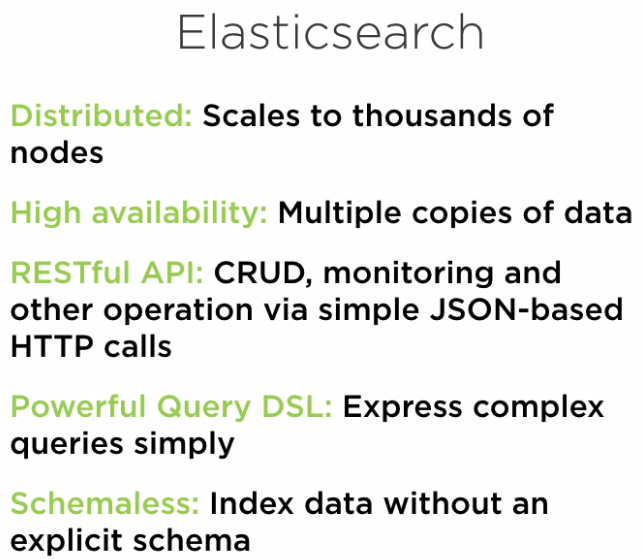
[Searching and Analyzing Data with Elasticsearch: Getting Started](https://app.pluralsight.com/library/courses/elasticsearch-analyzing-data/table-of-contents)



- moze byc uzyte do szybkiej analizy danych

- slowa sa grupowane od ilosci z odwolaniem do dokumentu – inverted index

- startuje w klastrze z 1 node z default



- by wiele klastrow, starujemy z nazwa klastra, elastic wysyla zapytanie do innych maszyn czy sa w tym klastrze



- do minuty miedzy doc indexed i serchable

- kazdy serwer (node) w klastrze wykonuje indexowanie dodanych dokumentow, kazdy pozwala na szukanie i operacje analizy

- kazdy nowe ma name i unique id – przydatne do operacji aministrowania na klastrze

- kazdy index do wyszukania dokumentu jest przechowywany w klastrze, jest unikalna nazwa pod ktora sa nody dolaczone, maszyny w klastrze w tej samej sieci

- dokumenty musza byc indexowane by potem szukane

- rozne typy i kategorie sa grupowane po indexie

- index to kolekcja podobnych dokumentow, nie musza dokladnie takie same byc

- index identyfikowany nazwa, dowolna ilosc indexow w klastrze

- w jednym indexie kilka typow dokumentow, to jakby logiczna czesc dokumentu w jednym indexie , np w indexie blog jest blog content i blog comments

- uzytkownik definiuje semantyki grup

- dokument z tym samym zbiorem pol nalezy do jednego typu

- dokument to jednostka informacji do indexowania, dok jako JSON,

- wszystkie dokumenty w indexie są, wiec wymagaja typu dokumentu i indexu, index i typ wyznaczaja zbior dokumentow

- moga byc duze dokumenty, wiec moga sie nie miescic na dysku 1 node, poza tym wolne gdy wszystkie zapytania szukania przez 1 node, lepiej na kilku na raz

- czyli index dzielimy na kilku node – **shards** your data , kazdy node ma tylko czesc indexu

- szukanie wspolbieznie na wielu node

- do tego kazdy shard ma replikacje – np. Po 2 kazdej czesci indexu – HA

- mozna od ) do wiele replik, jak duzo chcemy

- default – 5 shards i 1 replica , czyli kazdy shard 1 backup

Spr zycia klastra



Green – wszystko gra, yellow, jakas replica niegra, red – jakis shard nie gra

Nody :



Dalej jest CRUD operacje rest, bulk operacje, QUERY DSL i zapytania o agregacje

<https://app.pluralsight.com/player?course=elasticsearch-analyzing-data&author=janani-ravi&name=elasticsearch-analyzing-data-m2&clip=0&mode=live>

Centralized Logging with the Elastic Stack: Getting Started

* Przy instalacji zmieniam w elasticsearch.yml cluster, serwer I ip pod jakim serwer widoczny na zewnatrz (network.host)
* Musimy zwiekszyc dla el ilosc default in memory map



* Start i spr



* Wlaczenie uruchomienia es po kazdym boot

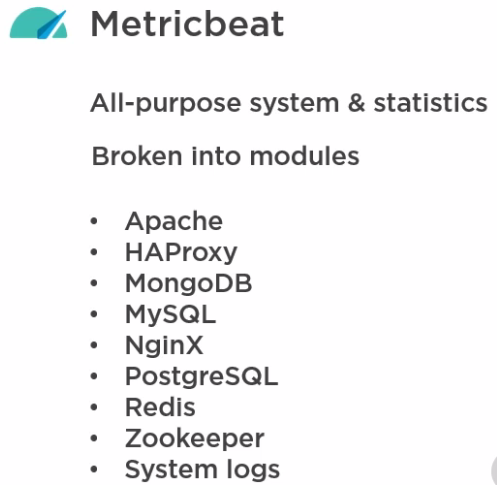


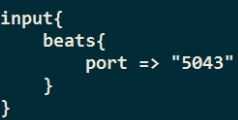
----

* 
* Pobranie z ELK, po index i wszystkie dokumenty



* 
* ------
* Kibana – to grafy i wizualizacje w Node.js
* W conf zmieniam server host i server name , ustawiam tez elasticseach.url
* 
* ---
* Dla danych z windows mamy windowsbeat, ale na lin metricsbeat , polaczenie do wielu systemow



* Mozemy dane o zuzyciu ram, procesor , dysk , memory, network, filesystem
* Logstash czyta wszystkie pliki .conf z katalogu
* 
* Dane z beats
* 
* Index moze byc dynamiczny, oparty na jakims polu, np. Entity\_name !!!!!!!!!!!!!!, tu kazda dana co z beat przychodzi ma metadata i w niej pole beat
* 
* Dodatkowow mozna document\_type okreslic
* 
* ----
* W wizualizacji Kibana jest podzial na subbucket, to dzieli dane na grupy danych
* Mozemy do dashboard dodac wykres
* U gory auto-refresh ustawiam co ile sec ma sie dashboard odswiezac
* Na linux moge uzyc Filebeat, do monitorowania plikow i metryk, bo na linux wszysatko w plikach dane o ram, cpu, dysku
* Mozna wysylac na logstash
* Dodatkowo packetbeat – analiza pakietow ilosci ze zrodel danych sql, redis, mongo, ..., do web traffic
* Mozna alerty wysylac z watcher, ale nie jest darmowy - trigger query, condition -> action

[Administering an Elasticsearch Cluster](https://app.pluralsight.com/library/courses/administering-elasticsearch-cluster/table-of-contents)









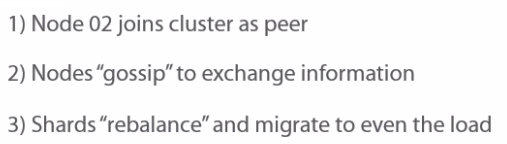


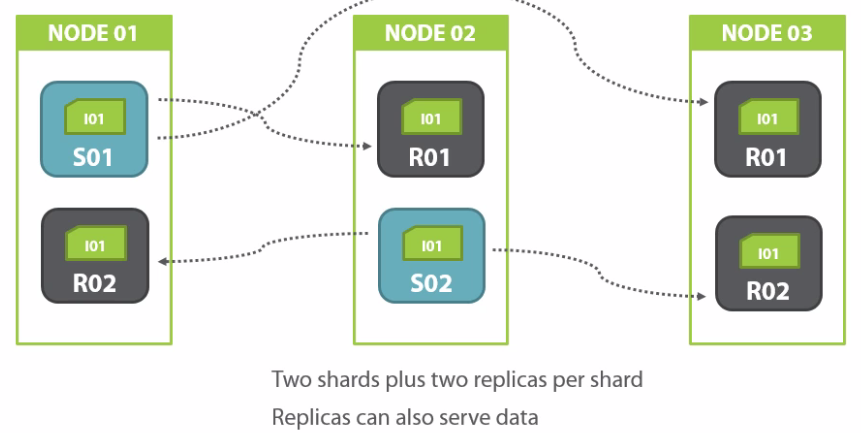
* Instalujemy elastic I start service

Konfiguracja na Ubuntu ()

--

Mam kilka shards na 1 node, pojawia sie 2gi, to czesc moze fizycznie przejsc na drugi – nody sie dogaduja by wymienic informacje, zachodzi rebalance i teraz 1 index na kilku nodach , to sie dzieje z automatu :





- tu nawet jak 2 nody padna, to dziala OK,

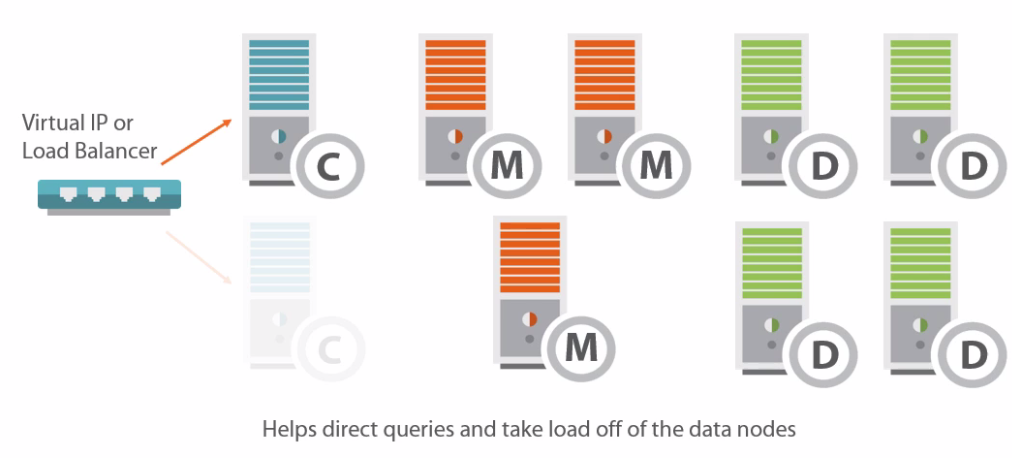
- jak node wysiada to ES automatycznie to wykrywa

- proboje zastapic brakujace shards and replicas na tych node co zostaly

--

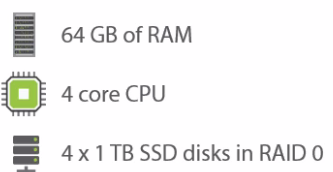
Na wejscie jest master/data/client node 1, ale nie jest to wydajne

Wazne rozdzielenie funkcji :

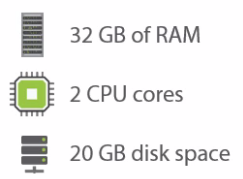


- maszynki duzo core CPU, data 64GB ram, master i client po 32 , jak najszybszy dysk , unikac network storage, szybkosc sieci najlepiej 10G, unikac klastru miedzy lokalizacjami , szybkosc jest najwolniejszego node,

Data :



Master/client :

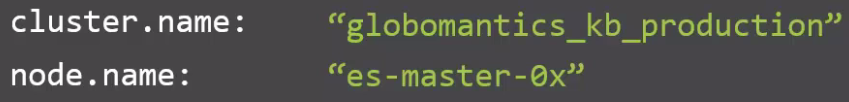


--do automatyzacji konfiguracji :

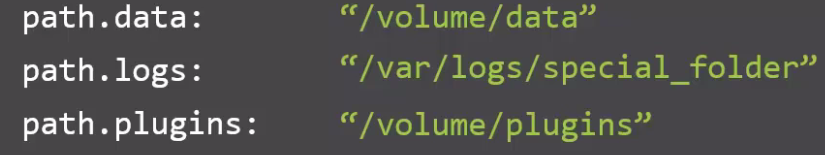


--

Elasticsearch.yml



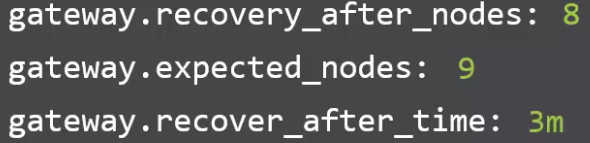
Opcjonalne nadpisanie sciezek :



Minimum 2 na quorum



Ponizsze flagi



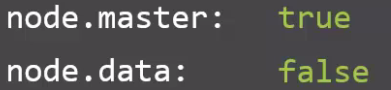
Jak wszystkie off i wstaje, to uwaza,ze wstal jak 8 on z 9ciu , czeka 3 minuty na wstanie , jak juz 8 to do 3 min na 9ty, jak powyzej to na 8my czeka

-

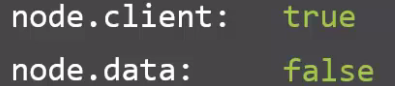
Z default ES automatycznie szuka innych ES nodes w sieci – multicasting, ciągle pull i czeka na response z noda co powinien byc w klastrze -> moze powodowac , ze jak na innych host bedzie taki sam cluster, to tam shards wysle, by uniknac okreslamy jakie nody pinga; to takie samo na wszystkich node powinno byc



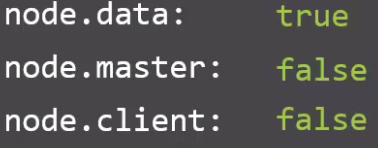
Master :



Client :



Data :



Dodatkowo, by zadan nie odbieral



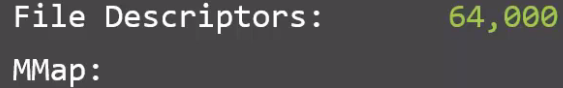
Ustawienie maszyny wirtualnej :



To dajac flage

 - to w /etc/environment

Dodatkowo



--

Tworze obraz z settings i pozostale z tego samego obrazu, ale inne name i hostname - /etc/hostname





- odkoment network.host na zewnetrzne ip



Head plugin :







---

Index mozna przez POST

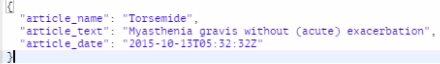


Moge tym powyzej strzelic pod : , by dodac index

Teraz na head widac rozklad shards

- jak chce dodac nowy dok, to pod





- po dodaniu w head widac ze ilosc dokumentow 2, bo 1 w shard i 1 w replica

- w head mozemy tez szukac

- jak wywale data node – shards przechodza z automatu

- jak wywale mastery 2 – to wszystko lezy, bo 2 z trzech, a 2 minimum by quorum bylo

- w head po prawej w status widzimy ile gok i jaki size, jak wywale index i zrobie nowy z replica 2 , to lacznie duzo wiecej miejsca

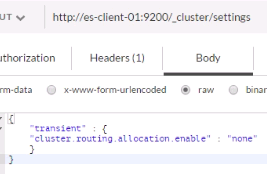
- wazna wielkosc doka i im wiekszy shard tym replikacja dluzsza

- rolowane inserty, widac pod tym samym aliasem w head

--

Settingd na ES z yaml mozemy zmieniac dynamicznie zamiast zmiany w pliku, ale nie wszystkie – bezpieczenstwo,

Te co transient nie zapisane



- po zmianie tej flagi wylanczam rebalancing i moge wylaczyc i updatowac dany node, potem zmieniam putem na all te flage , transient w pliku nie zmienia

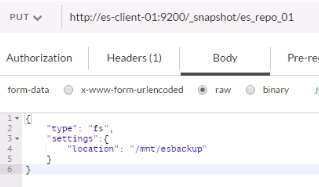
- dane jako snapshot, ale mozemy chciec zrobic backup danych i indexow na serwer jakis, mozemy tez w chmurze

Data i master musza wspoldzielic pamiec i musi byc serwer na backup, nie musi byc tam ES,

Wazne ustawienie sciezki repo w es.yml



Post by ustawic filesystem na snapshot



Wykonuje async post by zrobil snapshot

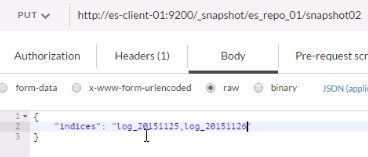


Potem ge spr status



Jak usune \_status, to mam dane tego snapshota/backupu

-Moge kolejny snap na wybrany index – PUT



- potem GET pokaze,ze zrobiony

- teraz ja usune insed – np. Z head i potem restore zrobie, z pobranym snapshot po GET



- wraca do normy

---------------------------------

Curator na osobnym serwerze dla backup/restore, to narzedzie command line

- mozna listowac indexy, pobierac starsze niz i na przyklad ich snapshot, usuwanie

------------------------

Monitorowanie

- mozna proste dane z head

- mozna z , zwraca kilka informacji

To daje podstawowy status, ale wiecej sie mozemy dowiedziec co jest nie tak z



- mozna pojedynczy node sprawdzac



Tu np. Dane o maszynce i ile danych przechowuje, ile indexow, zuzycie sprzetowe

- marvell jest do monitoringu klastrow , teraz w xpack

-- tu na wykresach dane o klastrze i o nodach