Otto Kupiainen

# Elastic Cloud -palvelun käyttö pilvi-palvelujen monitoroinnissa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mobile Solutions

Insinöörityö

29.5.2024

# Tiivistelmä


| | |
|---|---|
| Tekijä: | Otto Kupiainen |
| Otsikko: | Elastic Cloud -palvelun käyttö pilvipalvelujen monitoroinnissa |
| Sivumäärä: | 32 sivua |
| Aika: | 29.5.2024 |
| | |
| Tutkinto: | Insinööri (AMK) |
| Tutkinto-ohjelma: | Tieto- ja viestintätekniikka |
| Ammatillinen pääaine: | Mobile Solutions |
| Ohjaajat: | Lehtori Matti Peltoniemi |


---


Opinnäytetyössä tarkoituksena oli tutkia saatavilla olevia monitorointiratkaisuja pilvipalveluiden monitorointiin ja suunnitella sekä toteuttaa valittu Elastic Cloud -ratkaisu. Valitusta ratkaisusta toteutettiin ensin testiversio. Testiversion täyttäessä toimeksiantajan asettamat tavoitteet, toteutettaisiin monitorointiratkaisusta myös tuotantoversio. Tällä tuotantoympäristöllä korvattaisiin toimeksiantajan aikaisempi Elastic-toteutus. Toimeksiantajan kanssa tavoitteiksi asetettiin aiempaa helpompi ylläpito, kattavampi palvelun tilan monitorointi ja monitorointiin liittyvät hälytykset sekä laajempi lokien keräys häiriötilanteiden selvittämisen helpottamiseksi.

Markkinoilla saatavilla olevia vaihtoehtoja tutkittiin erityisesti julkisesti saatavilla olevan tiedon pohjalta. Saatavilla olevien vaihtoehtojen tutkimisen jälkeen päädyttiin ohjelmistopalveluna toteutettuun versioon monitorointiohjelmistosta. Valitusta vaihtoehdosta toteutettiin testiympäristö, jota hyödyntäen tutustuttiin käytännössä monitorointiratkaisun tarjoamiin ominaisuuksiin. Testiympäristö otettiin käyttöön tuotantokäyttöä vastaavaan tietojärjestelmien monitorointiin. Testiympäristö täytti sille asetetut vaatimukset, joten työn osana toteutettiin myös tuotantoympäristö. Valittu Elastic Cloud -monitorointiratkaisu täytti tavoitteet hyvin helpon käyttöönoton ja tiedon keräämisen takia. Sen perustavanlaatuiset ominaisuudet olivat tarpeeksi toimivia myös jatkuvan monitoroinnin ja häiriötilanteiden selvittämisen tueksi. Tuotantoympäristö konfiguroitiin käyttöönottovaiheessa hyvin samanlaiseksi kuin testiympäristö. Ympäristöä konfiguroitaessa valvonnan piiriin lisättiin kaikki tarpeelliseksi koetut toimeksiantajan palvelut niin saatavuuden kuin lokien ja muun tiedon keräämisen osalta.

Ohjelmistopalveluna toteutetun monitorointiohjelmiston voitiin todeta olevan hyvä ratkaisu organisaatiolle, joka ei halua käyttää merkittävästi omia resurssejaan tietojärjestelmiensä monitorointiin. Isännöimällä monitorointiratkaisun itse ja tarkemmalla konfiguraatiolla olisi todennäköisesti päästy vielä parempaan lopputulokseen, mutta tämä olisi vaatinut enemmän aikaa ja osaamista sekä aiheuttanut enemmän kustannuksia. Toteutettu ratkaisu oli helpompi ylläpitää kuin toimeksiantajan edeltävä ratkaisu, mutta se vaatii silti tasaisin väliajoin ammattitaitoisen käyttäjän ylläpitotoimia.



Avainsanat: monitorointi, pilvipalvelut, loki, keskitetty lokienhallinta, Elastic


---


Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

# Abstract


| | |
|---|---|
| Author: | Otto Kupiainen |
| Title: | Monitoring cloud applications with Elastic Cloud |
| Number of Pages: | 32 pages |
| Date: | 29th May 2024 |
| | |
| Degree: | Bachelor of Engineering |
| Degree Programme: | Information technology |
| Professional Major: | Mobile Solutions |
| Supervisors: | Matti Peltoniemi, Senior Lecturer |


---


The purpose of this thesis was to research available monitoring solutions for cloud computing and to plan and implement the selected solution. At first a test version was implemented for assessing the selected monitoring solution. When the test version was found to fulfill the requirements set by the thesis client, a production version of the selected monitoring solution was implemented. This production version would replace the existing Elastic solution used by the thesis client. Together with the thesis client, easier maintenance, better monitoring and logging coverage and alerting were selected as the goals of the project.

The available monitoring solutions were mainly evaluated with publicly available information. After the evaluation, a software as a service solution was selected. A test version of the selected solution was created to test its features in practice. The test version was taken into use with monitoring tasks similar to the planned production use case. The test version fulfilled the requirements and as agreed, a production version was created. The selected solution fulfilled the requirements well with its easy start of use and data ingesting. Once the production version of the selected monitoring solution was taken into use, its basic functionalities were found to be satisfactory in providing continuous monitoring and solving disruptions in systems. The production environment was configured very similar to the latest state of the test version. All resources of the thesis client that were deemed to need monitoring, started to be monitored with the production environment. This included logging, availability monitoring and other forms of data collection.

Using a software as a service monitoring solution was found to be a good choice for organizations that do not want to use a considerable amount of its own resources on system monitoring. An even better performing monitoring solution could have been probably possible by self-hosting it and configuring it in more detail. The implemented monitoring solution was easier to maintain than the previous one, but it still regularly needs maintenance from a knowledgeable user.



Keywords: Monitoring, Cloud Computing, Log, Centralized Logging, Elastic

# Sisällys

Lyhenteet

1 Johdanto 1

2 Tietojärjestelmien monitorointi ja valvottavuus 2

3 Monitorointiratkaisut ja pilvipalvelut 8

3.1 Yleisesti käytössä olevia vaihtoehtoja 8
3.2 Elastic-ohjelmisto 10
3.3 Pilvipalvelut 19

4 Elastic Cloud -toteutus 22

4.1 Projektin tausta ja tavoitteet 22
4.2 Toteutus 23
4.3 Käyttöönotossa ilmenneitä asioita 30

5 Yhteenveto 31

Lähteet 33

## Lyhenteet

APM: *Application performance monitoring*. Sovelluksen suorituskyvyn monitorointi.

ECS: *Elastic Common Schema*. Elasticin standardi tapahtumatiedon tietorakenteelle.

ES: *Elasticsearch.* Lyhenne Elasticsearchin nimestä.

ELK: *Elasticsearch, Logstash, Kibana*. Lyhenne Elastic Stackin, joka tunnettiin aikaisemmin nimellä ELK Stack, kolmen perustana olevan ohjelmiston nimistä.

HTTP: *Hypertext Transfer Protocol*. Hypertekstin siirtoprotokolla.

KQL: *Kibana Query Language*. Kibanan hakuihin käytettävä hakukieli.

REST: *Representational state transfer*. Arkkitehtuuriratkaisu ohjelmointirajapintojen toteuttamiseen.

# 1 Johdanto

Tämän opinnäytetyön tarkoituksena on tutkia tarjolla olevia ohjelmistoja ja palveluita pilvipalveluiden monitorointiin sekä toteuttaa valittu Elastic Cloud -monitorointiratkaisu.

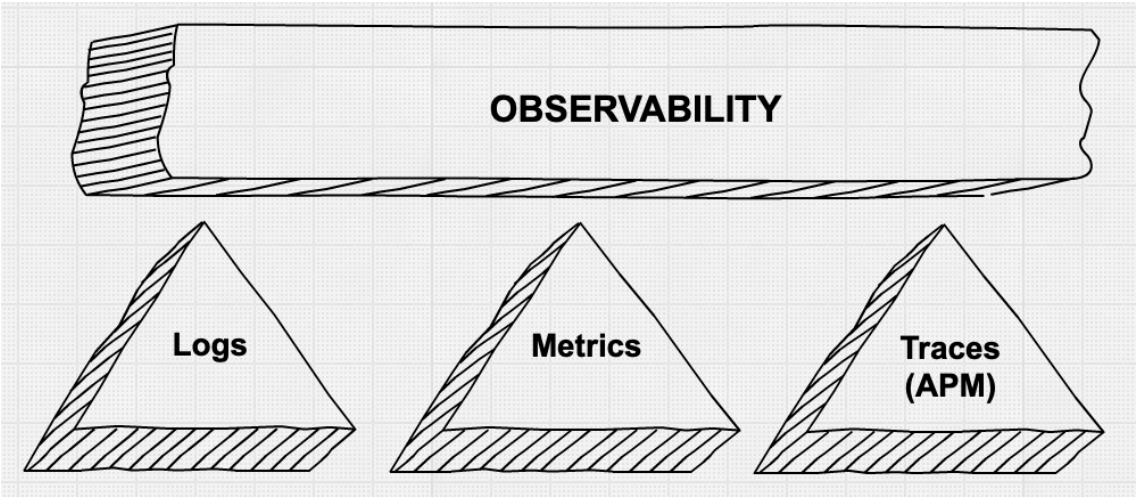
Työssä käsitellään monitorointiohjelmistojen käyttöä pilvessä isännöityjen, pääosin Linux-palvelinten ja niillä suoritettavien sovellusten monitorointiin. Keskeisiä käsitteitä ja teknologioita kuten lokienhallintaa, monitorointia ja pilvipalveluita tutkitaan toteutettavan käyttöönoton pohjustamiseksi. Toteutusvaiheessa suunnitellaan ja käyttöönotetaan valittu Elastic Cloud -pohjainen monitorointiratkaisu toimeksiantajan käyttöön. Elastic Cloud on pilvessä isännöity Elastic-kokonaisuus, joka rakentuu Elasticsearch-hakukoneen ympärille. Monitorointiratkaisusta toteutetaan ensin testiversio, jota hyödyntäen tutkitaan valitun Elastic Cloud -ratkaisun soveltuvuus toimeksiantajan käyttöön. Testiversion täyttäessä toimeksiantajan asettamat tavoitteet toteutetaan monitorointiratkaisusta myös tuotantoversio.

Toimeksiantajan käytössä oleva edellinen ratkaisu ei enää täytä toimeksiantajan palveluidensa monitoroinnille asettamia vaatimuksia. Nyt toteutettavalla monitorointiratkaisulla kerätään palvelinten metriikkaa, lokeja sekä tietoa sovellusten tilasta ja hyödynnetään kerättyä tietoa tilannekuvan muodostamiseen Elasticin tarjoamilla työkaluilla. Toteutetun ratkaisun esittelyssä keskitytään erityisesti sovellusten suorituskyvyn, suoritusvarmuuden ja palvelinten metriikan monitorointiin.

Opinnäytetyön toimeksiantajana toimii TrueMed Oy. TrueMed on Espoossa toimiva vuonna 2019 perustettu IT-alan yritys. TrueMed kehittää tekoälyratkaisuja pääasiassa lääkkeiden väärentämisen ehkäisyyn.

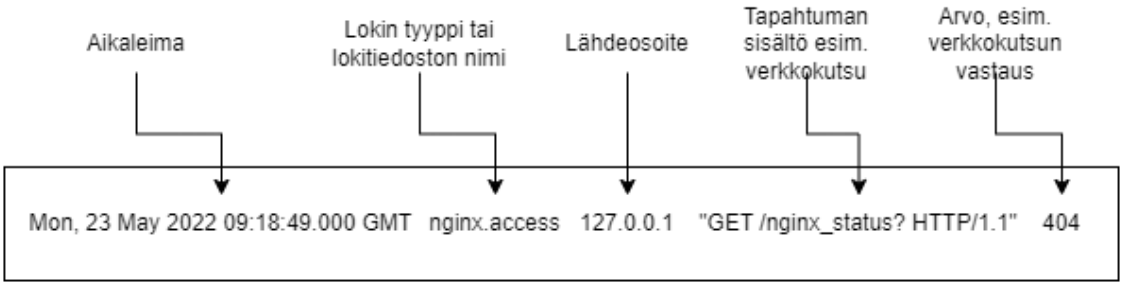
# 2 Tietojärjestelmien monitorointi ja valvottavuus

Järjestelmien omistajien, ylläpitäjien ja joskus myös käyttäjien täytyy olla tietoisia järjestelmän tilasta. Erityisesti ylläpitäjien täytyy seurata järjestelmään kohdistuvaa kuormaa ja siellä tapahtuvia asioita. Ylläpitäjien ja omistajien tulee kyetä jälkikäteen selvittämään, mitä järjestelmässä on tapahtunut, miksi ja kuka toimet on tehnyt. Tätä vaativat niin tietoturva, ohjelmistokehitys kuin tarve tietää järjestelmän tila ja saatavuus. Erityisesti tietoturvaan liittyvää tilannekuvan tarvetta ohjaavat siihen liittyvät lait, asetukset tai sopimusehdot. Tämän lisäksi saatavuudesta on usein ohjelmistoja ja järjestelmiä hankittaessa sovittu asiakkaan ja toimittajan välisissä sopimuksissa. Tavoiteltu valvottavuus voidaan saavuttaa lokeja, metriikkaa ja muuta tietoa keräämällä, tallentamalla ja keskitetysti analysoimalla. Alalla käytetään usein englanninkielistä termiä observability kuvaamaan sekä järjestelmien ja sovellusten valvottavuutta että siihen käytettäviä ratkaisuja. Valvottavuutta lähestytään usein tiedonkeruun kautta. Valvottavuuden kolme peruspilaria liittyvätkin oleellisesti monitoroinnissa hyödynnettävän tiedon keräämiseen. Kuten kuvassa 1 on esitetty, nämä kolme peruspilaria ovat lokit, metriikka ja niin sanotut jäljet (traces). Myös termiä havaittavuus voidaan käyttää puhuttaessa ylläpitäjän kyvystä tietää, mitä järjestelmässä tapahtuu. (1.)




Kuva 1. Valvottavuuden kolme peruspilaria: lokit, metriikka ja niin sanotut jäljet (traces) (1).

Lokilla tarkoitetaan aikajärjestyksessä talletettua kirjanpitoa tapahtumista ja niiden aiheuttajista. Siitä käy ilmi, mitä on tapahtunut, milloin ja kuka toimet on tehnyt. Kuvassa 2 on esitetty esimerkki lokin rakenteesta. Tässä työssä tarkoitettu loki syntyy, kun tapahtumat tietojärjestelmissä, sovelluksissa, tietoverkoissa ja tietosisällöissä kirjataan lokiin. Tätä prosessia kutsutaan lokitukseksi. (2.)




Kuva 2. Esimerkki yksittäisen lokimerkinnän rakenteesta.

Lokeja on erilaisia eri tarkoituksiin. Niiden pääasiallinen tarkoitus on tarjota tietoa siitä, mitä esimerkiksi tietojärjestelmässä on tapahtunut. Tapahtumat voivat olla ohjelmiston suorittamisessa tapahtuneita virheitä tai käyttäjän suorittamia toimintoja. Viestintäviraston (3) ohje lokien käyttämiseen listaa seuraavat lokityypit:

- Ylläpitolokiin kirjataan järjestelmän toimintaan tehdyt muutokset. Kerätty tieto muutoksista helpottaa versionhallinnassa.

- Käyttö- tai tapahtumalokiin kirjataan käyttäjien kirjautumistoiminta sekä tietoa järjestelmän prosesseista. Tämän kaltainen loki on hyvin yleinen useimmissa tietojärjestelmissä.

- Muutoslokiin kirjataan seurattavan tiedon muutokset, mukaan lukien lisäykset ja poistot.

- Virhelokiin kirjataan järjestelmässä ilmenneet virheet ja mahdollisuuksien mukaan niiden aiheuttajat. Kattavasti koostetun virhelokin perusteella

virheiden aiheuttajat on helpompi korjata verrattuna tilanteeseen, jossa tapahtuneista virheistä ei lokitietoa ole tallennettu.

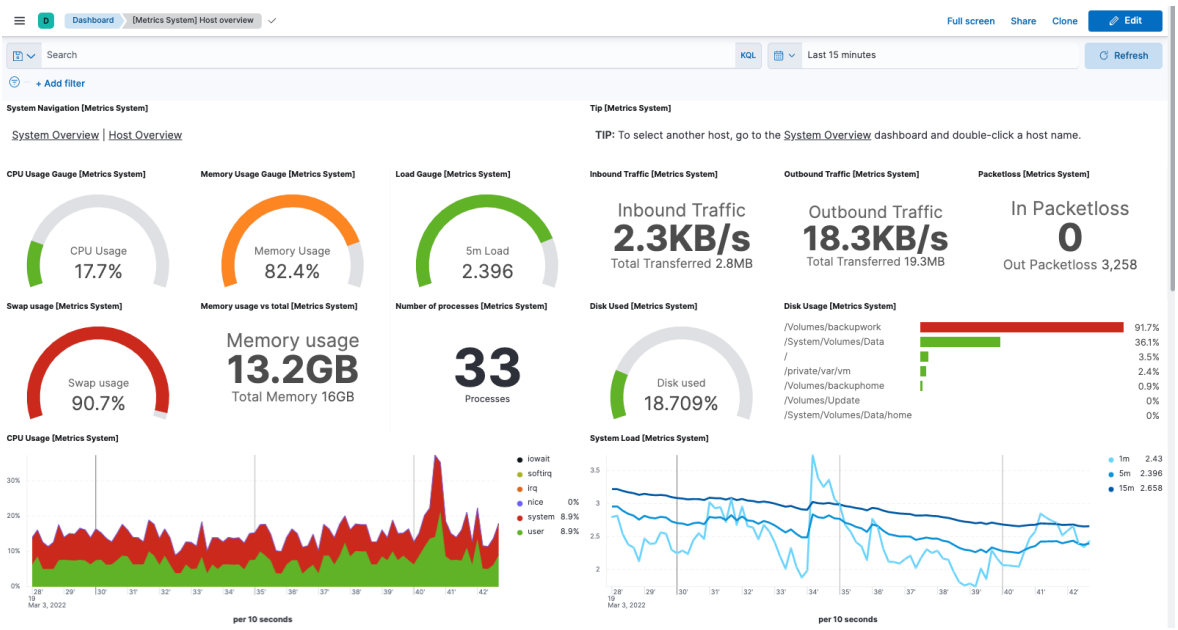
- Viestintälokia voidaan kerätä käydystä viestinnästä. Sen avulla voidaan esimerkiksi näyttää toteen tietyn viestinvaihdon olemassaolo esimerkiksi tietoturvapoikkeamia tutkittaessa.

- Haltijalokiin kirjataan, kenen hallussa jokin lokitettavaksi määritetty asia on ollut milloinkin. Näin voidaan toimia esimerkiksi fyysisten laitteiden tai salausavaimien hallussapitoa ja käyttöä seurattaessa.

- Pääsynvalvontalokiin kirjataan kirjautumiset ja kulkutunnisteiden käyttö, niin onnistuneet kuin epäonnistuneetkin. Näin saadaan kerättyä tärkeää tietoa turvallisuus- tai tietoturvapoikkeamien havaitsemista ja analysointia varten.

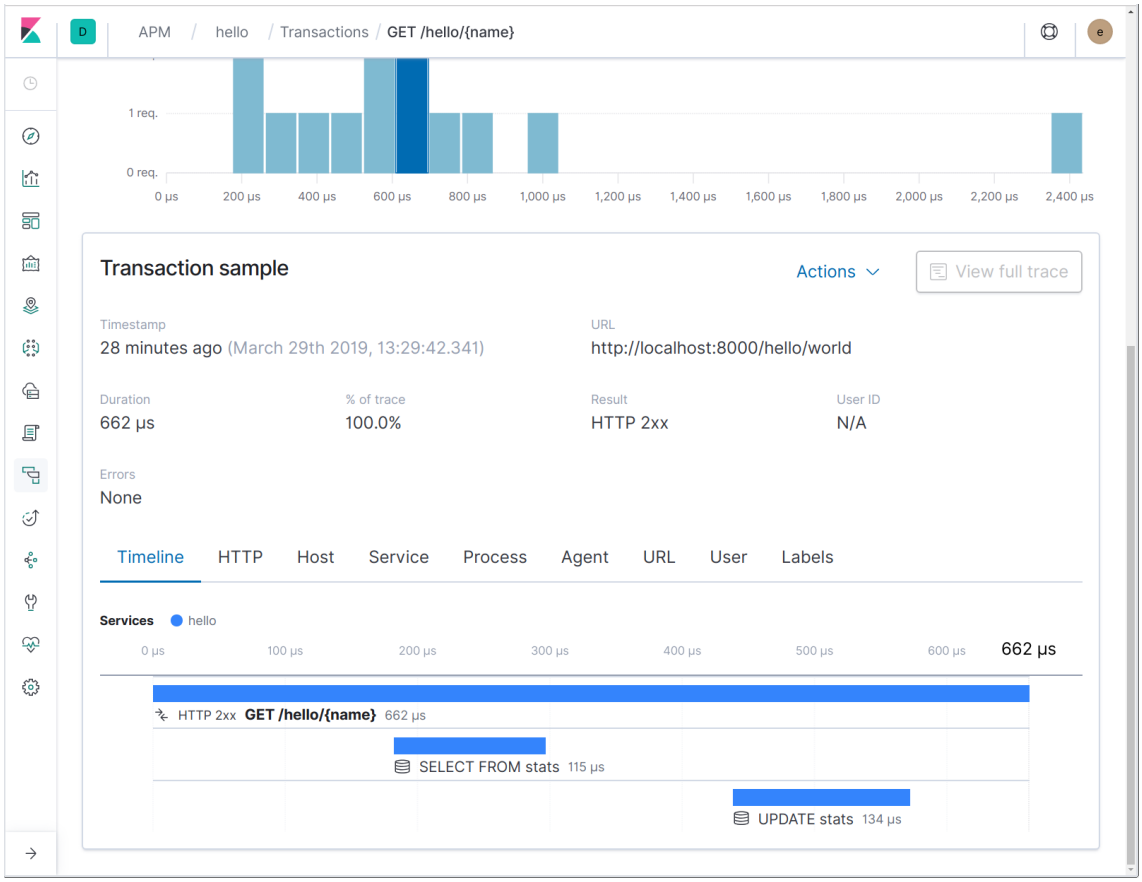
Tietojärjestelmän valvottavuutta rakennettaessa on siis tärkeää tunnistaa tärkeät tapahtumat ja tallettaa ne lokeihin. Edellisessä luettelossa esitellyt lokityypit yhdessä muodostavat valvottavuuden ensimmäisen tukipilarin. (1.)

Valvottavuuden toisella tukipilarilla, metriikalla, tarkoitetaan kohdeisännästä, usein palvelimelta tai muulta tietokoneesta mitattua resurssien käyttöä. Yleisesti mitattuja ja seurattuja metriikoita ovat prosessorien ja muistin kuormitus, verkkoliikenteen määrä ja tallennustilan käyttö. Nämä ovat vahvasti fyysiseen infrastruktuuriin liittyviä metriikoita, ja niitä useimmiten mitataankin yksittäistä fyysistä tietokonetta kohden. Virtualisoidun ympäristön tapauksessa metriikka mitataan ja esitetään yleensä käyttöjärjestelmän tasalla. Kerätty metriikka voidaan esittää visualisoituna. Kuvassa 3 on havainnollistettu yleisten metriikoiden visualisointi Elasticin Kibanalla.




Kuva 3. Monitoroitavan palvelimen metriikkaa Elasticin Kibana-ohjelmistolla visualisoituna (4).

Sovellusten monitorointia varten voidaan kerätä myös niin sanottuja jälkiä. Ne ovat jonkun toimen järjestelmässä kulkemaa reittiä kuvaavaa tietoa. Toimi voi olla joko järjestelmän käyttäjän suorittama tai järjestelmään ohjelmoitu prosessi. Kuvassa 4 on esitetty yksinkertainen jälki, jossa on mitattu käyttäjän verkkokutsuun ja sen seurauksena sovelluksen suorittamaan tietokantahakuun kulunutta aikaa. Käyttäjän suorittama toimi siis aiheutti verkkokutsun ja tietokantahaun, joista jäi monitorointiohjelmistoon yksityiskohtainen jälki. Tämän tyyppinen tieto on avuksi sovelluksen suorituskykyä optimoidessa tai suorituksessa ilmenneiden virheiden korjaamisessa. Sovellusten monitorointi on monitoroinnin osa-alue, jossa keskitytään sovelluksen tai sovellusten suorituskyvyn, suoritusvarmuuden ja käyttäjäkokemuksen monitorointiin. (5.)

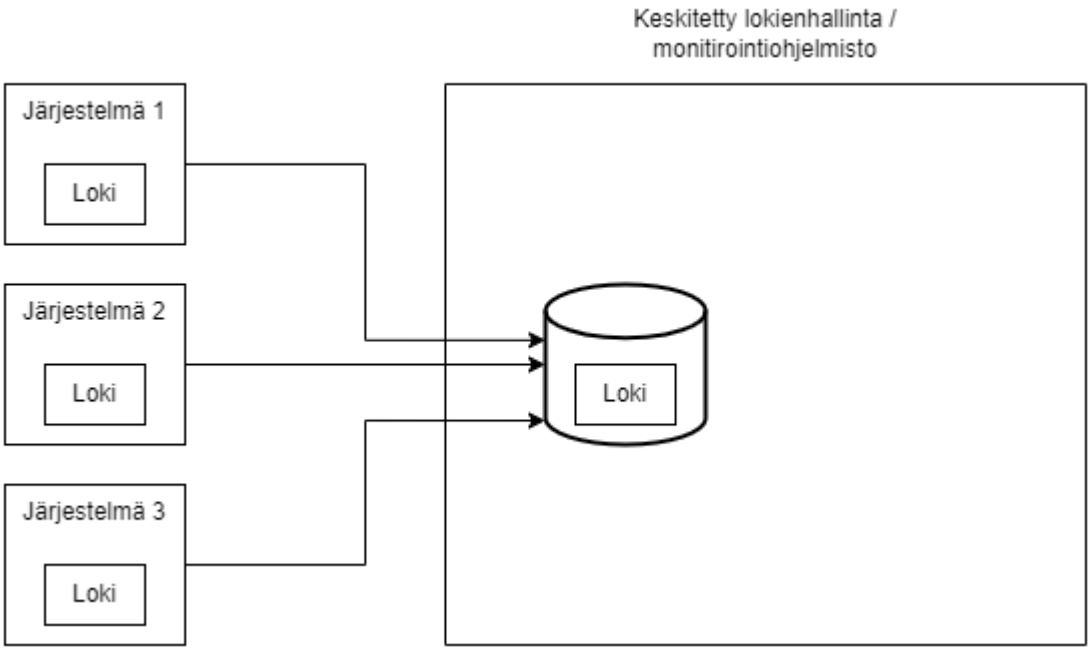



Kuva 4. Elasticin blogissaan julkaisema esimerkki APM:n transaktionäytteestä, jossa näkyy verkkokutsuun ja tietokantahakuun kulunut aika (6).

Jatkuvan valvottavuuden saavuttamiseksi tulee lokien tallentaminen ja analysointi automatisoida mahdollisimman pitkälle. Lokien ja niiden perusteella laukaistujen hälytysten avulla järjestelmän poikkeamat voidaan havaita mahdollisimman ajoissa. Helposti saatavilla oleva tieto poikkeamista nopeuttaa myös järjestelmien palauttamista. (7.)

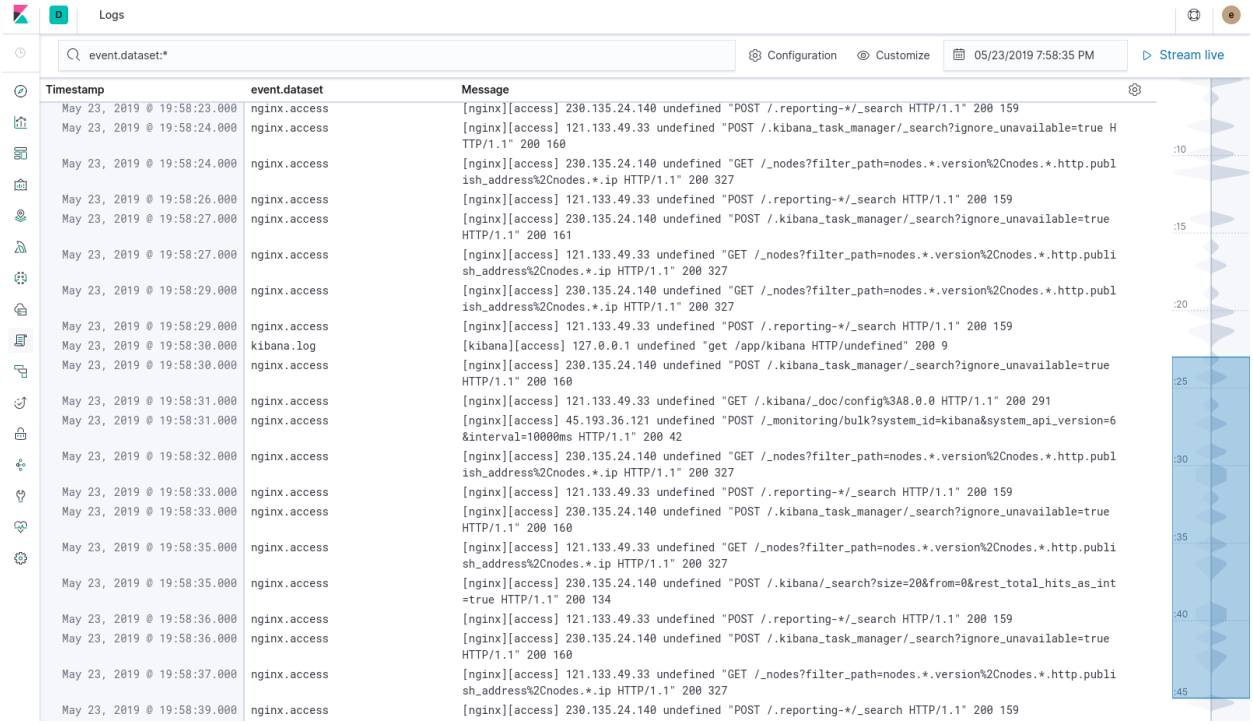
Monitorointiohjelmistoilla tarkoitetaan lokien ja metriikan seurantaan, tallentamiseen, hakemiseen ja analysointiin tarkoitettuja ohjelmistoja. Niitä kutsutaan usein myös keskitetyiksi lokienhallintajärjestelmiksi, ja tällöin toiminnan keskiössä ovat nimenomaan lokit. (2.) Lokit muodostavat useimmissa tapauksissa tärkeimmän tiedonlähteen järjestelmän tilannekuvaa muodostettaessa. Monitorointiohjelmistot voivat olla käyttäjänsä isännöimiä, tarjoajan isännöimiä tai niitä voi käyttää ohjelmistopalveluna (8). Usein pilvipalveluilla on omat monitorointityökalunsa palvelun sisällä. Palvelukohtaisilla monitorointityökaluilla ei ole

kuitenkaan mahdollista monitoroida muita palveluita. Kuvassa 5 on havainnollistettu keskitetyn lokienhallinnan periaatetta.




Kuva 5. Keskitetyn lokienhallinnan havainnollistaminen (2).

Elasticin (6) dokumentaatiossa esitettyä esimerkkiä voidaan pitää hyvänä yksinkertaisena esimerkkinä kohdeisännän monitoroinnista. Dokumentaatiossa Elastic-ohjelmisto otetaan käyttöön Linux-palvelimen monitoroinnissa. Tarvittava ohjelmisto asennetaan palvelimelle, josta se lähettää tarvittavan lokitiedon asetusten mukaisesti keskitettyyn lokienhallintaan Elastic Cloud -palveluun. Esimerkin konfiguraatiossa palvelimelta saadaan metriikkatietoa ja järjestelmälokeja, joita voidaan hyödyntää tilannekuvan luomiseen ja ylläpitämiseen kyseisestä järjestelmästä. Kuvassa 6 on esitetty ote esimerkin lokisyötteestä Elasticin Logs-näkymässä. (6).




Kuva 6. Kerättyjä lokeja Elasticin lokisyötteessä (9).

# 3 Monitorointiratkaisut ja pilvipalvelut

Tässä luvussa esitellään markkinoilla saatavilla olevia monitorointiratkaisuja sekä pilvipalvelujen periaatetta.

## 3.1 Yleisesti käytössä olevia vaihtoehtoja

Markkinoilla on käyttäjän saatavilla paljon erilaisia vaihtoehtoja, joista tässä esitellään merkittävimpiä. Gartner (10) on listannut sovellusten monitorointia käsittelevässä raportissaan alan tärkeimpiä toimijoita. Raportti on siis keskittynyt yhteen monitoroinnin osa-alueeseen, mutta kuvaa yleisellä tasolla markkinoita. Kuten kuvasta 7 on nähtävissä, Gartner on jaotellut toimijat nelikenttään vision kokonaisuuden (completeness of vision) ja toteutuskyvyn (ability to execute). mukaan.

Kuva 7. Monitorointiohjelmistoja toimittavien yritysten jakautuminen Gartnerin raportissa (10).

Johtaviksi toimijoiksi on nostettu suositut Dynatrace, Datadog, New Relic, Splunk ja Honeycomb. Näillä kaikilla toimijoilla on valikoimassaan monipuolisia työkaluja sovellusten, palvelinten ja infrastruktuurin monitorointiin. Yleensä ominaisuudet tarjotaan osana toimittajan alustaa, johon tieto kerätään ja jossa sitä voidaan sitten seurata ja analysoida. Kaikki nämä johtavat toimijat tarjoavat alustastaan erilaisia tilaustasoja ja alkuun pääsee useimmilla alustoilla ilmaiseksi. Edistyneemmät ominaisuudet vaativat yleensä maksulliselle tilaustasolle sitoutumista.

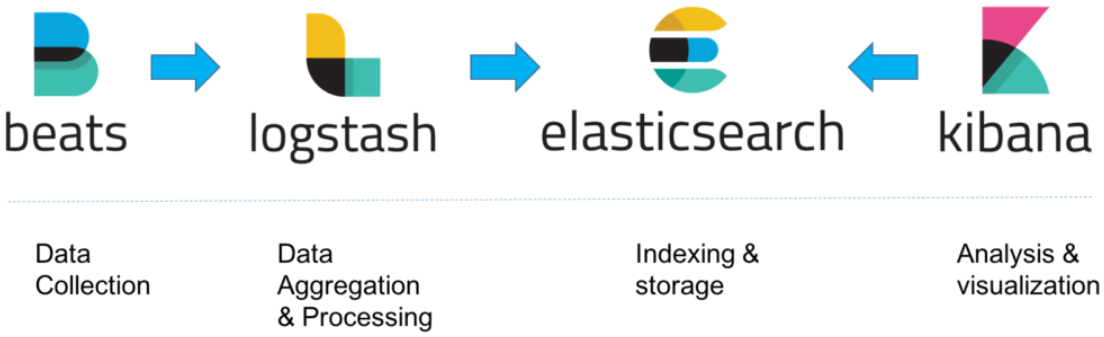
Raportissa haastajiksi asemoidut toimijat ovat suuria teknologiayrityksiä ja muidenkin kuin monitorointiin sopivien palveluiden tarjoajia. Niiden

monitorointiohjelmistotarjonta on keskittynyt joko heidän omille alustoilleen tai on suunnattu komplekseihin ympäristöihin suurille ja keskisuurille toimijoille. (10.) On huomattava, että tämän työn toteutuksen aikana haastajaksi profiloitu Cisco hankki yrityskaupalla Splunkin, yhden johtavista toimijoista (11).

Tässä työssä erityisesti käsiteltävä Elastic on asemoitu Gartnerin raportissa profiiliin visionäärit. Tämän kategorian toimijat poikkeavat selkeästi aiemmin esitellystä johtavien toimijoiden profiilista. Visionääreiksi luokitellut toimijat poikkeavat toisistaan enemmän. Osa tarjoaa vain monitorointiratkaisuja ja osa myös muita palveluita. Visionääreiksi profiloidut toimijat pyrkivät erottumaan johtavista kilpailijoista esimerkiksi hinnoittelulla tai avoimen lähdekoodin ohjelmistoilla. Yllättäviäkin yhtymäkohtia toimijoiden väliltä löytyy, sillä Datadog käyttää Elasticin Elasticsearchia osana alustansa taustajärjestelmää (12).

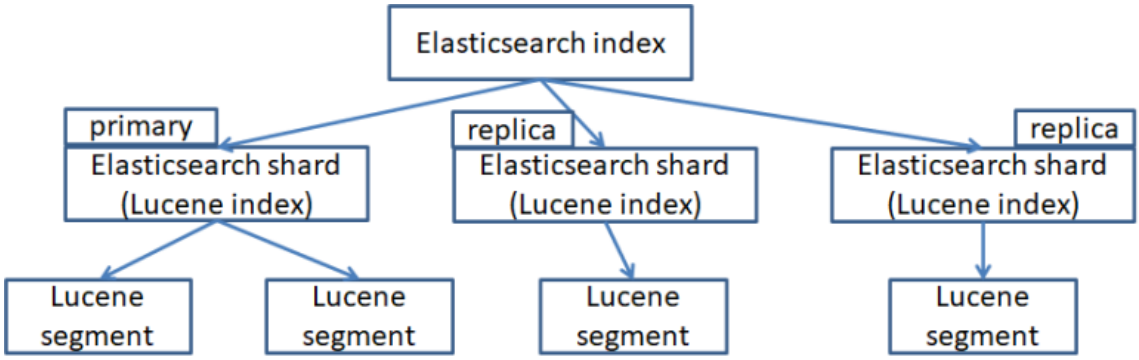
## 3.2 Elastic-ohjelmisto

Elastic Stack, aikaisemmin ELK Stack, on avoimen lähdekoodin projekteina syntynyt monitorointiohjelmisto ja keskitetty lokienhallintajärjestelmä. Sen keskiössä ovat Elasticsearch-hakukone ja analytiikkatyökalu (13), Logstash-tiedonkeruuohjelmisto (14) ja Kibana-analytiikka- ja visualisointityökalu (15). Kibana toimii yleensä Elastic-ratkaisujen graafisena käyttöliittymänä. Kuvassa 8 on esitetty Elastic Stackin tyypillinen tietovuo. Elasticin tuotevalikoimaan kuuluu myös esimerkiksi tietoturvatieto- ja tapahtumahallintatyökalu Elastic Security, joka tuo Kibanaan lisää työkaluja tiedon analysointiin tietoturvan kontekstissa (16).




Kuva 8. Tyypillinen tietovuo Elastic Stackiä käytettäessä (17).

Elasticsearch on avoimen lähdekoodin hakukone ja analytiikkatyökalu, joka pohjautuu Apachen Lucene-hakukoneeseen. Elasticsearch mahdollistaa käyttäjälleen todella nopean tiedon hakemisen ja skaalautuu helposti. Sitä voidaan käyttää usealle eri yksikölle (instance) hajautettuna. Näitä yksittäisiä Elasticsearch-yksiköitä Elastic kutsuu solmuiksi (node). Yhden Elasticsearch-kokonaisuuden solmut voidaan isännöidä eri palvelimilla, ja tämä onkin usein tavoiteltavaa häiriönsietokyvyn kannalta. Elasticsearchiin tallennettu tieto organisoidaan indekseiksi (index) useimmiten tiedon tyypin mukaan. Yksi indeksi muodostuu yhdestä tai useammasta sirpaleesta (shard), jotka ovat teknisesti käytännössä Lucene-indeksejä. Sirpaleet puolestaan pitävät sisällään yhden tai useampia Lucene-segmenttejä. (18.) Elasticsearch-indeksin rakennetta on havainnollistettu kuvassa 9.



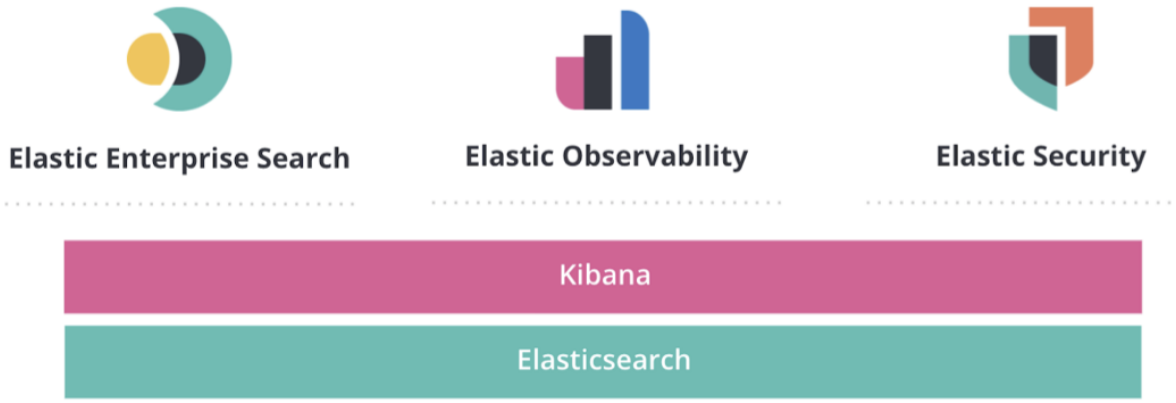

Kuva 9. Elasticsearch-indeksin rakenne (19).

Elasticsearch tukee useita eri tiedontallennuksen tasoja (data tiers). Elasticsearch-kokonaisuuden ei tarvitse sisältää kaikkia tasoja solmuissaan, vaan käyttäjä voi toteuttaa kokonaisuuden tarpeidensa ja resurssiensa mukaan. Yksittäisen indeksin tai tietotyypin ei myöskään tarvitse kulkea kaikkien kokonaisuudesta löytyvien tasojen kautta, vaan indeksin elinkaari voidaan määritellä indeksikohtaisesti. Tämän prosessin nimi on indeksin elinkaaren hallinta (index lifecycle management, ILM). (20.)

Elasticsearchin dokumentaatio (21) sisältää viisi tiedontallennuksen tasoa. Content tier eli sisältötaso on tarkoitettu muun kuin aikasarjatiedon tallentamiseen. Tämän tason Elasticsearch-solmu on yleensä optimoitu hakemista varten. Hot

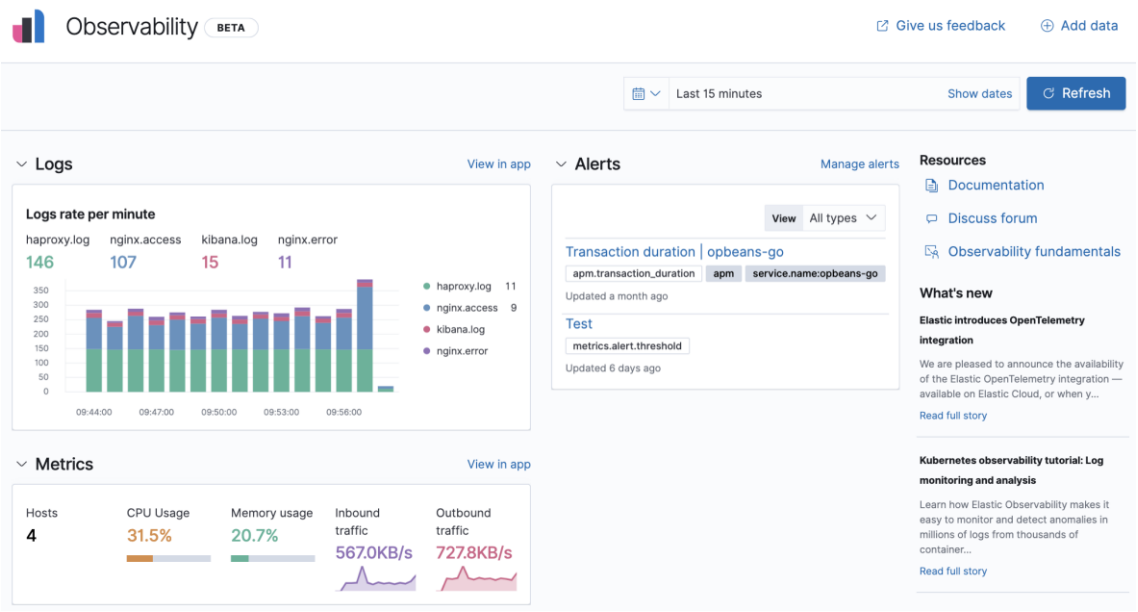
tier eli kuuma taso on välttämätön osa Elasticsearchia. Tieto tallennetaan indekseihin kuumalla tasolla. Myös kuuma taso soveltuu hyvin hakemiseen. Siksi siellä pidetään tuoretta tallennettu tietoa, jota todennäköisesti tullaan hakemaan. Warm tier eli lämmin taso on tarkoitettu tiedolle, jota vielä saatetaan hakea, mutta vähemmän kuin kuumalla tasolla pidettävää. Lämpimän tason kanssa käytettävän infrastruktuurin ei tarvitse olla yhtä tehokasta ja nopeaa kuin kuumalla tasolla. Cold tier eli kylmä taso on tarkoitettu tiedolle, jota täytyy hakea harvemmin, ja tämä taso onkin optimoitu tiedon säilytystä varten. Frozen tier eli jäätynyt taso on tarkoitettu tiedon pitkäaikaiseen säilytykseen silloin, kun tallennettua tietoa ei enää todennäköisesti tarvitse hakea. Viimeistään tällä tasolla tietoa ei enää säilytetä kokonaisina indekseinä vaan vedoksena (snapshot) sinne tallennetusta tiedosta. Tämä vähentää tarvittavaa tallennustilaa ja parantaa häiriönsietoa mutta hidastaa hakemista.

Elastic Cloud on ohjelmistopalveluna tarjottu versio Elastic Stackista. Ohjelmiston isännöinnin hallinnoi tällöin Elastic itse, mutta kaikki sen ominaisuudet ovat käyttäjän saatavilla. Luonnollisesti kaikkia ympäristön isännöintiin liittyviä ominaisuuksia hallinnoidakseen käyttäjän on käytettävä itse isännöityä Elastic Stack -toteutusta. Elastic on jakanut Elastic Stackin eri toiminnallisuudet Elastic Cloud -ohjelmistopalvelussa kolmeen osa-alueeseen: Elastic Enterprice Search, Elastic Observability ja Elastic Security. Elastic kutsuu näitä tuotteensa osa-alueita ratkaisuiksi (solution). Pohjalla on kuitenkin tässäkin tapauksessa sama teknologia kuin itse isännöidyissä Elastic-versioissa. Tätä on havainnollistettu kuvassa 10. (22.)




Kuva 10. Elastic Cloud -ohjelmistopalvelun kolme osa-aluetta ja niiden pohjana olevat teknologiat havainnollistettuna (22).

Toteutusmallista riippumatta Elastic-ratkaisussa on yleensä Elasticsearch tiedon tallennusta ja hakemista varten sekä Kibana tämän tiedon visualisointiin. Kuvassa 11 on esitetty esimerkki tiedon visualisoinnista Kibanalla. Kibanan sijasta käyttöliittymänä voidaan käyttää myös käyttäjän itsensä tai kolmannen osapuolen luomaa ratkaisua. Elasticsearch käyttää kommunikointiin REST-ohjelmointirajapintaa, joka mahdollistaa periaatteessa minkä tahansa yhteensopivan käyttöliittymän käyttämisen (23). REST (Representational State Transfer) on ohjelmistorajapintojen yleinen arkkitehtuurityyli. Siinä tiedon siirtotapana käytetään HTTP:tä, mutta esitystapaa ei ole sidottu (24). HTTP on tiedonsiirtoprotokolla, jota käytetään internetissä (25). Sitä käyttävistä ohjelmointirajapinnoista käytetään joskus myös termiä RESTful (26). Tiedon kerääminen Elasticsearchiin tallennettavaksi voidaan tehdä monella eri tavalla. Tiedon keräämistä käsitellään myöhemmin tässä luvussa.




Kuva 11. Näyttökuva Kibanan Discover-näkymästä, jossa on visualisoitu Kibanalla lokeja, metriikkaa ja muuta tietoa (27).

Elastic Cloud on tällä hetkellä saatavilla neljällä eri tilaustasolla: Standard, Gold, Platinum ja Enterprise. Elastic käyttää kaikilla tilaustasoilla tuntiveloitusta toteutuneen käytön mukaan. Tilauksen kustannuksia voi arvioida käyttämällä Elasticin laskurisivustoa. (28.)

Elastic Cloudin tilaustasoista Standard tarjoaa karsitun Elastic Stackin, joka on kuitenkin riittävä esimerkiksi tässä työssä käsiteltävään lokienhallintaan. Standard on Elastic Cloud -tilauksista edullisin, sen tilaamalla käyttäjä maksaa minimikonfiguraatiolla vain Elasticsearch-instanssin isännöintikulut. Mukana tulevasta Kibana-instanssista Elastic ei veloita erikseen. Esimerkkilaskelma hinnanmuodostuksesta Elasticin laskurisivustolta kuvassa 12.

Kuva 12. Esimerkki pienimmän mahdollisen Elastic Cloud Standard -ympäristön tuntilaskutuksesta (28).

Gold tarjoaa hieman enemmän ominaisuuksia Standardiin verrattuna, esimerkiksi usein hyvin olennaiset ilmoitukset hälytyksistä kolmansiin palveluihin. Platinum ja Enterprise tuovat huomattavasti enemmän ominaisuuksia, ja ne on selkeästi suunnattu isoille organisaatioille sekä käyttäjille, joilla on pieniä ja

keskisuuria organisaatioita korkeammat vaatimukset esimerkiksi saatavuuden, haettavuuden, tietojen menetyksen estämisen ja turvallisuuden osalta. Platinum ja Enterprise -tilauksiin sisältyy myös vuorokauden ympäri saatavilla oleva asiakaspalvelu sekä koneoppimis- ja tekoälytyökaluja. (29.)

On myös asiakkaan valittavissa, missä Elastic isännöi Elastic Cloud -ympäristön. Valittavina ovat julkisen pilven suuret toimijat Amazon Web Services, Microsoft Azure ja Google Cloud ja näiden tarjoamat saatavuusvyöhykkeet (availability zone). Pilvipalveluista ja julkisesta pilvestä kerrotaan tarkemmin luvussa 3.3.

Tiedon kerääminen – Elastic Agent, Elastic Beats ja APM-agentit

Tiedon, eli tässä tapauksessa lokisyötteen ja palvelinmetriikan, keräämiseen on Elasticsearchin tapauksessa kaksi ratkaisua. Perinteinen Beats ja uudempi, 2021 saataville tullut Elastic Agent (30). Beatsit ovat yksinkertaisia tiedonlähetysohjelmistoja (data shipper) ja niitä on jokaiselle Elasticin tietotyypille omansa. Elastic Agent taas kerää näitä kaikkia tietotyyppejä agentiin liitettyjen integraatioiden mukaan. Yleisesti voi sanoa, että Elastic Agent ja Fleet tähtäävät helpompaan käyttöönottoon ja hallintaan Beatsin tarjoaman ominaisuuksien räätälöinnin kustannuksella. (31.) Sovellusten monitorointiin Elastic tarjoaa APM-agentteja. Ne asennetaan ja konfiguroidaan haluttuun sovellukseen, josta ne lähettävät tietoa sovelluksen toiminnasta. (32.)

Elastic Beats

Beatsit ovat yksinkertaisia tiedonlähetysohjelmistoja (data shipper), ja niitä on jokaiselle Elasticin tietotyypille, kuten lokeille ja metriikalle, omansa. Beats mahdollistaa käyttäjälleen Elastic Agentia monipuolisemmat mahdollisuudet konfigurointiin, kuten lokien printtaus palvelimelle luotavaan tiedostoon. (31.) Elastic Beats -ohjelmistoperhe koostuu seitsemästä tiedonkeräysohjelmistosta:

- Filebeat yleisesti lokien keräämiseen

- Heartbeat saatavuustiedon keräämiseen
- Metricbeat metriikan keräämiseen
- Packetbeat verkkoliikennetiedon keräämiseen
- Winlogbeat Windows-lokien keräämiseen
- Functionbeat Pilvipalvelu-lokien keräämiseen
- Auditbeat tarkkailulokien (audit log) keräämiseen.

Beats-ohjelmistoja on siis erilaisille tietotyypeille omansa. (33.)

Beatsit täytyy päivittää, konfiguroida tai poistaa suoraan kohdepalvelimelta. (31.) Vaihtoehtoisesti nämä toimenpiteet voidaan suorittaa myös käyttämällä kolmannen osapuolen hallintatyökalua, kuten esimerkiksi Ansiblea. Ansible on sovellusten ja järjestelmien ylläpitoon tarkoitettu automaatio-ohjelmisto. Se ottaa SSH-yhteyden kohdeisäntään ja suorittaa sille konfiguroidut ylläpitotoimet komentorivillä. (34.)

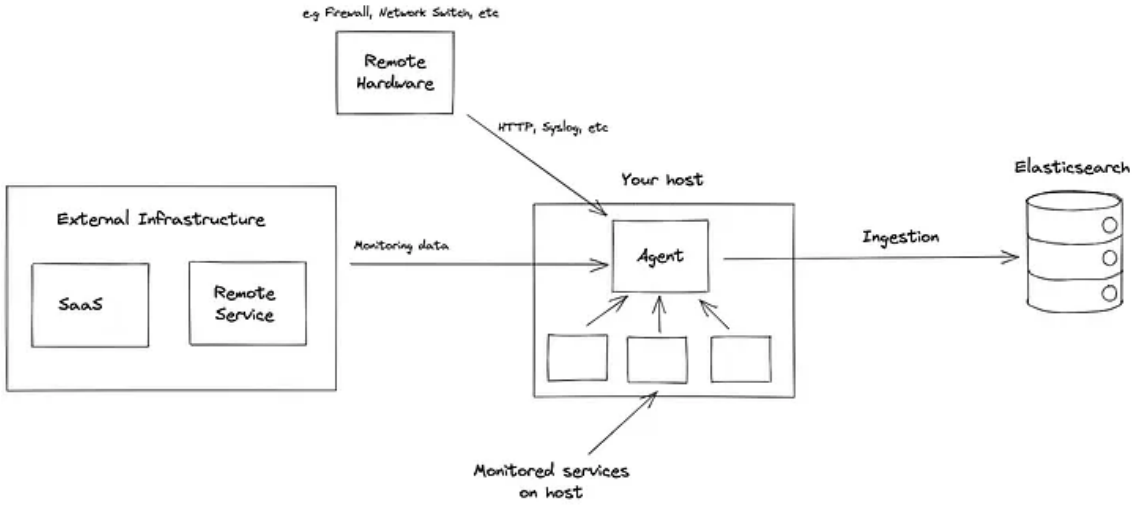
Elastic Agent ja Fleet-palvelin

Elastic Agent on ohjelmisto, jota käytetään tiedon keräämiseen kohdeisännistä (target host). Näitä voivat olla niin fyysiset palvelimet, virtuaaliset palvelimet kuin päätelaitteetkin. Elastic Agent kerää kohteestaan tietotyypit sen perusteella, mitä integraatioita siihen liitettyyn käytäntöön (Agent Policy) kuuluu. Käytännöt ovat konfiguraatiosäännöstöjä, joita voidaan liittää Agentiin. Agentiin täytyy aina olla liitettynä yksi käytäntö, ja yhden käytännön voi liittää useampaan agenttiin. Erityyppisille kohdeisännille kannattaakin olla erilaiset käytännöt. Tällöin kerättävän tietovuon voi optimoida kohteen mukaan. (35.)

Teknisesti Elastic Agent käyttää Beats-tiedonlähetysohjelmistoja varsinaiseen tiedon keräämiseen ja lähettämiseen. Se siis toimii hallintaohjelmistona niin, että sen käyttäjän ei tarvitse hallita käytettyjä Beats-ohjelmistoja. Käytäntöön

(agent policy) konfiguroidut integraatiot määrittävät sen, mitä Beats-ohjelmistoja Elastic Agent kohteeseensa asentaa. Tämä helpottaa käyttöönottoa sekä ylläpitoa merkittävästi, erityisesti, jos käyttäjä ei ole erityisen kokenut vastaavissa työsuoritteissa. (31.)

Fleet-palvelin on Elastic Cloud -ympäristöön kuuluva palvelin, joka pitää yhteyttä yksittäisiin agentteihin. Elastic Agent mahdollistaa myös etähallinnan Fleet-palvelimen kautta. Fleet toimii eräänlaisena isäntänä agenteilleen. Elastic Agent -ohjelmistoa on mahdollista käyttää myös ilman Fleet-palvelinta (stand alone), jolloin niiden hallinta on hoidettava itse.




Kuva 13. Esimerkki tietovuosta käytettäessä Elastic Agenttia ja Fleet-palvelinta (36).

APM

APM eli application performance monitoring, sovelluksen suorituskyvyn monitorointi, on osa Elasticin tarjoamia työkaluja monitorointiin. Se on nimensä mukaan tarkoitettu sovellusten monitorointiin, jolloin keskiössä on sovellusten suorituskyvyn ja suoritusvarmuuden tarkkailu. (32.)

Parhaimman valvottavuuden saavuttaa monitoroimalla sekä itse sovellusta, että käyttäjäsessioita. Tämän lisäksi voidaan käyttää aktiivista testaamista siihen tarkoitetulla testausohjelmistolla. Näiden avulla voidaan luoda ja suorittaa

ajastetusti simuloiden oikean käyttäjän toimia. Tähän voidaan käyttää Elasticin Synthetics-työkaluja tai muiden toimijoiden tuotteita kuten esimerkiksi Jenkins-ohjelmistoa (37).

## 3.3 Pilvipalvelut

Pilvipalveluilla tarkoitetaan verkkoyhteyden välityksellä palveluna tarjottavia tietotekniikan resursseja. Käytettävän resurssin kokoa voidaan säätää, ja resurssit on helppo jakaa usean käyttäjän kesken.

Pilvipalveluja luokitellaan usein palvelumallin mukaan. Keskeisimmät palvelumallit voidaan luokitella seuraavasti:

- Ohjelmistoresurssipalveluilla (Software as a Service, SaaS) tarkoitetaan verkon välityksellä käytettäviä ohjelmistoja ja sovelluksia. Käyttäjä saa tällöin useimmiten käytettäväkseen toimintavalmiin ratkaisun.

- Alustaresurssipalveluilla (Platform as a Service, PaaS) tarkoitetaan verkon välityksellä tarjottua sovelluskehitysympäristöä ja sen apuohjelmistoja. Käyttäjä voi toteuttaa oman ohjelmistonsa tarjotun alustan päällä.

- Infrastruktuuripalveluilla (Infrastructure as a Service, IaaS) tarkoitetaan verkon välityksellä käyttäjälle tarjottua tietokoneiden laskentatehoa, tallennustilaa ja verkkoyhteyksiä. Tällöin käyttäjälle jää vapaus toteuttaa koko järjestelmä omien tarpeidensa mukaan. (38.)

Pilvipalveluiden hyödyntäminen on siis mahdollista hyvinkin erilaisiin tarpeisiin. Pilvipalvelut voidaan jaotella myös niiden toteutusmallin mukaan. Joskus käytetään myös termiä hankintamalli. (7.)

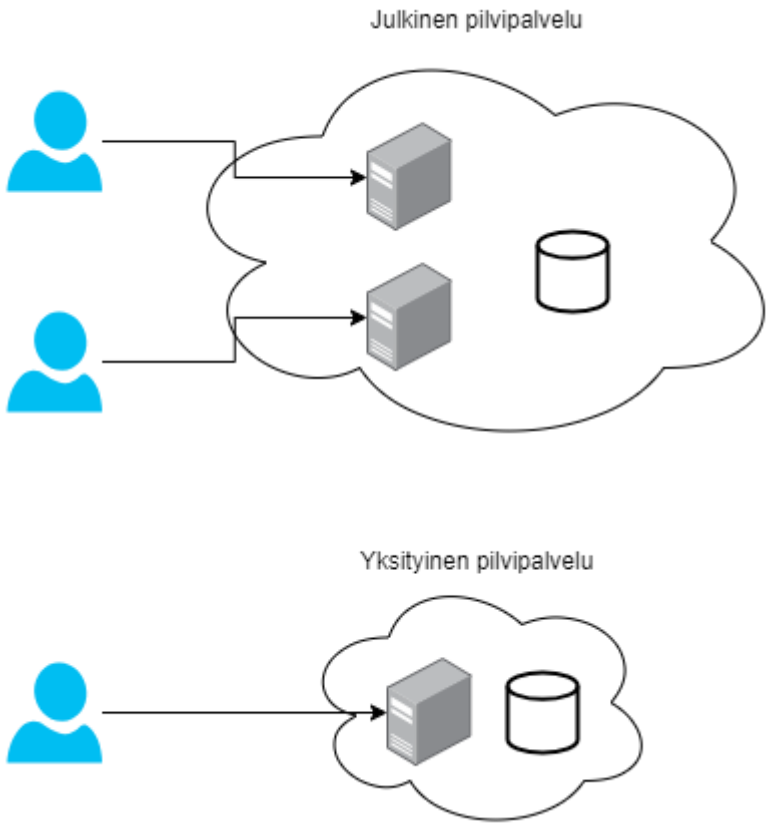
- Julkisella pilvellä tarkoitetaan toteutusmallia, jossa pilvi on saavutettavissa internetin kautta. Julkisen pilven tarjoajia ovat esimerkiksi Google

Cloud, Amazon Web Services ja Microsoft Azure. Julkisen pilven käyttäjiä ovat kaiken kokoiset organisaatiot.

- Yksityisellä pilvellä tarkoitetaan toteutusmallia, jossa pilvi on vain yhden organisaation käytössä ja jossa pilvi sijaitsee yksityisessä verkossa. Tällaisia toteutuksia on pääosin suurilla organisaatioilla.

- Hybridipilvellä tarkoitetaan toteutusmallia, jossa käytetään sekä julkista, että yksityistä pilveä. (39.)

Kuten kuvassa 14 on havainnollistettu, yksityisessä pilvessä kaikki resurssit ovat yhden organisaation käytössä. Julkisessa pilvessä resurssit ovat yleensä useamman organisaation käytössä.

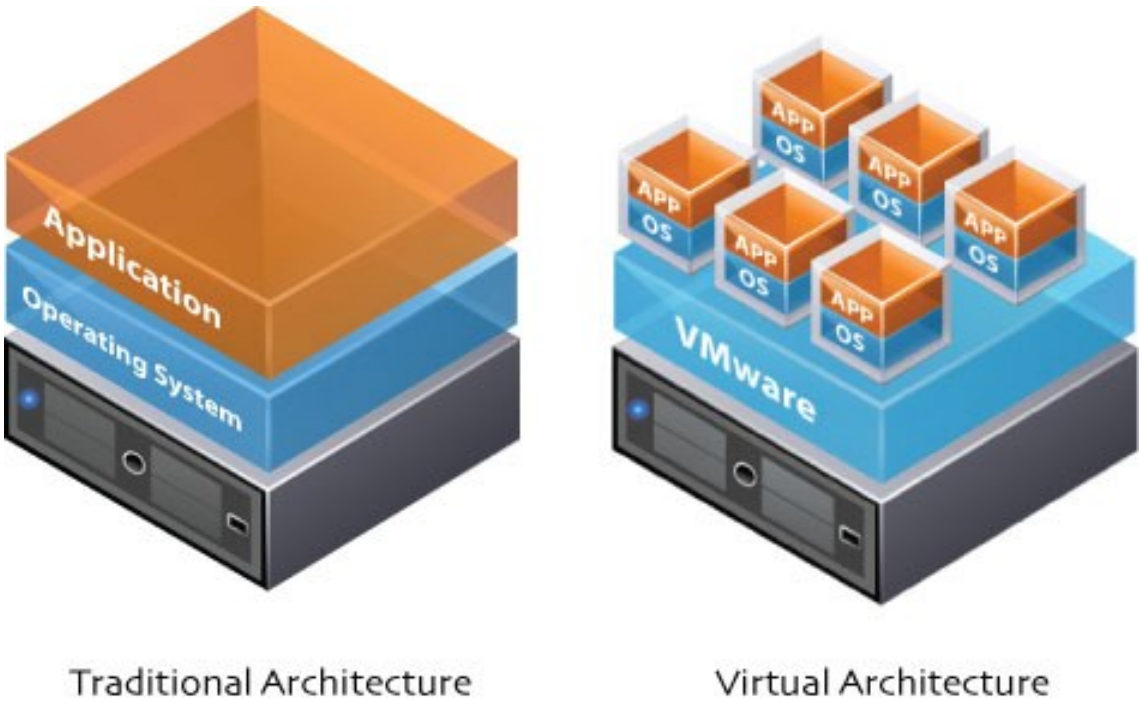



Kuva 14. Julkisen ja yksityisen pilven erot havainnollistettuna (39).

Virtuaalipalvelin

Virtuaalipalvelin on pilvessä isännöity virtualisoitu palvelin, joka jakaa fyysisen palvelininfrastruktuurin muiden virtuaalipalvelimien kanssa. Yhdellä fyysisellä palvelintietokoneella suoritetaan useampia virtualisoituja ilmentymiä (instance) käyttöjärjestelmästä, ja näille virtuaalipalvelimille varataan osansa fyysisistä resursseista kuten laskennasta, muistista ja tallennustilasta. Nämä virtualisoidut palvelinresurssit tarjotaan käyttäjälle pilvipalveluna. (40.)

Virtuaalipalvelin on hyvä esimerkki infrastruktuuripalvelusta. Virtuaalipalvelimen käyttäjällä on vapaus toteuttaa haluamansa kokonaisuus vähin rajoituksin. Käyttäjän valittavaksi jää esimerkiksi käyttöjärjestelmä, verkkoliikenteen konfigurointi ja mahdollisten sovellusten asentaminen. Pilvipalvelun tarjoaja voi tosin tarjota näillekin osa-alueille omia palvelujaan. Käyttäjällä on yleensä valittavissaan erilaisia vaihtoehtoja myös varsinaisen fyysisen infrastruktuurin toteutukseen. Vaihtoehdot luonnollisesti riippuvat pilvipalvelun tarjoajan valikoimasta ja saattavat vaikuttaa eri ratkaisujen toteuttamismahdollisuuksiin. Virtuaalipalvelimen ja perinteisen palvelimen ero on havainnollistettu kuvassa 15.




Kuva 15. Perinteisen palvelimen ja virtuaalipalvelimen ero havainnollistettuna (41).

Tässä opinnäytetyössä keskitytään julkiseen pilven käyttöön opinnäytetyön luonteesta sekä julkisen pilven saatavuudesta ja kustannuksista johtuen. Opinnäytetyössä tullaan hyödyntämään virtuaalipalvelinta monitorointiohjelmiston kohdeisäntänä.

# 4 Elastic Cloud -toteutus

Opinnäytetyössä toteuttiin TrueMed Oy:lle Elastic Cloud -ohjelmistopalvelun käyttöönoton kevään 2022 aikana. Toteutus oli täydessä tuotantokäytössä heinäkuussa 2022. Tämä raportti ei käsittele ympäristöön tämän jälkeen tehtyjä muutoksia.

## 4.1 Projektin tausta ja tavoitteet

TrueMed tarvitsi sen olemassa olevaa Elastic Stack -pohjaista monitorointiratkaisua toimivamman ratkaisun. Saatavilla olevien vaihtoehtojen tutkimisen jälkeen päädyttiin edellistä uudempaan ja ohjelmistopalveluna tarjottavaan Elastic-

versioon. Päätöksessä painoivat taustalla toimeksiantajan aiemmat kokemukset ja osaaminen Elasticista sekä mahdollisuus hyödyntää avoimen lähdekoodin versiota. TrueMed oli käyttänyt ensimmäisenä ratkaisunaan itse isännöityä avoimen lähdekoodin versiota Elasticista. Tämän ratkaisun tarkoituksena oli juoksevien kulujen pitäminen mahdollisimman pieninä. Itse isännöity versio oli kuitenkin työläs järjestelmä ylläpitää, joten TrueMed alkoi selvittää korvaavaa ratkaisua.

Projektin tavoitteiksi asetettiin aiempaa helpompi ylläpito, kattavampi palvelun tilan monitorointi ja siihen liittyvät hälytykset sekä laajempi lokien keräys häiriötilanteiden selvittämisen helpottamiseksi. Tavoitteiden täyttyessä uudella ratkaisulla korvattaisiin aikaisempi Elastic-toteutus.

Opinnäytetyönä toteutetun projektin resursointi toteutettiin yksinkertaisesti. Työn toteutus oli kokonaisuudessaan opinnäytetyön tekijän vastuulla ja siihen varattiin runsaasti aikaa muiden työtehtävien ohessa vuoden 2022 ensimmäisen puoliskon aikana. Toteutetun monitorointiratkaisun soveltuvuutta toimeksiantajan käyttöön valvoivat ja arvioivat kaksi vanhempaa yrityksen työntekijää. Heillä oli myös päätäntävalta asioissa, jotka vaikuttaisivat toimeksiantajalle syntyviin kuluihin.

## 4.2 Toteutus

### Valmistelut ja suunnittelu

Elastic Cloud ympäristön luontiin ja hallintaan vaadittava Elastic Cloud -tili luotiin aluksi Standard-tilauksella. Kuten luvussa 3.2 on mainittu, se on tilauksista edullisin ja soveltuu hyvin Elasticin ohjelmistopalvelun kokeilemiseen.

### Testiympäristö

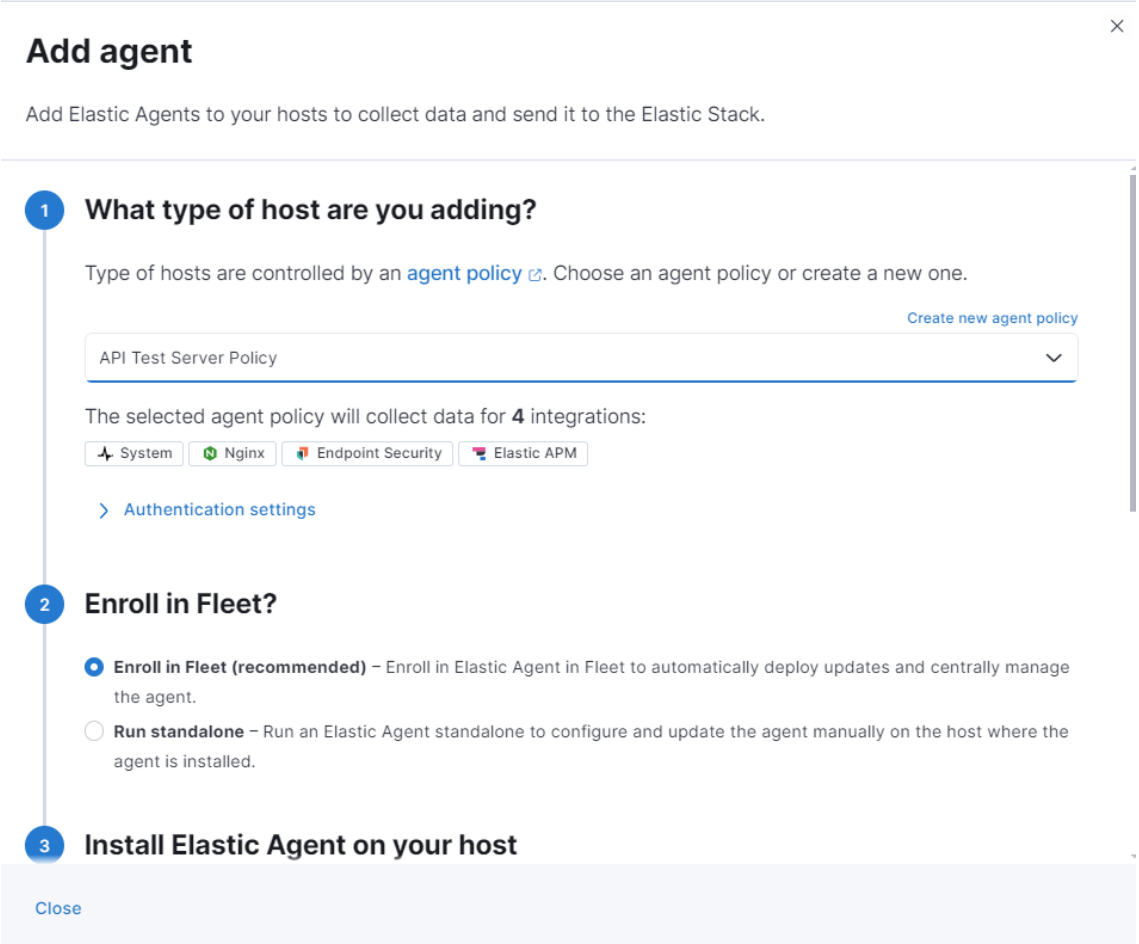
Elastic Cloud toteutusta varten luotiin testiympäristö, jossa toiminnallisuuksiin ja konfiguraatioon saattoi tutustua. Testiympäristö oli varustettu pienellä tallennustilalla yhdellä saatavuusvyöhykkeellä (availability zone). Käytetty Elasticsearch-

versio oli 8.1. Myöhemmin ympäristöön lisättiin myös Machine Learning -instanssi koneoppimisominaisuuksien testaamiseksi.

Testiympäristössä ominaisuuksia ryhdyttiin kokeilemaan sen perusteella, mitkä vaikuttivat hyödyllisiltä toimeksiantajalle keskeisien ongelmien ratkaisemiseksi. Ensimmäinen vaihe oli mahdollistaa lokisyötteen kerääminen järjestelmään. Tätä varten luotiin kopio eräästä käytössä olleesta palvelimesta, joka vastasi opinnäytetyön käyttötarkoitusta. Kopio toimitettiin Amazon Web Services -pilvipalvelusta hankitulle Elastic Compute Cloud -virtuaalipalvelimelle. Palvelimelle asennettiin Linux-pohjainen käyttöjärjestelmä, ja siellä suoritettiin Python-sovellusta.

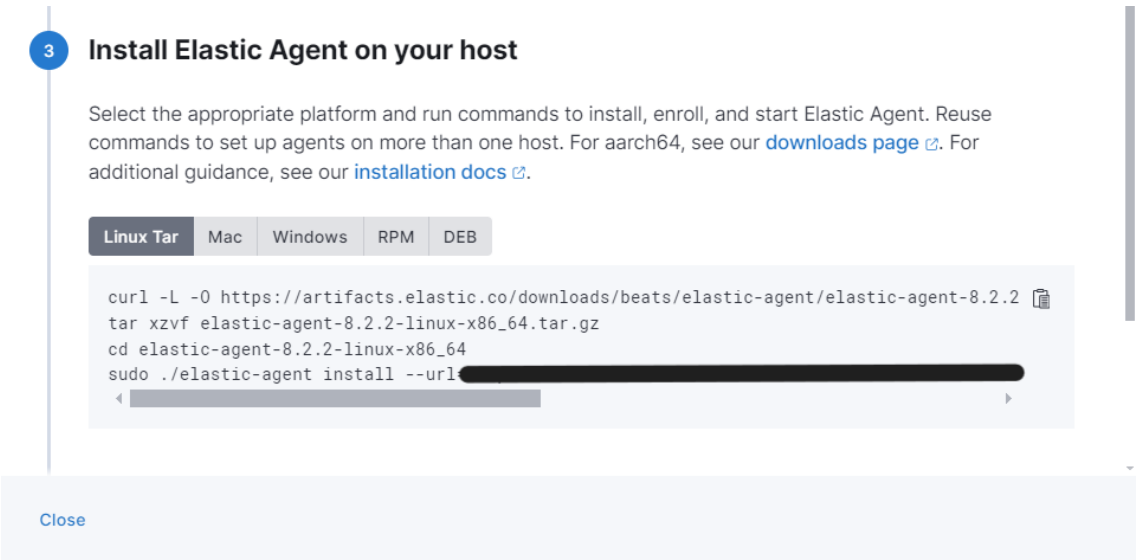
Lokisyötteen keräämisessä hyödynnettiin Elastic Agent -ohjelmistoa, joka asennettiin kohdepalvelimelle. Asentaminen oli helppoa ja yksinkertaista verrattuna aiemmin toimeksiantajan käytössä olleeseen Beats-pohjaiseen ratkaisuun.

Elastic Agentia asennettaessa oli valittava käytäntö, joka määrittää, mitä integraatioita agentti tulisi käyttämään. Tässä vaiheessa käytimme Elasticin (6) esimerkkiä mukailevaa käytäntöä. Valittavissa oli myös agentin lisäys Fleet-palvelimelle tai itse hallinnoitava agentti. Kuvasta 16 on nähtävissä tämä asennusvalikko.




Kuva 16. Kuvankaappaus Elastic agentin lisäyksestä.

Tässä vaiheessa Elastic Agent asennettiin palvelimelle ensin virheellisesti DEB-pakettina, jolloin sitä ei pysty päivittämään etänä, vaan päivittäessä vaaditaan uudelleenasennus palvelimelta käsin. Asensimme agentin myöhemmin uudestaan TAR-asennustiedostona, jolloin agentin etäpäivitykset ovat mahdollisia käyttäen Elastic Cloudin Fleet-palvelinta. Kuvassa 17 on valittuna suositeltu tapa. Käyttöliittymä antaa käyttäjälle leikepöydälle kopioitavaksi asennukseen tarvittavat komennot.

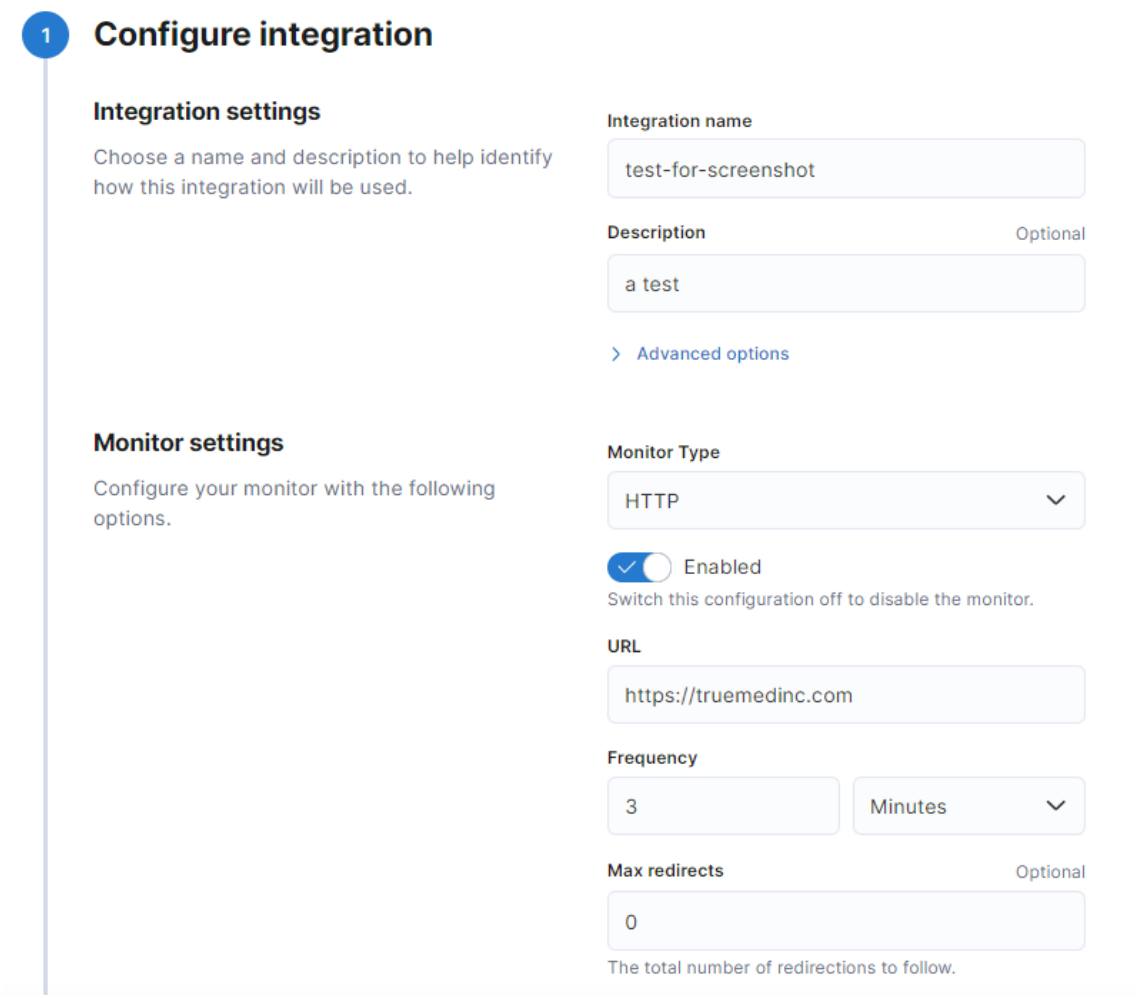



Kuva 17. Kuvankaappaus Elastic Agentin asennuspaketin valinnasta. Kuvassa tuotantovaiheen versionumero.

Palvelimella suoritettavaan Python-sovellukseen konfiguroitiin myös Elastic APM -agentti, jonka avulla Elastic Cloud -ympäristöön saatiin kerättyä tietoa palvelimella suoritettavan sovelluksen toiminnasta. Tämä APM-tieto oli tärkeässä osassa tässä projektissa käsiteltävien ominaisuuksien käytössä. Sovellukseen asennettiin ja konfiguroitiin APM Python Agentin versio 5.7.

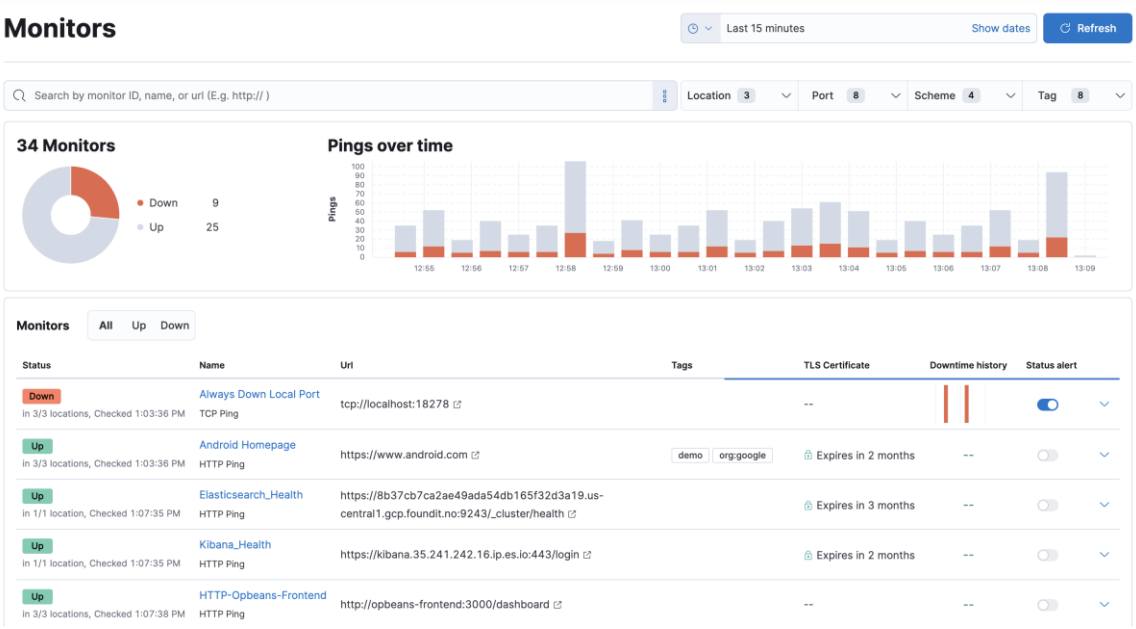
Saatavuuden monitorointi Elastic Synthetics -integraation avulla

Toimeksiantaja tarvitsi uuden tavan monitoroida palvelujemme saatavuutta, ja Elastic Synthetics vaikutti tähän sopivalta työkalulta. Päädyimme kokeilemaan sen varsin yksinkertaisesti konfiguroitavia HTTP-monitoreita. Ne tekevät määritetyin väliajoin verkkokutsun monitoroitavaan verkko-osoitteeseen ja tallentavat saadun vastauksen. Elastic Synthetics on Elastic-integraatio, eli se liitetään käytäntöön, joka puolestaan liitetään Elastic Agentiin. Hallinnan helpottamiseksi kaikki tarvittavat HTTP-monitorit luotiin samaan käytäntöön ja palvelimelle asennettiin Elastic Agent, johon tämä käytäntö oli liitetty. Kuvasta 18 on nähtävissä Elastic Synthetics HTTP -monitorin konfigurointi.




Kuva 18. Elastic Synthetics HTTP -monitorin lisäys.

Elastic Synthetics HTTP-monitorit tukevat myös helposti konfiguroitavia hälytyksiä. Jos saatu vastaus on epäonnistunut, esimerkiksi virhekoodi 500, lähettää Elastic konfiguroidulla tavalla hälytyksen. Kaikki Elastic-ympäristöön lisätyt monitorit ovat valvottavissa ja hallittavissa Kibanasta löytyvällä Uptime-sivulla. Tästä esimerkki kuvassa 19.




Kuva 19. Uptime-sivulla voidaan seurata monitorien tilaa ja hallita monitoreita (42).

## APM-virhehälytykset

Python-sovellukseen konfiguroitiin itse joitakin APM-viestejä osioihin, joissa koettiin erityisesti tarvittavan virheiden havaittavuutta. Ohjelmakoodiin lisättiin manuaalinen APM-viestin lähetys sovelluksen kohtaan, jossa tarkastimme ohjelmiston antaman vastauksen. Tiettyjen ehtojen täyttyessä lähetettiin viestin APM-lokeihin. Näin saatettiin varmasti lähettää viesti itse määritettyjen ehtojen mukaisesti. Kun tämän lisäksi Elasticissa konfiguroitiin tähän tiettyyn viestiin reagoiva hälytys, saatettiin helposti lisätä sovelluksen kriittisiin kohtiin hälytyksiä virheiden varalle. Esimerkkikoodissa 1 on havainnollistettu tätä prosessia.

```
def capture_specific_error(response):
     if response[error] == 1:
          elastic_apm.capture_message("A specific error message")
```

Esimerkkikoodi 1. Yksinkertaistettu Python-funktio, jonka avulla voidaan lisätä APM:n tapahtumalokiin tietty virhe. Elasticiin saapuvan APM-viestin perusteella voidaan myös niin haluttaessa laukaista hälytyksiä Kibanasta.

## Tavoitteiden toteutuminen ja testitulokset

Monitorointiratkaisun ylläpidossa ja hallinnoinnissa saavutettiin merkittäviä parannuksia verrattuna aiempaan ratkaisuun, jossa ei ollut esimerkiksi käytetty Elastic Agentia. Vaikka Elastic Cloud -ympäristön hallinnointi ei ollut täysin ongelmatonta tässäkään vaiheessa, oli se silti parannus aiempaan verrattuna.

Hälytykset saatavuuden muutoksissa olivat helposti konfiguroitavia, ja niistä saatiin aiempaa informatiivisempia yksilöimällä hälytysten sisältöä monitoroitavien ympäristöjen mukaan. Toimeksiantajan aiemmassa monitorointiratkaisussa oli ollut käytössä itse hallinnoitu Elastic Heartbeat -palvelin, johon verrattuna saavutettiin parannuksia niin saatavuuden monitoroinnin laadussa kuin sen ylläpitoon käytetyssä ajassa. Myös muut hälytykset koettiin hyödyllisiksi.

Lokien kattavamman keräämisen suhteen saavutettiin merkittäviä parannuksia Elastic Agentin ja Fleet-palvelimen käyttöönoton myötä. Elastic Agent asensi ja konfiguroi tarvittavat Beats-tiedonkeräysohjelmistot automaattisesti käytännön integraatioiden mukaan. Tämän myötä uuden lokitiedoston tai lokityypin monitorointiin ottaminen oli aiempaa helpompi prosessi.

Tuotantoympäristö

Testiympäristössä suoritetun testaamisen myötä projektille asetetut tavoitteet todettiin täyttyneiksi riittävällä tasolla. Tämän seurauksena päätettiin toteuttaa valitusta monitorointiratkaisusta toimeksiantajan käyttöön tuotantoversion.

Tuotantoympäristö luotiin hyvin samanlaiseksi kuin miksi testiympäristö oli tähän mennessä muodostunut. Tuotantovaiheessa siirryttiin Elastic Cloud -tilauksessa Platinum-tilaukseen täyden Elastic Support -teknisen tuen ja edistyneiden ominaisuuksien, kuten koneoppimistyökalujen, käyttöön saamiseksi. Ympäristöön konfiguroitiin Elasticsearch-yksiköt neljälle eri säilytystasolle sekä koneoppimisominaisuudet (machine learning instance). Eri tasojen Elasticsearch-yksiköt sijoitettiin kolmelle eri saatavuusvyöhykkeelle. Elasticsearch-versioksi oli tässä vaiheessa valittu 8.2, jossa pitäydyttiin pieniä päivityksiä (patch) lukuun ottamatta tuotantoympäristön käyttöönottovaiheen loppuun. Pienillä päivityksillä tarkoitetaan tässä esimerkiksi version 8.2.0 päivittämistä versioon 8.2.1.

Elastic Synthetics oli todettu testivaiheessa toimivaksi ratkaisuksi verkkosivujen ja vastaavasti verkkoon näkyvissä olevien palveluiden saatavuuden monitorointiin. Tuotantoympäristöön Elastic Synthetics -integraatiota konfiguroitaessa valvonnan piiriin lisättiin nyt kaikki tarpeelliseksi koetut palvelut. Monitoroitaville palvelimille asennettiin Elastic agentit. Agentteihin liitettiin tässä vaiheessa varsin maltillisesti tietoa keräävät käytännöt, joita tultaisiin tarpeiden mukaan päivittämään käyttäjän tiedontarvetta vastaaviksi.

Tuotantoympäristöön luotiin varoitussäännöt erilaisille virheille. Yksi toimeksiantajalle tärkeimmistä oli sääntö, jolla varoitettiin kohdepalvelinten eräässä prosessissa syntyvistä tuntemattomista virheistä. Tämä varoitus toteutettiin esimerkkikoodi 1:n tapaa mukaillen. Elasticin hälytykset mahdollistivat ongelmiin vastaamisen heti niiden ilmentyessä.

Hälytyksien personointi ja rikastaminen ECS-tietokenttiä hyödyntämällä

Kun tuotantoympäristö oli otettu käyttöön, alkoi saatavilla olevan lokisyötteen määrä olla tavoitteiden mukainen. Tästä tiedosta oli KQL-kieltä (Kibana Query Language) käyttämällä mahdollista hakea tarkempia tietoja tietyistä tapahtumista tilannekuvan kartuttamiseksi.

## 4.3 Käyttöönotossa ilmenneitä asioita

Elastic Cloud -ohjelmistopalvelun käyttöönotto on varsin helppoa ja alkuun pääsee todella nopeasti. Testiympäristön kohdalla saavutettiin tietty minimitason jo ensimmäisinä päivinä. Elastic Agent tekee uusien kohdeisäntien ottamisesta monitoroitavaksi yksinkertaista ja helposti toistettavaa. Dokumentaatiota ja ohjeita on saatavilla lähes joka aiheeseen, mutta ne ovat monesti pintapuolisia ja niissä on joskus vain yksinkertaisia esimerkkejä.

Hälytysten viestikentissä käytettävissä olevat hälytyskontekstin ECS-tietorakenteet ovat huonosti dokumentoituja ja vaativat joskus rakenteen selvittämistä manuaalisesti hälytysviestejä lähettämällä. Kibanan edistyneessä käytössä KQL:n

osaaminen edes jollain tasolla tulee välttämättömäksi, mutta Kibana tarjoaa usein myös tähän apua graafisessa käyttöliittymässään.

Indeksien, sirpaleiden ja tiedon elinkaaren hallinnointi on joskus vaikeaa ja työlästä, varsinkin käyttäjille, joilla ei ole kokemusta vastaavasta. Virallisestakin dokumentaatiosta on vain rajallisesti apua. Elastic Cloudia ei siis voida pitää ratkaisuna, joka on heti hankittaessa valmis käytettäväksi (plug and play). Tästä syystä Elastic Cloud ei välttämättä ole paras ratkaisu organisaatioille, joiden henkilöstöllä ei ole valmiuksia yllä kuvattujen ylläpitotoimien suorittamisesta tai jotka eivät ole valmiita tätä osaamista hankkimaan. Tämän työn toteuttajalle ja toimeksiantajan organisaatiolle tämä ei kuitenkaan ollut varsinainen ongelma, vaan ratkaisua on pystytty hyödyntämään järkevillä resursseilla. Ohjelmistopalvelun toteutusmallia, jossa Elastic hallinnoi myös Elasticsearch-kokonaisuuden indeksienhallinnan, ei toteutushetkellä ollut saatavilla, mutta raportin kirjoitushetkellä Elastic on sellaista tuomassa saataville.

# 5 Yhteenveto

IT-järjestelmän ylläpitäjän täytyy olla tietoinen järjestelmän tilasta. Jos järjestelmän tai palvelun suorituskyky on heikentynyt, siihen kohdistuu poikkeavaa kuormaa tai se ei ole ollenkaan saatavilla, tulee ylläpitäjän saada se tietoonsa mahdollisimman nopeasti. Automatisoidulla lokien keruulla, keskitetyllä lokienhallinnalla ja analyysilla sekä tarkoituksenmukaisilla hälytyksillä ylläpitäjä mahdollistaa poikkeamien havaitsemisen ja järjestelmän nopeamman palauttamisen.

Ohjelmistopalveluna toteutetun monitorointiratkaisun käyttö on organisaatioille yksi toimivaksi havaittu tapa aloittaa tai parantaa kattavaa toimintojensa valvontaa. Elastic Cloud -ohjelmistopalvelu mahdollistaa Elastic-ohjelmiston käyttöönoton ja käyttämisen itse isännöityä ratkaisua pienemmällä konfiguraation määrällä. Isännöimällä monitorointiratkaisun itse ja räätälöimällä konfiguraation tarkemmin olisi todennäköisesti päästy vielä parempaan lopputulokseen, mutta tämä olisi vaatinut enemmän aikaa ja osaamista sekä aiheuttanut enemmän kustannuksia. Kun monitorointiin ja keskitettyyn lokienhallintaan käytettävää

ohjelmistoa isännöi palveluntarjoaja, jää organisaatiolle enemmän resursseja omien järjestelmiensä ylläpitoon.

Toteutettu ratkaisu täyttää toimeksiantajan sille asettamat vaatimukset ja palvelee tarkoitustaan. Käytön aloittaminen oli varsin helppoa ja perustavanlaatuiset ominaisuudet olivat helposti konfiguroitavissa. Kibana on toimiva ja intuitiivinen käyttöliittymä, jonka avulla kerätyn tiedon analysointi ja hyödyntäminen onnistuu myös vähemmän kokeneelta käyttäjältä. Jotkut Elastic-ratkaisun ylläpidon ja käytön ominaisuudet eivät kuitenkaan olleet täysin ongelmattomia. Indeksien elinkaarenhallinta on välttämätön osa Elasticsearch-pohjaista ratkaisua, ja elinkaarenhallinnan toteuttaminen vaatii osaamista. Tätä ei kuitenkaan missään nimessä voi pitää syynä välttää Elasticsearch-pohjaisten ratkaisujen käyttöä, vaan vain yhtenä asiana, joka monitorointiratkaisun hankkijan tulee huomioida. Toteutettu ratkaisu on silti helpompi ylläpitää kuin toimeksiantajan edeltävä ratkaisu. Opinnäytetyössä toteutettua monitorointiratkaisua voitaneen siis pitää toimeksiantajan kannalta onnistuneena.

## Lähteet


1 Bragin, Tanya. 2019. Observability with the Elastic Stack. Verkkoaineisto. Elastic NV. <https://www.elastic.co/blog/observability-with-the-elastic-stack>. 28.2.2019. Luettu 28.03.2024.

2 Näin keräät ja käytät lokitietoja. 2023. Verkkoaineisto. Kyberturvallisuuskeskus. <https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/ohjeet-ja-oppaat/nain-keraat-ja-kaytat-lokitietoja>. 6.3.2023. Luettu 28.9.2023.

3 Lokien keräys ja käyttö. 2016. Verkkoaineisto. Viestintävirasto. <https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Lokitusohje.pdf>. Luettu 23.03.2024.

4 Get started with logs and metrics. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/observability/8.10/logs-metrics-get-started.html#add-nginx-integration>. Luettu 12.10.2023.

5 Ellingwood, Justin. 2017. An Introduction to Metrics, Monitoring, and Alerting. Verkkoaineisto. DigitalOcean. <https://www.digitalocean.com/community/tutorials/an-introduction-to-metrics-monitoring-and-alerting>. 5.12.2017. Luettu 23.03.2024.

6 Wilkins, Andrew. 2017. How to instrument your Go app with the Elastic APM Go agent. Verkkoaineisto. Elastic NV. <https://www.elastic.co/blog/how-to-instrument-your-go-app-with-the-elastic-apm-go-agent>. 15.4.2019. Luettu 27.03.2024.

7 Pilvipalveluiden tietoturva organisaatioille. Verkkoaineisto. Kyberturvallisuuskeskus. <https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/file/Pilvipalveluiden_tietoturva_organisaatioille.pdf>. Luettu 1.10.2023.

8 Application Performance Monitoring and Observability. Verkkoaineisto. Gartner. <https://www.gartner.com/reviews/market/application-performance-monitoring-and-observability>. Luettu 28.9.2023.

9 Logs app. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/infrastructure/guide/current/logs-ui-overview.html>. Luettu 12.10.2023.

10 Magic Quadrant for Application Performance Monitoring and Observability. 2023. Gartner. Ei julkinen, pyynnöstä saatavissa yrityksiltä.

11 Cisco to acquire Splunk. Verkkoaineisto. Cisco. <https://investor.cisco.com/news/news-details/2023/Cisco-to-Acquire-Splunk-to-Help-Make-Organizations-More-Secure-and-Resilient-in-an-AI-Powered-World/default.aspx>. Luettu 13.03.2024.

12 Elastic at Datadog. Verkkoaineisto. Elastic NV. <https://www.elastic.co/elasticon/conf/2017/sf/elastic-at-datadog>. Luettu 11.03.2024.

13 Elasticsearch. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html>. Luettu 30.9.2023.

14 Logstash. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/logstash/current/introduction.html>. Luettu 30.9.2023.

15 Kibana. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/kibana/7.0/introduction.html>. Luettu 30.9.2023.

16 Elastic Security. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/kibana/7.0/introduction.html>. Luettu 3.10.2023.

17 Wutzke, Michael. 2019. Elastic Stack and Elasticsearch basic tips. Verkkoaineisto. <https://www.michael-wutzke.com/elastic-stack-and-elasticsearch-basics-and-tips/>. 1.2.2019. Luettu 30.9.2023.

18 Data in: documents and indices. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/elasticsearch/reference/current/documents-indices.html>. Luettu 28.03.2024.

19 Toshev, Martin. 2022. Elasticsearch Internals. Verkkoaineisto. JVM Advent. <https://www.javaadvent.com/2022/12/elasticsearch-internals.html>. 21.12.2022. Luettu 28.03.2024.

20 ILM: Manage the index lifecycle. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/elasticsearch/reference/current/set-up-lifecycle-policy.html>. Luettu 28.03.2024.

21 Data tiers. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/elasticsearch/reference/current/data-tiers.html>. Luettu 28.03.2024.

22 Elastic Cloud. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/cloud/current/ec-getting-started.html>. Luettu 30.9.2023.

23 RESTful API. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/cloud/current/ec-restful-api.html>. Luettu 20.03.2024.

24 Pieni API-sanakirja. Verkkoaineisto. CGI. <https://www.cgi.com/fi/fi/blogi/pieni-api-sanakirja>. Luettu 20.03.2024.

25 HTTP. Verkkoaineisto. Mozilla. <https://developer.mozilla.org/en-US/docs/Web/HTTP>. Luettu 20.03.2024.

26 What is a RESTful API?. Verkkoaineisto. Amazon Web Services. <https://aws.amazon.com/what-is/restful-api/>. Luettu 20.03.2024.

27 Observability. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/kibana/current/observability.html>. Luettu 28.03.2024.

28 Elastic Cloud Pricing Calculator. Verkkoaineisto. Elastic NV. <https://cloud.elastic.co/pricing?elektra=pricing-page>. Luettu 07.03.2024.

29 Elastic Cloud subscription tiers. Verkkoaineisto. Elastic NV. <https://www.elastic.co/pricing>. Luettu 06.03.2024

30 Elastic Agent and Fleet make it easier to integrate your systems with Elastic. Verkkoaineisto. Elastic NV. <https://www.elastic.co/blog/elastic-agent-and-fleet-make-it-easier-to-integrate-your-systems-with-elastic>. Luettu 20.03.2024.

31 Elastic Agent. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/fleet/current/beats-agent-comparison.html>. Luettu 28.02.2024.

32 Application performance monitoring (APM). Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/observability/current/apm.html>. Luettu 08.04.2024.

33 What are Beats?. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html>. Luettu 23.04.2024.

34 How Ansible works. Verkkoaineisto. Red Hat. <https://www.ansible.com/overview/how-ansible-works>. Luettu 18.03.2024.

35 Elastic Agent Policy. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/fleet/current/agent-policy.html>. Luettu 06.03.2024.

36 Elastic Fleet server. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/fleet/current/fleet-overview.html>. Luettu 07.03.2024.

37 What is Jenkins?. Verkkoaineisto. Jenkins. <https://www.jenkins.io/doc/#what-is-jenkins>. Luettu 24.03.2024.

38 Pilviympäristöjen poikkeamanhallinta. Verkkoaineisto. Kyberturvallisuuskeskus.

<https://www.kyberturvallisuuskeskus.fi/sites/default/files/media/publication/Traficom%20Toimintaohje%20-%20Pilviymp%C3%A4rist%C3%B6jen%20poikkeamanhallinta.pdf>. Luettu 1.10.2023.

39 Pilvipalvelut. 2021. Verkkoaineisto. Tietoevry Oy. <https://www.tietoevry.com/fi/blogi/2021/05/pilvipalvelut-tieda-tarkeimmat-termit/>. 21.5.2021. Luettu 07.03.2024.

40 What is a virtual private server (VPS). Verkkoaineisto. IBM. <https://www.ibm.com/topics/vps>. Luettu 09.03.2024.

41 Which server you should use for virtualization. 2014. Verkkoaineisto. Snel.com <https://www.snel.com/which-server-you-should-use-for-virtualization/>. 19.12.2014. Luettu 07.04.2024.

42 Uptime and synthetic monitoring. Verkkoaineisto. Elastic NV. <https://www.elastic.co/guide/en/observability/8.2/monitor-uptime-synthetics.html>. Luettu 13.03.2024.