



Artur Gramacki

Uniwersytet Zielonogórski

Instytut Sterowania i Systemów Informatycznych

2019

# O systemie

- Historia

## Elasticsearch 7.4.1


October 23, 2019

Version	Original release date	Latest version	Release date	Maintenance Status <sup>[9]</sup>
0.4	2010-02-08	0.4.0	2010-02-08	No longer supported
0.5	2010-03-05 <sup>[10]</sup>	0.5.1	2010-03-09	No longer supported
0.6	2010-04-09 <sup>[11]</sup>	0.6.0	2010-04-09	No longer supported
0.7	2010-05-14 <sup>[12]</sup>	0.7.1	2010-05-17 <sup>[13]</sup>	No longer supported
0.8	2010-05-27 <sup>[14]</sup>	0.8.0	2010-05-27	No longer supported
0.9	2010-07-26 <sup>[15]</sup>	0.9.0	2010-07-26	No longer supported
0.10	2010-08-27 <sup>[16]</sup>	0.10.0	2010-08-27	No longer supported

5.4	2017-05-04 <sup>[61]</sup>	5.4.3	2017-06-27 <sup>[62]</sup>	Still supported
5.5	2017-07-06 <sup>[63]</sup>	5.5.3	2017-07-06 <sup>[64]</sup>	Still supported
5.6	2017-09-11 <sup>[65]</sup>	5.6.8	2018-02-20 <sup>[66]</sup>	Still supported
6.0	2017-11-14 <sup>[67]</sup>	6.0.1	2017-12-06 <sup>[68]</sup>	Still supported
6.1	2017-12-12 <sup>[69]</sup>	6.1.3	2018-01-16 <sup>[70]</sup>	Still supported
<b>6.2</b>	2018-02-06 <sup>[71]</sup>	6.2.2	2018-02-20 <sup>[1]</sup>	Latest

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

# 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](https://lucene.apache.org/core/8_2_0/demo/overview-summary.html#overview_description)

The screenshot shows the 'Overview' page for the Lucene 8.2.0 demo API. The page has a dark blue header with navigation links: OVERVIEW (highlighted), PACKAGE, CLASS, USE, TREE, DEPRECATED, and HELP. Below the header is a light gray bar with links: PREV, NEXT, FRAMES, NO FRAMES, and ALL CLASSES. The main content area has a white background. It starts with the title 'Lucene 8.2.0 demo API' in bold. Below the title is a paragraph: 'The demo module offers simple example code to show the features of Lucene.' followed by 'See: Description'. There is a section titled 'Packages' with a sub-header 'Packages' in an orange box. Below this is a table with two columns: 'Package' and 'Description'. The table contains two rows: one for 'org.apache.lucene.demo' with description 'Demo applications for indexing and searching.', and one for 'org.apache.lucene.demo.facet' with description 'Facets example code.'. Below the table is another paragraph: 'The demo module offers simple example code to show the features of Lucene.' followed by the title 'Apache Lucene - Building and Installing the Basic Demo'. At the bottom, there is a bulleted list of links: 'About this Document', 'About the Demo', 'Setting your CLASSPATH', 'Indexing Files', 'About the code', 'Location of the source', 'IndexFiles', and 'Searching Files'.

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](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



Products

Learn

Company

Pricing

# 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



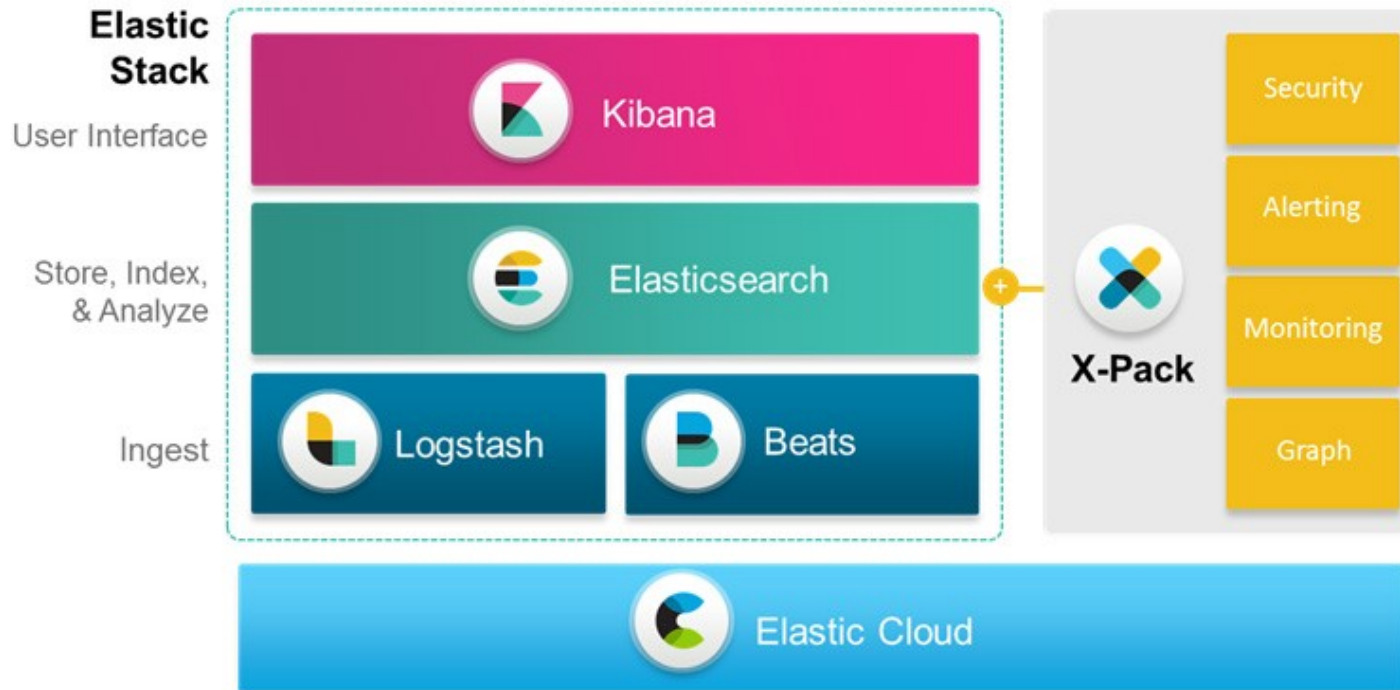


# O systemie

- 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 zaindeksowaniem dokumentu a stanem 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

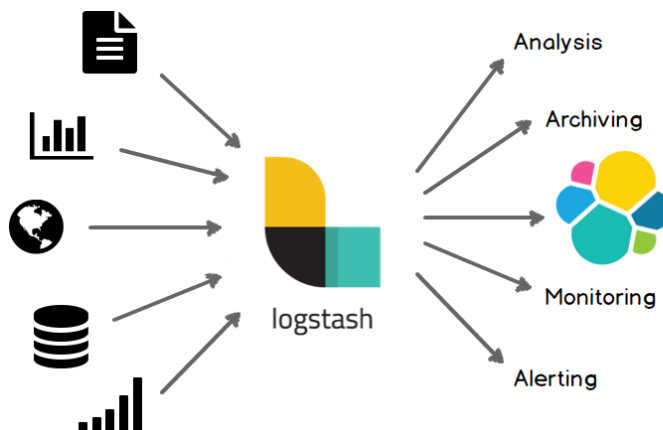


- Moduł open-source do analityki i wizualizacji danych przechowywanych w Elasticsearch
- Uruchamiany z poziomu przeglądarki (<http://localhost:5601>)
  - w zasadzie moduł bezkonfiguracyjny
- Bardzo rozbudowane możliwości graficznej prezentacji danych
  - w tym popularne obecnie tzw. dynamiczne kokpity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

# Komponenty – Logstash



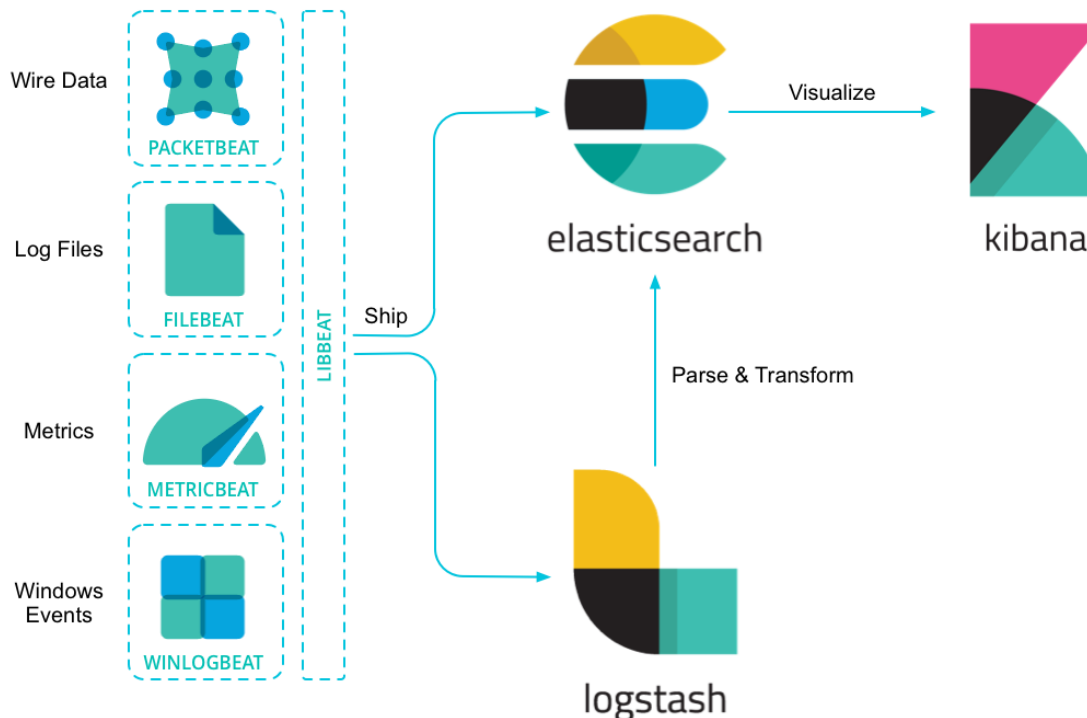
- Moduł open-source umożliwiający łączenie wielu strumieni danych wejściowych, filtrowanie każdego strumienia i przekazywania ich do Elasticsearch
  - stash – chomikować, kryjówka, ukryte zapasy
  - collection engine with real-time pipelining capabilities
- Ma możliwość unifikacji danych z różnych źródeł
- 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>

+ Elasticsearch Reference: 7.4 (current) ▾

+ Elasticsearch introduction

- **Getting started with Elasticsearch**

- Get Elasticsearch up and running
- Index some documents
- Start searching
- Analyze results with aggregations
- Where to go from here

+ Kibana Guide: 7.4 (current) ▾

[Introduction](#)

+ Set Up Kibana

- **Getting Started**

- Add sample data
- Explore Kibana using sample data
- + Build your own dashboard

+ Logstash Reference: 7.4 (current) ▾

[Logstash Introduction](#)







- **Getting Started with Logstash**

- Installing Logstash
- Stashing Your First Event
- Parsing Logs with Logstash
- Stitching Together Multiple Input and Output Plugins

# Dokumentacja, Getting Started

- Getting Started Videos

## Getting Started Videos

-  [Starting Elasticsearch](#) 
-  [Introduction to Kibana](#) 
-  [Logstash Starter Guide](#) 

# 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

# 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

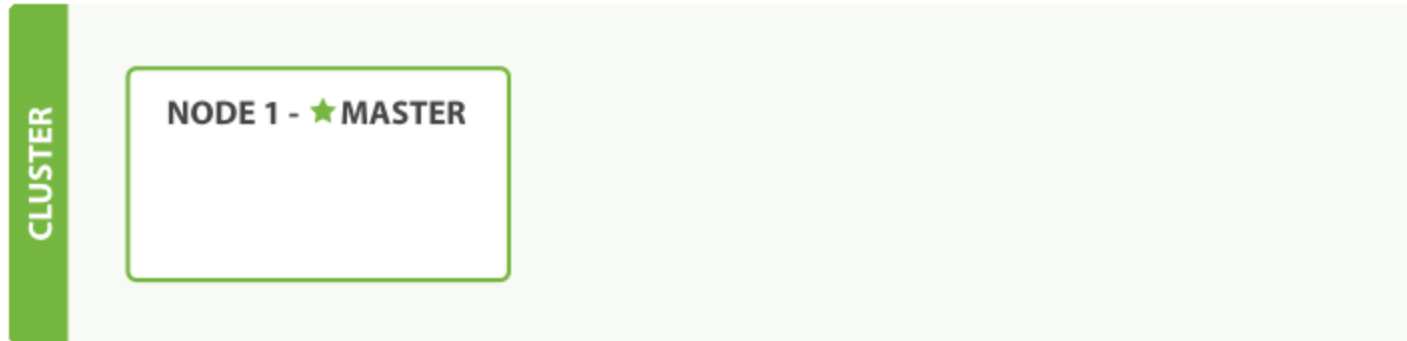
- **Shards (pol. odłamek, okruch, kawałek)**
  - mechanizm pozwalający dzielić indeks na mniejsze części zwane shard-ami
  - każdy shard stanowi samodzielny i niezależny indeks, który może być utworzony na dowolnym serwerze (node-ie)
  - poprawa wydajności, skalowanie horyzontalne, zrównoleglenie działania
  - mechanizm shard-ów jest całkowicie "przezroczysty" 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

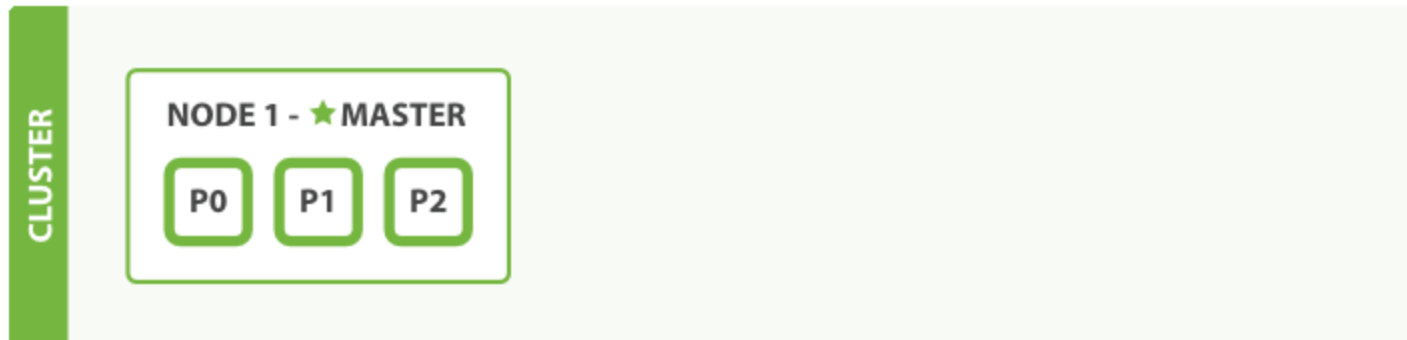
- Shards, Replicas, założenia, zasada działania
  - liczbę shard-ów oraz replik określa się w memencie definiowania indeksów
  - po zdefiniowaniu indeksu można zmieniać dynamicznie ilość replik
  - po zdefiniowaniu indeksu nie można już zmieniać ilości shard-ów
  - po zreplikowaniu, każdy indeks będzie 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 kompletna 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

- 

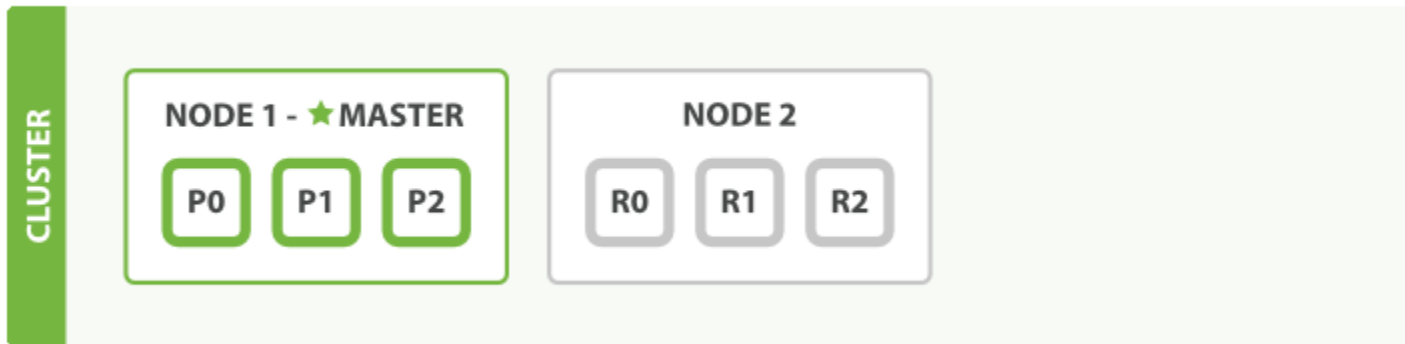


- Master tworzy 1 serwer (node), gdzie utworzono trzy podterminowe choby (primary choby)

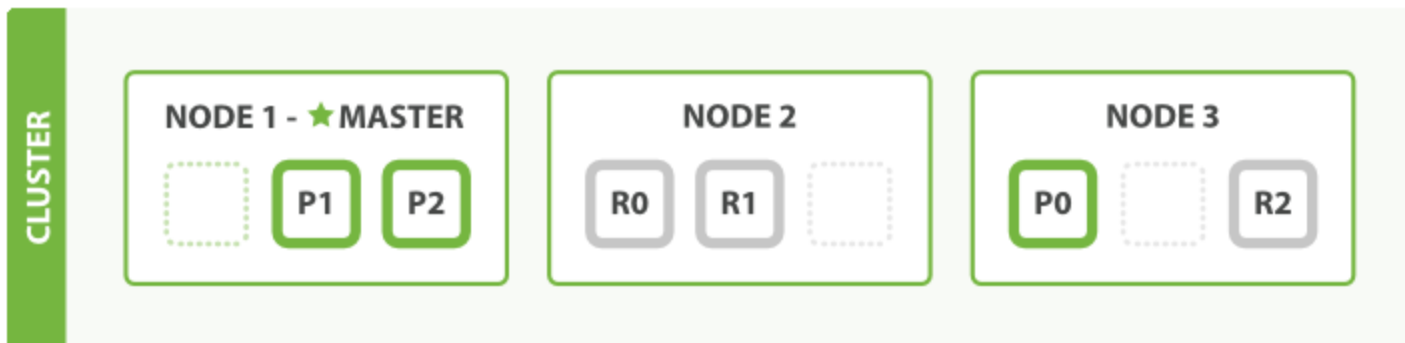


# Skalowanie horyzontalne - przykłady

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



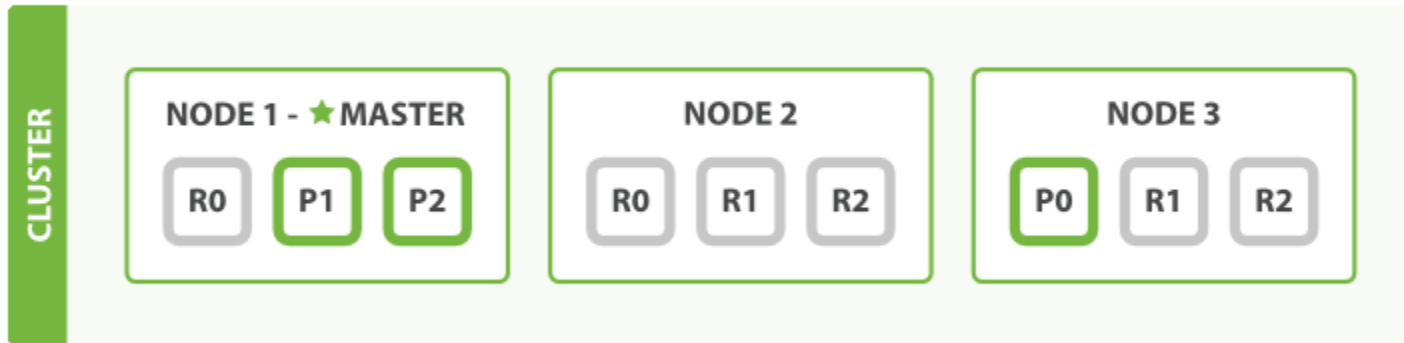
- Klaster tworzą 3 serwery (nodes). Po jakimś czasie z dwóch pierwszych serwerów zostało realokowanych do serwera 3,





# Skalowanie horyzontalne - przykłady

- Zwiększenie parametr number of replicas do 2



- Struktura klastra po awarii jednego serwera



# Skalowanie horyzontalne - oprogramowanie

- KOPF

The screenshot displays the KOPF cluster management interface. At the top, a navigation bar includes 'KOPF' and various tool icons: cluster, nodes, rest, aliases, analysis, percolator, warmers, and snapshot. The main dashboard shows cluster statistics: 2 nodes, 228,316 docs (with a +13 change), 22 shards, and 169.58MB (with a +61.42KB change). Below these are filter inputs for indices and nodes, and checkboxes for 'closed (0)' and 'special (1)' indices. A table below shows two indices: 'logstash-2015.03.23' and 'logstash-2015.03.24'. Each index has a 5-shard grid, with the 'Agamemnon' node (192.168.1.36:9300) having all shards assigned (shards 0-4 are green). A warning indicates 11 unassigned shards. The interface is dark-themed.

# 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>

The screenshot displays the Cerebro interface for an Elasticsearch cluster. At the top, navigation tabs include 'overview', 'nodes', 'rest', and 'more'. The status bar shows a refresh interval of 15 seconds and the URL 'http://localhost:9200'. The main overview section shows the following metrics:

- elasticsearch
- 1 nodes
- 3 indices
- 30 shards
- 1,008 docs
- 517.05KB

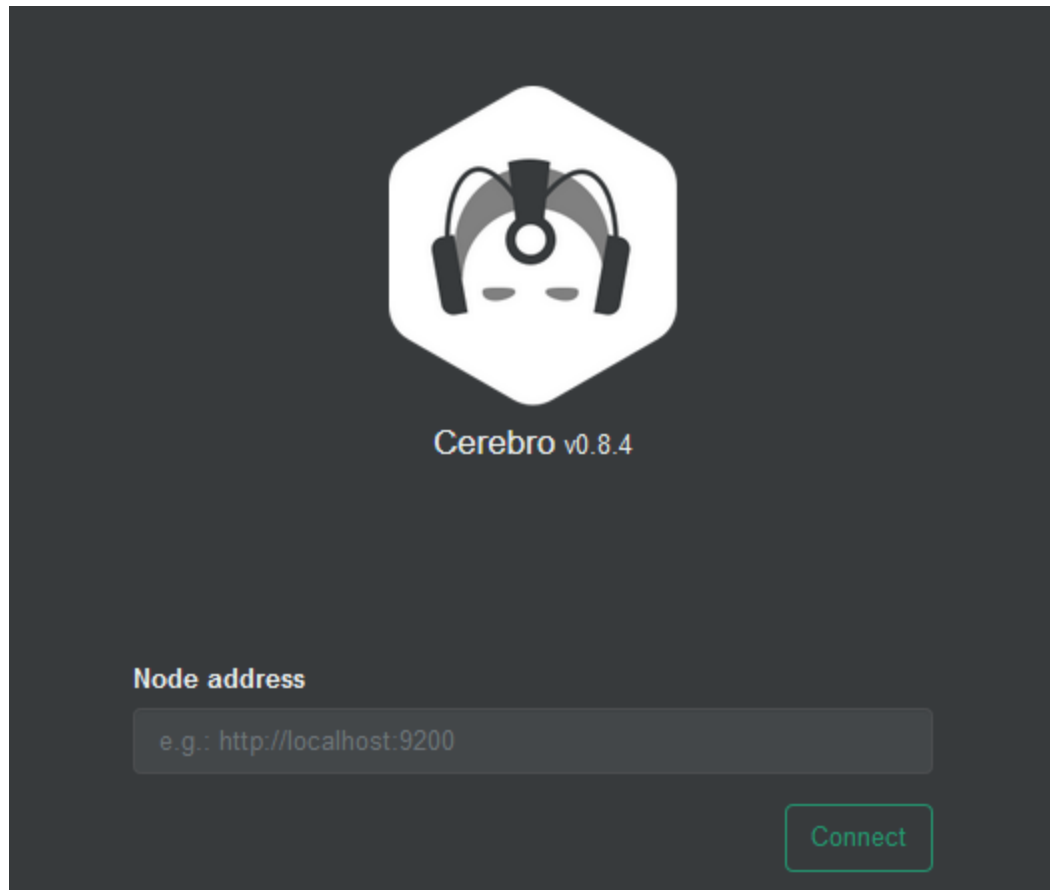
Below the overview, there are filters for indices (by name or alias) and nodes (by name). The main table lists the indices:

Index Name	Shards	Docs	Size
bank	5 * 2	1,000	474.78KB
movies	5 * 2	0	30.80KB
my_index	5 * 2	2	11.48KB

Each index entry includes a visual representation of its shards (0-4) and a node selection row. The node 'q0FBOx5' (127.0.0.1) is highlighted in green, indicating it is the selected node for the current view. The interface also shows a warning for 15 unassigned shards and various utility icons like lock, refresh, and sort.

# Cerebro

- <http://localhost:9000>



# Instalacja

- Wymaga Javy w wersji przynajmniej 8
- ZIP, TAR, DEB, RPM, MSI, Docker
  - 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
<b>home</b>	Elasticsearch home directory or <code>\$ES_HOME</code>	Directory created by unpacking the archive	
<b>bin</b>	Binary scripts including <code>elasticsearch</code> to start a node and <code>elasticsearch-plugin</code> to install plugins	<code>\$ES_HOME/bin</code>	
<b>conf</b>	Configuration files including <code>elasticsearch.yml</code>	<code>\$ES_HOME/config</code>	<code>ES_PATH_CONF</code>
<b>data</b>	The location of the data files of each index / shard allocated on the node. Can hold multiple locations.	<code>\$ES_HOME/data</code>	<code>path.data</code>
<b>logs</b>	Log files location.	<code>\$ES_HOME/logs</code>	<code>path.logs</code>
<b>plugins</b>	Plugin files location. Each plugin will be contained in a subdirectory.	<code>\$ES_HOME/plugins</code>	
<b>repo</b>	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	<code>path.repo</code>
<b>script</b>	Location of script files.	<code>\$ES_HOME/scripts</code>	<code>path.scripts</code>

# 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 kilkuliniowych 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\" } } }"
```

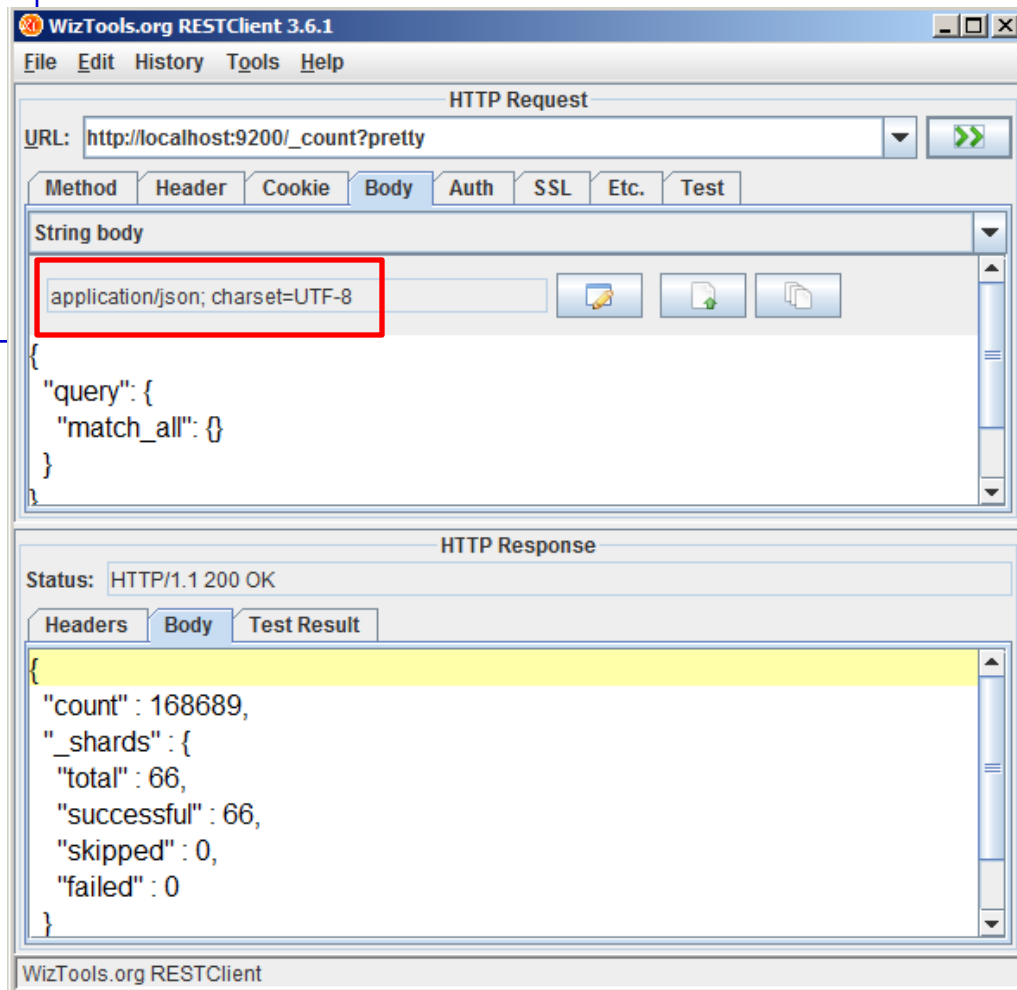
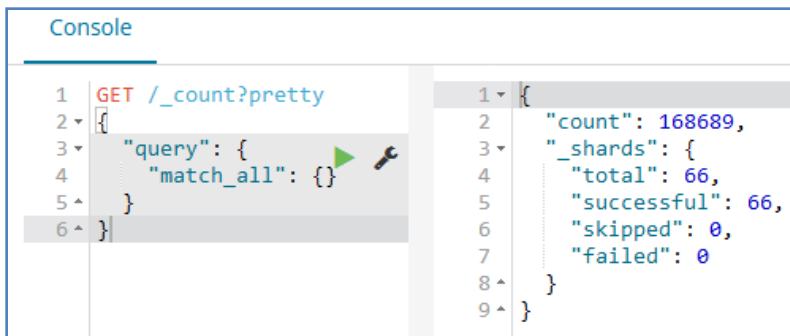
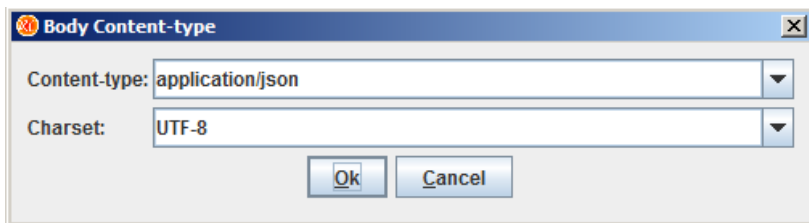
- Wniosek: cURL pod Windows trzeba zastąpić jakimś innym programem / nakładką graficzną, np.
  - <https://github.com/wiztools/rest-client>
  - Fiddler
  - Postman



# 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 Elasticsearch
  - [jvm.options](#) – konfiguracja JVM
  - [log4j2.properties](#) – konfiguracja logów

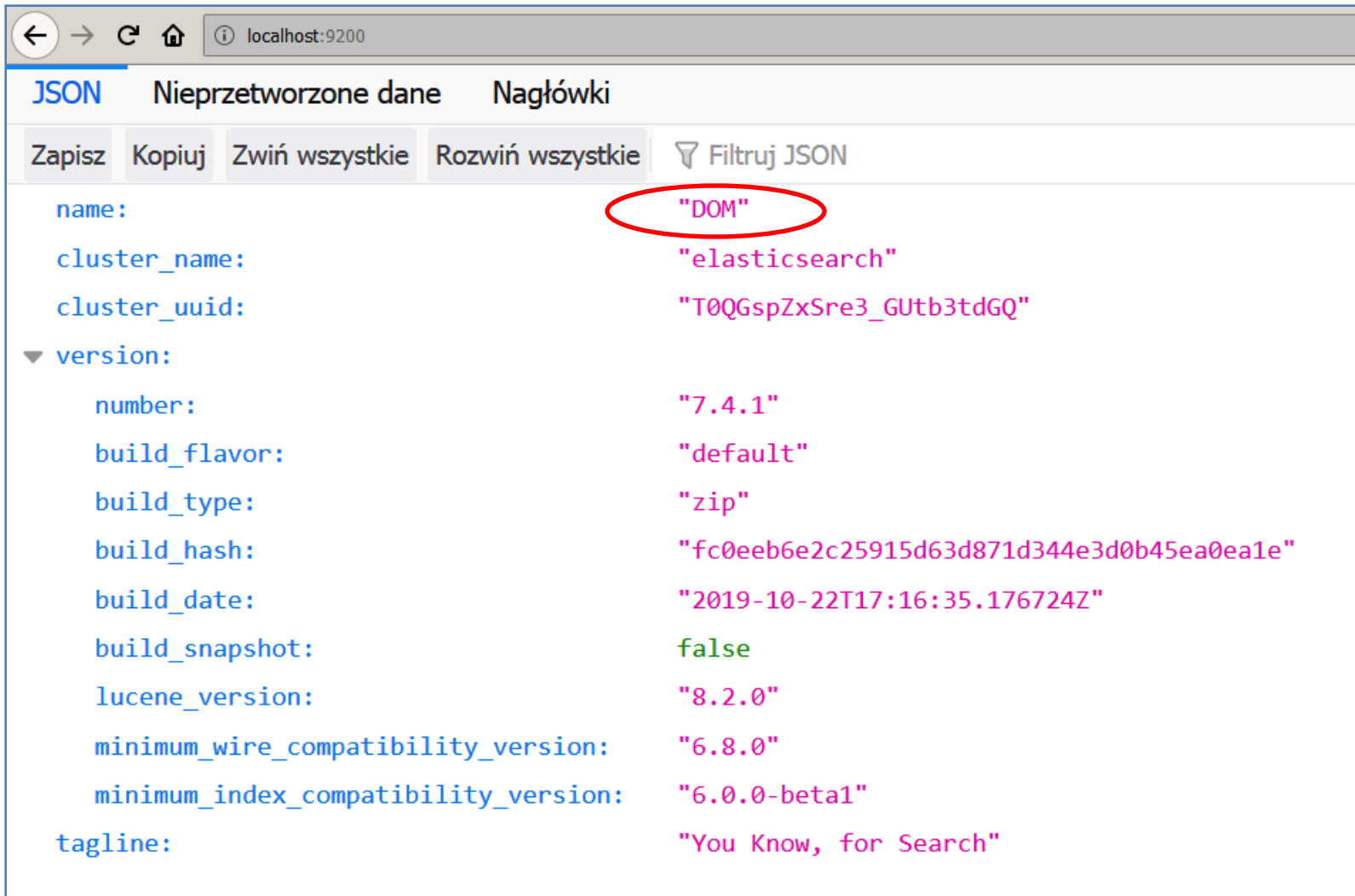
# Pierwsze uruchomienie

```
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 future release
[2019-10-31T00:39:44,064][INFO ][o.e.e.NodeEnvironment ] [DOM] using [1] data paths, mounts [[(c:)], net usable_space [93.858.9gb], types [NTFS]
[2019-10-31T00:39:44,082][INFO ][o.e.e.NodeEnvironment ] [DOM] heap size [990.7mb], compressed ordinary object pointers [true]
[2019-10-31T00:39:44,099][INFO ][o.e.n.Node ] [DOM] node name [DOM], node ID [2ukw8TNCQ9-feXZhw5KtxQ], cluster name [elasticsearch]
[2019-10-31T00:39:44,103][INFO ][o.e.n.Node ] [DOM] version[7.4.1], pid[4944], build[default/zip/fc0eeb6e2c25915de/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:CMSInitiatingOccupancyOnly=true, -XX:+UseCMSInitiatingOccupancyOnly, -Des.networkaddress.cache.ttl=60, -Des.networkaddress.cache.negative.ttl=10, -XX:CMSInitiatingOccupancyOnly=true, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dio.netty allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.io.tmpdir=C:\Users\ADMINI~1\AppData\Local\Temp\elasticsearch, -XX:+HeapDumpOnOutOfMemoryError, -XX:HeapDumpPath=C:\Program Files\Elastic\elasticsearch-7.4.1\logs/hs_err_pid%p.log, -Xlog:gc*,gc+age=trace,safepoint:file=logs/gc.log:utctime,pid,tags:filecount=32,filesize=64m, -Djava.locale.providers=COMPAT, -Dio.netty.allocator.type=unpooled, -XX:MaxDirectMemorySize=536870912, -Delasticsearch, -Des.path.home=c:\Program Files\Elastic\elasticsearch-7.4.1, -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\Elasticsearch-7.
s.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.
80bdacc5e8) 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.Security
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
scovery.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
t2sw}{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
/SSidgiCyMdT2sw}{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



The screenshot shows a web browser window at localhost:9200 displaying JSON data. The data is organized into a table with the following fields and values:

Field	Value
name:	"DOM"
cluster_name:	"elasticsearch"
cluster_uuid:	"T0QGspZxSre3_GUtb3tdGQ"
version:	
number:	"7.4.1"
build_flavor:	"default"
build_type:	"zip"
build_hash:	"fc0eeb6e2c25915d63d871d344e3d0b45ea0ea1e"
build_date:	"2019-10-22T17:16:35.176724Z"
build_snapshot:	false
lucene_version:	"8.2.0"
minimum_wire_compatibility_version:	"6.8.0"
minimum_index_compatibility_version:	"6.0.0-beta1"
tagline:	"You Know, for Search"

The value "DOM" for the 'name' field is circled in red in the original image.

# 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
```

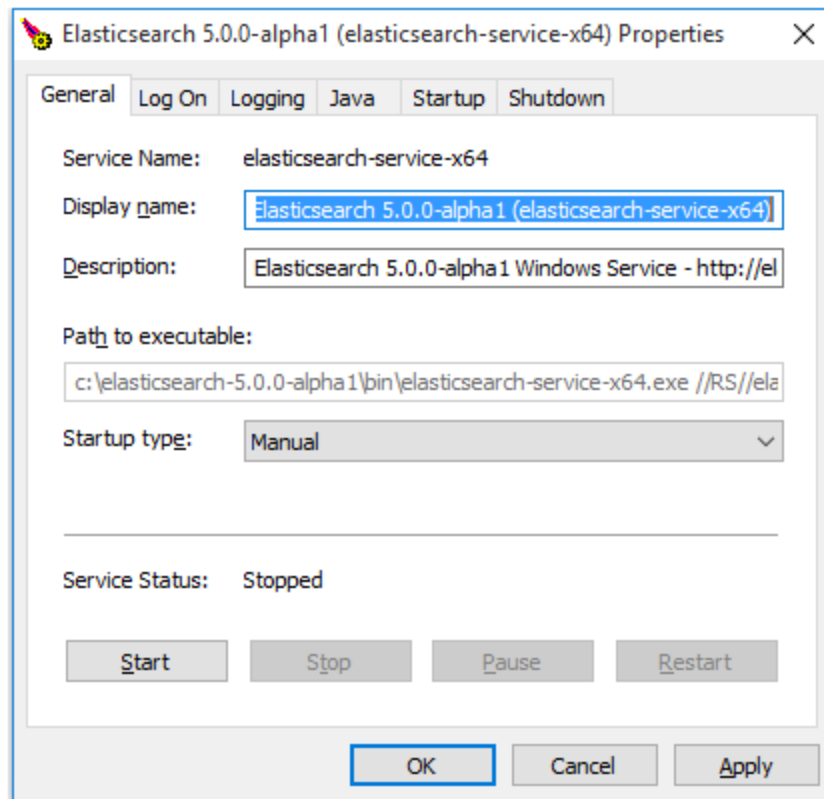
```
Installing service      : "elasticsearch-service-x64"
```

```
Using JAVA_HOME (64-bit): "c:\jvm\jdk1.8"
```

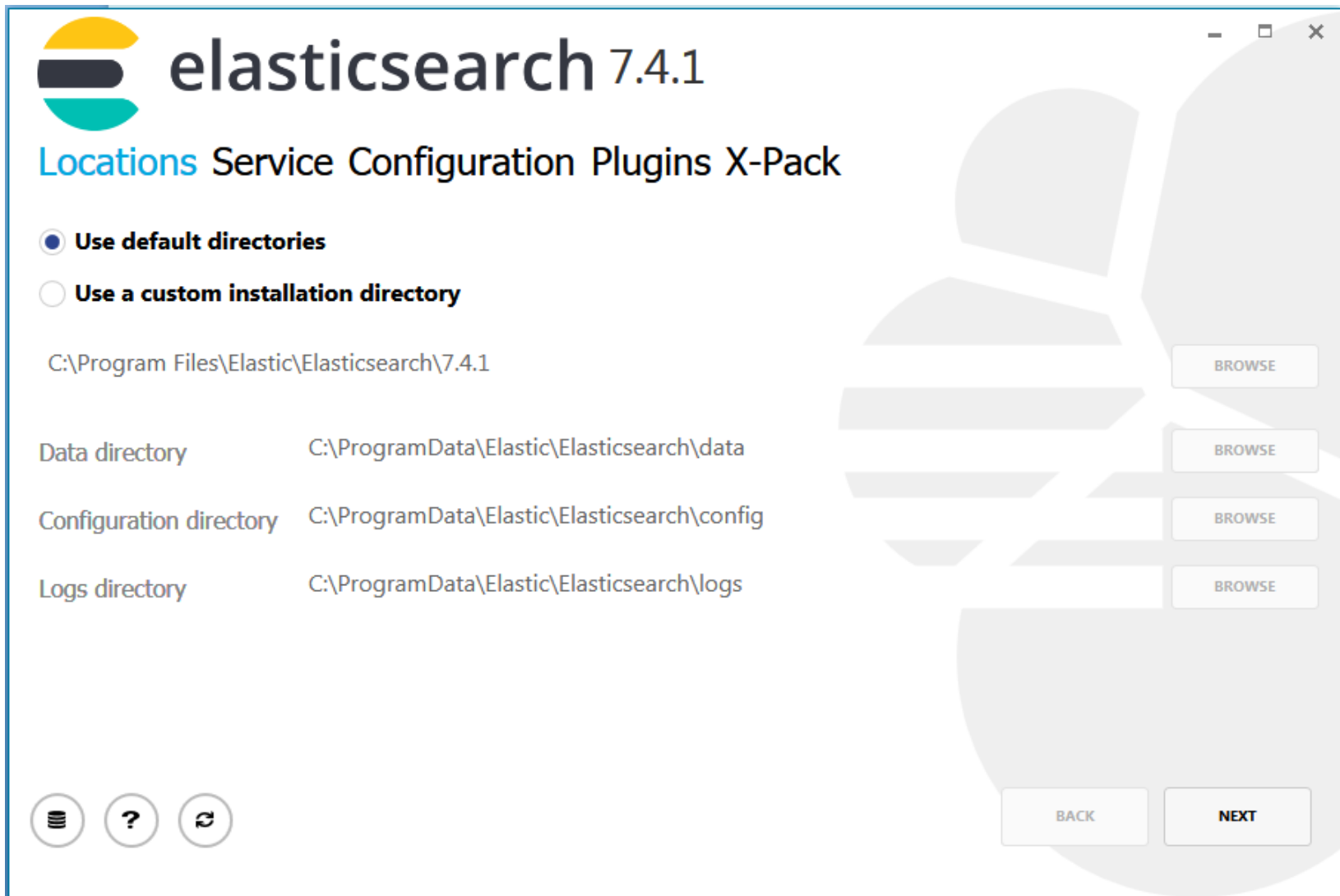
```
The service 'elasticsearch-service-x64' has been installed.
```

# Konfiguracja usługi w Windows

- elasticsearch-service-mgr.exe



# Instalacja MSI





# Instalacja MSI

**elasticsearch 7.4.1**

Locations **Service** Configuration Plugins X-Pack

Do not install as a service (start manually when needed)

**Install as a service**

**Account information**

Use Local System account

Use Network Service account

Existing user

**General properties**

Start the service after this installation is complete

Start the service when Windows starts (Automatic)

**BACK** **NEXT**

# Instalacja MSI

**elasticsearch 7.4.1**

Locations Service **Configuration** Plugins X-Pack

**Identifiers**

Cluster name:

Node name:

**Roles**

Data  Master  Ingest

**Memory** 2 GB/6 GB

Lock JVM memory

**Network** (Optional)

Network host:

HTTP port:  Transport port:

**Discovery** (Optional)

This is the first master in a new cluster

**Seed Hosts**

# Instalacja MSI


**elasticsearch 7.4.1**

Locations Service Configuration **Plugins X-Pack**

- Ingest Attachment Processor**  
The ingest attachment plugin lets Elasticsearch extract file attachments in common formats (such as PPT, XLS, and PDF) by using the Apache text and metadata extraction library Tika. You can use the ingest attachment plugin as a replacement for the mapper attachment plugin.
- ICU Analysis**  
The ICU Analysis plugin integrates the Lucene ICU module into Elasticsearch, adding extended Unicode support using the ICU libraries, including better analysis of Asian languages, Unicode normalization, Unicode-aware case folding, collation support, and transliteration.
- Japanese (kuromoji) Analysis**  
The Japanese (kuromoji) Analysis plugin integrates the Lucene kuromoji analysis module into Elasticsearch.
- Phonetic Analysis**  
The Phonetic Analysis plugin provides token filters which convert tokens to their phonetic representation using Soundex, Metaphone, and a variety of other algorithms.
- Smart Chinese Analysis**  
The Smart Chinese Analysis plugin integrates the Lucene Smart Chinese analysis module into Elasticsearch.
- Stempel Polish Analysis**  
The Stempel (Polish) Analysis plugin integrates the Lucene Stempel (Polish) analysis module into Elasticsearch.
- EC2 Discovery**  
The EC2 discovery plugin uses the AWS API for unicast discovery.
- Azure Discovery (Classic)**

Navigation buttons: **BACK**, **NEXT**, **PROXY**

# Instalacja MSI

 **elasticsearch 7.4.1**

Locations Service Configuration Plugins **X-Pack**

License

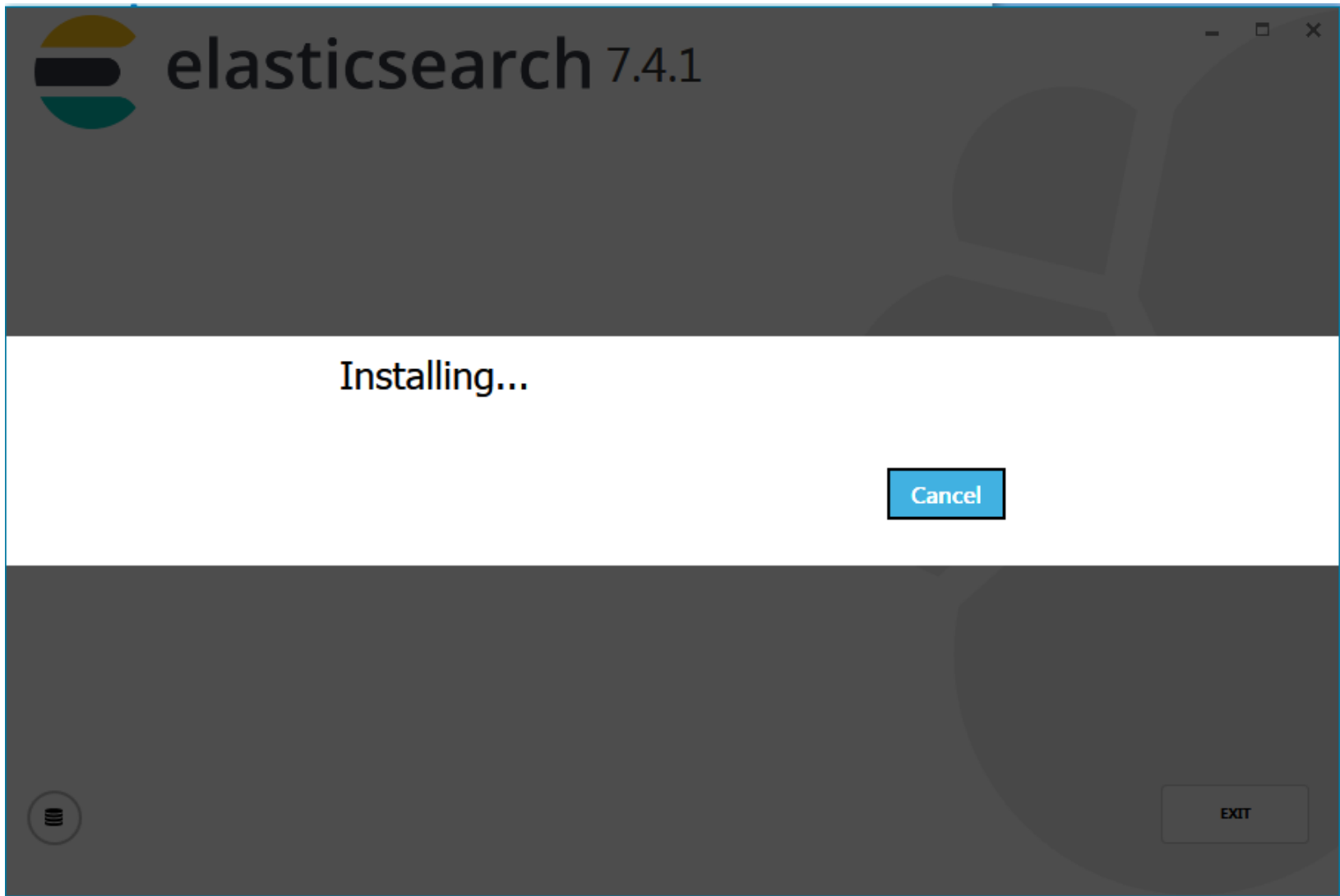
**Which license is for me?**

**Basic License**  
Access to all free X-Pack Basic features without an expiry date on the license.

**Trial License**  
Access to all X-Pack features for 30 days, including Machine Learning, Graph, Alerting, Security, and others.

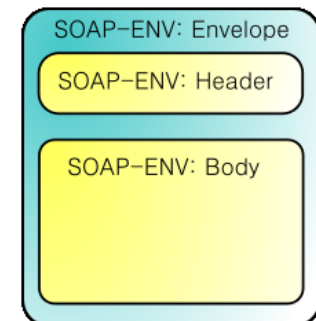
[Full overview of licences and subscriptions](#)

# 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 przesyłania, 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

`http://www.jakis-serwer.pl:8080/katalog1/katalog2/plik?parametr1=wartosc1&parametr2=wartosc2#fragment_dokumentu`

The diagram illustrates the structure of the URI `http://www.jakis-serwer.pl:8080/katalog1/katalog2/plik?parametr1=wartosc1&parametr2=wartosc2#fragment_dokumentu`. Brackets and vertical lines connect each part of the URI to its corresponding label below:

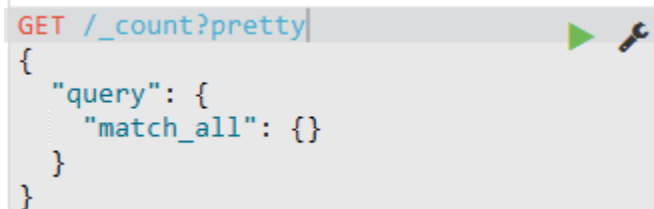
- `http://` is labeled as **schemat (protokół)**.
- `www.jakis-serwer.pl` is labeled as **host (nazwa serwera)**.
- `:8080` is labeled as **port**.
- `/katalog1/katalog2/plik` is labeled as **ścieżka do pliku**.
- `?parametr1=wartosc1&parametr2=wartosc2` is labeled as **zapytanie**.
- `#fragment_dokumentu` is labeled as **fragment**.

- 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": {}
  }
}'
```

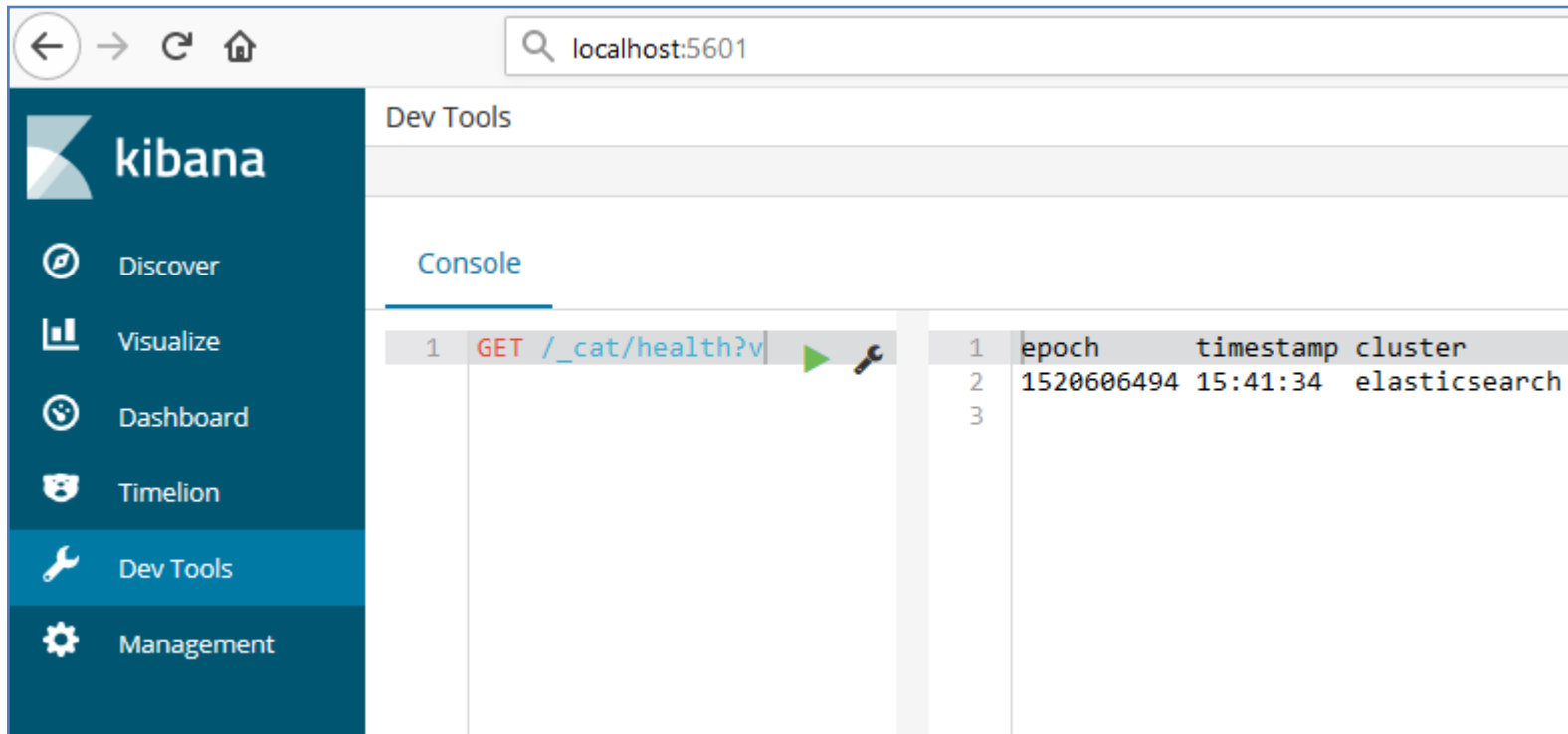


A screenshot of a REST client interface. The top bar shows the request method and path: `GET /_count?pretty`. Below the bar, the JSON response is displayed in a light gray background with syntax highlighting. The response is a JSON object with a `query` property containing a `match_all` object.

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

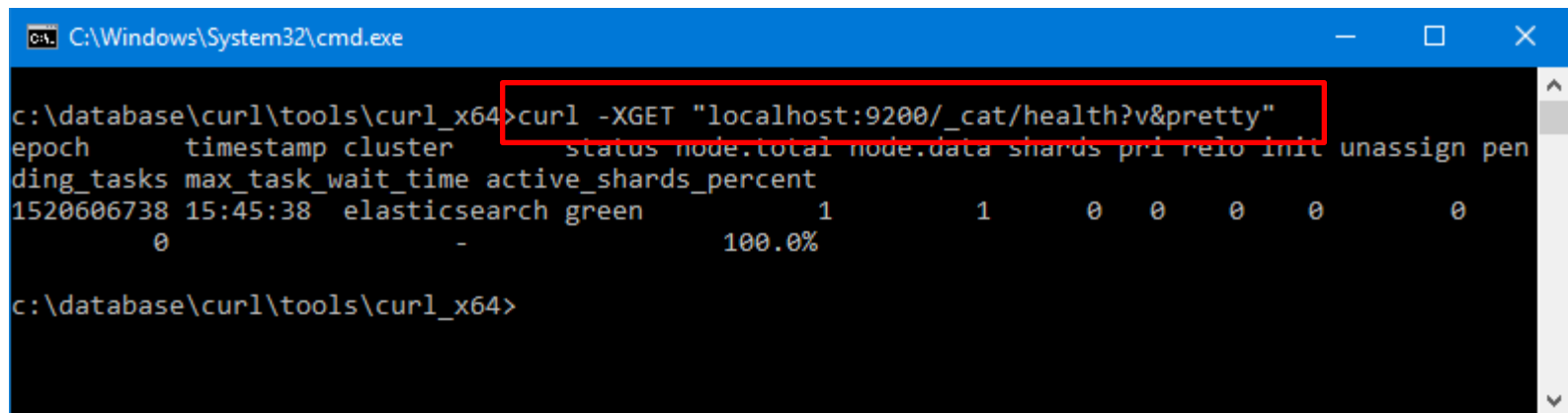


# Konsola w Kibana vs cURL



The screenshot shows the Kibana Dev Tools interface. The left sidebar contains navigation options: Discover, Visualize, Dashboard, Timelion, Dev Tools (selected), and Management. The main area is titled 'Dev Tools' and 'Console'. A request is shown: `1 GET /_cat/health?v`. The response is a table with the following data:

1	epoch	timestamp	cluster
2	1520606494	15:41:34	elasticsearch
3			



The screenshot shows a Windows command prompt window with the following command and output:

```
c:\database\curl\tools\curl_x64>curl -XGET "localhost:9200/_cat/health?v&pretty"
```

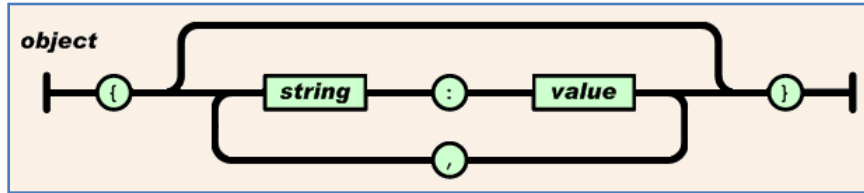
```
epoch      timestamp cluster      status node.total node.data shards pri relo init unassign pending_tasks max_task_wait_time active_shards_percent
1520606738 15:45:38  elasticsearch green          1          1      0  0    0  0          0
0          -          100.0%
```

# 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

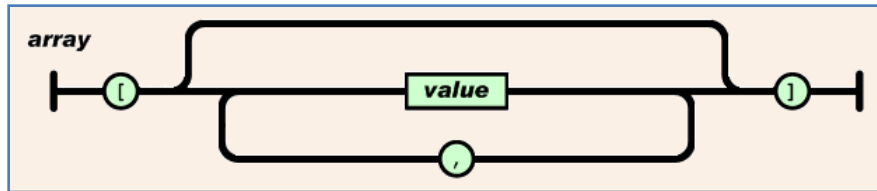
# JSON

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



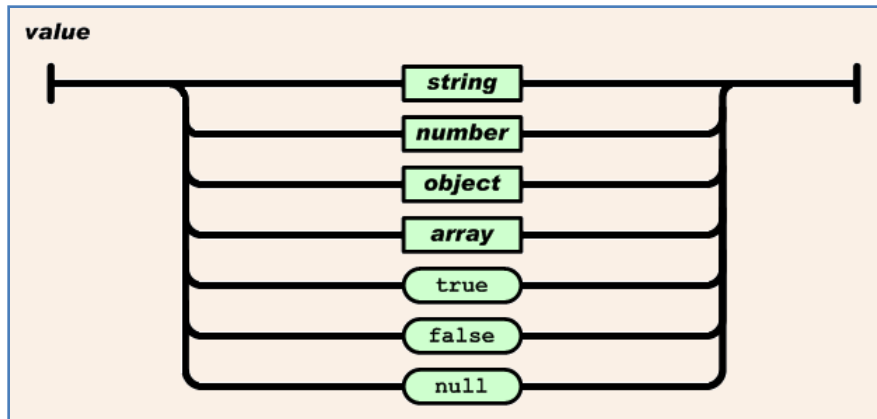
# JSON

- **Tabela** jest uporządkowanym zbiorem wartości



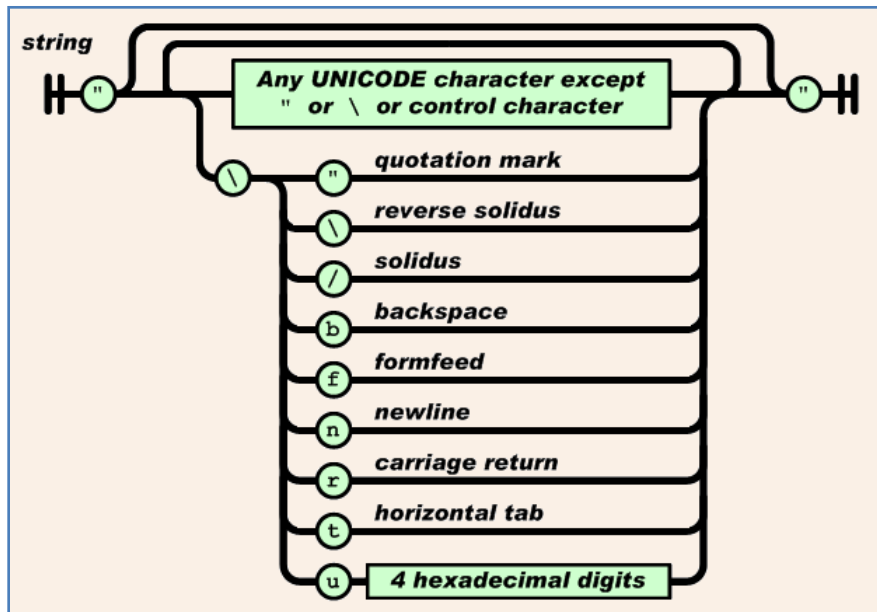
# JSON

- **Wartość** to łańcuch znakowy



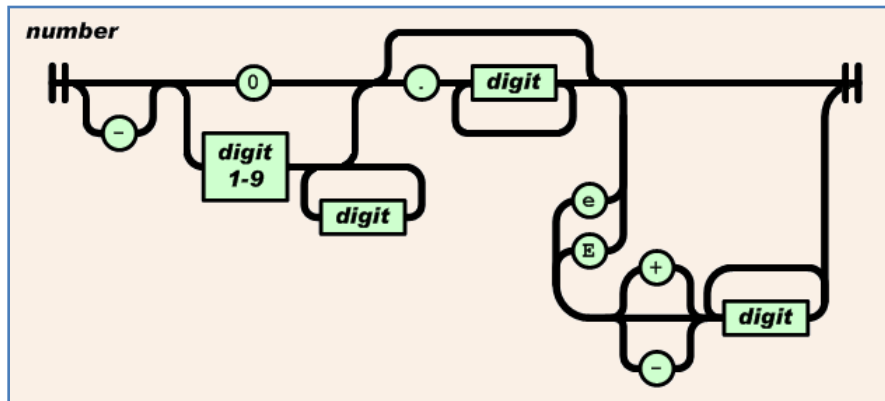
# JSON

- **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



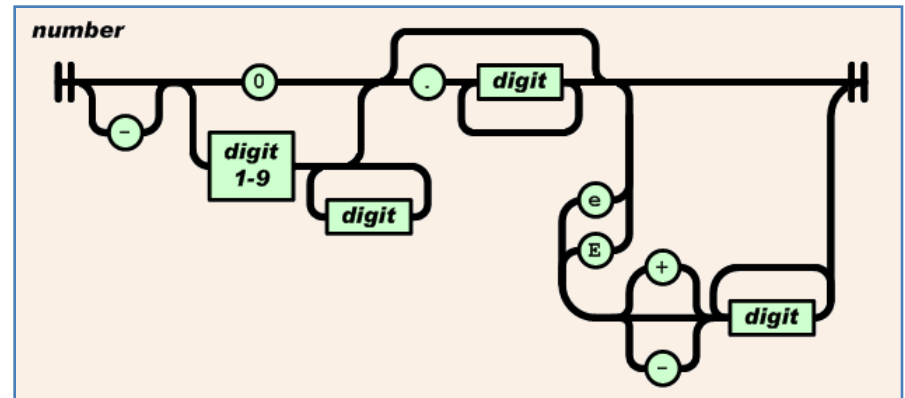
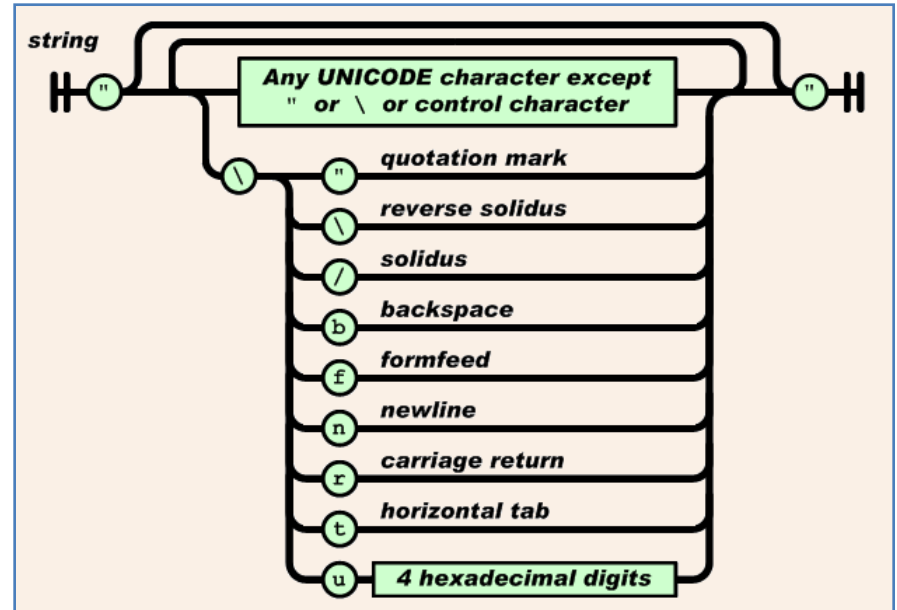
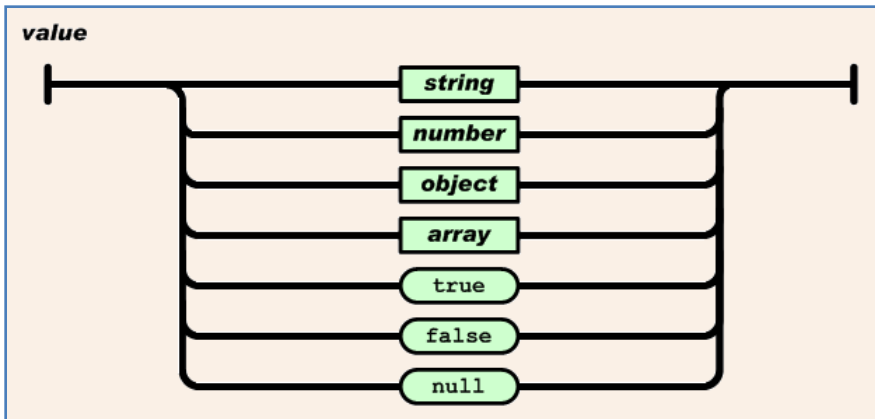
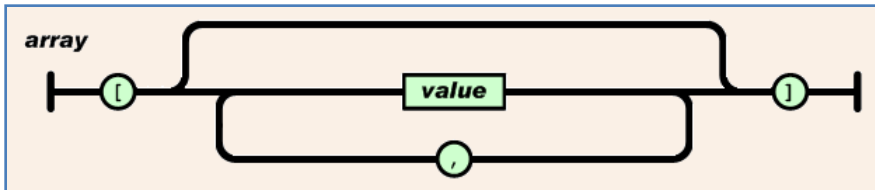
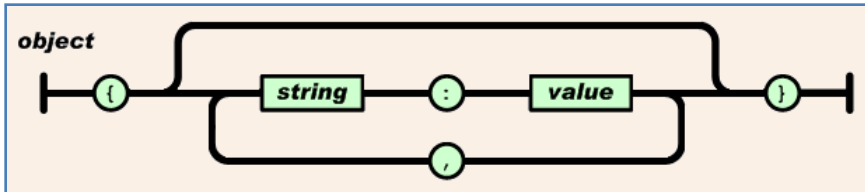
# JSON

- **Obiekt** jest nieuporządkowanym zbiorem par nazwa/wartość.
- **Tabela** jest uporządkowanym zbiorem wartości
- **Wartość** to łańcuch znakowy
- **łańcuch 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

- Łżejszy, 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

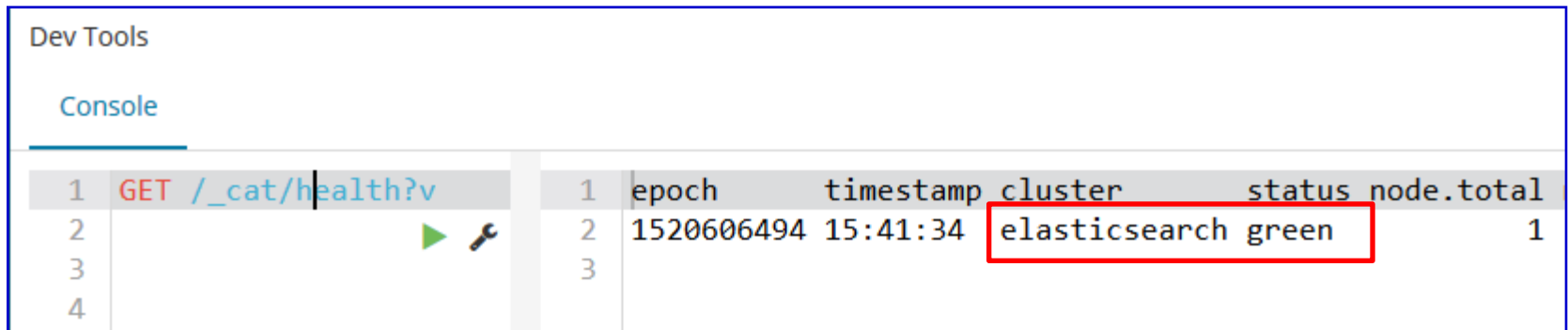
```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName>Jones</lastName>
  </employee>
</employees>
```

# 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

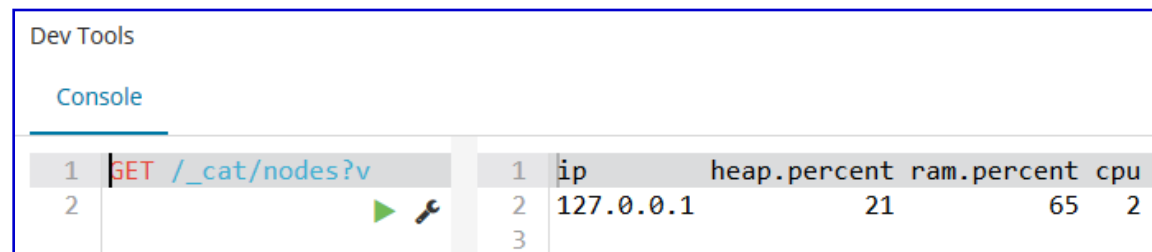
- „Kolorowe” stany klastra
  - Green - everything is good (cluster is fully functional)
  - Yellow - all data is available but some replicas are not yet allocated (cluster is fully functional)
  - Red - some data is not available for whatever reason (cluster is partially functional)



```
1 GET /_cat/health?v
2
3
4
```

epoch	timestamp	cluster	status	node.total
1520606494	15:41:34	elasticsearch	green	1

- Lista węzłów

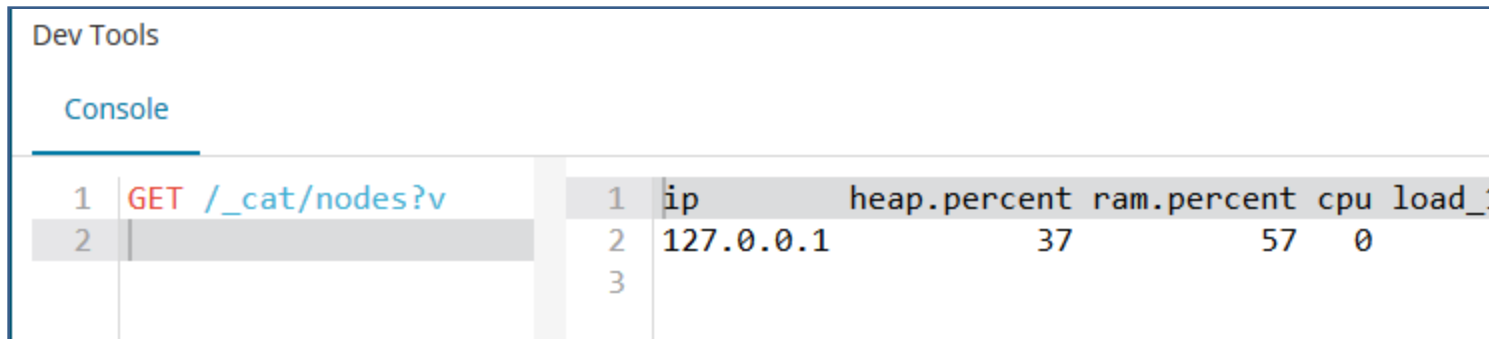


```
1 GET /_cat/nodes?v
2
3
```

ip	heap.percent	ram.percent	cpu
127.0.0.1	21	65	2

# Eksplorowanie klastra

- Lista węzłów



The screenshot shows the DevTools Console with a REST client request and its response. The request is a GET to `/_cat/nodes?v`. The response is a JSON array of node information.

1	GET	/_cat/nodes?v	1	ip	heap.percent	ram.percent	cpu	load_1
2			2	127.0.0.1	37	57	0	
			3					

# 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](#)

```
Lister - [t:\zajecia\Elasticsearch\datasets\shakespeare_6.0.json]
Plik Edytuj Opcje Kodowanie Pomoc 100 %
{"index":{"_index":"shakespeare","id":"111374"}}
{"type":"line","line_id":"111375","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.163","speaker":"LEONTES","text_entry":"0, peace, Pat"}
{"index":{"_index":"shakespeare","id":"111375"}}
{"type":"line","line_id":"111376","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.164","speaker":"LEONTES","text_entry":"Thou shouldst"}
{"index":{"_index":"shakespeare","id":"111376"}}
{"type":"line","line_id":"111377","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.165","speaker":"LEONTES","text_entry":"As I by thine"}
{"index":{"_index":"shakespeare","id":"111377"}}
{"type":"line","line_id":"111378","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.166","speaker":"LEONTES","text_entry":"And made betw"}
{"index":{"_index":"shakespeare","id":"111378"}}
{"type":"line","line_id":"111379","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.167","speaker":"LEONTES","text_entry":"But how, is t"}
{"index":{"_index":"shakespeare","id":"111379"}}
{"type":"line","line_id":"111380","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.168","speaker":"LEONTES","text_entry":"As I thought."}
{"index":{"_index":"shakespeare","id":"111380"}}
{"type":"line","line_id":"111381","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.169","speaker":"LEONTES","text_entry":"A prayer upor"}
{"index":{"_index":"shakespeare","id":"111381"}}
{"type":"line","line_id":"111382","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.170","speaker":"LEONTES","text_entry":"For him, I pa"}
{"index":{"_index":"shakespeare","id":"111382"}}
{"type":"line","line_id":"111383","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.171","speaker":"LEONTES","text_entry":"An honourable"}
{"index":{"_index":"shakespeare","id":"111383"}}
{"type":"line","line_id":"111384","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.172","speaker":"LEONTES","text_entry":"And take her"}
{"index":{"_index":"shakespeare","id":"111384"}}
{"type":"line","line_id":"111385","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.173","speaker":"LEONTES","text_entry":"Is richly not"}
{"index":{"_index":"shakespeare","id":"111385"}}
{"type":"line","line_id":"111386","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.174","speaker":"LEONTES","text_entry":"By us, a pair"}
{"index":{"_index":"shakespeare","id":"111386"}}
{"type":"line","line_id":"111387","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.175","speaker":"LEONTES","text_entry":"What! look up"}
{"index":{"_index":"shakespeare","id":"111387"}}
{"type":"line","line_id":"111388","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.176","speaker":"LEONTES","text_entry":"That eer I pu"}
{"index":{"_index":"shakespeare","id":"111388"}}
{"type":"line","line_id":"111389","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.177","speaker":"LEONTES","text_entry":"My ill suspic"}
{"index":{"_index":"shakespeare","id":"111389"}}
{"type":"line","line_id":"111390","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.178","speaker":"LEONTES","text_entry":"And son unto"}
{"index":{"_index":"shakespeare","id":"111390"}}
{"type":"line","line_id":"111391","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.179","speaker":"LEONTES","text_entry":"Is troth-pli"}
{"index":{"_index":"shakespeare","id":"111391"}}
{"type":"line","line_id":"111392","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.180","speaker":"LEONTES","text_entry":"Lead us from"}
{"index":{"_index":"shakespeare","id":"111392"}}
{"type":"line","line_id":"111393","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.181","speaker":"LEONTES","text_entry":"Each one dema"}
{"index":{"_index":"shakespeare","id":"111393"}}
{"type":"line","line_id":"111394","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.182","speaker":"LEONTES","text_entry":"Performd in t"}
{"index":{"_index":"shakespeare","id":"111394"}}
{"type":"line","line_id":"111395","play_name":"A Winters Tale","speech_number":38,"line_number":"5.3.183","speaker":"LEONTES","text_entry":"We were disse"}
{"index":{"_index":"shakespeare","id":"111395"}}
{"type":"line","line_id":"111396","play_name":"A Winters Tale","speech_number":38,"line_number":"","speaker":"LEONTES","text_entry":"Exeunt"}
```



# Przykładowe dane

- Podstawy, plik [shakespeare.json](#)
  - utworzenie mapowania (ang. mappings) poszczególnych pól
  - 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				
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.json

```
{ "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": -86.05928083 } }, "src": "PH", "dest": "MM", "srcdest": "PH:MM", "@tags": [ "success", "info" ], "utc_time": "2015-05-18T09:03:25.877Z", "referrer": "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.theacademvoofperformingartsandscience.org", "request": "/canhaz/gemini-7.gif", "url": "https://motion-media.theacademvoofperformingartsandscience.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 wwwoooooo\", \"xss\": \"<script>console.log(\\\"xss\\\")</script>\", \"headings\": [ \"<h3>f-i-j-n-l-qg</h5>\", \"http://facebook.com/success/loedewijk-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\", \"qg:type\": \"article\", \"qg:title\": \"CBS Crew Rat Fink\", \"qg: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 ...\", \"qg: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 Mauex\", \"qg:image\": \"http://IMAGES1.laweekly.com/imagex/cbs-crew-rat-fink/u/original/2430299/img_2049.jpg\", \"qg:image:height\": \"360\", \"qg:image:width\": \"480\", \"qg: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/imagex/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\", \"qg:type\": \"article\", \"qg:title\": \"Push and Retna in Koreatown\", \"qg: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 ...\", \"qg: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\", \"qg:image\": \"http://IMAGES1.laweekly.com/imagex/push-and-retna-in-koreatown/u/original/2430376/img_3671.jpg\", \"qg:image:height\": \"360\", \"qg:image:width\": \"480\", \"qg: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/imagex/push-and-retna-in-koreatown/u/original/2430376/img_3671.jpg\", \"twitter:site\": \"@laweekly\" }, { \"url\": \"http://www.laweekly.com/news/asylum-xueta-pdb-on-santa-monica-2368012\", \"qg:type\": \"article\", \"qg:title\": \"Asylum, Xueta, PDB on Santa Monica\", \"qg: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...\", \"qg:url\": \"http://www.laweekly.com/news/asylum-xueta-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\", \"qg:image\": \"http://images1.laweekly.com/imagex/asylum-xueta-pdb-on-santa-monica/u/original/2430137/img_5027.jpg\", \"qg:image:height\": \"360\", \"qg:image:width\": \"480\", \"qg:site_name\": \"LA Weekly\", \"twitter:title\": \"Asylum, Xueta, 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/imagex/asylum-xueta-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\", \"qg:type\": \"article\", \"qg:title\": \"Laurence Tribe Tangles with CBS and L.A. City Hall\", \"qg:description\": \"The United States Court of Appeals for the Ninth Circuit&#039;s Courtroom 3 - a miniature auditorium with comfortable, smoked salmon-colored seats - wa...\", \"qg: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\", \"qg: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&#039;s Courtroom 3 - a miniature auditorium with comfortable, smoked salmon-colored seats - wa...\", \"twitter:card\": \"summary\", \"twitter:site\": \"@laweekly\" } ], \"machine\": { \"qg\": \"win xp\", \"ram\": 3221225472 }, \"@version\": \"1\" }
```

# Przykładowe dane

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

```
Formatted JSON Data
{
  "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"
  ]
}
```

```
1 {
2   "index": {
3     "_index": "logstash-2015.05.18",
4     "_type": "log"
5   }
6 } {
7   "@timestamp": "2015-05-18T09:03:25.877Z",
8   "ip": "185.124.182.126",
9   "extension": "gif",
10  "response": "404",
11  "geo": {
12    "coordinates": {
13      "lat": 36.518375,
14      "lon": -86.05828083
15    },
16    "src": "PH",
17    "dest": "MM",
18    "srcdest": "PH:MM"
19  },
20  "@tags": [
21    "success",
22    "info"
23  ],
24  "utc_time": "2015-05-18T09:03:25.877Z",
25  "referer": "http://twitter.com/error/william-shepherd",
26  "agent": "Mozilla/5.0 (X11; Linux x86_64; rv:6.0a1) Gecko/2011042",
27  "clientip": "185.124.182.126",
28  "bytes": 804,
29  "host": "motion-media.theacademvofperformingartsandscience.org",
30  "request": "/canhaz/gemini-7.gif",
31  "url": "https://motion-media.theacademvofperformingartsandscience",
32  "@message": "185.124.182.126 - - [2015-05-18T09:03:25.877Z] \"GET",
33  "spaces": "this is a thing with lots of spaces",
34  "xss": "<script>console.log(\"xss\")</script>",
35  "headings": [
36    "<h3>f-i-j-nl-ng</h5>",
37    "http://facebook.com/success/lodewijk-van-den-berg"
38  ],
39 }
```

# 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","referrer":"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; 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; 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/asylum-ruets-pdb-on-santa-monica-
2368012","og:type":"article","og:title":"Asylum, 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/asylum-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/asylum-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":"Asylum, 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/asylum-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","cpu":"3221225472"},"@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	Yz-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	DN95cZkTTPafzMQQ9cqE6A	5	1	18524	0	90mb	90mb

# Przykładowe dane

- Podstawy, plik [accounts.zip](#)



The screenshot shows a Notepad window titled "Lister - [t\zajecia\Elasticsearch\datasets\accounts.json]". The window contains a list of JSON objects representing account records. Each record includes fields such as "index", "account\_number", "balance", "firstname", "lastname", "age", "gender", and "address". The records are sorted by account number from 1 to 107. The window also shows a menu bar with "Plik", "Edytuj", "Opcje", "Kodowanie", and "Pomoc", and a status bar at the bottom right indicating "2 %".

```

{
  "index": { "_id": "1" },
  "account_number": 1, "balance": 39225, "firstname": "Amber", "lastname": "Duke", "age": 32, "gender": "M", "address": "880 Holmes Lane", "employ
  "index": { "_id": "6" },
  "account_number": 6, "balance": 5686, "firstname": "Hattie", "lastname": "Bond", "age": 36, "gender": "M", "address": "671 Bristol Street", "emp
  "index": { "_id": "13" },
  "account_number": 13, "balance": 32838, "firstname": "Nanette", "lastname": "Bates", "age": 28, "gender": "F", "address": "789 Madison Street",
  "index": { "_id": "18" },
  "account_number": 18, "balance": 4180, "firstname": "Dale", "lastname": "Adams", "age": 33, "gender": "M", "address": "467 Hutchinson Court", "e
  "index": { "_id": "20" },
  "account_number": 20, "balance": 16418, "firstname": "Elinor", "lastname": "Ratliff", "age": 36, "gender": "M", "address": "282 Kings Place", "e
  "index": { "_id": "25" },
  "account_number": 25, "balance": 40540, "firstname": "Virginia", "lastname": "Ayala", "age": 39, "gender": "F", "address": "171 Putnam Avenue",
  "index": { "_id": "32" },
  "account_number": 32, "balance": 48086, "firstname": "Dillard", "lastname": "Mcperson", "age": 34, "gender": "F", "address": "702 Quentin Stre
  "index": { "_id": "37" },
  "account_number": 37, "balance": 18612, "firstname": "Mcgee", "lastname": "Mooney", "age": 39, "gender": "M", "address": "826 Fillmore Place", "
  "index": { "_id": "44" },
  "account_number": 44, "balance": 34487, "firstname": "Aurelia", "lastname": "Harding", "age": 37, "gender": "M", "address": "502 Baycliff Terra
  "index": { "_id": "49" },
  "account_number": 49, "balance": 29104, "firstname": "Fulton", "lastname": "Holt", "age": 23, "gender": "F", "address": "451 Humboldt Street", "
  "index": { "_id": "51" },
  "account_number": 51, "balance": 14097, "firstname": "Burton", "lastname": "Meyers", "age": 31, "gender": "F", "address": "334 River Street", "e
  "index": { "_id": "56" },
  "account_number": 56, "balance": 14992, "firstname": "Josie", "lastname": "Nelson", "age": 32, "gender": "M", "address": "857 Tabor Court", "emp
  "index": { "_id": "63" },
  "account_number": 63, "balance": 6077, "firstname": "Hughes", "lastname": "Owens", "age": 30, "gender": "F", "address": "510 Sedgwick Street", "
  "index": { "_id": "68" },
  "account_number": 68, "balance": 44214, "firstname": "Hall", "lastname": "Key", "age": 25, "gender": "F", "address": "927 Bay Parkway", "employe
  "index": { "_id": "70" },
  "account_number": 70, "balance": 38172, "firstname": "Deidre", "lastname": "Thompson", "age": 33, "gender": "F", "address": "685 School Lane", "
  "index": { "_id": "75" },
  "account_number": 75, "balance": 40500, "firstname": "Sandoval", "lastname": "Kramer", "age": 22, "gender": "F", "address": "166 Irvington Plac
  "index": { "_id": "82" },
  "account_number": 82, "balance": 41412, "firstname": "Concetta", "lastname": "Barnes", "age": 39, "gender": "F", "address": "195 Bayview Place"
  "index": { "_id": "87" },
  "account_number": 87, "balance": 1133, "firstname": "Hewitt", "lastname": "Kidd", "age": 22, "gender": "M", "address": "446 Halleck Street", "em
  "index": { "_id": "94" },
  "account_number": 94, "balance": 41060, "firstname": "Brittany", "lastname": "Cabrera", "age": 30, "gender": "F", "address": "183 Kathleen Cour
  "index": { "_id": "99" },
  "account_number": 99, "balance": 47159, "firstname": "Ratliff", "lastname": "Heath", "age": 39, "gender": "F", "address": "806 Rockwell Place",
  "index": { "_id": "102" },
  "account_number": 102, "balance": 29712, "firstname": "Dena", "lastname": "Olson", "age": 27, "gender": "F", "address": "759 Newkirk Avenue", "e
  "index": { "_id": "107" },
  "account_number": 107, "balance": 14097, "firstname": "Burton", "lastname": "Meyers", "age": 31, "gender": "F", "address": "334 River Street", "e

```



# 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"
}
```

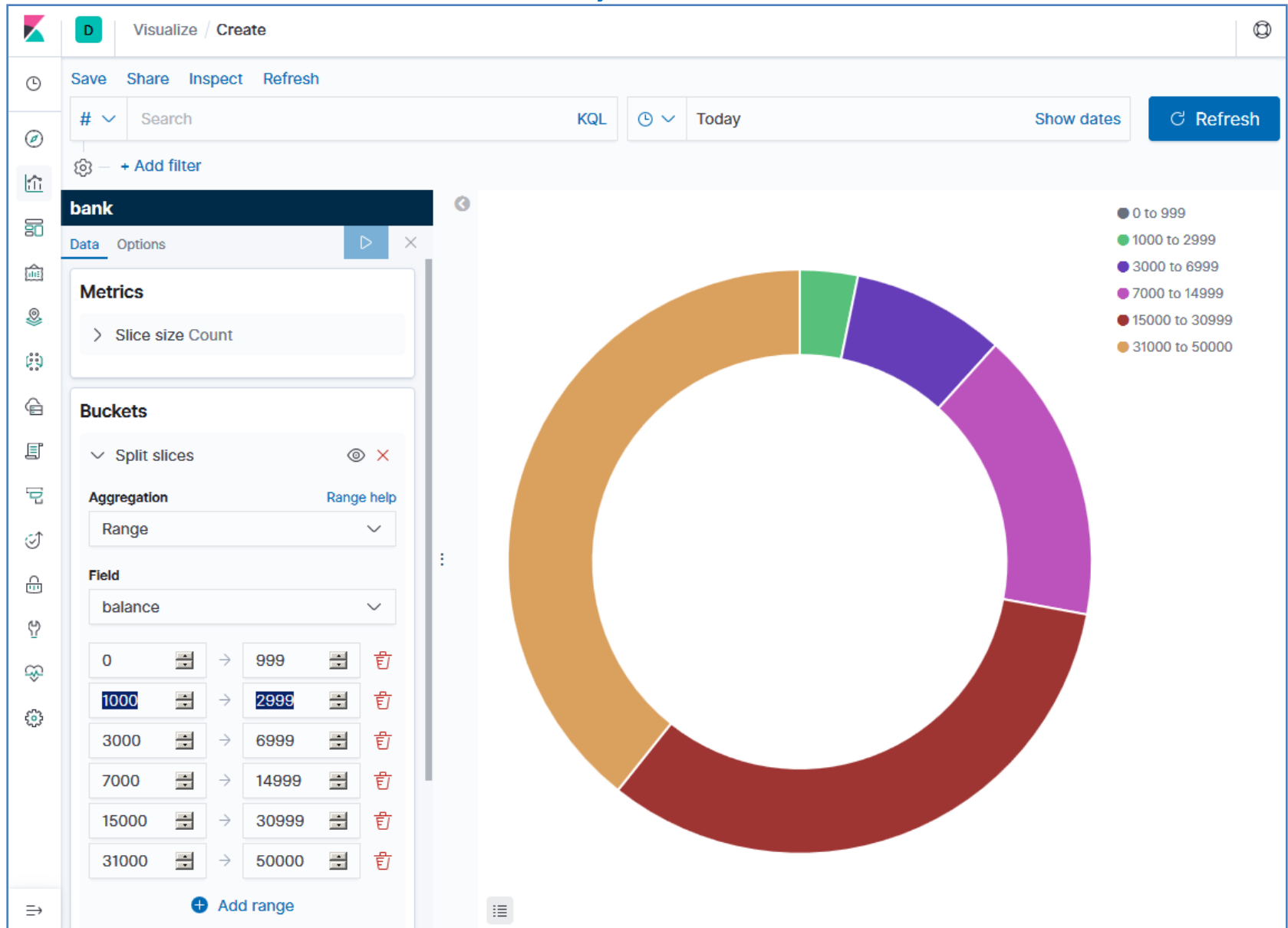
```
c:\Program Files\Elastic\curl-7.46.0-win64\bin>curl -XGET "localhost:9200/_cat/indices?v"
health status index          uuid                               pri rep docs.count docs.deleted store.size pri.store.size
yellow open   logstash-2015.05.20 rYz-xvAvTxC-XOAdBiWZBg  5  1    19000         0   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

# Kibana, Discover

The screenshot displays the Kibana Discover interface. At the top right, there are buttons for 'Lucene' and 'Refresh'. Below this, a search bar contains the query 'account\_number < 100 AND balance > 47500', with a 'KQL' button and another 'Refresh' button to its right. A 'Discover' tab is highlighted in the top left. On the left sidebar, a 'bank' filter is selected, and a list of 'Selected fields' includes account\_number, address, age, balance, city, email, employer, firstname, gender, lastname, and state. Below this, 'Available fields' includes \_id and \_index. The main area shows a table with 5 hits, with the count '5 hits' highlighted in a red box. The table columns are balance, account\_number, address, city, age, email, employer, firstname, gender, lastname, and state.

	balance	account_number	address	city	age	email	employer	firstname	gender	lastname	state
>	48,086	32	702 Quentin Street	Veguita	34	dillardmcperson@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	wilcoxseillers@confrenzy.com	Confrenzy	Wilcox	M	Sellers	MT
>	49,671	97	512 Cumberland Walk	Fredricktown	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

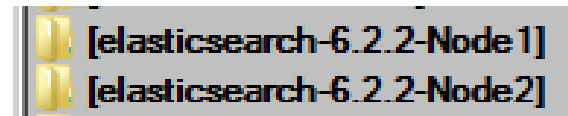
# Kibana, Visualize



# 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`

The screenshot displays the Kibana dashboard for an Elasticsearch cluster. At the top, navigation tabs include 'overview', 'nodes', 'rest', and 'more'. The current view is 'overview', showing cluster statistics: 'AG-cluster', 2 nodes, 1 index, 10 shards, 1,000 docs, and 966.54KB of data. Below the statistics, there are filters for indices and nodes. The main table shows the 'bank' index with 5 shards (2 per node) and 1,000 documents. The nodes are 'node-AG-1' and 'node-AG-2', both at IP 127.0.0.1. Each node's shard allocation is visualized with a row of five boxes numbered 0 to 4. On node-AG-1, shards 0, 1, 2, 3, and 4 are all highlighted in green, indicating they are in a 'green' state. On node-AG-2, shards 0, 1, 2, 3, and 4 are also highlighted in green. The 'heap' tab is selected for each node, showing a red bar representing memory usage.

# Jeden klaster, dwa node-y

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

The screenshot displays the Kibana interface for an Elasticsearch cluster. At the top, navigation tabs include 'overview', 'nodes', 'rest', and 'more'. The current view is 'overview', showing a cluster named 'AG-cluster' with 2 nodes, 2 indices, 20 shards, 112,393 documents, and 45.24MB of data. Below this, there are filters for indices and nodes, and a pagination control showing '1-2 of 2'.

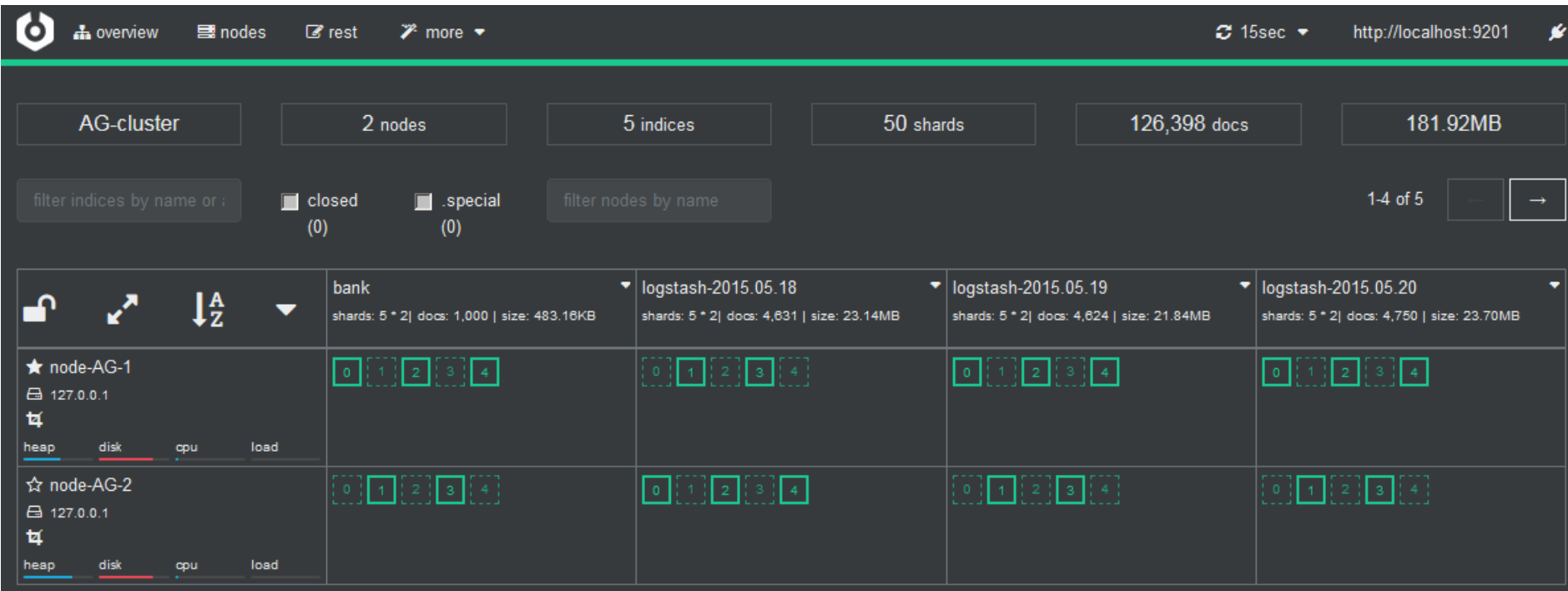
The main content area shows a table of indices. Two indices are visible: 'bank' and 'shakespeare'. The 'bank' index has 5 shards (2 replicas) and 1,000 documents. The 'shakespeare' index has 5 shards (2 replicas) and 112,393 documents. Below the index table, two nodes are listed: 'node-AG-1' and 'node-AG-2', both with IP address 127.0.0.1. Each node has a status bar with 'heap', 'disk', 'cpu', and 'load' indicators. The 'shakespeare' index is shown to be distributed across both nodes, with shards 0-4 on node-AG-1 and shards 0-4 on node-AG-2.

Index	Shards	Docs	Size
bank	5 * 2	1,000	483.16KB
shakespeare	5 * 2	112,393	22.11MB

Node	IP	Shards (bank)	Shards (shakespeare)
node-AG-1	127.0.0.1	0, 1, 2, 3, 4	0, 1, 2, 3, 4
node-AG-2	127.0.0.1	0, 1, 2, 3, 4	0, 1, 2, 3, 4

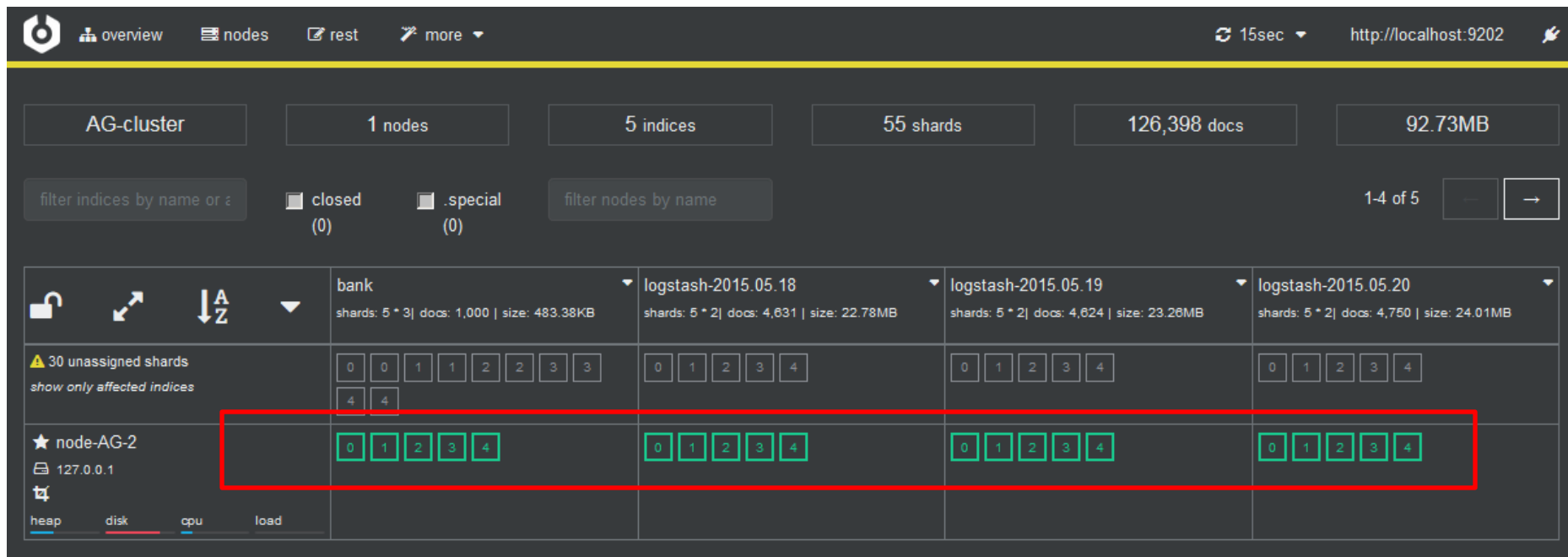
# Jeden klaster, dwa node-y

- 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/shakespeare/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 83MOZbNLSzGnjxxGdN4WWQ  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

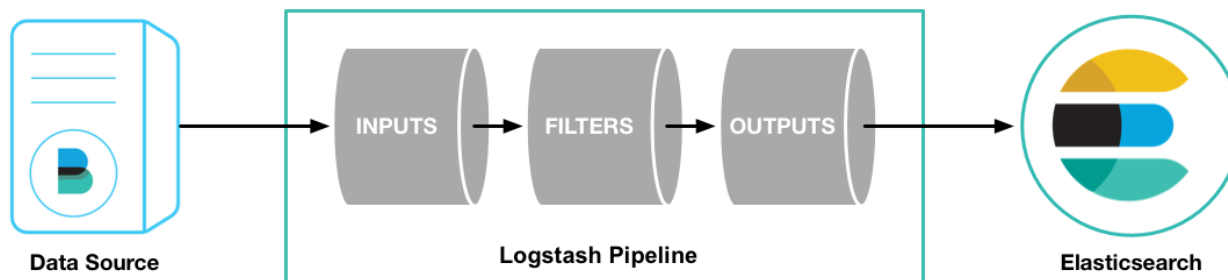
- wymagane: INPUTS, OUTPUTS

- 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 😊)

```
c:\Program Files\Elastic\logstash-6.2.2\bin>logstash -e "input { stdin { } } output { stdout { } }"
Sending Logstash's logs to c:/Program Files/Elastic/logstash-6.2.2/logs which is now configured via log4j2.pro
[2018-03-12T00:16:53,923][INFO ][logstash.modules.scaffold] Initializing module {:module_name=>"fb_apache", :d
"c:/Program Files/Elastic/logstash-6.2.2/modules/fb_apache/configuration"}
[2018-03-12T00:16:53,958][INFO ][logstash.modules.scaffold] Initializing module {:module_name=>"netflow", :dir
:/Program Files/Elastic/logstash-6.2.2/modules/netflow/configuration"}
[2018-03-12T00:16:54,123][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.queue
"c:/Program Files/Elastic/logstash-6.2.2/data/queue"}
[2018-03-12T00:16:54,132][INFO ][logstash.setting.writabledirectory] Creating directory {:setting=>"path.dead_
ue", :path=>"c:/Program Files/Elastic/logstash-6.2.2/data/dead_letter_queue"}
[2018-03-12T00:16:54,364][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because
command line options are specified
[2018-03-12T00:16:54,455][INFO ][logstash.agent ] No persistent UUID file found. Generating new UUID
bad064f4-1762-4ac7-a41b-67198c86081d", :path=>"c:/Program Files/Elastic/logstash-6.2.2/data/uuid"}
[2018-03-12T00:16:55,486][INFO ][logstash.runner ] Starting Logstash {"logstash.version"=>"6.2.2"}
[2018-03-12T00:16:58,462][INFO ][logstash.agent ] Successfully started Logstash API endpoint {:port=
[2018-03-12T00:16:59,403][INFO ][logstash.pipeline ] Starting pipeline {:pipeline_id=>"main", "pipeline
>4, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50}
[2018-03-12T00:16:59,635][INFO ][logstash.pipeline ] Pipeline started successfully {:pipeline_id=>"main"
>"#<Thread:0x6ab54065 run>"}
The stdin plugin is now waiting for input:
[2018-03-12T00:16:59,769][INFO ][logstash.agent ] Pipelines running {:count=>1, :pipelines=>["main"]}
hello world
2018-03-11T23:17:27.842Z DOM hello world
```

- 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_"=>{"dynam
ic_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 successfully {: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)

Dev Tools

Console

1	GET /_cat/indices?v	1	health	status	index	uuid	pri	rep	docs.count
2		2	yellow	open	bank	IFDCxNOzSxWPEiMD4RoFug	5	1	1000
3		3	yellow	open	my_index	RFAOPhuIRvmzUJrYclsoqw	5	1	1
4		4	yellow	open	logstash-2015.05.20	rYz-xvAvTxC-XOAdBiWZBg	5	1	19000
5		5	yellow	open	shakespeare	34B6PBOTSIuxcPxpZ5WtdQ	5	1	111396
6		6	green	open	.kibana	CEV-t-2SRaQRQ6866P-JhA	1	0	146
7		7	yellow	open	logstash-2018.03.13	1ni0nvs9QdikOAXjK5rF6w	5	1	2
8		8	yellow	open	logstash-2015.05.19	FW5rWPmuRdqISR_VAT486Q	5	1	18496
9		9	yellow	open	logstash-2015.05.18	JN95cZkTTPafzMQQ9cqE6A	5	1	18524



# 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
1	yellow	open	bank	IFDCxNOzSxWPEiMD4RoFug	5	1	1000	0
2	yellow	open	my_index	RFAOPhuIRvmzUJrYclsoqw	5	1	1	0
3	yellow	open	logstash-2015.05.20	rYz-xvAvTxC-XOAdBiWZBg	5	1	19000	0
4	yellow	open	shakespeare	34B6PBOTSIuxcPxpZ5WtdQ	5	1	111396	0
5	green	open	.kibana	CEV-t-2SRaqRQ6866P-JhA	1	0	146	7
6	yellow	open	logstash-2015.05.19	FW5rWpDmuRdqT5R_VAT486Q	5	1	18496	0
7	yellow	open	logstash-2018.03.13	ITqrtERvTa67Syd-jszc6w	5	1	100	0
8	yellow	open	logstash-2015.05.18	JN95cZkTTPafzMQQ9cqE6A	5	1	18524	0
9								
10								

```
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

1 hit

Gramacki

Uses lucene query syntax

Add a filter +

logstash-2018\*

Selected Fields

? \_source

Available Fields

@timestamp

@version

\_id

\_index

# \_score

\_type

host

message

path

? type

\_source

message: Artur Gramacki @version: 1 @timestamp: March 13th 2018, 13:38:06.691 host: DOM type: TutorialLOGs path: c:/Program Files/Elastic/data/logstash-tutorial-dataset \_id: B1peH2IB3vJmeVqPJ8EL \_type: doc \_index: logstash-2018.03.13 \_score: 2.773

Table JSON

View single document

@timestamp	March 13th 2018, 13:38:06.691
@version	1
_id	B1peH2IB3vJmeVqPJ8EL
_index	logstash-2018.03.13
# _score	2.773
_type	doc
host	DOM
message	Artur Gramacki
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 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	<code>clientip</code>
User ID	<code>ident</code>
User Authentication	<code>auth</code>
timestamp	<code>timestamp</code>
HTTP Verb	<code>verb</code>
Request body	<code>request</code>
HTTP Version	<code>httpversion</code>
HTTP Status Code	<code>response</code>
Bytes served	<code>bytes</code>
Referrer URL	<code>referrer</code>
User agent	<code>agent</code>

# 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

@timestamp March 13th 2018, 16:53:15.638

@version 1

\_id C04QIGIBQretUBI310dd

\_index logstash-2018.03.13

\_score 2.686

\_type doc

agent "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"

auth -

bytes 24747

clientip 87.169.99.232

host DOM

httpversion 1.1

ident -

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

referrer "https://www.google.de/"

request /presentations/puppet-at-loggly/puppet-at-loggly.pdf.html

response 200

timestamp 04/Jan/2015:05:23:41 +0000

type TutorialLOGs

verb GET

# Logstash

- Pobieranie danych z **pliku** (INPUT) i wrzucanie ich do Elasticsearch (OUTPUT). Dodatkowo dane zostaną poddane parsowaniu (FILTER)
  - użyjemy dodatkowo filtra **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

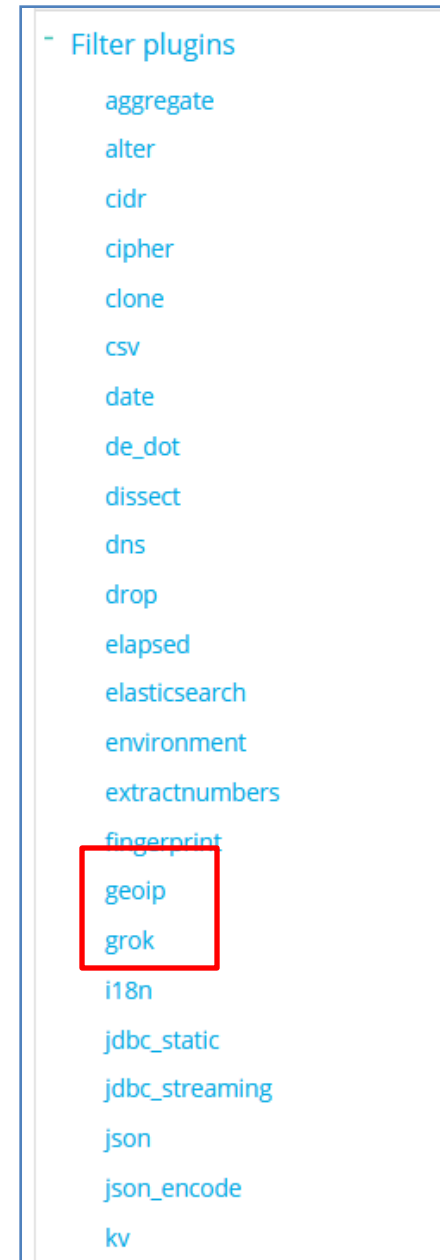
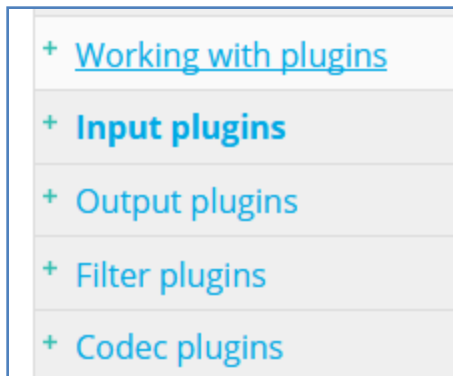
```
t message      @ @ [] * 87.169.99.232 - - [04/Jan/2015:05:23:41 +0000] "GET /presentations/puppet-at-loggly/puppet-at-loggly.pdf.html HTTP/1.1" 200 24747 "https://www.google.de/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36"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

```
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-snmptap (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)

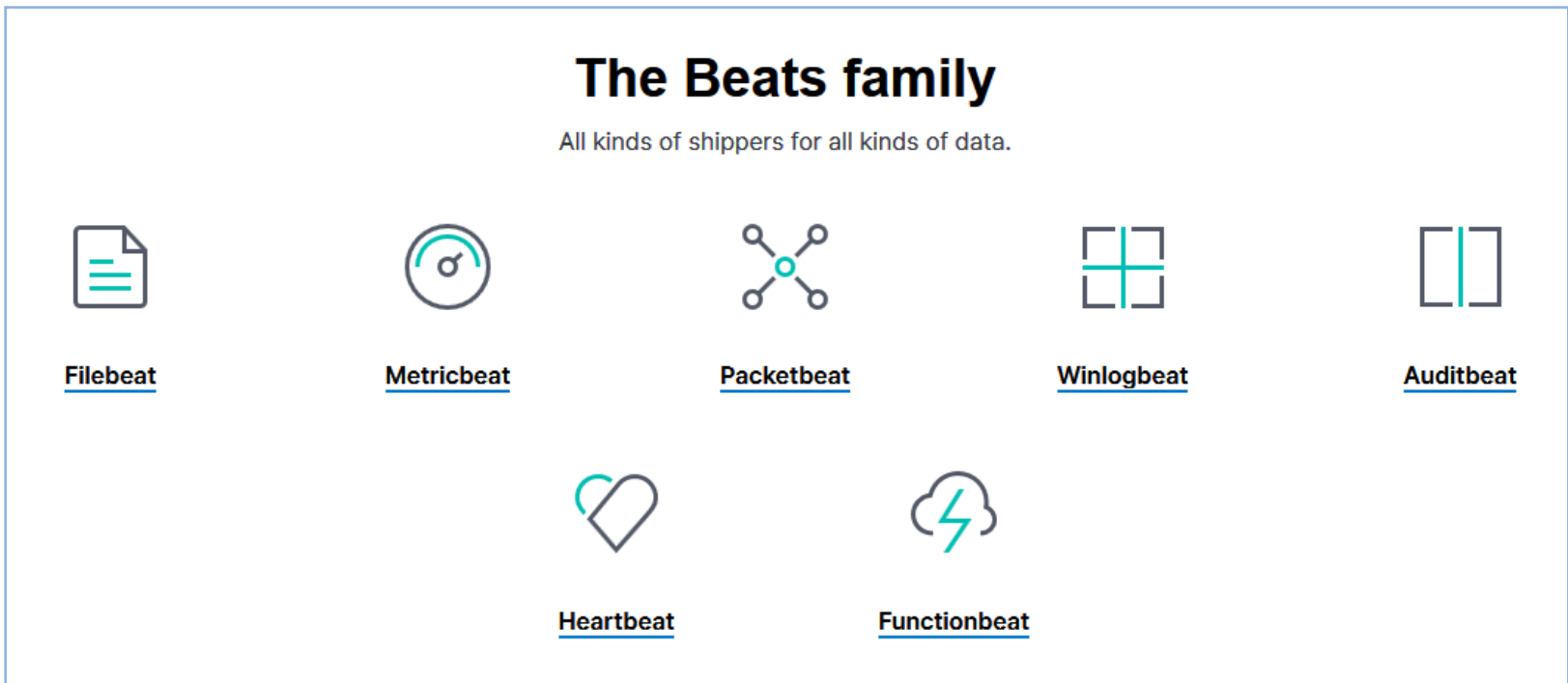
- Beats to współpracujące z produktami Elastic niezależne moduły (plugin-y), wspomagające akwizycję danych



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

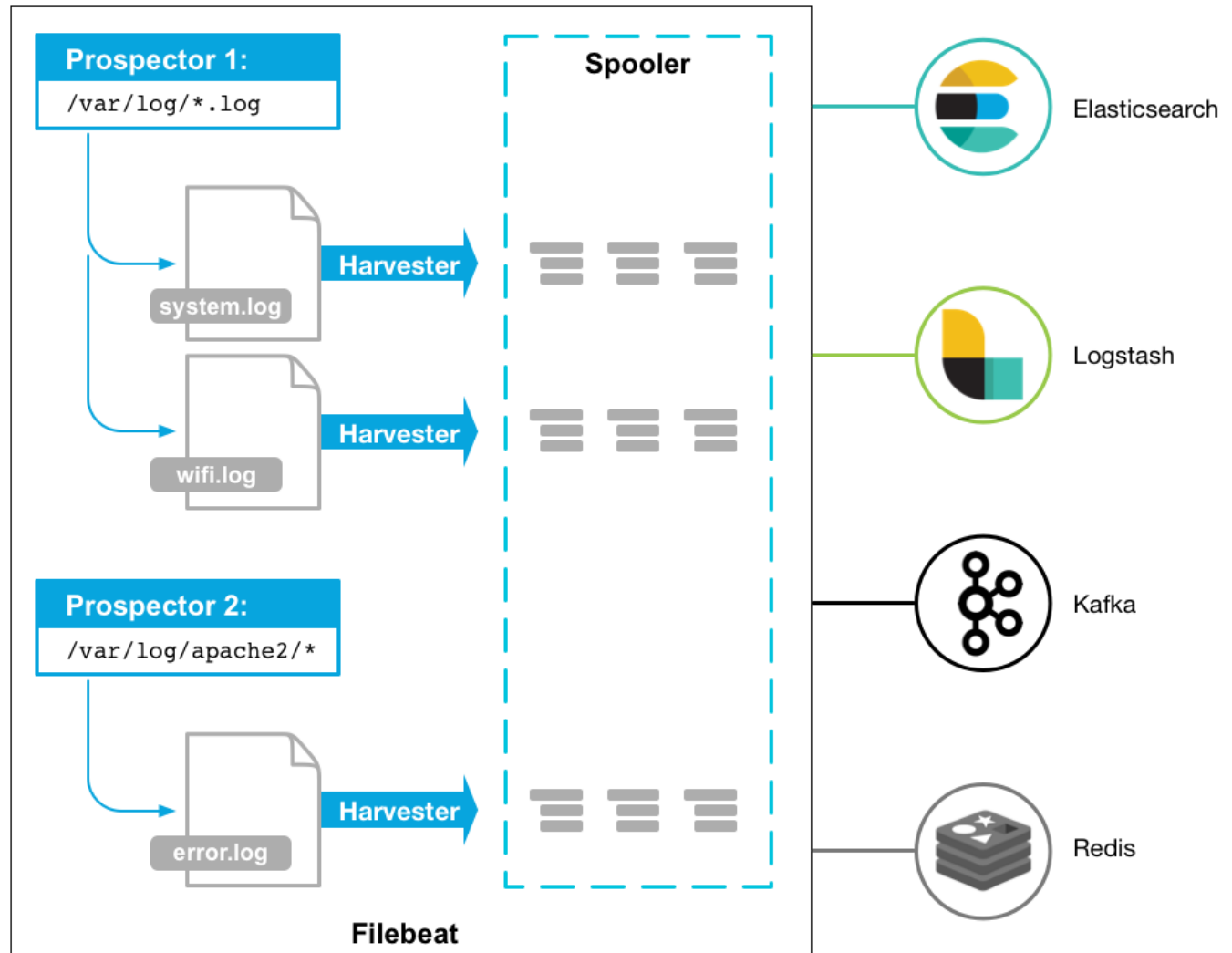
# Beats (data shippers)

- Rodzina Beats



# Filebeat

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

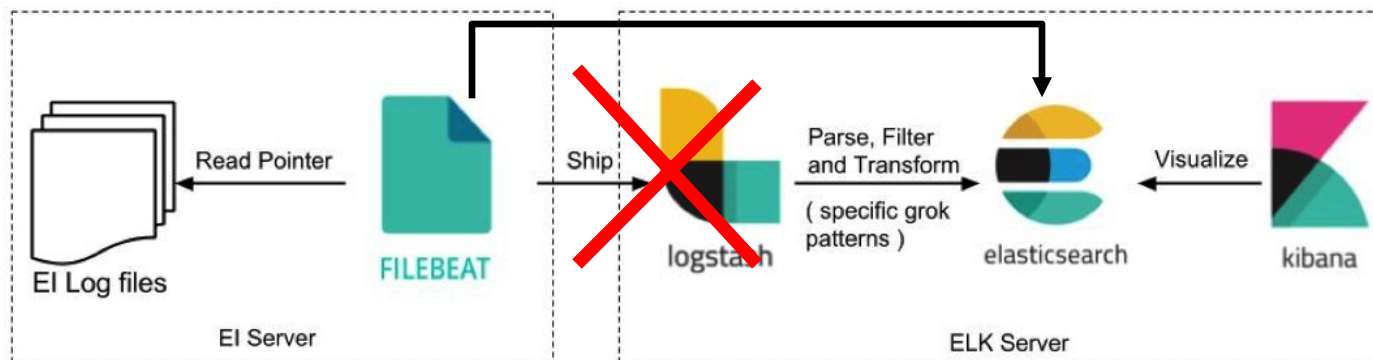


# Filebeat

- 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)

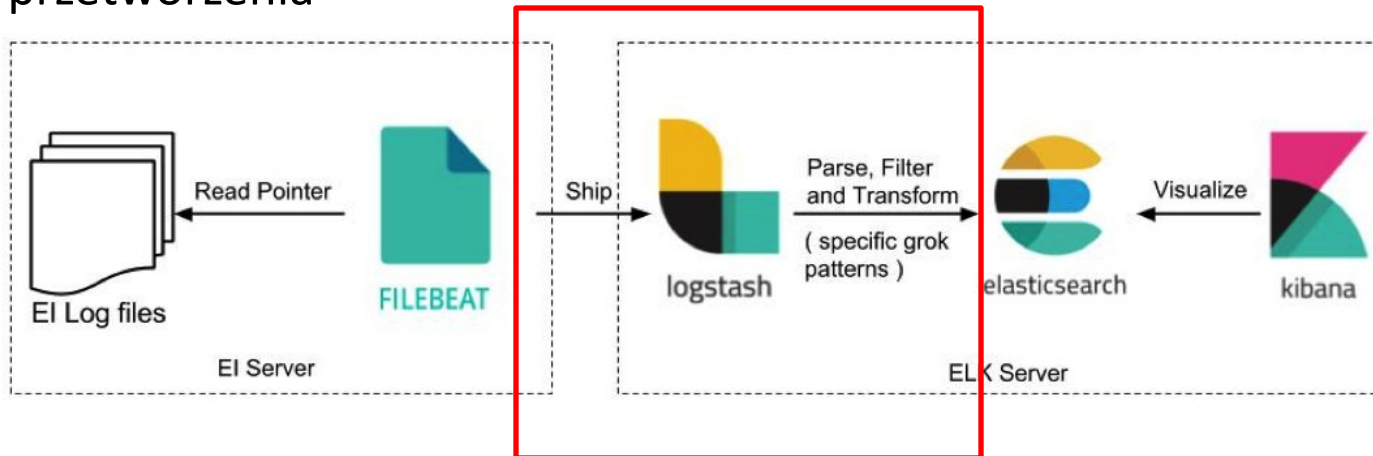
```
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/ima
83.149.9.216 - - [04/Jan/2015:05:13:42 +0000] "GET /presentations/logstash-monitorama-2013/ima
83.149.9.216 - - [04/Jan/2015:05:13:44 +0000] "GET /presentations/logstash-monitorama-2013/plu
83.149.9.216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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> cd c:\database\Filebeat\
PS C:\database\Filebeat> .\install-service-filebeat.ps1
.\install-service-filebeat.ps1 : File C:\database\Filebeat\install-service-filebeat.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ .\install-service-filebeat.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

```
c:\database\Filebeat>PowerShell.exe -ExecutionPolicy UnRestricted -File .\install-service-filebeat.ps1
```

Status	Name	DisplayName
Stopped	filebeat	filebeat

Faks	Umożliwia wysyłanie ...	Ręczny	Usługa sieciowa
filebeat		Działa	System lokalny
Foldery robocze	Ta usługa synchroniz...	Ręczny	Usługa lokalna



# 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
```

Faks	Umożliwia wysyłanie ...	Ręczny	Usługa sieciowa
filebeat	Działa	Automatyczny	System lokalny
Foldery robocze	Ta usługa synchroniz...	Reczny	Usługa lokalna

- 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

1	green	open	.kibana	JwM7_1z_QAiodNVL3ts-Ug	1	0	142	0	134.6kb	134.6kb
2	yellow	open	filebeat-6.2.2-2018.05.12	bY-bap63SG24VuIYu02esQ	3	1	100	0	48kb	48kb
3	yellow	open	shakespeare	AqkXLUInQRSpB+Nt05s87Q	5	1	111393	0	22mb	22mb
4	yellow	open	bank	yXN1fRsPTGqJ8CMpjwqcEg	5	2	1000	0	483.1kb	483.1kb
5	yellow	open	logstash-2015.05.20	Ei7jGmzTSouhfKY1lWfNZQ	5	1	4750	0	23mb	23mb
6	yellow	open	logstash-2015.05.19	uIMAoewOTGSgZ1JSNQNjDw	5	1	4624	0	21mb	21mb
7	yellow	open	logstash-2015.05.18	83MOZbNLSzGnjxGdN4WwQ	5	1	4631	0	22.4mb	22.4mb

# Filebeat – ćwiczenie 1

- Wyniki w Kibana

The screenshot shows the Kibana interface with the following components:

- Search Bar:** Search... (e.g. status:200 AND extension:PHP) with a search icon and "Uses lucene query syntax" text.
- Navigation Sidebar:** Discover, Visualize, Dashboard, Timelion, Dev Tools, Management.
- Filter Section:** filebeat-\*, Add a filter +, Selected Fields, Available Fields, @timestamp, t \_id, t \_index, # \_score, t \_type, t beat.hostname, t beat.name, t beat.version, t message, # offset, t prospector.type, t source.
- Time Range:** May 12th 2018, 00:30:32.810 - May 12th 2018, 00:45:32.810 — Auto.
- Histogram:** A bar chart showing the count of events per 30 seconds. The y-axis is labeled "Count" and ranges from 0 to 100. The x-axis is labeled "@timestamp per 30 seconds" and ranges from 00:32:00 to 00:45:00. A prominent bar is visible at approximately 00:43:00 with a count of about 100.
- Log Entries Table:**

Time	_source
▶ May 12th 2018, 00:44:47.952	@timestamp: May 12th 2018, 00:44:47.952 beat.name: DOM beat.hostname: DOM beat.version: 6.2.2 source: c:\Program Files\Elastic\data\logstash-tutorial-dataset offset: 24,480 message: b prospector.type: log _id: RLtgUWMBsOeFX_7zv-6 _type: doc _index: filebeat-6.2.2-2018.05.12 _score: -
▶ May 12th 2018, 00:44:47.952	@timestamp: May 12th 2018, 00:44:47.952 source: c:\Program Files\Elastic\data\logstash-tutorial-dataset offset: 24,484 message: b prospector.type: log beat.name: DOM beat.hostname: DOM beat.version: 6.2.2 _id: RrtgUWMBsOeFX_7zv-6 _type: doc _index: filebeat-6.2.2-2018.05.12 _score: -
▶ May 12th 2018, 00:44:47.952	@timestamp: May 12th 2018, 00:44:47.952 source: c:\Program Files\Elastic\da

# Filebeat – ćwiczenie 1

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

May 12th 2018, 16:19:49.662

```
@version: 1 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" source: c:\database\Filebeat_sample_data\logstash-tutorial-dataset offset: 16,591 host: ARTURG tags: beats_input_codec_plain_applied @timestamp: May 12th 2018, 16:19:49.662 prospector.type: log beat.version: 6.2.4 beat.name: ARTURG beat.hostname: ARTURG
```

Table JSON [View surrounding documents](#) [View single document](#)

@timestamp	May 12th 2018, 16:19:49.662
@version	1
_id	L1C4VGMBMqW7WcQy2jex
_index	filebeat-6.2.4-2018.05.12
_score	-
_type	doc
beat.hostname	ARTURG
beat.name	ARTURG
beat.version	6.2.4
host	ARTURG
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"
offset	16,591
prospector.type	log
source	c:\database\Filebeat_sample_data\logstash-tutorial-dataset
tags	beats_input_codec_plain_applied

# Filebeat – ćwiczenie 2

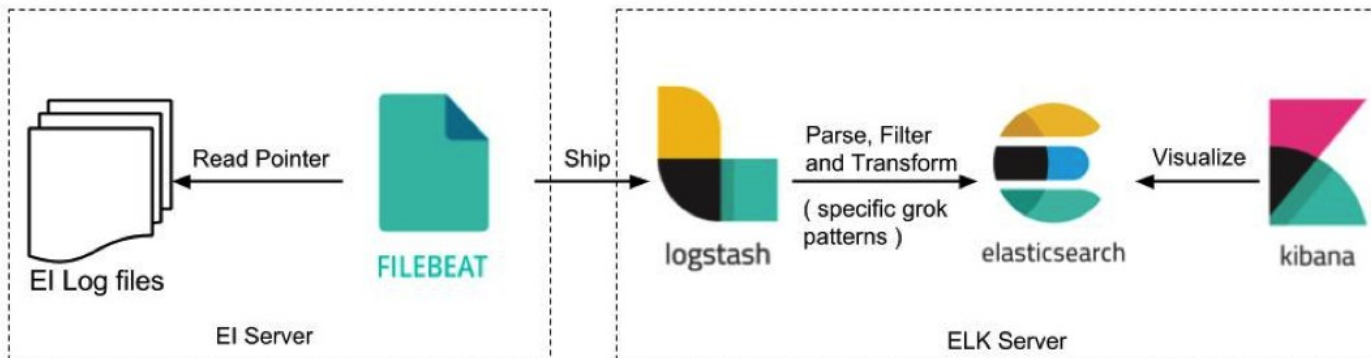
- Ćwiczenie 2, współpraca Filebeat z Logstash
  - konfiguracja Filebeat, aby korzystał z Logstash-a. **Uwaga:** albo Elasticsearch albo Filebeat. Oba wyjścia nie mogą być aktywne w tym samym czasie !

```
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

1	health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
2	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

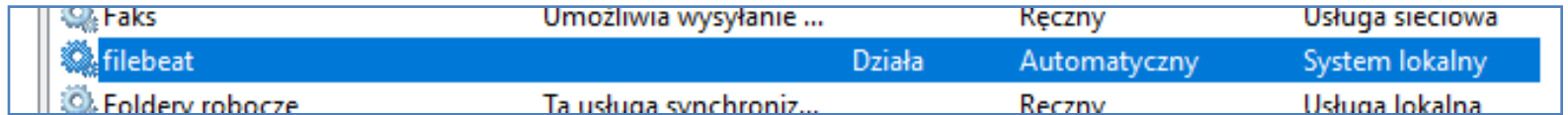
```
input {
  beats {
    port => 5044
  }
}

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



Faks	Umożliwia wysyłanie ...	Ręczny	Usługa sieciowa
filebeat	Działa	Automatyczny	System lokalny
Foldery robocze	Ta usługa synchroniz...	Ręczny	Usługa lokalna

- 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**

? clientip	🔍 📄 🗑️ * ⚠️ 108.174.55.234
? geopl.city_name	🔍 📄 🗑️ * ⚠️ Amsterdam
? geopl.continent_code	🔍 📄 🗑️ * ⚠️ EU
? geopl.country_code2	🔍 📄 🗑️ * ⚠️ NL
? geopl.country_code3	🔍 📄 🗑️ * ⚠️ NL
? geopl.country_name	🔍 📄 🗑️ * ⚠️ Netherlands
? geopl.ip	🔍 📄 🗑️ * ⚠️ 108.174.55.234
? geopl.latitude	🔍 📄 🗑️ * ⚠️ 52.35
? geopl.location.lat	🔍 📄 🗑️ * ⚠️ 52.35
? geopl.location.lon	🔍 📄 🗑️ * ⚠️ 4.9167
? geopl.longitude	🔍 📄 🗑️ * ⚠️ 4.9167
? geopl.postal_code	🔍 📄 🗑️ * ⚠️ 1091
? geopl.region_code	🔍 📄 🗑️ * ⚠️ NH
? geopl.region_name	🔍 📄 🗑️ * ⚠️ North Holland
? geopl.timezone	🔍 📄 🗑️ * ⚠️ Europe/Amsterdam
? host	🔍 📄 🗑️ * ⚠️ ARTURG
? httpversion	🔍 📄 🗑️ * ⚠️ 1.1
? ident	🔍 📄 🗑️ * ⚠️ -
t message	🔍 📄 🗑️ * 108.174.55.234 - - [04/Jan/2015:05:27:45 +0000] "GET /?flav=rss20 HTTP/1.1" 200 29941 "-" "-"
# offset	🔍 📄 🗑️ * 21,471

# 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

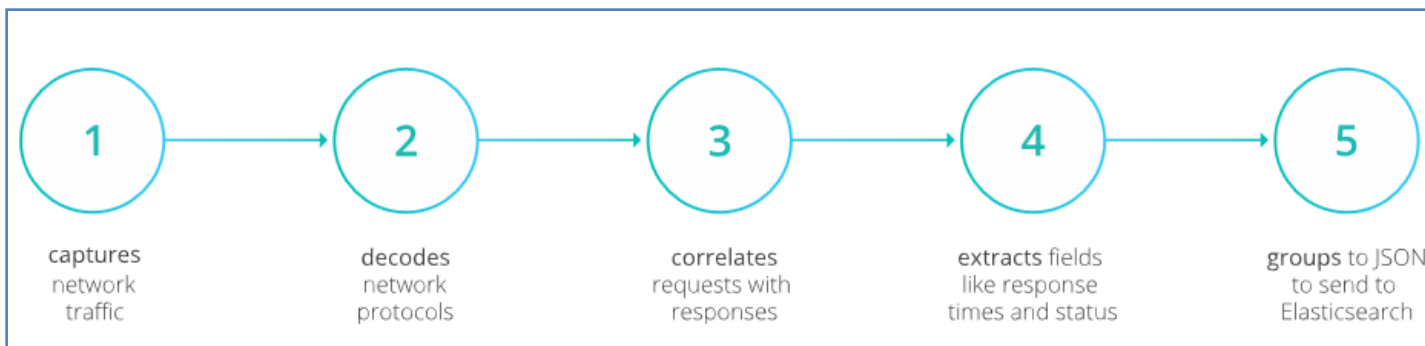
Step 3: Load the index template in Elasticsearch

Step 4: Set up the Kibana dashboards

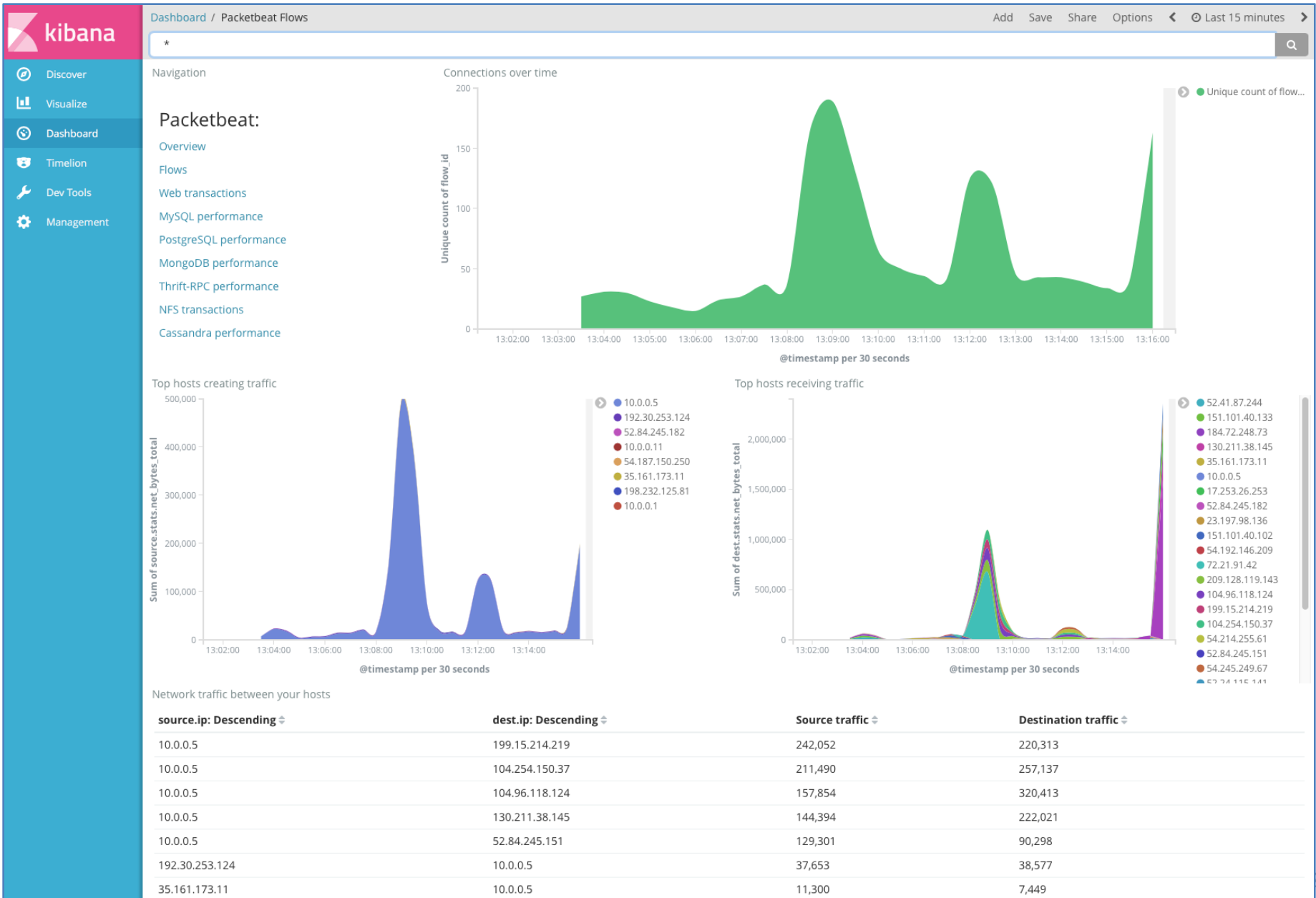
Step 5: Start Packetbeat

Step 6: View the sample Kibana dashboards

Repositories for APT and YUM



# Packetbeat





# 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": "2011-06-01"}
{"index":{"_id": "Hyperion"}}
{"name": "Hyperion", "author": "Dan Simmons", "release_date": "1989-05-26", "page_count": 482}
{"index":{"_id": "Dune"}}
{"name": "Dune", "author": "Frank Herbert", "release_date": "1965-06-01", "page_count": 604}
```

```
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 | ❶  
[LIKE pattern]] ❷
```

## SHOW COLUMNS [ FROM | IN ]?

```
[table identifier | ❶  
[LIKE pattern] ] ❷
```

## SHOW FUNCTIONS [LIKE pattern?]? ❶

## SHOW TABLES

```
[INCLUDE FROZEN]? ❶  
[table identifier | ❷  
[LIKE pattern ]]? ❸
```

```
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

<b>format</b>	<b>Accept HTTP header</b>	<b>Description</b>
<b>Human Readable</b>		
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
<b>Binary Formats</b>		
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": "sDXF1ZXJ5QW5kRmV0Y2gBAAAAAAAAAAAEWWdrRlVfSS1TbDYtcW9lc1FJNmlydw=="  
}
```



# 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

The screenshot shows the SQL Workbench/J interface connected to an Elasticsearch instance. The main window displays a table view for the 'elasticsearch.library' table. The table contains 24 rows of data, including columns for author, author.keyword, name.keyword, name, page\_count, and release\_date. The interface includes a menu bar (File, View, Workspace, Tools, Help), a toolbar with various icons, and a status bar at the bottom showing '6 Objects' and '2-24/24'.

author	author.keyword	name.keyword	name	page_count	release_date
Ursula K. Le Guin	Ursula K. Le Guin	The Left Hand of Darkness	The Left Hand of Darkness	304	1969-06-01 02:00:00
Robert A. Heinlein	Robert A. Heinlein	Starship Troopers	Starship Troopers	335	1959-12-01 02:00:00
Robert A. Heinlein	Robert A. Heinlein	The Moon is a Harsh Mistress	The Moon is a Harsh Mistress	288	1966-04-01 02:00:00
Ray Bradbury	Ray Bradbury	Fahrenheit 451	Fahrenheit 451	227	1953-10-15 02:00:00
Peter F. Hamilton	Peter F. Hamilton	Pandora's Star	Pandora's Star	768	2004-03-02 02:00:00
Orson Scott Card	Orson Scott Card	Ender's Game	Ender's Game	324	1985-06-01 03:00:00
Neal Stephenson	Neal Stephenson	Snow Crash	Snow Crash	470	1992-06-01 03:00:00
Margaret Atwood	Margaret Atwood	The Handmaid's Tale	The Handmaid's Tale	311	1985-06-01 03:00:00
Lois Lowry	Lois Lowry	The Giver	The Giver	208	1993-04-26 03:00:00
Kurt Vonnegut	Kurt Vonnegut	Slaughterhouse-Five	Slaughterhouse-Five	275	1969-06-01 02:00:00
James S.A. Corey	James S.A. Corey	Leviathan Wakes	Leviathan Wakes	561	2011-06-02 03:00:00
Isaac Asimov	Isaac Asimov	Foundation	Foundation	224	1951-06-01 02:00:00
Iain M. Banks	Iain M. Banks	Consider Phlebas	Consider Phlebas	471	1987-04-23 03:00:00
George Orwell	George Orwell	1984	1984	328	1985-06-01 03:00:00
Frank Herbert	Frank Herbert	Dune	Dune	604	1965-06-01 02:00:00
Frank Herbert	Frank Herbert	Dune Messiah	Dune Messiah	331	1969-10-15 02:00:00
Frank Herbert	Frank Herbert	Children of Dune	Children of Dune	408	1976-04-21 03:00:00
Frank Herbert	Frank Herbert	God Emperor of Dune	God Emperor of Dune	454	1981-05-28 03:00:00
Douglas Adams	Douglas Adams	The Hitchhiker's Guide to the Galaxy	The Hitchhiker's Guide to the Galaxy	180	1979-10-12 02:00:00
Dan Simmons	Dan Simmons	Hyperion	Hyperion	482	1989-05-26 03:00:00
Aldous Huxley	Aldous Huxley	Brave New World	Brave New World	268	1932-06-01 02:00:00
Alastair Reynolds	Alastair Reynolds	Revelation Space	Revelation Space	585	2000-03-15 02:00:00

# 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.

## - cat APIs

- cat aliases
- cat allocation
- cat count
- cat fielddata
- cat health
- cat indices
- cat master
- cat nodeattrs
- cat nodes
- cat pending tasks
- cat plugins
- cat recovery
- cat repositories
- cat task management
- cat thread pool
- cat shards
- cat segments
- cat snapshots
- cat templates

# Index APIs

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

## Index management:

- Create index
- Delete index
- Get index
- Index exists
- Close index
- Open index
- Shrink index
- Split index
- Clone index
- Rollover index
- Freeze index
- Unfreeze index

## Mapping management:

- Put mapping
- Get mapping
- Get field mapping
- Type exists

## Alias management:

- Add index alias
- Delete index alias
- Get index alias
- Index alias exists
- Update index alias

## Index settings:

- Update index settings
- Get index settings
- Analyze

## Index templates:

- Put index template
- Delete index template
- Get index template
- Index template exists

## Monitoring:

- Index stats
- Index segments
- Index recovery
- Index shard stores

## Status management:

- Clear cache
- Refresh
- Flush
- Synced flush
- Force merge

# 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. **C**reate, **R**ead, **U**ppdate, **D**elete)

- Wyświetlenie istniejących indeksów

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

- Utworzenie indeksu (**C**reate)

`PUT /customer?pretty`

- Wstawienie danych do indeksu (**C**reate, **U**ppdate)

- gdy wcześniej nie zdefiniujemy jawnie indeksu, zostanie on utworzony automatycznie

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

- Wyświetlenie danych o utworzonym dokumencie (**R**ead)

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

- Wykasowanie dokumentu (**D**elete)

`DELETE /customer/_doc/1?pretty`

- Wykasowanie indeksu (**D**elete)

`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 definiowania 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 wchodzi
  - pola meta (meta-fields), `_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 typu `text` (dla wsparcia przeszukiwania pełnotekstowego) oraz jako `keyword` (dla celów sortowania, agregacji)

# Mapowanie pól

- Przykład

```
PUT my_index ①
{
  "mappings": {
    "doc": { ②
      "properties": { ③
        "title": { "type": "text" }, ④
        "name": { "type": "text" }, ⑤
        "age": { "type": "integer" }, ⑥
        "created": {
          "type": "date", ⑦
          "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 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 (stylu)
  - 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**

```
1 {
2   "took": 10,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 1000,
12    "max_score": null,
13    "hits": [
14      {
15        "_index": "bank",
16        "_type": "account",
17        "_id": "0",
18        "_score": null,
19        "_source": {
20          "account_number": 0,
21          "balance": 16623,
22          "firstname": "Bradshaw",
23          "lastname": "Mckenzie",
24          "age": 29,
25          "gender": "F",
26          "address": "244 Columbus Place",
27          "employer": "Euron",
28          "email": "bradshawmckenzie@euron.com",
29          "city": "Hobucken",
30          "state": "CO"
31        },
32        "sort": [
33          0
34        ]
35      },
36      {
37        "_index": "bank",
38        "_type": "account",
39        "_id": "1",
40        "_score": null,
41        "_source": {
42          "account_number": 1,
43          "balance": 39225,
44          "firstname": "Amber",
45          "lastname": "Duke",
46          "age": 32,
```



# 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

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

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

```
1 {
2   "took": 10,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 1000,
12    "max_score": null,
13    "hits": [
14      {
15        "_index": "bank",
16        "_type": "account",
17        "_id": "0",
18        "_score": null,
19        "_source": {
20          "account_number": 0,
21          "balance": 16623,
22          "firstname": "Bradshaw",
23          "lastname": "Mckenzie",
24          "age": 29,
25          "gender": "F",
26          "address": "244 Columbus Place",
27          "employer": "Euron",
28          "email": "bradshawmckenzie@euron.com",
29          "city": "Hobucken",
30          "state": "CO"
31        },
32        "sort": [
33          0
34        ]
35      },
36      {
37        "_index": "bank",
38        "_type": "account",
39        "_id": "1",
40        "_score": null,
41        "_source": {
42          "account_number": 1,
43          "balance": 39225,
44          "firstname": "Amber",
45          "lastname": "Duke",
46          "age": 32,
```

# 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"]
}
```

```
1 {
2   "took": 9,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 1000,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "bank",
16        "_type": "account",
17        "_id": "25",
18        "_score": 1,
19        "_source": {
20          "account_number": 25,
21          "balance": 40540
22        }
23      },
```

```
{
  "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"
        }
      },
      {
        "_score": 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"
    }
  }
}
```

```
1 {
2   "took": 24,
3   "timed_out": false,
4   "_shards": {
5     "total": 5,
6     "successful": 5,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 1,
12    "max_score": 1,
13    "hits": [
14      {
15        "_index": "bank",
16        "_type": "account",
17        "_id": "20",
18        "_score": 1,
19        "_source": {
20          "account_number": 20,
21          "balance": 16418,
22          "firstname": "Elinor",
23          "lastname": "Ratliff",
24          "age": 36,
25          "gender": "M",
26          "address": "282 Kings Place",
27          "employer": "Scentric",
28          "email": "elinorratliff@scentric.com",
29          "city": "Ribera",
30          "state": "WA"
31        }
32      }
33    ]
34  }
35 }
```

# Eksplorowanie danych

- Przykłady wyszukiwania (bool query)

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

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

```
GET /bank/_search
{
  "query": {
    "bool": {
      "must_not": [
        { "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
```

```
15  "aggregations": {
16    "group_by_state": {
17      "doc_count_error_upper_bound": 20,
18      "sum_other_doc_count": 770,
19      "buckets": [
20        {
21          "key": "ID",
22          "doc_count": 27
23        },
24        {
25          "key": "TX",
26          "doc_count": 27
27        },
28        {
29          "key": "AL",
30          "doc_count": 25
31        },
32        {
33          "key": "MD",
34          "doc_count": 25
35        },
36        {
37          "key": "TN",
38          "doc_count": 23
39        },
40        {
41          "key": "MA",
42          "doc_count": 21
43        },
44        {
45          "key": "NC",
46          "doc_count": 21
47        },
48        {
49          "key": "ND",
50          "doc_count": 21
51        },
52        {
53          "key": "ME",
54          "doc_count": 20
55        },
56        {
57          "key": "MO",
58          "doc_count": 20
59        }
60      ]
61    }
}
```

# 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"
          }
        }
      }
    }
  }
}
```

```
"aggregations": {
  "group_by_state": {
    "doc_count_error_upper_bound": -1,
    "sum_other_doc_count": 918,
    "buckets": [
      {
        "key": "AL",
        "doc_count": 6,
        "average_balance": {
          "value": 41418.166666666664
        }
      },
      {
        "key": "SC",
        "doc_count": 1,
        "average_balance": {
          "value": 40019
        }
      },
      {
        "key": "AZ",
        "doc_count": 10,
        "average_balance": {
          "value": 36847.4
        }
      }
    ]
  }
}
```



# 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"
              }
            }
          }
        }
      }
    }
  }
}
```

```
15 ▾ "aggregations": {
16 ▾   "group_by_age": {
17 ▾     "buckets": [
18 ▾       {
19 ▾         "key": "20.0-30.0",
20 ▾         "from": 20,
21 ▾         "to": 30,
22 ▾         "doc_count": 451,
23 ▾         "group_by_gender": {
24 ▾           "doc_count_error_upper_bound": 0,
25 ▾           "sum_other_doc_count": 0,
26 ▾           "buckets": [
27 ▾             {
28 ▾               "key": "M",
29 ▾               "doc_count": 232,
30 ▾               "average_balance": {
31 ▾                 "value": 27374.05172413793
32 ▾               }
33 ▾             },
34 ▾             {
35 ▾               "key": "F",
36 ▾               "doc_count": 219,
37 ▾               "average_balance": {
38 ▾                 "value": 25341.260273972603
39 ▾               }
40 ▾             }
41 ▾           ]
42 ▾         }
43 ▾       },
44 ▾     {
45 ▾       "key": "30.0-40.0",
46 ▾       "from": 30,
47 ▾       "to": 40,
48 ▾       "doc_count": 504,
49 ▾       "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!"
    }
  }
}
```

```
1 - {
2   "took": 2,
3   "timed_out": false,
4   "_shards": {
5     "total": 1,
6     "successful": 1,
7     "skipped": 0,
8     "failed": 0
9   },
10  "hits": {
11    "total": 3,
12    "max_score": 0.44255546,
13    "hits": [
14      {
15        "_index": "my_index",
16        "_type": "my_type",
17        "_id": "3",
18        "_score": 0.44255546,
19        "_source": {
20          "title": "The quick brown fox jumps over the quick dog"
21        }
22      },
23      {
24        "_index": "my_index",
25        "_type": "my_type",
26        "_id": "1",
27        "_score": 0.42327404,
28        "_source": {
29          "title": "The quick brown fox"
30        }
31      },
32      {
33        "_index": "my_index",
34        "_type": "my_type",
35        "_id": "2",
36        "_score": 0.3081844,
37        "_source": {
38          "title": "The quick brown fox jumps over the lazy dog"
39        }
40      }
41    ]
42  }
43 }
```

# 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"
        }
      }
    ]
  }
}
```

# 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

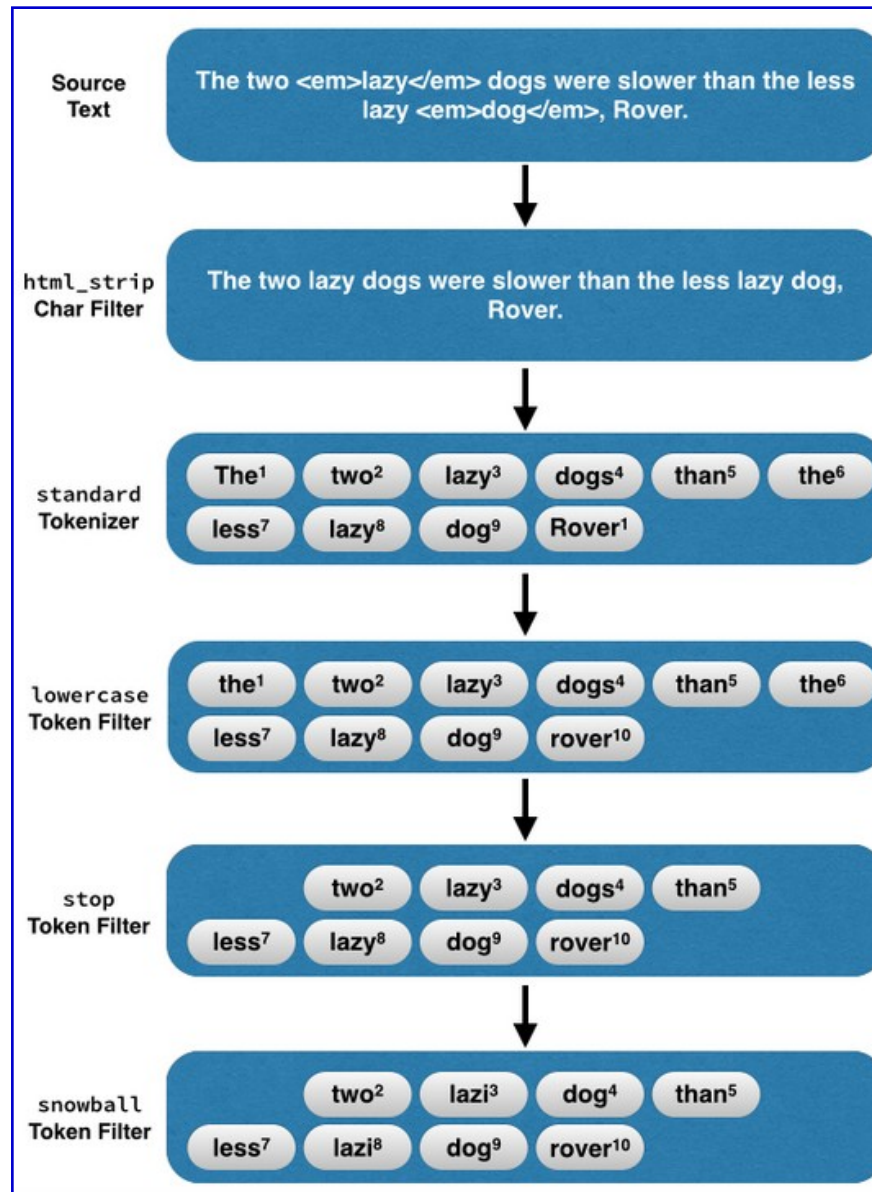
# Radzimy sobie z żywym językiem

- 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

- 1 We define the `std_english` analyzer to be based on the `standard` analyzer, but configured to remove the pre-defined list of English stopwords.
- 2 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 ]
- 3 The `my_text.english` field uses the `std_english` analyzer, so English stop words will be removed. The resulting terms are: [ old, brown, cow ]

```
POST my_index/_analyze
{
  "field": "my_text", 4
  "text": "The old brown cow"
}
```

```
POST my_index/_analyze
{
  "field": "my_text.english", 5
  "text": "The old brown cow"
}
```

```
PUT my_index
{
  "settings": {
    "analysis": {
      "analyzer": {
        "std_english": { 1
          "type": "standard",
          "stopwords": "_english_"
        }
      }
    },
    "mappings": {
      "_doc": {
        "properties": {
          "my_text": {
            "type": "text",
            "analyzer": "standard", 2
            "fields": {
              "english": {
                "type": "text",
                "analyzer": "std_english" 3
              }
            }
          }
        }
      }
    }
  }
}
```

# 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_"  
        }  
      }  
    }  
  }  
}
```

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

```
POST my_index/_analyze
```

```
{  
  "analyzer": "my_english_analyzer",  
  "text": "The 2 QUICK Brown-Foxes jumped over the 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"
        }
      }
    }
  }
}

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

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

```
gram Files\Elastic\elasticsearch-6.2.2, -Des.path.conf=c:\Program Files\Elastic\elasticsearch-6.2.2\config]  
[2018-03-17T20:27:27,238][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [aggs-matrix-stats]  
[2018-03-17T20:27:27,239][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [analysis-common]  
[2018-03-17T20:27:27,239][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [ingest-common]  
[2018-03-17T20:27:27,239][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [lang-expression]  
[2018-03-17T20:27:27,240][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [lang-mustache]  
[2018-03-17T20:27:27,240][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [lang-painless]  
[2018-03-17T20:27:27,240][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [mapper-extras]  
[2018-03-17T20:27:27,241][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [parent-join]  
[2018-03-17T20:27:27,241][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [percolator]  
[2018-03-17T20:27:27,241][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [rank-eval]  
[2018-03-17T20:27:27,242][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [reindex]  
[2018-03-17T20:27:27,242][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [repository-url]  
[2018-03-17T20:27:27,242][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [transport-netty4]  
[2018-03-17T20:27:27,242][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded module [tribe]  
[2018-03-17T20:27:27,243][INFO ][o.e.p.PluginsService ] [rU1Eg_t] loaded plugin [analysis-stempel]  
[2018-03-17T20:27:33,148][INFO ][o.e.d.DiscoveryModule ] [rU1Eg_t] using discovery type [zen]  
[2018-03-17T20:27:34,340][INFO ][o.e.n.Node ] initialized  
[2018-03-17T20:27:34,341][INFO ][o.e.n.Node ] [rU1Eg_t] starting ...
```



# Radzimy sobie z żywym językiem

- Pakiet Stempel (Stempel Polish Analysis Plugin)

```
POST _analize
{
  "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>