Keskitetty lokitiedon käsittelypalvelu

Elastic-tuotteilla

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ammattikorkeakoulututkinnon opinnäytetyö

Riihimäki, Tietotekniikka

syksy, 2016

Joona Nieminen

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

TIIVISTELMÄ

RIIHIMÄKI
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

Tekijä Joona Nieminen **Vuosi** 2016

Työn nimi Keskitetty lokitiedon käsittelypalvelu

TIIVISTELMÄ

Tämä opinnäytetyö käsittelee keskitettyä lokitiedon hallintaan ja käsittelyyn tarkoitetun järjestelmän käyttöönottoa ja säätämistä toimeksiantajan tarpeisiin sopivalla tavalla. Työssä käydään läpi sitä, miten kyseisiä ohjelmia ja palveluita asennetaan, säädetään ja käytetään. Työn toimeksiantajana toimi pieniohjelmisto talo Hämeenlinnasta, jolla oli tarve saada omien palvelimien ja tuotteiden tietoja selkeästi ja tehokkaasti tutkittavaksi. Lisäksi järjestelmän tuli toimia hyvänä tukena yrityksen eri osastoille.

Työn tavoitteena oli saada toimivat ohjelmistot ja näkemykset, joista saadaan seurattua omien ohjelmien lokia ja palvelimien resursseja. Työssä käytettiin toimeksiantajan valitseman yrityksen (Elastic) tuotteita ja sovellettiin Elastic-yrityksen virallisia dokumentaatioita tuotteista sekä omaa tietoa Windows- ja Linux-palvelimista. Työssä käytetty materiaali on pääosin Elastic-yrityksen omaa.

Yritys oli aloittanut hankkeen jo edellisenä vuonna, joten aihe valikoitui minulle tätä kautta.

Työn tuloksena yrityksen eri osastoilla on tehokkaat työkalut helppoon ja selkeään tapaan tutkia omien tuotteiden lokia tai palvelimien resursseja. Ohjelmat ovat myös helppoja ja selkeitä asentaa, säätää ja käyttää ympäristökohtaisesti.

Avainsanat Loki, Elastic, Kibana, Big data

Sivut 40 s. + liitteet 12 s.

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

RIIHIMÄKI
Degree Programme in Information Technology
Software technology

Author Joona Nieminen **Year** 2016

Subject of Bachelor's thesis Centralized log data processing service

ABSTRACT

This thesis is about deployment of the centralized log data processing service and on adjusting it for the company's needs. The Thesis describes installing, adjusting and using these softwares and services. The Thesis was commissioned by a small software house that had a major need for a clear and powerful method for following servers and the state of their own products. The Service was also to provide good support for the company's different departments.

The Objective of this thesis was to create working software and dashboards, where one could follow software logs and the server's hardware info. We used in this project the softwares and services, which the commissioner had choosen. Official documentation of Elastic was used while doing this thesis, as well as the author's own knowledge of the Windows- and Linux-servers. Mostly all the material's that were used in this thesis project came from Elastic.

The Commisioner had started this project last year, so the subject of this thesis was easy to choose.

As a result of this thesis project, powerful tool was proviced for the company for an easy and clear method for examining the product logs or the server hardware info. Different forms of software are also easy and clear to install, adjust and use in different environments.

Keywords Log, Kibana, Elastic, Big data

Pages **40** p. + appendices 12 p.

SANASTO

TERMI	NIMI	SELITYS
AD	Active Directory	Käyttäjätietokanta ja hakemistopalvelu. Palveluun tallennetaan käyttäjätiedot, tunnukset ja oikeudet. Palvelusta tarkistetaan käyttäjän oikeudet ja tunnukset.
Big data		Tarkoittaa suurten sekalaisten tietomassojen keräämistä, säilyttämistä ja analysointia ja niiden muuntamista ”järkeväksi”, tekniikkaa ja tiedettä hyväksi käyttäen
CSV-tiedosto		XML-formaattia yksinkertaisempi taulukkoformaatti.
Data		Koneellisesti luettavassa, viestittävässä tai käsiteltävässä muodossa oleva tieto.
DNS	Domain Name System	Internetin nimipalvelujärjestelmä.
HTTP	Hypertext Transfer Protocol	Protokolla jota käytetään tiedonsiirtoon.
Indeksi	Index	Indeksi on kuin tietokanta relaatiotietokannassa. Indeksi on määritelty usealle tyypille. Indeksi on looginen nimiavaruus, mikä kuvaa yhtä tai useampaan ensisijaista ”shardia” ja jolla voi olla nolla tai useampi replika ”shard”.
JSON	JavaScript Object Notation	On yksinkertainen tiedonvälityksen tiedostomuoto.
Klusteri	Cluster	Sisältää Elasticsearchissa ainakin yhden noden. Jokaisella klusterilla on yksi päänode, jonka klusteri valitsee ja joka voidaan korvata, jos päänode rikkoontuu.
LDAP	Lightweight directory Access Protocol	Käyttäjähakemisto. Protokolla, joka määrittelee tietyt palvelut, erilliset ohjelmat hoitavat hakemiston toteutuksen.
Lumberjack		Logstash-forwarderin käyttämä tiedonsiirto-protokolla.
MongoDB		Avoimen lähdekoodin dokumentaatio-orientoitu tietokanta

MySQL		Relaatiotietokantaohjelmisto
Node		Yksittäinen instanssi joka pyörii erillisellä koneella
OSI-Malli	Open Systems Interconnection	Malli, jota käyttäen pitäisi tietoliikenne järjestelmät suunnitella. Koostuu seitsemästä kerroksesta, Fyysinen-, siirtoyhteys-, verkko-, kuljetus-, istunto-, esitystapa- ja soveluskerros
PKI	Public Key Infrastructure	PKI-järjestelmissä varmentaja allekirjoittaa digitaalisesti julkisen avaimen, jota voidaan jakaa käyttäjille, ja täten tunnistaa järjestelmässä luotetuksi.
Primary Shard	Ensisijainen ”sirpale”	Jokainen dokumentti säilötään yhteen ensisijaiseen sirpaleeseen. Kun dokumentti indeksoidaan, se indeksoidaan ensiksi ensisijaiselle sirpaleelle, jonka jälkeen vasta kaikille replika sirpaleille. Vakiona indeksissä on viisi ensisijaista sirpaletta. Riippuen dokumenttien määrästä voidaan määrää tiputtaa tai nostaa. Ensisijaisien sirpaleiden määrää ei voi muuttaa, jos indeksi on jo luotu.
Raakadata		Asioiden lähettämää analysoimatonta tietoa, jota analysoimalla saadaan aikaiseksi reaaliaikaista analytiikkaa, jonka avulla voidaan esimerkiksi tehostaa liiketoimintaa, kohdistamalla mainontaa käyttäjä kohtaisesti.
Replica Shard	Replika ”sirpale”	Jokaisella ensisijaisella sirpaleella on nolla tai useampi replika sirpale. Kopio ensisijaisesta sirpaleesta. Replikalla on kaksi tarkoitusta: Jos ensisijaiseen sirpaleeseen tulee vikaa, on tallella täysi kopio ja hakujen tehokkuuden nostaminen.
SaaS	Software as a Service	Ohjelmiston hankinta palveluna, oston sijaan. Käyttöönotto käyttäjälle helppoa. Usein halvempi ratkaisu pienille yrityksille kuin omat palvelimet.
Shard	”Sirpale”	Sirpale on yksi Lucene instanssi. Sirpale on matalan-tason ”työläinen”,

		jota elasticsearch ohjailee automaattisesti. Indeksi viittaa ensisijaisiin ja replika sirpaleisiin. Sirpaleisiin ei tarvitse viitata suoraan lukumäärän päättämisen jälkeen, sen sijaan koodin pitäisi työskennellä vain indeksin kanssa. Elasticsearch jakaa sirpaleita kaikkien samassa klusterissa olevien nodejen kanssa, ja voi myös siirtää sirpaleita automaattisesti nodesta toiseen, jos nodeen tulee häiriö tai jos nodejen lukumäärää kasvatetaan.
SMTP	Simple Mail Transfer Protocol	Sähköpostin välitys protokolla.
TCP/IP-viitemalli		Tietoliikenneverkkojen viitemalli. Sisältää neljä porrasta, joista kolme on samaa kuin OSI-mallissa. Sisältää soveluskerroksen, kuljetuskerroksen, verkkokerroksen ja peruskerroksen.
Template		JSON-muotoinen pohja Beats -ohjelmille
WinPcap		Ohjelma, jonka Packetbeat tarvitsee, jotta saadaan TCP/IP-viitemallin Peruskerroksen dataa tallennettua.
YAML		Yksinkertainen merkintäkieli
ZIP		Tiedonpakkaus menetelmä

SISÄLLYS

1 JOHDANTO

Viime vuosien aikana lokien määrä on kasvanut rajusti eikä kaikkea tätä dataa ole mahdollista järkevästi käydä palvelin kerrallaan tutkimassa.

Lokien määrä ja tarve niiden tutkimiselle on kasvanut rajusti viime vuosina. Lokien seurannasta on tullut erittäin työlästä ja sekavaa. Opinnäytetyöni tarkoitus on ratkaista tämä ongelma Big data-ajattelutavalla.

Toimeksiantaja on seurannut palvelimien resursseja, virheilmoituksia ja omien tuotteiden lokeja suoraan palvelimelta tai asiakkaan huomautuksesta. Tapa on hidas ja kömpelö, minkä vuoksi hukataan työntekijöiden kallista aikaa. Tähän haluttiin muutos.

Opinnäytetyön tarkoituksena on asentaa, ottaa käyttöön ja määritellä toimeksiantajan valitseman Elastic-yrityksen tuotteita yrityksen liiketoimintaa tukevaksi rakenteeksi. Järjestelmän on tarkoitus toimia helpdeskin, ohjelmistokehityksen ja tekniikan tukena, kehityksessä, vian selvityksessä tai ennakoinnissa.

Elastic-yrityksen tuotteilla saadaan aikaiseksi järjestelmä, mistä kaikki tarpeellinen tieto nähdään vaivattomasti, ja voidaan luoda hälytyksiä erilaisista ongelmatilanteista. Järjestelmän käyttöönoton jälkeen voidaan palvelimia seurata yhdestä paikasta, minkä vuoksi palvelimilla ei tarvitsisi käydä kuin ongelman ratkonnassa.

Toimeksiantaja oli jo asentanut osan Elastic-yrityksen tuotteista tullessani töihin. Asennetut tuotteet olivat datan jälkikäsittelyyn (Logstash), datan indeksointiin ja analysointiin (Elasticsearch), datan graafiseen esittämiseen (Kibana) ja halutut lisäosat (Elastalert ja head).

Työssä keskitytään edellä mainittujen ohjelmien säätämisen lisäksi asentamaan ja säätämään erilaisia datan lähettimiä (Beats-tuotteet).

Ympäristö johon työtäni tullaan käyttämään, koostuu testi- ja tuotantopalvelimista, joita on hallinnoitu virtualisointi alustalla, nimeltään VMware vSphere. Toimeksiantajani suurin tarve on seurata tuotantoympäristöjä. Tuotantoympäristö koostuu 39 Linux- ja Windows-ympäristöstä, joista suurin osa on asiakkaille myytyjä palveluita ja loput omia tuotantopalvelimia. Testi koostuu 53 Testipalvelimesta/työasemasta.

2 KEHITYSTYÖN TIETOPERUSTA

Tässä luvussa kerrotaan lokien perusteista, Big datasta. Koska toimeksiantaja oli valinnut Elastic-yrityksen tuotteet käytettäväksi, kerron niistä ja kilpailusta alalla.

2.1 Loki

Tietotekniikassa lokia tuotetaan lähes kaikkialla, ja viimeistään ongelmatilanteessa loki on myös korvaamatonta. Ongelmatilanteen tullessa lokeista on helppoa tarkastaa, mitä ja missä tapahtui tietyllä ajanjaksolla. Jokaisella järjestelmällä tai ohjelmalla pitäisi olla omat lokitietonsa, ja jotta lokit pysyvät eheinä, pitäisi muun muassa estää jälkikäsittely. Myös lokitietojen tarkastelu ja käsittely pitäisi lokiin merkitä. (Viestintävirasto, n.d.)

2.1.1 Lokin rakenne

Jotta loki olisi mahdollisimman hyvää ja tarkkaa, pitäisi löytyä ainakin seuraavat kohdat:

- aikaleima, koska tapahtuma tapahtui
- tapahtuma, mitä tapahtui tai yritettiin tehdä
- tekijä, kuka asiaa teki
- millaiset käyttöoikeudet hänellä oli
- miltä koneelta tai mistä tapahtuma tehtiin
- kohde, johon toiminta kohdistui
- onnistuiko tapahtuma vai ei.

Riippuen tarpeesta, lokissa voi/kannattaa olla myös lisätietoja. (Viestintävirasto, n.d.)

Lokeissa pitäisi välttää seuraavia tietoja:

- henkilötunnukset
- arkaluonteiset henkilötiedot
- luottokorttien numerot
- salasanat
- järjestelmien väliset tunnistetiedot
- valtuutustiedot
- viestiliikennettä henkilöiden väliltä.

Jos lokitus on huonoa tai vääränlaista, voidaan helposti rikkoa esimerkiksi yksilön tietosuojaa. Tästä syystä lokien käsittelyssä tulisi olla pääsynvalvonta, varmuuskopiointi, merkintä käsittelystä ja myös hälytyksen säätö. (Viestintävirasto, n.d.)

2.1.2 Käyttö

Lokien tallennusstahtia tai säilytysaikaa ei pystytä määrittelemään ennakkoon tarkasti, ja parhaiten säädöstä selviää kokeilemalla. Jos tallennustahti

on liian suuri, dataa saattaa kertyä liian paljon järjestelmälle, ja järjestelmä tukkiutuu. Jos tallennustahti on liian hidas, voi tärkeä tieto jäädä saamatta. Jos säilytysaika on suuri, dataa saattaa kertyä tarpeettoman paljon ja se kuluttaa levytilaa turhaan. Jos aika on liian pieni, voidaan helposti menettää tarvitut tiedot.

Erilaisia lokitasoja on useita erityyppisiä, riippuen hieman mitä lokittajaa ohjelma käyttää, mutta yleisimmät ovat SEVERE/FATAL/CRITICAL, ERROR, WARNING, INFO ja DEBUG.

Eri luokkaan kuuluvat lokit ja niiden yleiset merkitykset:
- SEVERE/FATAL/CRITICAL = Kriittinen virhe, joka aiheuttaa toimenpiteitä ja pitää korjata heti
- ERROR = Tapahtuma, jota kannattaa ruveta tutkimaan ja korjaamaan mahdollisimman pian
- WARNING = Tapahtuma, joka ei välttämättä aiheuta toimenpiteitä tutkimisen lisäksi
- INFO = Merkintä tapahtuneesta onnistuneesta toiminnosta, esimerkiksi ohjelman käynnistyminen ja sammuminen
- DEBUG = Pitäisi käyttää vain testaukseen tai ongelman tutkimiseen.

(Oracle, n.d.)

Lokitiedoilla on yleensä elinkaari, joka määrittelee, kuinka kauan tieto pysyy tärkeänä tai minkä jälkeen ei tiedolla tehdä mitään. Lokien poistamiselle pitäisi myös olla jokin menettely.

Vastuu lokien käsittelystä on viime kädessä organisaation ylimmällä johdolla, ensisijaisesti kuitenkin tehtävään nimetyt pääkäyttäjät. Jotta seurattavuus varmistuisi, yksittäisillä pääkäyttäjillä ei pitäisi olla muokkausoikeutta lokienhallintajärjestelmään. (Viestintävirasto, n.d.)

2.2 Big data

Big dataksi kutsutaan tietoa, joka koostuu suuren raakadatan analysoiduista tiedoista, joita voidaan reaaliajassa hyödyntää esimerkiksi nettisivujen käyttäjien hakuehdoituksissa tai kohdistetussa mainonnassa. Big data saattaa sisältää kaikenlaista dataa, esimerkiksi tekstiä, sensoritietoa, ääntä, videoita ja lokeja.

Datamäärät ovat kasvaneet viimeisien vuosien aikana merkittävästi, ja niiden ylläpitämisessä on suuri työ, jotta data on käytettävissä eikä vain raakadataa, jolla ei ilman suurta työtä tehdä mitään. Noin 90 % luodusta datasta on analysoimatonta.

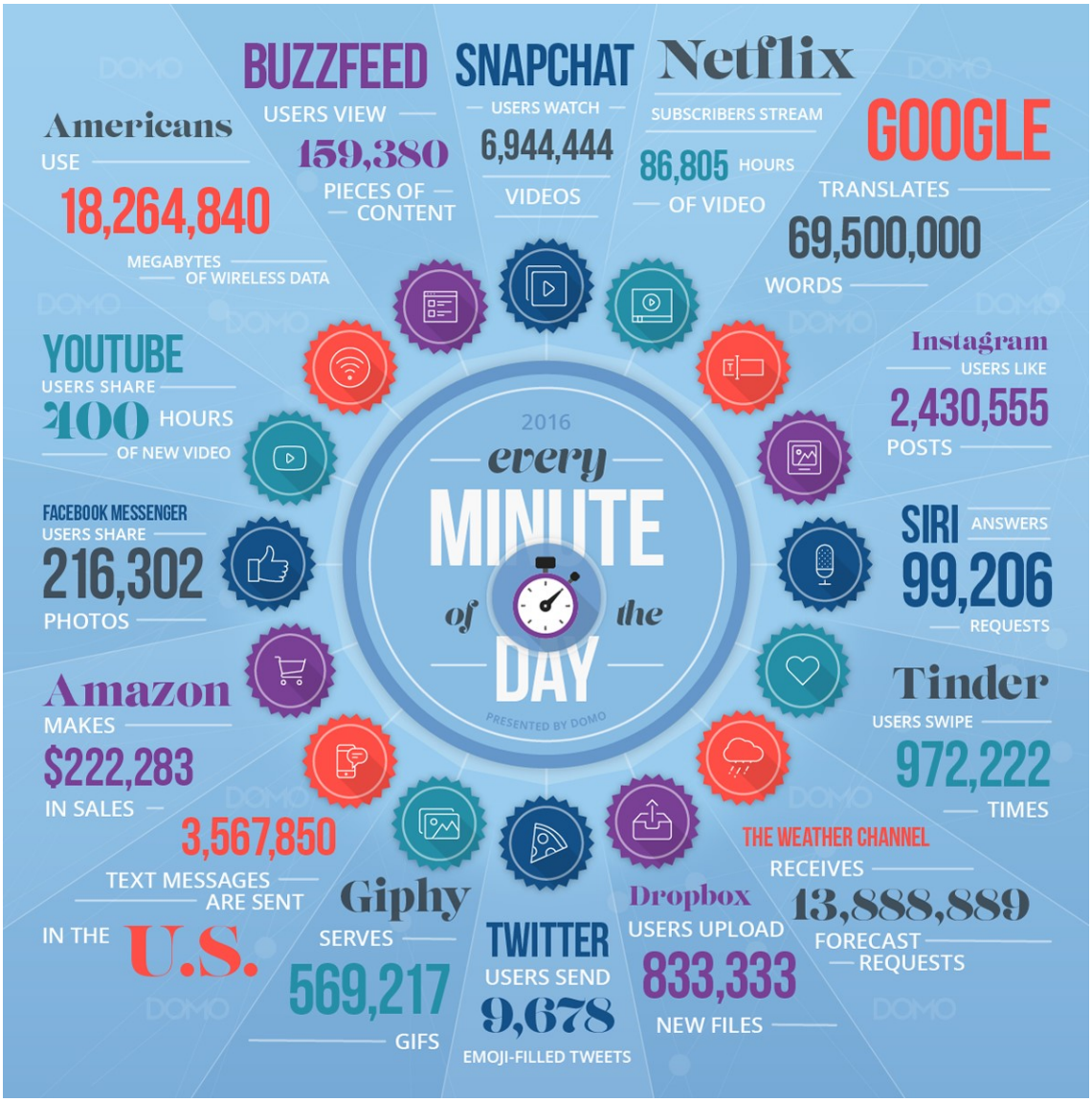
Vuonna 2013 tehdyn tutkimuksen mukaan vuosien 2011-2013 aikana, dataa oli syntynyt noin 90 % kaikesta vuoteen 2013 saakka syntyneestä datasta (Dragland, 2013).

Vuonna 2015 julkaistun verkkotekstin mukaan maailman datamäärä kasvaa noin 40 % vuodessa ja noin 50 kertaiseksi 2020 vuoteen mennessä (Waal-

Montgomery, 2015). Vuoden aikana maailmassa luodaan dataa noin 2,5 triljoonaa tavua, mikä on verrattavissa noin 10 miljoonaan yksikerroksiseen blu-ray levyyn (25GB) (Walker, 2015).

Erään lähteen mukaan datan määrä on kasvanut seuraavasti:
- Vuonna 1992 dataa on luotu 100GB/päivä
- Vuonna 1997 dataa on luotu 100GB/tunti
- Vuonna 2002 dataa on luotu 100GB/sekunti
- Vuonna 2013 dataa on luotu 28 875GB/sekunti
- Vuonna 2018 dataa tullaan luomaan arvion mukaan 50 000GB/sekunti (Walker, 2015)

Alla olevasta kuvasta (Kuva 1) selviää, kuinka paljon dataa sosiaalisessa mediassa luodaan minuutissa vuonna 2016.



Kuva 1. Datan määrä sosiaalisessa mediassa (James, 2016)

Big datan käsittelyyn tarvitaan tehokkaita työkaluja, jotta data saadaan analysoitua tehokkaasti. Yksi suosituimmista Big data-työkaluista on Apache Hadoop. Toimeksiantajan valitsemalla yrityksellä, Elasticilla, on tähän oma

sovelluksensa Elasticsearch for Hadoop, mistä kerron lisää luvun 2.3.1 kohdassa 10. Vaikkakin toimeksiantajalla merkintöjä syntyy vain kymmeniä tuhansia päivässä, voidaan ajattelutapaa ja toimintaa pitää hyvin Big data-maisena.

2.3 Elastic

Yksi perustajista, Shay Banon julkaisi Elasticsearchin ensimmäiset koodirivit Open-Source lisenssillä jakoon yleisenä hakukoneena jo vuonna 2010. Shay ennakoi tarpeen ohjelmistolle, jolla pystyisi hallinnoimaan suurta määrää dataa helposti. Steven Schuurman, Uri Boness ja Simon Willnauer jakoivat Shayn näkemyksen ja perustivat Shayn kanssa Elastic-yhtiön vuonna 2012 Amsterdamissa. Tämän jälkeen mukaan on lähtenyt myös Kibanan, Logstashin ja Beatsin luojat. (Elastic Co, n.d.)

Elasticin tuotteita käyttävät monen muun lisäksi esimerkiksi Netflix, Facebook, GitHub, Microsoft ja Activision, he käyttävät tuotteita esimerkiksi varmistaakseen mainostuksen toimivan, Microsoft Azuren hakutoimintona ja sosiaalisen tiedon muuttamiseen dataksi. (Elastic Co, n.d.)

2.3.1 Elastic-tuotteet

Tuotteita on mahdollista käyttää joko paikallisesti asentamalla tai Elasticin tarjoamassa virtuaaliympäristössä Software as a Service-palveluna.

Tuotteet on lajiteltu kahteen pakettiin, “The Elastic Stack” ja ”X-Pack”.

”The Elastic Stack” -paketti sisältää tuotteet, joilla saadaan jo täysin toimiva järjestelmä aikaiseksi. Pakettiin kuuluu neljä ohjelmaa:

1. Elasticsearch, indeksointi, analysointi ja säilytys
2. Kibana, datan graafiseen esittämiseen
3. Logstash, datan jälkikäsittely
4. Beats, datan lähettäminen ympäristöistä.

Ilmaislisenssin, avoimen lähdekoodin sijasta ottavalle tulee myös Marvel-tuotteesta monitorointi-ominaisuus käyttöön.

”X-Pack” -paketti sisältää edellä mainittujen lisäksi tuotteet, joilla palvelusta saadaan huomattavasti tehokkaampi. Yllä mainittujen tuotteiden lisäksi tulee viisi tuotetta:

1. Marvel, monitorointiin ja analysointiin
2. Shield, tietoturvallisuuteen
3. Watcher, hälytyksiin
4. Reporting, raportointiin
5. Graph, datasuhteiden analysointiin.

(Elastic Co, n.d.)

Lisäksi Elastic tarjoaa myös Elasticsearch for Hadoop, joka on tarkoitettu suuren Big datan käsittelyyn.

Kerron alla lisää yllä mainituista tuotteista.

1. Elasticsearch on hakukone-ohjelma, joka käyttää Apache Lucenen kirjastoa sisäisesti. Luo dokumenteista ”varaston”, jossa jokainen dokumentin kenttä on indeksoitu, analysoitu ja haettavissa. Tulen myöhemmin työssäni (Luku 4.1) kertomaan Elasticsearchin käytöstä. (Elastic Co, n.d.)

2. Logstash on ohjelma, jolla voidaan keskittää kaiken datan prosessointi ja esimuokkaus ennen kuin se lähetetään Elasticsearchiin. Ympäristöstä mistä dataa halutaan lähettää, pitää asentaa logstash forwarder. (Elastic Co, n.d.)

3. Kibana on ohjelma joka muuttaa Elasticsearchin ”varaston” visuaaliseksi tiedoksi. Tietoa voi muokata, etsiä ja näyttää miten haluaa. Tulen myöhemmin työssäni (Luku 4.2) kertomaan Kibanan käytöstä. (Elastic Co, n.d.)

4. Beats on kevyt datan siirto ohjelma, jolla voidaan lähettää tietoa suoraan Elasticsearchiin tai logstashin kautta Elasticsearchiin.

   Se jakautuu kuuten erilaiseen Beatsiin, jotka ovat:
- Filebeat, joka lähettää halutut lokitiedot Logstashiin jatkoprosessoitavaksi, tai suoraan Elasticsearchiin anyloistavaksi.
- Metricbeat on yhdistelmä Packetbeat-, Winlogbeat- ja Topbeat -ohjelmista. Metricbeat lähettää tiedon tietokoneesta, prosesseista, tietoliikenteestä, Apachen moduuleista, tietokannoista, Ngix-palvelusta, Redis-palvelusta ja Zookeeper-palvelusta.
- Packetbeat, joka lähettää tietoliikenteestä tiedon suoraan Logstashiin tai Elasticsearchiin. Packetbeat esimerkiksi seuraa http-liikennettä tietyissä porteissa ja indeksoi tiedon ja muodostaa tästä tietopaketin, joka lähetetään eteenpäin.
- Winlogbeat, joka lähettää Windowsin event-tiedot eteenpäin, suoraan Elasticsearchiin tai Logstashiin.
- Topbeat, joka kerää ja lähettää tietoja tietokoneen prosessorin, muistin, tallennustilan ja ohjelmien tilasta suoraan Elasticsearchiin tai Logstashiin.
- Libbeat, joka on pohja uudelle beatsille, jos haluaa itse tehdä omaan tarpeeseen paremman. (Elastic Co, n.d.)

Tulen kertomaan myöhemmin työssäni Filebeatin (Luku 4.3.4), Packetbeatin (Luku 4.3.3), Winlogbeatin (Luku 4.3.1) ja Topbeatin (Luku 4.3.2) käytöstä.

5. Marvel on Elasticsearchin monitorointi- ja analysointi-sovellus. Reaaliaikaisen tietojen keräämisen lisäksi Marvel säilyttää myös vanhat suorituskyky tiedot. Marvelilla pystytään katsomaan esimerkiksi indeksoinnin viiveettä tai haun tehokkuutta reaaliajassa tai vertailla sitä menneeseen tietoon (Elastic Co, n.d.).

6. Shield on turvallisuusohjelma Elastic-tuotteisiin. Shieldillä saadaan Elasticsearchin klustereihin salasanasuojaus, roolipohjaisen pääsyn hallinta, IP-suodatus, käyttäjien seuranta, viestien autentikointi ja SSL/TLS-salaus. Shieldin liitännäisen avulla Kibanaan saadaan erillinen kirjautuminen, jonka avulla voidaan määritellä kunkin käyttäjän käyttöoikeudet. Ohjelmaan tunnistautuminen tapahtuu oletuksena käyttäjänimi + salasana yhdistelmällä, mutta jos järjestelmä on monimutkaisempi ja tarvitaan parempaa tietoturvaa, voidaan käyttäjät sallia myös, Shieldin hallituista rooleista, käyttämällä PKI-sertifikaatteja, IP-suodatuksen kautta tai integroida LDAP:in tai Active Directoryn:n kautta. (Elastic Co, n.d.)

7. Watcher-ohjelmalla luodaan haku Elasticsearchiin ja luodaan siitä halutunlainen hälytys. Pystyy myös luomaan ”liipaisimia” jotka laukaisevat seuraavan ”watcherin”. (Elastic Co, n.d.)

8. Reporting-ohjelmalla saadaan kaikista Kibanassa tehdyistä visuaalisoinneista tai näkemyksistä tehtyä raportit. Raportin luonnin voi myös ajastaa ja käskeä lähettämään raportti sähköpostilla. Raportit ovat optimoituja tulostamiseen, muokattavissa ja PDF-formaatissa. (Elastic Co, n.d.)

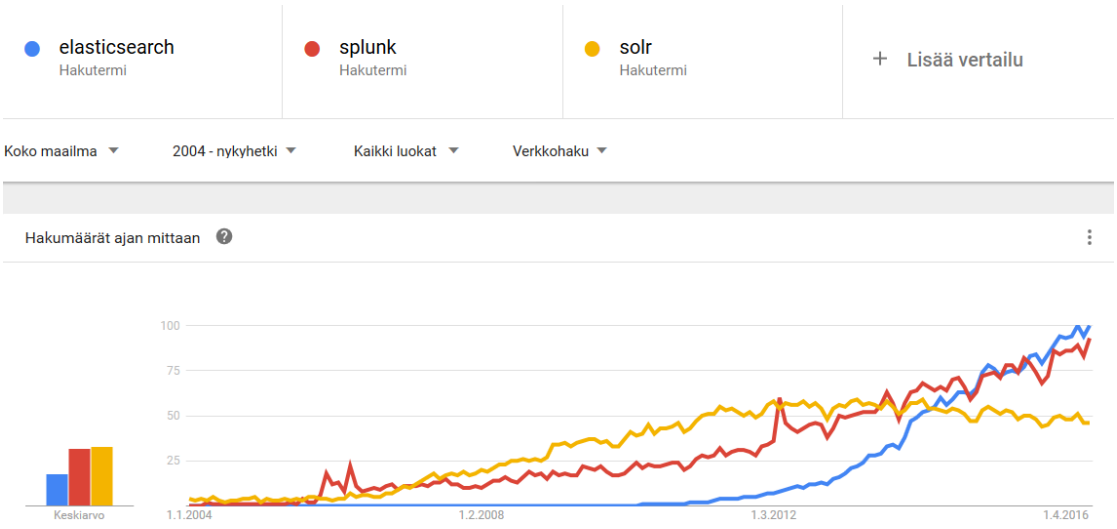
9. Graph on ohjelma datojen suhteiden analysointiin. Suhteiden tutkiminen tapahtuu Kibanassa visuaalisesti. Se helpottaa epäilyttävien tapahtumien löytämistä ja tuottaa myös reaaliaikaisen suosittelun. (Elastic Co., n.d.)

10. Elasticsearch for Hadoop on tarkoitettu todellisen Big datan analysointiin. ES-Hadoopin on kehitetty tuomaan Elasticsearchin ja hadoopin tärkeimmät ominaisuudet yhteen, eli Elasticsearchin reaaliaikainen haku ja Hadoopin Big datan tehokas analysointi. Tarkoitettu suurelle datamassalle.

2.4 Elasticsearchin kilpailijat

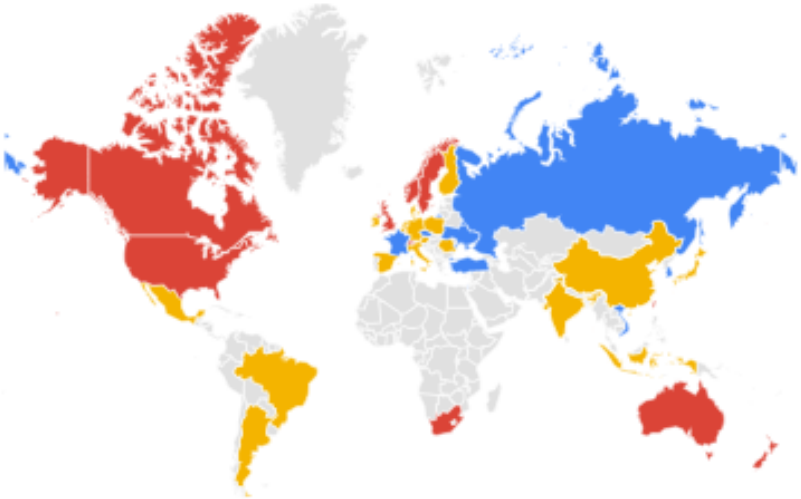
Elasticsearch on järjestelmän tärkein ohjelma, koska kyseinen ohjelma indeksoi ja analysoi datan. Tästä syystä vertailu tehdään kyseiseen tuotteeseen.

Kilpailijoita haku-alusta-alalla on paljon, mutta suosituimpia ovat Elasticsearch, Splunk ja Apache Solr. Alla kuva tuotteiden hakumääristä (Kuva 2).



Kuva 2. Hakutulokset. (Google, n.d.)

Elasticsearch on kotoisin Hollannista, mistä johtuen ohjelma on Euroopassa suosittu, mutta esimerkiksi Pohjois-Amerikassa melkein tuntematon. Alla kuva ohjelmien suosiosta ympäri maailman (Kuva 3).



Kuva 3. Ohjelmien suosio maailmalla. (Google, n.d.)

Koska toimeksiantajani oli valinnut Elasticsearchin käytettäväksi ohjelmaksi, kohdistuu vertailut Elasticsearchiin.

1. Splunk

Pakettiin kuuluu vastaavat ohjelmistot kuin Elastic-yrityksen tuotteet Elasticsearch, Logstash ja Kibana. Ohjelmia on helppo ja yksinkertainen käyttää. Yritys perustettu vuonna 2003.

Vertailua Elasticsearchiin:
- ohjelma on maksullinen
- asiakkaista suuret yritykset etusijalla

- ohjelmaan tulee vähemmän säätöä, jos käyttäjä tai osasto määrät kasvavat
- ohjelman light versiossa maksimi määrät käyttäjille ja tiedon määrälle
- ohjelma huomattavasti yksinkertaisempi ja helpompi käyttää, kuin Elasticsearch
- järjestelmässä tieto yksilöidään vasta järjestelmässä eikä ennen lähetystä niin kuin Elasticsearch/beats yhtälössä
- ohjelmasta saadaan vietyä näkemyksiä PDF-tiedostoiksi (Minkä saa Elasticsearchiin maksusta).

(Zhitnitsky, n.d.)

2. Apache Solr

Tunnettu ja suosittu avoimen lähdekoodin haku-alusta, joka käyttää Apache Lucenea. Julkaistu vuonna 2008 ja SolrCloud-pilvipalvelu julkaistiin vuonna 2012. (Apache, n.d.)

Vertailua Elasticsearchiin:

- Solr mahdollisti shardien jakamisen vuonna 2013, ilman että tarvitsee indeksoida kaikkea uudestaan
- Solr on ollut pidempään markkinoilla kuin Elasticsearch, joten yhteisö on suurempi ja tietoa löytyy helpommin
- Solr käyttää XML-, CSV- ja JSON-muotoja
- Solrissa Java Management Extension-tuki
- molemmat ohjelmista ovat ilmaisia
- Solriin ei pysty asentamaan lisäosia esimerkiksi suoraan GitHubista
- Solr tarvitsee palvelimelle ZooKeeper-palvelimen toimiakseen.
- Solrissa web-käyttöliittymä mukana.

(Tan, n.d.) & (Think Big Analytics, n.d.)

Hyvin samankaltainen Elasticsearchin kanssa.

3 KEHITTÄMISTYÖN TAVOITE JA TARKOITUS

Opinnäytetyön tarkoitus on helpottaa SAAS-palveluiden ja omien palvelimien seurantaa. Palvelimilta seurataan yrityksen tuotteiden tuottamia lokeja, Windows-event tietoja ja ympäristön laitteistotason tietoja. Palvelimien seuranta on ennen tapahtunut siten, että on täytetty Excel-dokumenttia palvelimilta käsin (Laitteistotaso) tietyin aikavälein tai asiakas/työntekijä on ilmoittanut asiasta. Tuotteiden lokeja on seurattu myös suoraan palvelimilta, ja yleensä vasta kun joku taho on ongelmasta ilmoittanut.

Elasticin palveluista työnantajani oli jo asentanut Kibanan, Logstashin ja Elasticsearchin, johon oli lisäksi asennettu head- ja elastalert-lisäosat.

Tässä luvussa kerrotaan työn aloittamisesta, suunnittelusta ja toteuttamisesta.

3.1 Aloitus

Toimeksiantajalla oli todellinen tarve saada ympäristöjen ja/tai tuotteiden seurantaa yksinkertaisemmaksi ja tehokkaammaksi, jotta työaikaa kuluisi tähän mahdollisimman vähän, myös ongelmiin olisi hyvä puuttua jo mahdollisesti ennen kuin asiakas ongelmaa huomaa.

Koska toimeksiantajan datamäärät olivat pienet ja merkintöjä syntyi päivässä vain kymmeniä tuhansia, heidän tarpeisiinsa riitti pelkkä Elasticsearch eikä tarvittu esimerkiksi ES-Hadooppia.

Aiheen varmistuttua aloin tutustua Elastic-yrityksen tuotteisiin ja niiden mahdollisuuksiin. Tutustumisen yhteydessä kävi ilmi, että tarpeisiimme riittävät jo löytyvät Elasticsearch ja Kibana, ja näiden lisäksi erilaiset Beats-ohjelmat.

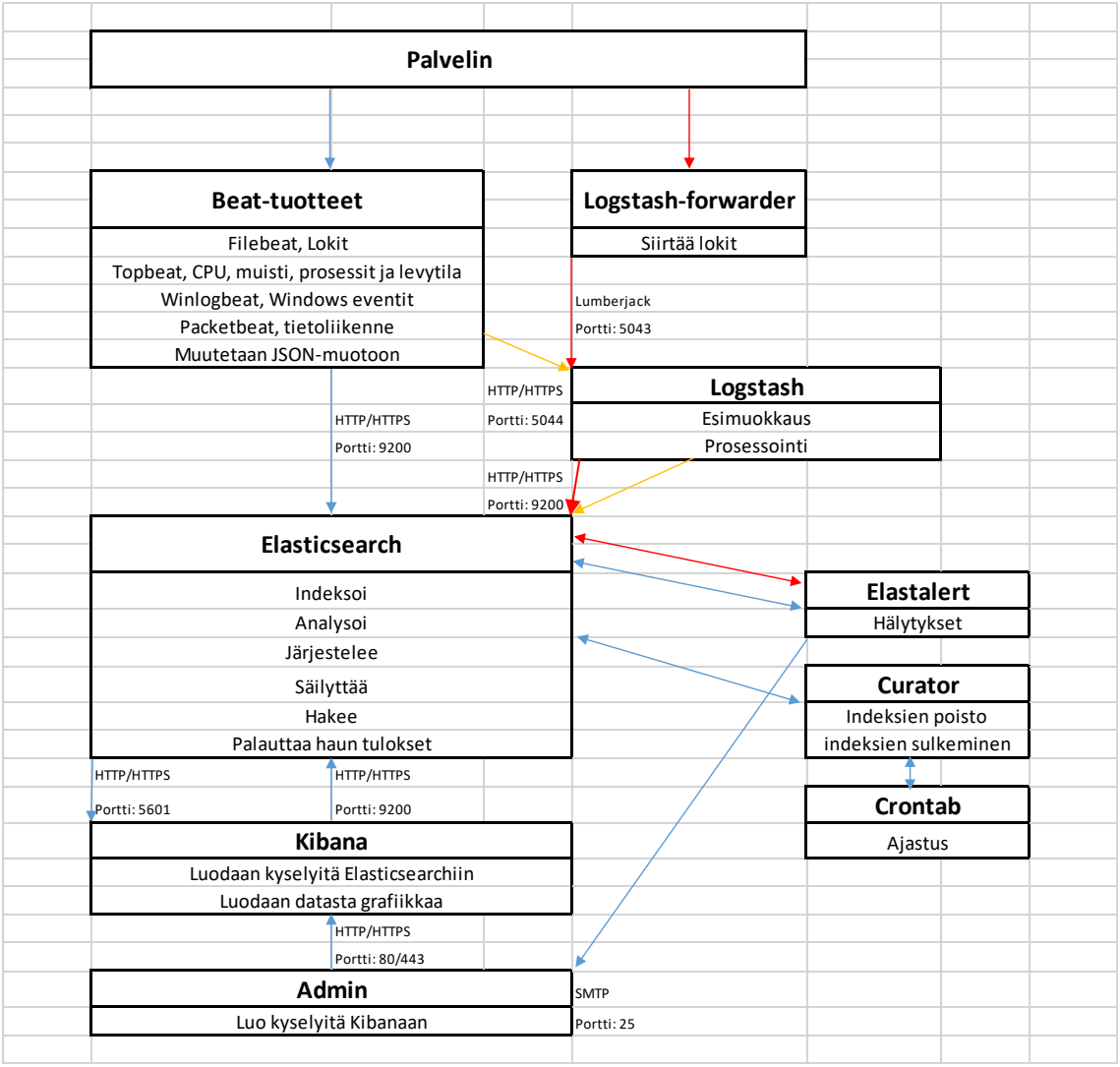
Aiheesta pidettiin useampi palaveri, joissa käytiin läpi sitä, millaisia tarpeita kullakin käyttäjällä ohjelmaan olisi. Tarpeet vaihtelivat omien lokien tarkkailusta ja seurannasta ympäristöjen tietoihin. Kibanan käyttäjinä tulisi ensisijaisesti olemaan tekniikan-työntekijät, mutta järjestelmällä on tarvetta myös ohjelmistokehittelijöillä ja helpdeskille.

Vanha toimintaperiaate on todettu hitaaksi ja tehottomaksi, varsinkin seurannan näkökulmasta. Uudella järjestelmällä saataisiin huomattava parannus ympäristöjen seurantaan, asiakaspalveluun tehokas apu ongelmanratkonnan tueksi ja ohjelmistosuunnittelijoille hyvä tapa seurata omien ohjelmien lokeja.

3.2 Suunnittelu

Alkuperäinen suunnitelma selviää alla olevasta kuvasta (Kuva 4), punainen reitti. Kuten edellisessä luvussa (Luku 3.1) mainittiin, alkuperäisestä suunnitelmasta poikettiin ja todettiin, että Beats-tuotteilla saadaan monipuolisemmin ja helpommin dataa kuin logstash- ja logstash-forwarder-ohjelmilla. Tästä syystä suunniteltiin sinisen reitin mukainen ympäristö. Mahdollisesti tiedon määrän kasvaessa joudutaan siirtymään käyttämään oransseja reittiä.

Ympäristöille haluttiin myös hälytyksiä, jotka lähettävät tietyille ryhmille sähköpostia, jos jokin arvo täyttyy.



Kuva 4. Toimintamalli

Yllä olevassa kuvassa (Kuva 4), näkyvät myös protokollat ja portit, joita ympäristössä käytetään.

Ohjelmien asentaminen suunniteltiin niin, että testipuolelle tulee vain Topbeat ja jos tietyistä testiympäristöistä halutaan seurata lokeja, niin Filebeat lisäksi. Topbeat säädetään lähettämään mahdollisimman vähän tietoa, mutta silti tarpeeksi, että seurantaa voi joka päivä tehdä.

Tuotantopuolelle tulee Filebeat, Topbeat, Winlogbeat ja mahdollisesti Packetbeat (Jos halutaan seurata esimerkiksi omien nettisivujen käyttöä). Beat-tuotteet säädetään lähettämään tietoa mahdollisimman vähän, mutta silti niin paljon, että voidaan lähes reaaliajassa tietoa tarkkailla.

Kibanaan haluttiin näkemykset, joista olisi helppo ja tehokasta tarkkailla niin testi- kuin tuotanto-ympäristöjen levytilaa, sekä prosessorin ja muistin käyttöä. Tuotantopalvelimista haluttiin myös Windows eventit ja oman tuotteen lokit seurattaviksi. Toimeksiantajan tekniikan-tiloihin haluttiin näytölle näkyviin näkemys, josta selviäisi pelkällä vilkaisulla ympäristöjen ja tuotteiden tilat.

3.3 Toteuttaminen

Ohjelmiin perehtymisen jälkeen ohjelmia aloitettiin asentamaan testiympäristöihin.

Asentamisen jälkeen tuotteiden käyttäytymistä ja mahdollisuuksia tutkittiin ja todettiin ohjelmien lähettävän oletuksena liikaa dataa, ainakin testi-ympäristöistä. Usean konfiguraatio-testin jälkeen todettiin alla olevien säätöjen olevan lähellä haluttuja arvoja, jotta data pysyy tarpeeksi reaaliaikaisena, mutta silti indeksit pysyvät tarpeeksi pieninä.

Ohjelmien säätäminen toteutettiin niin, että testipuolelta Topbeat (laitteistotieto) lähettää tietoja vain kuuden tunnin välein ja tuotantopuolella maksimissaan 30 minuutin välein. Tuotantopuolen muut beat-tuotteet (Winlog-, file- tai packetbeat) lähettävät tietonsa aina kun muutoksia tapahtuu tai maksimissaan 10 minuutin välein.

Lopulliset säädöt tehdään datamäärän kasvaessa ja järjestelmän käytön lisääntyessä, jotta tiedetään, kuinka paljon dataa halutaan tai kuinka kauan sitä halutaan säilyttää.

Säilytysajan säätely toteutettiin Curator-nimisellä ohjelmalla, johon säädettiin ajat, minkä jälkeen indeksit joko suljettiin tai poistettiin. Curator ajetaan päivittäin käyttäen Crontab-ohjelmaa, mistä kerron lisää luvussa 4.1.3. Curator-ohjelmasta kerron lisää luvussa 4.1.2

Kibanaan luotiin työntekijöiden kuvauksien perusteella useampi näkemys ja visualisointi eri tarpeisiin.

Visualisoinnit toteutettiin tekemällä erilaisia visualisointeja pylväsdiagrammeista lähtien, pelkkiin lukumääriin saakka (Esimerkiksi merkintöjen lukumäärä).

Näkemyksissä etusijalla oli käyttäjien tarpeet. Yksi näkemys, mistä selviää palvelimien resurssit ja Windows eventit, luotiin työntekijälle, joka on tarkkaillut ennenkin ympäristöjä. Tekniikan-tiloihin tulevaan näyttöön tehtiin oma näkemys, mistä selvisi palvelimien resurssien lisäksi viimeisimmät windows eventit ja omien tuotteiden lokien merkinnät joissa tasona vakavammat kuin INFO (katso tasot kappaleesta 2.1.2). Ohjelmistokehittäjille ei ole vielä näkemystä tehty, koska omien tuotteiden lokiin tarvitaan vielä muutoksia, samaisesta syystä helpdesk-työntekijöillekkään ei ole näkemystä tehty.

Hälytykset koko ympäristöön saatiin luotua Elastalert-ohjelmalla, joka oli jo asennettuna työtä aloitettaessa. Ohjelmaan luotiin halutut hälytykset, joista toinen tarkkailee kiintolevyn käyttöastetta ja toinen tarkkailee lokimerkintöjä. Lisää Elastalert-ohjelmasta luvussa 4.1.4

Nykyinen ympäristö on kuvattu yllä olevassa kuvassa (Kuva 4), sininen reitti.

4 TUOTOKSEN SÄÄTÄMINEN JA KÄYTTÄMINEN

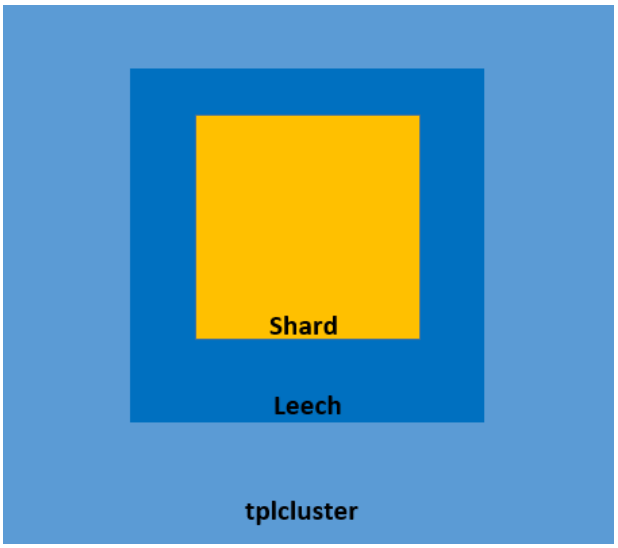
Luvussa käydään läpi käytettyjen tuotteiden asentamista, säätämistä ja käyttöä.

4.1 Elasticsearch 1.7.2

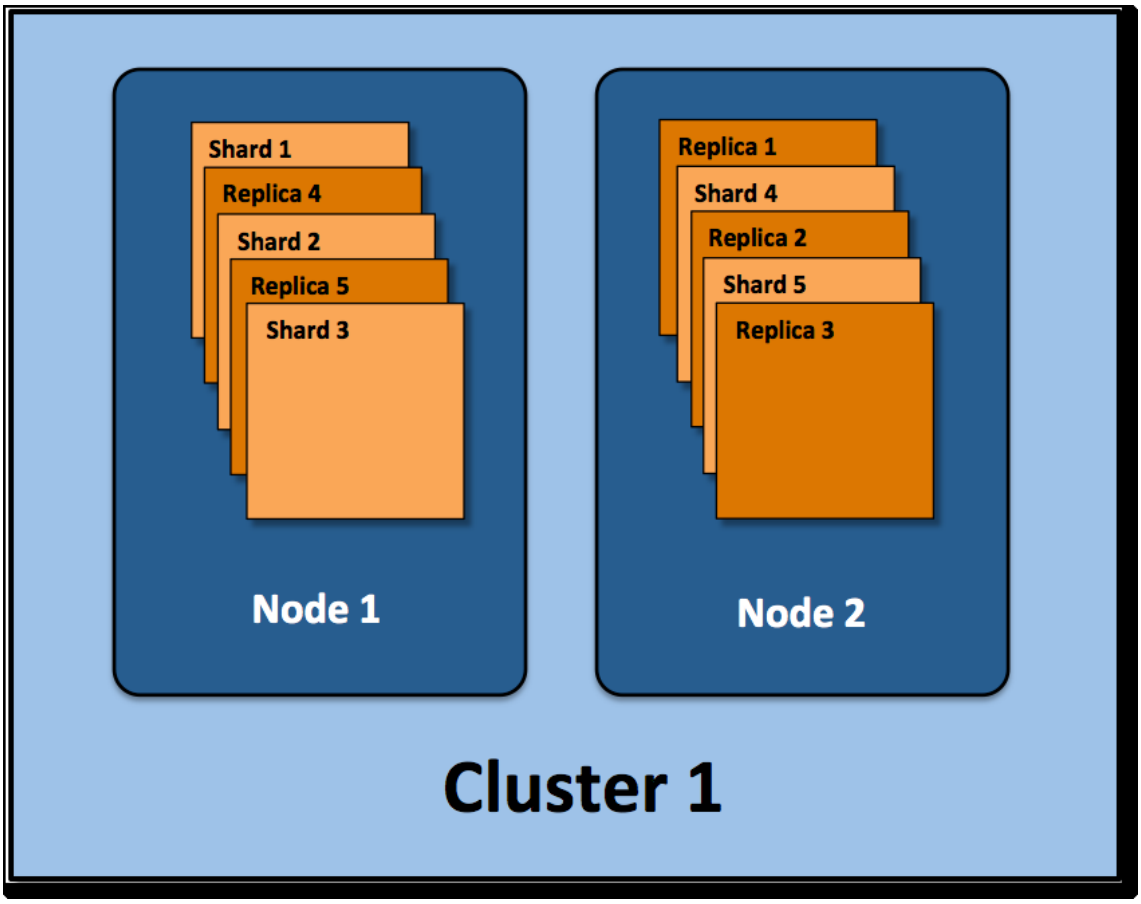
Kyseessä on ohjelma, joka hoitaa datan analysoinnin ja luo ”datavaraston”, johon Kibana luo hakuja ja hakee tiedot itsellensä.

Aloittaessani työni oli Elasticsearch, Elasticsearch-head ja Elastalert-lisäosat jo asennettu toimintaan. Elasticsearch-head-lisäosalla sai Elasticsearchiin verkkokäyttöliittymän. Elastalert-ohjelmalla saadaan tehtyä halutuista asioista hälytyksiä, jotka lähettävät ilmoitukset esimerkiksi sähköpostiin. Elasticsearch-headista ja Elastalertista kerron alla olevissa luvuissa (Luvut 4.1.1 ja 4.1.4) lisää.

Klusteri on oletuksena alla olevan kuvan (Kuva 6) mukaisesti, mutta vain yhdellä nodella. Toimeksiantajalla klusteria on muokattu niin, että shardeja on vain yksi eikä yhtäkään replikaa (Kuva 5).



Kuva 5. Toimeksiantajan klusteri

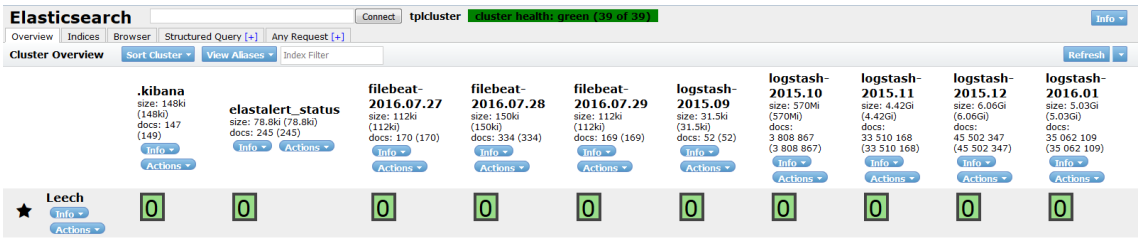


Kuva 6. Klusterin oletusrakenne kahdella nodella. (Hundley, 2015)

4.1.1 Elasticsearch-head

1. Overview-välilehti

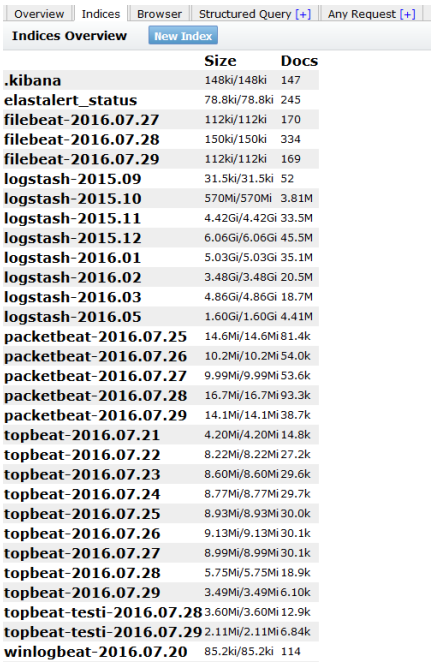
Käyttöliittymän Overview-välilehdeltä (Kuva 7) näkymästä näkee suoraan klusterin (tplcluster) tilan, indeksit (filebeat, logstash jne.) ja noden (Leech) tilanteen. Jokaisesta indeksistä voi tarkistaa statuksen ja metatiedot, jotka näkyvät JSON-muotoisena. Jokaisella indeksillä on toiminnot, uudelleen nimeäminen, päivittäminen, ajaminen, optimointi, ”snapshotin” tekeminen, analysoijan testaaminen, sammuttaminen ja poistaminen. Myös Nodesta saa JSON-muotoiset yhteenvedot ja sen pystyy sammuttamaan.



Kuva 7. Elasticsearch-head lisäosan Overview-välilehti.

2. Indices-välilehti

Indices-välilehdeltä näkee indeksit, niiden koot ja dokumenttien määrän (Kuva 8).

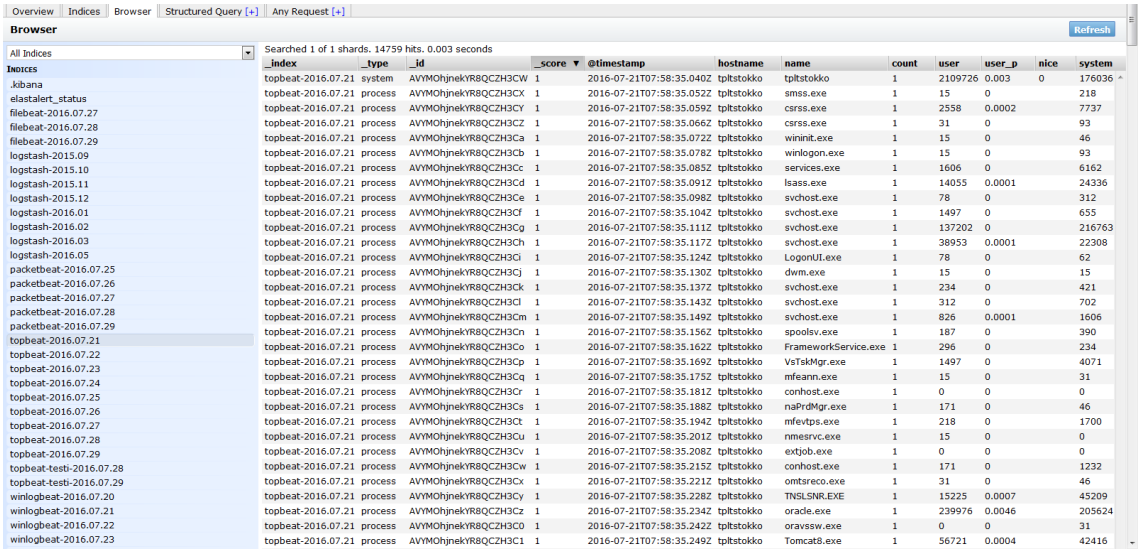


	Size	Docs
.kibana	148ki/148ki	147
elastalert_status	78.8ki/78.8ki	245
filebeat-2016.07.27	112ki/112ki	170
filebeat-2016.07.28	150ki/150ki	334
filebeat-2016.07.29	112ki/112ki	169
logstash-2015.09	31.5ki/31.5ki	52
logstash-2015.10	570Mi/570Mi	3.81M
logstash-2015.11	4.42Gi/4.42Gi	33.5M
logstash-2015.12	6.06Gi/6.06Gi	45.5M
logstash-2016.01	5.03Gi/5.03Gi	35.1M
logstash-2016.02	3.48Gi/3.48Gi	20.5M
logstash-2016.03	4.86Gi/4.86Gi	18.7M
logstash-2016.05	1.60Gi/1.60Gi	4.41M
packetbeat-2016.07.25	14.6Mi/14.6Mi	81.4k
packetbeat-2016.07.26	10.2Mi/10.2Mi	54.0k
packetbeat-2016.07.27	9.99Mi/9.99Mi	53.6k
packetbeat-2016.07.28	16.7Mi/16.7Mi	93.3k
packetbeat-2016.07.29	14.1Mi/14.1Mi	38.7k
topbeat-2016.07.21	4.20Mi/4.20Mi	14.8k
topbeat-2016.07.22	8.22Mi/8.22Mi	27.2k
topbeat-2016.07.23	8.60Mi/8.60Mi	29.6k
topbeat-2016.07.24	8.77Mi/8.77Mi	29.7k
topbeat-2016.07.25	8.93Mi/8.93Mi	30.0k
topbeat-2016.07.26	9.13Mi/9.13Mi	30.1k
topbeat-2016.07.27	8.99Mi/8.99Mi	30.1k
topbeat-2016.07.28	5.75Mi/5.75Mi	18.9k
topbeat-2016.07.29	3.49Mi/3.49Mi	6.10k
topbeat-testi-2016.07.28	3.60Mi/3.60Mi	12.9k
topbeat-testi-2016.07.29	2.11Mi/2.11Mi	6.84k
winlogbeat-2016.07.20	85.2ki/85.2ki	114

Kuva 8. Elasticsearch-head, indices-välilehti.

3. Browser-välilehti

Browser-välilehdeltä näkee kaikki indeksit ja niiden sisältämän tiedon (Kuva 9).

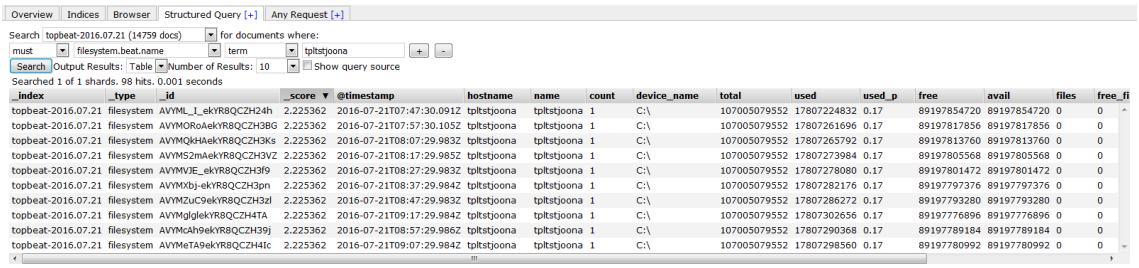


_index	_type	_id	_score	@timestamp	hostname	name	count	user	user_p	nice	system
topbeat-2016.07.21	system	AVYMOhjnekYR8QCZH3CW	1	2016-07-21T07:58:35.040Z	tpltstokko	tpltstokko	1	2109726	0.003	0	176036
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3CX	1	2016-07-21T07:58:35.052Z	tpltstokko	smss.exe	1	15	0		218
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3CY	1	2016-07-21T07:58:35.059Z	tpltstokko	csrss.exe	1	2558	0.0002		7737
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3CZ	1	2016-07-21T07:58:35.066Z	tpltstokko	csrss.exe	1	31	0		93
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Ca	1	2016-07-21T07:58:35.072Z	tpltstokko	wininit.exe	1	15	0		46
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cb	1	2016-07-21T07:58:35.078Z	tpltstokko	winlogon.exe	1	15	0		93
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cc	1	2016-07-21T07:58:35.085Z	tpltstokko	services.exe	1	1606	0		6162
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cd	1	2016-07-21T07:58:35.091Z	tpltstokko	lsass.exe	1	14055	0.0001		24336
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Ce	1	2016-07-21T07:58:35.098Z	tpltstokko	svchost.exe	1	78	0		312
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cf	1	2016-07-21T07:58:35.104Z	tpltstokko	svchost.exe	1	1497	0		655
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cg	1	2016-07-21T07:58:35.111Z	tpltstokko	svchost.exe	1	137202	0		216763
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Ch	1	2016-07-21T07:58:35.117Z	tpltstokko	svchost.exe	1	38953	0.0001		22308
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Ci	1	2016-07-21T07:58:35.124Z	tpltstokko	LogonUI.exe	1	78	0		62
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cj	1	2016-07-21T07:58:35.130Z	tpltstokko	dwm.exe	1	15	0		15
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Ck	1	2016-07-21T07:58:35.137Z	tpltstokko	svchost.exe	1	234	0		421
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cl	1	2016-07-21T07:58:35.143Z	tpltstokko	svchost.exe	1	312	0		702
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cm	1	2016-07-21T07:58:35.149Z	tpltstokko	svchost.exe	1	826	0.0001		1606
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cn	1	2016-07-21T07:58:35.156Z	tpltstokko	spoolsv.exe	1	187	0		390
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Co	1	2016-07-21T07:58:35.162Z	tpltstokko	FrameworkService.exe	1	296	0		234
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cp	1	2016-07-21T07:58:35.169Z	tpltstokko	VsTskMgr.exe	1	1497	0		4071
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cq	1	2016-07-21T07:58:35.175Z	tpltstokko	mfeann.exe	1	15	0		31
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cr	1	2016-07-21T07:58:35.181Z	tpltstokko	conhost.exe	1	0	0		0
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cs	1	2016-07-21T07:58:35.188Z	tpltstokko	naPrdMgr.exe	1	171	0		46
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Ct	1	2016-07-21T07:58:35.194Z	tpltstokko	mfevtps.exe	1	218	0		1700
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cu	1	2016-07-21T07:58:35.201Z	tpltstokko	nmesrvc.exe	1	15	0		0
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cv	1	2016-07-21T07:58:35.208Z	tpltstokko	extjob.exe	1	0	0		0
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cw	1	2016-07-21T07:58:35.215Z	tpltstokko	conhost.exe	1	171	0		1232
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cx	1	2016-07-21T07:58:35.221Z	tpltstokko	omtsreco.exe	1	31	0		46
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cy	1	2016-07-21T07:58:35.228Z	tpltstokko	TNSLSNR.EXE	1	15225	0.0007		45209
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3Cz	1	2016-07-21T07:58:35.234Z	tpltstokko	oracle.exe	1	239976	0.0046		205624
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3C0	1	2016-07-21T07:58:35.242Z	tpltstokko	oravssw.exe	1	0	0		31
topbeat-2016.07.21	process	AVYMOhjnekYR8QCZH3C1	1	2016-07-21T07:58:35.249Z	tpltstokko	Tomcat8.exe	1	56721	0.0004		42416

Kuva 9. Elasticsearch-head, browser-välilehti.

4. Structured Query-välilehti

Structured Query-välilehdellä pystyy tekemään hakuja haluttuun indeksiin, ja tulostamaan tiedot taulukkoon, JSON- tai CSV-muotoisena (Kuva 10).



Kuva 10. Elasticsearch-head, Structuder Query-välilehti.

5. Any request-välilehti

Any Request-välilehdeltä voidaan indekseille tehdä erilaisia käskyjä. Esimerkiksi poistaa kaikki tietyn kuukauden indeksit kerralla (Kuva 11).

Kuva 11. Any Request-välilehti, jossa komento jolla poistaisi kaikki winlogbeat indeksit heinäkuulta.

4.1.2 Curator 3.5.1

Linux-palvelimelle, jossa Elasticsearch sijaitsee, on asennettu myös Curator-palvelu, joka helpottaa indeksien ylläpitoa, myös suoraan palvelimelta käsin.

Asennus tapahtui komennolla *pip install elasticsearch-curator==3.5.1.*

Asennuksen jälkeen Curator oli käyttövalmis. Curatorin komennot joita tarvitsimme, olivat *show, delete* ja *close*. Curatorin käyttö oli helppoa heti alusta ja komennot olivat yksinkertaisia.

Show-komento, jolla näkee kaikki halutut indeksit: *Curator show indices –regex ”mitähalutaannähdä”*.

Delete-komento, jolla poistetaan yli 30 päivää vanhat indeksit: *curator delete indices --older-than 30 --time-unit days --timestring %Y.%m.%d --regex “mitähalutaanpoistaa”*.

Close-komento, joka sulkee yli viisi päivää vanhat indeksit: *curator close indices --older-than 5 --time-unit days --timestring %Y.%m.%d --regex “Mitkähalutaansulkea”*.

Palvelimelle tehty shell-tiedosto, joka sulkee ja poistaa testi- ja tuotantoindeksit. Testipuolella indeksit suljetaan, kun indeksit ovat neljä päivää vanhoja ja yli kahdeksan päivää vanhat poistetaan (Kuva 12). Tuotantopuolella indeksejä säilytetään hieman kauemmin ja indeksit suljetaan vasta 10 päivän jälkeen ja poistetaan vasta 20 päivää vanhat.

```
#!/bin/bash
curator close indices --older-than 10 --time-unit days --timestring %Y.%m.%d --regex topbeat-tuotanto*
curator close indices --older-than 4 --time-unit days --timestring %Y.%m.%d --regex topbeat-testi*
curator close indices --older-than 10 --time-unit days --timestring %Y.%m.%d --regex packetbeat-tuotanto*
curator close indices --older-than 4 --time-unit days --timestring %Y.%m.%d --regex packetbeat-testi*
curator close indices --older-than 10 --time-unit days --timestring %Y.%m.%d --regex winlogbeat-tuotanto*
curator close indices --older-than 4 --time-unit days --timestring %Y.%m.%d --regex winlogbeat-testi*
curator close indices --older-than 10 --time-unit days --timestring %Y.%m.%d --regex filebeat-tuotanto*
curator close indices --older-than 4 --time-unit days --timestring %Y.%m.%d --regex filebeat-testi*
curator delete  indices --closed-only --older-than 20 --time-unit days --timestring %Y.%m.%d --regex topbeat-tuotanto*
curator delete  indices --closed-only --older-than 8 --time-unit days --timestring %Y.%m.%d --regex topbeat-testi*
curator delete  indices --closed-only --older-than 20 --time-unit days --timestring %Y.%m.%d --regex packetbeat-tuotanto*
curator delete  indices --closed-only --older-than 8 --time-unit days --timestring %Y.%m.%d --regex packetbeat-testi*
curator delete  indices --closed-only --older-than 20 --time-unit days --timestring %Y.%m.%d --regex winlogbeat-tuotanto*
curator delete  indices --closed-only --older-than 8 --time-unit days --timestring %Y.%m.%d --regex winlogbeat-testi*
curator delete  indices --closed-only --older-than 20 --time-unit days --timestring %Y.%m.%d --regex filebeat-tuotanto*
curator delete  indices --closed-only --older-than 8 --time-unit days --timestring %Y.%m.%d --regex filebeat-testi*
```

Kuva 12. Shell-tiedoston sisältö

Tiedosto on ajastettu ajettavaksi joka päivä kello 7 käyttämällä crontab-ohjelmaa (Kuva 13), mistä kerron lisää seuraavassa luvussa (Luku 4.1.3).

```
0 7 * * * /root/curator/boot.sh
```

Kuva 13. Crontab-ohjelman asetukset.

4.1.3 Crontab

Crontab-ohjelma löytyy useimmista Linux-versioista valmiina ja on helppo säätää. Ohjelmaa käytetään komennoilla crontab -e ja -l, joista e:llä voidaan muokata jo olemassa olevaa crontab-tiedostoa, ja jos tiedostoa ei ole, luodaan se samalla ja l:ällä nähdään tiedostossa olevat ajastukset.

Ajastus tehdään luomalla Crontab-tiedostoon rivi, jossa ensimmäiset viisi kohtaa ovat aika ja viimeinen kohta on ajettava tiedosto (Kuva 14). Arvot annetaan alla olevan kuvan mukaisesti (Kuva 14). Tähtien tilalla voidaan myös käyttää @daily, @hourly ja niin edelleen arvoja (Kuva 15).

```
# Minute   Hour   Day of Month       Month          Day of Week       Command
# (0-59)  (0-23)     (1-31)    (1-12 or Jan-Dec)  (0-6 or Sun-Sat)
    0        2          12             *                 *          /usr/bin/find
```

Kuva 14. Arvo periaatteet (Pantz, 2007).

```
string          meaning
------          -------
@reboot         Run once, at startup.
@yearly         Run once a year, "0 0 1 1 *".
@annually       (same as @yearly)
@monthly        Run once a month, "0 0 1 * *".
@weekly         Run once a week, "0 0 * * 0".
@daily          Run once a day, "0 0 * * *".
@midnight       (same as @daily)
@hourly         Run once an hour, "0 * * * *".
```

Kuva 15. String-arvoiset ajastukset (Pantz, 2007).

4.1.4 Elastalert

Palvelimelle asennettu myös avoimen lähdekoodin Elastalert-ohjelma, jolla voidaan tehdä hälytyksiä Elasticsearchissa olevasta datasta.

Ohjelmalle tehdään erilaisia sääntöjä, joiden perusteella ohjelma lähettää ilmoituksen esimerkiksi sähköpostiin.

Ohjelman oma säätötiedosto on yksikertainen käyttää ja säätää. Säätötiedostosta piti muuttaa tarpeen mukaan kuvien mukaisia kohtia (Kuva 16-18).

Elastalertille luodaan Elasticsearchiin oma indeksi ajamalla komento *elastalert-create-index* ja seuraamalla ohjeita.

```
# This is the folder that contains the rule yaml files
# Any .yaml file will be loaded as a rule
rules_folder: rules

# How often ElastAlert will query elasticsearch
# The unit can be anything from weeks to seconds
run_every:
  minutes: 1

# ElastAlert will buffer results from the most recent
# period of time, in case some log sources are not in real time
buffer_time:
#  days: 20
  minutes: 1
```

Kuva 16. Muutettavia kohtia 1

Rules_folder: Määritetään kansio, josta sääntöjä ajetaan.
Run_every: Määritetään aikaväli, kuinka usein säännöt ajetaan.
Buffer_time: Määritetään puskurille aika.

```
# The index on es_host which is used for metadata storage
# This can be a unmapped index, but it is recommended that you run
# elastalert-create-index to set a mapping
writeback_index: elastalert_status
#writeback_index: elastalert_testi


# If an alert fails for some reason, ElastAlert will retry
# sending the alert until this time period has elapsed
alert_time_limit:
   minutes: 1
```

Kuva 17. Muutettavia kohtia 2

writeback_index: Määritetään mitä indeksiä halutaan käyttää.
Alert_time_limit: Määritetään aikaväli jonka jälkeen hälytys läheteään uudelleen.

```
notify_email: joona.nieminen@       .fi
from_addr: Loki@       .fi
smtp_host: exchange.       .fi
email_reply_to: joona.nieminen@    _    .fi
```

Kuva 18. Muutettavia kohtia 3

Notify_email: Määritetään vastaanottajan sähköposti.
From_addr: Määritetään lähettäjän sähköposti.
smtp_host: Määritetään SMTP-palvelin.
email_reply_to: Määritetään kenelle vastaukset lähtevät.

Sääntöjä voidaan tehdä erityyppisiä, joissa jokaisessa on hieman erilaiset YAML-tiedostot.

Erilaisia sääntöjä:

1. Any, joka laukaisee hälytyksen aina kun määritetty arvo toteutuu.
2. Blacklist, johon luodaan lista, jonka arvoja vertaillaan indeksien tietoon, ja jos määritetty arvo löytyy, hälytys laukeaa.
3. Whitelist on kuin blacklist, mutta jos määritettyä arvoa ei löydy indeksien tiedoista, laukeaa hälytys.
4. Change laukeaa aina kun määritelty arvo vaihtuu.
5. Frequency laukaisee hälytyksen, kun määritellyn aikavälin aikana tapahtuu määritetyn määrän tapahtumia.
6. Spike, johon määritellään piikin koko, jonka ylittyessä hälytys laukeaa. Voidaan määritellä, seurataanko piikkiä ylös, alas vai molempiin suuntiin.
7. Flatlineen määritellään aikaväli, jonka aikana tapahduttava määritetyn määrän tapahtumia.
8. New Termiin määritellään kenttiä, joista tietoa vertaillaan ja jos uusi tietue ilmaantuu, laukeaa hälytys.
9. Cardinalityyn määritellään kenttä, johon on maksimissaan tai minimissään tultava määritelty määrä yksilöllisiä arvoja

(Yelp, 2014)

Omiin tarpeisiin riitti ”Any”- ja ”Frequency” -tyypit, jotka laukeavat joko tietyn ajan aikana, kun tietty määrä asioita on tapahtunut tai kaikesta, mikä menee arvoihin.

Anyllä tarkkaillaan levytilaa ja lähetetään sähköpostia, jos levytilan käyttö ylittää 80 % rajan. Frequencyllä lähetetään sähköpostia, jos tulee ERROR-, WARNING- tai SEVERE-merkintöjä tietyn verran tietyssä ajassa.

Säännöt ovat yksinkertaisia ja alla esimerkkinä Any-sääntö joka ilmoittaa, jos testipalvelimen kiintolevyn käyttö on yli 80 % (Kuva 19 - 22).

```
# (Optional)
# Elasticsearch host
es_host: elastic.       .fi

# (Optional)
# Elasticsearch port
es_port: 9200

# (OptionaL) Connect with SSL to elasticsearch
#use_ssl: True

# (Optional) basic-auth username and password for elasticsearch
es_username: testi
es_password: testi
```

Kuva 19. Muutettavia kohtia säännössä 1

es_host: Elasticsearch palvelimen osoite, ei pakollinen
es_port: Elasticsearchin portti, ei pakollinen
use_ssl: Käyttääkö SSL yhteyttä, ei pakollinen.

```
# (Required)
# Rule name, must be unique
name: Testipalvelimien tallennustilan alert.

#Any triggeröityy jokaisesta tapahtumasta
type: any

# (Required)
# Index to search, wildcard supported
index: topbeat-testi*
```

Kuva 20. Muutettavia kohtia säännössä 2

name: Säännön nimi, pakollinen
type: Säännön tyyppi, pakollinen
index: Mitä indeksiä tarkkaillaan, pakollinen.

```
# (Required)
# A list of elasticsearch filters used for find events
# These filters are joined with AND and nested in a filtered query
# For more info: http://www.elasticsearch.org/guide/en/elasticsearch/reference/$
filter:
- range:
    fs.used_p:
      from: 0.8


include: ["fs.used_p", "beat.hostname", "fs.total", "fs.avail", "tags"]

realert:
  minutes: 30
```

Kuva 21. Muutettavia kohtia säännöissä 3

filter: Suodattimet, pakolliset
- range: Voi asettaa välin koska laukaisee säännön
    - fs.used_p: On topbeatin luoma kenttä, jota seurataan (kovalevyn käyttö prosenteissa)
        - from: raja, jonka jälkeen sääntö laukeaa. Voidaan myös asettaa from – to arvot

include: Mitkä kentät haluaa sisällyttää hälytyksen mukana
realert: Uudelleen hälytyksen aikataulu.

```
# (Required)
# The alert is use when a match is found
alert:
- "email"

alert_subject: "Jossain testipalvelimessa on levytila vähissä"

# (required, email specific)
# a list of email addresses to send alerts to
email:
- "joona.nieminen@      .fi"
```

Kuva 22. Muutettavia kohtia säännöissä 4

alert: Minkä tyyppinen hälytys lähetetään. Olemassa useita erilaisia tyyppejä, jotka eriteltynä alla
- ”email”: Valittu hälytyksen tyyppi

alert_subject: Sähköpostin aihe
email: Sähköposti johon viesti lähetetään, pakollinen.

Erilaisia hälytystyyppejä on:
1. Command, ajaa halutun komennon
2. Email, lähettää sähköpostin
3. Jira, avaa Jiraan tiketin
4. OpsGenie, lähettää OpsGenieen hälytyksen
5. SNS, lähettää ilmoituksen Amazonin Simple Notification Serviceen käyttämällä Botoa
6. HipChat, lähettää ilmoituksen määriteltyyn HipChat-huoneeseen
7. Slack, lähettää ilmoituksen määriteltyyn Slack-kanavaan
8. Telegram, lähettää ilmoituksen määritellylle telegram-käyttäjälle tai kanavalle

9. PaperDuty, lähettää ilmoituksen PaperDuty-palveluun
10. VictorOps, lähettää ilmoituksen VictorOps-palveluun
11. Gitter, lähettää ilmoituksen määritellylle Gitter-kanavalle
12. ServiceNow, luo uuden tapahtuman ServiceNow-palveluun
13. Debug, luo tiedoston käyttämällä Pythonin lokittajaa.

(Yelp, 2014)

Ohjelmaa ajetaan menemällä asennus kansioon ja ajetaan käsky: *python -m elastalert.elastalert* . Lisäksi käskyssä on käytetty kohtia verbose ja rule. Verbose antaa ajosta INFO-tason viestejä näkyviin (Kuva 23) ja rulella voidaan ajaa vain tiettyä sääntöä, esimerkiksi *--rule rules/filesystem_tuotanto.yaml*.

```
Queried rule Tuotantopalvelimien tallennustilan alert. from 8-19 12:13 EEST to 8
-19 12:14 EEST: 0 hits
INFO:elastalert:Queried rule Tuotantopalvelimien tallennustilan alert. from 8-19
 12:13 EEST to 8-19 12:14 EEST: 0 hits
Queried rule Tuotantopalvelimien tallennustilan alert. from 8-19 12:14 EEST to 8
-19 12:15 EEST: 0 hits
INFO:elastalert:Queried rule Tuotantopalvelimien tallennustilan alert. from 8-19
 12:14 EEST to 8-19 12:15 EEST: 0 hits
Queried rule Tuotantopalvelimien tallennustilan alert. from 8-19 12:15 EEST to 8
-19 12:16 EEST: 0 hits
```

Kuva 23. Esimerkki INFO-tason viesteistä säännön ajossa

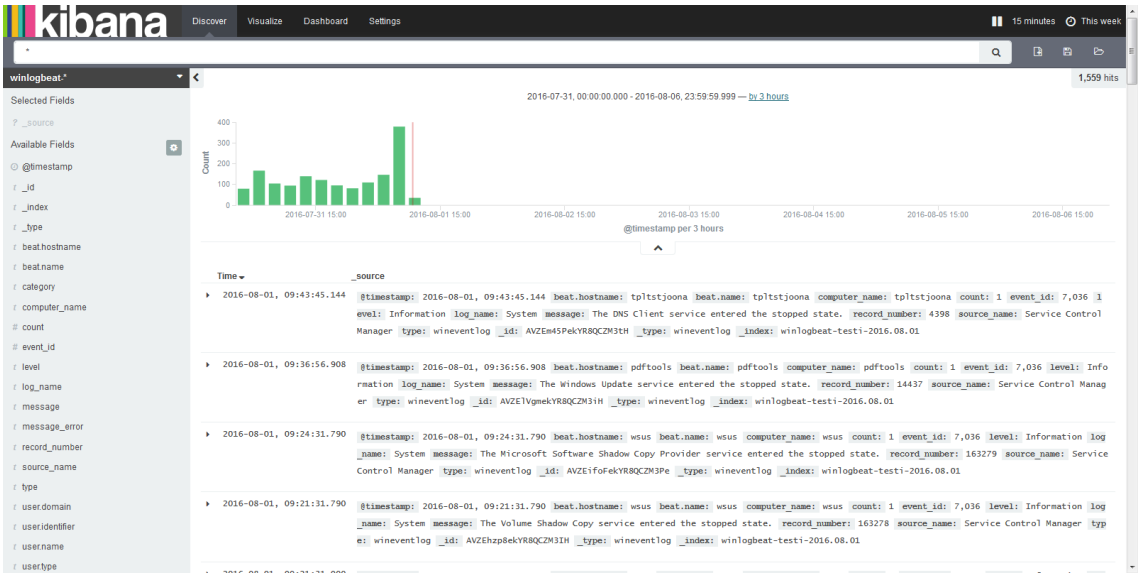
4.2 Kibana 4.1.2

Kyseessä on ohjelma, joka tuottaa Elasticsearchin indeksoidusta datasta visuaalista dataa.

Aloittaessani työt oli Kibana (Kuva 24) asennettu palvelimelle ja todettu toimivaksi ja tutkittu käyttöliittymää.

Tarkoituksenani oli saada Kibanaan visuaalisesti näkyviin palvelimien data helposti ja selkeästi. Toimeksiantajaa kiinnostaa erityisesti SaaS-palvelimien resurssit, mahdolliset häiriöt Windows eventeissä ja yrityksen tuotteiden lokit. Tarkoituksena järjestelmällä on vähentää palvelimilla turhaa käyntiä, palvelimella käytäisiin vain ongelmatilanteissa, jotka näkisi Kibanasta.

Kibana jakautuu neljään välilehteen, Discover, Visualize, Dashboard ja Settings, joista kerron alla lisää.



Kuva 24. Kibana

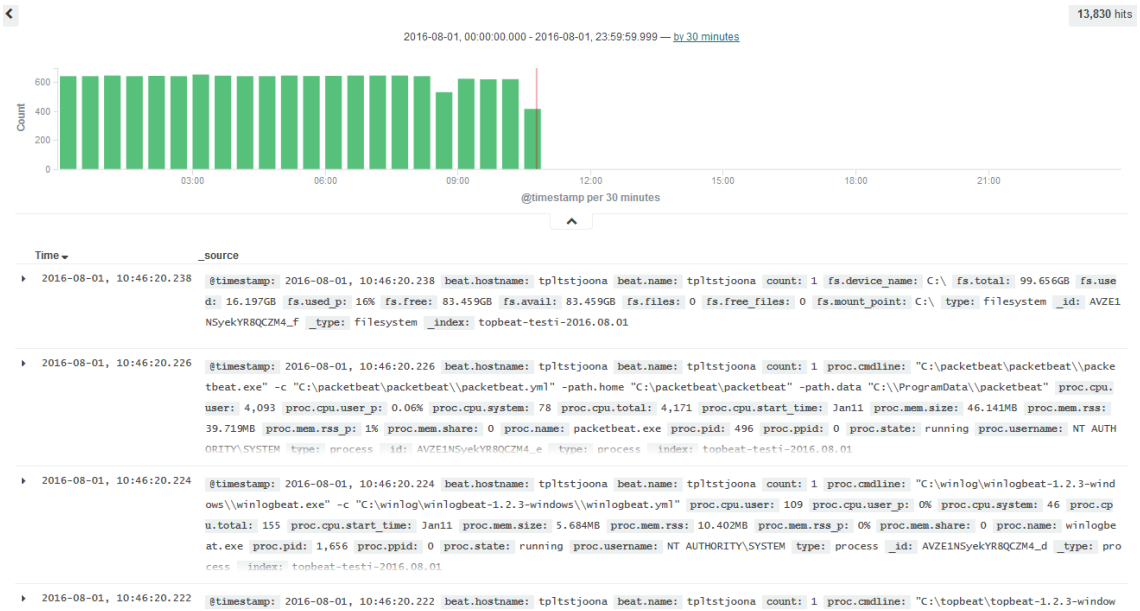
1. Discover-välilehti

Discover-välilehden ylälaidassa on hakukenttä (Kuva 25), mistä dataan voi tehdä erilaisia kyselyitä. Kyselyillä voidaan hakea esimerkiksi vain tietyn tietokoneen tiedot näkyviin. Hakukentän vieressä on kolme painiketta, joista voidaan luoda, tallentaa ja avata erilaisia hakuja.

Sivun oikeassa ylälaidassa on automaattisen päivityksen aikataulut ja aikaväli, jolta tieto näytetään.

Kuva 25. Hakukenttä ja sivun ylälaita.

Sivun keskeltä löytyy data ja merkintöjen lukumäärä (Kuva 26). Datasta näkee Elasticsearchin tekemät indeksoinnit harmaalla taustavärillä.



Kuva 26. Data ja merkintöjen lukumäärä.

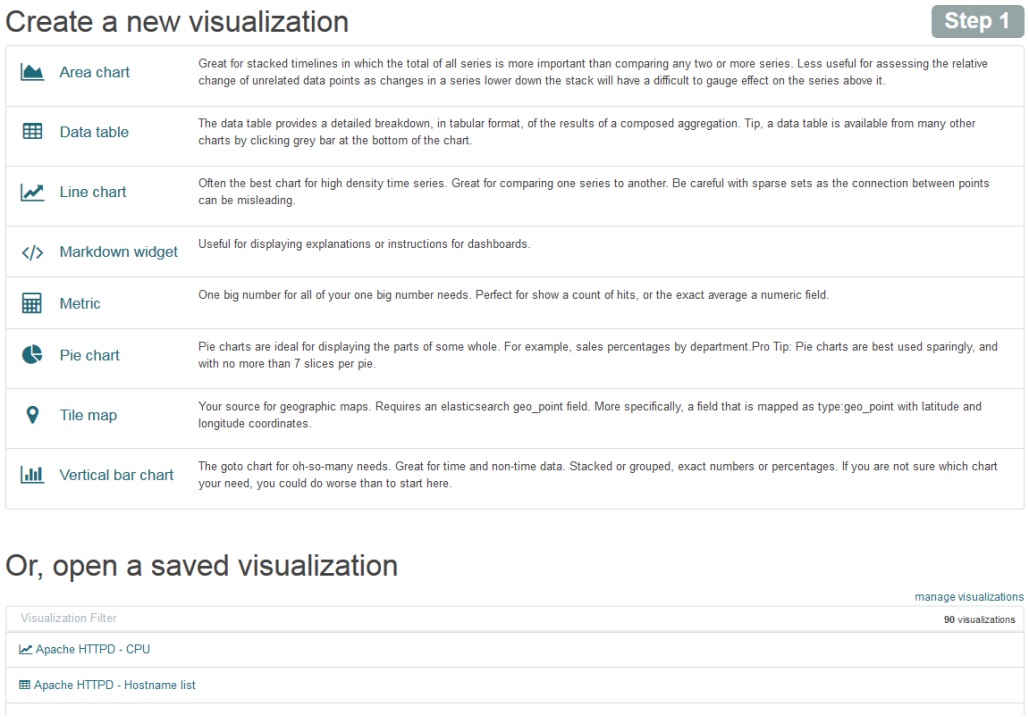
Sivun vasemmassa laidassa (Kuva 27) on Elasticsearchin tekemät indeksit, joista valitsemalla voidaan myös luoda hakuja dataan.

Kuva 27. Sivun vasenlaita

2. Visualize-välilehti

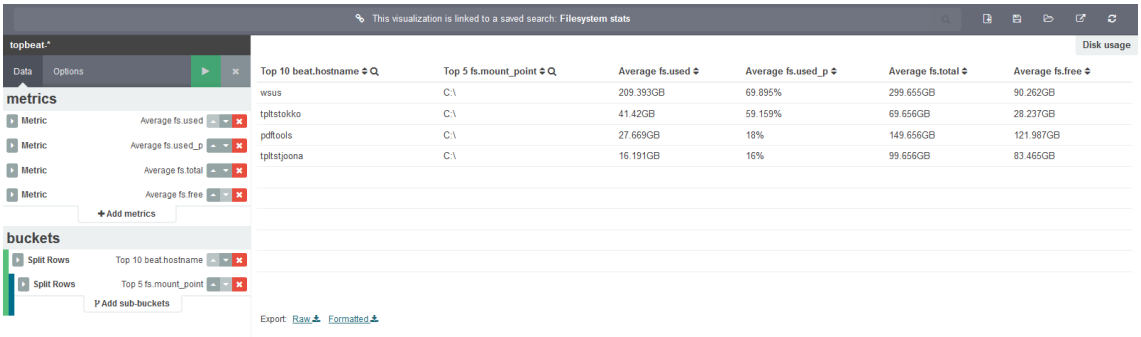
Visualize-välilehdellä saadaan tehtyä erilaisia visuaalisia tehosteita erilaisille hauille (Kuva 28). Erilaiset tehosteet voidaan tehdä joko Discoveryssä tallennetuille hauille, luomalla uusia hakuja indekseihin tai muokata jo olevia visuaalisia tehosteita. Visualizessa tehtyjä tehosteita käytetään Dashboard -välilehdellä, haluttuja näkemyksiä luodessa.

Elastic tarjoaa valmiita visuaalisia tehosteita ohjelmiinsa. Esimerkiksi Beats-tuotteiden tehosteet saadaan haettua Linux-palvelimelle komennolla *curl -L -O http://download.elastic.co/beats/dashboards/beats-dashboards-1.2.3.zip,* jonka jälkeen ladattu paketti puretaan komennolla *unzip beats-dashboards-1.2.3.zip.* Tämän jälkeen siirrytään *beats-dashboards* kansioon ja ajetaan siellä *./load.sh* -komento, komennon ajamisen jälkeen Kibanaan saadaan hyviä pohjia tehosteille.



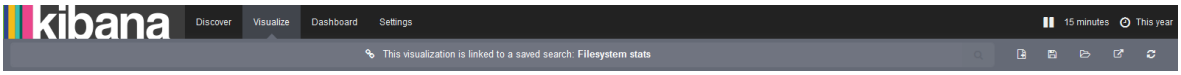
Kuva 28. Visualize -välilehti

Data visualisoidaan hakemalla haun eri indeksejä halutuilla arvoilla halutuun järjestykseen (Kuva 29).



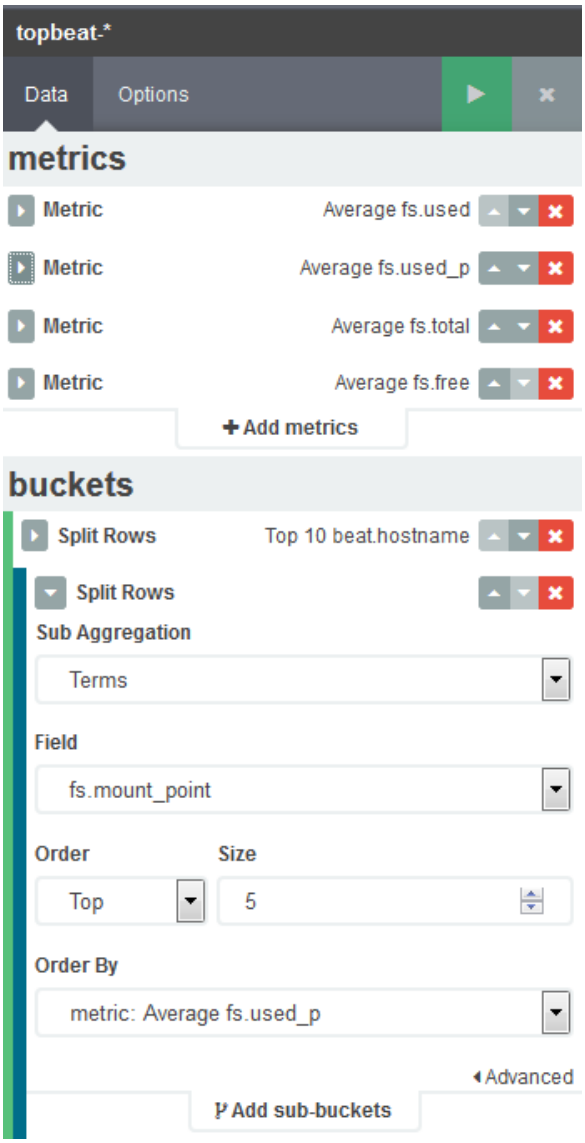
Kuva 29. Datan visualisointi.

Sivun ylälaidassa on palkki, johon haun voi tehdä, mutta alla-olevassa kuvassa on linkitys Discover-välilehdellä tehtyyn hakuun (Kuva 30). Hakupalkin vieressä on painikkeet, joista voidaan luoda uusi tehoste, avata tehoste tai tallentaa, jakaa tai päivittää kyseinen tehoste.



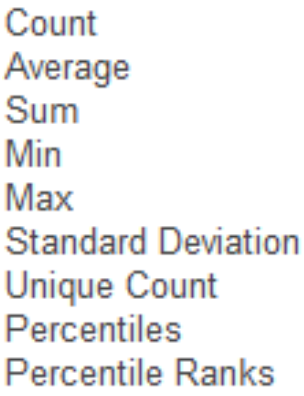
Kuva 30. Hakupalkki

Sivun vasempaan laitaan luodaan haluttuja mittareita ja rivejä, jotta saadaan haluttu tulos tehosteeseen (Kuva 31).

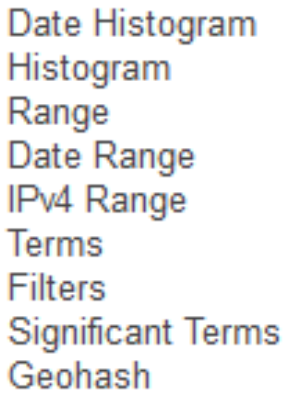


Kuva 31. Mittarit ja rivit

Mittareiden ja rivien koosteet riippuvat indeksoidusta ja analysoidusta datasta. Mittareihin saa valittua laskutavan alla olevan kuvan mukaisesti (Kuva 32). Riveihin saa valittua termit, joista se etsii alla olevan kuvan mukaisesti (Kuva 33). Jokaisella termillä ja laskutavalla on omat lisätietonsa, haku kohteet, kentät tai arvot, joita muokkaamalla saadaan datasta yksilöidympää.

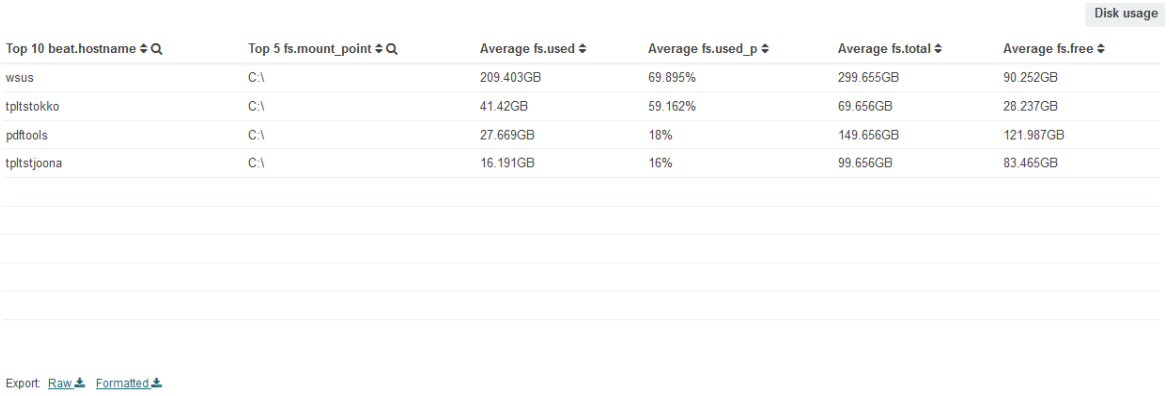


Kuva 32. Laskutavat



Kuva 33. Rivien lisätermit

Sivun keskeltä näkee millaisen tiedon valitut hakuarvot palauttavat (Kuva 34). Taulukon/Kuvion alta voi kyseisen taulukon/kuvion tiedot viedä CSV-tiedostoksi.



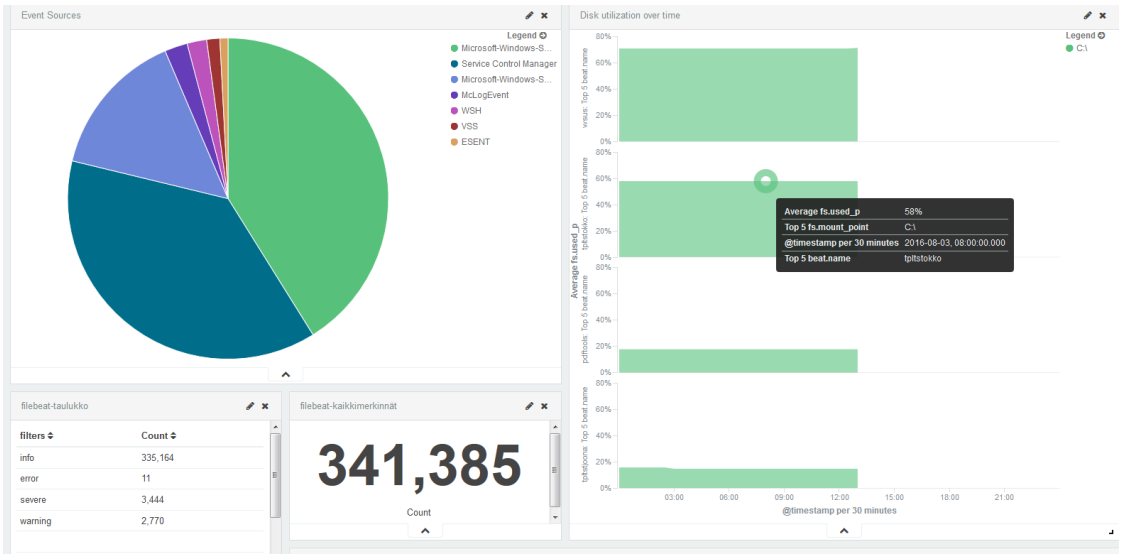
Kuva 34. Valittujen hakuarvojen tulos.

3. Dashboard-välilehti

Dashboard-välilehdellä voidaan luoda halutunlaisia näkemyksiä, johon Visualize-välilehdellä tehdyt tehosteet tai Discovery-välilehdellä tallennetut haut asetetaan. Sivun ylälaidassa on hakupalkki, johon voi luoda hakuja. Hakupalkin vierestä löytyy painikkeet uuden näkemyksen luomiselle, vanhan avaamiselle tai nykyisen näkemyksen tallentamiselle, jakamiselle ja tehosteen lisäämiselle (Kuva 35).

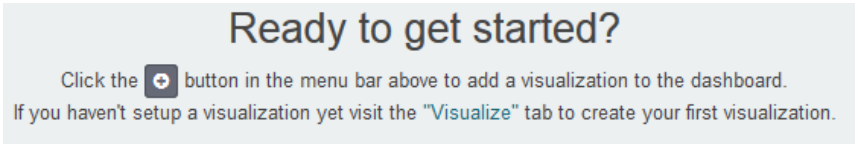
Kuva 35. Dashboard-hakupalkki.

Yritykselle luotu näkemys, mistä näkee kaiken tarpeellisen kerralla (Kuva 36). Näkemyksestä näkee kerralla kaikkien Beats-ohjelmien lähettämiä tietoja.

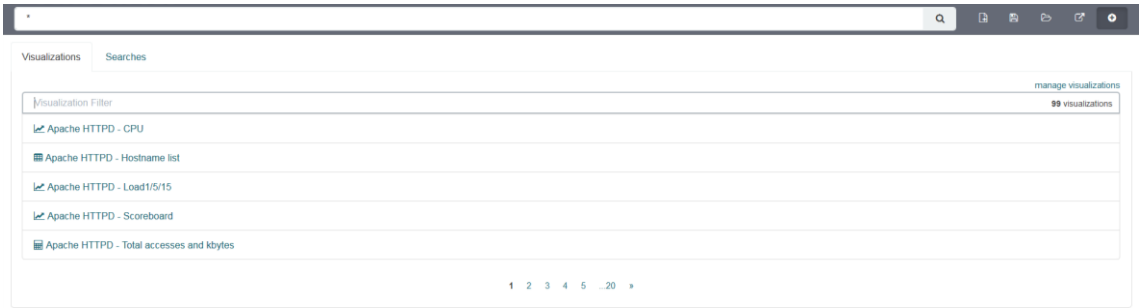


Kuva 36. Esimerkki näkymästä.

Uusi näkymä luodaan lisäämällä haluttuja Visualize-tehosteita haluttuun järjestykseen painamalla + näppäintä (Kuva 37). Näppäimen painon jälkeen avautuu lista tehdyistä tehosteista ja välilehdeltä löytyy lista tehdyistä hauista (Kuva 38), näitä valitsemalla ilmestyy valittu tehoste/haku näkemykseen.



Kuva 37. Uuden näkymän luonti.



Kuva 38. Lista tehosteista ja hauista

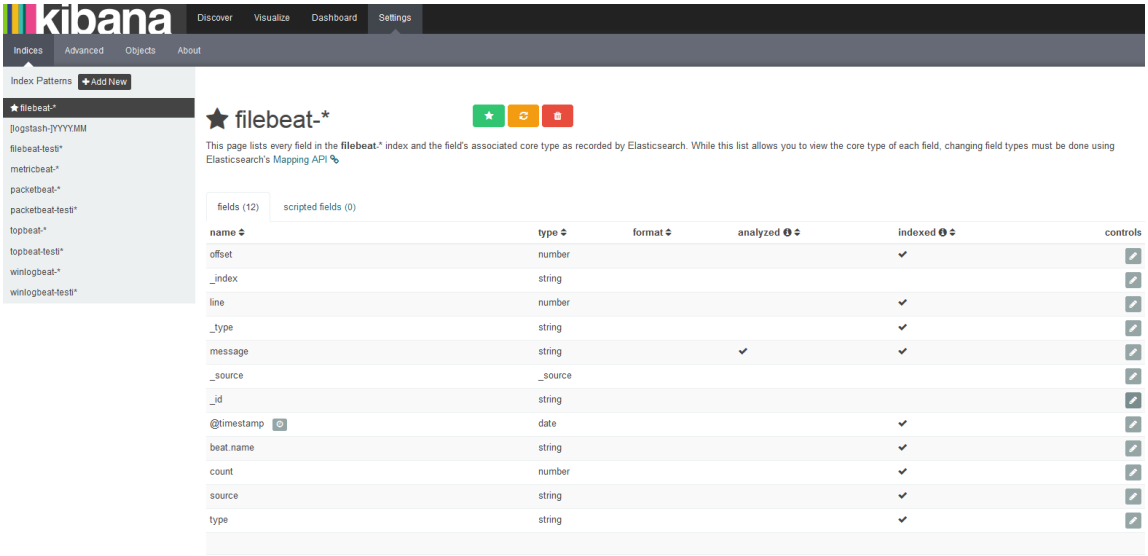
Jokaista tehostetta pystyy muokkaamaan ja poistamaan suoraan näkymästä painamalla tehosteen ylälaidassa kynä- tai X-merkkiä (kuva 39).

Kuva 39. Painikkeet

4. Settings-välilehti

Settings-välilehti jakautuu neljään osioon, Indices, Advanced, Objects ja About.

Indices kohdassa luodaan ja säädetään indeksien malleja, joita Elasticsearchista haetaan, ja johon discover-välilehdessä pystyy hakuja luomaan (Kuva 40). Kun haku on luotu, näyttää Kibana indeksoidut ja analysoidut kentät listassa (Kuva 40).



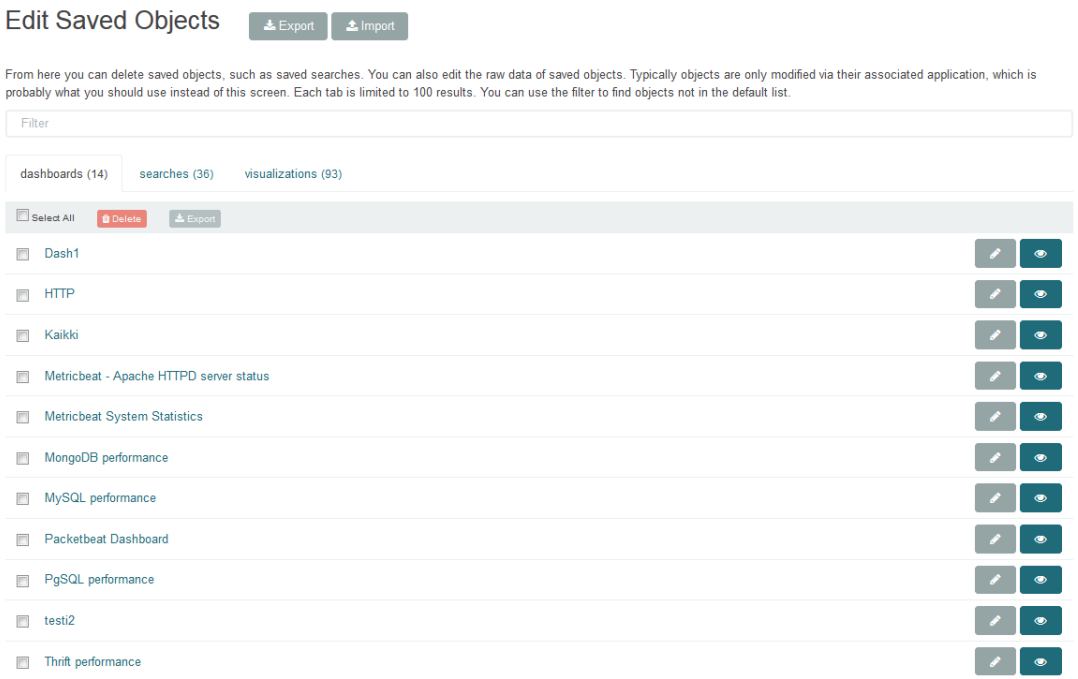
Kuva 40. Haetut mallit, indeksoidut ja analysoidut kentät

Advanced kohdassa säädetään Kibanan asetuksia (Kuva 41).

Name	Value	Actions
query:queryString:options Options for the lucene query string parser	{ "analyze_wildcard": true }	
sort:options Options the Elasticsearch sort parameter	{ "unmapped_type": "boolean" }	
dateFormat (Default: *MMM Do YYYY, HH:mm:ss.SSS*) When displaying a pretty formatted date, use this format	YYYY-MM-DD, HH:mm:ss.SSS	
dateFormat:scaled Values that define the format used in situations where timebased data is rendered in order, and formatted timestamps should adapt to the interval between measurements. Keys are ISO 8601 intervals: http://en.wikipedia.org/wiki/ISO_8601#Time_intervals	[["", "hh:mm:ss.SSS"], ["PT1S", "HH:mm:ss"], ["PT1M", "HH:mm"], ["PT1H", "YYYY-MM-DD HH:mm"], ["P1DT", "YYYY-MM-DD"], ["P1YT", "YYYY"]]	
defaultIndex (Default: *null*) The index to access if no index is set	filebeat-*	
metaFields Fields that exist outside of _source to merge into our document when displaying it	_source, _id, _type, _index	
discover:sampleSize The number of rows to show in the table	500	
fields:popularLimit The top N most popular fields to show	10	
format:numberPrecision Round numbers to this many decimal places	3	
histogram:barTarget Attempt to generate around this many bar when using "auto" interval in date histograms	50	
histogram:maxBars Never show more than this many bar in date histograms, scale values if needed	100	
visualization:tileMap:maxPrecision The maximum geoHash precision displayed on tile maps: 7 is high, 10 is very high, 12 is the max. Explanation of cell dimensions: http://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-bucket-geohashgrid-aggregation.html#_cell_dimensions_at_the_equator	7	

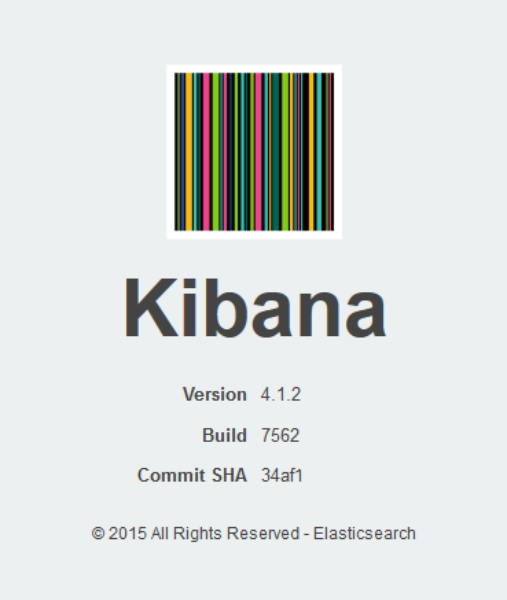
Kuva 41. Esimerkki asetuksista.

Objects-välilehdeltä objektien tuonnin ja viennin lisäksi, voi muokata, katsoa, viedä ja poistaa tehtyjä näkemyksiä, hakuja tai visualisointeja (Kuva 42).



Kuva 42. Objects

About kohta kertoo Kibanasta lisätietoja (Kuva 43).



Kuva 43. About

4.3 Beats 1.2.3

Beatsit ovat ohjelmia, joilla järjestelmien tiedot kerätään ja lähetetään. Käytössä Winlogbeat, Topbeat, Packetbeat ja Filebeat. Jokainen säädetään käyttäen YAML (.yml)- ja template-tiedostoja. Ohjelmat lähettävät tiedon JSON-muodossa Elasticsearchiin.

Beats-tuotteet ovat pieniä zip-paketteja, jotka ovat kooltaan kahden ja neljän megabitin väliltä. Yritykselle luotu omat paketit, jotta asentaminen on mahdollisimman helppoa.

Windows ympäristöissä halutut zip-paketit puretaan haluttuun kansioon, jonka jälkeen ajetaan mukana tullut *install-service-XXXbeat.ps1*-tiedosto, tämän ajamisen jälkeen kyseinen Beats-ohjelma on asennettu palveluksi.

Linux ympäristöissä halutut Beats-ohjelmat haetaan käskyllä: *curl -L -O https://download.elastic.co/beats/XXXbeat/XXXbeat-1.2.3-x86_64.rpm* haun jälkeen ajetaan käsky: *sudo rpm -vi XXXbeat-1.2.3-x86_64.rpm.* Käskyn jälkeen siirrytään etc/XXXbeat/-kansioon ja tarkistetaan yml-tiedostosta asetukset oikeiksi, jonka jälkeen käynnistetään komennolla: *sudo /etc/init.d/XXXbeat start* .

Ohjelmat säädetään siten, että aluksi otetaan ympäristöistä mahdollisimman paljon tietoa, pienen hetken päästä pystytään määrittelemään tiedon tarpeellisuus, aikaväli jolta tieto halutaan ja päivitystaajuus huomattavasti tarkemmaksi.

Kaikki säätötiedot koostuvat ohjelmakohtaisesta osuudesta ja yleisestä osuudesta, joka kaikissa ohjelmissa on lähes sama.

Yleisellä osuudella voidaan säätää tulos(Output), lähettäjä(Shipper) ja lokitus(Logging). Liitteenä yleinen osuus (Liite 1).

Työtä tehdessä todettiin Elasticsearch riittäväksi vastaanottajaksi eikä logstashia tarvita väliin, joten yleisistä asetuksista pitää muuttaa vain tuloskohdasta Elasticsearchin osoite oikeaksi ja indeksin nimi viittaamaan haluttuun indeksiin (Kuva 44) ja lähettäjä-kohdassa laittaa halutessa jokin ”merkki”, jotta saadaan kohdistettua hakuja helpommin (Kuva 45). Lokitusta käytettiin vain, jos asennus ei onnistunut.

Oikeanlaisella merkkaamisella saadaan aikaiseksi haluttuja hakuja huomattavasti helpommalla. Merkkaaminen onnistuu lähettäjä kohdasta kahdella tapaa, joko laitetaan ”merkki” tai muokataan nimeä (Kuva 45).

```
############################# Output ########################################

# Configure what outputs to use when sending the data collected by the beat.
# Multiple outputs may be used.
output:

  ### Elasticsearch as output
  elasticsearch:
    # Array of hosts to connect to.
    # Scheme and port can be left out and will be set to the default (http and 9200)
    # In case you specify and additional path, the scheme is required: http://localhost:9200/path
    # IPv6 addresses should always be defined as: https://[2001:db8::1]:9200
    hosts: ["elastic.       .fi:9200"]

    # Optional protocol and basic auth credentials.
    #protocol: "https"
    #username: "admin"
    #password: "s3cr3t"

    # Number of workers per Elasticsearch host.
    #worker: 1

    # Optional index name. The default is "topbeat" and generates
    # [topbeat-]YYYY.MM.DD keys.
    index: "topbeat-testi"
```

Kuva 44. Tuloksen muutokset

```
############################# Shipper ########################################

shipper:
  # The name of the shipper that publishes the network data. It can be used to group
  # all the transactions sent by a single shipper in the web interface.
  # If this options is not defined, the hostname is used.
  #name:tpltstjoona-testi

  # The tags of the shipper are included in their own field with each
  # transaction published. Tags make it easy to group servers by different
  # logical properties.
  tags: ["testi"]
```

Kuva 45. Lähettäjän muutokset

4.3.1 Winlogbeat 1.2.3

Winlogbeatin säätöosuus on yksinkertainen (Kuva 46). Alkuperäisissä säädöissä, Security ja System lokeista otetaan kaikki merkinnät asentamisesta nykyhetkeen.

```
############################## Winlogbeat ###################################
winlogbeat:
  # The registry file is where Winlogbeat persists its state so that the beat
  # can resume after shutdown or an outage. The default is .winlogbeat.yml
  # in the directory in which it was started.
  registry_file: C:/ProgramData/winlogbeat/.winlogbeat.yml

  # List of event logs to monitor.
  #
  # Optionally, ignore_older may be specified to filter events that are older
  # then the specified amount of time. If omitted then no filtering will
  # occur. Valid time units are "ns", "us" (or "µs"), "ms", "s", "m", "h"
  event_logs:
    - name: Application
      ignore_older: 72h
    - name: Security
    - name: System

  # Diagnostic metrics that can retrieved through a web interface if a
  # bindaddress value (host:port) is specified. The web address will be
  # http://<bindaddress>/debug/vars
  #metrics:
  #  bindaddress: 'localhost:8123'
```

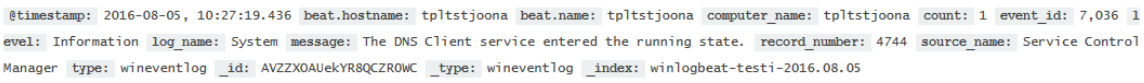
Kuva 46. Winlogbeatin säätöosuus.

Säätötiedostosta muokattu vain event_logs-kohtaa, jossa asetettu unohtamaan vanhemmat kuin 72 tuntia vanhat (Kuva 47). Jos kyseisessä kohdassa ei ole asetettu unohtamaan vanhoja, niin Winlogbeat lähettää kaikki Windows eventit tietokoneen asennuksesta lähtien.

```
event_logs:
  - name: Application
    ignore_older: 72h
  - name: Security
    ignore_older: 72h
  - name: System
    ignore_older: 72h
```

Kuva 47. Muokattu osuus

Kibanassa Winlogbeatin lähettämä data näyttää alla olevan kuvan mukaiselta (Kuva 48).



Kuva 48. Event kohtainen data.

4.3.2 Topbeat 1.2.3

Topbeatin säätöosuus on yksinkertainen (Kuva 49). Alkuperäisissä säädöissä tieto lähetetään 10 sekunnin välein, jokaista ohjelmaa seurataan ja statistiikkaa kerätään järjestelmästä, prosesseista ja tiedostojärjestelmästä.

```
############################## Input ########################################
input:
  # In seconds, defines how often to read server statistics
  period: 10

  # Regular expression to match the processes that are monitored
  # By default, all the processes are monitored
  procs: [".*"]

  # Statistics to collect (all enabled by default)
  stats:
    # per system statistics, by default is true
    system: true

    # per process statistics, by default is true
    process: true

    # file system information, by default is true
    filesystem: true

    # cpu usage per core, by default is false
    cpu_per_core: false
```

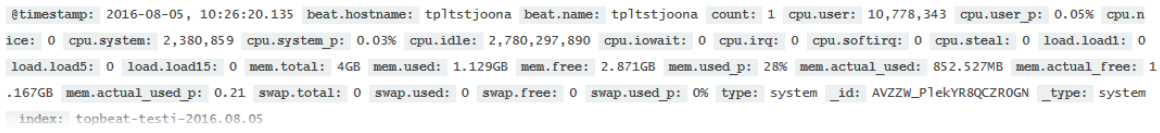
Kuva 49. Topbeatin säätöosuus.

Topbeat-ohjelmaa muokattiin yritykselle sopivammaksi ja säätöjä muutettiin tuotanto-palvelimille alla olevan kuvan mukaisesti (Kuva 50). Yrityksen testi-palvelimilta tiedot otetaan vielä harvemmin.

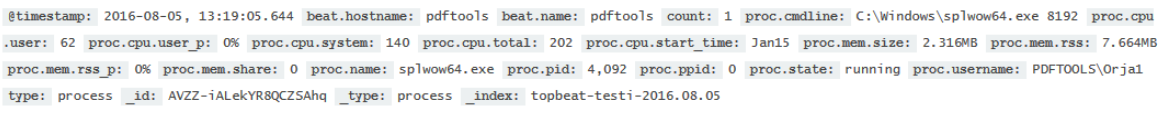
```
############################## Input ########################################
input:
  # In seconds, defines how often to read server statistics
  period: 600

  # Regular expression to match the processes that are monitored
  # By default, all the processes are monitored
  procs: [".*"]

  # Statistics to collect (all enabled by default)
  stats:
    # per system statistics, by default is true
    system: true

    # per process statistics, by default is true
    process: true

    # file system information, by default is true
    filesystem: true

    # cpu usage per core, by default is false
    cpu_per_core: true
```

Kuva 50. Muutetut säädöt.

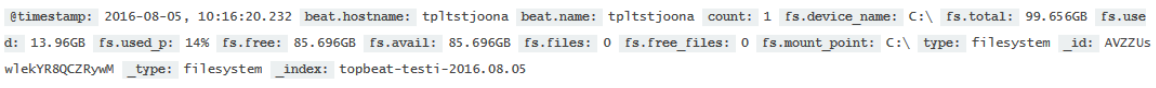
Kibanassa Topbeatin lähettämä tieto näyttää alla olevien kuvien mukaiselta (Kuva 51 - 53).



Kuva 51. Järjestelmä-data



Kuva 52. Prosessi-data



Kuva 53. Tiedostojärjestelmä-data

4.3.3 Packetbeat 1.2.3

Packetbeatin säätöosuus on hieman hankalampi ja monimuotoisempi kuin muissa ja koska säätöosuus on huomattavasti pidempi kuin muissa on Packetbeatin säätöosuus liitteenä (Liite 2). Packetbeat myös vaatii asennettavaksi WinPcap-ohjelman toimiakseen. Yrityksellä ei Packetbeatille suoranaista tarvetta ollut, mutta ohjelmaa voi käyttää esimerkiksi yrityksen omien verkkosivujen tarkkailuun.

Packetbeat voidaan säätää portti- ja protokollakohtaisesti. Protokollat jotka löytyvät suoraan säätötiedostosta ovat muun muassa DNS, HTTP, MySQL ja mongoDB.

Yritykselle säädettäessä voitiin suurin osa protokollista unohtaa ja jättää jäljelle vain DNS ja HTTP.

Packetbeat lähettämä data näyttää alla olevan kuvan mukaiselta (Kuva 54).

Kuva 54. Packetbeatin lähettämä data

4.3.4 Filebeat 1.2.3

Koska ohjelmalla pystyttiin lähettämään suoraan tietoa Elasticsearchiin, todettiin Logstashin ja Logstash-forwarderin olevan turhia, tutkitulla testidatalla.

Filebeatin säätöosuus on monipuolinen ja pitkä, joten säätöosuus on liitteenä (Liite 3).

Säädöt ovat yksinkertaisia, lukuun ottamatta useamman rivin loki merkintää. Säädöt käyvät pitkälti oletusarvoisesti, mutta muutamia kohtia muutettu (Kuva 55 – 58).

```
filebeat:
  # List of prospectors to fetch data.
  prospectors:
    # Each - is a prospector. Below are the prospector specific configurations
    -
      # Paths that should be crawled and fetched. Glob based paths.
      # To fetch all ".log" files from a specific level of subdirectories
      # /var/log/*/*.log can be used.
      # For each file found under this path, a harvester is started.
      # Make sure not file is defined twice as this can lead to unexpected behaviour.
      paths:
        - C:\testi\*
        #- c:\programdata\elasticsearch\logs\*

      # Configure the file encoding for reading files with international characters
      # following the W3C recommendation for HTML5 (http://www.w3.org/TR/encoding).
      # Some sample encodings:
      #   plain, utf-8, utf-16be-bom, utf-16be, utf-16le, big5, gb18030, gbk,
      #    hz-gb-2312, euc-kr, euc-jp, iso-2022-jp, shift-jis, ...
      #encoding: plain

      # Type of the files. Based on this the way the file is read is decided.
      # The different types cannot be mixed in one prospector
      #
      # Possible options are:
      # * log: Reads every line of the log file (default)
      # * stdin: Reads the standard in
      input_type: log
```

Kuva 55. Muutettu sijaintia mistä lokit haetaan.

```
# Ignore files which were modified more then the defined timespan in the past.
# In case all files on your system must be read you can set this value very large.
# Time strings like 2h (2 hours), 5m (5 minutes) can be used.
ignore_older: 48h
```

Kuva 56. Muutettu aikaa, ettei asentaessa lataa kaikkia lokeja.

```
# Scan frequency in seconds.
# How often these files should be checked for changes. In case it is set
# to 0s, it is done as often as possible. Default: 10s
#scan_frequency: 10s
```

Kuva 57. Mahdollisuus muuttaa aikaa jonka välein tarkastetaan uudet rivit.

```
  # Mutiline can be used for log messages spanning multiple lines. This is common
  # for Java Stack Traces or C-Line Continuation
  multiline:

    # The regexp Pattern that has to be matched. The example pattern matches all lines starting with [
    pattern: '^[[:space:]]'

    # Defines if the pattern set under pattern should be negated or not. Default is false.
    negate: false


    # Match can be set to "after" or "before". It is used to define if lines should be append to a pattern
    # that was (not) matched before or after or as long as a pattern is not matched based on negate.
    # Note: After is the equivalent to previous and before is the equivalent to to next in Logstash
    match: after

    # The maximum number of lines that are combined to one event.
    # In case there are more the max_lines the additional lines are discarded.
    # Default is 500
    max_lines: 100

    # After the defined timeout, an multiline event is sent even if no new pattern was found to start a new event
    # Default is 5s.
    timeout: 1s
```

Kuva 58. Useamman rivin lokitus

Kibanassa Filebeatin lähettämä data näyttää alla olevan kuvan mukaiselta (Kuva 59)

```
@timestamp: 2016-10-04, 09:52:35.709 beat.hostname: maximus beat.name: maximus count: 1 fields: - input_type: log message: INFO: #T
IMER ##item# excuteQuery: 0,421 s offset: 63,930 source: D:\Tomcat\logs\maximus-stderr.2016-10-04.log tags: Tuotanto type: log _id: AV
eOOrjWekYR8QCZpEV1 _type: log _index: filebeat-tuotanto-2016.10.04
```

Kuva 59. Filebeatin lähettämä data

5 JOHTOPÄÄTÖKSET JA POHDINTA

Tässä luvussa käydään lävitse nykyistä toimintatapaa, kohdattuja ongelmia, niiden ratkaisuja, mahdollisia jatkotoimenpiteitä ja työn yhteenveto.

5.1 Uusi toimintatapa vastaan vanha

Työn tuloksena toimeksiantajalla on laaja järjestelmä, jolla ympäristöistä saadaan data tuotua yhteen paikkaan, indeksoituna, analysoituna ja tarkkailukin tapahtuu yhdestä paikasta, Kibanasta.

Uusi järjestelmä mahdollistaa ennakoivan ongelmanratkonnan lähes reaaliajassa, vanhalla tavalla ennakointi varsinkin palvelimien ongelmissa oli heikkoa, koska palvelimien tarkkailu tapahtui noin kerran kuussa. Koska tarkkailu oli näin harvoin, saattoi tarkastuksien välillä tapahtua jotain merkittävää, joka huomattiin vasta kun seuraavan kerran palvelimella käytiin.

Entinen toimintaperiaate palvelimien resurssien tarkistamiselle oli se että, käytiin fyysisesti palvelimella tarkastamassa tilanne, johon kului turhan paljon aikaa ja vaivaa. Uudella järjestelmällä resurssien tarkistaminen tapahtuu suoraan Kibanasta. Järjestelmässä on myös hälytystoiminto, joka on toteutettu Elastalert-ohjelmistolla, joka hälyttää, jos ympäristöjen tallennustilaa on käytetty yli 80 %.

Ennen omien tuotteiden lokia tutkittiin vasta, kun joku asiasta ilmoitti, eli ongelma oli saattanut tapahtua jo huomattavasti aikaisemmin eikä sitä ole pystynyt huomaamaan, koska ei palvelimella ole ollut tarvetta käydä. Uudella järjestelmällä voidaan tutkia tiettyjä lokitasoja tai kaikkia lokeja kerralla menemättä palvelimelle. Järjestelmässä on myös hälytys toteutettu Elastalert-ohjelmalla, joka hälyttää, jos tietyn tasoisia merkintöjä on tullut tietyn ajan sisään tarpeeksi.

Tekniikan-tiloihin tulevasta näytöstä järjestelmän pääkäyttäjät näkevät pelkällä silmäilyllä ympäristöjen ja palveluiden tilanteen, tällä tavalla pystytään tarkkailemaan lokimerkintöjen ja resurssien tilannetta lähes huomaamattomasti.

Uudella järjestelmällä tavoitellaan tilannetta, jossa:

- Järjestelmäasiantuntijat pystyisivät ongelmiin puuttumaan jo ennen, kun vika ilmentyisi asiakkaalle. Palvelimien ja palveluiden seuranta tapahtuisi vain järjestelmäasiantuntijoiden tilassa olevan näytön ja sähköposti hälytyksien avulla.
- Ohjelmistokehittäjät saisivat omista ohjelmistaan selvän kuvan, miten ohjelmat toimivat ja missä menee normaali raja ilmoituksista ja merkinnöistä.
- Helpdesk-työntekijät voisivat helposti seurata tapahtumien kulkua suoraan Kibanasta, että mitä on tehty, tapahtunut ja kenen toimesta.

5.2 Kohdatut ongelmat ja ratkaisut

Alla kohdattuja ongelmia ja niiden ratkaisuja.

1. Curator 4.0

Curator 4.0 ei aluksi toiminut, joten palvelimelle asennettiin Pythoniin setup-toolsit komennolla *yum install python-setuptools*. Tämän jälkeen huomattiin, ettei Curator 4.0 ollutkaan yhteensopiva nykyisen Elasticsearch version 1.7.2 kanssa, joten asensin 3.5.1 version Curatorista.

2. Logstash 1.5.4

Toimeksiantajalla oli Logstash-ohjelma jo asennettuna, mutta koska Beats-ohjelmilla pystytään lähettämään dataa suoraan Elasticsearchiin, niin todettiin Logstashin olevan turha välissä. Jos datan määrä nousee rajusti, tulee Logstash ehkä olemaan tarpeellinen.

3. Usean rivin loki

Filebeat-ohjelmaa jouduttiin säätämään useasti, jotta Kibanaan saatiin järkevää tietoa. Todettiin, että paras vaihtoehto on kuvan 58 mukaiset asetukset. Koska lokit eivät ole aivan yhtenäisiä kaikkialla niin ei voida käyttää esimerkiksi aikamerkintää merkkinä uudelle riville.

4. Metricbeat 5.0 (Alpha) testaaminen

Metricbeat-ohjelma olisi yhdistelmä Topbeat ja Packetbeat ohjelmista, joilla voisi seurata Apachen tuotteita tehokkaasti järjestelmän ohella. Ohjelmaa asentaessa huomattiin, että Elasticin tekemiä näkemyksiä ei voinut olla samaan aikaan kuin muiden Beats-tuotteiden, vaan rikkoivat toisensa.

5.3 Jatkokehitys

Alla mahdollisia kehityskohteita, jolla järjestelmää voitaisiin vielä jatkaa monipuolisemmaksi.

1. Lokien seuranta

Aikatauluista johtuen ei omien tuotteiden lokeihin saatu yksilöityä istuntotunnusta, joten ei voinut testata kuinka seuraaminen onnistuu Kibanassa, heti kuin uuden version lokitukseen saa, pystyy asian toteamaan ja testaamaan, että toimiiko.

2. Ympäristön suorituskyky

Aikatauluista johtuen ei tuotteiden asentamista ja säätämistä kaikille ympäristöille onnistuttu tekemään. Tulevien palaverien jälkeen tiedetään kunkin Beat-ohjelman datan tallennustarpeet ja -määrät, jonka jälkeen voidaan tuotteet kaikkialle asentaa ja todentaa, että onko beat-tuotteet riittäviä, vai tarvitseeko logstash -ohjelman lisäksi.

3. Ympäristön seuranta

Palvelua käyttävien kanssa keskustellessa tuli ilmi, kuinka sitä olisi hyvä seurata esimerkiksi isosta näytöstä joka olisi selkeästi nähtävillä yrityksen

tiloissa. Aikataulusta johtuen ei näyttöä keretty asentamaan lopulliselle paikallensa. Näytössä näkemys, josta pelkällä vilkaisulla näkee palvelimien ja tuotteiden tilanteen.

4. Elastalert palveluksi

Aikataulun puitteissa en kerennyt Elastalertista tehdä palvelua, jolloin ei palvelua tarvitsisi käydä manuaalisesti ajamassa palvelimella, vaan ohjelma pyörisi itse yötä päivää.

5. Palvelimen siirto

Elasticsearch-palvelin pitäisi siirtää eri verkkoon, jotta data saadaan lähetettyä kaikilta tuotanto-ympäristöiltä Elasticsearchiin.

5.4 Käyttöönotto

Käyttöönotto suoritettiin heti kun säädökset oli testattu toimiviksi ja tarpeeksi laajoiksi.

Käyttöönottoa varten luotiin erilliset asennuspaketit tuotannolle ja testille, jotta asentaminen olisi mahdollisimman nopeaa ja helppoa. Erona asennuspaketeilla oli Beats-tuotteiden asetuksissa. Asetuksien erona oli lähetystaajuus, indeksointi ja merkinnät.

Järjestelmän täydellistä käyttöönottoa ei keretty aikataulun puitteissa suorittamaan, mutta kaikki testiympäristöt ja osa tuotantoympäristöistäkin, saatiin jo järjestelmän piiriin.

Käyttöönotto tuotannon-ympäristöille on suunniteltu jo toteutettavaksi, mutta itse Elasticsearch-palvelinta pitää siirtää toiseen verkkoon, jotta kaikki tuotanto-ympäristöt saadaan lähettäämään dataa järjestelmään.

Käyttöönotto suoritetaan heti kun palvelin saadaan siirrettyä oikeaan verkkoon, koska järjestelmä ei vaadi palvelimilta muutakuin Beats-tuotteiden asentamisen eikä esimerkiksi hetkellistä käyttökatkoa.

5.5 Yhteenveto

Opinnäytetyössä oli tavoitteena saada yrityksen tuotteiden lokit ja ympäristöjen tiedot samaan paikkaan selvästi ja tehokkaasti seurattavaksi, käyttämällä Elastic-yrityksen tuotteita.

Yrityksessä palvelimia ja palveluita tarkkailtiin ennen manuaalisesti palvelimilta käsin, joka oli työlästä ja hidasta. Opinnäytetyöni tulos toi mielestäni tähän ongelmaan helpottavan ratkaisun.

Mielestäni aiheena Big data ja sen hallinnointi on erittäin mielenkiintoinen ja kehittyvä tietotekniikan ala, joka on myös yrityksille erityisen tärkeää. Nykyään kaikki asiat synnyttävät jonkinlaista dataa itsestään tai tekemisistään, joten datamäärät saattavat olla todella suuria ja hankalia käsitellä.

Työssä asentamani tuotteet tuntuivat olevan oikein tehokas ratkaisu tähän ongelmaan.

Ennen opinnäytetyötäni en aiheesta juurikaan mitään tiennyt, koska Elastic Co. ei ole suurimmasta päästä yrityksenä, joten dokumentaationkaan löytäminen ei aina ollut helppoa. Silti onnistuin mielestäni työssäni oikein hyvin ja löysin helpottavan ratkaisun yritykselle, jolla asentaminen saatiin vielä yksinkertaisemmaksi, kuin alun perin oli suunniteltu.

Kokemuksena opinnäytetyöni oli erittäin opettavainen ja hyödyllinen. Työtä tehdessäni opin paljon Big datasta, sen hallinnoinnista ja mahdollisuuksista yrityksien käytössä. Lisäksi opinnäytetyöhön liittyen opin paljon yrityksien lokitustavoista ja niiden seurannan tarpeista, jotta asiakasta voidaan palvella paremmin.

Opinnäytetyön tuloksena yrityksellä on järjestelmä, mistä voidaan palvelimia seurata selkeästi ja tehokkaasti. Palvelimilla ei tarvitse enää käydä, ellei järjestelmästä tule hälytystä tai huomaa Kibanassa jotain epäilyttävää. Lopulta sekä helpdesk, että ohjelmistosuunnittelijat voivat käyttää järjestelmää tehokkaana lisänä työnsä tukena. Aikataulun puitteissa ei omien tuotteiden lokia keretty muokkaamaan siten, että olisi saatu järkevät näkemykset myös helpdeskille ja ohjelmistosuunnittelijoille.

Lähdeluettelo

Apache. (n.d.). *Apache Solr 6.1.0 Documentation*. Retrieved Elokuu 25, 2016, from Apache Lucene: http://lucene.apache.org/solr/6_1_0/index.html

Dragland, Å. (2013, toukokuu 22). *Big Data - for better or worse*. Retrieved Syyskuu 28, 2016, from SINTEF: http://www.sintef.no/en/latest-news/big-data--for-better-or-worse/

Elastic Co. (n.d.). *About*. Retrieved Heinäkuu 26, 2016, from Elastic: https://www.elastic.co/about

Elastic Co. (n.d.). *Customers*. Retrieved Heinäkuu 27, 2016, from Elastic: https://www.elastic.co/use-cases

Elastic Co. (n.d.). *Products: Beats*. Retrieved Heinäkuu 27, 2016, from https://www.elastic.co/products/beats

Elastic Co. (n.d.). *Products: Kibana*. Retrieved Heinäkuu 26, 2016, from Elastic: https://www.elastic.co/products/kibana

Elastic Co. (n.d.). *Products: Logstash*. Retrieved Heinäkuu 26, 2016, from Elastic: https://www.elastic.co/products/logstash

Elastic Co. (n.d.). *Products: Marvel*. Retrieved Heinäkuu 27, 2016, from Elastic: https://www.elastic.co/products/marvel

Elastic Co. (n.d.). *Products: Reporting*. Retrieved Syyskuu 20, 2016, from Elastic: https://www.elastic.co/products/reporting

Elastic Co. (n.d.). *Products: Shield*. Retrieved Heinäkuu 27, 2016, from Elastic: https://www.elastic.co/products/shield

Elastic Co. (n.d.). *Products: Watcher*. Retrieved Heinäkuu 27, 2016, from Elastic: https://www.elastic.co/products/watcher

Elastic Co. (n.d.). *Subscriptions*. Retrieved Heinäkuu 26, 2016, from Elastic: https://www.elastic.co/subscriptions

Elastic Co. (n.d.). *You know, for Search...* Retrieved Heinäkuu 26, 2016, from Elastic: https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html

Elastic Co. (n.d.). *Products: Graph*. Retrieved Heinäkuu 27, 2016, from Elastic: https://www.elastic.co/products/graph

Google. (n.d.). *Vertaa*. Retrieved Elokuu 25, 2016, from Google Trends: https://www.google.fi/trends/explore?date=all&q=elasticsearch,splunk,solr

Hundley, B. (2015, Heinäkuu 29). *Optimizing Elasticsearch: How Many Shards per Index?* Retrieved Heinäkuu 29, 2016, from QBOX: https://qbox.io/blog/optimizing-elasticsearch-how-many-shards-per-index

James, J. (2016, Kesäkuu 28). *Data Never Sleeps 4.0.* Retrieved Syyskuu 28, 2016, from DOMO: https://www.domo.com/blog/2016/06/data-never-sleeps-4-0/

Oracle. (n.d.). *Class Level*. Retrieved Syyskuu 28, 2016, from Oracle Docs: https://docs.oracle.com/javase/7/docs/api/java/util/logging/Level.html

Pantz. (2007, Elokuu 13). *Cron and Crontab usage and examples*. Retrieved Elokuu 19, 2016, from Pantz.org: https://www.pantz.org/software/cron/croninfo.html

Tan, K. (n.d.). *The Feature Smackdown*. Retrieved Elokuu 25, 2016, from Apache Solr vs Elasticsearch: http://solr-vs-elasticsearch.com/

Think Big Analytics. (n.d.). Retrieved Elokuu 25, 2016, from Solr vs Elasticsearch: https://thinkbiganalytics.com/solr-vs-elastic-search/

Waal-Montgomery, M. d. (2015, Tammikuu 15). *World's data volume to grow 40% per year & 50 times by 2020: Aureus*. Retrieved Syyskuu 28, 2016, from E27: https://e27.co/worlds-data-volume-to-grow-40-per-year-50-times-by-2020-aureus-20150115-2/

Walker, B. (2015, Huhtikuu 5). *Every Day Big Data Statistics – 2.5 Quintillion Bytes of Data Created Daily*. Retrieved Syyskuu 28, 2016, from VCloudNews: http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/

Viestintävirasto. (n.d.). *Kyberturvallisuus*. Retrieved Heinäkuu 27, 2016, from Viestintävirasto: https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2016/03/ttn20160 3091742.html

Yelp. (2014). *Rule Types and Configuration Options*. Retrieved Elokuu 25, 2016, from Elastalert: https://elastalert.readthedocs.io/en/latest/ruletypes.html#rule-types

Yelp. (2014). *Rule Types and Configuration Options*. Retrieved Syyskuu 21, 2016, from Elastalert: https://elastalert.readthedocs.io/en/latest/ruletypes.html#alerts

Zhitnitsky, A. (n.d.). *Splunk vs ELK: The Log Management Tools Decision Making Guide*. Retrieved Elokuu 25, 2016, from Takipi Blog: http://blog.takipi.com/splunk-vs-elk-the-log-management-tools-decision-making-guide/

YLEISENOSUUDEN YML-TIEDOSTO

```
################  Output  ################
# Configure what outputs to use when sending the data collected by the beat.
# Multiple outputs may be used.
output:

  ### Elasticsearch as output
  elasticsearch:
    # Array of hosts to connect to.
    # Scheme and port can be left out and will be set to the default (http and 9200)
    # In case you specify and additional path, the scheme is required: http://localhost:9200/path
    # IPv6 addresses should always be defined as: https://[2001:db8::1]:9200
    hosts: ["elastic.XXXXXXX.fi:9200"]

    # Optional protocol and basic auth credentials.
    #protocol: "https"
    #username: "admin"
    #password: "s3cr3t"

    # Number of workers per Elasticsearch host.
    #worker: 1

    # Optional index name. The default is "topbeat" and generates
    # [topbeat-]YYYY.MM.DD keys.
    index: "topbeat-testi"

    # A template is used to set the mapping in Elasticsearch
    # By default template loading is disabled and no template is loaded.
    # These settings can be adjusted to load your own template or overwrite existing ones
    #template:

      # Template name. By default the template name is topbeat.
      #name: "topbeat"

      # Path to template file
      #path: "topbeat.template.json"

      # Overwrite existing template
      #overwrite: true

    # Optional HTTP Path
    #path: "/elasticsearch"

    # Proxy server url
    #proxy_url: http://proxy:3128

    # The number of times a particular Elasticsearch index operation is attempted. If
```

```
  # the indexing operation doesn't succeed after this many retries, the events are
   # dropped. The default is 3.
   #max_retries: 3

   # The maximum number of events to bulk in a single Elasticsearch bulk API index
request.
   # The default is 50.
   #bulk_max_size: 50

   # Configure http request timeout before failing an request to Elasticsearch.
   #timeout: 90

   # The number of seconds to wait for new events between two bulk API index requests.
   # If `bulk_max_size` is reached before this interval expires, addition bulk index
   # requests are made.
   #flush_interval: 1

   # Boolean that sets if the topology is kept in Elasticsearch. The default is
   # false. This option makes sense only for Packetbeat.
   #save_topology: false

   # The time to live in seconds for the topology information that is stored in
   # Elasticsearch. The default is 15 seconds.
   #topology_expire: 15

   # tls configuration. By default is off.
   #tls:
     # List of root certificates for HTTPS server verifications
     #certificate_authorities: ["/etc/pki/root/ca.pem"]

     # Certificate for TLS client authentication
     #certificate: "/etc/pki/client/cert.pem"

     # Client Certificate Key
     #certificate_key: "/etc/pki/client/cert.key"

     # Controls whether the client verifies server certificates and host name.
     # If insecure is set to true, all server host names and certificates will be
     # accepted. In this mode TLS based connections are susceptible to
     # man-in-the-middle attacks. Use only for testing.
     #insecure: true

     # Configure cipher suites to be used for TLS connections
     #cipher_suites: []

     # Configure curve types for ECDHE based cipher suites
     #curve_types: []

     # Configure minimum TLS version allowed for connection to logstash
```

```
    #min_version: 1.0

    # Configure maximum TLS version allowed for connection to logstash
    #max_version: 1.2


### Logstash as output
#logstash:
  # The Logstash hosts
  #hosts: ["loki.XXXXX.fi:5044"]

  # Number of workers per Logstash host.
  #worker: 1

  # Set gzip compression level.
  #compression_level: 3

  # Optional load balance the events between the Logstash hosts
  #loadbalance: true

  # Optional index name. The default index name depends on the each beat.
  # For Packetbeat, the default is set to packetbeat, for Topbeat
  # top topbeat and for Filebeat to filebeat.
  #index: topbeat-log

  # Optional TLS. By default is off.
  #tls:
    # List of root certificates for HTTPS server verifications
    #certificate_authorities: ["C:\topbeat\topbeat-1.2.3-windows\cert.crt"]

    # Certificate for TLS client authentication
    #certificate: "/etc/pki/client/tls.pem"

    # Client Certificate Key
    #certificate_key: "/etc/pki/client/cert.key"

    # Controls whether the client verifies server certificates and host name.
    # If insecure is set to true, all server host names and certificates will be
    # accepted. In this mode TLS based connections are susceptible to
    # man-in-the-middle attacks. Use only for testing.
    #insecure: true

    # Configure cipher suites to be used for TLS connections
    #cipher_suites: []

    # Configure curve types for ECDHE based cipher suites
    #curve_types: []
```

```
 ### File as output
  #file:
    # Path to the directory where to save the generated files. The option is mandatory.
    #path: "/tmp/topbeat"

    # Name of the generated files. The default is `topbeat` and it generates files: `topbeat`,
`topbeat.1`, `topbeat.2`, etc.
    #filename: topbeat

    # Maximum size in kilobytes of each file. When this size is reached, the files are
    # rotated. The default value is 10 MB.
    #rotate_every_kb: 10000

    # Maximum number of files under path. When this number of files is reached, the
    # oldest file is deleted and the rest are shifted from last to first. The default
    # is 7 files.
    #number_of_files: 7

  ### Console output
  # console:
    # Pretty print json event
    #pretty: false

################  Shipper  ################

shipper:
  # The name of the shipper that publishes the network data. It can be used to group
  # all the transactions sent by a single shipper in the web interface.
  # If this options is not defined, the hostname is used.
  #name:

  # The tags of the shipper are included in their own field with each
  # transaction published. Tags make it easy to group servers by different
  # logical properties.
  #tags: ["service-X", "web-tier"]

  # Uncomment the following if you want to ignore transactions created
  # by the server on which the shipper is installed. This option is useful
  # to remove duplicates if shippers are installed on multiple servers.
  #ignore_outgoing: true

  # How often (in seconds) shippers are publishing their IPs to the topology map.
  # The default is 10 seconds.
  #refresh_topology_freq: 10

  # Expiration time (in seconds) of the IPs published by a shipper to the topology map.
  # All the IPs will be deleted afterwards. Note, that the value must be higher than
```

```
  # refresh_topology_freq. The default is 15 seconds.
  #topology_expire: 15

  # Internal queue size for single events in processing pipeline
  #queue_size: 1000

  # Configure local GeoIP database support.
  # If no paths are not configured geoip is disabled.
  #geoip:
    #paths:
    #  - "/usr/share/GeoIP/GeoLiteCity.dat"
    #  - "/usr/local/var/GeoIP/GeoLiteCity.dat"

################  Logging  ################

# There are three options for the log ouput: syslog, file, stderr.
# Under Windos systems, the log files are per default sent to the file output,
# under all other system per default to syslog.
logging:

  # Send all logging output to syslog. On Windows default is false, otherwise
  # default is true.
  #to_syslog: true

  # Write all logging output to files. Beats automatically rotate files if rotateeverybytes
  # limit is reached.
  to_files: false

  # To enable logging to files, to_files option has to be set to true
  files:
    # The directory where the log files will written to.
    path: C:\topbeat\topbeat-1.2.3-windows

    # The name of the files where the logs are written to.
    name: mybeat.log

    # Configure log file size limit. If limit is reached, log file will be
    # automatically rotated
    rotateeverybytes: 10485760 # = 10MB

    # Number of rotated log files to keep. Oldest files will be deleted first.
    #keepfiles: 7
  # Enable debug output for selected components. To enable all selectors use ["*"]
  # Other available selectors are beat, publish, service
  # Multiple selectors can be chained.
  #selectors: [ ]
  # Sets log level. The default log level is error.
  # Available log levels are: critical, error, warning, info, debug
  level: info
```

PACKETBEATIN SÄÄTÖOSUUS

```
################  Packetbeat Configuration Example  ################

# This file contains an overview of various configuration settings. Please consult
# the docs at https://www.elastic.co/guide/en/beats/packetbeat/current/packetbeat-configuration.html
# for more details.

# The Packetbeat shipper works by sniffing the network traffic between your
# application components. It inserts meta-data about each transaction into
# Elasticsearch.

################  Sniffer  ################

# Select the network interfaces to sniff the data. You can use the "any"
# keyword to sniff on all connected interfaces.
interfaces:
  device: 0

################  Protocols  ################
protocols:
  dns:
    # Configure the ports where to listen for DNS traffic. You can disable
    # the DNS protocol by commenting out the list of ports.
    ports: [53]

    # include_authorities controls whether or not the dns.authorities field
    # (authority resource records) is added to messages.
    # Default: false
    include_authorities: true
    # include_additionals controls whether or not the dns.additionals field
    # (additional resource records) is added to messages.
    # Default: false
    include_additionals: true

    # send_request and send_response control whether or not the stringified DNS
    # request and response message are added to the result.
    # Nearly all data about the request/response is available in the dns.*
    # fields, but this can be useful if you need visibility specifically
    # into the request or the response.
    # Default: false
    # send_request:  true
    # send_response: true

  http:
    # Configure the ports where to listen for HTTP traffic. You can disable
    # the HTTP protocol by commenting out the list of ports.
    ports: [80, 8080, 8000, 5000, 8002]
```

```
# Uncomment the following to hide certain parameters in URL or forms attached
  # to HTTP requests. The names of the parameters are case insensitive.
  # The value of the parameters will be replaced with the 'xxxxx' string.
  # This is generally useful for avoiding storing user passwords or other
  # sensitive information.
  # Only query parameters and top level form parameters are replaced.
  # hide_keywords: ['pass', 'password', 'passwd']

memcache:
  # Configure the ports where to listen for memcache traffic. You can disable
  # the Memcache protocol by commenting out the list of ports.
  ports: [11211]

  # Uncomment the parseunknown option to force the memcache text protocol parser
  # to accept unknown commands.
  # Note: All unknown commands MUST not contain any data parts!
  # Default: false
  # parseunknown: true

  # Update the maxvalue option to store the values - base64 encoded - in the
  # json output.
  # possible values:
  #    maxvalue: -1  # store all values (text based protocol multi-get)
  #    maxvalue: 0   # store no values at all
  #    maxvalue: N   # store up to N values
  # Default: 0
  # maxvalues: -1

  # Use maxbytespervalue to limit the number of bytes to be copied per value element.
  # Note: Values will be base64 encoded, so actual size in json document
  #       will be 4 times maxbytespervalue.
  # Default: unlimited
  # maxbytespervalue: 100

  # UDP transaction timeout in milliseconds.
  # Note: Quiet messages in UDP binary protocol will get response only in error case.
  #       The memcached analyzer will wait for udptransactiontimeout milliseconds
  #       before publishing quiet messages. Non quiet messages or quiet requests with
  #       error response will not have to wait for the timeout.
  # Default: 200
  # udptransactiontimeout: 1000

mysql:
  # Configure the ports where to listen for MySQL traffic. You can disable
  # the MySQL protocol by commenting out the list of ports.
  ports: [3306]

pgsql:
  # Configure the ports where to listen for Pgsql traffic. You can disable
  # the Pgsql protocol by commenting out the list of ports.
```

```
 ports: [5432]

  redis:
    # Configure the ports where to listen for Redis traffic. You can disable
    # the Redis protocol by commenting out the list of ports.
    ports: [6379]

  thrift:
    # Configure the ports where to listen for Thrift-RPC traffic. You can disable
    # the Thrift-RPC protocol by commenting out the list of ports.
    ports: [9090]

  mongodb:
    # Configure the ports where to listen for MongoDB traffic. You can disable
    # the MongoDB protocol by commenting out the list of ports.
    ports: [27017]

################  Processes  ################

# Configure the processes to be monitored and how to find them. If a process is
# monitored then Packetbeat attempts to use it's name to fill in the `proc` and
# `client_proc` fields.
# The processes can be found by searching their command line by a given string.
#
# Process matching is optional and can be enabled by uncommenting the following
# lines.
#
#procs:
#  enabled: false
#  monitored:
#    - process: mysqld
#      cmdline_grep: mysqld
#
#    - process: pgsql
#      cmdline_grep: postgres
#
#    - process: nginx
#      cmdline_grep: nginx
#
#    - process: app
                #      cmdline_grep: gunicorn
```

FILEBEATIN SÄÄTÖOSUUS

```
################  Filebeat Configuration Example  ################

################  Filebeat  ################
filebeat:
  # List of prospectors to fetch data.
  prospectors:
    # Each - is a prospector. Below are the prospector specific configurations
    -
      # Paths that should be crawled and fetched. Glob based paths.
      # To fetch all ".log" files from a specific level of subdirectories
      # /var/log/*/*.log can be used.
      # For each file found under this path, a harvester is started.
      # Make sure not file is defined twice as this can lead to unexpected behaviour.
      paths:
        - /var/log/*.log
        #- c:\programdata\elasticsearch\logs\*

      # Configure the file encoding for reading files with international characters
      # following the W3C recommendation for HTML5
(http://www.w3.org/TR/encoding).
      # Some sample encodings:
      #   plain, utf-8, utf-16be-bom, utf-16be, utf-16le, big5, gb18030, gbk,
      #    hz-gb-2312, euc-kr, euc-jp, iso-2022-jp, shift-jis, ...
      #encoding: plain

      # Type of the files. Based on this the way the file is read is decided.
      # The different types cannot be mixed in one prospector
      #
      # Possible options are:
      # * log: Reads every line of the log file (default)
      # * stdin: Reads the standard in
      input_type: log

      # Exclude lines. A list of regular expressions to match. It drops the lines that are
      # matching any regular expression from the list. The include_lines is called before
      # exclude_lines. By default, no lines are dropped.
      # exclude_lines: ["^DBG"]

      # Include lines. A list of regular expressions to match. It exports the lines that are
      # matching any regular expression from the list. The include_lines is called before
      # exclude_lines. By default, all the lines are exported.
      # include_lines: ["^ERR", "^WARN"]

      # Exclude files. A list of regular expressions to match. Filebeat drops the files that
      # are matching any regular expression from the list. By default, no files are dropped.
      # exclude_files: [".gz$"]

      # Optional additional fields. These field can be freely picked
```

```
# to add additional information to the crawled log files for filtering
    #fields:
    #  level: debug
    #  review: 1

    # Set to true to store the additional fields as top level fields instead
    # of under the "fields" sub-dictionary. In case of name conflicts with the
    # fields added by Filebeat itself, the custom fields overwrite the default
    # fields.
    #fields_under_root: false

    # Ignore files which were modified more then the defined timespan in the past.
    # In case all files on your system must be read you can set this value very large.
    # Time strings like 2h (2 hours), 5m (5 minutes) can be used.
    #ignore_older: 0

    # Close older closes the file handler for which were not modified
    # for longer then close_older
    # Time strings like 2h (2 hours), 5m (5 minutes) can be used.
    #close_older: 1h

    # Type to be published in the 'type' field. For Elasticsearch output,
    # the type defines the document type these entries should be stored
    # in. Default: log
    #document_type: log

    # Scan frequency in seconds.
    # How often these files should be checked for changes. In case it is set
    # to 0s, it is done as often as possible. Default: 10s
    #scan_frequency: 10s

    # Defines the buffer size every harvester uses when fetching the file
    #harvester_buffer_size: 16384

    # Maximum number of bytes a single log event can have
    # All bytes after max_bytes are discarded and not sent. The default is 10MB.
    # This is especially useful for multiline log messages which can get large.
    #max_bytes: 10485760

    # Mutiline can be used for log messages spanning multiple lines. This is common
    # for Java Stack Traces or C-Line Continuation
    #multiline:

      # The regexp Pattern that has to be matched. The example pattern matches all lines
starting with [
      #pattern: ^\[

      # Defines if the pattern set under pattern should be negated or not. Default is false.
      #negate: false
```

```
# Match can be set to "after" or "before". It is used to define if lines should be append to a pattern
    # that was (not) matched before or after or as long as a pattern is not matched based on negate.
    # Note: After is the equivalent to previous and before is the equivalent to to next in Logstash
    #match: after

    # The maximum number of lines that are combined to one event.
    # In case there are more the max_lines the additional lines are discarded.
    # Default is 500
    #max_lines: 500

    # After the defined timeout, an multiline event is sent even if no new pattern was found to start a new event
    # Default is 5s.
    #timeout: 5s

  # Setting tail_files to true means filebeat starts readding new files at the end
  # instead of the beginning. If this is used in combination with log rotation
  # this can mean that the first entries of a new file are skipped.
  #tail_files: false

  # Backoff values define how agressively filebeat crawls new files for updates
  # The default values can be used in most cases. Backoff defines how long it is waited
  # to check a file again after EOF is reached. Default is 1s which means the file
  # is checked every second if new lines were added. This leads to a near real time crawling.
  # Every time a new line appears, backoff is reset to the initial value.
  #backoff: 1s

  # Max backoff defines what the maximum backoff time is. After having backed off multiple times
  # from checking the files, the waiting time will never exceed max_backoff idenependent of the
  # backoff factor. Having it set to 10s means in the worst case a new line can be added to a log
  # file after having backed off multiple times, it takes a maximum of 10s to read the new line
  #max_backoff: 10s

  # The backoff factor defines how fast the algorithm backs off. The bigger the backoff factor,
  # the faster the max_backoff value is reached. If this value is set to 1, no backoff will happen.
  # The backoff value will be multiplied each time with the backoff_factor until max_backoff is reached
  #backoff_factor: 2
```

```
    # This option closes a file, as soon as the file name changes.
    # This config option is recommended on windows only. Filebeat keeps the files it's
reading open. This can cause
    # issues when the file is removed, as the file will not be fully removed until also
Filebeat closes
    # the reading. Filebeat closes the file handler after ignore_older. During this time no
new file with the
    # same name can be created. Turning this feature on the other hand can lead to loss
of data
    # on rotate files. It can happen that after file rotation the beginning of the new
    # file is skipped, as the reading starts at the end. We recommend to leave this option
on false
    # but lower the ignore_older value to release files faster.
    #force_close_files: false

  # Additional prospector
  #-
    # Configuration to use stdin input
    #input_type: stdin

# General filebeat configuration options
#
# Event count spool threshold - forces network flush if exceeded
#spool_size: 2048

# Enable async publisher pipeline in filebeat (Experimental!)
#publish_async: false

# Defines how often the spooler is flushed. After idle_timeout the spooler is
# Flush even though spool_size is not reached.
#idle_timeout: 5s

# Name of the registry file. Per default it is put in the current working
# directory. In case the working directory is changed after when running
# filebeat again, indexing starts from the beginning again.
registry_file: "C:/ProgramData/filebeat/registry"

# Full Path to directory with additional prospector configuration files. Each file must
end with .yml
# These config files must have the full filebeat config part inside, but only
# the prospector part is processed. All global options like spool_size are ignored.
# The config_dir MUST point to a different directory then where the main filebeat
config file is in.
#config_dir:
```