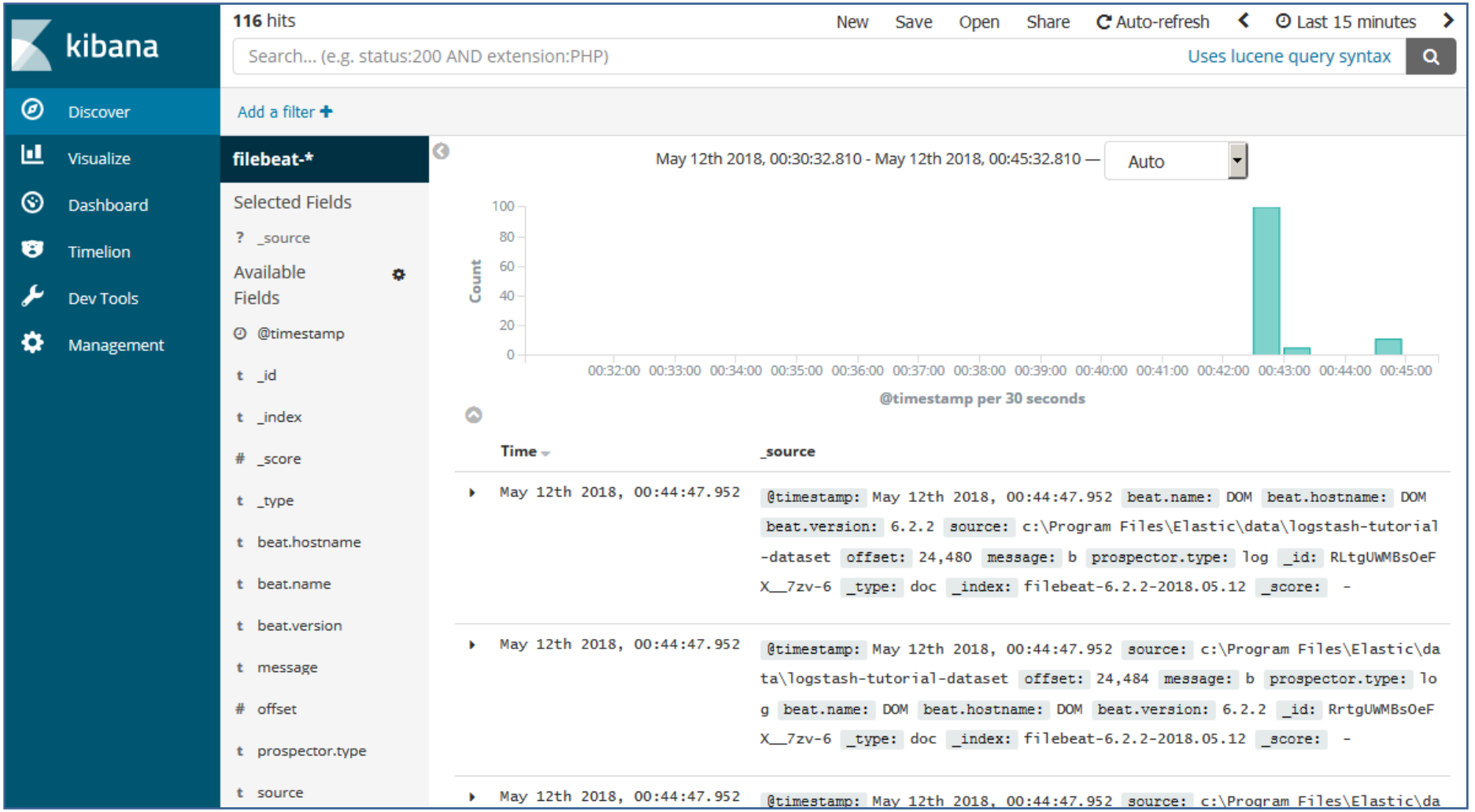


- Krok 4: W tym momencie Filebeat jest gotowy do wysyłania logów pobranych ze wskazanego wejścia (INPUT) i wysłanie ich na wskazane wyjście (OUTPUT). Oglądamy wyniki w Kibanie



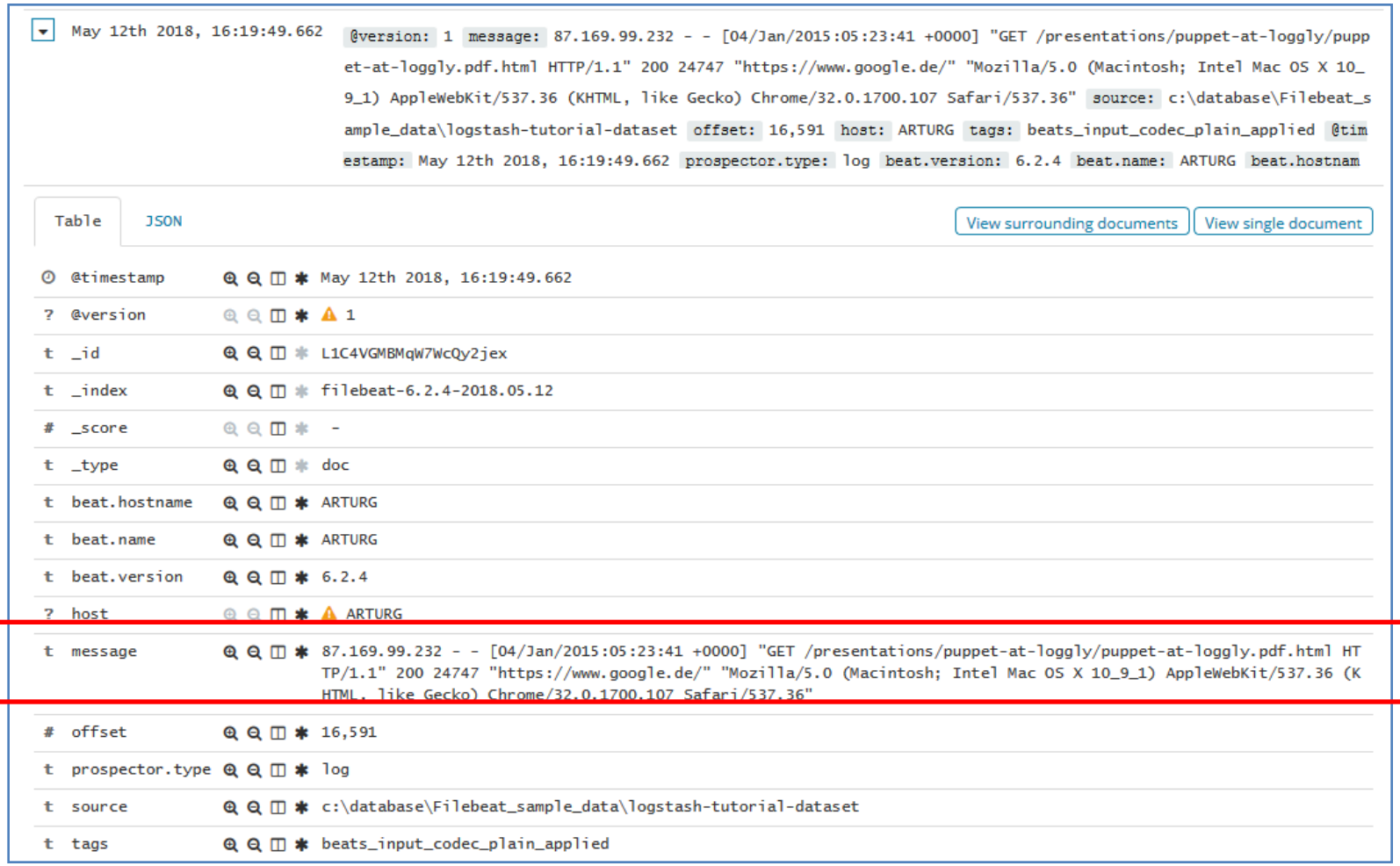
Filebeat – ćwiczenie 1

- Wyniki w Kibana



Filebeat – ćwiczenie 1

- Wynik w Kibana. Dane nie zostały w żaden sposób przetworzone. Weszły do Elasticsearch jako strumień tekstu



Filebeat – ćwiczenie 2

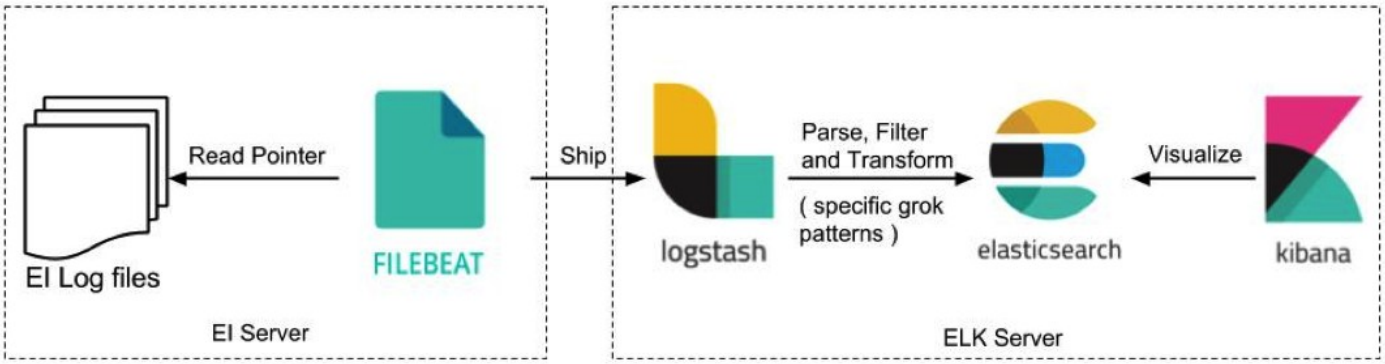
- Ćwiczenie 2, współpraca Filebeat z Logstash
  - konfiguracja Filebeat, aby korzystał z Logstash-a. Uwaga: albo Elasticsearch albo Filebeat. Oba wyjścia nie mogą być aktywne w tym samym czasie !

```
output.elasticsearch:
  hosts: ["192.168.1.42:9200"]
```

To trzeba **zakomentować**

```
#------------------------------ Logstash output ---
output.logstash:
  hosts: ["127.0.0.1:5044"]
```

To trzeba **odkomentować.** Na wskazanym porcie Logstash będzie nasłuchiwał, jakie informacje nadaje mu Filebeat.



Filebeat – ćwiczenie 2

- Krok 1: pobranie Filebeat oraz instalacja go jak usługi Windows (jak w ćwiczeniu 1)
- Krok 2: przygotowanie (poprawienie z ćwiczenia 1) pliku filebeat.yml
  - zakomentowanie wpisów w sekcji „Elasticsearch output”
  - Odkomentowanie wpisów w sekcji „Logstash output”

Filebeat – ćwiczenie 2

- Krok 3: załadowanie tzw. index template (zapisane jest tam, jak poszczególne pola z logów mają być interpretowane / analizowane). Uruchamiamy następujące polecenie

```
filebeat setup --template -E output.logstash.enabled=false
-E output.elasticsearch.hosts=["localhost:9200"]
```

```
c:\database\Filebeat>filebeat setup --template -E output.logstash.enabled=false
-E output.elasticsearch.hosts=["localhost:9200"]
Loaded index template
```

Filebeat – ćwiczenie 2

- Krok 4: Instalacja w Kibanie przykładowych dashboard-ów. Uruchamiamy następujące polecenie:

```
filebeat setup --dashboards
```

  - uwaga: gdy w pliku konfiguracyjnym będzie następujący wpis, to ładowanie będzie odbywać się automatycznie w momencie startowania usługi Filebeat

```
setup.dashboards.enabled: true
```

  - po prawidłowym załadowaniu dashboard-ów widzimy, że powstał dedykowany do tego indeks

```
health status index    uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   .kibana  7rh3-JuBSrWbm72-c_YfHg   1   0        141           11    222.4kb        222.4kb
```

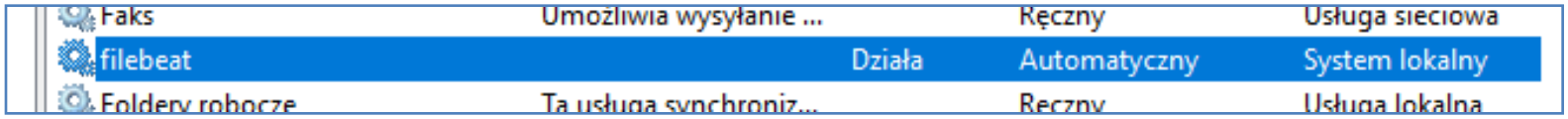
Filebeat – ćwiczenie 2

- Krok 4: przygotowanie pliku konfiguracyjnego dla Logstash-a

```
input {
  beats {
    port => 5044
  }
}

filter {
    grok {
        match => { "message" => "%{COMBINEDAPACHELOG}"}
    }
    geoip {
        source => "clientip"
    }
}

output {
  elasticsearch {
    hosts => "localhost:9200"
    manage_template => false
    index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"
  }
}
```

Filebeat – ćwiczenie 2

- Krok 5: uruchomienie całości
  - najpierw uruchamiamy usługę Filebeat



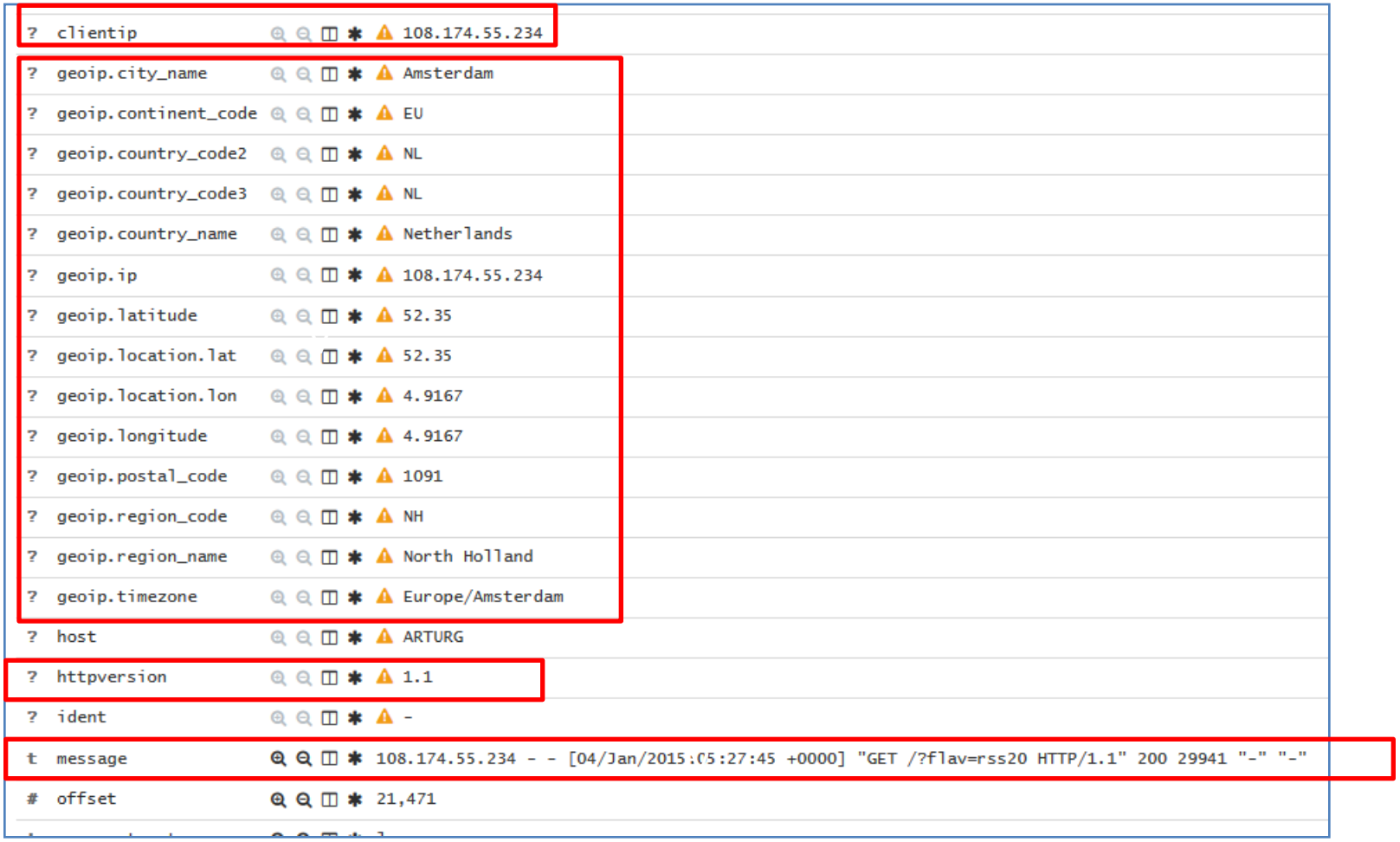
  - potem uruchamiamy Logstash

```
logstash -f logstash_filebeat.conf
```

  - w Kibanie potwierdzamy, że dane zostały zaindeksowane i przefiltrowane

Filebeat – ćwiczenie 2

- Wynik w Kibana. Dane zostały przetworzone przez filtry GROK oraz GEOIP



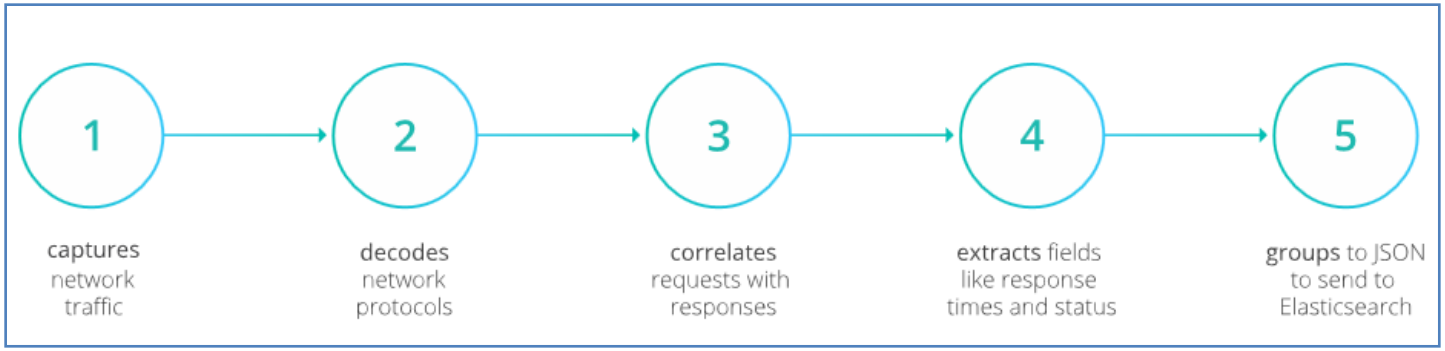
Packetbeat

- Lekki analizator pakietów sieciowych, wysyłającym dane bezpośrednio do Logstash lub Elasticsearch, umożliwiając w ten sposób monitorowanie bieżących działań w różnych aplikacjach
- W wersji 7.5 wspierane są następujące protokoły:
  ICMP (v4 and v6), DHCP (v4), DNS, HTTP, AMQP 0.9.1, Cassandra, Mysql, PostgreSQL, Redis, Thrift-RPC, MongoDB, Memcache, NFS, TLS

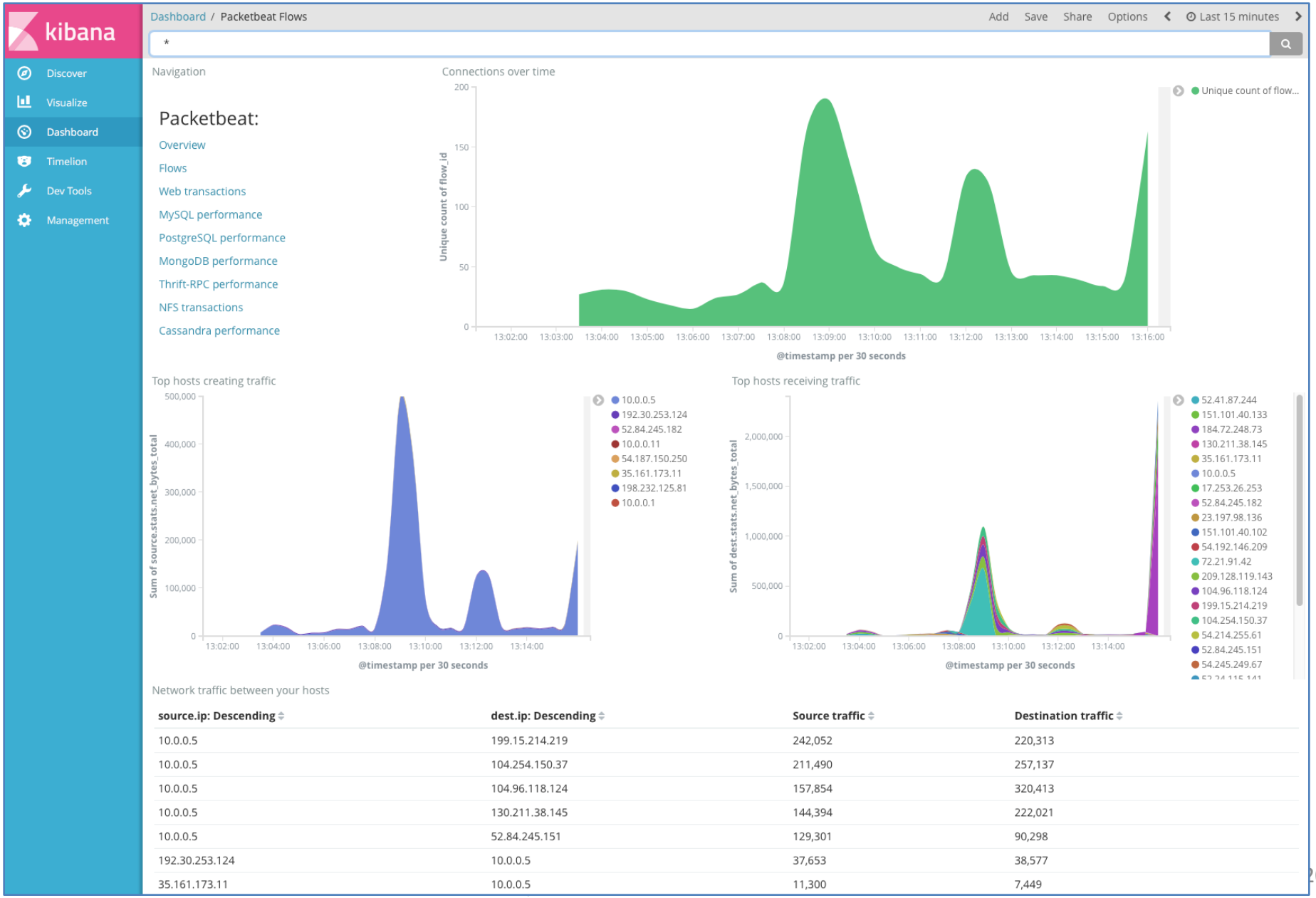
Packetbeat

- https://www.elastic.co/guide/en/beats/packetbeat/current/packetbeat-getting-started.html
  - do pracy wymagana jest biblioteka libpcap (przechwytuje ruch sieciowy)

- **Getting started with Packetbeat**
  - Step 1: Install Packetbeat
  - Step 2: Configure Packetbeat
  - Step 3: Load the index template in Elasticsearch
  - Step 4: Set up the Kibana dashboards
  - Step 5: Start Packetbeat
  - Step 6: View the sample Kibana dashboards
  - Repositories for APT and YUM



Packetbeat


kibana
Discover
Visualize
Dashboard
Timelion
Dev Tools
Management
Dashboard / Packetbeat Flows
Add
Save
Share
Options
Last 15 minutes
*
Navigation
Packetbeat:
Overview
Flows
Web transactions
MySQL performance
PostgreSQL performance
MongoDB performance
Thrift-RPC performance
NFS transactions
Cassandra performance
Connections over time
Unique count of flow_id
Unique count of flow...
@timestamp per 30 seconds
Top hosts creating traffic
Sum of source.stats.net_bytes_total
10.0.0.5
192.30.253.124
52.84.245.182
10.0.0.11
54.187.150.250
35.161.173.11
198.232.125.81
10.0.0.1
@timestamp per 30 seconds
Top hosts receiving traffic
Sum of dest.stats.net_bytes_total
52.41.87.244
151.101.40.133
184.72.248.73
130.211.38.145
35.161.173.11
10.0.0.5
17.253.26.253
52.84.245.182
23.197.98.136
151.101.40.102
54.192.146.209
72.21.91.42
209.128.119.143
104.96.118.124
199.15.214.219
104.254.150.37
54.214.255.61
52.84.245.151
54.245.249.67
@timestamp per 30 seconds
Network traffic between your hosts
source.ip: Descending
dest.ip: Descending
Source traffic
Destination traffic
10.0.0.5 199.15.214.219 242,052 220,313
10.0.0.5 104.254.150.37 211,490 257,137
10.0.0.5 104.96.118.124 157,854 320,413
10.0.0.5 130.211.38.145 144,394 222,021
10.0.0.5 52.84.245.151 129,301 90,298
192.30.253.124 10.0.0.5 37,653 38,577
35.161.173.11 10.0.0.5 11,300 7,449

SQL REST API, SQL Access

- X-Pack includes a SQL feature to execute SQL queries against Elasticsearch indices and return results in tabular formats

```
PUT /library/book/_bulk?refresh
{"index":{"_id": "Leviathan Wakes"}}
{"name": "Leviathan Wakes", "author": "James S.A. Corey", "release_date": "2011-0
{"index":{"_id": "Hyperion"}}
{"name": "Hyperion", "author": "Dan Simmons", "release_date": "1989-05-26", "page
{"index":{"_id": "Dune"}}
{"name": "Dune", "author": "Frank Herbert", "release_date": "1965-06-01", "page_c
```

```
POST /_sql?format=txt
{
    "query": "SELECT * FROM library WHERE release_date < '2000-01-01'"
}
```

```
    author     |     name      |  page_count   | release_date
---------------+---------------+---------------+------------------------
Dan Simmons    |Hyperion       |482            |1989-05-26T00:00:00.000Z
Frank Herbert  |Dune           |604            |1965-06-01T00:00:00.000Z
```

SQL CLI

- SQL z linii poleceń

```
$ ./bin/elasticsearch-sql-cli
```

```
$ ./bin/elasticsearch-sql-cli https://some.server:9200
```

```
$ ./bin/elasticsearch-sql-cli https://sql_user:strongpassword@some.server:9200
```

```
sql> SELECT * FROM library WHERE page_count > 500 ORDER BY page_count DESC;
     author      |        name        |  page_count   | release_date
-----------------+--------------------+---------------+---------------
Peter F. Hamilton|Pandora's Star      |768            |1078185600000
Vernor Vinge     |A Fire Upon the Deep|613            |707356800000
Frank Herbert    |Dune                |604            |-144720000000
Alastair Reynolds|Revelation Space    |585            |953078400000
James S.A. Corey |Leviathan Wakes     |561            |1306972800000
```

SQL Translate API

- The SQL Translate API accepts SQL in a JSON document and translates it into native Elasticsearch queries

```
POST /_sql/translate
{
    "query": "SELECT * FROM library ORDER BY page_count DESC",
    "fetch_size": 10
}
```

SQL Commands

DESCRIBE TABLE

Describe a table.

SELECT

Retrieve rows from zero or more tables.

SHOW COLUMNS

List columns in table.

SHOW FUNCTIONS

List supported functions.

SHOW TABLES

List tables available.

```
DESCRIBE
    [table identifier | 1
    [LIKE pattern]]     2
```

```
SHOW COLUMNS [ FROM | IN ]?
    [table identifier | 1
    [LIKE pattern] ]    2
```

```
SHOW FUNCTIONS [LIKE pattern?]? 1
```

```
SHOW TABLES
    [INCLUDE FROZEN]?   1
    [table identifier | 2
    [LIKE pattern ]]?   3
```

```
SELECT select_expr [, ...]
[ FROM table_name ]
[ WHERE condition ]
[ GROUP BY grouping_element [, ...] ]
[ HAVING condition]
[ ORDER BY expression [ ASC | DESC ] [, ...] ]
[ LIMIT [ count ] ]
[ PIVOT ( aggregation_expr FOR column IN ( value [ [ AS ] alias ] [, ...] ) ) ]
```

SQL Translate API

```
{
    "size" : 10,
    "docvalue_fields" : [
        {
            "field": "release_date",
            "format": "epoch_millis"
        }
    ],
    "_source": {
        "includes": [
            "author",
            "name",
            "page_count"
        ],
        "excludes": []
    },
    "sort" : [
        {
            "page_count" : {
                "order" : "desc",
                "missing" : "_first",
                "unmapped_type" : "short"
            }
        }
    ]
}
```

Response Data Formats

format	`Accept` HTTP header	Description
	Human Readable	
`csv`	`text/csv`	Comma-separated values
`json`	`application/json`	JSON (JavaScript Object Notation) human-readable format
`tsv`	`text/tab-separated-values`	Tab-separated values
`txt`	`text/plain`	CLI-like representation
`yaml`	`application/yaml`	YAML (YAML Ain't Markup Language) human-readable format
	Binary Formats	
`cbor`	`application/cbor`	Concise Binary Object Representation
`smile`	`application/smile`	Smile binary data format similar to CBOR

Response Data Formats

- TXT

```
POST /_sql?format=txt
{
    "query": "SELECT * FROM library ORDER BY page_count DESC",
    "fetch_size": 5
}
```

```
     author      |        name         |  page_count   |      release_date
-----------------+---------------------+---------------+------------------------
Peter F. Hamilton|Pandora's Star       |768            |2004-03-02T00:00:00.000Z
Vernor Vinge     |A Fire Upon the Deep|613            |1992-06-01T00:00:00.000Z
Frank Herbert    |Dune                 |604            |1965-06-01T00:00:00.000Z
Alastair Reynolds|Revelation Space     |585            |2000-03-15T00:00:00.000Z
James S.A. Corey |Leviathan Wakes      |561            |2011-06-02T00:00:00.000Z
```

Response Data Formats

- JSON

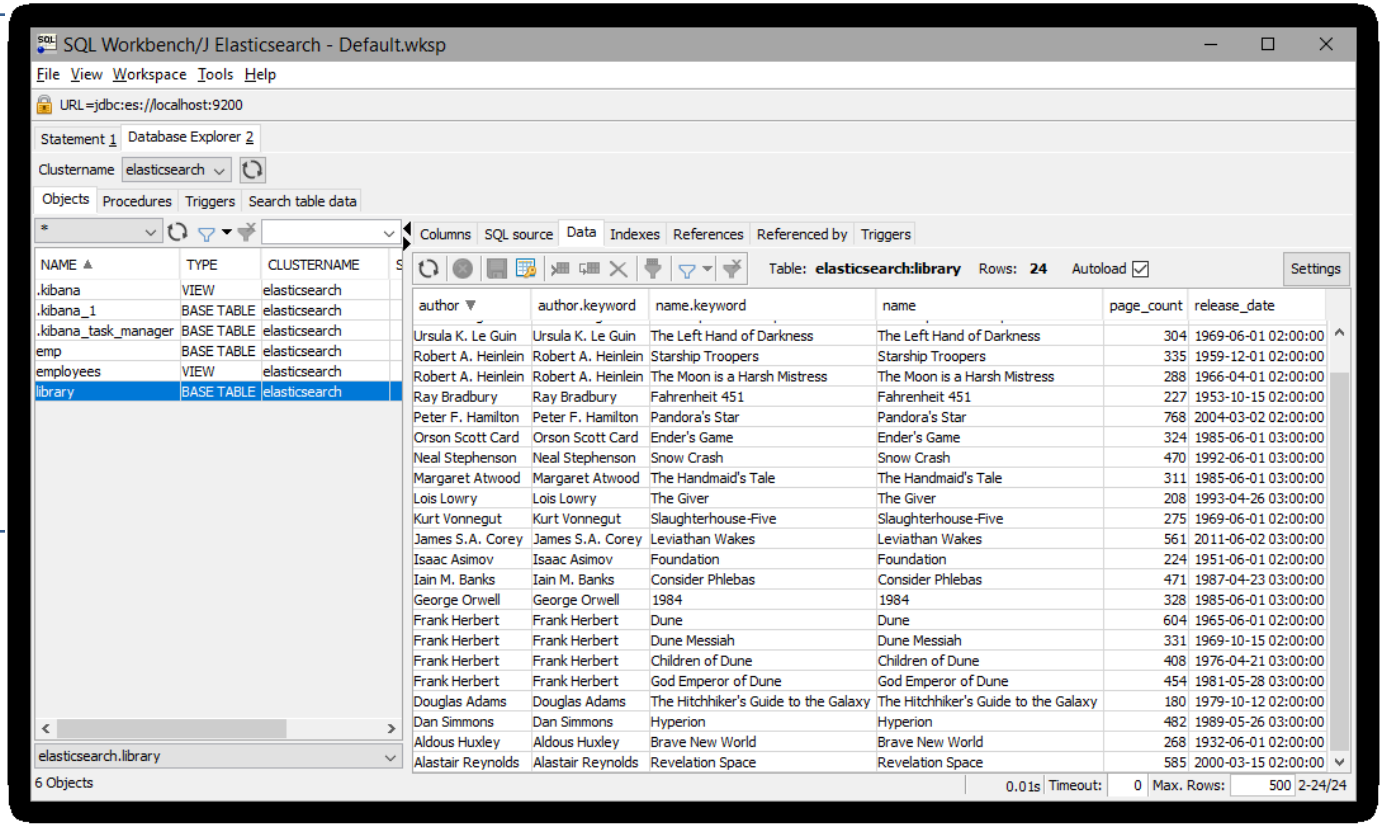
```
POST /_sql?format=json
{
    "query": "SELECT * FROM library ORDER BY page_count DESC",
    "fetch_size": 5
}
```

```
{
    "columns": [
        {"name": "author",       "type": "text"},
        {"name": "name",         "type": "text"},
        {"name": "page_count",   "type": "short"},
        {"name": "release_date", "type": "datetime"}
    ],
    "rows": [
        ["Peter F. Hamilton",  "Pandora's Star",       768, "2004-03-02T00:00:00.
        ["Vernor Vinge",       "A Fire Upon the Deep", 613, "1992-06-01T00:00:00.
        ["Frank Herbert",      "Dune",                 604, "1965-06-01T00:00:00.
        ["Alastair Reynolds",  "Revelation Space",     585, "2000-03-15T00:00:00.
        ["James S.A. Corey",   "Leviathan Wakes",      561, "2011-06-02T00:00:00.
    ],
    "cursor": "sDXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAAAEWWWdrRlVfSS1TbDYtcW9lc1FJNmlYdw==
}
```

SQL Client Applications

- Dostęp za pomocą JDBC oraz ODBC

- DBeaver
- DbVisualizer
- Microsoft Excel
- Microsoft Power BI Desktop
- Microsoft PowerShell
- MicroStrategy Desktop
- Qlik Sense Desktop
- SQuirreL SQL
- SQL Workbench
- Tableau Desktop



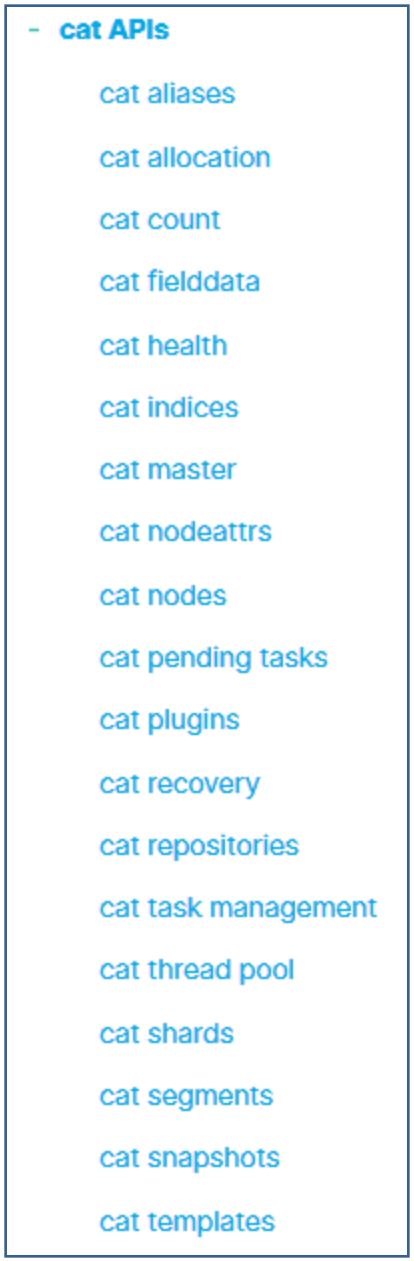
REST APIs

- API conventions
- cat APIs
- Cluster APIs
- Cross-cluster replication APIs
- Document APIs
- Enrich APIs
- Graph Explore API
- Index APIs
- Index lifecycle management APIs
- Ingest APIs

- Info API
- Licensing APIs
- Machine learning anomaly detection APIs
- Machine learning data frame analytics APIs
- Migration APIs
- Reload Search Analyzers API
- Rollup APIs
- Search APIs
- Security APIs
- Snapshot lifecycle management APIs
- Transform APIs
- Watcher APIs

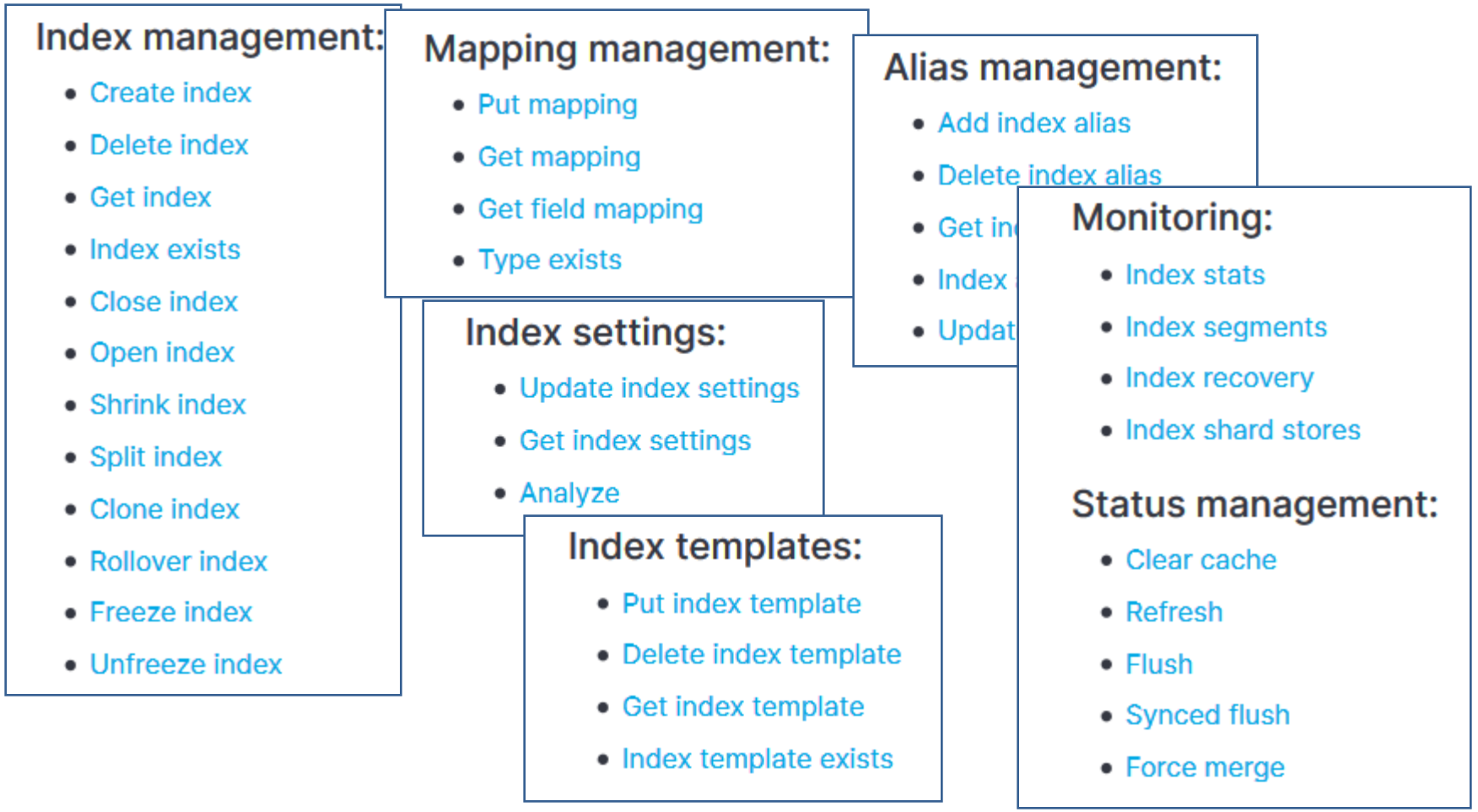
cat APIs

- Wyświetlanie różnych informacji, np:
  - o indeksach
  - o statusie klastra
  - o węzłach
  - o szardach
  - itd.



Index APIs

- Index APIs are used to manage individual indices, index settings, aliases, mappings, and index templates



Praca z indeksami

- Polecenia cURL, automatyczna konwersja w Kibana

```
curl -XPUT 'http://localhost:9200/my_index/my_type/1' -H 'Content-Type: application/json' -d '{
   "user" : "Artur Gramacki",
   "post_date" : "2018-03-16T14:12:12",
   "message" : "trying out Elastic Search"
}'
```

- Wykonujemy Copy-Paste powyższego

```
PUT /my_index/my_type/1
{
   "user" : "Artur Gramacki",
   "post_date" : "2018-03-16T14:12:12",
   "message" : "trying out Elastic Search"
}
```

```
{
  "_index": "my_index",
  "_type": "my_type",
  "_id": "1",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 0,
  "_primary_term": 1
}
```

Praca z indeksami
CRUD (ang. Create, Read, Update, Delete)

- Wyświetlenie istniejących indeksów

  GET /_cat/indices?v (parametr v (verbose) – wyświetlają się nagłówki)

- Utworzenie indeksu (Create)

  PUT /customer?pretty

- Wstawienie danych do indeksu (Create, Update)
  - gdy wcześniej nie zdefiniujemy jawnie indeksu, zostanie on utworzony automatycznie

  ```
  PUT /customer/_doc/1?pretty
  {
     "name": "John Doe"
  }
  ```

- Wyświetlenie danych o utworzonym dokumencie (Read)

  GET /customer/_doc/1?pretty
  (lub gdy tylko same dane: GET / customer /_doc/1/_source)

- Wykasowanie dokumentu (Delete)

  DELETE /customer/_doc/1?pretty

- Wykasowanie indeksu (Delete)

  DELETE /customer?pretty

Praca z indeksami

```
PUT /customer?pretty

PUT /customer/_doc/1?pretty
{
  "name": "John Doe"
}

GET /customer/_doc/1?pretty

DELETE /customer/_doc/1?pretty

DELETE /customer?pretty

GET /_cat/indices?v
```

```
{
  "acknowledged": true,
  "shards_acknowledged": true,
  "index": "customer"
}
```

```
{
  "_index": "customer",
  "_type": "_doc",
  "_id": "1",
  "_version": 1,
  "found": true,
  "_source": {
    "name": "John Doe"
  }
}
```

```
{
  "_index": "customer",
  "_type": "_doc",
  "_id": "1",
  "_version": 1,
  "result": "created",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 0,
  "_primary_term": 1
}
```

```
{
  "_index": "customer",
  "_type": "_doc",
  "_id": "1",
  "_version": 2,
  "result": "deleted",
  "_shards": {
    "total": 2,
    "successful": 1,
    "failed": 0
  },
  "_seq_no": 1,
  "_primary_term": 1
}
```

```
{
  "acknowledged": true
}
```

```
health status index    uuid                   pri rep docs.count
yellow open   customer 2Xz5cXW4TNqitgiDxm7HHw   5   1          1
```

Modyfikowanie danych

- Indeksowanie / zastępowanie danych

```
PUT /customer/_doc/1?pretty
{
  "name": "John Doe"
}
```

```
PUT /customer/_doc/1?pretty
{
  "name": "Jahn Doe"
}
```

- Nie trzeba jawnie podawać ID, wówczas zostanie on wygenerowany automatycznie

```
PUT /customer/_doc/?pretty
{
  "name": "John Doe"
}
```

Modyfikowanie danych

- Uaktualnianie dokumentów / kasowanie dokumentów

```
POST /customer/_doc/1/_update?pretty
{
  "doc": { "name": "Jane Doe" }
}
```

```
POST /customer/_doc/1/_update?pretty
{
  "doc": { "name": "Jane Doe", "age": 20 }
}
```

```
POST /customer/_doc/1/_update?pretty
{
  "script" : "ctx._source.age += 5"
}
```

```
DELETE /customer/_doc/2?pretty
```

Modyfikacja danych

- Kasowanie wszystkich dokumentów z indeksu

```
POST /customer/_delete_by_query
{
    "query": {
        "match_all": {}
    }
}
```

Modyfikowanie danych

- Tryb wsadowy
  - _bulk API

```
POST /customer/_doc/_bulk?pretty
{"index":{"_id":"1"}}
{"name": "John Doe" }
{"index":{"_id":"2"}}
{"name": "Jane Doe" }
```

```
POST /customer/_doc/_bulk?pretty
{"update":{"_id":"1"}}
{"doc": { "name": "John Doe becomes Jane Doe" } }
{"delete":{"_id":"2"}}
```

Mapowanie

- Mapowanie to proces definiowana jak dokument i jego pola są przechowywane i indeksowane w ES. Na przykład używamy mapowania aby zdefiniować:
  - które pola znakowe powinny być traktowane jako pełnotekstowe
  - które pola zawierają liczby, daty, lokalizację geograficzną
  - format daty
  - itp.
- Każdy indeks posiada zapisany sposób mapowania (mapping type). W skład mapowania wchodzą
  - pola meta (meta-fileds), _index, _type, _id, _version, _source
  - zwykłe pola

Mapowanie pól

- Typy pól
  - typy proste: text, keyword, date, long, double, boolean , ip
  - typy wspierające hierarchiczną naturę formatu JSON: object, nested
  - typy specjalizowane, jak np.: geo_point, geo_shape, or completion
- Czasami wygodnie jest zaindeksować jakieś pole na kilka sposobów
  - np. jakieś pole może być typy text (dla wsparcia przeszukiwania pełnotekstowego) oraz jako keyword (dla celów sortowania, agregacji)

Mapowanie pól

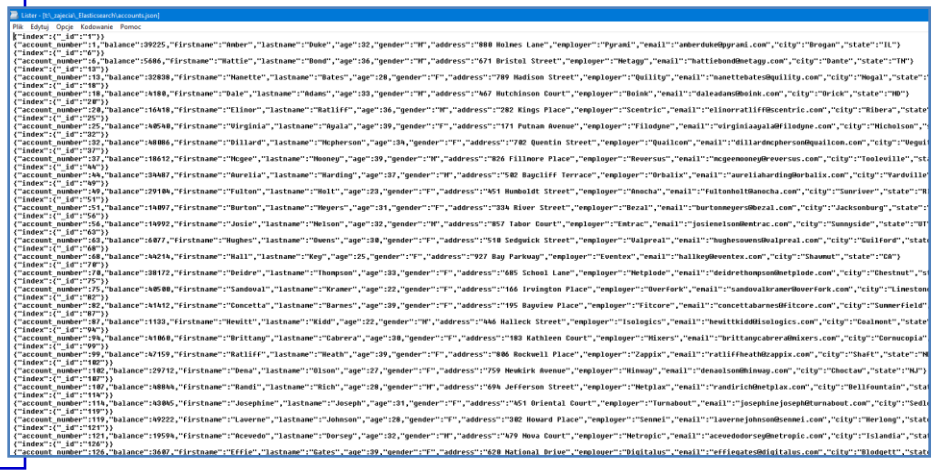
- Przykład

```
PUT my_index ❶
{
  "mappings": {
    "doc": { ❷
      "properties": { ❸
        "title":    { "type": "text"  }, ❹
        "name":     { "type": "text"  }, ❺
        "age":      { "type": "integer" }, ❻
        "created":  {
          "type":   "date", ❼
          "format": "strict_date_optional_time||epoch_millis"
        }
      }
    }
  }
}
```

Eksplorowanie danych

- Ładowanie danych z pliku

```
{
    "account_number": 0,
    "balance": 16623,
    "firstname": "Bradshaw",
    "lastname": "Mckenzie",
    "age": 29,
    "gender": "F",
    "address": "244 Columbus Place",
    "employer": "Euron",
    "email": "bradshawmckenzie@euron.com",
    "city": "Hobucken",
    "state": "CO"
}
```



```
c:\database\curl\tools\curl_x64>curl -H "Content-Type: application/json" -XPOST
"localhost:9200/bank/account/_bulk?pretty&refresh" --data-binary "@accounts.json"> log

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  616k  100  379k  100  237k  1158k   723k --:--:-- --:--:-- --:--:-- 1881k

c:\database\curl\tools\curl_x64>curl "localhost:9200/_cat/indices?v"

health status index uuid                   pri rep docs.count docs.deleted store.size ri.store.size
yellow open   bank  QDMExiDUTM-8u-QG_mVFIQ   5   1       1000            0    473.9kb        473.9kb
```

Eksplorowanie danych

- API do wyszukiwania dokumentów. Można używać 2 metod (styli)
  - parametry wysłane jako REST request URI
    - _search – szukanie
    - q=* – wszystkie dokumenty w indeksie
    - sort=account_number:asc – kierunek sortowania
    - pretty – wynik, jako "pretty-printed JSON"

```
GET /bank/_search?q=*&sort=account_number:asc&pretty
```

  - parametry wysłane jako REST request body

Eksplorowanie danych

- **took** – time in milliseconds for Elasticsearch to execute the search
- **timed_out** – tells us if the search timed out or not
- **_shards** – tells us how many shards were searched, as well as a count of the successful/failed searched shards
- **hits** – search results
- **hits.total** – total number of documents matching our search criteria
- **hits.hits** – actual array of search results (defaults to first 10 documents)
- **hits.sort** - sort key for results (missing if sorting by score)
- **hits._score** and **max_score** - ignore these fields for now


```
{
  "took": 10,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": null,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "0",
        "_score": null,
        "_source": {
          "account_number": 0,
          "balance": 16623,
          "firstname": "Bradshaw",
          "lastname": "Mckenzie",
          "age": 29,
          "gender": "F",
          "address": "244 Columbus Place",
          "employer": "Euron",
          "email": "bradshawmckenzie@euron.com",
          "city": "Hobucken",
          "state": "CO"
        },
        "sort": [
          0
        ]
      },
      {
        "_index": "bank",
        "_type": "account",
        "_id": "1",
        "_score": null,
        "_source": {
          "account_number": 1,
          "balance": 39225,
          "firstname": "Amber",
          "lastname": "Duke",
          "age": 32,
```


- Jak na poprzednim slajdzie ale tym razem w trybie REST request body (wynik identyczny)

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "sort": [
    { "account_number": "asc" }
  ]
}
```

Eksplorowanie danych

- Język zapytań (Query DSL - Domain-Specific Language)

```
GET /bank/_search
{
  "query": { "match_all": {} }
}
```

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "size": 1
}
```

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "from": 10,
  "size": 10
}
```

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "sort": { "balance": { "order": "desc" } }
}
```

Eksplorowanie danych

- Przykłady wyszukiwania

```
GET /bank/_search
{
  "query": { "match_all": {} },
  "_source": ["account_number", "balance"]
}
```


```
{
  "took": 9,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": 1,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "25",
        "_score": 1,
        "_source": {
          "account_number": 25,
          "balance": 40540
        }
      },
```



```
{
  "took": 5,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1000,
    "max_score": null,
    "hits": [
      {
        "_index": "bank",
        "_type": "account",
        "_id": "0",
        "_score": null,
        "_source": {
          "account_number": 0,
          "balance": 16623,
          "firstname": "Bradshaw",
          "lastname": "Mckenzie",
          "age": 29,
          "gender": "F",
          "address": "244 Columbus Place",
          "employer": "Euron",
          "email": "bradshawmckenzie@euron.com",
          "city": "Hobucken",
          "state": "CO"
        },
        "sort": [
          0
        ]
```

Eksplorowanie danych

- Przykłady wyszukiwania

```
GET /bank/_search
{
  "query": {
    "match": {
      "account_number": 20
    }
  }
}
```

```
GET /bank/_search
{
  "query": {
    "match": {
      "address": "mill lane"
    }
  }
}
```

```
GET /bank/_search
{
  "query": {
    "match_phrase": {
      "address": "mill lane"
    }
  }
}
```

Eksplorowanie danych

- Przykłady wyszukiwania (bool query)

```
GET /bank/_search
{
  "query": {
    "bool": {
      "must": [
        { "match": { "address": "mill" } },
        { "match": { "address": "lane" } }
      ]
    }
  }
}
```

```
GET /bank/_search
{
  "query": {
    "bool": {
      "must_not": [
        { "match": { "address": "mill" } },
        { "match": { "address": "lane" } }
      ]
    }
  }
}
```

```
GET /bank/_search
{
  "query": {
    "bool": {
      "should": [
        { "match": { "address": "mill" } },
        { "match": { "address": "lane" } }
      ]
    }
  }
}
```

```
GET /bank/_search
{
  "query": {
    "bool": {
      "must": [
        { "match": { "age": "40" } }
      ],
      "must_not": [
        { "match": { "state": "ID" } }
      ]
    }
  }
}
```

Eksplorowanie danych

- Filtrowanie danych

```
GET /bank/_search
{
  "query": {
    "bool": {
      "must": { "match_all": {} },
      "filter": {
        "range": {
          "balance": {
            "gte": 20000,
            "lte": 30000
          }
        }
      }
    }
  }
}
```

Eksplorowanie danych

- Agregowanie danych

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_state": {
      "terms": {
        "field": "state.keyword"
      }
    }
  }
}
```

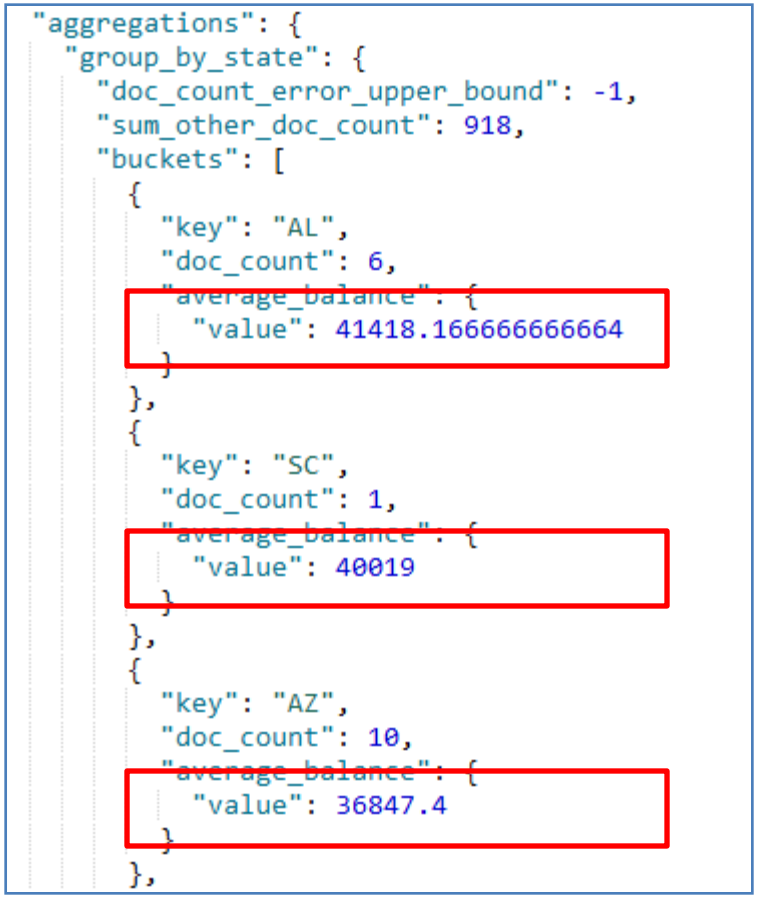
```
SELECT state, COUNT(*)
FROM bank
GROUP BY state
ORDER BY COUNT(*) DESC
```


```
"aggregations": {
  "group_by_state": {
    "doc_count_error_upper_bound": 20,
    "sum_other_doc_count": 770,
    "buckets": [
      {
        "key": "ID",
        "doc_count": 27
      },
      {
        "key": "TX",
        "doc_count": 27
      },
      {
        "key": "AL",
        "doc_count": 25
      },
      {
        "key": "MD",
        "doc_count": 25
      },
      {
        "key": "TN",
        "doc_count": 23
      },
      {
        "key": "MA",
        "doc_count": 21
      },
      {
        "key": "NC",
        "doc_count": 21
      },
      {
        "key": "ND",
        "doc_count": 21
      },
      {
        "key": "ME",
        "doc_count": 20
      },
      {
        "key": "MO",
        "doc_count": 20
      }
    ]
  }
```

Eksplorowanie danych

- Agregowanie danych, bardziej ekstremalne przykłady

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_state": {
      "terms": {
        "field": "state.keyword",
        "order": {
          "average_balance": "desc"
        }
      },
      "aggs": {
        "average_balance": {
          "avg": {
            "field": "balance"
          }
        }
      }
    }
  }
}
```



Eksplorowanie danych

- Agregowanie danych, bardziej ekstremalne przykłady

```
GET /bank/_search
{
  "size": 0,
  "aggs": {
    "group_by_age": {
      "range": {
        "field": "age",
        "ranges": [
          {
            "from": 20,
            "to": 30
          },
          {
            "from": 30,
            "to": 40
          },
          {
            "from": 40,
            "to": 50
          }
        ]
      },
      "aggs": {
        "group_by_gender": {
          "terms": {
            "field": "gender.keyword"
          },
          "aggs": {
            "average_balance": {
              "avg": {
                "field": "balance"
              }
            }
          }
        }
      }
    }
  }
}
```

```
"aggregations": {
  "group_by_age": {
    "buckets": [
      {
        "key": "20.0-30.0",
        "from": 20,
        "to": 30,
        "doc_count": 451,
        "group_by_gender": {
          "doc_count_error_upper_bound": 0,
          "sum_other_doc_count": 0,
          "buckets": [
            {
              "key": "M",
              "doc_count": 232,
              "average_balance": {
                "value": 27374.05172413793
              }
            },
            {
              "key": "F",
              "doc_count": 219,
              "average_balance": {
                "value": 25341.260273972603
              }
            }
          ]
        }
      },
      {
        "key": "30.0-40.0",
        "from": 30,
        "to": 40,
        "doc_count": 504,
        "group_by_gender": {
```

Eksplorowanie danych

- Przeszukiwanie pełnotekstowe
  - term-based queries
  - full-text queries

```
DELETE /my_index
PUT /my_index
{ "settings": { "number_of_shards": 1 }}
POST /my_index/my_type/_bulk
{"index":{"_id":1}}
{"title":"The quick brown fox"}
{"index":{"_id":2}}
{"title":"The quick brown fox jumps over the lazy dog"}
{"index":{"_id":3}}
{"title":"The quick brown fox jumps over the quick dog"}
{"index":{"_id":4}}
{"title":"Brown fox brown dog"}

GET /my_index/my_type/_search
{
  "query": {
    "match": {
      "title": "QUICK!"
    }
  }
}
```

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 3,
    "max_score": 0.44255546,
    "hits": [
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "3",
        "_score": 0.44255546,
        "_source": {
          "title": "The quick brown fox jumps over the quick dog"
        }
      },
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "1",
        "_score": 0.42327404,
        "_source": {
          "title": "The quick brown fox"
        }
      },
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "2",
        "_score": 0.3081844,
        "_source": {
          "title": "The quick brown fox jumps over the lazy dog"
        }
      }
    ]
  }
}
```

Eksplorowanie danych

- Przeszukiwanie pełnotekstowe

```
GET /my_index/my_type/_search
{
  "query": {
    "match": {
      "title": "BROWN DOG!"
    }
  }
}
```


```
  "hits": {
    "total": 4,
    "max_score": 0.58571666,
    "hits": [
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "4",
        "_score": 0.58571666,
        "_source": {
          "title": "Brown fox brown dog"
        }
      },
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "2",
        "_score": 0.39922094,
        "_source": {
          "title": "The quick brown fox jumps over the lazy dog"
        }
      },
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "3",
        "_score": 0.39922094,
        "_source": {
          "title": "The quick brown fox jumps over the quick dog"
        }
      },
      {
        "_index": "my_index",
        "_type": "my_type",
        "_id": "1",
        "_score": 0.12503365,
        "_source": {
          "title": "The quick brown fox"
        }
      }
    ]
  }
}
```

https://marcobonzanini.com/2015/02/09/phrase-match-and-proximity-search-in-elasticsearch/

Eksplorowanie danych

- Przeszukiwanie przybliżone (proximity matching)
  - używana jest tutaj macierz TDM (Term Document Matrix) zbudowana z pewnymi parametrami domyślnymi. Podejście to będzie omawiane w dalszej części kursu

Doc ID	Content
1	This is a brown fox
2	This is a brown dog
3	This dog is really brown
4	The dog is brown but this document is very very long
5	There is also a white cat
6	The quick brown fox jumps over the lazy dog

```
POST /test/articles/1
{"content":"This is a brown fox"}
POST /test/articles/2
{"content":"This is a brown dog"}
POST /test/articles/3
{"content":"This dog is really brown"}
POST /test/articles/4
{"content":"The dog is brown but this document is very very long"}
POST /test/articles/5
{"content":"There is also a white cat"}
POST /test/articles/6
{"content":"The quick brown fox jumps over the lazy dog"}
```

```
GET /test/articles/_search
{
    "query": {
        "match_phrase": {
            "content": {
                "query": "brown dog",
                "slop": 3
            }
        }
    }
}
```

Pos	Doc ID	Content	Score
1	2	This is a brown dog	0.9547657
2	4	The dog is brown but this document is very very long	0.2727902

Radzimy sobie z żywym językiem (Dealing with Human Language)

- Czy dokładne dopasowywanie przeszukiwanych dokumentów do wydanego zapytania jest dobrym podejściem?
  - Query: quick brown fox Doc: fast brown foxes
  - Query: lubimy komputery Doc: komputer da się lubić
- Problemy
  - różne niechciane znaki przystankowe i znaki diakrytyczne
  - liczba mnoga i pojedyncza
  - zdrobnienia, odmiana słów (fleksja)
  - stoplista
  - synonimy, polisemia
  - literówki, inne gramatyczne i ortograficzne błędy

Radzimy sobie z żywym językiem

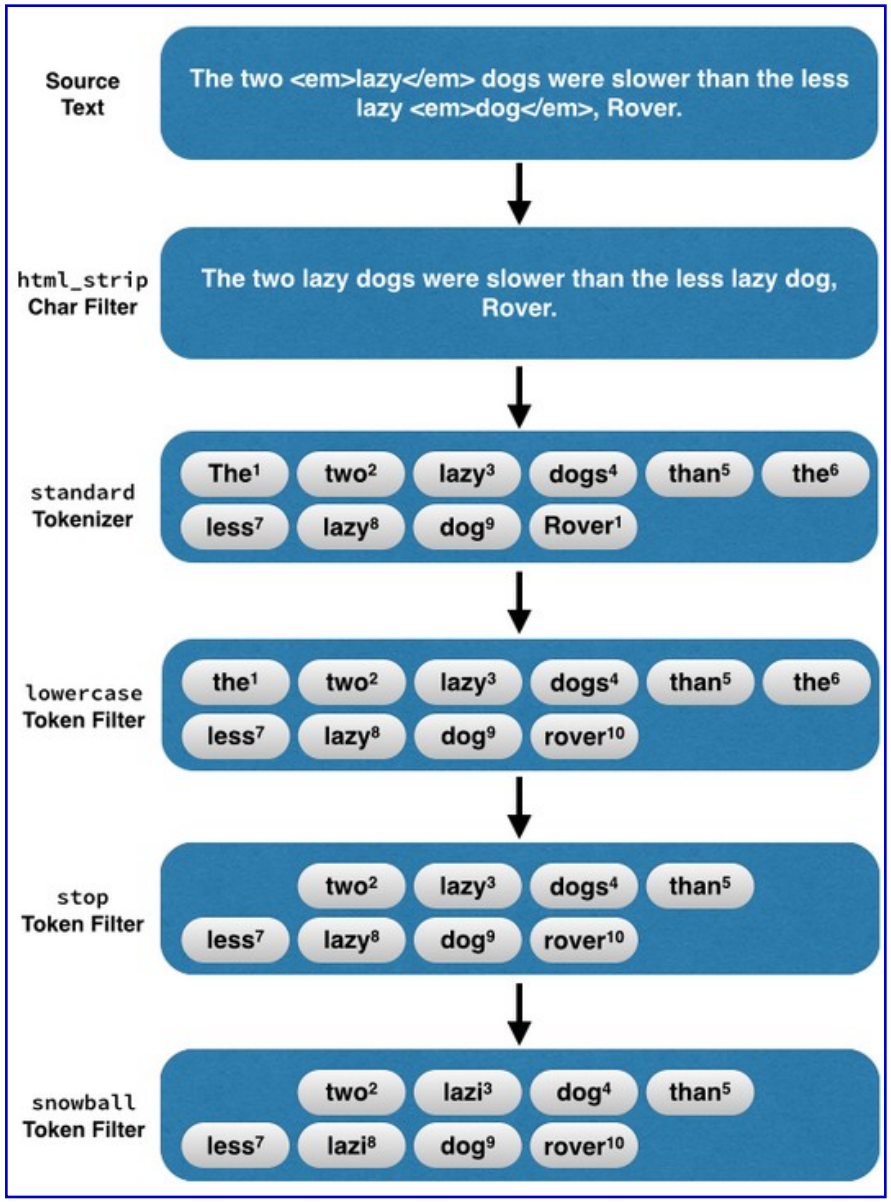
- Elasticsearch posiada wbudowane automatyczne analizatory tekstu (language analyzers) posiadające wsparcie dla wielu języków (ale na razie nie j. polski ☹)
  - Arabic, Armenian, Basque, Brazilian, Bulgarian, Catalan, Chinese, Czech, Danish,Dutch, English, Finnish, French, Galician, German, Greek, Hindi, Hungarian, Indonesian, Irish, Italian, Japanese, Korean, Kurdish, Norwegian, Persian, Portuguese, Romanian, Russian, Spanish, Swedish, Turkish, and Thai
- Wykonywane zadania
  - podział na pojedyncze tokeny, np. The quick brown foxes → [The, quick, brown, foxes]
  - zamiana na małe litery, np. The → the
  - usunięcie słów ze stoplisty, np. [The, quick, brown, foxes] → [quick, brown, foxes]
  - konwersja tokenów do postaci rdzenia znaczeniowego, np. foxes → fox
  - inne transformacje typowe dla danego języka, np. John's → john, außerst → ausserst, l'eglise → eglis

Radzimy sobie z żywym językiem

- Działanie analizatora tekstów
  - podstawowy cel: podział tekstu na tokeny

  - CharacterFilters: np. zamiana na małe litery, usunięcie znaczników HTML, można uruchomić wiele filtrów działających sekwencyjnie
  - Tokenizer: z przekształconego w poprzednik kroku strumienia tekstu wyodrębnia tokeny. Dodatkowo zapamiętywana jest pozycja tokenu w stringu (przydaje się przy bardziej zaawansowanych opcjach wyszukiwania)
  - TokenFilters: dalsza obróbka tokenów, np. zastosowanie stemmer-a, usunięcie tokenów krótszych niż podana długość, rozpoznanie synonimów, usunięcie apostrofów, usunięcie tokenów ze stoplisty, można uruchomić wiele filtrów działających sekwencyjnie

Radzimy sobie z żywym językiem



Radzimy sobie z żywym językiem

- Analyzers
  - Configuring built-in analyzers
  - Standard Analyzer
  - Simple Analyzer
  - Whitespace Analyzer
  - Stop Analyzer
  - Keyword Analyzer
  - Pattern Analyzer
  - Language Analyzers
  - Fingerprint Analyzer
  - Custom Analyzer

- **Tokenizers**
  - Standard Tokenizer
  - Letter Tokenizer
  - Lowercase Tokenizer
  - Whitespace Tokenizer
  - UAX URL Email Tokenizer
  - Classic Tokenizer
  - Thai Tokenizer
  - NGram Tokenizer
  - Edge NGram Tokenizer
  - Keyword Tokenizer
  - Pattern Tokenizer
  - Simple Pattern Tokenizer
  - Simple Pattern Split Tokenizer
  - Path Hierarchy Tokenizer

- **Token Filters**
  - Standard Token Filter
  - ASCII Folding Token Filter
  - Flatten Graph Token Filter
  - Length Token Filter
  - Lowercase Token Filter
  - Uppercase Token Filter
  - NGram Token Filter
  - Edge NGram Token Filter
  - Porter Stem Token Filter
  - Shingle Token Filter
  - Stop Token Filter
  - Word Delimiter Token Filter
  - Word Delimiter Graph Token Filter
  - Stemmer Token Filter
  - Stemmer Override Token Filter
  - Keyword Marker Token Filter
  - Keyword Repeat Token Filter
  - KStem Token Filter
  - Snowball Token Filter
  - Phonetic Token Filter

- Synonym Token Filter
- Synonym Graph Token Filter
- Compound Word Token Filters
- Reverse Token Filter
- Elision Token Filter
- Truncate Token Filter
- Unique Token Filter
- Pattern Capture Token Filter
- Pattern Replace Token Filter
- Trim Token Filter
- Limit Token Count Token Filter
- Hunspell Token Filter
- Common Grams Token Filter
- Normalization Token Filter
- CJK Width Token Filter
- CJK Bigram Token Filter
- Delimited Payload Token Filter
- Keep Words Token Filter
- Keep Types Token Filter
- Classic Token Filter
- Apostrophe Token Filter
- Decimal Digit Token Filter
- Fingerprint Token Filter
- Minhash Token Filter

Radzimy sobie z żywym językiem

- Wbudowane analizatory
  - Standard Analyzer The standard analyzer divides text into terms on word boundaries, as defined by the Unicode Text Segmentation algorithm. It removes most punctuation, lowercases terms, and supports removing stop words
  - Simple Analyzer The simple analyzer divides text into terms whenever it encounters a character which is not a letter. It lowercases all terms
  - Whitespace Analyzer The whitespace analyzer divides text into terms whenever it encounters any whitespace character. It does not lowercase terms
  - Stop Analyzer The stop analyzer is like the simple analyzer, but also supports removal of stop words
  - Keyword Analyzer The keyword analyzer is a “noop” analyzer that accepts whatever text it is given and outputs the exact same text as a single term
  - Pattern Analyzer The pattern analyzer uses a regular expression to split the text into terms. It supports lower-casing and stop words
  - Language Analyzers Elasticsearch provides many language-specific analyzers like english or french
  - Fingerprint Analyzer The fingerprint analyzer is a specialist analyzer which creates a fingerprint which can be used for duplicate detection

Radzimy sobie z żywym językiem

- Analizatory użytkowników

❶ We define the `std_english` analyzer to be based on the `standard` analyzer, but configured to remove the pre-defined list of English stopwords.

❷ ❹ The `my_text` field uses the `standard` analyzer directly, without any configuration. No stop words will be removed from this field. The resulting terms are: `[ the, old, brown, cow ]`

❸ ❺ The `my_text.english` field uses the `std_english` analyzer, so English stop words will be removed. The resulting terms are: `[ old, brown, cow ]`

```
PUT my_index
{
  "settings": {
    "analysis": {
      "analyzer": {
        "std_english": { ❶
          "type":      "standard",
          "stopwords": "_english_"
        }
      }
    }
  },
  "mappings": {
    "_doc": {
      "properties": {
        "my_text": {
          "type":     "text",
          "analyzer": "standard", ❷
          "fields": {
            "english": {
              "type":     "text",
              "analyzer": "std_english" ❸
            }
          }
        }
      }
    }
  }
}
```

```
POST my_index/_analyze
{
  "field": "my_text", ❹
  "text": "The old brown cow"
}

POST my_index/_analyze
{
  "field": "my_text.english", ❺
  "text": "The old brown cow"
}
```

Radzimy sobie z żywym językiem

- Przykład użycia, Standard Analyzer

```
POST _analyze
{
  "analyzer": "standard",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

```
[ the, 2, quick, brown, foxes, jumped, over, the, lazy, dog's, bone ]
```

```
PUT my_index
{
  "settings": {
    "analysis": {
      "analyzer": {
        "my_english_analyzer": {
          "type": "standard",
          "max_token_length": 5,
          "stopwords": "_english_"
        }
      }
    }
  }
}

POST my_index/_analyze
{
  "analyzer": "my_english_analyzer",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

```
[ 2, quick, brown, foxes, jumpe, d, over, lazy, dog's, bone ]
```

Radzimy sobie z żywym językiem

- Wbudowane tokenizatory
  - Standard Tokenizer The standard tokenizer divides text into terms on word boundaries, as defined by the Unicode Text Segmentation algorithm. It removes most punctuation symbols. It is the best choice for most languages
  - Letter Tokenizer The letter tokenizer divides text into terms whenever it encounters a character which is not a letter
  - Lowercase Tokenizer The lowercase tokenizer, like the letter tokenizer, divides text into terms whenever it encounters a character which is not a letter, but it also lowercases all terms
  - Whitespace Tokenizer The whitespace tokenizer divides text into terms whenever it encounters any whitespace character
  - UAX URL Email Tokenizer The uax_url_email tokenizer is like the standard tokenizer except that it recognises URLs and email addresses as single tokens
  - Classic Tokenizer The classic tokenizer is a grammar based tokenizer for the English Language
  - Thai Tokenizer The thai tokenizer segments Thai text into words

Radzimy sobie z żywym językiem

- Wbudowane tokenizatory
  - N-Gram Tokenizer The ngram tokenizer can break up text into words when it encounters any of a list of specified characters (e.g. whitespace or punctuation), then it returns n-grams of each word: a sliding window of continuous letters, e.g. quick → [qu, ui, ic, ck]
  - Edge N-Gram Tokenizer The edge_ngram tokenizer can break up text into words when it encounters any of a list of specified characters (e.g. whitespace or punctuation), then it returns n-grams of each word which are anchored to the start of the word, e.g. quick → [q, qu, qui, quic, quick]

Radzimy sobie z żywym językiem

- Filtrowanie tokenów, przykład stemmer-a dla języka angielskiego

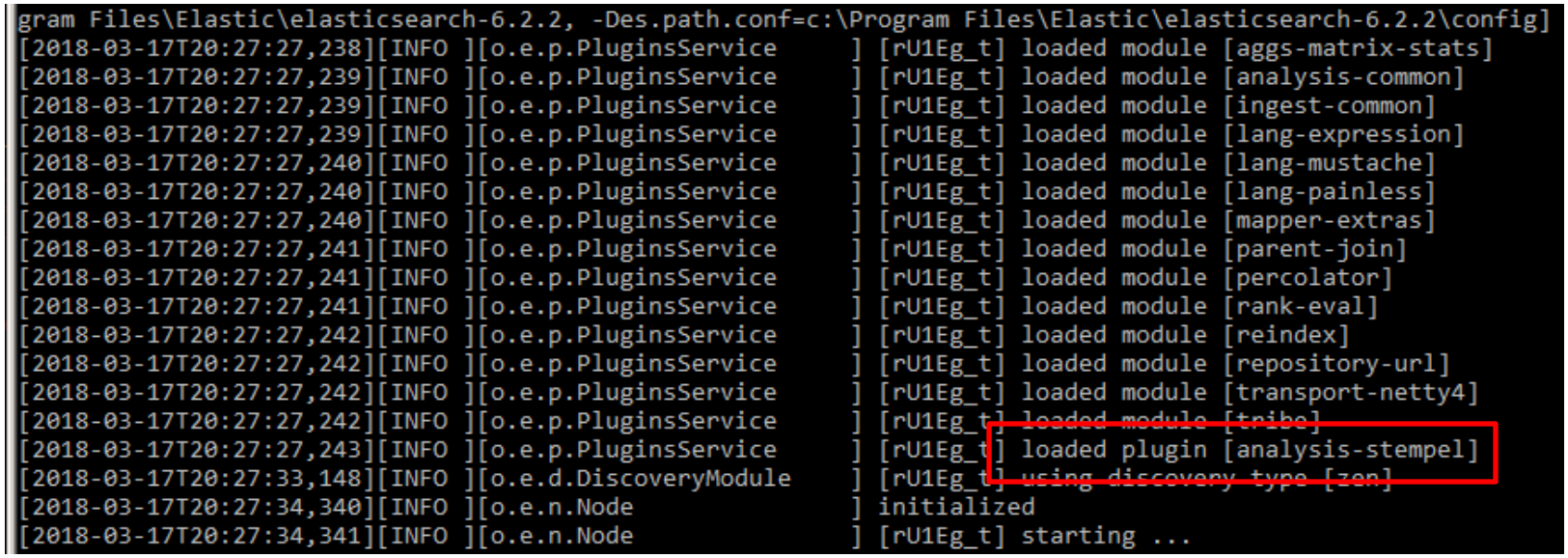
```
DELETE  /my_index

PUT /my_index
{
    "settings": {
        "analysis" : {
            "analyzer" : {
                "my_analyzer" : {
                    "tokenizer" : "standard",
                    "filter" : ["standard", "lowercase", "my_stemmer", "stop"]
                }
            },
            "filter" : {
                "my_stemmer" : {
                    "type" : "stemmer",
                    "name" : "english"
                }
            }
        }
    }
}

POST /my_index/_analyze
{
  "analyzer": "my_analyzer",
  "text": "The 2 QUICK Brown-Foxes jumped over the lazy dog's bone."
}
```

```
[2, quick, brown, fox, jump, over, lazi, dog, bone]
```

Radzimy sobie z żywym językiem

- Pakiet Stempel (Stempel Polish Analysis Plugin)
  - zawiera analizator oraz stemmer dostosowany do języka polskiego

```
bin\elasticsearch-plugin install file:///C:/path/to/plugin.zip

-> Downloading file:///t:/_zajecia/_Elasticsearch/analysis-
stempel-6.2.2.zip
[=================================================] 100%
-> Installed analysis-stempel
```



Radzimy sobie z żywym językiem

- Pakiet Stempel (Stempel Polish Analysis Plugin)

```
POST _analyze
{
  "analyzer": "polish",
  "text": "W niedzielę wielki finał turnieju Raw Air. Wszystko
wskazuje jednak na to, że w ostatnim konkursie możemy być
świadkami wielkiego chaosu. Wszystko przez zmianę regulaminu
rozgrywania konkursu. W drugiej serii zawodnicy będą skakać
zgodnie z klasyfikacją turnieju Raw Air."
}
```

```
[niedzielić, wielki, finał, turniej, raw, air, wskazuje,
ostatni, konkurs, móc, świadkami, wielki, chaos, zmiana,
regulamin, rozgrywać, konkurs, drugi, seria, zawodnik, skakać,
zgodny, klasyfikacja, turniej, raw, air,]
```

Tutoriale

- https://github.com/elastic/examples
  - Home for Elasticsearch examples available to everyone. It's a great way to get started
  - sporo przykładów dla nie najnowszych już wersji ES, jednak wydaje się, że powinny działać też w nowszych wersjach
- https://www.tutorialspoint.com/elasticsearch/index.htm