Politechnika Łódzka

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

Instytut Informatyki Stosowanej

**PRACA DYPLOMOWA MAGISTERSKA**

System ciągłego gromadzenia, przetwarzania i wizualizacji zdarzeń z wybranego

systemu informatycznego przy zastosowaniu oprogramowania ELK.

A system for continuous collection, processing and visualization

of events from a selected IT system using ELK software.

Autor: Szymon Woyda

Numer albumu: 227458

Opiekun pracy:

dr inż. Łukasz Sturgulewski

Łódź, Styczeń, 2020

**Streszczenie**

Powinno zmieścić się na jednej stronie A4 (½ po polsku, ½ po angielsku) – stanowi praktycznie skrót z rozdziału „Cel i zakres pracy” i zawiera opis tego jaki był cel pracy oraz co jest zawarte w poszczególnych rozdziałach części teoretycznej oraz tego, co zawiera cześć projektowa.

Monitoring, Automatyzacja, Kontrola, Bezpieczeństwo, Precyzyjna reakcja

Słowa kluczowe: Monitoring, Automatyzacja, Kontrola, Bezpieczeństwo, Precyzyjna reakcja

**Abstract**

To samo streszczenie lecz w języku angielskim.

Keywords: tutaj podać po przecinku 5 słów kluczowych związanych z niniejszą pracą dyplomową w języku angielskim.

SPIS TREŚCI

[Cel i zakres pracy 4](#_Toc33553896)

[Wstęp 6](#_Toc33553897)

[Rozdział 1 8](#_Toc33553898)

[Elastic Search 9](#_Toc33553899)

[Kibana 9](#_Toc33553900)

[Rozdział 2 10](#_Toc33553901)

[Początki Elastic Stack 10](#_Toc33553902)

[Beat 10](#_Toc33553903)

[Kto używa Elastic Stack? 11](#_Toc33553904)

[Salesforce 11](#_Toc33553905)

[CERN 11](#_Toc33553906)

[Green Man Gaming 11](#_Toc33553907)

[Konkurenci Elastic Stack 12](#_Toc33553908)

[Rozdział 3 13](#_Toc33553909)

[Logi lub wskaźniki (Logs or Metrics) - decyzja pojęciowa 13](#_Toc33553910)

[Dlaczego metryki 14](#_Toc33553911)

[Dlaczego logi 15](#_Toc33553912)

[Co ze śledzeniem? 15](#_Toc33553913)

[Jakiej metody powinienem zastosować? 16](#_Toc33553914)

[Rozdział 4 17](#_Toc33553915)

[Krótka lekcja historii 17](#_Toc33553916)

[Co to są Beats? 17](#_Toc33553917)

[Filebeat 17](#_Toc33553918)

[Packetbeat 18](#_Toc33553919)

[Metricbeat 18](#_Toc33553920)

[Heartbeat 18](#_Toc33553921)

[Auditbeat 19](#_Toc33553922)

[Winlogbeat 19](#_Toc33553923)

[Rozdział 5 20](#_Toc33553924)

[Instalacja stosu ELK na CentOS 7 20](#_Toc33553925)

[Rozdział 6 45](#_Toc33553926)

[Rozdział 7 54](#_Toc33553927)

[Rozdział 8 67](#_Toc33553928)

[Rozdział 9 79](#_Toc33553929)

[Podsumowanie i wnioski 91](#_Toc33553930)

[Literatura 92](#_Toc33553931)

[https://blog.gutek.pl/2017/01/16/elastic-wstep/ 93](#_Toc33553932)

[Wykaz rysunków 94](#_Toc33553933)

[Słownik skrótów i pojęć 96](#_Toc33553934)

# Cel i zakres pracy

Moja praca jest została dla administratorów tworzących bądź utrzymujących oprogramowanie, którzy działają zdalnie. Współpracują z różnymi urządzeniami, zbierają dane z czujników i zapewniają obsługę dla użytkownika. Pewnego dnia coś idzie nie tak i system nie działa zgodnie z oczekiwaniami. Może to nie być identyfikacja urządzeń lub brak otrzymywania danych z czujników, lub może właśnie wystąpił błąd czasu wykonania z powodu błędu w kodzie. Dzięki mojej pracy będzie w stanie wiedzieć na pewno.

Zaimplementuję w kodzie systemu odpowiednie punkty kontrolne, w których, jeśli system zwróci nieoczekiwany wynik, po prostu oflaguje go i powiadomi programistę. To jest koncepcja logowania.

Rejestrowanie umożliwia administratorom zrozumienie, co faktycznie robią systemy i jaki jest przepływ pracy. Duża część życia twórców oprogramowania oraz administratorów to monitorowanie, rozwiązywanie problemów i debugowanie. Rejestrowanie sprawia, że jest to znacznie łatwiejszy i płynniejszy proces.

Wizualizacja określonych zarejestrowanych danych ma następujące zalety:

* Zdalnie monitoruje działanie systemu.
* Przekazuj informacje w sposób przejrzysty i wydajny za pomocą grafik statystycznych, wykresów i grafik informacyjnych.
* Wydobywa wiedzę z danych wizualizowanych w postaci różnych wykresów.
* Podejmuje niezbędne działania, aby ulepszyć system.

Istnieje wiele sposobów wizualizacji surowych danych. W językach programowania Python i R istnieje wiele bibliotek, które mogą pomóc w wykreślaniu wykresów.

Celem niniejszej pracy magisterskiej jest zaplanowanie i opracowanie kompleksowego systemu monitorującego pracę systemu IT w średniej wielkości przedsiębiorstwie. Monitoring będzie obejmował najczęściej używanych typów urządzeń i aplikacji klienckich.

Topologia sieciowa zostanie zbudowana przy użyciu kontenerów Docker oraz LXD. Zostaną użyte rozwiązania Docker Compose oraz Vagrant. Stos technologiczny oparty będzie na ELK (ElasticSearch, Logstash, Kibana) oraz EFK, wraz z użyciem oprogramowania agregującego logi FluentD. Jest to oprogramowanie typu open source o bardzo bogatych możliwościach. Dzięki jego zastosowaniu administratorzy są w stanie w sposób bezpieczny i nieinwazyjny dowolną topologię sieci, zbudowaną z dowolnych urządzeń większości producentów oferowanych na rynku.

Ważnym aspektem pracy będzie skupienie się na wysokiej dostępności oraz ciągłej pracy urządzeń. Kolektor logów będzie zdecentralizowany oraz zduplikowany, odporny na awarie oraz w pełni aktualizowany, tak aby w trakcie wgrywania na niego updatów logi były nadal zbierane.

ElasticSearch oraz Kibana będą uruchomione przy użyciu kontenerów Docker na moim lokalnym komputerze. Dodatkowo będę uruchamiał klienckie maszyny wirtualne aby na nich postawić usługi do monitorowania logów systemowych oraz metryk. Będą to systemy operacyjne Linux: Ubuntu oraz CentOS. Usługi monitorujące na tych systemach to będą między innymi HeartBeat, PacketBeat, AuditBeat. Wszystkie zostaną dokładnie opisane w części teoretycznej oraz dokładnie przeanalizowane w części praktycznej. Tam również zostanie przeprowadzona kompleksowa instalacja, konfiguracja i weryfikacja poprawności działania. W miedzy czasie użyta przeze mnie technologia zostanie porównana do największych konkurentów na rynku oraz zostaną zaprezentowane wady i zalety wybranego przeze mnie rozwiązania na tle innych produktów.

W swojej pracy magisterskiej korzystam z ELK na licencji open source wraz z podstawowymi dodatkami. Na szczęście wersja bazowa umożliwia darmowe zarządzanie bezpieczeństwem przy użyciu dodatku X-Pack. Bez zaimplementowanych mechanizmów logowania każdy użytkownik może dostać się do webowego panelu zarządzania usługami. Po konfiguracji X-Pack będzie wymagane podanie nazwy użytkownika i hasła. Klientem logującym zdarzenia do ELK będzie maszyna wirtualna Ubuntu z zainstalowanym pakietem Filebeat. Elastic Search i Kibana zostaną uruchomione w kontenerach Docker. Zostaną wygenerowane certyfikaty oraz klucze prywatne. Aby używać dodatku X-Pack należy aktywować szyfrowanie SSL a co za tym wygenerować certyfikaty i klucze prywatne, które są konieczne do zestawienia tego szyfrowanego połączenia. W sytuacji kiedy na produkcji mamy wiele różnych indeksów: FileBeat, PacketBeat, WinLogBeat, Heartbeat itp. każdemu z użytkowników możemy nadać uprawnienia do odczytywania tylko niektórych z nich. Tak aby administracyjni odpowiedzialni za Windows mieli podgląd do WinLogBeata a administratorzy z innych zespołów mieli podgląd do logów z baz danych czy serwerów z systemem Linux.

# Wstęp

Środowisko IT staje się coraz bardziej skomplikowane, a administratorzy starają się nadążać za postępem aby w dalszym ciągu być świadomi aktywności wszystkich elementów tej układanki. Każdy moment niepoprawnie działającej infrastruktury przedsiębiorstwa powoduje wymierne straty materialne i wizerunkowe. Żadna szanująca się firma nie może pozwolić sobie na to aby system informatyczny dotknęła poważna awaria. Dzięki dobrze skonfigurowanemu systemowi monitoringu administratorzy są wstanie szybciej i precyzyjniej działać aby usunąć małą usterkę aby ta nie dokonała spustoszenia w innej, często krytycznej, części infrastruktury IT. Oczywiście na rynku dostępne są gotowe narzędzia do monitorowania różnych producentów, jednakże często są drogie i nie w pełni kompatybilne ze sobą. Kompleksowy system monitorujący wymaga aby wszystkie urządzenia w sieci raportowały do jednego komputera, który zbierze interesujące dane i kompleksowo je przetworzy a następnie z wizualizuje aby były one przejrzyste i czytelne do interpretacji przez użytkownika końcowego, w tym wypadku administratora. W celu zaprojektowania takiego mechanizmu zostanie użyta technologia ELK (ElasticSearch, Logstash, Kibana), która jest bezpłatna i w modelu open source. Ważnym aspektem będzie zachowanie pełnej wysokiej dostępności z naciskiem na ciągłości pracy systemu monitorującego, w rozumieniu bez przerw. Prace związane z utrzymaniem technologii ELK i jej ciągłym aktualizowaniu nie będą powodowały przerwania ciągłości zbierania logów i zdarzeń zachodzących w monitorowanym systemie komputerowym. Praca dyplomowa obejmować będzie szerokie spektrum monitorowania sieci, baz danych, serwerów oraz pozostałych urządzeń składających się na współczesny system informatyczny.

Wybrany temat jest ważny i istotny ze względu na fakt, że monitoring jest ciągły i nieprzerwany - odbywa się 24 godziny na dobę. Nawet w najlepiej zaprojektowanym systemie mogą się pojawić awarie a dzięki dobrej i stałej opiece jest spora szansa aby zniwelować ich potencjalne skutki. Monitorowanie IT obejmuje najważniejsze usługi i procesy w firmie, bez których biznes nie będzie działał poprawnie. Zabezpieczenie ich nie jest kwestią wyboru, ale odpowiedzialności. Monitorowanie IT jest nieocenionym źródłem informacji na temat większości procesów firmy, całego środowiska IT, zasobów, danych, ruchu i niektórych schematów. Umożliwia prawidłową diagnozę procesów, pozwala lepiej przewidywać i dostrzegać możliwości ich usprawnienia i modernizacji. To wygodny, szybki i niezawodny system ostrzegania. Niepokojące sygnały i powiadomienia przychodzą na pocztę elektroniczną, SMS - w dowolnej i optymalnej formie. Zapewnia to zdolność reagowania niemal w czasie rzeczywistym na wszelkie niepokojące zjawiska. Najgorsze, co może być, to dowiedzieć się o niepowodzeniu od klientów. Nie każde narzędzie można dostosować do konkretnych potrzeb i cech danej firmy. Monitoring IT jest inny. Może i powinien reagować na konkretne oczekiwania biznesowe i pomagać w osiągnięciu indywidualnych celów.

Niektóre schematy się powtarzają, ale nie warto patrzeć na nie szablonowo. Monitorowanie IT daje możliwość automatyzacji niektórych działań - w wielu przypadkach interwencja IT nie będzie konieczna. Ogólnie można powiedzieć, że odciąża zespół IT od wielu problemów, co daje dodatkowe możliwości i oszczędza dużo czasu. Monitoring jest nie tylko źródłem wiedzy - w rzeczywistości jest opcją kontrolną, dzięki której administrator nie musi działać w ciemno ale ma realną szansę na zwiększenie wydajności i produktywności. Skuteczne monitorowanie pomaga obniżyć koszty firmy - w tym koszty naprawy możliwych awarii i sytuacji krytycznych, których konsekwencje mogą być trudne do odkręcenia. Pomaga także w lepszym planowaniu budżetu firmy. Umożliwia bieżącą dokładną analizę procesów, zachowania użytkowników, śledzenie trendów w infrastrukturze IT, co pozwoli ustalić harmonogram działań i prac, a to bezpośrednio wpływa na całą organizację. dzięki systemowi raportowania uzyskujemy dokładny obraz aktualnej sytuacji w firmie. Administrator zna nie tylko ogólny zarys, ale także wszystkie niezbędne szczegóły, które najczęściej są fundamentalne. Monitorowanie IT daje możliwość świadczenia usług na stałym, wysokim poziomie. W końcu jeżeli firma działa płynnie - ma to pozytywny wpływ na jej klientów. Monitoring daje możliwość zrównoważonego rozwoju firmy, sugeruje kierunek zmian - na podstawie zweryfikowanych danych, a nie domysłów.

Stos ELK to zbiór trzech programów typu open source, które pomagają w uzyskiwaniu wglądu w czasie rzeczywistym na temat danych, które mogą być ustrukturyzowane lub nieustrukturyzowane. Można wyszukiwać i analizować dane za pomocą jego narzędzi z niezwykłą łatwością i wydajnością.

Społeczność zgromadzone wokół ELK znacznie się powiększa oraz liczba przypadków użycia wzrosta.

Niezależnie od tego, czy chodziło o znalezienie najlepszych N wyników w dżungli dokumentów tekstowych, analizowanie zdarzeń związanych z bezpieczeństwem, czy swobodne wycinanie i krojenie wskaźników, społeczność światowa wciąż przekraczała kolejne granice.

# Rozdział 1

Logstash to bardzo dobre narzędzie do monitorowania serwerów i usług w środowiskach produkcyjnych. Aplikacje w środowisku produkcyjnym generują różnego rodzaju dane dziennika, takie jak dzienniki dostępu, dzienniki błędów itp. Logstash może liczyć lub analizować liczbę błędów, dostępu lub innych zdarzeń za pomocą wtyczek filtrów. Ta analiza i liczenie mogą być wykorzystane do monitorowania różnych serwerów i ich usług.

Wszystko zaczęło się od Shaya Banona, który rozpoczął ją jako projekt open source, Elasticsearch, następca Compassa, który zyskał popularność, by stać się jednym z najlepszych silników baz danych open source. Później, w oparciu o rozproszony model pracy, Kibana została wprowadzona do wizualizacji danych obecnych w Elasticsearch. Wcześniej, w celu wprowadzenia danych do Elasticsearch, były potoki danych (stream), które dostarczyły konkretnego wejścia, za pomocą którego dane były wprowadzane do Elasticsearch. Jednak wraz z rosnącą popularnością ta konfiguracja wymagała narzędzia, dzięki któremu można było wprowadzać dane do Elasticsearch i mieć możliwość dokonywania różnych transformacji danych w celu uzyskania danych nieustrukturyzowanych, które miałyby pełną kontrolę nad sposobem ich przetwarzania. W oparciu o tę przesłankę narodził się Logstash, który następnie został włączony do stosu i razem te trzy narzędzia, Elasticsearch, Logstash i Kibana zostały nazwane stosem ELK.

Dane są odczytywane za pomocą Logstash i indeksowane do Elasticsearch. Później używana jest Kibana do odczytania indeksów z Elasticsearch i wizualizacji ich za pomocą wykresów i list. W swojej pracy analizuję te składniki oddzielnie i ich rolę, jaką odgrywają w tworzeniu stosu.

Logstash nie wysyła danych z jednego końca na drugi; pomaga administratorowi w zbieraniu surowych danych i modyfikowaniu/filtrowaniu ich w celu przekształcenia ich w coś sensownego, sformatowanego i zorganizowanego. Zaktualizowane dane są następnie wysyłane do Elasticsearch. Jeśli nie ma dostępnej wtyczki umożliwiającej odczytywanie danych z konkretnego źródła, zapisywanie danych w określonym miejscu lub modyfikowanie ich na swój sposób, Logstash jest na tyle elastyczny, że pozwala na pisanie własnych wtyczek.

Krótko mówiąc, Logstash jest Open Source, bardzo elastyczny, bogaty w wtyczki, może odczytywać dane z wybranej lokalizacji, normalizować je zgodnie ze zdefiniowanymi konfiguracjami i wysyłać je do określonego miejsca docelowego zgodnie z wymaganiami.

## Elastic Search

Wszystkie dane odczytane przez Logstash są wysyłane do Elasticsearch w celu indeksowania. Jest to o wiele więcej niż tylko indeksowanie. Elasticsearch służy nie tylko do indeksowania danych, ale jest to wyszukiwarka pełnotekstowa, wysoce skalowalna, rozproszona i oferuje wiele innych rzeczy. Elasticsearch zarządza i utrzymuje dane w formie indeksów, oferuje możliwość wyszukiwania, dostępu i agregowania danych za pomocą swoich interfejsów API. Elasticsearch opiera się na Lucene, dostarczając tym samym wszystkie funkcje, które oferuje Lucene.

## Kibana

Używa API Elasticsearch do odczytywania/zapytywania danych z indeksów Elasticsearch do wizualizacji i analizy w formie wykresów, diagramów i tabel. Ma postać aplikacji internetowej, zapewniającej wysoce konfigurowalny interfejs użytkownika, który pozwala na zapytywanie o dane, tworzenie wielu wykresów do wizualizacji oraz nadawanie rzeczywistego sensu przechowywanym danym.

Po zbudowaniu solidnego stosu ELK, w miarę upływu czasu, miało miejsce kilka ważnych i złożonych wymagań, takich jak uwierzytelnianie, bezpieczeństwo, powiadomienia itp. To wymaganie doprowadziło do powstania kilku innych narzędzi, takich jak Watcher (dostarczanie powiadomień i alarmów na podstawie zmian w danych), Shield (uwierzytelnianie i autoryzacja dla zabezpieczenia klastrów), Marvel (monitorowanie statystyk klastra), ES- Hadoop, Curator i Graph jako wymagania.

# Rozdział 2

## Początki Elastic Stack

Wszystkie zadania związane z odczytem danych zostały wykonane przy użyciu programu Logstash, ale jest to pochłaniające zasoby. Ponieważ Logstash działa na JVM, zużywa dużą ilość pamięci. Społeczność zdała sobie sprawę z potrzeby poprawy i uczynienia procesu pipelowania przyjaznym dla zasobów i lekkim. W 2015 roku narodził się projekt Packetbeat, który polegał na stworzeniu sieciowego analizatora pakietów, który mógłby czytać z różnych protokołów, analizować dane i przesyłać je do Elasticsearch. Ze względu na swoją lekkość, powstał nowy pomysł na Beats. Beats są napisane w języku programowania Go. Projekt bardzo się rozwinął, a teraz stos ELK nie był niczym więcej niż tylko Elasticsearch, Logstash i Kibana, ale Beats stał się również ważnym elementem.

## Beat

Odczytuje dane, przetwarza je i może przesłać do Elasticsearch lub Logstash. Różnica polega na tym, że są one lekkie, służą konkretnemu celowi i są instalowane jako środki. Dostępnych jest niewiele bitów, takich jak Topbeat, Filebeat, Packetbeat i tak dalej, które są obsługiwane i dostarczane przez Elastic.co oraz duża liczba bitów już napisanych przez społeczność. Jeśli masz jakieś szczególne wymagania, możesz napisać swój własny Beat używając biblioteki libbeat.

W prostych słowach, Beaty mogą być traktowane jako bardzo lekkie środki do wysyłania danych do Logstash'a lub Elasticsearch'a i oferować Ci infrastrukturę wykorzystującą bibliotekę libbeat do tworzenia własnych Beatów. Razem Elasticsearch, Logstash, Kibana i Beats stają się Elastycznym stosem, formalnie znanym jako ELK Stack. Elastic Stack nie tylko dodał Beaty do swojego zespołu, ale będzie zawsze używał tej samej wersji. Wersja startowa stosu Elastic Stack będzie miała postać 5.0.0 i ta sama wersja będzie miała zastosowanie do wszystkich komponentów.

Ta wersja i metoda wydania jest nie tylko dla Elastic Stack, ale także dla innych narzędzi z rodziny Elastic. Ze względu na tak wiele narzędzi pojawił się problem unifikacji, w którym każde narzędzie miało swoją własną wersję i każda wersja nie była kompatybilna ze sobą, co prowadziło do problemu. Aby rozwiązać ten problem, teraz wszystkie narzędzia zostaną zbudowane, przetestowane i wydane razem. Wszystkie te komponenty odgrywają znaczącą rolę w tworzeniu potoku. Podczas gdy Beats i Logstash są używane do zbierania danych, analizowania ich i wysyłania, Elasticsearch tworzy indeksy, które w końcu są wykorzystywane przez Kibana do tworzenia wizualizacji. Podczas gdy Elastic Stack pomaga w tworzeniu potoku, inne narzędzia dodają zabezpieczenia, powiadomienia, monitorowanie i inne takie możliwości do konfiguracji.

## Kto używa Elastic Stack?

W ciągu ostatnich kilku lat wdrożenia Elastic Stack rozwijały się bardzo szybko. W tym rozdziale rozważymy kilka studiów przypadków, aby zrozumieć, w jaki sposób Elastic Stack przyczynił się do tego rozwoju.

## Salesforce

Opracowali nowy plugin o nazwie ELF (Event Log Files), który zbiera dane z dzienników Salesforce w celu umożliwienia kontroli aktywności użytkowników. Celem było przeanalizowanie danych, aby zrozumieć zachowania użytkowników i trendy w usługach Salesforce.

Plugin ten upraszcza konfigurację stosu i pozwala na pobranie ELF w celu uzyskania indeksowanych i wreszcie sensownych danych, które mogą być wizualizowane za pomocą Kibana. Ta implementacja wykorzystuje Elasticsearch, Logstash i Kibana.

## CERN

Jest nie tylko jeden przypadek zastosowania Elastic Stack, który pomógł CERN-owi (Europejskiej Organizacji Badań Jądrowych), ale pięć. W CERN Elastic Stack jest wykorzystywany do następujących celów:

* Messaging
* Monitorowanie danych
* Benchmarking w chmurze
* Monitorowanie infrastruktury
* Monitorowanie pracy

Wiele pulpitów Kibana jest wykorzystywanych przez CERN do wielu wizualizacji.

## Green Man Gaming

Jest to platforma gier online, na której dostawcy gier publikują swoje gry. Strona internetowa chciała dokonać zmiany poprzez udowodnienie lepszej rozgrywki. Zaczęli używać Elastic Stack do analizy logów, wyszukiwania i analizy danych dotyczących rozgrywki.

Zaczęli od założenia Kibana dashboardów, aby uzyskać wgląd w liczbę graczy według kraju i waluty używanej przez graczy. To pomogło im zrozumieć i usprawnić wsparcie i pomoc w celu zapewnienia lepszej reakcji.

Oprócz tych studiów przypadków, Elastic Stack jest używany przez wiele innych firm, aby uzyskać wgląd w dane, które posiadają. Czasami, nie wszystkie komponenty są używane, to znaczy, nie wszystkie razy użyto by Beata i skonfigurowano by Logstash. Czasami używana jest tylko kombinacja Elasticsearch i Kibana.

Jeśli przyjrzeć się użytkownikom w organizacji, wszystkie tytuły, od których oczekuje się dużej analizy danych, inteligencji biznesowej, wizualizacji danych, analizy dzienników itp. mogą wykorzystać Elastic Stack dla swojej technicznej przewagi. Kilka z tych tytułów to data scientists, DevOps i tak dalej.

## Konkurenci Elastic Stack

Cóż, błędem byłoby nie napisanie o bezpośrednich konkurentach Elastic Stack, ponieważ Elastic Stack stał się w ostatnich latach silną konkurencją dla wielu innych narzędzi na rynku i szybko rośnie.

Open Source

* Graylog
* InfluxDB

Pozostałe

* Logscape
* Logscene
* Splunk
* Sumo Logic

Konkurencja

Kibany

* Grafana
* Graphite

Elastic

* Lucene/Solr
* Sphinx

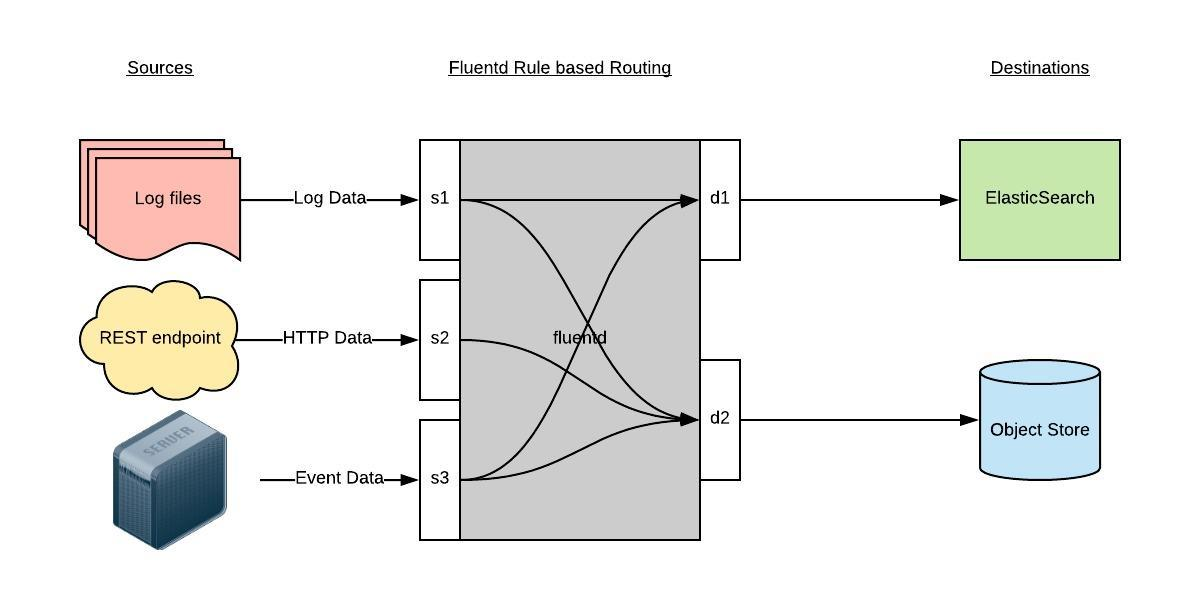
Większość z nich porównuje się z zarządzaniem logami, podczas gdy Elastic Stack to coś znacznie więcej. Oferuje on możliwość analizowania każdego rodzaju danych, a nie tylko logów.

# Rozdział 3

FluentD to wydajny agregator logów. Jest napisany w Ruby i bardzo dobrze się. W przypadku większości małych i średnich wdrożeń FluentD jest szybki i zużywa stosunkowo minimalne zasoby. W projekcie FluentD-Bit zostało poprawione skalowanie i zmniejszone zużycie zasobów. W mojej pracy przeanalizuję FluentD, ponieważ jest to rozwiązanie dosyć dobrze dopracowane i szeroko stosowane.

FluentD usuwa dzienniki z danego zestawu źródeł, przetwarza je (konwertuje do formatu danych strukturalnych), a następnie przekazuje je do usług, które przechowują obiekty np. Elasticsearch, FluentD integruje się z ponad 300 serwisami oferującymi magazynowanie i analizę logów.

* FluentD pobiera dane z wielu źródeł.
* Strukturyzuje i taguje dane.
* Następnie wysyła dane do wielu miejsc docelowych na podstawie pasujących tagów.



Rysunek 1 https://platform9.com/wp-content/uploads/2018/09/fluentd-architecture.jpg

## Logi lub wskaźniki (Logs or Metrics) - decyzja pojęciowa

Istnieją nieskończone sugestie, najlepsze praktyki i wskazówki, jak utrzymać stabilność środowisk produkcyjnych i zapobiegać przestojom usług. Zawsze będą problemy, które muszą zostać wykryte wcześnie, prawidłowo i szybko rozwiązane oraz wyciągnięte wnioski na przyszłość. Aby osiągnąć te cele, środowiska produkcyjne jest ściśle monitorowa każde wydarzenie rejestrowane i badane.  
  
Infrastruktura nawet podstawowej aplikacji w chmurze składa się z wielu możliwych punktów awarii - potencjalnie obejmujących usługi, kontenery, interfejsy użytkownika i integracje.

Obecnie monitorowanie oprogramowania odbywa się zazwyczaj za pomocą jednej z trzech metod: dzienników, metryk lub kombinacji obu tych metod. Metody te pomagają w gromadzeniu i przetwarzaniu danych produkcyjnych, ale słaba implementacja może powodować chaos, zniekształcać istotne informacje i utrudniać rozwiązywanie problemów.

## Dlaczego metryki

Nawet prosta aplikacja w chmurze opiera się na kilku komponentach, z których wszystkie są wdrażane w środowisku, które zespołom DevOps bardzo trudno kontrolować. Jeśli jeden z tych komponentów nie działa zgodnie z oczekiwaniami, cała aplikacja może być zagrożona. Metryki pomagają mierzyć funkcjonalność komponentu i definiować progi użycia wymagającego uwagi. Dane dają inżynierom DevOps możliwość oceny wartości usługi w czasie i zapewniają ciągły obraz całego środowiska. Istnieje nieskończona liczba metryk, których można użyć do oceny aplikacji, dlatego ważne jest, aby określić funkcjonalność krytyczną dla firmy i odpowiednio zbudować plan metryk.

Podstawowe wskaźniki, takie jak przepustowość transakcji i czas odpowiedzi, mają zastosowanie do wszystkich aplikacji, a liczba kliknięć na sekundę lub nowi użytkownicy miesięcznie są wykorzystywani do bardziej wyrafinowanych przypadków użycia. Dane są nie tylko istotne dla kodu, ale mogą być również stosowane do kontenerów hostujących usługi. Wskaźniki, takie jak zadania / zużycie / pamięć i przepustowość sieci, pomagają zespołom DevOps zrozumieć prędkość i wydajność systemu oraz określić poziom gotowości do gwałtownych skoków lub ciągłego obciążenia. W przypadku aplikacji bezserwerowych wskaźniki są absolutnie kluczowe - czas uruchamiania kontenera, czas odpowiedzi i średni czas wykonania kontenera odzwierciedlają użycie aplikacji i zdolność platformy do zaspokojenia potrzeb aplikacji. Metryki są stosunkowo łatwe do wdrożenia, ale po wdrożeniu mogą stanowić wyzwanie skalowania w miarę wzrostu ilości danych i wymaganej infrastruktury. Istnieje jednak kilka narzędzi, które mogą monitorować usługi w chmurze, a zebrane informacje są wykorzystywane do pomiarów. Gdy ładowanie usług wymaga skalowania, te narzędzia monitorowania wiedzą, jak zbierać te same dane dla nowych instancji, metryki automatycznie zawierają nowe dane i wymagają zerowej interwencji manualnej. Korzystanie z metryk ma swoje wady. Aby uzyskać dane dla każdej metryki, należy wygenerować zdarzenie dla każdego wystąpienia mierzonych działań. Projektowanie i wdrażanie tych zdarzeń jest dodatkowym zadaniem w każdym zadaniu programistycznym, należy również wziąć pod uwagę narzut usługi - w tym zużycie pamięci i czas przestoju usługi.

Ponadto, jak sugerowano powyżej, metryki są łatwe do tworzenia i przechowywania, więc niedoświadczone zespoły mogą popełnić błąd, tworząc zbyt wiele i mogą nie być w stanie wybrać odpowiednich metryk. Pomiary są przydatne do identyfikowania trendów, powiązania zachowania aplikacji z grupami zdarzeń i przewidywania wad systemu - działania, które pomagają uniknąć problemów z klientem, szczególnie w zakresie wydajności.

## Dlaczego logi

Wskaźniki mają kluczowe znaczenie dla przeglądu zachowania oprogramowania wdrażanego w chmurze i wpływają na decyzje dotyczące usprawnienia procesów wdrażania i konserwacji.

Jednak wielu programistów uważa dane za niewystarczające, a czasem nawet nieużyteczne. Podczas gdy metryki pokazują tendencje i skłonności usługi lub aplikacji, dzienniki koncentrują się na konkretnych zdarzeniach. Celem dzienników jest zachowanie jak największej ilości informacji - głównie technicznych - na temat konkretnego zdarzenia. Informacje w dziennikach można wykorzystać do badania incydentów i pomocy w analizie pierwotnych przyczyn błędów lub defektów, ale także w przypadku rosnącej liczby dodatkowych przypadków użycia.

Innym aspektem, w którym metryki różnią się od dzienników, jest to, że dzienniki mogą być unikalne w każdym zespole badawczo-rozwojowym (na przykład dzienniki aplikacji) i mają strukturę zgodną z potrzebami zespołu dochodzeniowego lub systemu, który je gromadzi i analizuje. Dzienniki dotyczą niektórych innych aspektów monitorowania - identyfikowania prób naruszenia bezpieczeństwa i niewłaściwego wykorzystania funkcji aplikacji oraz prowadzenia dokumentacji na potrzeby zgodności z prawem. Ale dzienniki nie są łatwe w użyciu. Wymagają większego miejsca do przechowywania i mają bardziej skomplikowane procedury przetwarzania niż metryki. Niepoprawnie zaimplementowane zawierają dużą ilość bezużytecznych danych, które ukrywają fragmenty informacji rzeczywiście potrzebne do procesu analizy.

Dane wyjściowe dziennika powinny być planowane i testowane, tak jak każda inna funkcjonalność aplikacji, aby w przypadku wypychania niezbędne informacje były dostępne, jasne i przydatne. Aby były skuteczne, dzienniki powinny spełniać określone standardy, takie jak wyświetlanie czytelnego dla człowieka języka i daty w przejrzystym formacie, podkreślanie błędów i kontekst dla każdego rekordu.

## Co ze śledzeniem?

Śledzenie to kolejny sposób śledzenia statusu środowiska, umożliwiający rejestrowanie na poziomie programisty. Gdy dzienniki są skonfigurowane do śledzenia poziomu, cała komunikacja, zdarzenia i dane są rejestrowane, tworząc wiele różnych typów rekordów. Większość z nich nie jest zlokalizowana, co oznacza, że nie są czytelne, a niektóre mogą nawet ujawnić poufne informacje.

Istnieje kilka podejść, że śledzenie jest właściwym sposobem rejestrowania wszystkich działań w ekosystemie, ale tylko wtedy, gdy jest to zrobione prawidłowo. Jeśli śledzenie nie jest zgodne z wyraźnym zestawem reguł, ogromna ilość danych w dziennikach przesłania ważne dane i wymaga głębszego zbadania w celu zebrania odpowiednich informacji. Niepoprawna implementacja śledzenia może również wpływać na wydajność systemu, ponieważ w dziennikach rejestrowana jest ogromna ilość danych, a każda akcja jest dokładnie dokumentowana. Śledzenie jest zalecane tylko dla zaawansowanych użytkowników, którzy naprawdę potrzebują danych niskiego poziomu.

## Jakiej metody powinienem zastosować?

Jak wyjaśniłem powyżej, metryki i dzienniki odpowiadają dwóm różnym potrzebom aplikacji w chmurze i oba mają kluczowe znaczenie dla firmy. Metryki można używać do monitorowania wydajności, rozpoznawania ważnych wydarzeń i ułatwiania przewidywania przyszłych przerw. Dzienniki są zwykle używane do rozwiązywania problemów, ale także do analizy zachowania użytkowników, wskaźników aplikacji i rosnącej liczby dodatkowych przypadków użycia.

Pomiary pomagają wskazać punkty usprawnienia procesów i umożliwiają widok aplikacji z lotu ptaka. Dzienniki są szczególnie przydatne, gdy stają się praktyczne - jeśli aplikacja napotyka wiele problemów funkcjonalnych i stale wymaga dogłębnego zbadania.

Dobrą wiadomością jest to, że zespoły DevOps niekoniecznie muszą wybierać jedną metodę zamiast drugiej. Dzienniki i metryki mogą być używane w tandemie. Zła wiadomość jest taka, że opanowanie obu metod monitorowania wymaga obsługi ogromnej ilości danych oraz możliwości odfiltrowania nieistotnych informacji i skupienia się na tym, co jest istotne i istotne dla utrzymania aplikacji.  
  
Istnieje kilka narzędzi zaprojektowanych w celu rozwiązania tych problemów, nadzorowania procesu monitorowania i wydobywania istotnych danych. Narzędzia te wdrażają różne mechanizmy gromadzenia, analizy i wyświetlania danych w sposób, który pomaga badać problemy i rozumieć konsekwencje.

Odgrywanie tak istotnej roli w świecie Big Data, logi i metryki wymagają głębokiego rozwiązania w zakresie wydobywania informacji i analizy segmentów, aby zespoły R&D i DevOps nie musiały same filtrować danych. ELK Stack robi dokładnie to, pomagając przechowywać i analizować duże zbiory danych, a następnie wyszukiwać i wyświetlać trendy oraz informacje o aplikacji i jej zasobach.

# Rozdział 4

Stos ELK, który tradycyjnie składał się z trzech głównych elementów - Elasticsearch, Logstash i Kibana, od dawna odszedł od tej kompozycji i może być teraz używany w połączeniu z czwartym elementem zwanym „Beats” - rodziną dostawców kłód do różnych zastosowań . To odejście doprowadziło do zmiany nazwy stosu na Elastyczny. Ta część pracy zawiera informacje dotyczące rozpoczęcia pracy z różnymi Beats - Filebeat, Packetbeat, Metricbeat, Auditbeat, Heartbeat, Winlogbeat i Functionbeat.

## Krótka lekcja historii

W scentralizowanym logowaniu potok danych składa się z trzech głównych etapów: agregacji, przetwarzania i przechowywania. W stosie ELK za pierwsze dwa etapy tradycyjnie odpowiedzialny był Logstash, koń roboczy stosu. Wykonywanie tych zadań kosztuje. Ze względu na nieodłączne problemy związane ze sposobem zaprojektowania Logstash często występowały problemy z wydajnością, szczególnie w przypadku skomplikowanych rurociągów, które wymagają dużej ilości przetwarzania.  
Pojawił się także pomysł outsourcingu części obowiązków Logstash, w szczególności ekstrakcji danych, do innych narzędzi. Jak opisałem w tym artykule, pomysł pojawił się najpierw w Lumberjack, a później w forwarderze Logstash. W końcu i kilka cykli rozwoju później wprowadzono nowy i ulepszony protokół, który stał się podstawą tego, co obecnie nazywa się rodziną „Beats”.

## Co to są Beats?

Beaty to zbiór lekkich (zasobooszczędnych, bez zależności, małych) i dostarczających logi typu open source, które działają jako agenty zainstalowane na różnych serwerach w infrastrukturze w celu gromadzenia logów lub metryk. Mogą to być pliki dziennika (Filebeat), dane sieciowe (Packetbeat), metryki serwera (Metricbeat) lub dowolny inny rodzaj danych, które mogą być gromadzone przez rosnącą liczbę bitów opracowywanych zarówno przez Elastic, jak i przez społeczność. Po zebraniu dane są wysyłane bezpośrednio do Elasticsearch lub do Logstash w celu dodatkowego przetworzenia.  
Beaty są oparte na frameworku Go o nazwie libbeat - bibliotece służącej do przekazywania danych, co oznacza, że nowe beaty są opracowywane i wnoszone przez społeczność cały czas.

## Filebeat

Jak sama nazwa wskazuje, służy do zbierania i wysyłania plików dziennika, a także jest najczęściej używanym rytmem. Jednym z faktów, które sprawiają, że Filebeat jest tak wydajny, jest sposób radzenia sobie z ciśnieniem wstecznym - więc jeśli Logstash jest zajęty, Filebeat spowalnia jego szybkość odczytu i przyspiesza rytm po zakończeniu spowolnienia.

Filebeat może być zainstalowany w prawie każdym systemie operacyjnym, w tym jako kontener Docker, a także jest wyposażony w wewnętrzne moduły dla określonych platform, takich jak Apache, MySQL, Docker i inne, zawierające domyślne konfiguracje i obiekty Kibana dla tych platform.

## Packetbeat

Analizator pakietów sieciowych Packetbeat był pierwszym wprowadzonym rytmem. Packetbeat przechwytuje ruch sieciowy między serwerami i jako taki może być wykorzystywany do monitorowania aplikacji i wydajności. Packetbeat można zainstalować na monitorowanym serwerze lub na własnym serwerze dedykowanym. Packetbeat śledzi ruch w sieci, dekoduje protokoły i rejestruje dane dla każdej transakcji. Protokoły obsługiwane przez Packetbeat obejmują: DNS, HTTP, ICMP, Redis, MySQL, MongoDB, Cassandra i wiele innych.

Packetbeat to program do przesyłania i analizowania danych o otwartym kodzie źródłowym dla pakietów sieciowych zintegrowanych ze stosem ELK (Elasticsearch, Logstash i Kibana). Packetbeat, członek rodziny dostawców logów Elastic (Filebeat, Topbeat, Libbeat, Winlogbeat), dostarcza metryki monitorowania w czasie rzeczywistym w Internecie, bazie danych i innych protokołach sieciowych poprzez monitorowanie rzeczywistych pakietów przesyłanych przewodowo.

Monitorowanie pakietów danych za pomocą ELK Stack może pomóc w wykrywaniu nietypowych poziomów ruchu sieciowego i nietypowych charakterystyk pakietów, identyfikowaniu źródeł i miejsc docelowych pakietów, wyszukiwaniu określonych ciągów danych w pakietach i tworzeniu przyjaznego dla użytkownika pulpitu nawigacyjnego z wnikliwymi statystykami. Monitorowanie pakietów może uzupełniać inne środki bezpieczeństwa (takie jak tworzenie pulpitów menedżerskich SIEM) i pomóc skrócić czas reakcji na złośliwe ataki.

## Metricbeat

Niezwykle popularny beat, który zbiera i raportuje różne wskaźniki na poziomie systemu dla różnych systemów i platform. Metricbeat obsługuje również wewnętrzne moduły do zbierania statystyk z określonych platform. Możesz skonfigurować częstotliwość, z jaką Metricbeat zbiera metryki, oraz jakie konkretne metryki należy zbierać, korzystając z tych modułów i ustawień podrzędnych zwanych zestawami metryk.

## Heartbeat

Przeznaczone do „monitorowania czasu pracy”. W skrócie, Heartbeat robi usługi sprawdzania, czy są one osiągalne, czy nie - przydatne jest na przykład sprawdzenie, czy czas działania usługi jest zgodny z umową SLA. Wszystko, co musisz zrobić, to dostarczyć Heartbeat listę adresów URL i metryki czasu działania, aby wysłać je na stos, bezpośrednio do Elasticsearch lub Logstash w celu wzbogacenia przed indeksowaniem.

## Auditbeat

Może służyć do kontrolowania aktywności użytkowników i procesów na serwerach Linux. Podobnie jak inne tradycyjne narzędzia do inspekcji systemu (systemd, auditd), Auditbeat może być wykorzystywany do identyfikowania naruszeń bezpieczeństwa - zmian plików, zmian konfiguracji, złośliwego zachowania itp.

Monitorowanie tego, co dzieje się w systemie, jest kluczem do jego ochrony. Zbudowano szereg narzędzi lub demonów, takich jak systemd, icrond i auditd, aby pomóc użytkownikom Linuxa śledzić zmiany plików, a także monitorować i uzyskiwać dostęp do procesów uruchomionych w systemie.  
  
Ponieważ auditd wykonuje kontrolę na poziomie jądra i poniżej przestrzeni użytkownika, możesz podłączyć się do dowolnej operacji systemu, którą chcesz, co znacznie utrudnia penetrację systemu. Dlatego w przypadku zastosowań bezpieczeństwa oraz ze względu na łatwość użycia i wszechstronność, to narzędzie zyskało popularność. Dla użytkowników ELK moduł Audit został dodany do Metricbeat jako alternatywa dla korzystania z audytu dostarczonego przez dystrybucję, ale to rozwiązanie było nieco ograniczone. W programie Elastic Stack 6.0 (beta) nowy użytkownik dziennika kontroli o nazwie Auditbeat jest teraz dostępny dla użytkowników ELK.

## Winlogbeat

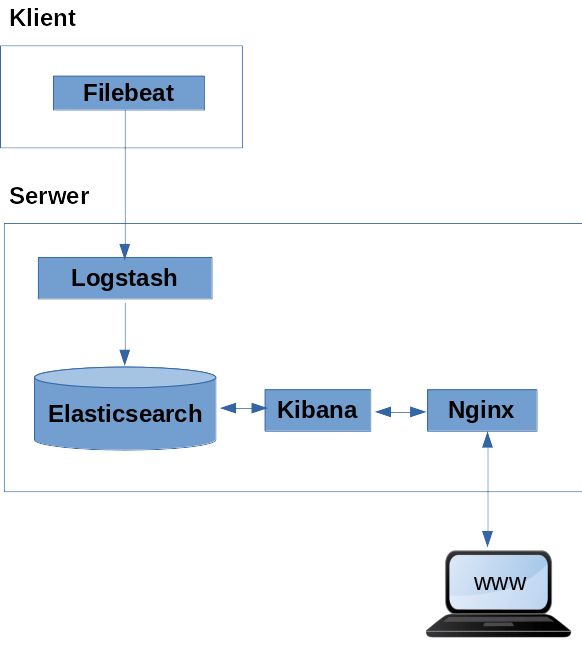
Będzie interesował tylko administratorów lub inżynierów Windows, ponieważ jest to rytm zaprojektowany specjalnie do zbierania dzienników zdarzeń Windows. Można go wykorzystać do analizy zdarzeń bezpieczeństwa, zainstalowanych aktualizacji i tak dalej.

Dzienniki zdarzeń Windows zawierają wiele informacji o środowiskach Windows i są wykorzystywane do wielu celów. Administratorzy systemu i menedżerowie IT mogą używać dzienników zdarzeń do monitorowania aktywności w sieci i zachowania aplikacji. Dzienniki zdarzeń mogą być niezwykle cennymi zasobami do wykrywania incydentów bezpieczeństwa przez zespoły InfoSec, gdy zapewniają zgodność z wymogami regulacyjnymi, takimi jak SOX, HIPAA i PCI.

Chociaż konfiguracja i przeglądanie informacji zapisanych w systemach Windows jest dość łatwa, nie jest tak łatwo odpowiednio wyszukiwać i analizować te informacje z powodu dużej ilości danych, które zwykle są w to zaangażowane. Właśnie tam może się przydać stos ELK. Jest przeznaczony do przetwarzania dużych zbiorów danych i szybkiej analizy komunikatów dziennika. Możliwość korzystania z wizualizacji i pulpitów nawigacyjnych Kibana to ogromna zaleta i kolejny powód, dla którego ELK stał się preferowaną bronią z wyboru podczas logowania do systemu Windows.

# Rozdział 5

## Instalacja stosu ELK na CentOS 7



Rysunek 2

* Elastic Search – Gromadzenie logówWymagana JAVA 8 lub nowszaMetody instalacji:
  + Tar ball (.zip/tar.gz)
  + Ręcznie RPM
  + Przy użyciu repozytorium Yum
* Porty
  + 9200 (REST)
  + 9300 (Komunikacja węzłowa / Node communication)
* Plik konfiguracyjny: /etc/elasticsearch/elasticsearch.yml
* Folder plików logów: /var/log/elasticsearch
* Tweak to enable Journal logging

Dostępnych jest kilka metod instalacji jednakże najbardziej wygodnym i preferowanym jest użycie repozytorium, ponieważ dzięki temu pobieramy i instalujemy najnowszą wersję oprogramowania. W teorii jest to wersja najlepsza na dany moment w której na bieżąco nieeliminowane są błędy. Czasami zdarzają się wyjątki, że najnowsza wersja nie jest kompatybilna z jednym z pozostałych komponentów jednakże są to sytuacje sporadyczne. Instalacja oprogramowania z repozytorium pozwala również na płynne aktualizowanie software w przyszłości, gdy pojawi się taka potrzeba.

Yml jest plikiem konfiguracyjnym zawierającym język znaczników. AML (rekursywny akronim „YAML Ain't Markup Language”) to czytelny dla człowieka język serializacji danych. Jest powszechnie używany do plików konfiguracyjnych i aplikacji, w których dane są przechowywane lub przesyłane. YAML jest ukierunkowany na wiele takich samych aplikacji komunikacyjnych jak Extensible Markup Language (XML), ale ma minimalną składnię, która celowo różni się od SGML. Używa zarówno wcięcia w stylu Python, aby wskazać zagnieżdżenie, jak i bardziej kompaktowego formatu, który używa [] do listy i {} dla map czyniących YAML 1.2 nadzbiorem JSON.

**Kibana – Wizualizacja logów**

* Metody instalacji:
  + Tar ball (.zip/tar.gz)
  + Ręcznie RPM
  + Przy użyciu repozytorium Yum
* Porty
  + 5601 (localhost)
  + użycie Nginx jako reverse (zwrotny serwer) proxy
* Plik konfiguracyjny: /etc/kibana/kibana.yml
* Folder plików logów: uruchomienie Journal (journalctl -f –unit=kibana)

**Nginx – Reverse (zwrotny serwer) Proxy**

* /etc/nginx/nginx.conf
  + Usunięcie bloku domyślnego serwera
* *etc*/nginx/conf.d/kibana.conf

server {

listen 80;

server\_name server.example.com;

location / {

proxy\_pass http://localhost:5601;

}

**Logstash – Przechwytywanie, przetwarzanie i indeksacja logów**

* Porty
  + 5044 (localhost)

**Filebeat – Zbieranie i przesyłanie danych**

* Metody instalacji:
  + Tar ball (.zip/tar.gz)
  + Ręcznie RPM
  + Przy użyciu repozytorium Yum
* Plik konfiguracyjny: /etc/filebeat/filebeat.yml
* Folder plików logów:
  + /var/log/filebeat
  + uruchomienie Journal (journalctl -f –unit=filebeat)

Instrukcje konfiguracyjne pobrałem z repozytorium github.com/JustMeAndOpenSource, które były opracowane na podstawie ze strony elastic.co/start.

Tworzę dwie maszyny wirtualne przy użyciu oprogramowania VirtualBox. Do automatycznego utworzenia i początkowej konfiguracji sieciowej używam aplikacji Vagrant. Jest fantastyczne narzędzie do automatyzacji pracy administratora. Poniżej prezentuję plik Vagrantfile.

# vi: set ft=ruby :

ENV['VAGRANT\_NO\_PARALLEL'] = 'yes'

Vagrant.configure(2) do |config|

config.vm.provision "shell", path: "bootstrap.sh"

config.vm.define "server" do |server|

server.vm.box = "centos/7"

server.vm.hostname = "server.example.com"

server.vm.network "private\_network", ip: "172.42.42.10"

server.vm.provider "virtualbox" do |vb|

vb.name = "server"

vb.memory = 1024

vb.cpus = 1

end

end

config.vm.define "client" do |client|

client.vm.box = "centos/7"

client.vm.hostname = "client.example.com"

client.vm.network "private\_network", ip: "172.42.42.20"

client.vm.provider "virtualbox" do |vb|

vb.name = "client"

vb.memory = 512

vb.cpus = 1

end

end

end

Zostaną utworzone dwie maszyny CentOS 7. Jedna będzie serwerem a druga klientem. Będą miały na stałe przypisane adresy IP: 172.42.42.10 oraz 172.42.42.20. Serwer będzie posiadał 1 GB pamięci RAM, a klient 512 MB. Obydwa serwery będą miały przypisane po jednym procesorze. VM Boxy (pliki instalacyjne) pobierane są bezpośrednio z repozytoriów Vagrant. Podczas instalacji na jej końcowym etapie zostanie wywołany plik bootstrap.sh. Poniżej prezentuje jego zawartość.

#!/bin/bash

# Update the system

echo "[TASK 1] Updating the system"

yum update -y >/dev/null 2>&1

# Install desired packages

echo "[TASK 2] Installing desired packages"

yum install -y -q vim redhat-lsb-core net-tools bind-utils >/dev/null 2>

# Set up global aliases and exports

echo "[TASK 3] Creating global aliases and functions"

cat >>/etc/bashrc <<EOF

# Generated by Vagrant

alias vi='vim'

alias sudo='sudo '

export EDITOR=vim

export TERM=xterm

# Generated by Vagrant

EOF

echo "colorscheme elflord" >> /etc/vimrc

# Enable password authentication

echo "[TASK 4] Enabling password authentication in sshd config"

sed -i 's/^PasswordAuthentication no/PasswordAuthentication yes/' /etc/>

systemctl reload sshd

# Disable SELinux

echo "[TASK 5] Disable SELinux"

setenforce 0

sed -i --follow-symlinks 's/^SELINUX=enforcing/SELINUX=disabled/' /etc/>

# Set Root password

echo "[TASK 6] Set root password"

echo "admin" | passwd --stdin root >/dev/null 2>&1

# Disable and stop firewalld

echo "[TASK 5] Disable and stop firewalld"

systemctl disable firewalld >/dev/null 2>&1

systemctl stop firewalld

# Update hosts file

echo "[TASK 6] Update /etc/hosts file"

cat >>/etc/hosts<<EOF

172.42.42.10 server.example.com server

172.42.42.20 client.example.com client

EOF

W zadaniu 1 (task 1) aktualizowany jest system. Następnie w zadaniu 2 instalowane są wymagane pakiety. Zadanie 3 aktualizuje globalne aliasy i funkcje. Zadanie 4 uruchamia uwierzytelnianie hasłem protokołu zdalnego zarządzania SSH. Domyślnie podczas stawiania maszyny wirtualnej przy użyciu Vagrant uwierzytelnianie hasłem nie jest uruchomione. To ustawienie zostało wprowadzone aby było możliwe ustanowienie zdalnego połączenia z prawami administratora root. Krok 5 dezaktywuje SELinux oraz linuksowego firewalla (zaporę sieciową). Oczywiście możliwe jest otworzenie wybranych portów aby umożliwić komunikację ELK między klientem a serwerem. Jednak do zastosowań testowych najprościej było po prostu zdezaktywować zaporę sieciową. To rozwiązanie na produkcji jest wysoce niezalecane ponieważ infrastruktura otwarta jest na wszelkie zagrożenia. Ostatnim krokiem jest edycja pliku etc/host aby zamiast adresów logicznych IP można było używać nazw domenowych.

**[szymon@LinuxManjaro vagrant-provisioning]$** *ls*

bootstrap.sh Vagrantfile

**[szymon@LinuxManjaro vagrant-provisioning]$** *vagrant up*

Bringing machine 'server' up with 'virtualbox' provider...   
Bringing machine 'client' up with 'virtualbox' provider…

==> server: Importing base box 'centos/7'...   
==> server: Matching MAC address for NAT networking…

==> server: Setting the name of the VM: server   
==> server: Clearing any previously set network interfaces...   
==> server: Preparing network interfaces based on configuration…

   server: Adapter 1: nat Adapter 2: hostonly   
==> server: Forwarding ports… 22 (guest) => 2222 (host) (adapter 1)   
==> server: Running 'pre-boot' VM customizations...   
==> server: Booting VM...   
==> server: Waiting for machine to boot. This may take a few minutes...   
   server: SSH address: 127.0.0.1:2222 SSH username: vagrant   
   server: SSH auth method: private key

server: Vagrant insecure key detected. Vagrant will automatically replace

server: this with a newly generated keypair for better security.

server: Inserting generated public key within guest...   
 server: Removing insecure key from the guest if it's present...   
 server: Key inserted! Disconnecting and reconnecting using new SSH key…

==> server: Machine booted and ready!   
==> server: Checking for guest additions in VM...   
   server: No guest additions were detected on the base box for this VM! Guest additions are required for forwarded ports, shared folders, host only networking, and more. If SSH fails on this machine, please install the guest additions and repackage the box to continue.

==> server: Setting hostname...   
==> server: Configuring and enabling network interfaces...   
==> server: Rsyncing folder: /home/szymon/play/elk/vagrant-provisioning/vagrant   
==> server: Running provisioner: shell…

   server: [TASK 1] Updating the system   
   server: [TASK 2] Installing desired packages   
   server: [TASK 3] Creating global aliases and functions   
   server: [TASK 4] Enabling password authentication in sshd config   
   server: [TASK 5] Disable SELinux   
   server: [TASK 6] Set root password   
   server: [TASK 7] Disable and stop firewalld   
   server: [TASK 8] Update /etc/hosts file

Edycja pliku /ect/hosts na komputerze użytym do hostowania maszyn wirtualnych.

127.0.0.1 localhost

127.0.1.1 LinuxManjaro

::1 localhost ip6-localhost ip6-loopback

ff02::1 ip6-allnodes

ff02::2 ip6-allrouters

172.42.42.10 server.example.com server

172.42.42.20 client.example.com client

Teraz mogę zalogować się do maszyn wirtualnych.

**[szymon@LinuxManjaro** **vagrant-provisioning]$** *ssh root@server*

The authenticity of host 'server (172.42.42.10)' can't be established.   
ECDSA key fingerprint is SHA256:bpcGW71ifmDeU3McxojVed28gnrkr3aT0Nf81o142RY.   
Are you sure you want to continue connecting (yes/no/[fingerprint])? **Yes**

Warning: Permanently added 'server,172.42.42.10' (ECDSA) to the list of known hosts.

root@server's password: **admin**

Po zatwierdzeniu certyfikatu oraz podaniu hasła roota poprawnie zalogowałem się do CentOS 7, który pełni rolę serwera. Sprawdzam wersję Linux oraz czy SELinux jest dezaktywowany.

**[root@server ~]#** *lsb\_release -dirc*   
Distributor ID: CentOS   
Description:    CentOS Linux release 7.7.1908 (Core)   
Release:        7.7.1908   
Codename:       Core

**[root@server ~]#** *sestatus*   
SELinux status:                 enabled   
SELinuxfs mount:                /sys/fs/selinux   
SELinux root directory:         /etc/selinux   
Loaded policy name:             targeted   
Current mode:                   permissive   
Mode from config file:          disabled   
Policy MLS status:              enabled   
Policy deny\_unknown status:     allowed   
Max kernel policy version:      31

Sprawdzam również status daemona zapory sieciowej.

**[root@server ~]#** *sudo systemctl status firewalld*   
● firewalld.service - firewalld - dynamic firewall daemon   
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled;   
vendor preset: enabled)   
  Active: inactive (dead)   
    Docs: man:firewalld(1)

Następnie zalogowałem się do maszyny wirtualnej client. Powtórzyłem tę samą procedurę i analogicznie sprawdziłem, czy odpowiednie usługi są również dezaktywowane. Na tym etapie jestem gotowy do dalszej pracy.

Na maszynie wirtualnej serwera rozpoczynam instalowanie wymaganych zależności.

**[root@server ~]#** *yum install java-1.8.0-openjdk*  
Loaded plugins: fastestmirror   
Transaction ID : 4   
Begin time     : Thu Feb 20 22:07:17 2020   
Begin rpmdb    : 369:97cc1fa63ce763b3c23ab5ecd4f5bc7f578e0d8c   
End time       :            22:07:31 2020 (14 seconds)   
End rpmdb      : 432:21e417d11c05602f835642730f94bef259c19dee   
User           : root <root>   
Return-Code    : Success   
Command Line   : install java-1.8.0-openjdk   
Transaction performed with:   
   Installed     rpm-4.11.3-40.el7.x86\_64                      @base   
   Installed     yum-3.4.3-163.el7.centos.noarch               @base   
   Installed     yum-plugin-fastestmirror-1.1.31-52.el7.noarch @base   
Packages Altered:   
   Dep-Install alsa-lib-1.1.8-1.el7.x86\_64                                @base   
   Dep-Install atk-2.28.1-1.el7.x86\_64                                    @base   
   Dep-Install cairo-1.15.12-4.el7.x86\_64                                 @base   
   Dep-Install copy-jdk-configs-3.3-10.el7\_5.noarch                       @base   
   Dep-Install dejavu-fonts-common-2.33-6.el7.noarch                      @base   
   Dep-Install dejavu-sans-fonts-2.33-6.el7.noarch                        @base   
   Dep-Install fontconfig-2.13.0-4.3.el7.x86\_64                           @base   
   Dep-Install fontpackages-filesystem-1.44-8.el7.noarch                  @base   
   Dep-Install fribidi-1.0.2-1.el7\_7.1.x86\_64                             @updates  
   Dep-Install gdk-pixbuf2-2.36.12-3.el7.x86\_64                           @base   
   Dep-Install giflib-4.1.6-9.el7.x86\_64                                  @base   
   Dep-Install graphite2-1.3.10-1.el7\_3.x86\_64                            @base   
   Dep-Install gtk-update-icon-cache-3.22.30-3.el7.x86\_64                 @base   
   Dep-Install gtk2-2.24.31-1.el7.x86\_64                                  @base   
   Dep-Install harfbuzz-1.7.5-2.el7.x86\_64                                @base   
   Dep-Install hicolor-icon-theme-0.12-7.el7.noarch                       @base   
   Dep-Install jasper-libs-1.900.1-33.el7.x86\_64                          @base   
   Install     java-1.8.0-openjdk-1:1.8.0.242.b08-0.el7\_7.x86\_64          @updates  
   Dep-Install java-1.8.0-openjdk-headless-1:1.8.0.242.b08-0.el7\_7.x86\_64 @updates  
   Dep-Install javapackages-tools-3.4.1-11.el7.noarch                     @base   
   Dep-Install jbigkit-libs-2.0-11.el7.x86\_64                             @base   
   Dep-Install libICE-1.0.9-9.el7.x86\_64                                  @base   
   Dep-Install libSM-1.2.2-2.el7.x86\_64                                   @base   
   Dep-Install libX11-1.6.7-2.el7.x86\_64                                  @base   
   Dep-Install libX11-common-1.6.7-2.el7.noarch                           @base   
   Dep-Install libXau-1.0.8-2.1.el7.x86\_64                                @base   
   Dep-Install libXcomposite-0.4.4-4.1.el7.x86\_64                         @base   
   Dep-Install libXcursor-1.1.15-1.el7.x86\_64                             @base   
   Dep-Install libXdamage-1.1.4-4.1.el7.x86\_64                            @base   
   Dep-Install libXext-1.3.3-3.el7.x86\_64                                 @base   
   Dep-Install libXfixes-5.0.3-1.el7.x86\_64                               @base   
   Dep-Install libXft-2.3.2-2.el7.x86\_64                                  @base   
   Dep-Install libXi-1.7.9-1.el7.x86\_64                                   @base   
   Dep-Install libXinerama-1.1.3-2.1.el7.x86\_64                           @base   
   Dep-Install libXrandr-1.5.1-2.el7.x86\_64                               @base   
   Dep-Install libXrender-0.9.10-1.el7.x86\_64                             @base   
   Dep-Install libXtst-1.2.3-1.el7.x86\_64                                 @base   
   Dep-Install libXxf86vm-1.1.4-1.el7.x86\_64                              @base   
   Dep-Install libfontenc-1.1.3-3.el7.x86\_64                              @base   
   Dep-Install libglvnd-1:1.0.1-0.8.git5baa1e5.el7.x86\_64                 @base   
   Dep-Install libglvnd-egl-1:1.0.1-0.8.git5baa1e5.el7.x86\_64             @base   
   Dep-Install libglvnd-glx-1:1.0.1-0.8.git5baa1e5.el7.x86\_64             @base   
   Dep-Install libjpeg-turbo-1.2.90-8.el7.x86\_64                          @base   
   Dep-Install libthai-0.1.14-9.el7.x86\_64                                @base   
   Dep-Install libtiff-4.0.3-32.el7.x86\_64                                @base   
   Dep-Install libwayland-client-1.15.0-1.el7.x86\_64                      @base   
   Dep-Install libwayland-server-1.15.0-1.el7.x86\_64                      @base   
   Dep-Install libxcb-1.13-1.el7.x86\_64                                   @base   
   Dep-Install libxshmfence-1.2-1.el7.x86\_64                              @base   
   Dep-Install lksctp-tools-1.0.17-2.el7.x86\_64                           @base   
   Dep-Install mesa-libEGL-18.3.4-6.el7\_7.x86\_64                          @updates  
   Dep-Install mesa-libGL-18.3.4-6.el7\_7.x86\_64                           @updates  
   Dep-Install mesa-libgbm-18.3.4-6.el7\_7.x86\_64                          @updates  
   Dep-Install mesa-libglapi-18.3.4-6.el7\_7.x86\_64                        @updates  
   Dep-Install pango-1.42.4-4.el7\_7.x86\_64                                @updates  
   Dep-Install pcsc-lite-libs-1.8.8-8.el7.x86\_64                          @base   
   Dep-Install pixman-0.34.0-1.el7.x86\_64                                 @base   
   Dep-Install python-javapackages-3.4.1-11.el7.noarch                    @base   
   Dep-Install python-lxml-3.2.1-4.el7.x86\_64                             @base   
   Dep-Install ttmkfdir-3.0.9-42.el7.x86\_64                               @base   
   Dep-Install tzdata-java-2019c-1.el7.noarch                             @updates  
   Dep-Install xorg-x11-font-utils-1:7.5-21.el7.x86\_64                    @base   
   Dep-Install xorg-x11-fonts-Type1-7.5-9.el7.noarch                      @base

Kolejnym krokiem jest instalacja ElasticSearch.

Sprawdzam która wersja java została zainstalowana.

**[root@server ~]#** *java -version*    
openjdk version "1.8.0\_242"   
OpenJDK Runtime Environment (build 1.8.0\_242-b08)   
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)

Na początku należy zaimportować klucz. Następnie utworzyć repozytorium Yum.

**[root@server ~]#** *rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch*

**[root@server ~]#** *cat >>/etc/yum.repos.d/elk.repo<<EOF   
> [ELK-6.x]   
> name=ELK repository for 6.x packages   
> baseurl=https://artifacts.elastic.co/packages/6.x/yum   
> gpgcheck=1   
> gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch   
> enabled=1   
> autorefresh=1   
> type=rpm-md   
> EOF*

**[root@server ~]#** *yum install elasticsearch*   
Loaded plugins: fastestmirror   
Loading mirror speeds from cached hostfile   
Resolving Dependencies   
--> Running transaction check   
---> Package elasticsearch.noarch 0:6.8.6-1 will be installed   
--> Finished Dependency Resolution

Dependencies Resolved   
========================================================================   
Package              Arch          Version        Repository      Size   
========================================================================   
Installing:   
elasticsearch        noarch        6.8.6-1        ELK-6.x        142 M   
Transaction Summary   
========================================================================   
Install  1 Package   
  
Total download size: 142 M   
Installed size: 229 M   
Is this ok [y/d/N]: **y**

Downloading packages:   
elasticsearch-6.8.6.rpm                            | 142 MB   00:18        
Running transaction check   
Running transaction test   
Transaction test succeeded   
Running transaction   
Creating elasticsearch group... OK   
Creating elasticsearch user... OK   
 Installing : elasticsearch-6.8.6-1.noarch                         1/1    
### NOT starting on installation, please execute the following statements to configure elasticsearch service to start automatically using systemd   
sudo systemctl daemon-reload   
sudo systemctl enable elasticsearch.service   
### You can start elasticsearch service by executing   
sudo systemctl start elasticsearch.service   
Created elasticsearch keystore in /etc/elasticsearch   
 Verifying  : elasticsearch-6.8.6-1.noarch                         1/1    
  
Installed:   
 elasticsearch.noarch 0:6.8.6-1                                           
Complete!

Następnie aktywuję i uruchamiam usługę elasticsearch.

**[root@server ~]#** *systemctl daemon-reload*   
**[root@server ~]#** *systemctl enable elasticsearch*   
Created symlink from /etc/systemd/system/multi-user.target.wants/elastic  
search.service to /usr/lib/systemd/system/elasticsearch.service.   
**[root@server ~]#** *systemctl start elasticsearch*   
**[root@server ~]#** *systemctl status elasticsearch*   
● elasticsearch.service - Elasticsearch   
  Loaded: (/usr/lib/systemd/system/elasticsearch.service; enabled)  
  Active: **active (running)** since Thu 2020-02-20 22:38:40 UTC; 6s ago   
    Docs: http://www.elastic.co   
Main PID: 6388 (java)   
  CGroup: /system.slice/elasticsearch.service   
          ├─6388 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX...   
          └─6443 /usr/share/elasticsearch/modules/x-pack-ml/platform...

Na razie nie ma konieczności zmiany w plikach konfiguracyjny. Natomiast jeżeli pojawi się taka konieczność to za pomocą tej komendy można wylistować pliki konfiguracyjne, które dotyczą usługi elasticsearch.

**[root@server ~]#** *rpm -qc elasticsearch*   
/etc/elasticsearch/elasticsearch.yml   
/etc/elasticsearch/jvm.options   
/etc/elasticsearch/log4j2.properties   
/etc/elasticsearch/role\_mapping.yml   
/etc/elasticsearch/roles.yml   
/etc/elasticsearch/users   
/etc/elasticsearch/users\_roles   
/etc/init.d/elasticsearch   
/etc/sysconfig/elasticsearch   
/usr/lib/sysctl.d/elasticsearch.conf   
/usr/lib/systemd/system/elasticsearch.service

Za pomocą poniższego polecenia można sprawdzić lokalizację przechowywania logów oraz jakie domyślne logi zostały już zebrane.

**[root@server ~]#** *ls /var/log/elasticsearch*   
elasticsearch\_access.log   
elasticsearch\_audit.log   
elasticsearch\_deprecation.log   
elasticsearch\_index\_indexing\_slowlog.log   
elasticsearch\_index\_search\_slowlog.log   
elasticsearch.log   
gc.log.0.current

**[root@server ~]#** *netstat -ntlp*

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program

tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN 1/systemd

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN 6989/sshd

tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN 2591/master

tcp6 0 0 :::111 :::\* LISTEN 1/systemd

**tcp6 0 0 127.0.0.1:9200 :::\* LISTEN 6388/java**

tcp6 0 0 ::1:9200 :::\* LISTEN 6388/java

**tcp6 0 0 127.0.0.1:9300 :::\* LISTEN 6388/java**

tcp6 0 0 ::1:9300 :::\* LISTEN 6388/java

tcp6 0 0 :::22 :::\* LISTEN 6989/sshd

tcp6 0 0 ::1:25 :::\* LISTEN 2591/master

Usługa Elastic Search poprawnie wystartowała teraz mogę zabrać się za instalację usługi Kibana. Podejmuję analogiczne kroki. Powtarzające czynności będę pomijał w treści pracy. Udokumentowane zostaną różnice względem instalacji Elastic Search.

**[root@server ~]#** *rpm -qc kibana*  
/etc/kibana/kibana.yml

**[root@server ~]#** *netstat -ntlp*

tcp        0      0 127.0.0.1:5601    0.0.0.0:\*    LISTEN      6598/node  
  
Następnie instaluję usługę Nginx. Konieczne jest również doinstalowanie zależności.

**[root@server ~]#** *yum install -y epel-release*  
**[root@server ~]#** *yum install -y nginx*

**[root@server ~]#** *yum history info 9*   
Command Line   : install -y epel-release   
Transaction performed with:   
   Installed     rpm-4.11.3-40.el7.x86\_64                      @base   
   Installed     yum-3.4.3-163.el7.centos.noarch               @base   
   Installed     yum-metadata-parser-1.1.4-10.el7.x86\_64       @anaconda   
   Installed     yum-plugin-fastestmirror-1.1.31-52.el7.noarch @base   
Packages Altered:   
   Install epel-release-7-11.noarch @extras   
[root@server ~]# yum history info 10   
Command Line   : install -y nginx   
Transaction performed with:   
   Installed     rpm-4.11.3-40.el7.x86\_64                      @base   
   Installed     yum-3.4.3-163.el7.centos.noarch               @base   
   Installed     yum-metadata-parser-1.1.4-10.el7.x86\_64       @anaconda   
   Installed     yum-plugin-fastestmirror-1.1.31-52.el7.noarch @base   
Packages Altered:   
   Dep-Install centos-indexhtml-7-9.el7.centos.noarch            @base   
   Dep-Install centos-logos-70.0.6-3.el7.centos.noarch           @base   
   Dep-Install gd-2.0.35-26.el7.x86\_64                           @base   
   Dep-Install gperftools-libs-2.6.1-1.el7.x86\_64                @base   
   Dep-Install libXpm-3.5.12-1.el7.x86\_64                        @base   
   Install     nginx-1:1.16.1-1.el7.x86\_64                       @epel   
   Dep-Install nginx-all-modules-1:1.16.1-1.el7.noarch           @epel   
   Dep-Install nginx-filesystem-1:1.16.1-1.el7.noarch            @epel   
   Dep-Install nginx-mod-http-image-filter-1:1.16.1-1.el7.x86\_64 @epel   
   Dep-Install nginx-mod-http-perl-1:1.16.1-1.el7.x86\_64         @epel   
   Dep-Install nginx-mod-http-xslt-filter-1:1.16.1-1.el7.x86\_64  @epel   
   Dep-Install nginx-mod-mail-1:1.16.1-1.el7.x86\_64              @epel   
   Dep-Install nginx-mod-stream-1:1.16.1-1.el7.x86\_64            @epel  
  
Przygotowuję konfigurację serwera proxy. Usuwam domyślny blok konfiguracji serwera w pliku /etc/nginx/nginx.conf i umieszczam własną konfigurację serwera w oddzielnym pliku /etc/nginx/conf.d/kibana.conf

**[root@server ~]#** *vi /etc/nginx/nginx.conf*

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log;

pid /run/nginx.pid;

include /usr/share/nginx/modules/\*.conf;

events {

worker\_connections 1024;

}

http {

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 65;

types\_hash\_max\_size 2048;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

include /etc/nginx/conf.d/\*.conf;

~~server {~~

~~listen 80 default\_server;~~

~~listen [::]:80 default\_server;~~

~~server\_name \_;~~

~~root /usr/share/nginx/html;~~

~~include /etc/nginx/default.d/\*.conf;~~

~~location / {~~

~~}~~

~~error\_page 404 /404.html;~~

~~location = /40x.html {~~

~~}~~

~~error\_page 500 502 503 504 /50x.html;~~

~~location = /50x.html {~~

~~}~~

~~}~~

}

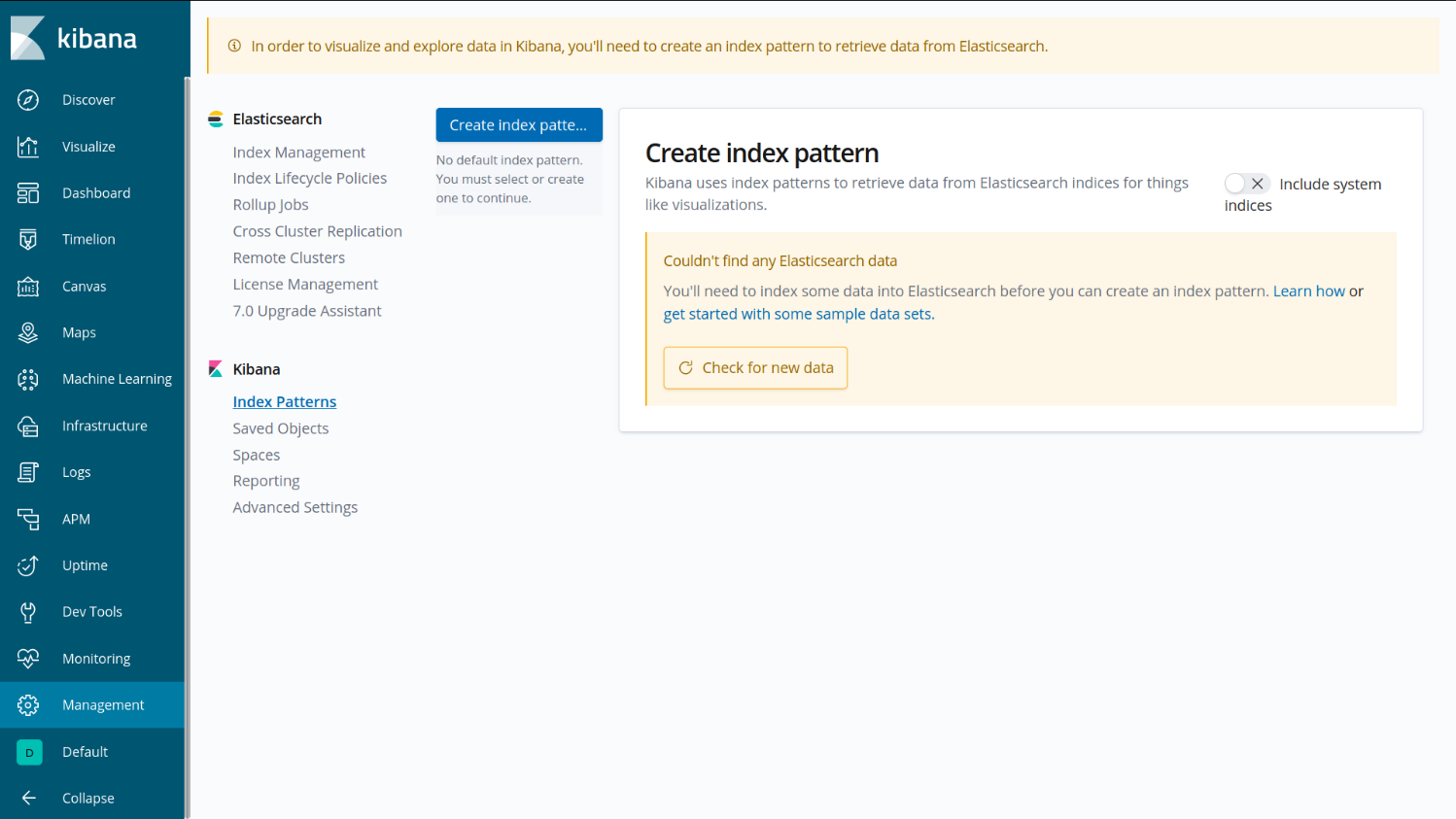
**[root@server ~]#** *cat >>/etc/nginx/conf.d/kibana.conf<<EOF   
> server {   
>     listen 80;   
>     server\_name server.example.com;   
>     location / {   
>         proxy\_pass http://localhost:5601;   
>     }   
> }   
> EOF*

Teraz mogę spokojnie zaktywować i wystartować usługę nginx.

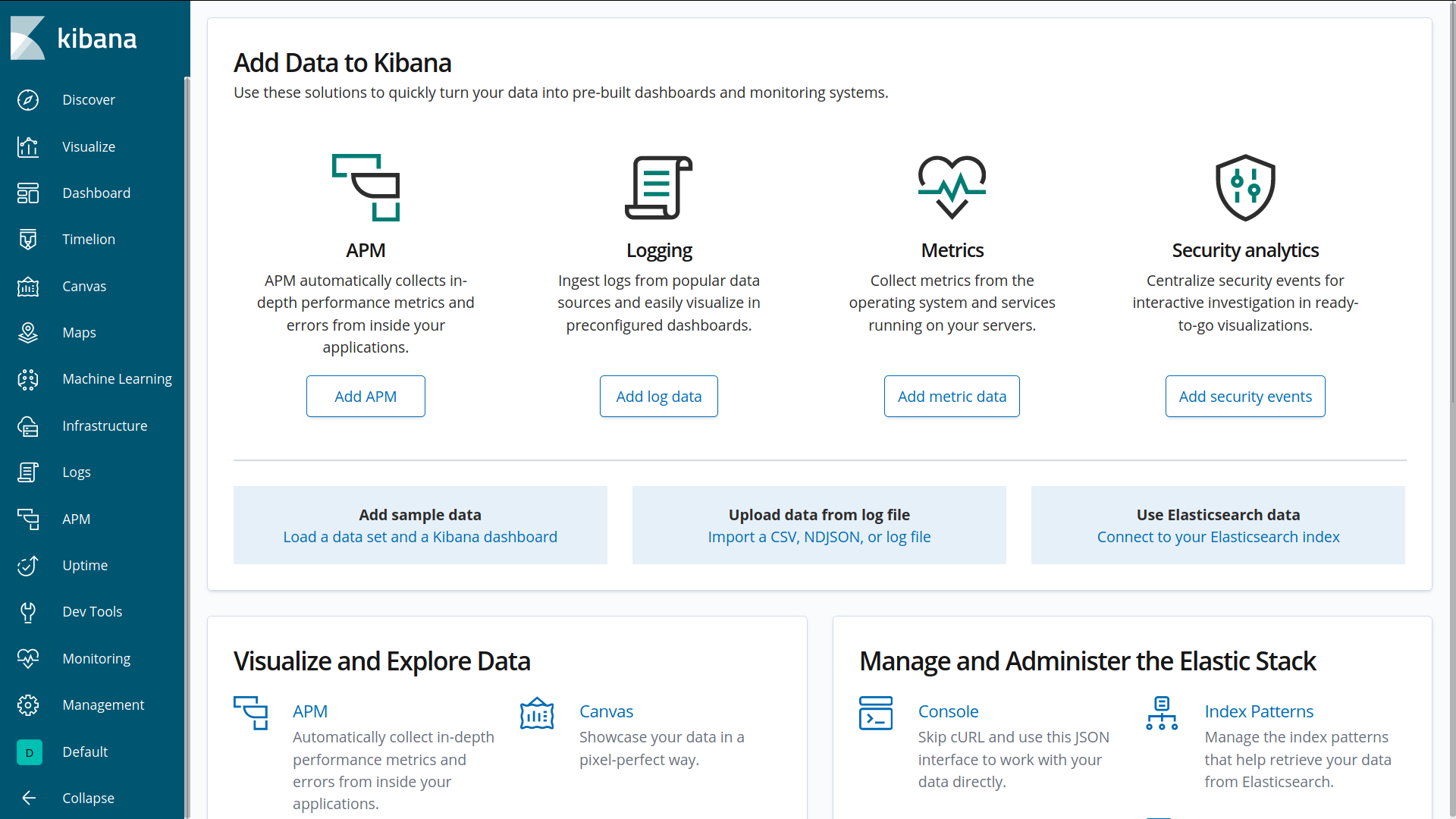
**[root@server ~]#** *systemctl enable nginx*   
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.s  
ervice to /usr/lib/systemd/system/nginx.service.   
**[root@server ~]#** *systemctl start nginx*    
**[root@server ~]#** *systemctl status nginx*   
● nginx.service - The nginx HTTP and reverse proxy server   
  Loaded: (/usr/lib/systemd/system/nginx.service; enabled; vendor preset:disabled)  
  Active: **active (running)** since Thu 2020-02-20 23:46:05 UTC; 6s ago   
 Process: 6865 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)   
 Process: 6863 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)   
Main PID: 6867 (nginx)   
  CGroup: /system.slice/nginx.service   
          ├─6867 nginx: master process /usr/sbin/nginx   
          └─6868 nginx: worker process

**[root@server ~]#** *netstat -ntlp*  
tcp        0      0 0.0.0.0:80      0.0.0.0:\*       LISTEN      6867/nginx: master

Po wpisaniu url: server.example.com w wyszukiwarce hosta pojawia nam się panel do zarządzania Kibana. Obecnie jest nic tam nie ma gdyż nie zostały utworzone żadne wzory indeksowania. Niedługo się to jednak zmieni, gdyż w dalszej części pracy tworzył będą indeksy i stopniowo będą pojawiały się logi, które będzie można wizualizować przy użyciu oprogramowania Kibana.



Rysunek 3



Rysunek 4

Do Kibana wrócę później. Zajmę się teraz instalacją Logstash. Instalacja Logstash jest bardzo analogiczna do instalcji Elastic Search i Kibana dlatego też przejdę od razu do konfiguracji.

Podczas instalacji Logstash postanowiłem wygenerować certyfikat OpenSSL. Jest to certyfikat, który wygasa dopiero po upływie 10 lat (3560 dni). Algorytm szyfrowania to RSA-2048

**[root@server ~]#** *openssl req -subj '/CN=server.example.com/' -x509 -days  
3650 -nodes -batch -newkey rsa:2048 -keyout /etc/pki/tls/private/logsta  
sh.key -out /etc/pki/tls/certs/logstash.crt*   
Generating a 2048 bit RSA private key   
...+++ ...+++ writing new private key to '/etc/pki/tls/private/logstash.key' -----

Teraz zamierzam utworzyć plik konfiguracyjny aby umożliwić usłudze Logstash korzystanie z wygenerowanego certyfikatu. Będą tam zawarte również informacje o tym czego Logstash ma nasłuchiwać, jak przetwarzać dane wejściowe oraz gdzie je dalej przesyłać do przechowywania

**[root@server ~]#** *sudo vi /etc/logstash/conf.d/01-logstash-simple.conf*

input {

beats {

port => 5044

ssl => true

ssl\_certificate => "/etc/pki/tls/certs/logstash.crt"

ssl\_key => "/etc/pki/tls/private/logstash.key"

}

}

filter {

if [type] == "syslog" {

grok {

match => {

"message" => "%{SYSLOGTIMESTAMP:syslog\_timestamp} %{SYSLOGHOST:syslog\_hostname} %{DATA:syslog\_program}(?:\[%{POSINT:syslog\_pid}\])?: %{GREEDYDATA:syslog\_message}"

}

add\_field => [ "received\_at", "%{@timestamp}" ]

add\_field => [ "received\_from", "%{host}" ]

}

syslog\_pri { }

date {

match => [ "syslog\_timestamp", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]

}

}

}

output {

elasticsearch {

hosts => "localhost:9200"

index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"

}

}

Ten plik konfiguracyjny ma trzy sekcje: wejściowa (input), filtrująca (filter), oraz wyjściowa (output).

W sekcji wejściowej mamy informację o tym czego Logstash nasłuchuje. Na jakich portach i protokołach działają usługi oraz jakimi certyfikatami się legitymują. Sekcja filtrująca sprawdza każdy log i wyszukuje słów kluczowych a następnie dokonuje parsowania tekstu. Sekcja wyjściowa nadaje stempel czasowy oraz zawiera informacje o tym gdzie przetworzone logi mają być składowane. W znacznej większości przypadków kolejnym krokiem jest usługa Elastic Search. W tym miejscu nadawana jest również nazwa indeksu. Jeżeli jest taka potrzeba to każda z sekcji mogłaby być oddzielnym plikiem konfiguracyjnym jednak w moim przypadku wszystko jest razem.

Startuję usługę Logstash.

**[root@server ~]#** *systemctl enable logstash*   
Created symlink from /etc/systemd/system/multi-user.target.wants/logstash.service to /etc/systemd/system/logstash.service.   
**[root@server ~]#** *systemctl start logstash && systemctl status logstash*    
● logstash.service - logstash   
  Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabled)   
  Active: **active (running)** since Fri 2020-02-21 01:25:33 UTC; 5s ago   
Main PID: 7178 (java)   
  CGroup: /system.slice/logstash.service   
          └─7178 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX...  
**[root@server ~]#** *netstat -ntlp*

tcp6       0      0 :::5044          :::\*       LISTEN      7178/java

Na ten moment po stronie serwera wszystko jest gotowe. Teraz zabieram się do konfiguracji maszyny wirtualnej CentOS 7, która jest klientem.

**[szymon@LinuxManjaro** **vagrant-provisioning]$** *ssh root@client*

Instaluje FileBeat z repozytorium.

**[root@client ~]#** *cat >>/etc/yum.repos.d/elk.repo<<EOF*

**[root@client ~]#** *yum install -y filebeat*

Kopiuję certyfikat SSL z server.example.com

**[root@client ~]#** *scp server.example.com:/etc/pki/tls/certs/logstash.crt   
 /etc/pki/tls/certs/*   
The authenticity of host 'server.example.com (172.42.42.10)' can't be established.  
ECDSA key fingerprint is SHA256:bpcGW71ifmDeU3McxojVed28gnrkr3aT0Nf81o142RY.   
ECDSA key fingerprint is MD5:96:e0:e4:70:3f:5c:03:f0:bc:f8:9d:a1:d7:29:45:ca.   
Are you sure you want to continue connecting (yes/no)? **yes**   
Warning: Permanently added 'server.example.com,172.42.42.10' (ECDSA) to the list of known hosts.   
root@server.example.com's password: **admin**    
logstash.crt                          100% 1119   863.3KB/s   00:00

Sprawdzam czy certyfikat poprawnie się przekopiował.

**[root@client ~]#** *ls /etc/pki/tls/certs*   
ca-bundle.crt        logstash.crt     Makefile   
ca-bundle.trust.crt  make-dummy-cert  renew-dummy-cert

Następnie edytuję plik konfiguracyjny filebeat.

**[root@client ~]#** *vi /etc/filebeat/filebeat.yml*

#=========================== Filebeat inputs =============================

filebeat.inputs:

- type: log

# Change to true to enable this input configuration.

enabled: ~~false~~ **true**

# Paths that should be crawled and fetched. Glob based paths.

paths:

~~- /var/log/\*.log~~

**- /var/log/messages**

**- /var/log/secure**

#-------------------------- Elasticsearch output ------------------------------

**#**output.elasticsearch:

# Array of hosts to connect to.

~~hosts: ["localhost:9200"]~~

**#**hosts: ["localhost:9200"]

# Enabled ilm (beta) to use index lifecycle management instead daily indices.

#ilm.enabled: false

# Optional protocol and basic auth credentials.

#protocol: "https"

#username: "elastic"

#password: "changeme"

#----------------------------- Logstash output --------------------------------

~~#output.logstash:~~

output.logstash:

# The Logstash hosts

~~#hosts: ["localhost:5044"]~~

hosts: ["**server.example.com**:5044"]

# Optional SSL. By default is off.

# List of root certificates for HTTPS server verifications

~~#ssl.certificate\_authorities: ["/etc/pki/root/ca.pem"]~~

ssl.certificate\_authorities: ["/etc/pki/**tls/certs/logstash.crt**"]

# Certificate for SSL client authentication

#ssl.certificate: "/etc/pki/client/cert.pem"

# Client Certificate Key

#ssl.key: "/etc/pki/client/cert.key"

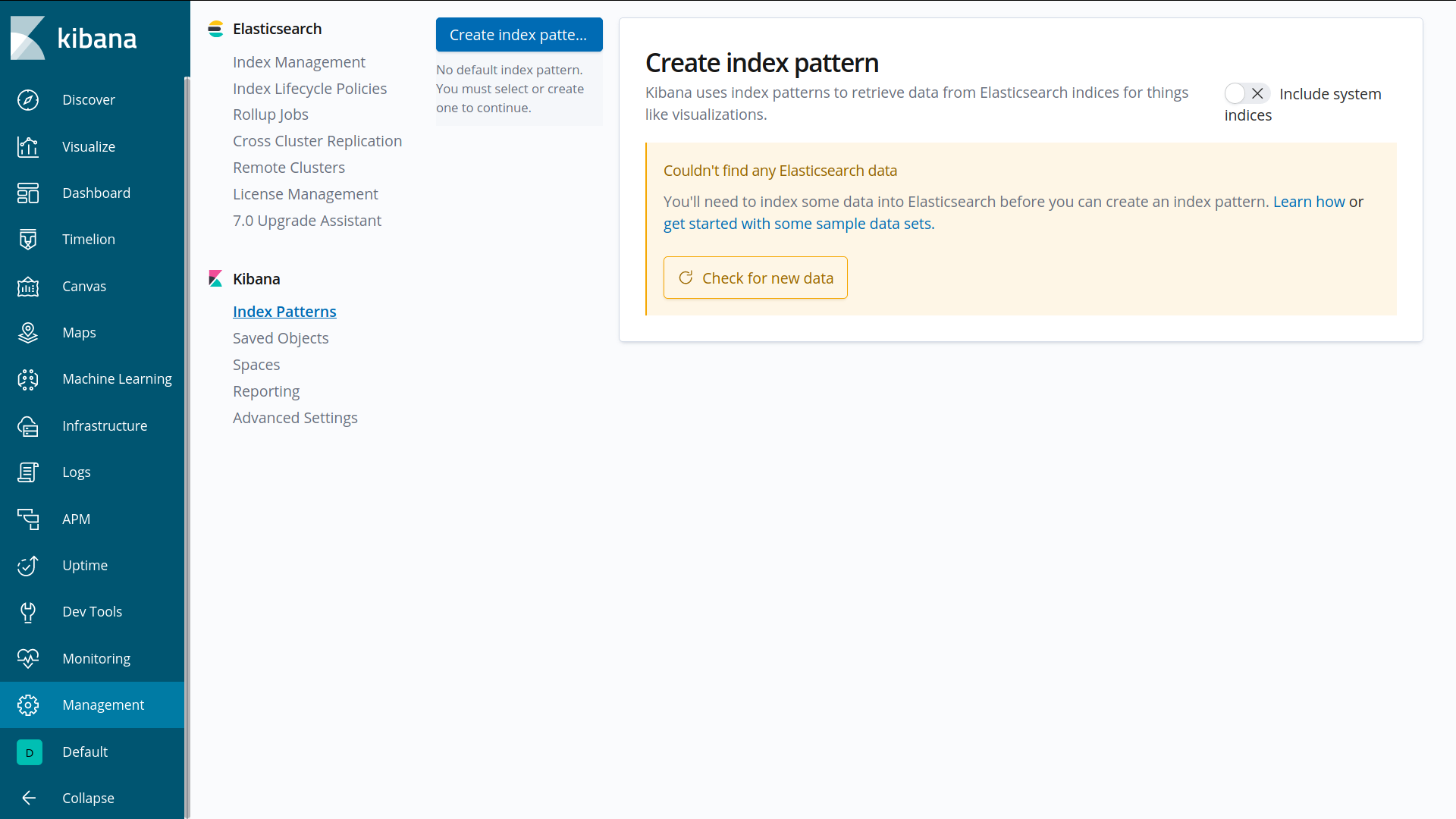
Folder /var/log/secure przechowuje logi, które są zweryfikowane certyfikatem.

Oraz w sekcji output nie chcę aby logi były przesyłane bezpośrednio do Elastic Search tylko aby były

przekierowane do Logstash.

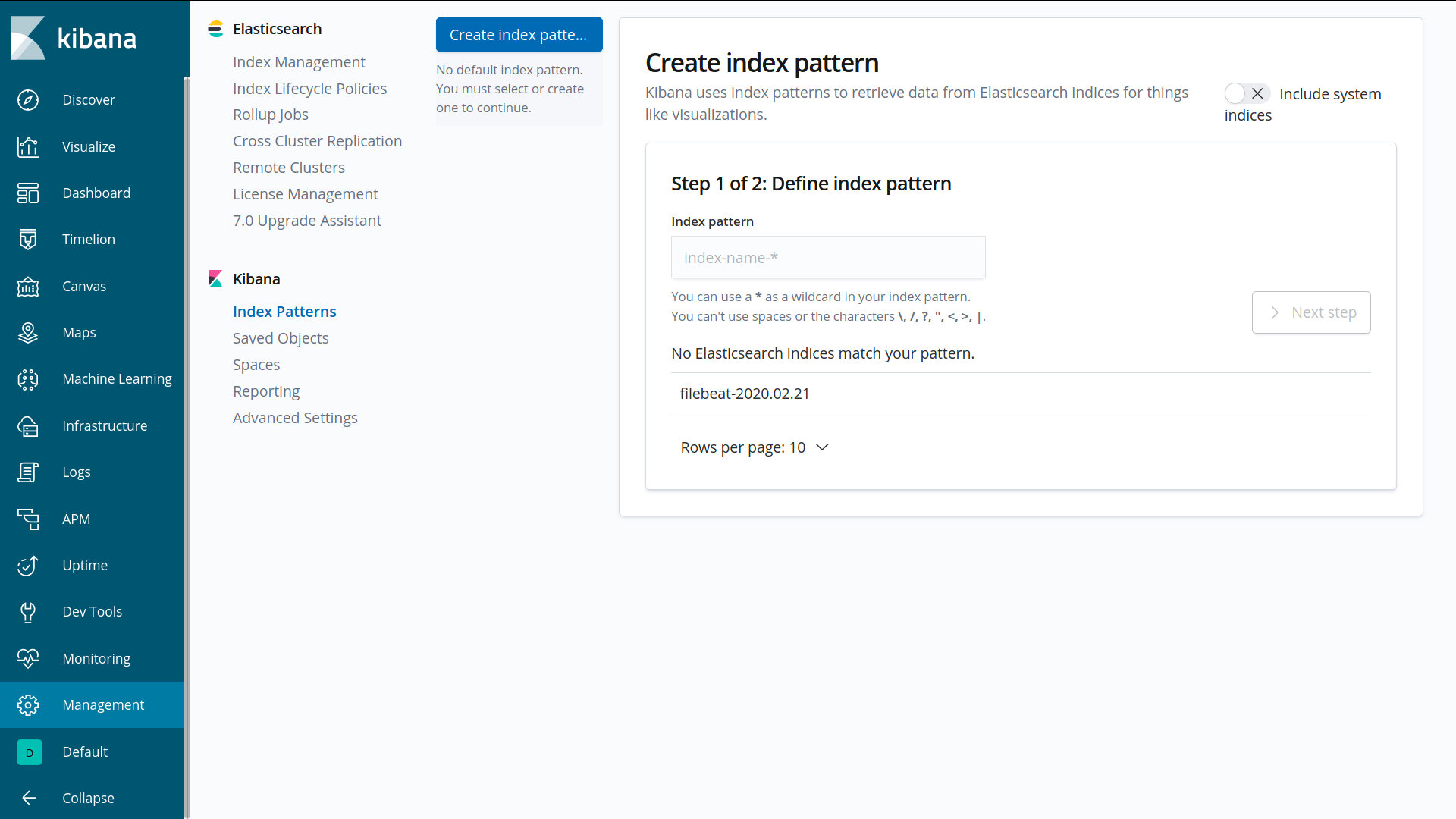
**[root@client ~]#** *systemctl enable filebeat*   
Created symlink from /etc/systemd/system/multi-user.target.wants/filebeat.service to /usr/lib/systemd/system/filebeat.service.   
**[root@client ~]#** *systemctl start filebeat*

Teraz wracam do Dashboard Kibana.



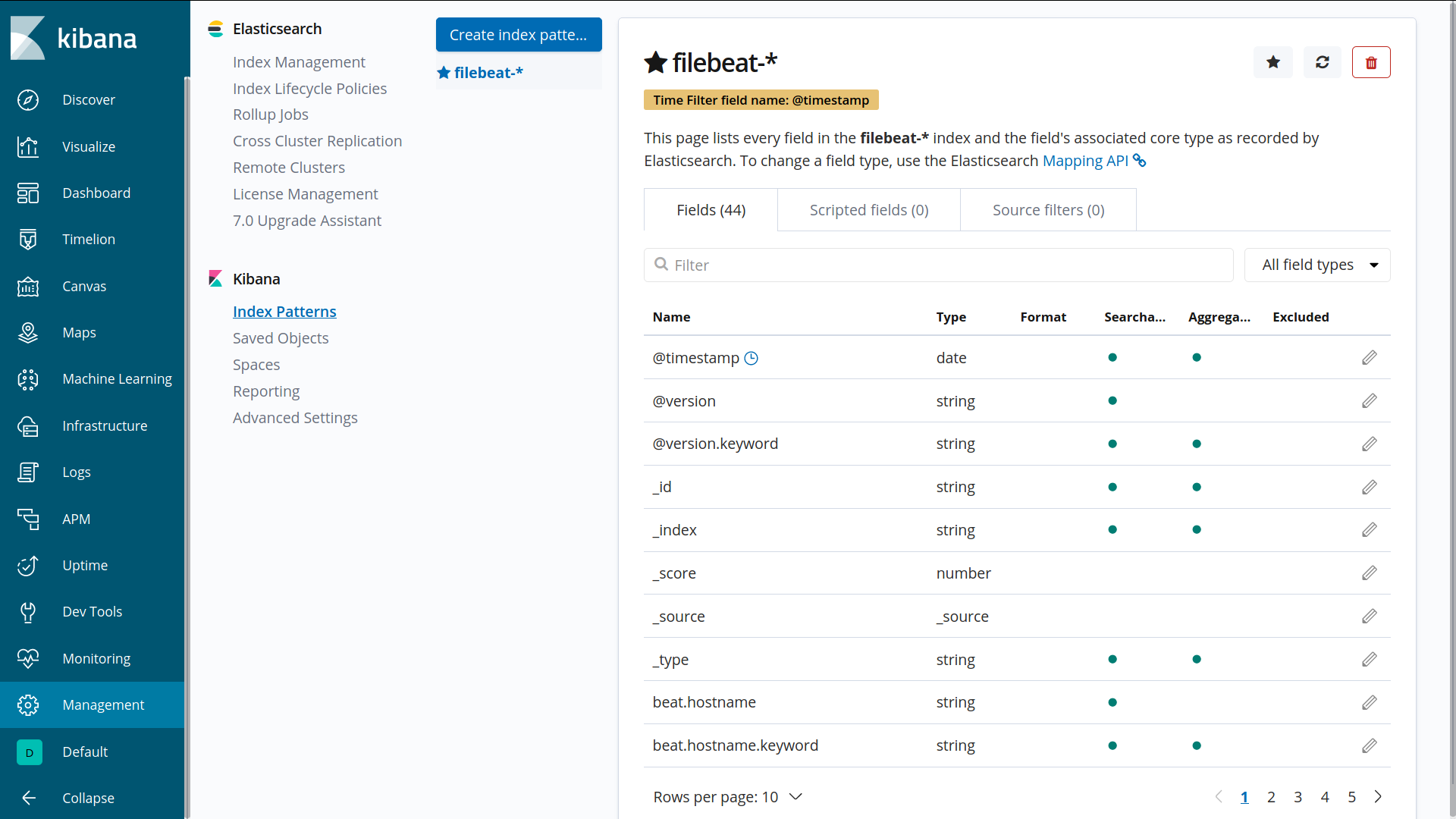
Rysunek 5

Po naciśnięciu przycisku „Check for new data” pojawił się indeks wygenerowany przez Filebeat.

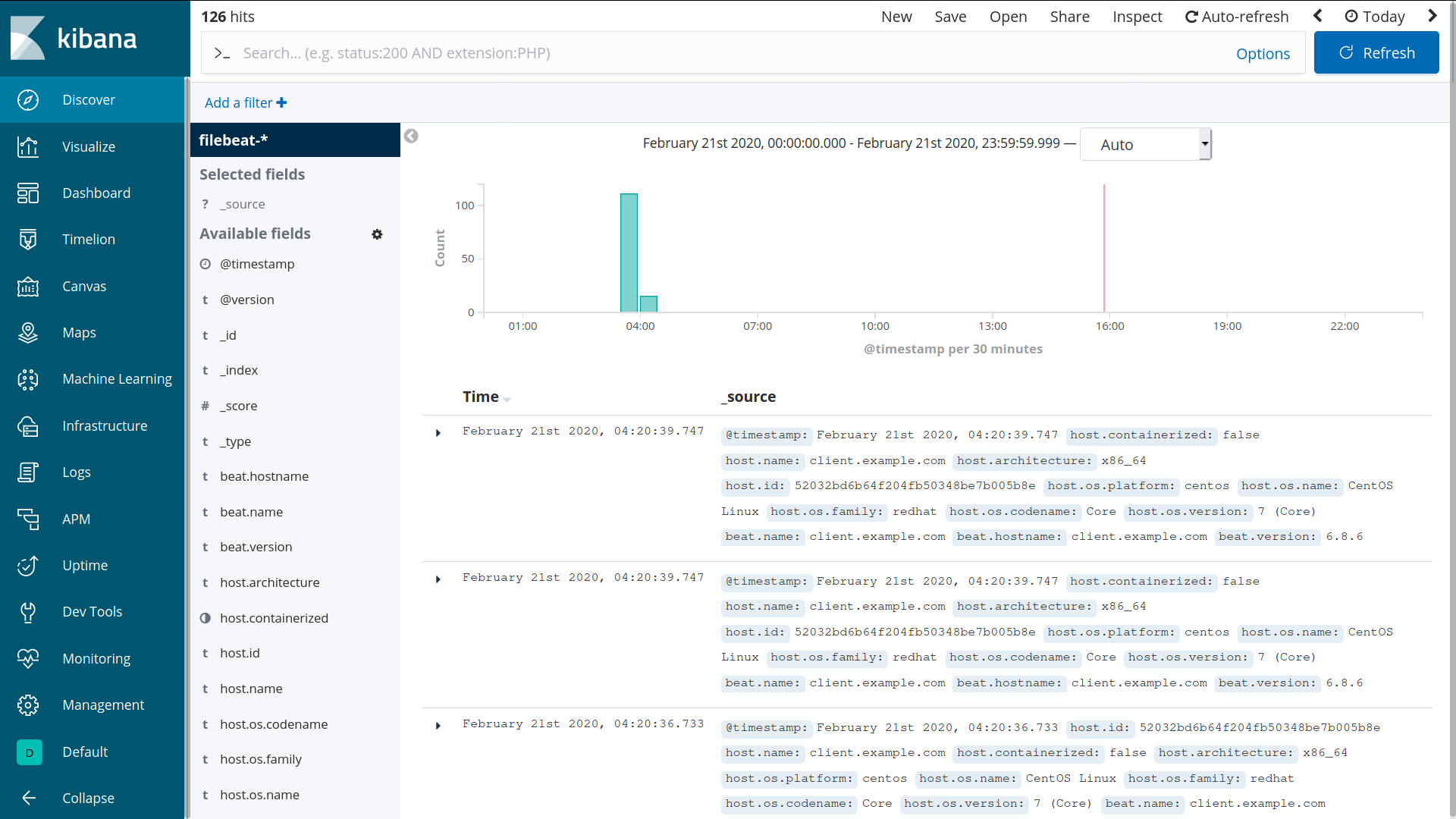


Rysunek 6

W polu tekstowym wpisuję „filebeat-\*” a następnie naciskam przycisk „Next step”.



Potem z rozwijanej listy wybieram „@timestamp”, następnie przycisk „Create index patern”. W ten sposób został utworzony indeks widoczny obok.



Rysunek 7

W zakładce Discover można zobaczyć wpadające logi.

* Można filtrować logi po wybranych w zakładce „Available fields” parametrach.
* Zarówno w polu „search” można wyszukiwać we wszystkich logach słów kluczowych.
* Można również wpisywać bardziej szczegółowe zapytania takie jak: pokaż logi, które jako źródło mają folder secure (source: "/var/log/secure/").
* Również działają zapytania łączone. (source: "/var/log/secure/" and root)

To rozwiązanie z wyszukiwaniem jest analogiczne do tego jak podobna funkcjonalność została zaimplementowana w konkurencyjnym oprogramowaniu Splunk. Jednakże w stosie ELK jest to rozwiązanie bogatsze i o większych możliwościach. Ponadto ELK jest oprogramowaniem na licencji open source co również bardziej przemawia na korzyść rozwiązania zaimplementowanego w mojej pracy magisterskiej w porównaniu ze stosunkowo drogim oprogramowaniem Splunk.

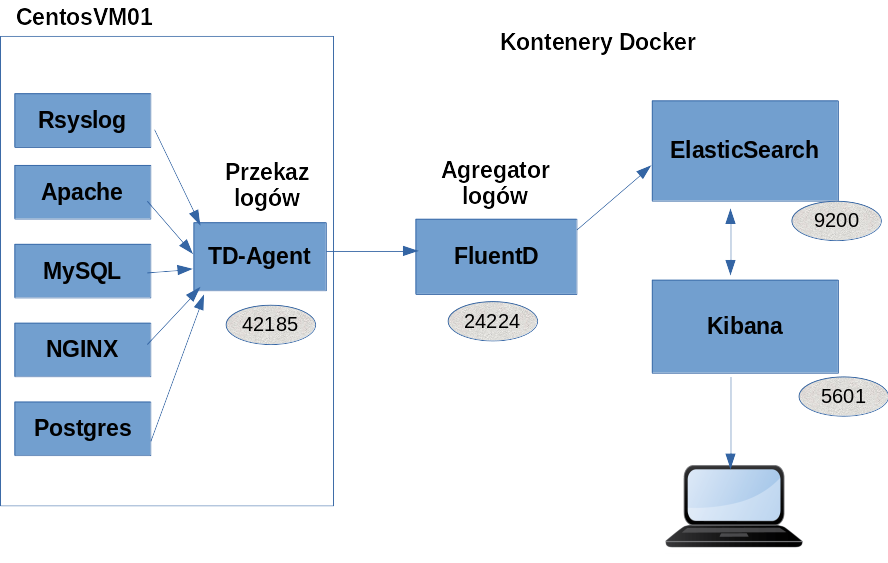
Kolejne paragrafy mojej pracy nadal będą poświęcone konfigurowaniu kolejnych funkcjonalności. Dotychczas rozwiązanie było implementowane na systemie operacyjnym CentOS 7 jednak w dalszej perspektywie zostanie poruszony temat maszyny wirtualnej Ubuntu oraz kontenerach Docker.

# Rozdział 6

W tym rozdziale skonfiguruję stos EFK: ElasticSearch, FluentD i Kibana.

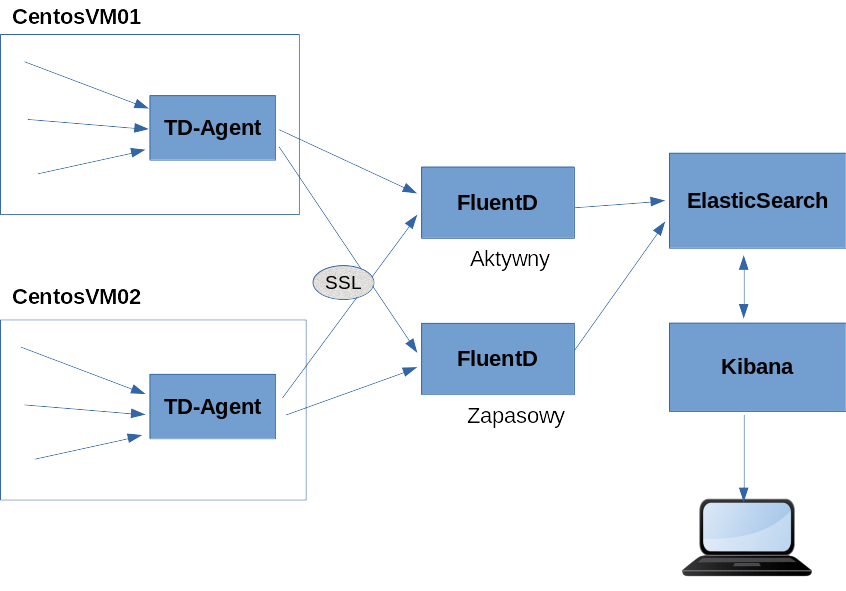
Chociaż w temacie mojej pracy występuje stos ELK to w tym rozdziale skupię się na EFK czyli Logstash zamieniam na FluentD. Dzięki takiemu zabiegowi będę mógł zaprezentować wysokość dostępność rozwiązania poprzez duplikacje kontenerów Dockera, które pełnią rolę agregatorów logów i pracują w architekturze Master-Slave. Jest to kluczowe rozwiązanie gdy administratorowi zależy na ciągłości i niezawodności w działaniu infrastruktury gromadzącej logi. Dzięki takiemu zabiegowi niestraszne są przerwy w dostawie usługi i chwilowe awarie. Unikam wtedy sytuacji, kiedy mam krytyczny, pojedynczy punkt awarii (single point of failure). Zostaną zaimplementowane również bufory (TD-Agent, log forwarder), które przez pewien czas będą gromadziły pojawiające się logi i kiedy usługa FluentD z powrotem stanie się aktywna po awarii to wszystkie informacje i zdarzenia szybko i sprawnie znajdą się w systemie agregującym. FluentD jest również bardzo lekkim i nienadmiernie odciążającym pamięć RAM oraz procesor rozwiązaniem.

Wszystkie usługi EFK będą instalowane i konfigurowane na kontenerach Docker.

Logi będą generowane przez usługi zaimplementowane ma klienckiej maszynie wirtualnej CentOS 7. Logi te będą przekazywane przez TD-Agent do agregatora logów FluentD. Tam będą parsowane, pobierane przez ElasticSearch. Pod odpowiednim prze-procesowaniu do akcji wkroczy Kibana, która te przeprocesowane logi zwizualizuje. TD-Agent przyjmuje najróżniejszego rodzaju logi: systemowe, wiadomości folderu var, Apache, mySQL, nginx, postgres i wiele, wiele innych. Tych źródeł generujących logi i wiadomości może być bardzo dużo.

Rysunek 8

Jak już pisałem chwilę wcześniej FluentD może działać w trybie klastra.

Kiedy jeden z kontenerów FluentD padnie drugi natychmiastowo przejmuje jego rolę i cała infrastruktura przez cały czas może działać w trybie awaryjnym opartym tylko o jedną aktywną usługę. Wtedy wszystkie TD-Agent przekazywać będą logi tylko do jednego FluentD. Natomiast kiedy padnie drugi FluentD logi tymczasowo będą buforowane, przechowywane w TD-Agent. Tak aby kiedy jeden z FluentD powrócił do działalności mógłby za jednym razem otrzymać wszystkie zaległe logi i je zagregować. Domyślnie dane przesyłane są czystym tekstem natomiast jest taka możliwość aby połączenie między TD-Agent a FluentD było w pełni szyfrowane za pomóc certyfikatu SSL. Jest to rozwiązanie opcjonalne znacząco zwiększające poufność i bezpieczeństwo przesyłanych danych.

Rysunek 9

Poniżej prezentuję plik konfiguracyjny Docker Compose.

version: '2.2'

services:

fluentd:

build: ./fluentd

container\_name: fluentd

volumes:

- ./fluentd/conf:/fluentd/etc

ports:

- "24224:24224"

- "24224:24224/udp"

# Add this to /etc/sysctl.conf for making it permanent

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:6.5.4

container\_name: elasticsearch

environment:

- bootstrap.memory\_lock=true

- "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"

ulimits:

memlock:

soft: -1

hard: -1

volumes:

- esdata1:/usr/share/elasticsearch/data

ports:

- 9200:9200

kibana:

image: docker.elastic.co/kibana/kibana:6.5.4

container\_name: kibana

environment:

ELASTICSEARCH\_URL: "http://elasticsearch:9200"

ports:

- 5601:5601

depends\_on:

- elasticsearch

volumes:

esdata1:

driver: local

Ten plik konfiguracyjny przedstawia instalację trzech usług FluentD, ElasticSearch oraz Kibana. Kibana działa na porcie 5601 i jest przypięta do lokalnego komputera. Podobnie działa ElasticSearch na porcie 9200 oraz FluentD z portem 24224. Wszystkie usługi nasłuchują na protokole TCP oraz FluentD nasłuchuje protokołu TCP oraz UDP. Obrazy usług ElasticSearch oraz Kibana pobierane są oficjalnych repozytoriów natomiast obraz FluentD budowany jest ze zmodyfikowanego przeze mnie obrazu Docker. Niżej prezentuję dockerfile FluentD.

FROM fluent/fluentd   
RUN ["gem", "install", "fluent-plugin-elasticsearch", "--no-rdoc", "--no-ri"]

Do oficjalnego obrazu FluentD musi zostać doinstalowana wtyczka ElasticSearch. Dopóki ta wtyczka nie zostanie zainstalowana ElasticSearch i FluentD nie będą się wzajemnie widziały.

Niżej prezentuję również kolejny plik konfiguracyjny FluentD.

<source>

@type forward

port 24224

</source>

# Store Data in Elasticsearch

<match \*.\*\*>

@type copy

<store>

@type elasticsearch

host elasticsearch

port 9200

include\_tag\_key true

tag\_key @log\_name

logstash\_format true

flush\_interval 10s

</store>

</match>

Definiuję tutaj port źródłowy z którego ElasticSearch będzie pobierał logi z FluentD. Jakakolwiek wiadomość bądź informacja nie pojawi się na tym porcie ma natychmiastowo zostać przekopiowana i przekazana dalej. Dzięki temu, że aplikacje uruchomione są za pomocą docker-compose mogą się wzajemnie ze sobą komunikować używając jedynie nazw swoich kontenerów.

Nadszedł czas aby rozpocząć usługi poprzez wpisanie odpowiedniego polecenia.

**[szymon@LinuxManjaro** **docker-efk]$** *docker-compose up -d*

**[szymon@LinuxManjaro** **docker-efk]$** *docker-compose ps*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Command | State | Ports |
| elasticsearch | /usr/local/bin/docker | Up | 0.0.0.0:9200->9200/tcp, 9300/tcp |
| fluentd | /bin/entrypoint.sh/bin/sh | Up | 0.0.0.0:24224->24224/tcp, 0.0.0.0:24224->24224/udp, 5140/tcp |
| kibana | /usr/local/bin/kibana | Up | 0.0.0.0:5601→5601/tcp |

Świetnie, mam aktywne kontenery. Sprawdzę teraz logi każdego z kontenerów.

**[szymon@LinuxManjaro** **docker-efk]$** *docker-compose logs -f kibana*

Attaching to kibana

kibana           | {"type":"log","@timestamp":"2020-02-24T11:06:06Z","tags":["status","plugin:security@6.5.4","info"],"pid":1,"state":"green","message": "Status changed from yellow to green - Ready","prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}

Status zmienił się z żółtego na zielony co oznacza, że usługa działa i czeka na ElasticSearch.

**[szymon@LinuxManjaro** **docker-efk]$** *docker-compose logs -f elasticsearch*  
elasticsearch    | [2020-02-24T11:06:14,452][INFO][o.e.c.m.MetaDataMappingService] [SNd4pFR] [logstash-2020.02.24/WXHY9Z6hSMiM2rs93f3Jaw] create\_mapping [fluentd]

Również tutaj to wygląda w porządku. Widać, że ElasticSearch oczekuje logów od FluentD.

**[szymon@LinuxManjaro** **docker-efk]$** *docker-compose logs -f fluentd*  
fluentd          | 2020-02-24 [info]: #0 listening port port=24224 bind="0.0.0.0"   
fluentd          | 2020-02-24 [info]: #0 fluentd worker is now running worker=0

FluentD również zameldował swoją gotowość do pracy.

**[szymon@LinuxManjaro docker-efk]$** *sudo netstat -nltp | grep docker*

tcp6 0 0 :::9200 :::\* LISTEN 6008/docker-proxy

tcp6 0 0 :::24224 :::\* LISTEN 5899/docker-proxy

tcp6 0 0 :::5601 :::\* LISTEN 6175/docker-proxy

Kiedy już część z kontenerami Dockera działa poprawnie zabieram się za konfigurację części z kliencką maszyną wirtualną. Zainstaluję na niej TD-Agent oraz skonfiguruję syslog. Jako, że we wcześniejszej części mojej pracy komputery klienckie były stawiane w VirtualBoxie przy pomocy oprogramowania Vagrant teraz postanowiłem użyć LXD. Jest to bardzo lekkie rozwiązanie wirtualizacyjne. Nie jest to Docker, jest to troszkę inny koncept. Nie jest to pełna wirtualizacja. Wykorzystywane są rzeczywiste podzespoły komputera tzw. bare metal. Nie jest to ani symulacja ani emulacja ani wirtualizacja. Rozwiązanie LXD wykorzystuje fizyczny kernel urządzenia na którym jest uruchomione. Docker powstał w celu izolowania procesów. LXD działa na całym systemie operacyjnym i ma dostęp do wszystkich jego zasobów. Inaczej niż to jest rozwiązane w maszynach wirtualnych gdzie na stałe należy przydzielić danej maszynie pewną ilość pamięci RAM. Kontenery LXD mogą używać elastycznie całej pamięci RAM wymiennie z innymi programami uruchomionymi na tym samym komputerze hosta.

**[szymon@LinuxManjaro** **play]$** *sudo lxc launch images:centos/7 efkclient*  
**[szymon@LinuxManjaro play]$** *sudo lxc start efkclient*

**[szymon@LinuxManjaro play]$** *sudo lxc list*

+-----------+---------+----------------------+------+-----------+-----------+

| NAME | STATE | IPV4 | IPV6 | TYPE | SNAPSHOTS |

+-----------+---------+----------------------+------+-----------+-----------+

| efkclient | RUNNING | 10.77.248.187 (eth0) | | CONTAINER | 0 |

+-----------+---------+----------------------+------+-----------+-----------+

**[szymon@LinuxManjaro** **play]$** *sudo lxc exec efkclient bash*

**[root@efkclient ~]#** *cat /etc/redhat-release*    
CentOS Linux release 7.7.1908 (Core)

**[root@efkclient ~]#** *uname -r*   
5.4.15-2-MANJARO  
**[root@efkclient ~]#** *free -m*

total used free shared buff/cache available

Mem: 15907 66 15759 7 81 15841

Swap: 5892 0 5892  
Dzięki tym poleceniom mogłem przekonać się, że kontener LXC korzysta bezpośrednio z kernela hosta.

Instaluję aplikację umożliwiającą wykonywanie poleceń z prawami administratora.

**[root@efkclient ~]#** *yum install sudo -y*

Kolejne polecenie doda repozytorium oraz zainstaluje wymagany pakiet.

**[root@efkclient ~]# $** *curl -L https://toolbelt.treasuredata.com/sh/install-redhat-td-agent3.sh | sh*

To polecenie zaktywuje usługę.

**[root@efkclient ~]#** *systemctl enable td-agent*

Przed rozpoczęciem usługi należy ją jeszcze skonfigurować. Należy poinformować o tym gdzie znajduje się FluentD z którym TD-Agent ma współpracować.

<source>

@type syslog

@id input\_syslog

port 42185

tag efkclient.system

</source>

<match \*.\*\*>

@type forward

@id forward\_syslog

<server>

host 10.3.168.127

</server>

</match>

Analogicznie do pliku konfiguracyjnego fluent.conf w sekcji źródła mam informację o typie logu, nasłuchującego portu oraz nazwy urządzenia, które nasłuchuje. Jeśli byłoby wiele źródeł logów to również można by dodać więcej analogicznych sekcji source. W tym miejscu jeżeli FluentD było by zduplikowane w roli aktywny-pasywny w sekcji match należałoby dodać kolejny adres logiczny IP.

Teraz mogę wystartować usługę TD-Agent.

**[root@efkclient ~]#** *systemctl start td-agent*

Logi można podglądać w folderze /var/log. Tam też jest widziany bufor, o którym pisałem wcześniej.

**[root@efkclient td-agent]#** *pwd && ls*   
/var/log/td-agent   
buffer  td-agent.log

Teraz skonfiguruję syslog tak aby był przekazywany do TD-Agent.

**[root@efkclient td-agent]#** *vi /etc/rsyslog.conf*

Idę do końca pliku i umieszczam tam następującą linie konfiguracji.

\*.\* @127.0.0.1:42185

Znaczy to nie mniej nie więcej tyle, że cokolwiek przejdzie przez rsyslog ma być przekazywane na port na którym uruchomiony jest TD-Agent. Pojedynczy @ oznacza, że prekazywany będzie tylko komunikat UDP, jeżeli były by dwa znaki @@ oznaczałoby to aby przekazywane były zarówno komunikaty UDP jak i TCP.

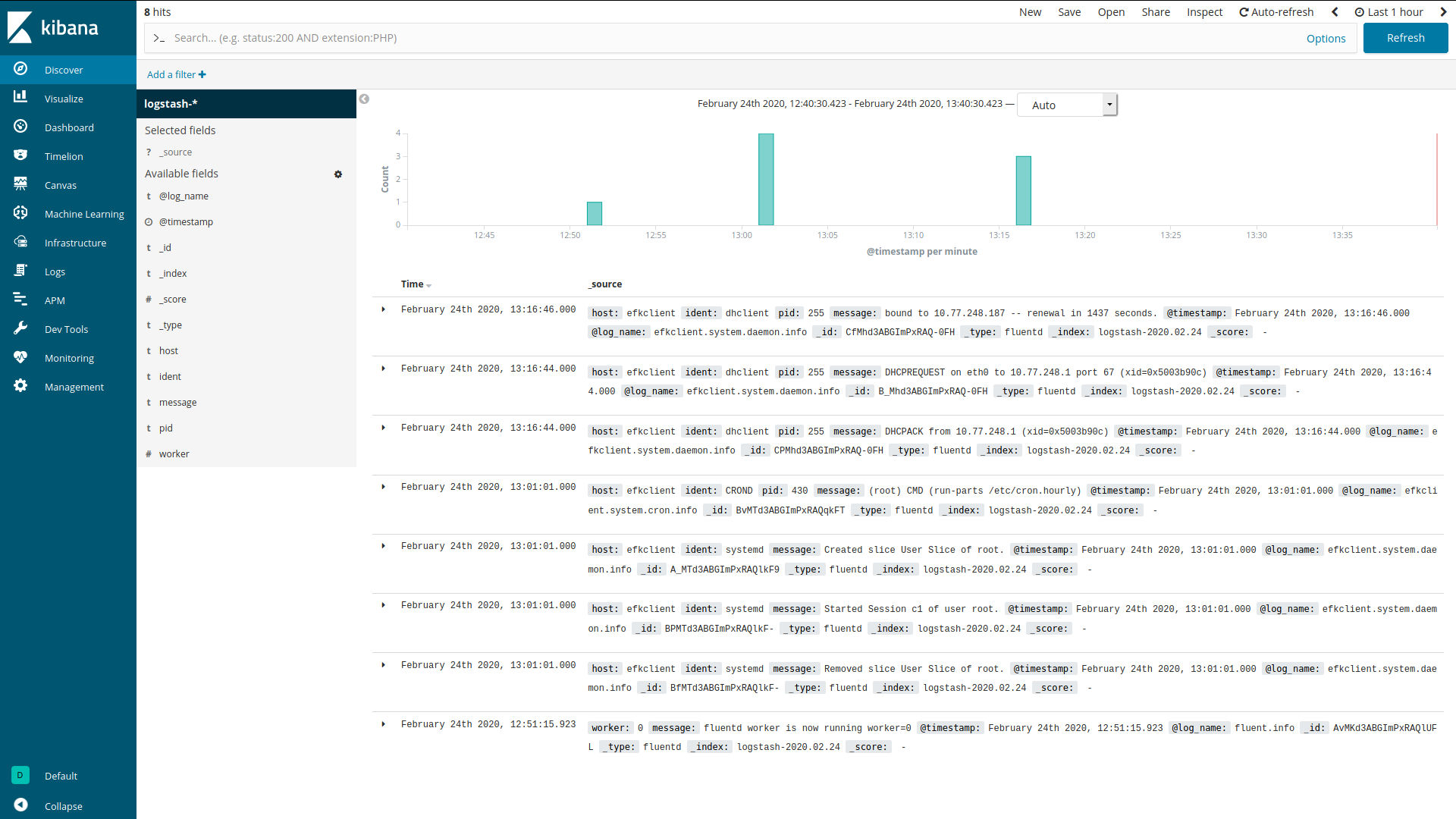
Teraz jest pora na restart usługi rsyslog.

**[root@efkclient td-agent]#** *systemctl restart rsyslog*

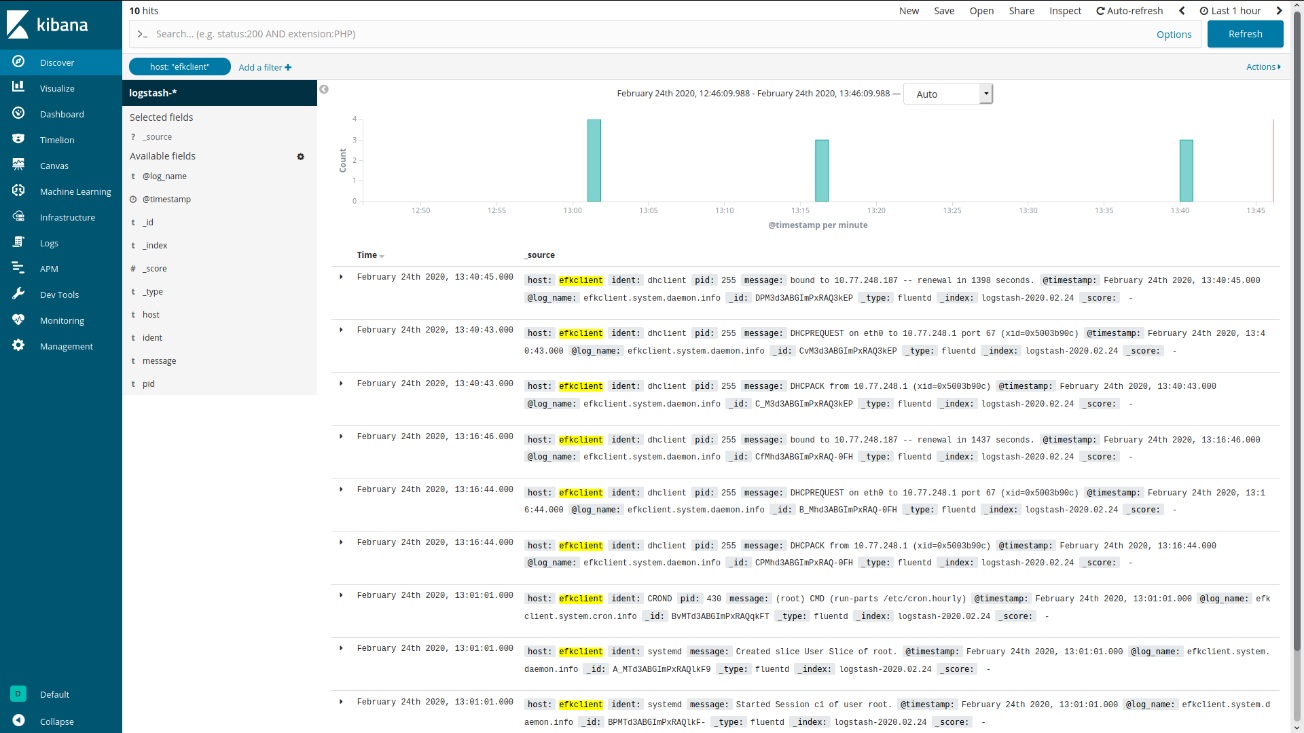
**[root@efkclient td-agent]#** *yum install -y net-tools*

**[root@efkclient td-agent]#** *netstat -nlup*   
Active Internet connections (only servers)   
Proto Recv-Q Send-Q Local Address    Foreign Address  State       PID/Program name       
udp        0      0 0.0.0.0:68       0.0.0.0:\*            255/dhclient       
udp        0      0 0.0.0.0:57544           0.0.0.0:\*            317/rsyslogd       
udp        0      0 0.0.0.0:42185           0.0.0.0:\*            402/ruby

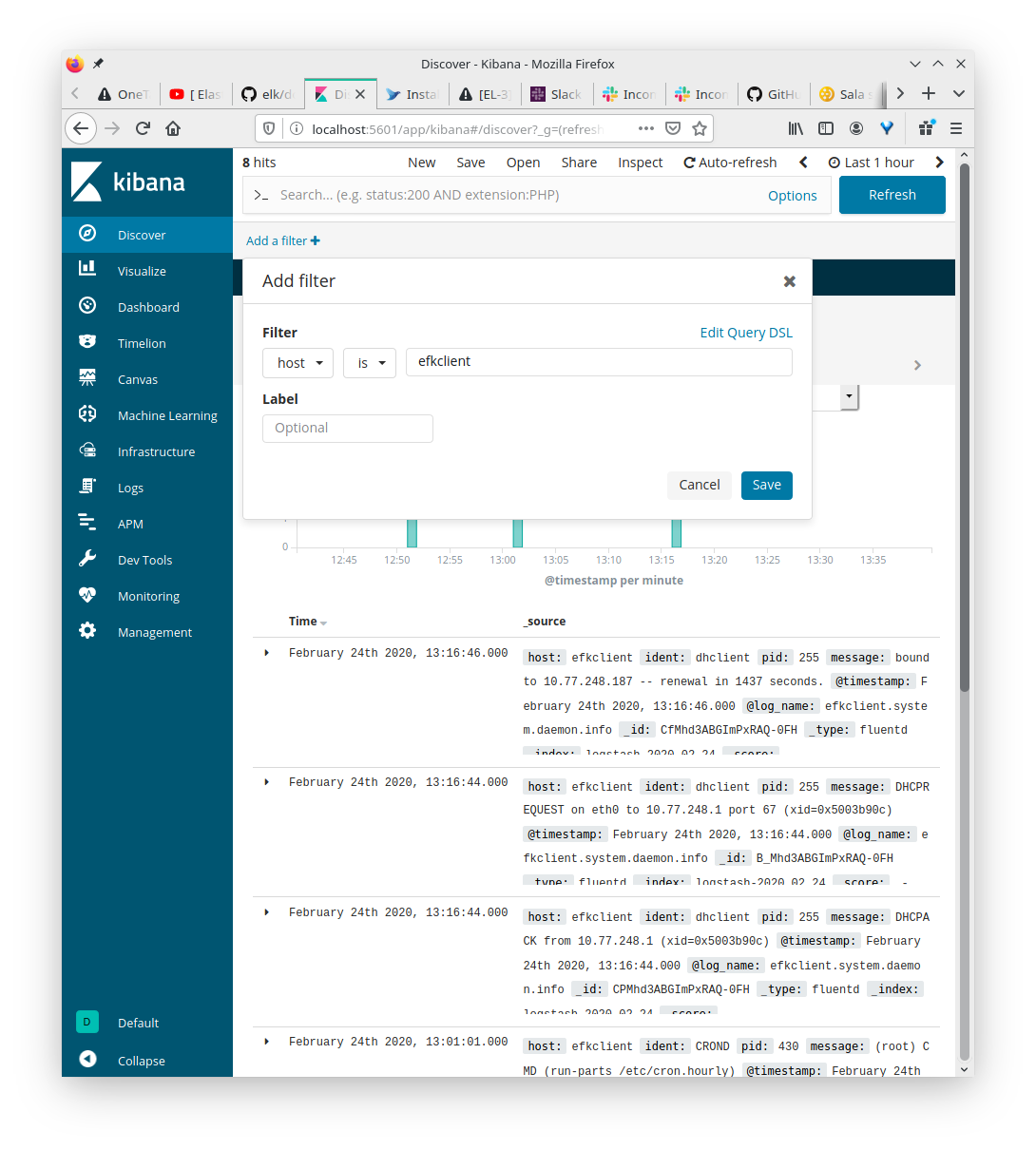
Na ten moment poprawnie skonfigurowałem rsyslog, który przekazuje logi do TD-Agent, który następnie przesyła ja dalej do FluentD, dalej znajdują się w ElasticSearch a na końcu trafiają do Kibany. Sprawdzę teraz czy widoczne są w panelu webowym. Po dodaniu odpowiedniego wzoru indeksowania mogę podziwiać w sekcji Discover logi wygenerowane przez rsyslog.

Z tych logów można wyczytać, że FluentD pracuje jako worker 0 oraz, że logi systemowe są generowane przez kontener LXC efkclient. Mogę dodać również pewne filtry. Filtry wybiorą logi zawierające to słowo kluczowe i zaznaczą je na żółto.

Rysunek 10



Rysunek 11

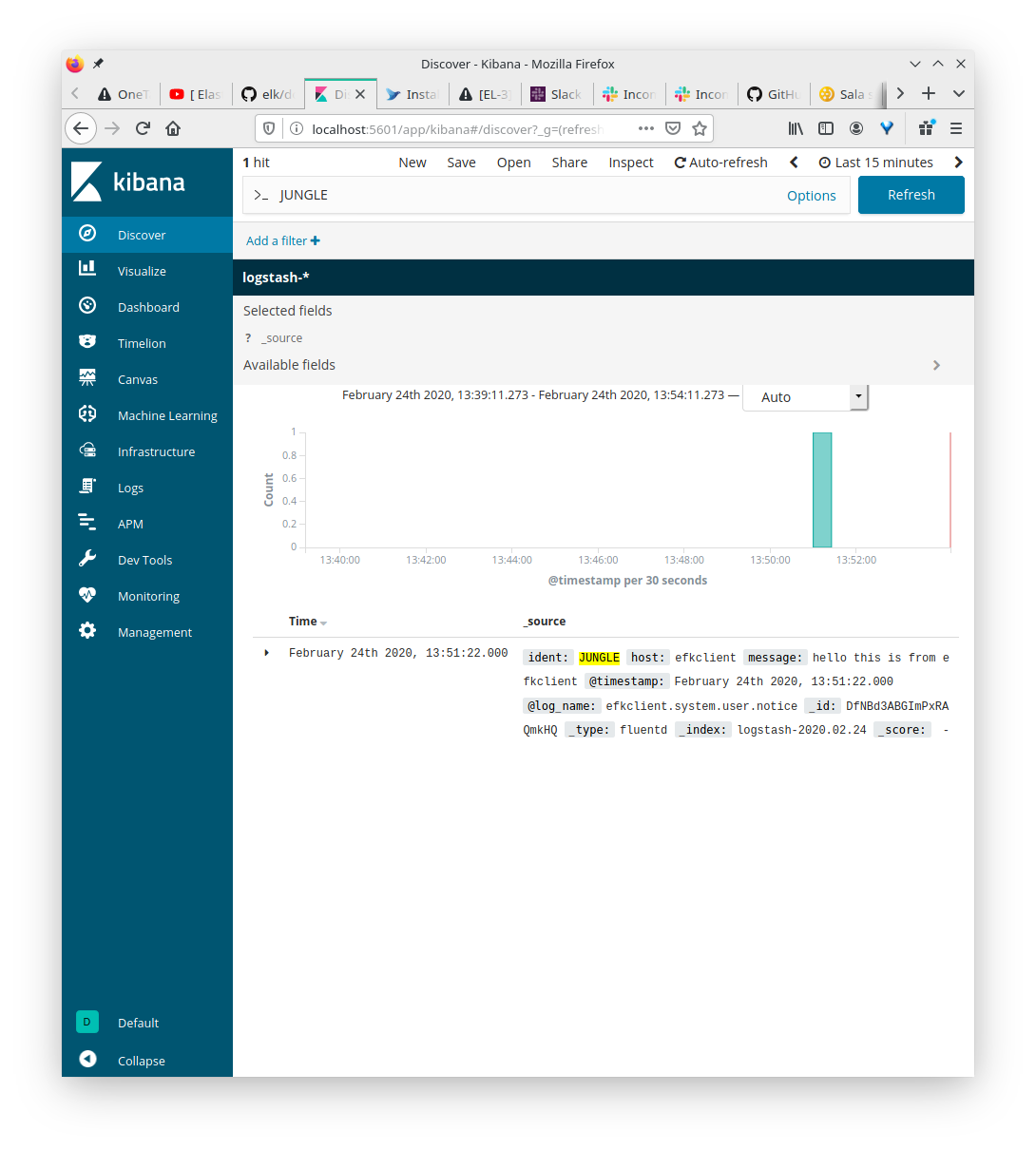


Rysunek 12

Wygeneruję teraz jeden log i sprawdzę czy pojawił się on w zakładce Discover w Kibana.

**[root@efkclient ~]#** *logger -t JUNGLE hello this is from efkclient*

**[root@efkclient ~]#** *tail /var/log/messages*   
Feb 24 12:51:22 efkclient JUNGLE: hello this is from efkclient



Rysunek 13

Sukces! Log jest widoczny w panelu webowym Kibana.

# Rozdział 7

Na skonfigurowanych wcześniej kontenerach Docker ElasticSearch i Kibana oraz maszynie wirtualnej CentOS 7 zaprezentuję monitorowanie metryk oraz logów systemowych.

Do poprawnego działania wtyczki należy sprawdzić czy wszystkie wymagane zależności zostały zainstalowane. Na tapet należy wziąć ingest-geoip. Zazwyczaj ta funkcjonalność jest zwykle domyślnie już zainstalowana w kontenerze z ElasticSearch jednak zawsze należy się upewnić.

**[szymon@LinuxManjaro** **docker]$** *docker-compose exec elasticsearch ls* /usr/share/elasticsearch/plugins  
**[szymon@LinuxManjaro** **docker]$** *docker-compose exec elasticsearch /usr/share/elasticsearch/bin/elasticsearch-plugin install --batch ingest-geoip*

Należy pamiętać również aby albo dodać odpowiednią zasadę na zaporze sieciowej albo do celów testowych zupełnie wyłączyć deamon firewall.

**[vagrant@centosvm01 ~]$** *sudo systemctl status firewalld*

● firewalld.service - firewalld - dynamic firewall daemon

Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)

Active: inactive (dead)

Docs: man:firewalld(1)

Na klienckiej maszynie wirtualnej CentOS 7 muszę pobrać oraz zainstalować usługę Filebeat.

**[vagrant@centosvm01 ~]$** *curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.1.1-x86\_64.rpm*

**[vagrant@centosvm01 ~]$** *sudo rpm -vi filebeat-7.1.1-x86\_64.rpm*

Po pobraniu i zainstalowaniu należy wyedytować konfigurację aby wskazać adres IP serwera na który FileBeat ma przesyłać logi. Edytuję sekcję Kibana oraz ElasticSearch.

**[vagrant@centosvm01 ~]$** *sudo vi /etc/filebeat/filebeat.yml*

#============================== Kibana =====================================

# Kibana Host

# Scheme and port can be left out and will be set to the default (http and 5601)

~~#host: "localhost:5601"~~

**host: "10.3.168.127:5601"**

#-------------------------- Elasticsearch output ------------------------------

output.elasticsearch:

# Array of hosts to connect to.

**hosts: ["10.3.168.127:9200"]**

~~hosts: ["localhost:9200"]~~

Należy również zaktywować i konfigurować moduł odpowiadający za logi systemowe.  
**[vagrant@centosvm01 ~]$** *sudo filebeat modules help*

Manage configured modules

Usage:

filebeat modules [command]

Available Commands:

disable Disable one or more given modules

enable Enable one or more given modules

list List modules

[vagrant@centosvm01 ~]$ sudo filebeat modules list   
Enabled:   
Disabled:   
apache   
auditd   
elasticsearch   
haproxy   
icinga   
iis   
iptables   
kafka   
kibana   
logstash   
mongodb   
mysql   
nginx   
osquery   
postgresql   
redis   
santa   
suricata   
system   
traefik   
zeek

**[vagrant@centosvm01 ~]$** *sudo filebeat modules enable system*   
Enabled system

Testuję czy konfiguracja jest poprawna.

**[vagrant@centosvm01 ~]$** *sudo filebeat test config*   
Config OK

Sprawdzam czy FileBeat widzi ElasticSearch oraz Kibana.

**[vagrant@centosvm01 ~]$** *sudo filebeat test output*

elasticsearch: http://10.3.168.127:9200...

parse url... OK

connection...

parse host... OK

dns lookup... OK

addresses: 10.3.168.127

dial up... OK

TLS... WARN secure connection disabled

talk to server... OK

Version: 7.1.1

Jeżeli wszystko jest okej mogę dokonywać setup.

**[vagrant@centosvm01 ~]$** *sudo filebeat setup*   
Index setup complete.   
Loading dashboards (Kibana must be running and reachable)   
Loaded dashboards   
Loaded machine learning job configurations

**[vagrant@centosvm01 ~]$** *sudo systemctl enable filebeat*   
Created symlink from /etc/systemd/system/multi-user.target.wants/filebeat.service to /usr/lib/systemd/system/filebeat.service.

**[vagrant@centosvm01 ~]$** *sudo systemctl start filebeat*

Per analogia instaluję MetricBeat.

**[vagrant@centosvm01 ~]$** *sudo vi /etc/metricbeat/modules.d/system.yml*

# Module: system

# Docs: https://www.elastic.co/guide/en/beats/metricbeat/7.1/metricbeat-module-system.html

- module: system

period: 10s

metricsets:

- cpu

- load

- memory

- network

- process

- process\_summary

- socket\_summary

#- core

#- diskio

#- socket

process.include\_top\_n:

by\_cpu: 5 # include top 5 processes by CPU

by\_memory: 5 # include top 5 processes by memory

- module: system

period: 1m

metricsets:

- filesystem

- fsstat

processors:

- drop\_event.when.regexp:

system.filesystem.mount\_point: '^/(sys|cgroup|proc|dev|etc|host|lib)($|/)'

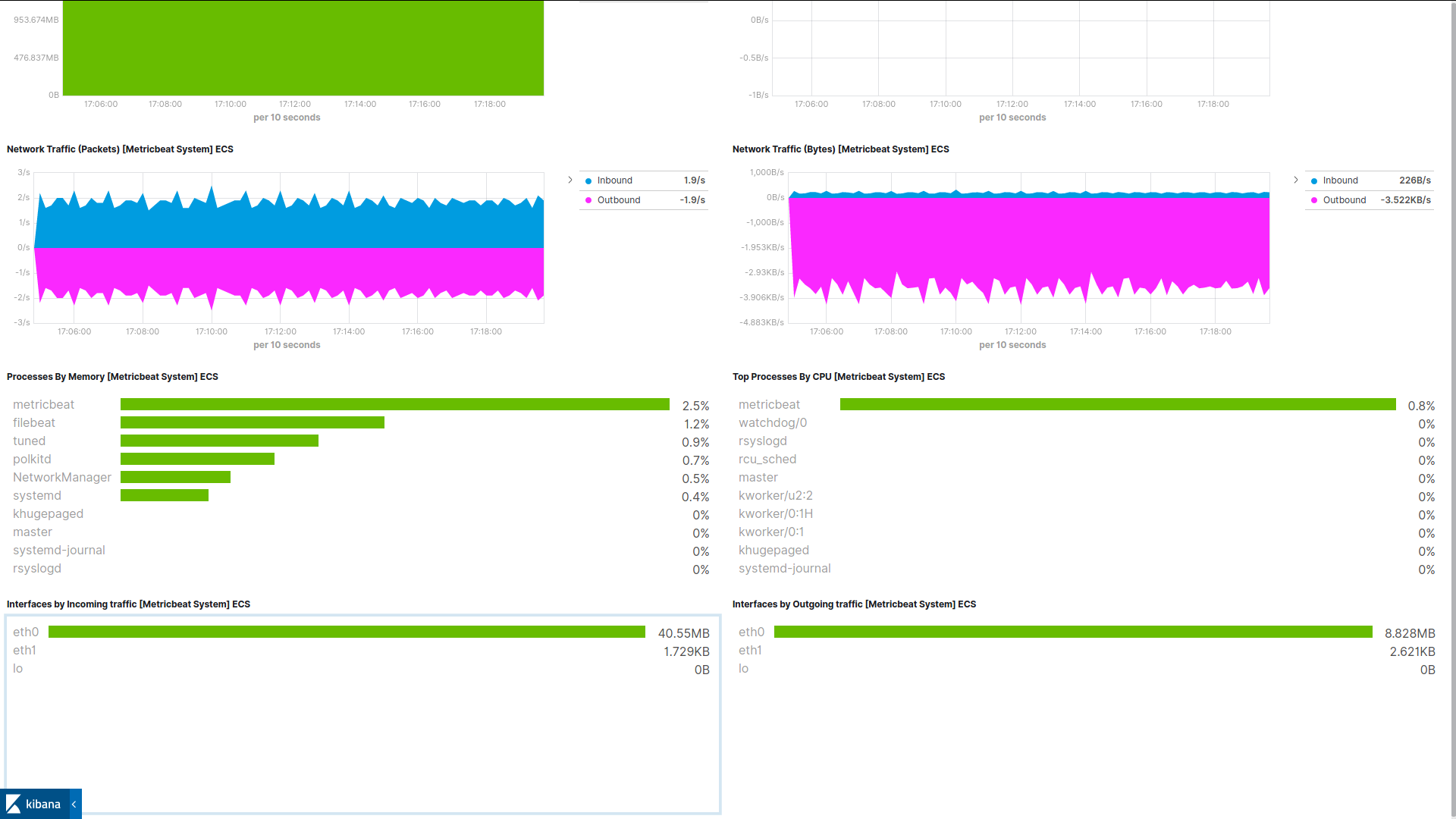
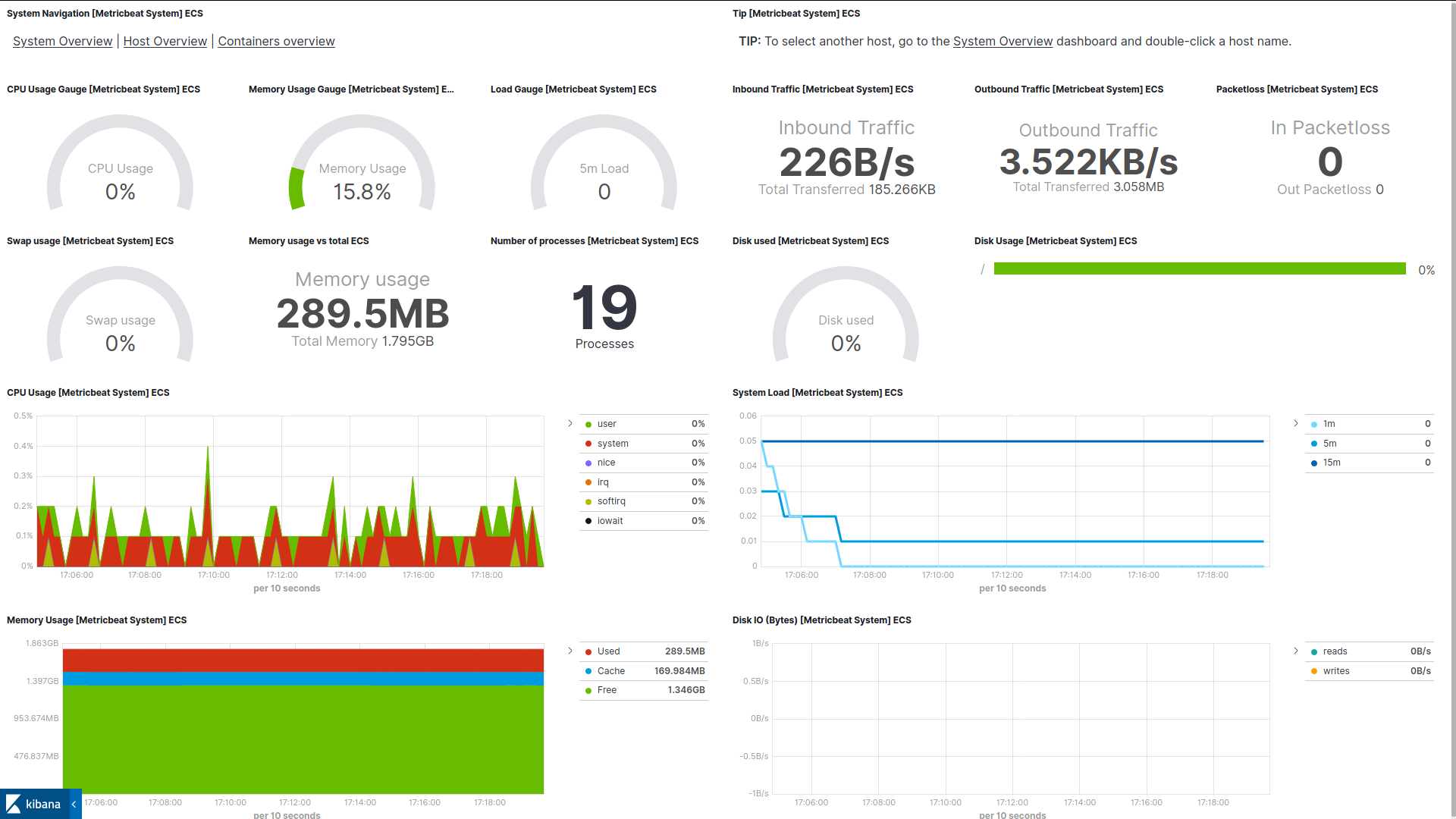
- module: system

period: 15m

metricsets:

- uptime

MetricBeat jest tak skonfigurowane aby co 10 sekund wysyłało informację o obciążeniu procesora, zużyciu pamięci RAM, ruchu sieciowym. Co minutę analizowany jest system plików oraz statystyka plików. Co 15 minut aktualizowany jest moduł uptime. Prezentowane są również po 5 aplikacji, które najbardziej obciążają procesor oraz kolejne 5, które najbardziej obciążają pamięć RAM.



Rysunek 14 [Metricbeat System] Host overview ECS

Przekazywanie i przetwarzanie metryk i logów Apache.

Wykorzystuję do tego usługę FileBeat, którą już wcześniej zainstalowałem.

Teraz przede mną instalacja usługi Apache a potem wystarczy tylko skonfigurować i uruchomić odpowiedni moduł dedykowany do jej obsługi obsługi.

**[vagrant@centosvm01 ~]$** *sudo filebeat modules enable apache*   
Enabled apache  
**[vagrant@centosvm01 ~]$** *sudo vi /etc/filebeat/modules.d/apache.yml*

# Module: apache

# Docs: https://www.elastic.co/guide/en/beats/filebeat/7.1/filebeat-module-apache.html

- module: apache

# Access logs

access:

enabled: true

# Set custom paths for the log files. If left empty,

# Filebeat will choose the paths depending on your OS.

~~#var.paths:~~

**var.paths: ["/var/log/httpd/access\_log"]**

# Error logs

error:

enabled: true

# Set custom paths for the log files. If left empty,

# Filebeat will choose the paths depending on your OS.

~~#var.paths:~~

**var.paths: ["/var/log/httpd/error\_log"]**

**[vagrant@centosvm01 ~]$** *sudo yum install httpd*  
Loaded plugins: fastestmirror   
Determining fastest mirrors   
\* base: ftp.icm.edu.pl   
\* extras: ftp.icm.edu.pl   
\* updates: ftp.icm.edu.pl   
base                                             | 3.6 kB     00:00        
extras                                           | 2.9 kB     00:00        
updates                                          | 2.9 kB     00:00      

(1/4): base/7/x86\_64/group\_gz                      | 165 kB   00:00        
(2/4): extras/7/x86\_64/primary\_db                  | 159 kB   00:00        
(3/4): base/7/x86\_64/primary\_db                    | 6.0 MB   00:01        
(4/4): updates/7/x86\_64/primary\_db                 | 6.7 MB   00:01        
Resolving Dependencies   
--> Running transaction check   
---> Package httpd.x86\_64 0:2.4.6-90.el7.centos will be installed   
--> Processing Dependency: httpd-tools = 2.4.6-90.el7.centos for package  
: httpd-2.4.6-90.el7.centos.x86\_64   
--> Processing Dependency: system-logos >= 7.92.1-1 for package: httpd-2  
.4.6-90.el7.centos.x86\_64   
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.6-90.e  
l7.centos.x86\_64   
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd  
-2.4.6-90.el7.centos.x86\_64   
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4  
.6-90.el7.centos.x86\_64   
--> Running transaction check   
---> Package apr.x86\_64 0:1.4.8-5.el7 will be installed   
---> Package apr-util.x86\_64 0:1.5.2-6.el7 will be installed   
---> Package centos-logos.noarch 0:70.0.6-3.el7.centos will be installed   
---> Package httpd-tools.x86\_64 0:2.4.6-90.el7.centos will be installed   
---> Package mailcap.noarch 0:2.1.41-2.el7 will be installed   
--> Finished Dependency Resolution   
  
Dependencies Resolved   
========================================================================   
Package           Arch        Version                  Repository Size   
========================================================================   
Installing:   
httpd             x86\_64      2.4.6-90.el7.centos      base      2.7 M   
Installing for dependencies:   
apr               x86\_64      1.4.8-5.el7              base      103 k   
apr-util          x86\_64      1.5.2-6.el7              base       92 k   
centos-logos      noarch      70.0.6-3.el7.centos      base       21 M   
httpd-tools       x86\_64      2.4.6-90.el7.centos      base       91 k   
mailcap           noarch      2.1.41-2.el7             base       31 k   
  
Transaction Summary   
========================================================================   
Install  1 Package (+5 Dependent packages)   
  
Total download size: 24 M   
Installed size: 31 M

Downloading packages:   
warning: /var/cache/yum/x86\_64/7/base/packages/apr-1.4.8-5.el7.x86\_64.rp  
m: Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY   
Public key for apr-1.4.8-5.el7.x86\_64.rpm is not installed   
(1/6): apr-1.4.8-5.el7.x86\_64.rpm                  | 103 kB   00:00        
(2/6): apr-util-1.5.2-6.el7.x86\_64.rpm             |  92 kB   00:00        
(3/6): httpd-tools-2.4.6-90.el7.centos.x86\_64.rpm  |  91 kB   00:00        
(4/6): mailcap-2.1.41-2.el7.noarch.rpm             |  31 kB   00:00        
(5/6): httpd-2.4.6-90.el7.centos.x86\_64.rpm        | 2.7 MB   00:02        
(6/6): centos-logos-70.0.6-3.el7.centos.noarch.rpm |  21 MB   00:07        
------------------------------------------------------------------------   
Total                                      3.4 MB/s |  24 MB  00:07        
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7   
Importing GPG key 0xF4A80EB5:   
Userid     : "CentOS-7 Key (CentOS 7 Official Signing Key) <security@centos.org>"   
Fingerprint: 6341 ab27 53d7 8a78 a7c2 7bb1 24c6 a8a7 f4a8 0eb5   
Package    : centos-release-7-6.1810.2.el7.centos.x86\_64 (@anaconda)   
From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7   
Is this ok [y/N]: y   
Running transaction check   
Running transaction test   
Transaction test succeeded   
Running transaction   
Warning: RPMDB altered outside of yum.   
 Installing : apr-1.4.8-5.el7.x86\_64                               1/6    
 Installing : apr-util-1.5.2-6.el7.x86\_64                          2/6    
 Installing : httpd-tools-2.4.6-90.el7.centos.x86\_64               3/6    
 Installing : centos-logos-70.0.6-3.el7.centos.noarch              4/6    
 Installing : mailcap-2.1.41-2.el7.noarch                          5/6    
 Installing : httpd-2.4.6-90.el7.centos.x86\_64                     6/6    
 Verifying  : mailcap-2.1.41-2.el7.noarch                          1/6    
 Verifying  : httpd-tools-2.4.6-90.el7.centos.x86\_64               2/6    
 Verifying  : apr-util-1.5.2-6.el7.x86\_64                          3/6    
 Verifying  : httpd-2.4.6-90.el7.centos.x86\_64                     4/6    
 Verifying  : apr-1.4.8-5.el7.x86\_64                               5/6    
 Verifying  : centos-logos-70.0.6-3.el7.centos.noarch              6/6    
  
Installed:   
 httpd.x86\_64 0:2.4.6-90.el7.centos                                       
  
Dependency Installed:   
 apr.x86\_64 0:1.4.8-5.el7                                                 
 apr-util.x86\_64 0:1.5.2-6.el7                                            
 centos-logos.noarch 0:70.0.6-3.el7.centos                                
 httpd-tools.x86\_64 0:2.4.6-90.el7.centos                                 
 mailcap.noarch 0:2.1.41-2.el7                                            
  
Complete!

Utworzę teraz szybką stronę internetową w pliku index.html i postawię ją na utworzonym przed chwilą serwerze Apache.

**[vagrant@centosvm01 ~]$** *sudo su -*

**[root@centosvm01 ~]#** *echo "<h1>Just me and Open Source</h1>" > /var/www/html/index.html*

**[root@centosvm01 ~]#** *systemctl enable httpd*   
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.s  
ervice to /usr/lib/systemd/system/httpd.service.   
**[root@centosvm01 ~]#** *systemctl start httpd*

**[vagrant@centosvm01 ~]$** *sudo filebeat setup*  
Index setup complete.   
Loading dashboards (Kibana must be running and reachable)   
Loaded dashboards

Rysunek 15 Loaded machine learning job configurations

Mogę przeanalizować z jakiej lokalizacji była wyszukiwana moja strona internetowa oraz z jakich przeglądarek korzystali użytkownicy. Ile było poprawnych wyszukań, ile było wyszukań błędnych. Jakie numery odpowiedzi / błędów były generowane przez serwer webowy Apache.

Korzystając z tego, że MetricBeat jest już zainstalowane edytuje tylko aktywowane moduły i dodaje do nich Apache.

**[vagrant@centosvm01 ~]$** *sudo metricbeat modules list*

Enabled:   
apache   
system   
  
Disabled:   
aerospike   
aws   
ceph   
couchbase   
couchdb   
docker   
dropwizard   
elasticsearch   
envoyproxy   
etcd   
golang   
graphite   
haproxy   
http   
jolokia   
kafka   
kibana   
kubernetes   
kvm   
logstash   
memcached   
mongodb   
mssql   
munin   
mysql   
nats   
nginx   
php\_fpm   
postgresql   
prometheus   
rabbitmq   
redis   
traefik   
uwsgi   
vsphere   
windows   
zookeeper

**[vagrant@centosvm01 ~]$** *sudo vi /etc/metricbeat/modules.d/apache.yml*

# Module: apache

# Docs: https://www.elastic.co/guide/en/beats/metricbeat/7.1/metricbeat-module-apache.html

- module: apache

~~#metricsets:~~

**metricsets:**

~~# - status~~

**- status**

period: 10s

hosts: ["http://127.0.0.1"]

#username: "user"

#password: "secret"

Co 10 sekund będzie sprawdzane czy strona internetowa jest aktywna. Natomiast aby nadzorować status bardziej dokładnie należy odkomentować metricsets i wprowadzić również zmiany w konfiguracji Apache.

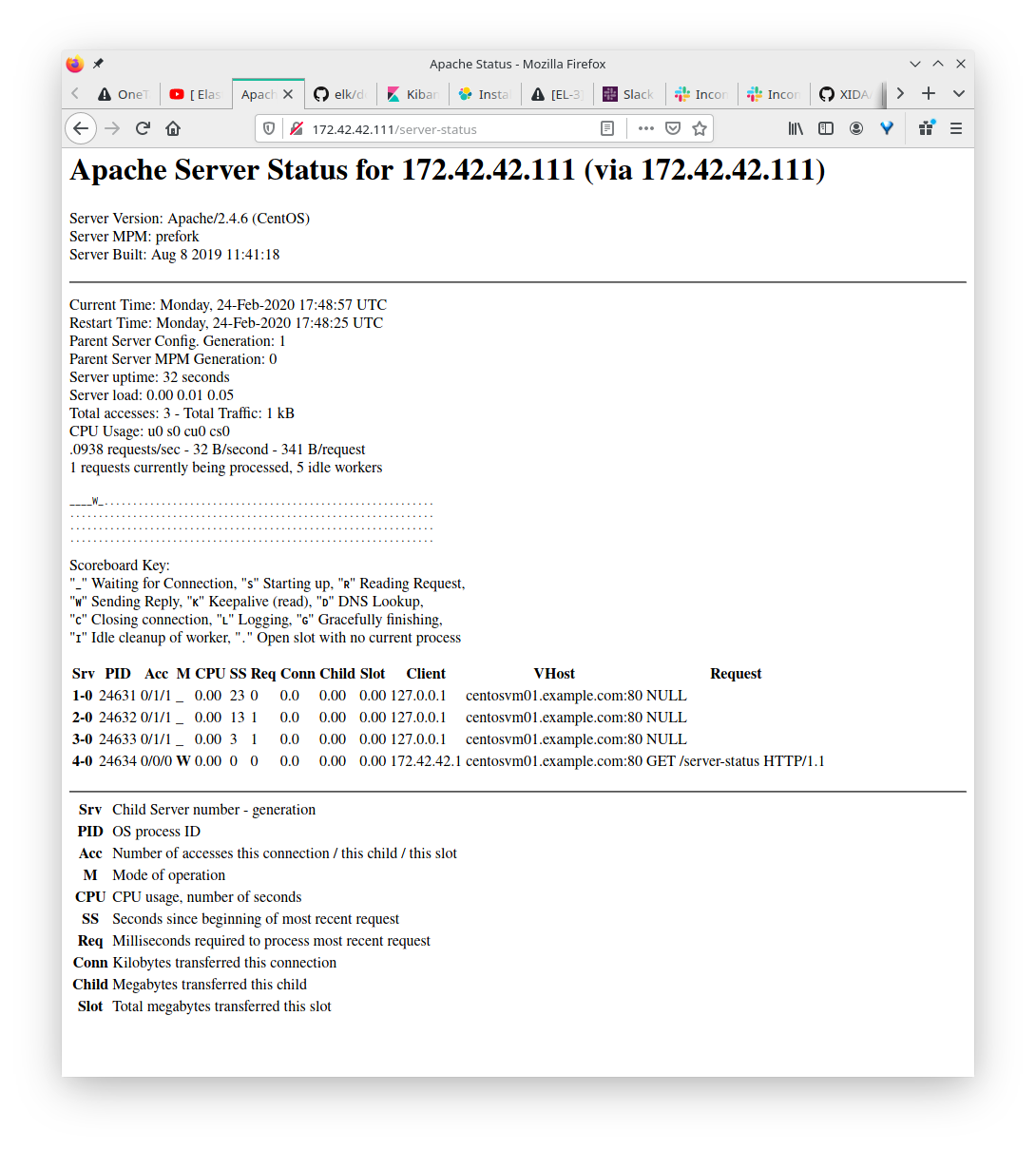
Domyślnie usługa MetricBeat sprawdza podstronę http://localhost/server-status której obecnie nie ma i należy ją utworzyć.

**[vagrant@centosvm01 ~]$** *sudo su -*   
Last login: Mon Feb 24 16:51:17 UTC 2020 on pts/0   
**[root@centosvm01 ~]#** *vi /etc/httpd/conf.modules.d/00-base.conf*

LoadModule status\_module modules/mod\_status.so  
**[root@centosvm01 ~]#** *vi /etc/httpd/conf/httpd.conf*

Idę do końca pliku i dodaję trzy linijki konfiguracji.

<Location /server-status>   
 SetHandler server-status   
</Location>  
[**root@centosvm01 ~]#** *systemctl restart httpd*



Rysunek 16



Rysunek 17

# Rozdział 8

Monitorowanie usług przy użyciu HeartBeat, AuditBeat oraz PacketBeat. Wszystkie usługi będą zainstalowane na klienckiej maszynie wirtualnej Ubuntu. Pobieram i instaluję klucz publiczny aplikacji HeartBeat.

**vagrant@ubuntuvm01**:**~**$ *wget -qO - https://artifacts.elastic.co/GPG-KEY-el  
asticsearch | sudo apt-key add -*   
OK

Instaluje dodatkowe wymagane zależności.

**vagrant@ubuntuvm01**:**~**$ *sudo apt-get install apt-transport-https*

Dodaję definicję repozytorium.

**vagrant@ubuntuvm01**:**~**$ *echo "deb https://artifacts.elastic.co/packages/7.  
x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-7.x.list*

Aktualizuję repozytorium i instaluję Heartbeat.

**vagrant@ubuntuvm01**:**~**$ *sudo apt-get update && sudo apt-get install heartbeat-elastic*

W pliku konfiguracyjnym /etc/heartbeat/heartbeat.yml edytuję odpowiednie sekcje aby poinformować usługę gdzie znajduje się ElasticSearch oraz gdzie należy szukać Kibana.

Następnie konfiguruję monitory Heartbeat.

**vagrant@ubuntuvm01**:**~**$ *cd /etc/heartbeat/monitors.d/ && ls*   
sample.http.yml.disabled  sample.tcp.yml.disabled   
sample.icmp.yml.disabled

Producent oprogramowania daje trzy przykładowe pliki na których można wzorować się przy konfigurowaniu własnego monitora.

**vagrant@ubuntuvm01**:**/etc/heartbeat/monitors.d**$ *sudo vi tcpdemo.yml*

- type: tcp

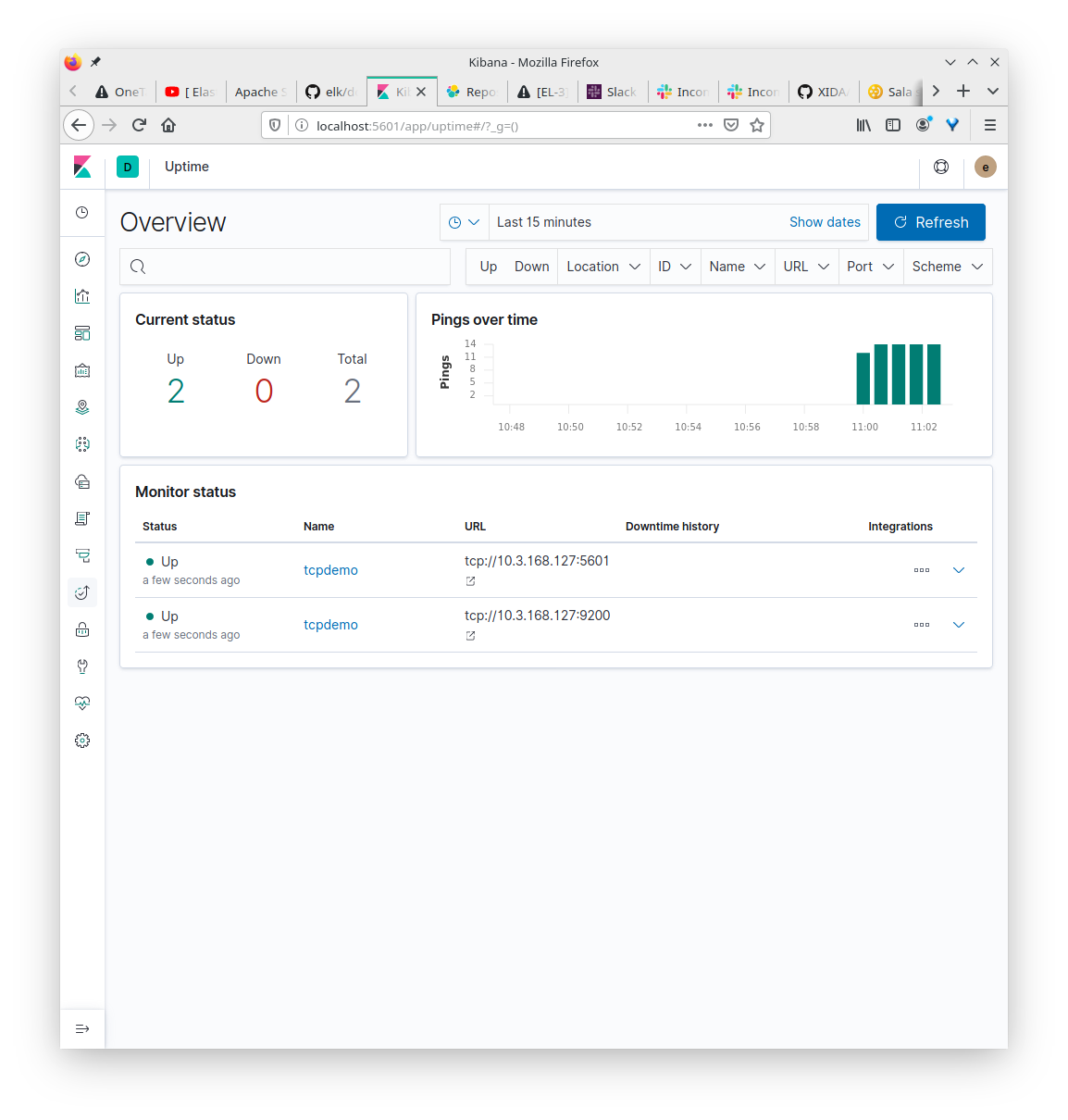
name: tcpdemo

enabled: true

schedule: "@every 5s"

hosts: ["10.3.168.127"]

ports: [9200, 5601]



Rysunek 18

Ten pulpit prezentuje że są monitorowane dwie usługi. Obydwie pochodzą z monitora, który utworzyłem „tcpdemo”. Więc ta usługa działa tak jakby sprawdzała bicie serca, czy wszystkie usługi monitorujące, które ustawił administrator są aktywne.

Następnie modyfikuje monitor oraz dodaje następujące linijki konfiguracji.

- type: tcp

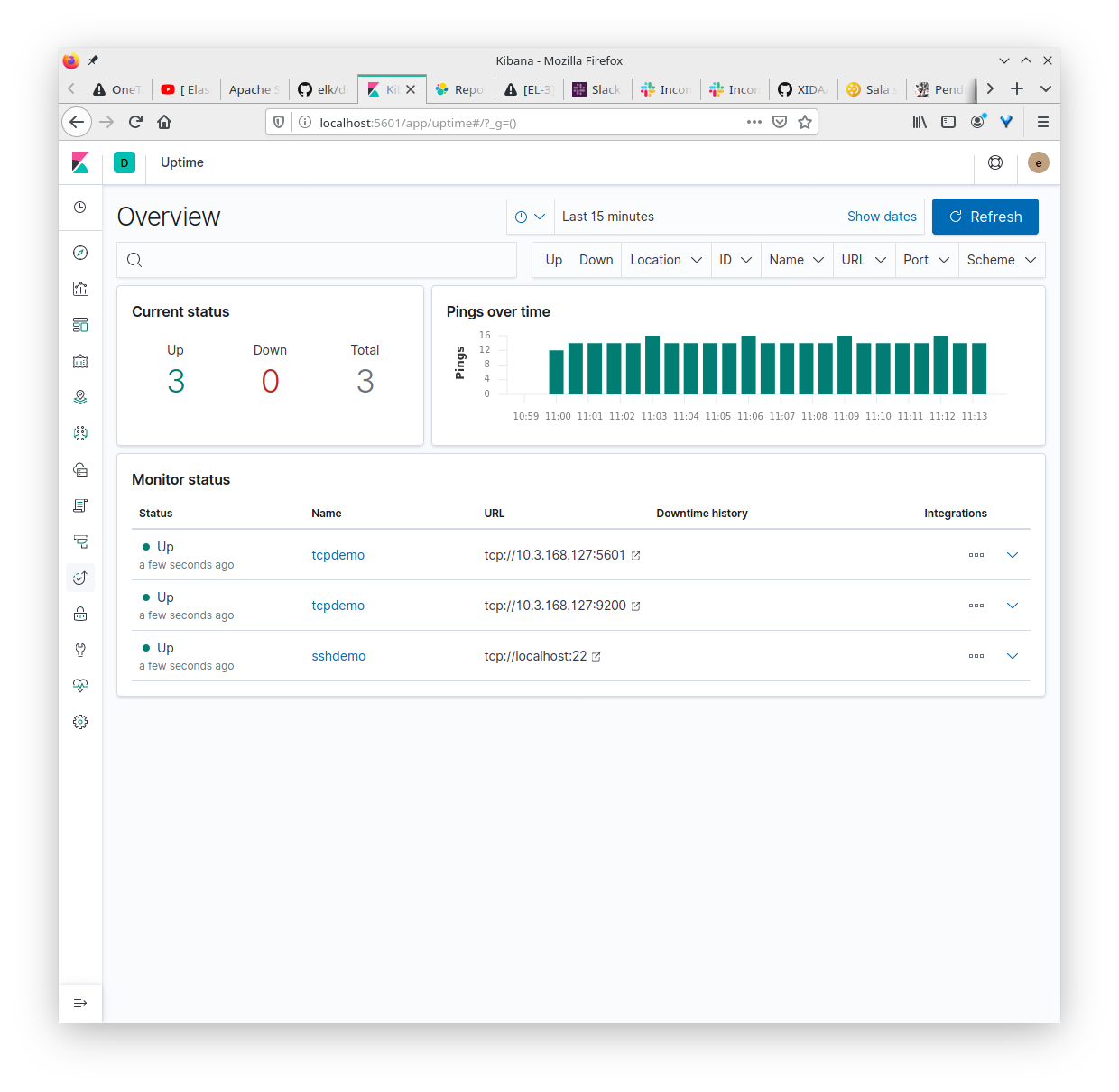
name: sshdemo

enabled: true

schedule: "@every 5s"

hosts: ["localhost"]

ports: [22]

Kolejna edycja pliku konfiguracyjnego i kolejne dodane linijki. Oczywiście kolejne monitory można dodawać również w pojedynczych plikach ale mnie jest wygodniej w jednym.Teraz przechodzę do prezentacji usługi Auditbeat.

Rysunek 19

Muszę upewnić się czy mam zainstalowaną i aktywnie działającą usługę auditd.

**vagrant@ubuntuvm01**:**~**$ *sudo systemctl status auditd*

Sprawdzić czy na maszynie wirtualnej znajduję się folder /var/log/audit

**vagrant@ubuntuvm01**:**~**$ *sudo ls /var/log/audit/audit.log*   
/var/log/audit/audit.log  
**vagrant@ubuntuvm01**:**~**$ *auditbeat version*   
auditbeat version 7.6.0 (amd64), libbeat 7.6.0 [6a23e8f8f30f5001ba344e4e  
54d8d9cb82cb107c built 2020-02-05 23:03:32 +0000 UTC]  
**vagrant@ubuntuvm01:~$** *auditbeat help*

Usage:

auditbeat [flags]

auditbeat [command]

Available Commands:

enroll Enroll in Kibana for Central Management

export Export current config or index template

help Help about any command

keystore Manage secrets keystore

run Run auditbeat

setup Setup index template, dashboards and ML jobs

show Show modules information

test Test config

version Show current version info

- module: file\_integrity

paths:

- /bin

- /usr/bin

- /sbin

- /usr/sbin

- /etc

- module: system

datasets:

- host # General host information, e.g. uptime, IPs

- login # User logins, logouts, and system boots.

- package # Installed, updated, and removed packages

- process # Started and stopped processes

- socket # Opened and closed sockets

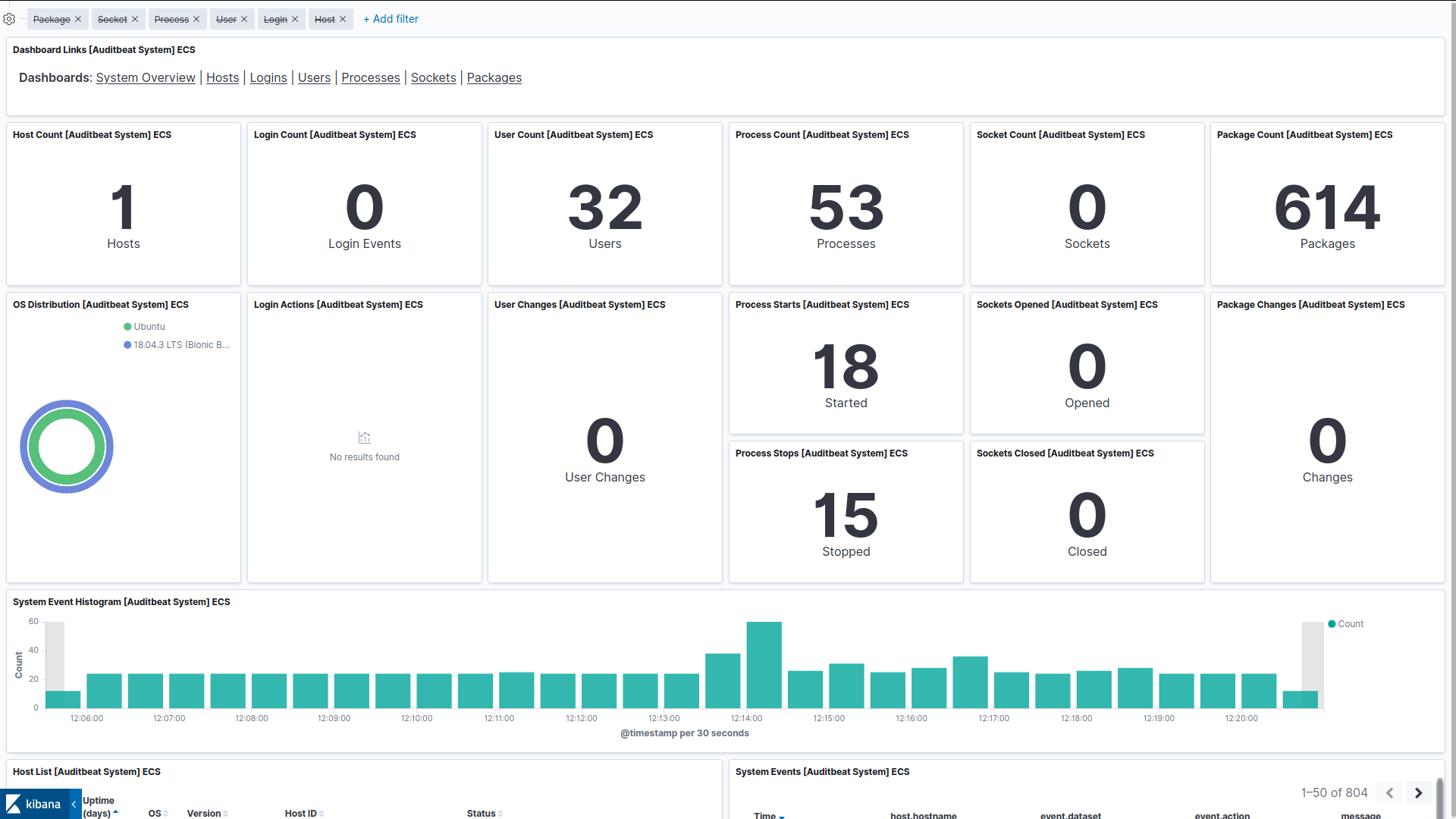
- user # User information

Jest to fragment pliku konfiguracyjnego Auditbeat.yml który dotyczy sprawdzania spójności danych: plików w określonych folderach. Jeżeli są tworzone nowe pliki lub kiedy istniejące pliki są edytowane lub usuwane jest generowany log, który następnie jest możliwy do wychwycenia i przeanalizowania w Kibana. Jak również w pakiecie znajduje się moduł systemowy, który nadzoruje wszystkie procesy, logowania użytkowników, jak długo system jest aktywny, kiedy ostatnio był aktualizowany, restartowany. Jakiekolwiek transakcje związane z instalowaniem nowych usług lub aplikacji. Otwarte i zamknięte gniazda sieciowe (sockety).

W pliku konfiguracyjnym muszę odkometować oraz zmienić wartość na true również linijkę dotyczącą panelu. Bez tej zmiany panel Kibana nie będzie w pełni funkcjonalny w obsłudze Filebeat.

**setup.dashboards.enabled**: true

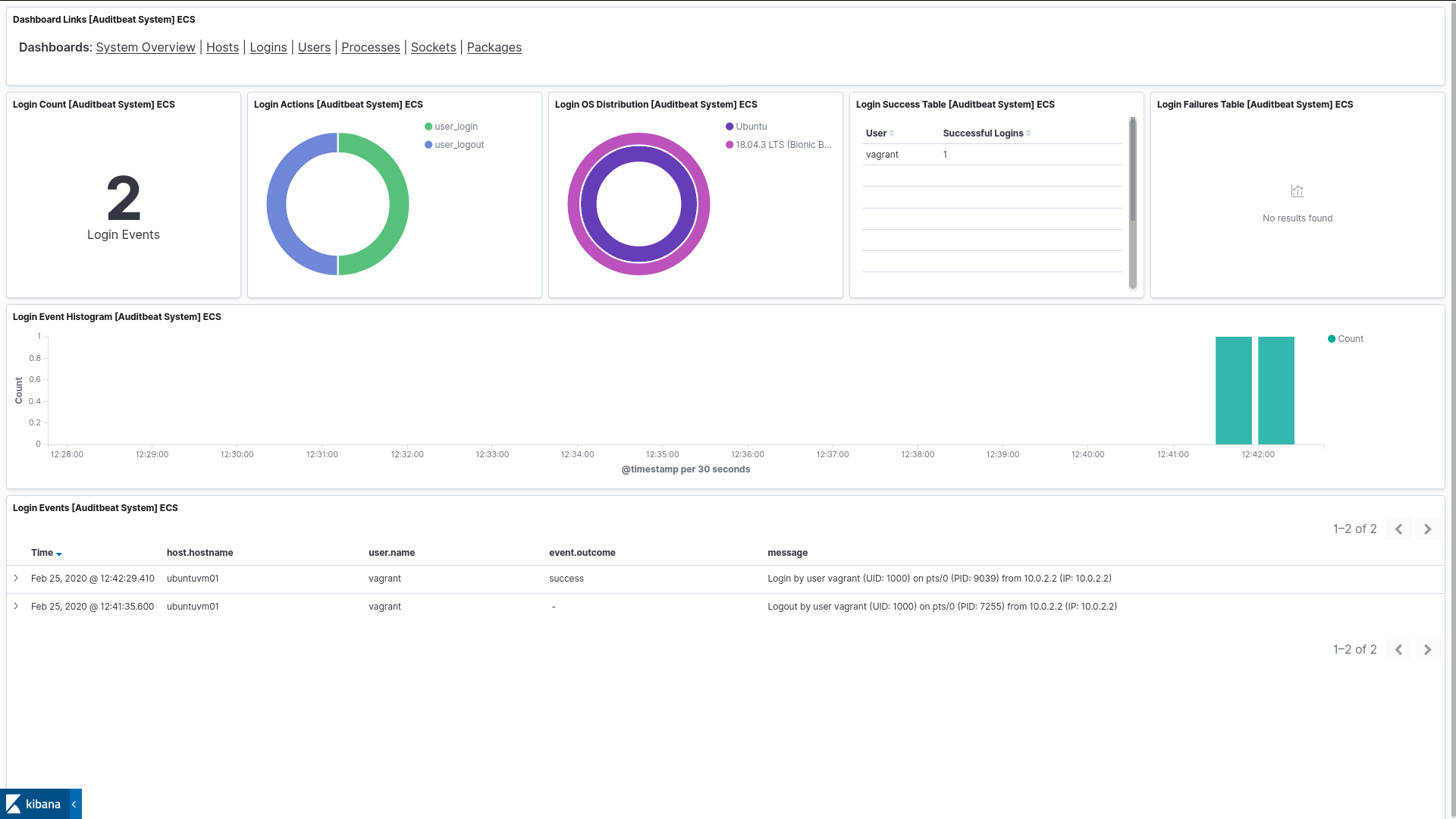
Oczywiście edytuje plik konfiguracyjny również informująć AuditBeat jaki jest adres IP ElasticSearch oraz Kibana.



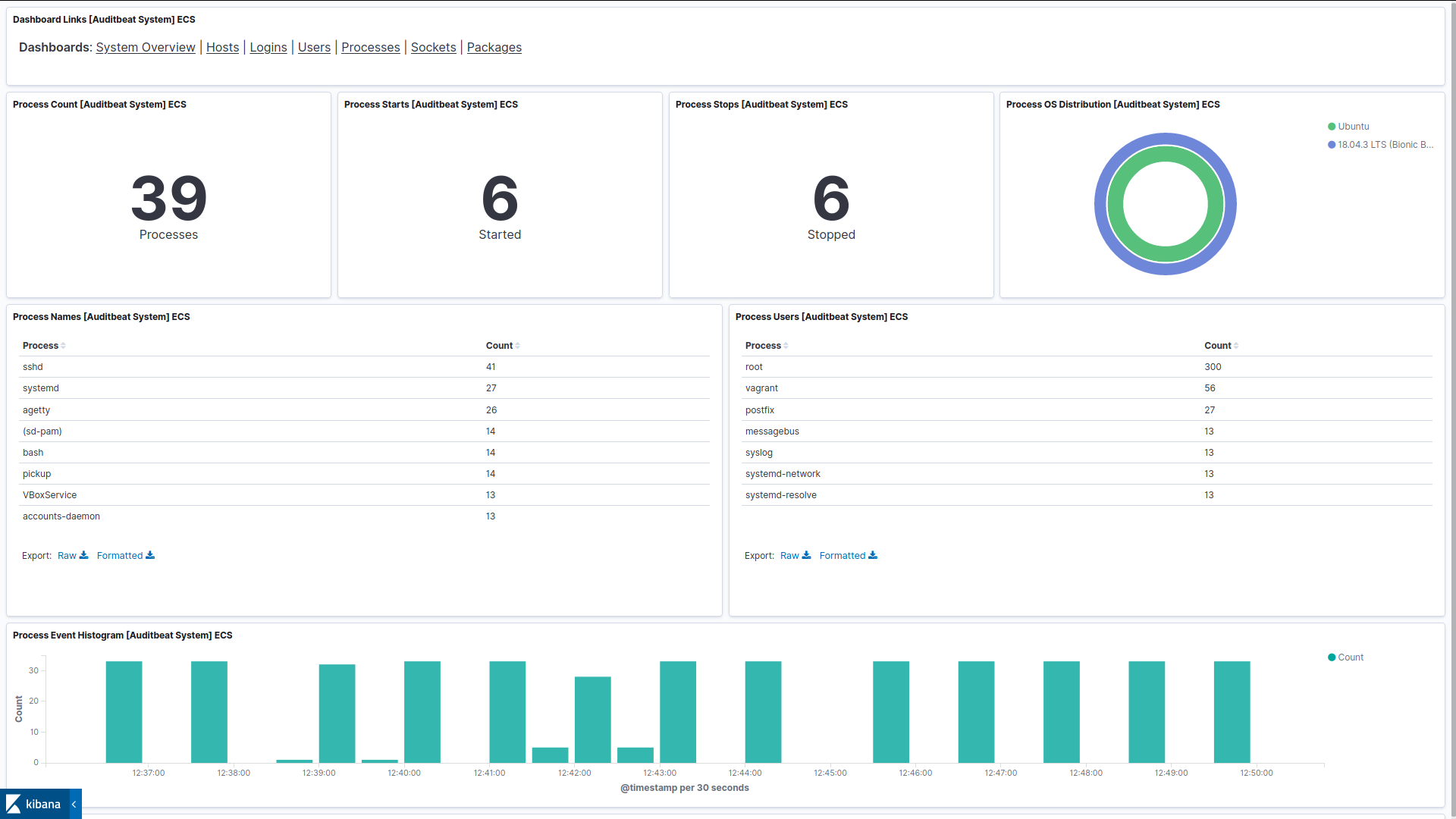
Rysunek 20

Panel pokazuje, że w systemie są 32 użytkowników. Oznacza to tylko nie mniej nie więcej tyle, że większość z usług uruchomianych w systemie operacyjnym ma przypisanego do siebie użytkownika. Są to konta, które pomagają sprawnie zarządzać uprawnieniami danej aplikacji.

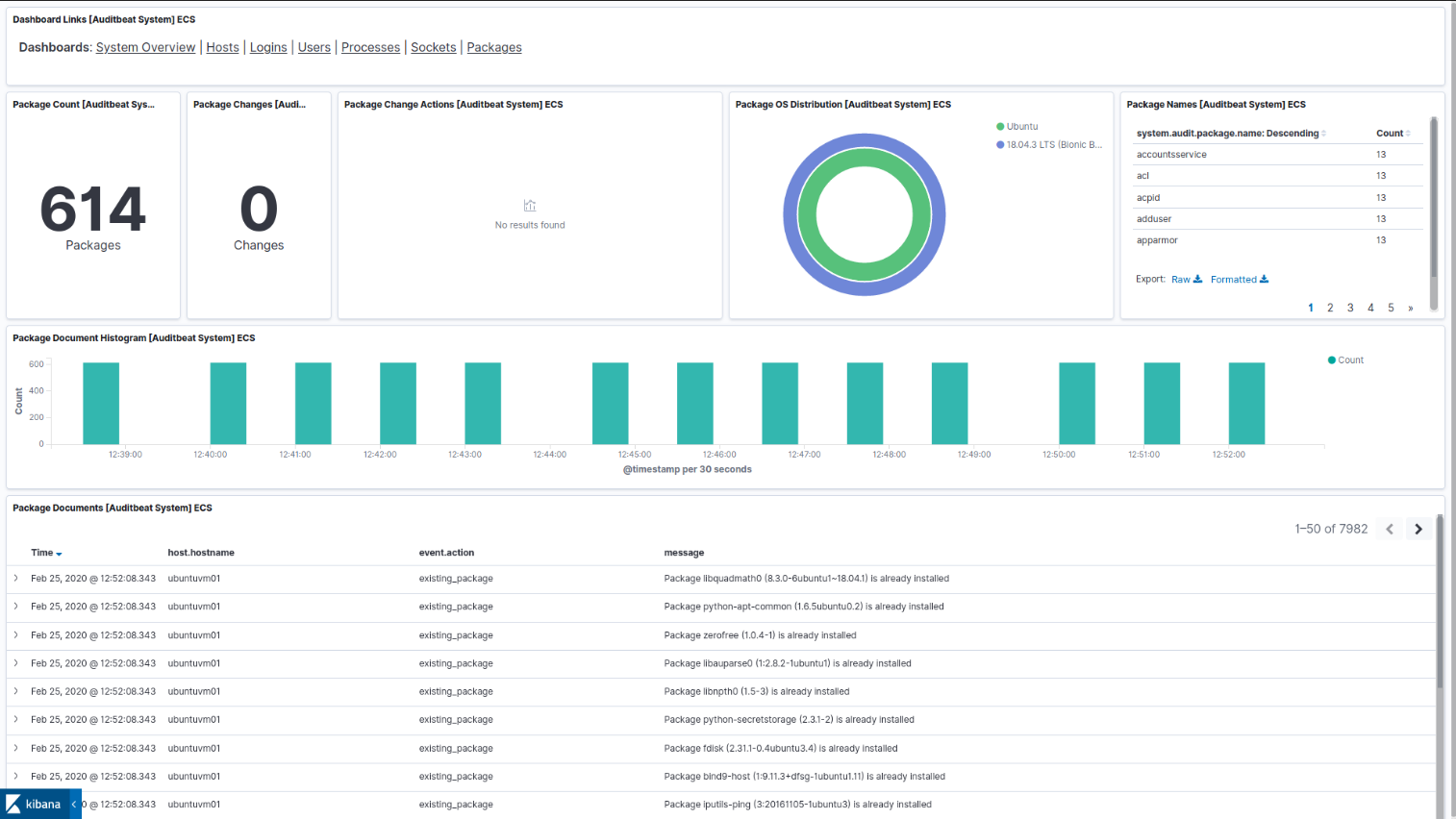
**vagrant@ubuntuvm01**:**~**$ *sudo cat /etc/passwd*   
root:x:0:0:root:/root:/bin/bash   
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin   
bin:x:2:2:bin:/bin:/usr/sbin/nologin   
sys:x:3:3:sys:/dev:/usr/sbin/nologin   
sync:x:4:65534:sync:/bin:/bin/sync   
games:x:5:60:games:/usr/games:/usr/sbin/nologin   
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin   
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin   
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin   
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin   
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin   
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin   
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin   
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin   
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin   
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin   
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin   
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin   
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif: /usr/sbin/nologin   
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve: /usr/sbin/nologin   
syslog:x:102:106::/home/syslog:/usr/sbin/nologin   
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin   
\_apt:x:104:65534::/nonexistent:/usr/sbin/nologin   
lxd:x:105:65534::/var/lib/lxd/:/bin/false   
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin   
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin   
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin   
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin   
pollinate:x:110:1::/var/cache/pollinate:/bin/false   
vagrant:x:1000:1000:,,,:/home/vagrant:/bin/bash   
ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash   
postfix:x:111:117::/var/spool/postfix:/usr/sbin/nologin



Rysunek 21



Rysunek 22

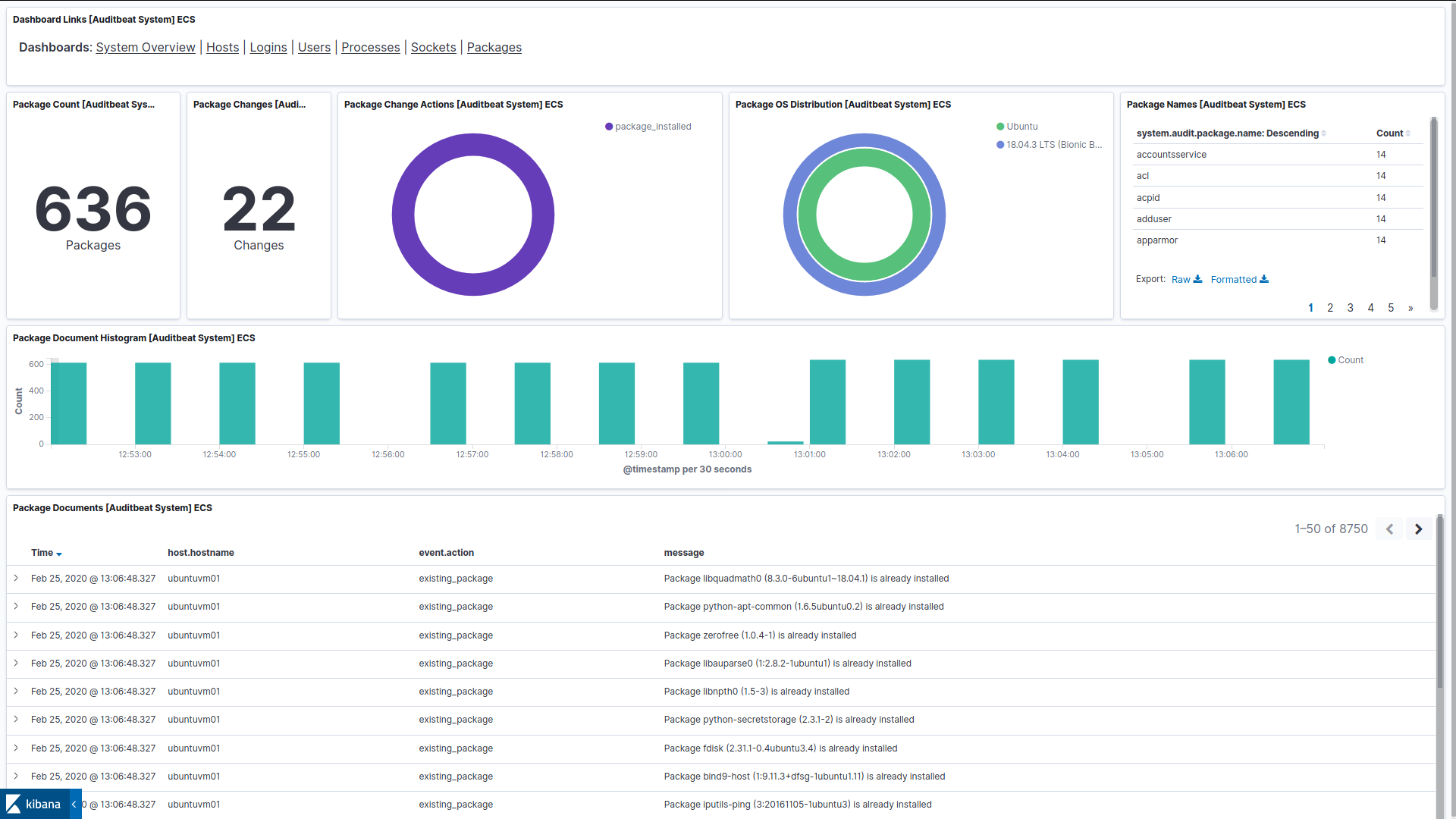


Rysunek 23

Przeprowadzę test AuditBeat i zainstaluję aplikację neovim.

**vagrant@ubuntuvm01:~$** *sudo apt-get install neovim*

Tym jednym poleceniem zainstalowałem wiele zależności, pakietów i aplikacji. A oto jak zmienił się panel Kibana.



Rysunek 24

Według panelu nastąpiło 22 zmian pakietów. Nie ważne czy to było usunięcie, aktualizacja czy dodanie pakietu ale była zmiana. Więcej szczegółów jest do przeanalizowania poniżej.

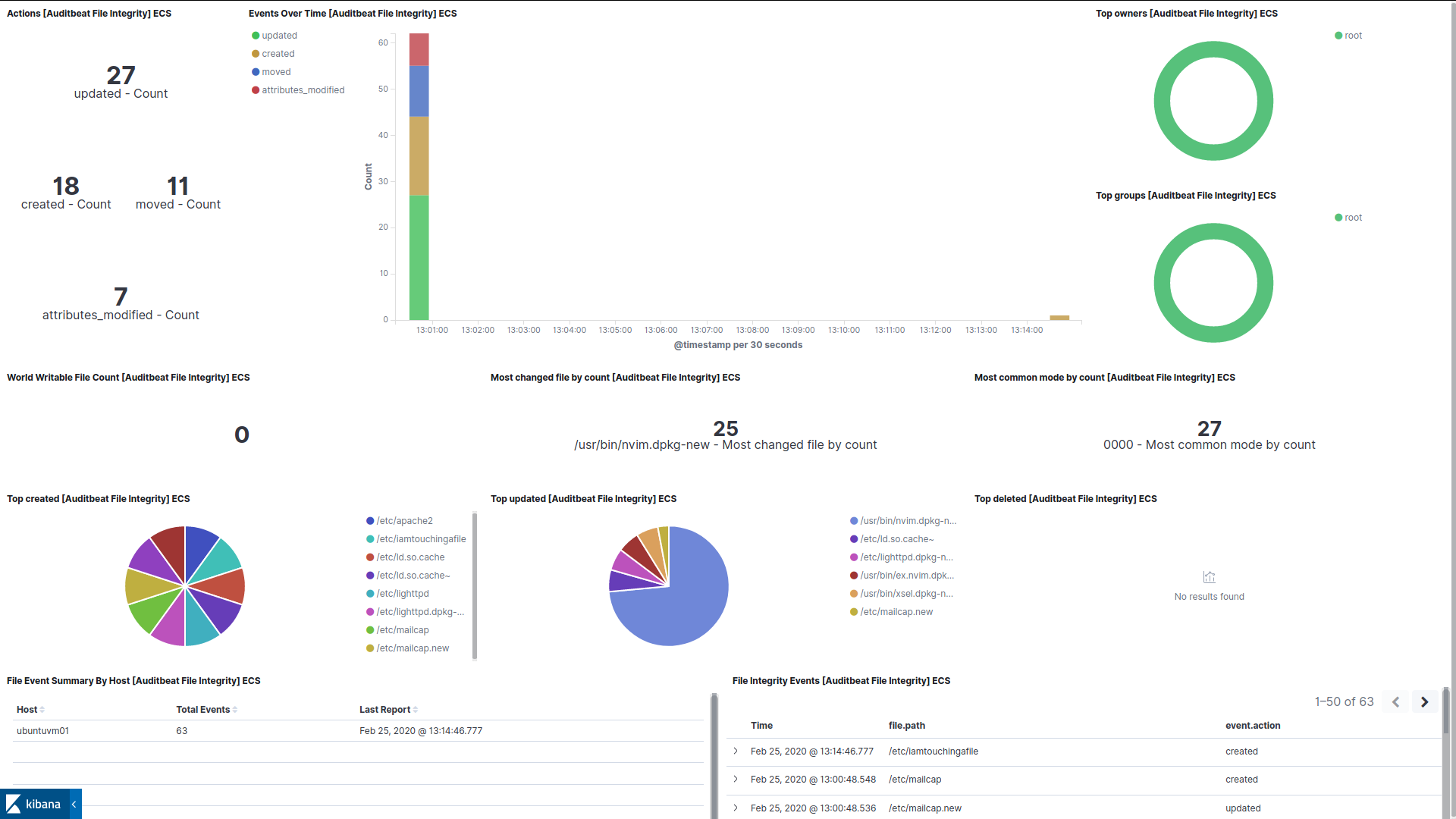
Integralność danych.



Rysunek 25

Utworzę teraz plik w jednej z monitorowanych lokalizacji i zobaczę czy ta zmiana zostanie zanotowana przez system.

**vagrant@ubuntuvm01**:**~**$ *sudo touch /etc/iamtouchingafile*

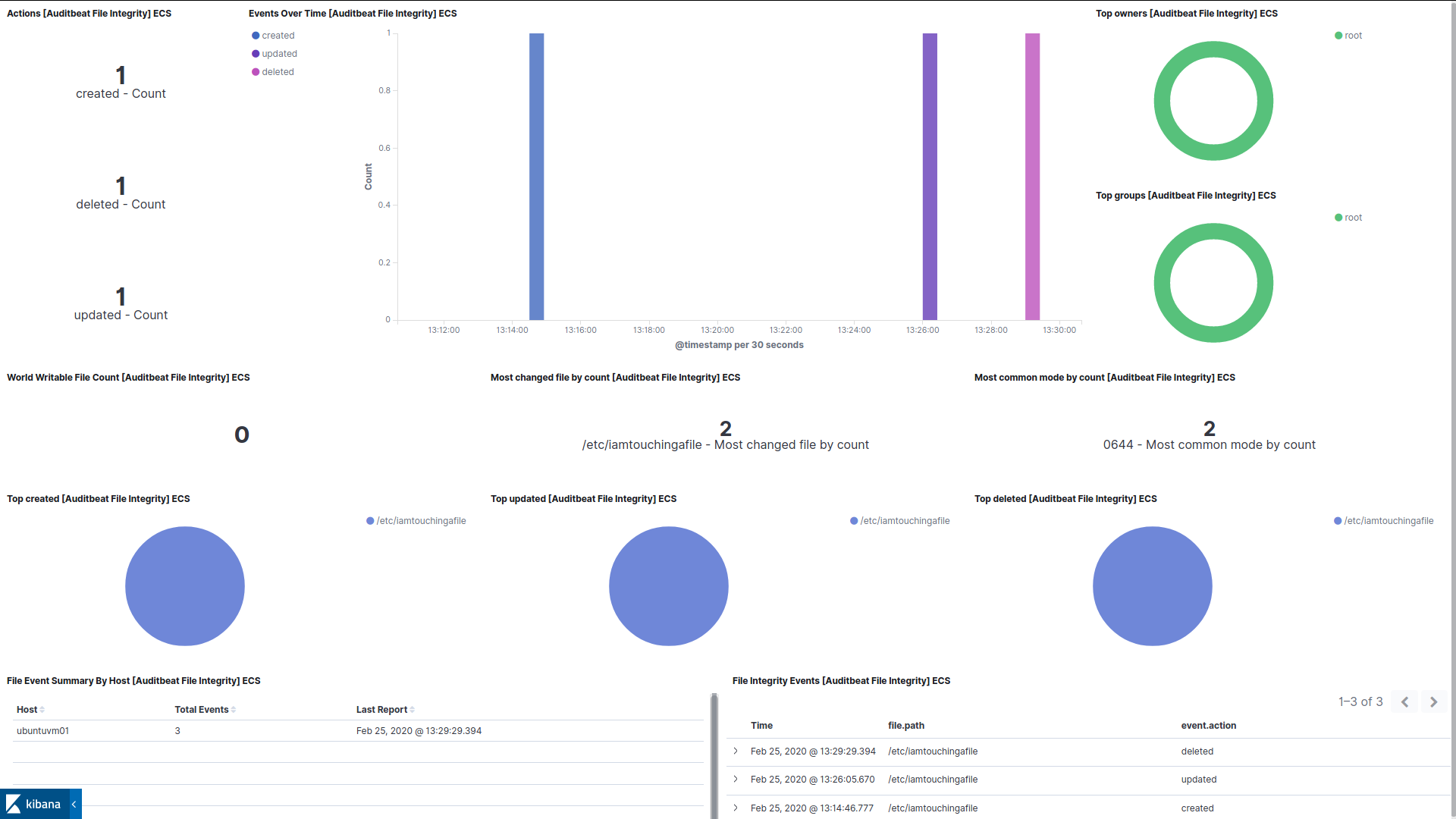


Rysunek 26

Załóżmy, że ten plik iamtouchingafile jest bardzo ważny i dokładnie należy monitorować każdą zmianę dokonaną w tym pliku.

**vagrant@ubuntuvm01:~$** *echo hello | sudo tee -a /etc/iamtouchingafile*    
hello

**vagrant@ubuntuvm01:~$** *sudo rm /etc/iamtouchingafile*



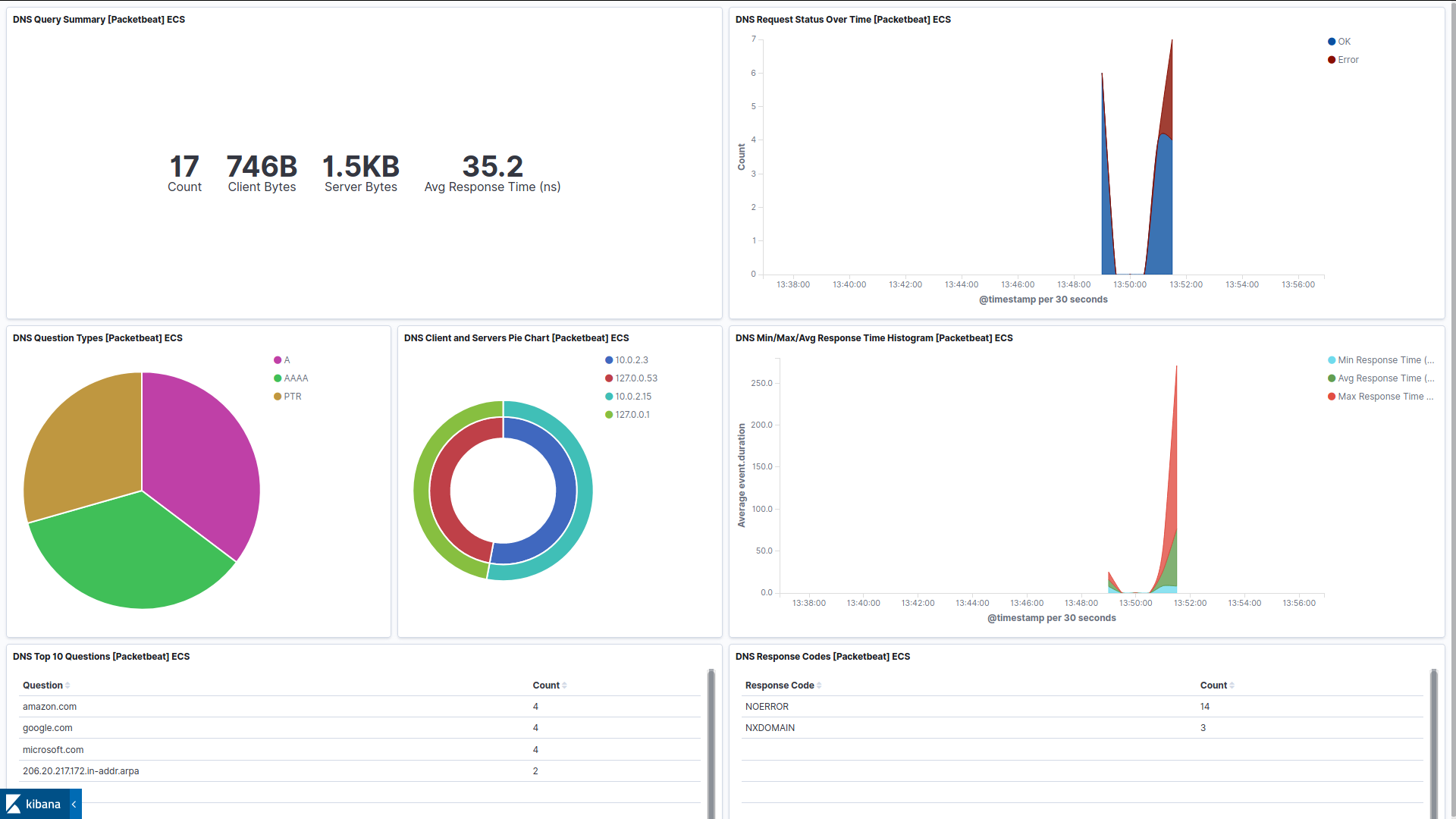
Rysunek 27

Następnie przejdę do konfiguracji i prezentacji PacketBeat.  
**vagrant@ubuntuvm01:~$** *ping google.com*   
PING google.com (172.217.20.206) 56(84) bytes of data.   
64 bytes from waw02s08.1e100.net (172.217.20.206): icmp\_seq=1 ttl=63 time=10.2 ms   
64 bytes from waw02s08.1e100.net (172.217.20.206): icmp\_seq=2 ttl=63 time=10.4 ms   
64 bytes from waw02s08.1e100.net (172.217.20.206): icmp\_seq=3 ttl=63 time=10.9 ms   
^C   
--- google.com ping statistics ---   
3 packets transmitted, 3 received, 0% packet loss, time 2003ms   
rtt min/avg/max/mdev = 10.216/10.538/10.928/0.317 ms

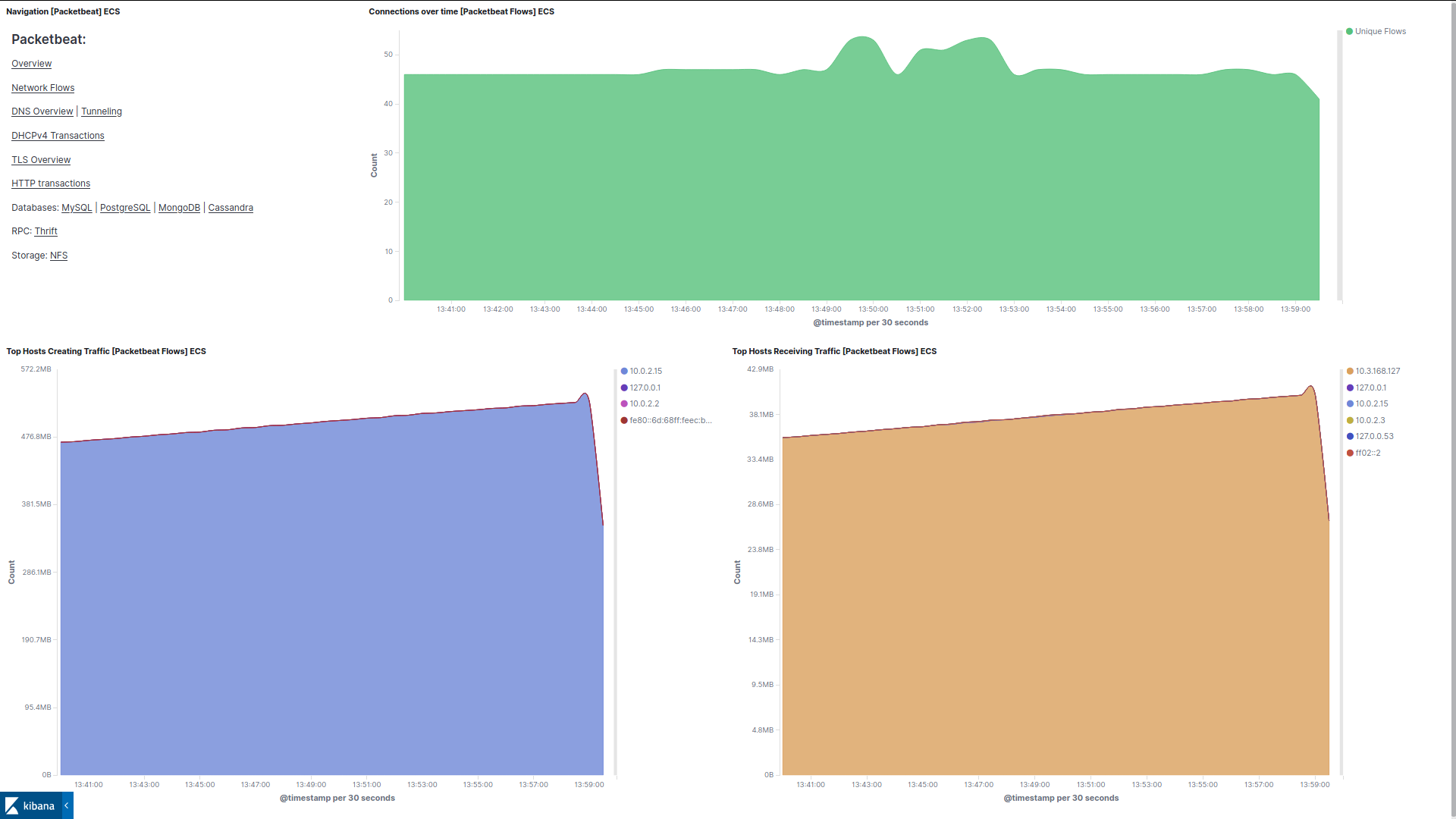
**vagrant@ubuntuvm01**:**~**$ ping microsoft.com   
PING microsoft.com (40.76.4.15) 56(84) bytes of data.   
^C   
--- microsoft.com ping statistics ---   
12 packets transmitted, 0 received, 100% packet loss, time 11261ms

vagrant@ubuntuvm01:**~**$ ping amazon.com  
--- amazon.com ping statistics ---   
3 packets transmitted, 3 received, 0% packet loss, time 2002ms   
rtt min/avg/max/mdev = 124.139/124.443/124.837/0.410 ms  
****

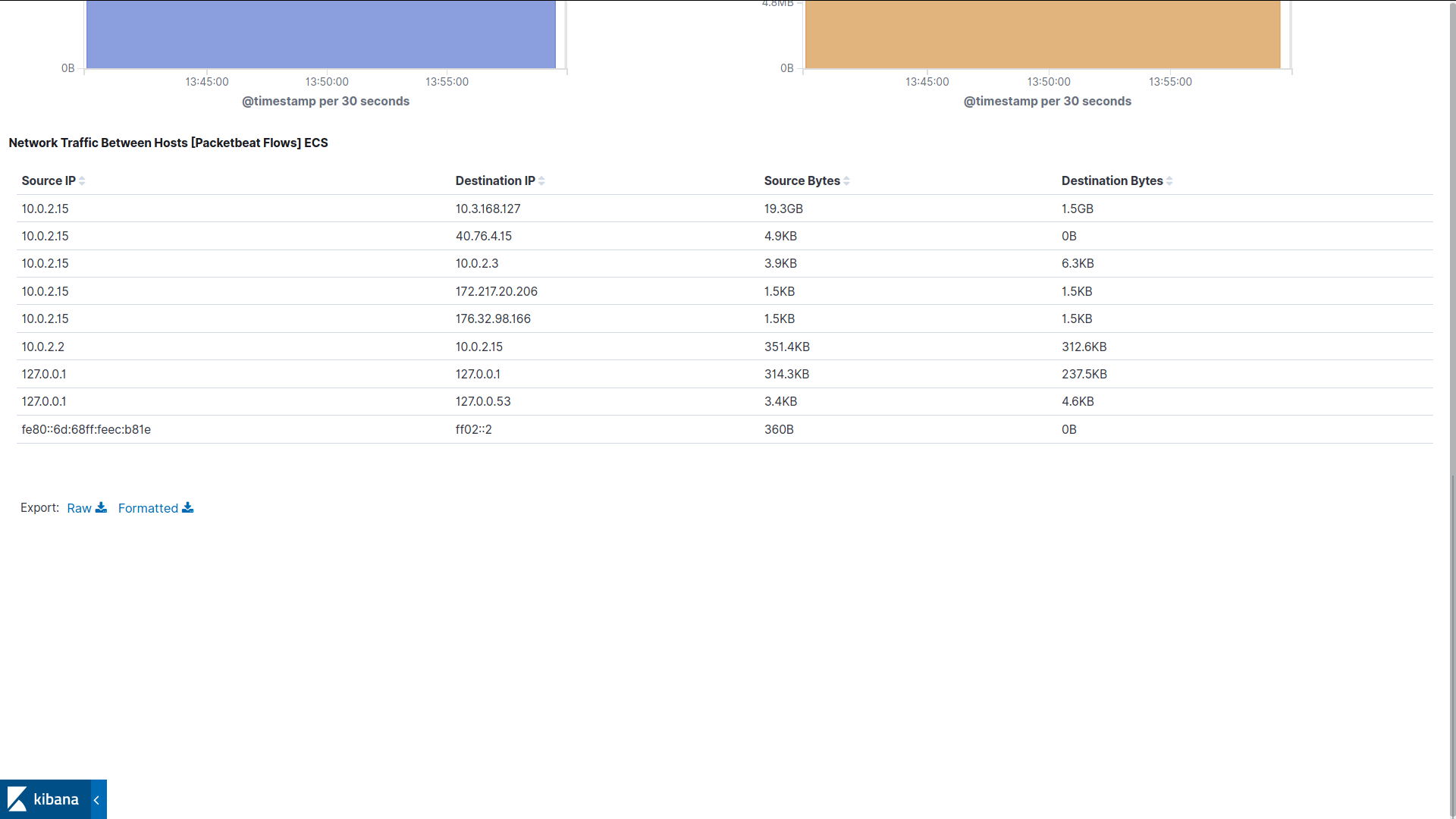
Rysunek 28



Rysunek 29



Rysunek 30 Przepływy sieciowe



Rysunek 31

**vagrant@ubuntuvm01:~$** *ip a s*

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default

link/ether 02:6d:68:ec:b8:1e brd ff:ff:ff:ff:ff:ff

inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3

valid\_lft 75310sec preferred\_lft 75310sec

inet6 fe80::6d:68ff:feec:b81e/64 scope link

valid\_lft forever preferred\_lft forever

3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default

link/ether 08:00:27:0c:22:4c brd ff:ff:ff:ff:ff:ff

inet 172.42.42.121/24 brd 172.42.42.255 scope global enp0s8

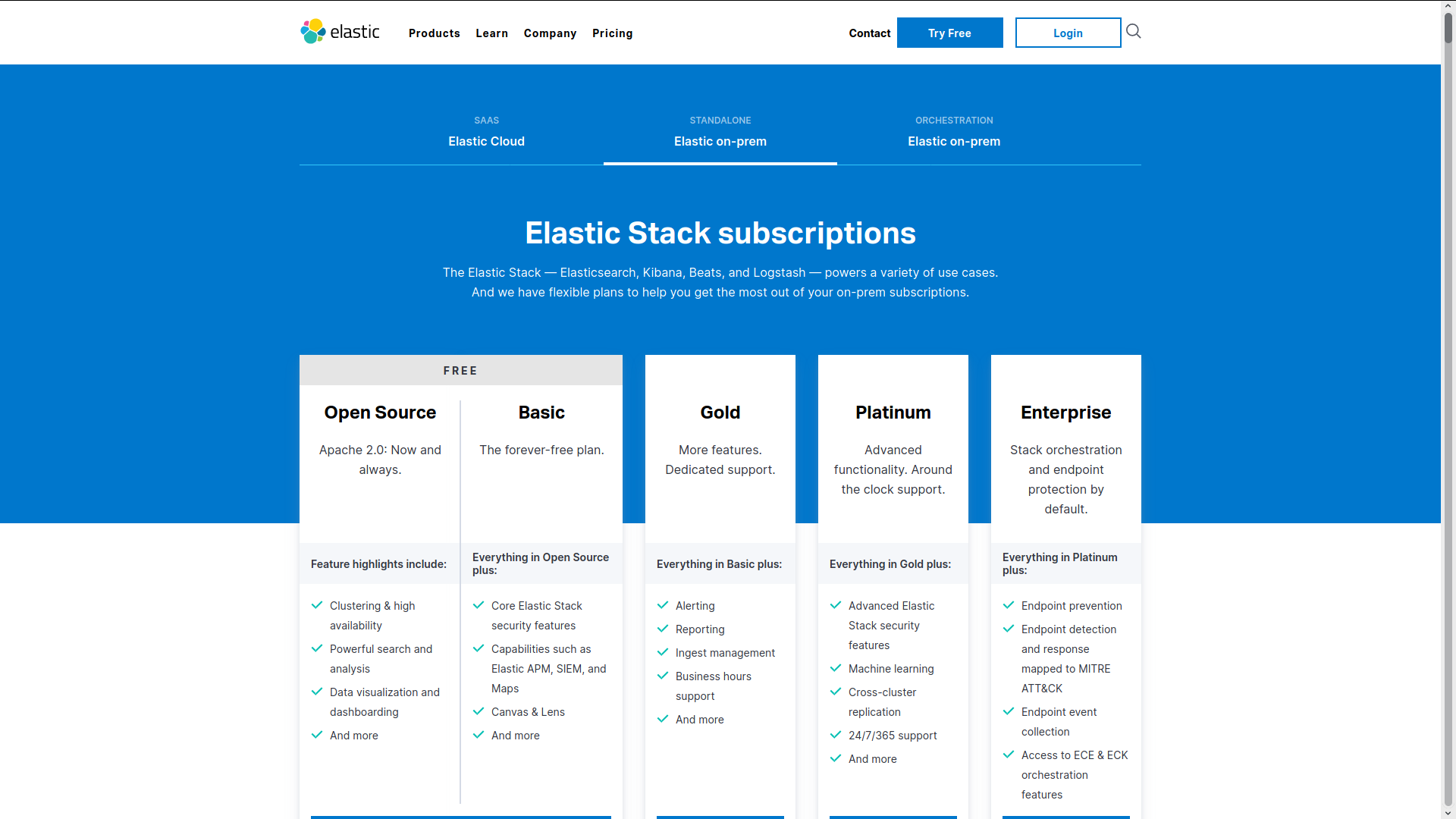
valid\_lft forever preferred\_lft forever

inet6 fe80::a00:27ff:fe0c:224c/64 scope link

valid\_lft forever preferred\_lft forever

# Rozdział 9

W tym rozdziale poruszę temat instalacji, konfiguracji weryfikacji i testowanie dodatku X-Pack. Dodatek ten służy do zabezpieczenia zasobów oraz limitacji uprawnień w zależności od przypisanej użytkownikowi roli. Limity te są ustanawiane dla webowego panelu usług ElasticSearch i Kibana.



Rysunek 32

Ja w swojej pracy magisterskiej korzystam z ELK na licencji open source wraz z podstawowymi dodatkami. Na szczęście wersja bazowa umożliwia darmowe zarządzanie bezpieczeństwem przy użyciu dodatku X-Pack. Bez zaimplementowanych mechanizmów logowania każdy użytkownik może dostać się do webowego panelu zarządzania usługami. Po konfiguracji X-Pack będzie wymagane podanie nazwy użytkownika i hasła. Klientem logującym zdarzenia do ELK będzie maszyna wirtualna Ubuntu z zainstalowanym pakietem Filebeat. Elastic Search i Kibana zostaną uruchomione w kontenerach Docker. Zostaną wygenerowane certyfikaty oraz klucze prywatne. Aby używać dodatku X-Pack należy aktywować szyfrowanie SSL a co za tym wygenerować certyfikaty i klucze prywatne, które są konieczne do zestawienia tego szyfrowanego połączenia.

Na początku wystartuje kontener Docker z Elastic Search. Odpalam go z parametrami deamon oraz pojedynczy węzeł (bez klastra). Obraz pobieram z oficjalnego repozytorium Docker. Kontener ten będzie potrzebny tylko i wyłącznie do wygenerowania certyfikatu, potem zostanie zlikwidowany.

**[szymon@LinuxManjaro** **play]$** *docker run -d -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:7.4.2*   
68b7decee1499ff931eca62561044e7bb54f7f60bfb5d51c0cd0b563164240be

**[szymon@LinuxManjaro play]$** *docker ps*

CONTAINER ID IMAGE COMMAND PORTS NAMES

68b7decee149 elasticsearch:7.4.2 "/usr/local/bin/dock…" 9200/tcp, 9300/tcp amazing\_yonath

Będę wykonywał teraz zdalne komendy bashowe na tym kontenerze przy użyciu exec.

**[szymon@LinuxManjaro** **play]$** *docker exec -it amazing\_yonath bash*

**[root@68b7decee149 elasticsearch]#** *pwd*/usr/share/elasticsearch

Będę generował teraz certyfikat oraz parę kluczy.

**[root@68b7decee149 elasticsearch]#** *bin/elasticsearch-certutil ca*

This tool assists you in the generation of X.509 certificates and certificate signing requests for use with SSL/TLS in the Elastic stack. The 'ca' mode generates a new 'certificate authority' This will create a new X.509 certificate and private key that can be used to sign certificate when running in 'cert' mode. Use the 'ca-dn' option if you wish to configure the 'distinguished name'   
of the certificate authority By default the 'ca' mode produces a single PKCS#12 output file which holds:   
   \* The CA certificate   
   \* The CA's private key   
If you elect to generate PEM format certificates (the -pem option), then  
the output will be a zip file containing individual files for the CA certificate and private key.

Please enter the desired output file [elastic-stack-ca.p12]:    
Enter password for elastic-stack-ca.p12 :

**[root@68b7decee149 elasticsearch]#** *ls*   
LICENSE.txt     **bin**     elastic-stack-ca.p12  **logs**   
NOTICE.txt      **config**  **jdk**                   **modules**   
README.textile  **data**    **lib**                   **plugins**

Wychodzę z trybu bash exec poleceniem exit.

Należy teraz przekopiować wygenerowany certyfikat do maszyny hosta.

**[szymon@LinuxManjaro** **play]$** *docker cp amazing\_yonath:/usr/share/elastics  
earch/elastic-stack-ca.p12 .*

***[*szymon@LinuxManjaro** **play]$** *ls*elastic-stack-ca.p12

Teraz spokojnie mogę usunąć ten kontener.

**[szymon@LinuxManjaro** **play]$** *docker stop amazing\_yonath*  
**[szymon@LinuxManjaro** **play]$** *docker rm amazing\_yonath*  
Teraz będę klonować repozytorium Git.

**[szymon@LinuxManjaro** **play]$** *git clone https://github.com/justmeandopensource/elk*

**[szymon@LinuxManjaro** **play]$** *cd elk/docker && ls*   
docker-compose-v7.1.1.yml  docker-compose-xpack.yml  docker-compose.yml

**[szymon@LinuxManjaro** **docker]$** *vi docker-compose-xpack.yml*

version: '3'

services:

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:7.4.2

container\_name: elasticsearch

environment:

- node.name=elasticsearch

- discovery.seed\_hosts=elasticsearch

- cluster.initial\_master\_nodes=elasticsearch

- cluster.name=docker-cluster

- bootstrap.memory\_lock=true

- "ES\_JAVA\_OPTS=-Xms512m -Xmx512m"

- xpack.security.enabled=true

- xpack.security.transport.ssl.enabled=true

- xpack.security.transport.ssl.keystore.type=PKCS12

- xpack.security.transport.ssl.verification\_mode=certificate

- xpack.security.transport.ssl.keystore.path=elastic-stack-ca.p12

- xpack.security.transport.ssl.truststore.path=elastic-stack-ca.p12

- xpack.security.transport.ssl.truststore.type=PKCS12

ulimits:

memlock:

soft: -1

hard: -1

volumes:

- ./elastic-stack-ca.p12:/usr/share/elasticsearch/config/elastic-stack-ca.p12

- esdata1:/usr/share/elasticsearch/data

ports:

- 9200:9200

kibana:

image: docker.elastic.co/kibana/kibana:7.4.2

container\_name: kibana

environment:

ELASTICSEARCH\_URL: "http://elasticsearch:9200"

ELASTICSEARCH\_USERNAME: "kibana"

ELASTICSEARCH\_PASSWORD: "kibana"

ports:

- 5601:5601

depends\_on:

- elasticsearch

volumes:

esdata1:

driver: local

Plik ten zawiera konfigurację związaną z dodatkiem X-Pack. Aktywacja samego pakietu oraz szyfrowania SSL Dodana jest również lokalizacja pliku certyfikatu. Bardzo ważne jest aby nazwa wygenerowanego certyfikatu była tożsamą z nazwą zapisaną w konfiguracji. Sam certyfikat dodany jest również do Dockera do jego pamięci nieulotnej, trwałej (volumes, persistant). Plik konfiguracyjny zawiera również login i hasło umożliwiające usłudze Kibana zalogowanie się do usługi Elastic Search. Nie będę startował dwóch usług jednocześnie gdyż do poprawnej działalności Kibana wymagane jest najpierw utworzenie odpowiednie użytkownika w Elastic Search.

Również muszę skopiować wygenerowany certyfikat do odpowiedniej lokalizacji.

**[szymon@LinuxManjaro** **play]$** *cp elastic-stack-ca.p12 ./elk/docker*

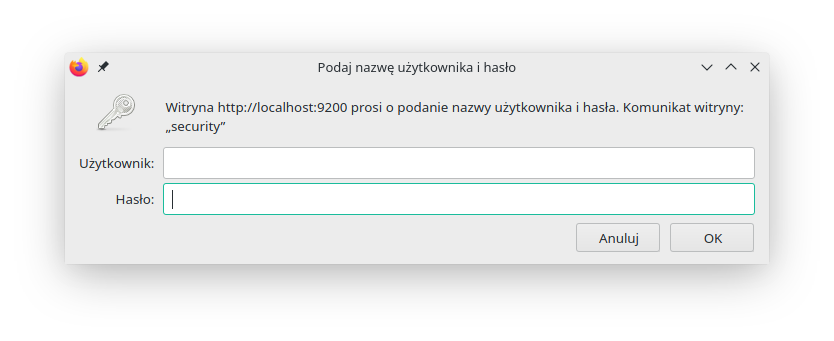
**[szymon@LinuxManjaro** **docker]$** *docker-compose -f docker-compose-xpack.yml up -d elasticsearch*

Creating network "docker\_default" with the default driver   
Creating volume "docker\_esdata1" with local driver   
Creating elasticsearch ... done

Na razie uruchamiam tylko elasticsearch, ponieważ muszę się najpierw do niego zalogować aby utworzyć odpowiednich wymaganych administracyjnych użytkowników.

**[szymon@LinuxManjaro** **docker-xpack]$** *docker-compose ps*   
   Name          Command               State                Ports          
----------------------------------------------------------------------------------  
elasticsearch   /usr/local/bin/docker   Up      0.0.0.0:9200->9200/tcp, 9300/tcp

Skoro elasticsearch jest uruchomiona możemy podjąć próbę dostępu do panelu webowego. W polu wyszukiwarki wpisuję url: http:/localhost:9200/ i pojawia się okno do logowania. Jeszcze nie został utworzony, żaden użytkownik jest nie jest to możliwe aby ta próba logowania została zakończona poprawnie.



Rysunek 33

{"error":{"root\_cause":[{"type":"security\_exception","reason":"unable to authenticate user [] for REST request [/]","header":{"WWW-Authenticate":"Basic realm=\"security\" charset=\"UTF-8\""}}],"type":"security\_exception","reason": "unable to authenticate user [] for REST request [/]","header":{"WWW-Authenticate":"Basic realm=\"security\" charset=\"UTF-8\""}},"status":401}

Teraz zaloguję się do ElasticSearch używając konsoli poprzez skrypty bashowe.

**[szymon@LinuxManjaro** **docker-xpack]$** *docker-compose exec elasticsearch bash*

**[root@71cf3ecf4d9d elasticsearch]#** *pwd*   
/usr/share/elasticsearch

Korzystam rozwiązania umożliwiającego dokonanie konfiguracji haseł.

**[root@71cf3ecf4d9d elasticsearch]#** *bin/elasticsearch-setup-passwords --help*

Sets the passwords for reserved users

Commands

--------

auto - Uses randomly generated passwords

interactive - Uses passwords entered by a user

Non-option arguments:

command

Option Description

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

-h, --help show help

-s, --silent show minimal output

-v, --verbose show verbose output

Jak widać z instrukcji, którą oferuje oprogramowanie są dwie możliwości nadania haseł albo automatyczna (przypisywane są hasła losowe) albo interaktywna, która pozwala nadać administratorowi oddzielne sprecyzowane przez niego hasło do każdej roli administracyjnej usługi ElasticSearch. Ja zdecydowałem się wybrać tę drugą opcję.

**[root@71cf3ecf4d9d elasticsearch]#** *bin/elasticsearch-setup-passwords interactive*   
Initiating the setup of passwords for reserved users elastic,apm\_system,  
kibana,logstash\_system,beats\_system,remote\_monitoring\_user.   
You will be prompted to enter passwords as the process progresses.   
Please confirm that you would like to continue [y/N] **y**

Enter password for [elastic]: **elastic**  
Reenter password for [elastic]: **elastic**  
Enter password for [apm\_system]: **elastic**  
Reenter password for [apm\_system]: **elastic**  
Enter password for [kibana]: **kibana**  
Reenter password for [kibana]: **kibana**  
Enter password for [logstash\_system]: **elastic**  
Reenter password for [logstash\_system]: **elastic**  
Enter password for [beats\_system]: **elastic**  
Reenter password for [beats\_system]: **elastic**  
Enter password for [remote\_monitoring\_user]: **elastic**  
Reenter password for [remote\_monitoring\_user]: **elastic**  
Changed password for user [apm\_system]   
Changed password for user [kibana]   
Changed password for user [logstash\_system]   
Changed password for user [beats\_system]   
Changed password for user [remote\_monitoring\_user]   
Changed password for user [elastic]

Po ustaleniu haseł wylogowuję się z kontenera.

Powtarzam próbę logowania do panelu webowego. Wpisuję ustawione przeze mnie hasło i po poprawnym zalogowaniu otrzymuję informację o szczegółach uruchomionej usługi.

{

"name" : "elasticsearch",

"cluster\_name" : "docker-cluster",

"cluster\_uuid" : "h\_rlG838TFGE6YXuGD8-Hw",

"version" : {

"number" : "7.4.2",

"build\_flavor" : "default",

"build\_type" : "docker",

"build\_hash" : "2f90bbf7b93631e52bafb59b3b049cb44ec25e96",

"build\_date" : "2019-10-28T20:40:44.881551Z",

"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"

}

Kiedy już usługa ElasticSearch działa poprawnie jest to moment na wystartowanie kontenera, który dostarczał będzie usługę Kibana. Przypomnę tylko, że w konfiguracji docker-compose.yml były zawarte dane do logowania Kibana do ElasticSearch.

kibana:

image: docker.elastic.co/kibana/kibana:7.4.2

container\_name: kibana

environment:

ELASTICSEARCH\_URL: "http://elasticsearch:9200"

ELASTICSEARCH\_USERNAME: "kibana"

ELASTICSEARCH\_PASSWORD: "kibana"

ports:

- 5601:5601

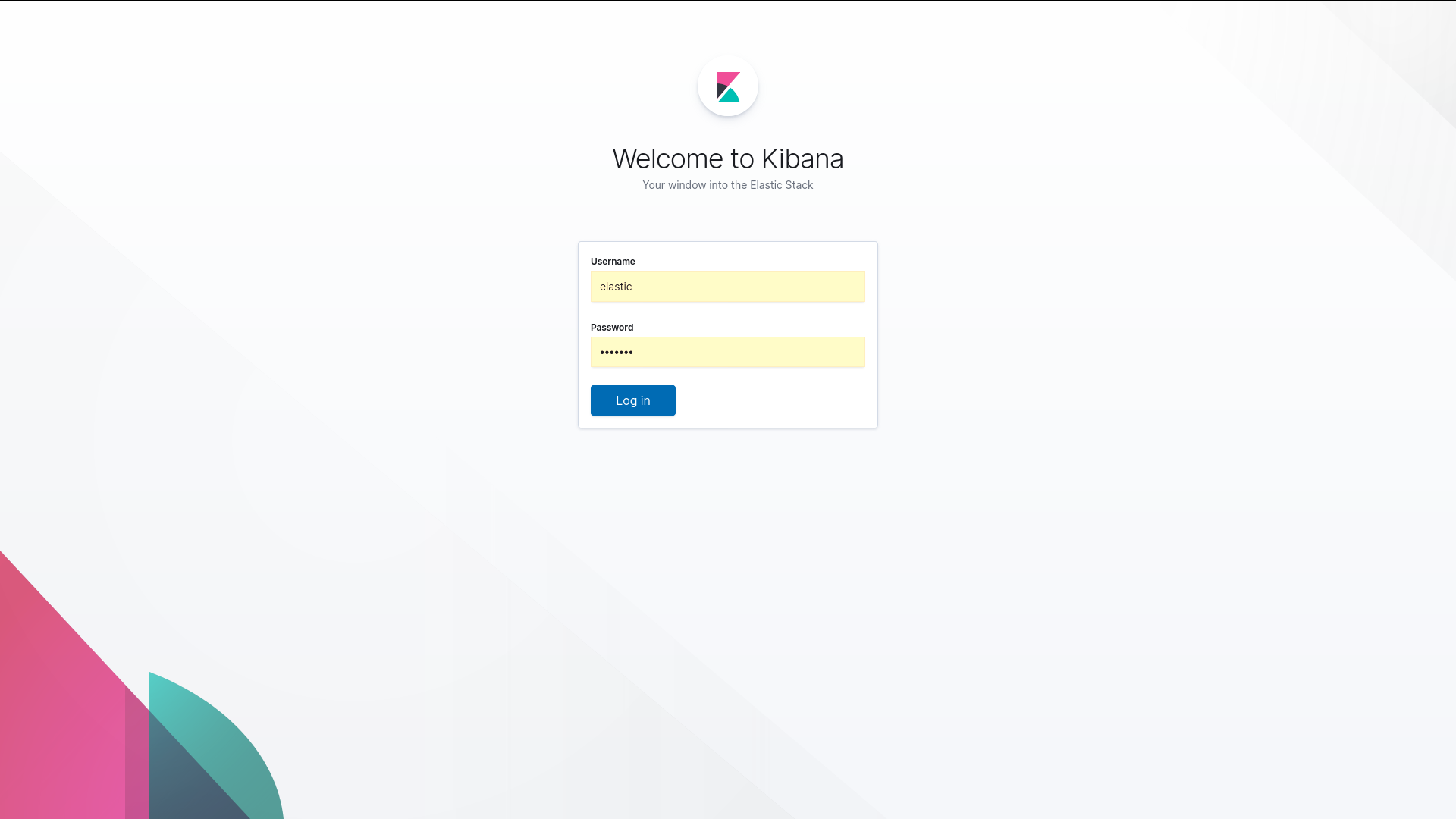
depends\_on:

- elasticsearch

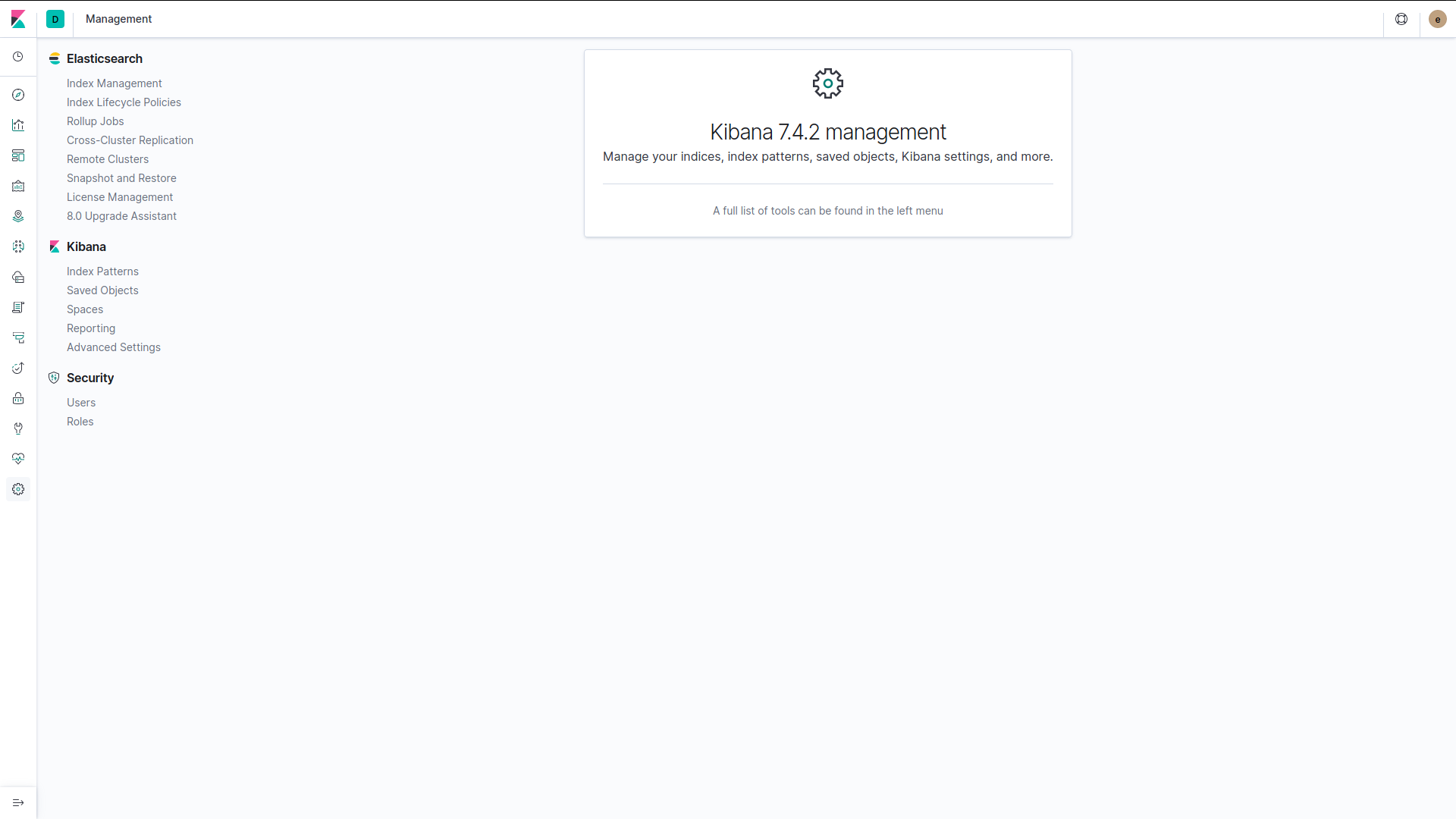
**[szymon@LinuxManjaro** **docker-xpack]$** docker-compose up -d kibana

elasticsearch is up-to-date   
Creating kibana ... done

Logowanie do panelu webowego Kibana na url: http://localhost:5601



Rysunek 34

Dzięki aktywowaniu dodatku X-Pack w zakładce Management pojawiła się dodatkowa opcja Security. 

Rysunek 35

Tam mam listę wszystkich użytkowników oraz ról utworzonych w systemie.

Teraz zaprezentuję w jaki sposób można tworzyć nowe role i przypisywać do nich nowo utworzonych użytkowników.

W pliku konfiguracyjnym FileBeat na maszynie wirtualnej Ubuntu w sekcji Output należy umieścić informacje o loginie i haśle do ElasticSearch. W innym wypadku usługa FileBeat się nie zautoryzuje i nie będzie mogła przesyłać żadnych logów.

**vagrant@ubuntuvm01**:**~**$ *sudo filebeat test output*             
elasticsearch: http://10.3.168.127:9200...   
 parse url... OK   
 connection...                                                            
   parse host... OK   
   dns lookup... OK                                                       
   addresses: 10.3.168.127                                                
   dial up... OK   
 TLS... WARN secure connection disabled                                   
 talk to server... ERROR 401 Unauthorized: {"error":{"root\_cause":[{"type":"security\_exception","reason":"missing authentication credentials for REST request [/]","header":{"WWW-Authenticate":"Basic realm=\"security\" charset=\"UTF-8\""}}],"type":"security\_exception","reason":"missing authentication credentials for REST request [/]","header":{"WWW-Authenticate":"Basic realm=\"security\" charset=\"UTF-8\""}},"status":401}

#-------------------------- Elasticsearch output ------------------------------

output.elasticsearch:

# Array of hosts to connect to.

hosts: ["10.3.168.127:9200"]

# Protocol - either `http` (default) or `https`.

#protocol: "https"

# Authentication credentials - either API key or username/password.

#api\_key: "id:api\_key"

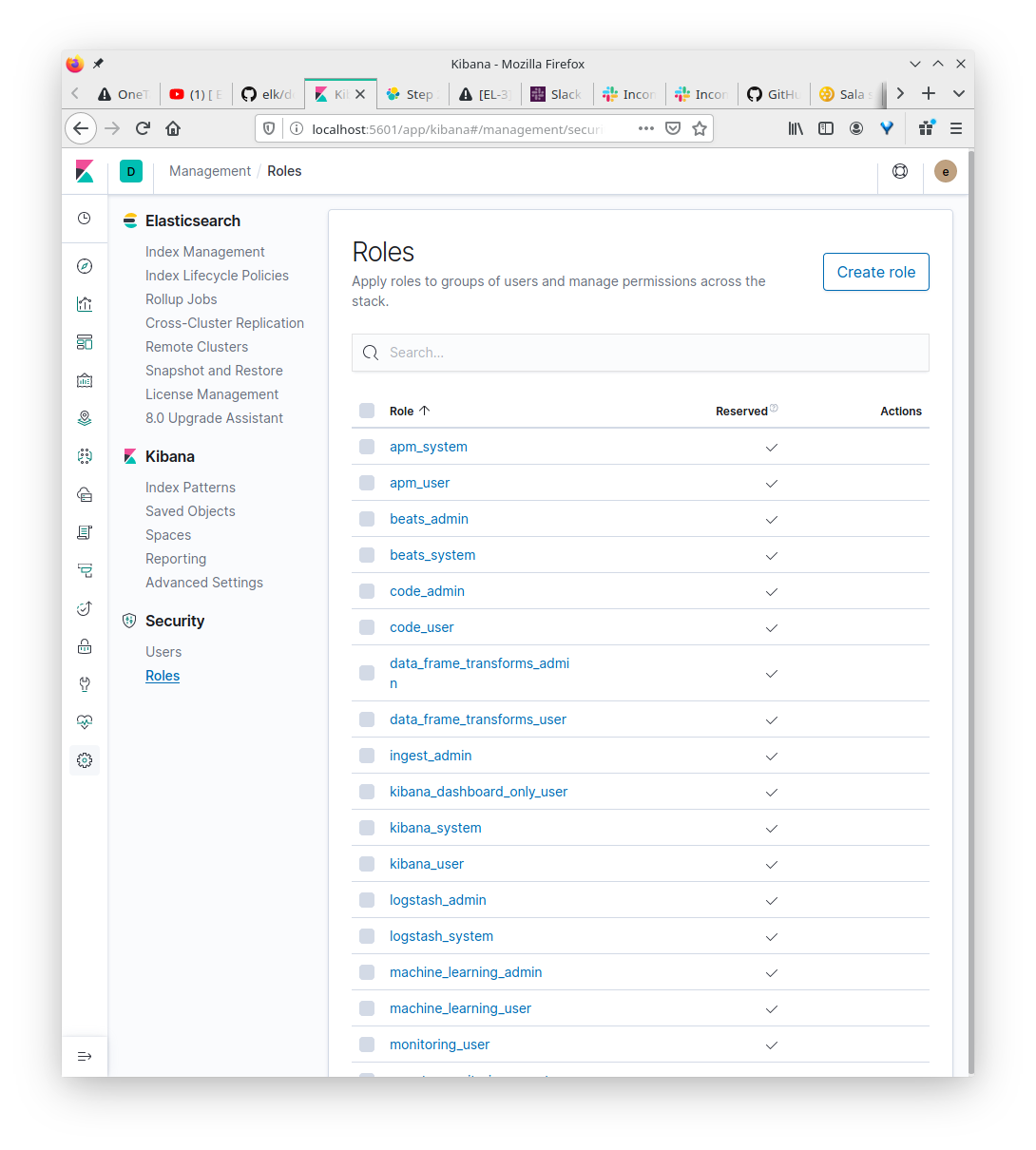
~~#username: "elastic"~~

~~#password: "changeme"~~

username: "elastic"

password: "elastic"

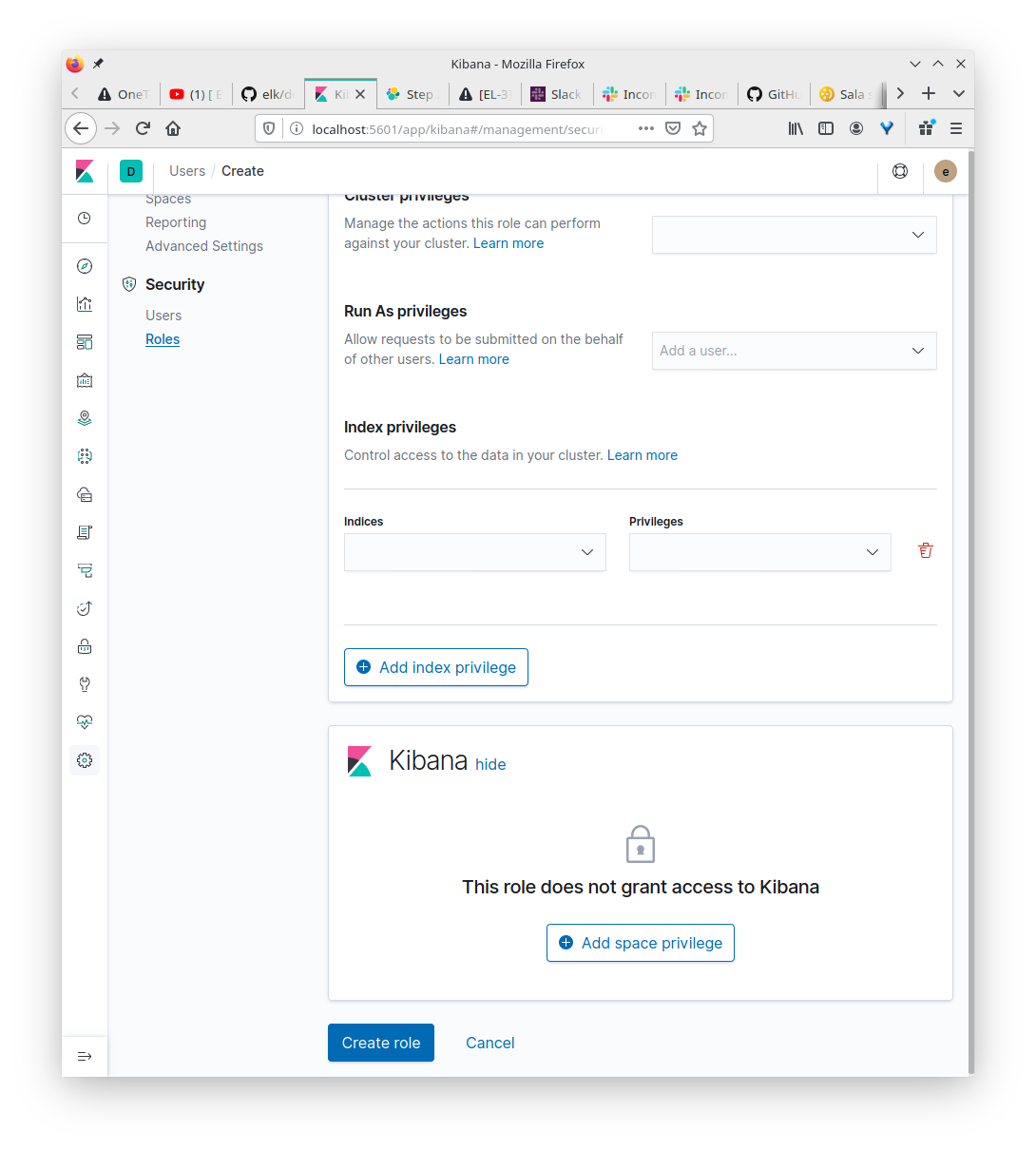
Teraz zaprezentuję w jaki sposób można zarządzać dostępem do logów, indeksów i wzorów dla różnych użytkowników. Wrócę do zakładki Management i pod zakładki Roles, postanowiłem utworzyć nową rolę poprzez naciśniecie przycisku „Create role”.



Rysunek 36

Role name: demo

Wybieram opcję „Add space privilege”



Rysunek 37

Spaces: DefaultPrivilege: Read

Spaces dotyczy jakie opcje są widoczne dla konkretnej roli. Privilege informuje o tym jakie prawa są nadane roli. Czy jest to tylko odczytywanie parametrów czy również ich edytowanie.

Teraz utworzę użytkownika i przypiszę mu tę rolę. Naciskam przycisk „Create user”.

Username: demo

Password: demodemo

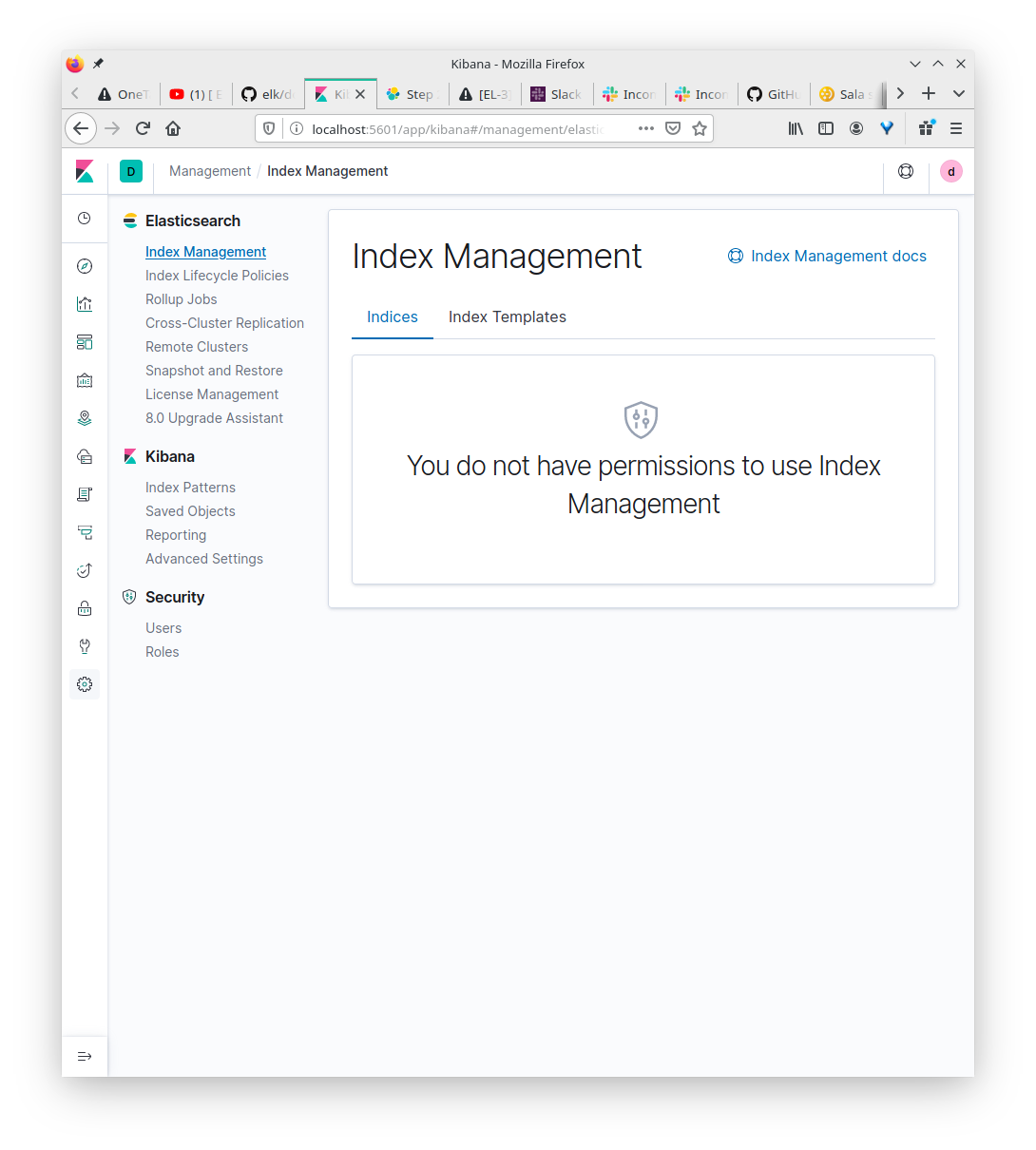
Full name: demo

Email adress: demo@demo.com

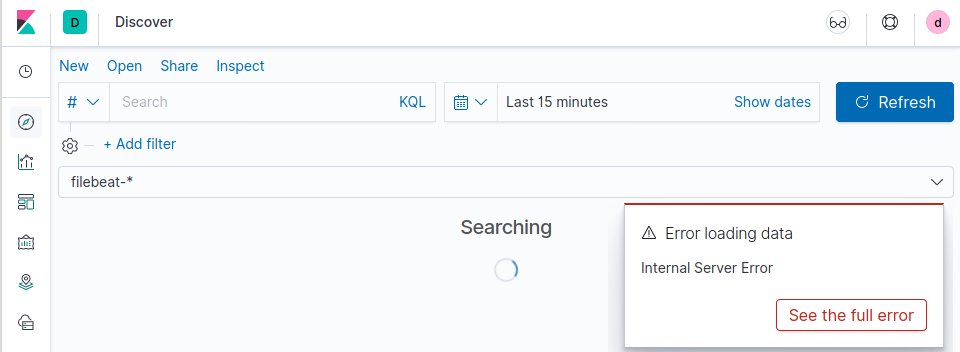
Roles: demo

Teraz się wyloguję i zaloguję jako użytkownik demo.

Wchodząc do zakładki Index Management pojawia mi się informacja o braku uprawnień.



Rysunek 38

Zakładka Discovery nie jest możliwa do zaprezentowania i wyświetlany jest wewnętrzny błąd serwera.

Rysunek 39

Wylogowuję się z konta demo. Loguję z powrotem na konto elastic oraz modyfikuję uprawnienia roli demo, tak aby działały wymagane funkcjonalności. Nadaję roli demo prawo czytania indeksu Filebeat.

Po tej zmianie wewnętrzny błąd serwera zniknął i w zakładce Discovery mogę oglądać logi. W sytuacji kiedy na produkcji mamy wiele różnych indeksów: FileBeat, PacketBeat, WinLogBeat, Heartbeat itp. każdemu z użytkowników możemy nadać uprawnienia do odczytywania tylko niektórych z nich. Tak aby administracyjni odpowiedzialni za Windows mieli podgląd do WinLogBeata a administratorzy z innych zespołów mieli podgląd do logów z baz danych czy serwerów z systemem Linux.

## Podsumowanie i wnioski

Elasticsearch to rozproszony silnik RESTful służący do wyszukiwania i analizy, który jest w stanie rozwiązać rosnącą liczbę przypadków użycia. Będąc sercem elastycznego stosu, centralnie przechowuje twoje dane, dzięki czemu możesz odkryć oczekiwane i odkryć nieoczekiwane. Elasticsearch pozwala przeprowadzać i łączyć wiele rodzajów wyszukiwań - ustrukturyzowane, nieustrukturyzowane, geo, metryczne itp. Opiera się na języku programowania Java, który umożliwia Elasticsearch działanie na różnych platformach. Umożliwia użytkownikom przeglądanie bardzo dużej ilości danych z bardzo dużą prędkością.

Logstash to otwarty proces przetwarzania danych po stronie serwera, który pobiera dane z wielu źródeł jednocześnie, przekształca je, a następnie wysyła do ulubionych „skrytek” (takich jak Elasticsearch). Dane są często rozproszone lub wyciszone w wielu systemach w wielu formatach. Logstash obsługuje różnorodne dane wejściowe, które jednocześnie pobierają zdarzenia z wielu wspólnych źródeł. Łatwe pobieranie danych z dzienników, metryk, aplikacji internetowych, magazynów danych i różnych usług AWS - wszystko w sposób ciągły i strumieniowy. Logstash ma wtykową platformę zawierającą ponad 200 wtyczek. Miksuj, dopasowuj i koordynuj różne wejścia, filtry i wyjścia, aby działać w harmonii z potokiem.

Kibana to otwarta platforma analityczna i wizualizacyjna zaprojektowana do współpracy z Elasticsearch. Korzystasz z Kibana do wyszukiwania, przeglądania i interakcji z danymi przechowywanymi w indeksach Elasticsearch. Możesz łatwo przeprowadzić zaawansowaną analizę danych i wizualizować dane na różnych wykresach, tabelach i mapach. Kibana ułatwia zrozumienie dużych ilości danych. Prosty interfejs oparty na przeglądarce umożliwia szybkie tworzenie i udostępnianie dynamicznych pulpitów nawigacyjnych wyświetlających zmiany zapytań Elasticsearch w czasie rzeczywistym.

Dzięki aktywowaniu dodatku X-Pack w zakładce Management pojawiła się dodatkowa opcja Security. Tam mam listę wszystkich użytkowników oraz ról utworzonych w systemie.

## Literatura

Źródła drukowane

1. Abecadłowski A., *Wybrane zagadnienia niektórych problemów*, [w:] Ekspress Ilustrowany nr 7/2012 z 21 grudnia 2012, Agora, Warszawa 2012, str. 12-13
2. Kowalski J., *Wybrane aspekty niektórych zagadnień*, wydanie III poprawione, WNT, Warszawa 2011
3. [RFC 4253 - The Secure Shell (SSH) Transport Layer Protocol](http://tools.ietf.org/html/rfc4253)

Źródła elektroniczne

1. http://www.polska.pl, *Polska strona główna* (odsyłacz sprawdzony 21.12.2014)
2. http://www.iis.p.lodz.pl, *Instytut Informatyki Stosowanej - strona główna* (odsyłacz sprawdzony 29.12.2014)

Literatura powinna być ponumerowana w nawiasach kwadratowych po przednim posortowaniu w porządku alfabetycznym. Warto odróżnić źródła drukowane od elektronicznych. Można to zrobić dokonując osobnej numeracji i rozdziału na dwa spisy: osobno źródła drukowane [1] osobno źródła elektroniczne oznaczenie poprzez dodanie litery „i” w numeracji [1i]. Standardy takie jak RFC są dokumentami drukowanym.

Książki:

Autor, *tytuł*, wydawnictwo, wydanie, miejsce wydania rok

Kowalski J., *Wybrane aspekty niektórych zagadnień*, wydanie III poprawione, WNT, Warszawa 2011

Artykuły:

Autor, *tytuł*, wydawnictwo, nazwa periodyku lub zbioru, wydawnictwo, miejsce wydania rok, strony na których umieszczony był artykuł

Abecadłowski A., *Wybrane zagadnienia niektórych problemów*, [w:] Ekspress Ilustrowany nr 7/2012 z 21 grudnia 2012, Agora, Warszawa 2012, str. 12-13

Odsyłacz internetowy:

URL, *opis*, (data weryfikacji)

http://www.polska.pl, *Polska strona główna* (odsyłacz sprawdzony 21.12.2014)

## <https://blog.gutek.pl/2017/01/16/elastic-wstep/>

<https://luktom.net/pl/e617-monitorowanie-ruchu-na-routerach-cisco-przy-pomocy-netflow-i-programu-ntop>

Vishal Sharma., *Beginning Elastic Stack*, wydanie I, Apress, 2016

Clinton Gormley, Zachary Tong, Elasticsearch: The Definitive Guide, O'Reilly Media, 2015

Adam Józefiok, GNS3. Emulowanie sieci komputerowych Cisco, Helion, 2017

Ben Piper, Sieci Cisco w miesiąc. Podręcznik administratora, Helion, 2018

Barrie Sosinsky, Sieci komputerowe. Biblia, Helion, 2013

## Wykaz rysunków

[Rysunek 1 https://platform9.com/wp-content/uploads/2018/09/fluentd-architecture.jpg 13](#_Toc33553480)

[Rysunek 2 20](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553481)

[Rysunek 3 37](#_Toc33553482)

[Rysunek 4 37](#_Toc33553483)

[Rysunek 5 42](#_Toc33553484)

[Rysunek 6 42](#_Toc33553485)

[Rysunek 7 43](#_Toc33553486)

[Rysunek 8 45](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553487)

[Rysunek 9 46](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553488)

[Rysunek 10 52](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553489)

[Rysunek 11 53](#_Toc33553490)

[Rysunek 12 53](#_Toc33553491)

[Rysunek 13 53](#_Toc33553492)

[Rysunek 14 [Metricbeat System] Host overview ECS 58](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553493)

[Rysunek 15 Loaded machine learning job configurations 63](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553494)

[Rysunek 16 66](#_Toc33553495)

[Rysunek 17 66](#_Toc33553496)

[Rysunek 18 68](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553497)

[Rysunek 19 69](file:///C:\Users\Szymon\Desktop\praca_dyplomowa(v_10-lukart)_v2.docx#_Toc33553498)

[Rysunek 20 71](#_Toc33553499)

[Rysunek 21 72](#_Toc33553500)

[Rysunek 22 73](#_Toc33553501)

[Rysunek 23 73](#_Toc33553502)

[Rysunek 24 74](#_Toc33553503)

[Rysunek 25 74](#_Toc33553504)

[Rysunek 26 75](#_Toc33553505)

[Rysunek 27 75](#_Toc33553506)

[Rysunek 28 76](#_Toc33553507)

[Rysunek 29 77](#_Toc33553508)

[Rysunek 30 Przepływy sieciowe 77](#_Toc33553509)

[Rysunek 31 78](#_Toc33553510)

[Rysunek 32 79](#_Toc33553511)

[Rysunek 33 83](#_Toc33553512)

[Rysunek 34 86](#_Toc33553513)

[Rysunek 35 87](#_Toc33553514)

[Rysunek 36 88](#_Toc33553515)

[Rysunek 37 89](#_Toc33553516)

[Rysunek 38 89](#_Toc33553517)

[Rysunek 39 90](#_Toc33553518)

## Słownik skrótów i pojęć

Najważniejsze skróty należy rozwinąć i przetłumaczyć (jeśli można i jest sens) oraz opisać w jednym/dwóch zdaniach. Wszystkie powinny być posortowane w porządku alfabetycznym. Poniżej przykład.

**SNMP** (ang. *Simple Network Management Protocol* - Prosty Protokół Zarządzania Siecią) standard protokołu używanego do nadzoru i zarządzania różnymi elementami sieci komputerowej.

**OŚWIADCZENIA**

**(wydrukowane w webdziekanatu)**

Ogólne wytyczne dotyczące prac dyplomowych oraz dokumenty i wzory związane z pracą dyplomową:

http://www.iis.p.lodz.pl/prace\_dyplomowe.html

<http://www.iis.p.lodz.pl/prace_dyplomowe_1.html>

<http://www.iis.p.lodz.pl/prace_dyplomowe_2.html>

Do pracy musi być dołączona **płyta CD** z pracą oraz ewentualnie innymi materiałami (źródła elektroniczne, kod napisanych programów, pliki konfiguracyjne urządzeń itp.).

W pracy oraz na obronie muszą znaleźć się odpowiedzi na poniższe pytania:

* Czy treść pracy odpowiada tematowi określonemu w tytule?
* Czy i w jakim zakresie praca stanowi nowe ujęcie problemu?
* Sposób wykorzystania pracy (publikacja, udostępnienie instytucjom, materiał źródłowy)?
* **Co jest wkładem własnym dyplomanta w niniejszą pracę dyplomową**?