

## **Pilvipalvelun ketterän kehittämisen työkalut**

Pasi Aarnilähde

Haaga-Helia ammattikorkeakoulu

Amk-opinnäytetyö

2022

Tradenomin tutkinto

## Tiivistelmä

**Tekijä(t)**

Pasi Aarnilähde

**Tutkinto**

Tradenomi

**Raportin/Opinnäytetyön nimi**

Pilvipalvelun ketterän kehittämisen työkalut

**Sivu- ja liitesivumäärä**

66 + 0

Päiväkirjamuotoinen opinnäytetyö on omalta työpaikalta tehty työ, missä kuvataan ensinnäkin työn vaatimuksia, mutta myös työn haasteita sekä pohditaan oman osaamisen kehittymistä aikajaksolla. Tekijä kuvaa päivittäisiä työtehtäviään pilvipalveluiden ja ohjelmistokehityksen parissa sekä työn vaatimia ketterän kehityksen työkaluja. Opinnäytetyössä alustetaan työpaikan työympäristö sekä työtehtävät, vaadittu osaaminen, työpaikan sidosryhmät sekä työntekijän vuorovaikutustaidot työpaikalla.

Tekijä asettaa jokaiselle seurattavalle aikajakson työpäivälle työtehtävät, ja tavoitteet sekä päivän ja viikon päätteeksi pohtii tavoitteiden toteutumista sekä oman työn vaativuutta ja omaa kehittymistä. Aikajakso ajoittuu 10-viikon ajalle 14.2-24.4.2022 välisenä aikana. Työviikkojen päätteeksi tekijä pohtii omien havaintojen sekä tietoperustan pohjalta viikon kulkua ja vertailee kuinka viikon tavoitteet vastaavat oman kehityksen odotuksiin sekä asetettujen tavoitteiden toteutumiseen. Kehitysteemat perustuvat tietoperustan aiheinkin oleviin teoksiin, jotka liittyvät ketterän kehityksen kehitysprosessiin Scrumiin ja ohjelmistokehityksen työkaluihin, kuin myös työssäoppimiseen sekä työmenetelmiin. Jakson aikana analysoidaan työtehtävissä käytettyjä työmenetelmiä, omaa oppimista pilvipalveluista sekä ketterästä kehittämisestä, ja pohditaan erilaisia työtapoja haasteiden ratkaisemiseksi ja kartoitetaan mitä muita menetelmiä ratkaisuissa voitaisiin mahdollisesti käyttää.

Keskeinen tietoperusta kehitysteemojen tueksi löytyvät kolmesta teoksesta jotka liittyvät ohjelmistokehityksessä käytettäviin työkaluihin ja pääsääntöiseen pilviympäristöön minne kehitystyötä tehdään, kuin myös tekijän henkilökohtaiseen elämään ohjaavalla tietotyöläisen oppaalla joka liittyy tekijän parempaan elämänhallintaan ja parempaan keskittymiseen projektikehityksen aikana.

Lopuksi tekijä pohtii omaa kehittymistä, oppimista, sekä ammattitaitoaan ja henkilökohtaista elämäänsä, erityisesti henkisen tasapainon löytämistä ohjelmistokehityksen aikana.

**Asiasanat**

Työmenetelmät, Työssäoppiminen, Pilvipalvelut, Ketterät menetelmät

## Sisällys

1	Johdanto .....	1
1.1	Käsitteet.....	3
2	Lähtötilanteen kuvaus .....	5
2.1	Oman nykyisen työn analyysi.....	5
2.1.1	Terraform .....	8
2.1.2	Kubernetes.....	9
2.1.3	Helm pakettienhallinnan työkalu.....	10
2.1.4	Microsoft Visual Studio Code .....	10
2.2	Sidosryhmät työpaikalla .....	11
2.3	Vuorovaikutus työpaikalla .....	12
3	Päiväkirjaraportointi.....	13
3.1	Viikko 7 .....	13
3.2	Viikko 8 .....	18
3.3	Viikko 9 .....	22
3.4	Viikko 10 .....	27
3.5	Viikko 11 .....	32
3.6	Viikko 12 .....	38
3.7	Viikko 13 .....	44
3.8	Viikko 14 .....	50
3.9	Viikko 15 .....	54
3.10	Viikko 16 .....	58
4	Pohdinta ja päätelmät.....	62
	Lähteet .....	65

## 1 Johdanto

Kuvaan tässä päiväkirjamuotoisessa opinnäytetyössä työtehtäviäni ohjelmistokehittäjänä 10-viikon ajalta. Lisäksi alustan lähtötilanteen, käytettävän työympäristön ja sidosryhmät, sekä pilvipalvelussa vaaditut ketterän kehityksen työkalut ja menetelmät. Opinnäytetyö tehtiin 14.2.2022 ja 24.4.2022 välisenä aikana.

Työnantajani on Netum Oy, joka on jokin aika sitten Nasdaq Helsingissä (First North Growth Market Finland) pörssiin listautunut Tamperelainen IT-palvelutalo, jolla on yli 20-vuoden kokemus erilaisista IT-hankkeista. Netumin vahvuuksia ovat markkinoilla heidän LegacyToDigi -konsepti, joka yhdistää perinteiset tietotekniikkaratkaisut uusimpiin digitaalisiin sovelluksiin. Netumin palvelutarjoama kattaa kyberturvallisuuspalvelut, data- ja analytiikkapalvelut, digipalvelut, järjestelmien jatkuvuuspalvelut, integraatiopalvelut sekä johdon konsultointipalvelut.

Työympäristöni koostuu erilaisista sidosryhmistä, joiden kanssa pääsääntöisesti työskentelen. Sisäiset sidosryhmät muodostavat kokonaisuuden, mikä tarjoaa erilaisia palveluita niin sisäisten kuin ulkoistenkin sidosryhmien tarpeisiin. Minun sidosryhmäni tehtävänä on tarjota kehitys, tuki ja uusimmat digitaalisemmat ratkaisumallit vanhojen järjestelmien korvaajaksi.

Työkaluina pilviohjelmistokehityksen puolella käytetään pääsääntöisesti ketterän kehityksen työkaluja, jotka ovat yleensä kolmannen osapuolen työkaluja. Työkaluja eli ohjelmistoja, joilla kehitys tapahtuu, on paljon ja valikoituvat projektin sisällön perusteella mitä aletaan tekemään, ja millainen on päämäärä.

Oman työtehtäväni tietoperusta koostuu ohjelmistokehityksen alaisista teoksista kuten Terraform Cookbook joka antaa hyvän alun Terraformilla toteutettavaan pilviympäristön luontiin sekä Just Sopivasti Ketterä – ketterällä kehityksellä kohti parempaa elämää joka avaa työn sekä muun elämän merkitystä parempaan elämänhallintaan ja parempaan tietotyöhön, kuin myös Azure Security Handbook joka antaa teorian tietoturvallisen pilviympäristön luontiin, eli suoranaista ohjelmointikieltä ei ole käytössä, vaan työ tapahtuu pääsääntöisesti kolmannen osapuolen (Infrastructure as Code) eli IaC-kehittäjätyökaluilla kuten ohjelmistokehityksen puolella käytetään.

Kehitysteemat perustuvat tietoperustan alaisiin teoksiin jotka liittyvät pilvijärjestelmäkehityksen työkalujen parempaan käyttöön, sekä myös oman elämänhallinnan parempaan hallintaan.

Toimin järjestelmäasiantuntijan nimikkeellä ohjelmistokehittäjänä asiakasprojektissa, jota tulen opinnäytetyössä kutsumaan nimellä Projekti X. Nykyisessä työssäni olen työskennellyt vuoden verran. Työtehtävissäni vaaditaan laajaa ymmärrystä Microsoft Azuren pilvipalveluympäristöstä, sekä kolmannen osapuolen IaC-työkaluista.

Projektin ulkopuolista osaamista vaaditaan myös projektityöskentelystä, suunnittelusta, toteuttamisesta sekä testaamisesta.

Valitsemani tietoperusta pohjautuu ammattikirjallisuuden teoksiin kuten:

- Krief, M. 2020. Terraform Cookbook. Valitsin tämän kirjan koska se on perusteos Terraform työkalun opetteluun.
- Niemi A. & Hietaniemi J. 2020. Just sopivasti ketterä – ketterällä kehityksellä kohti parempaa elämää. Teos avaa ketterän kehityksen ajattelutapaa ja tarjoaa konkreettisia työkaluja oman työympäristön parantamiseen kuin myös elämän hallintaan projektikehityksen aikana.
- Ots, K. 2021. Azure Security Handbook. Valitsin tämän teoksen, koska se avaa parhaita käytäntöjä pilvipalveluiden tietoturvaan ja niiden hallintaan.

## 1.1 Käsitteet

### AKS-klusteri

on Azure Kubernetes Servicen tuottama klusterikokonaisuus eli rypäs, joka voidaan mieltää joukoksi tietokoneita.

### API-rajapinta

Application Programming Interface on ohjelmointirajapinta sovellusten väliseen tiedonvaihtoon.

### Assosiaatio

Kahden instanssin välinen relaatio.

### Backend

Terraformissa käytettävän projektin päätaustatiedosto mihin tallentuu tärkeät ominaisuudet projektin kehityksessä.

### Debuggaaminen

Koodista suoritettavaa virheenjäljittämistä, jolla paikallistetaan koodissa esiintyviä virheitä.

### Devops

Ohjelmistokehityksen ja toimituksen automatisointimenetelmä ketterään kehittämiseen.

### Hypervisor

Virtualisointitekniikka, joka sallii ohjelmien ajamisen virtuaalikoneissa.

### IaC-työkalu

Infrastructure as Code eli työkalu koodaamalla tarkoitetun infrastruktuurin käyttöönottamiseksi.

### Infrastruktuuri

Infra on tekninen kokoelma laitteita tai resursseja, joilla tuotetaan palveluja.

### Ketterä kehitys

Projektin työmenetelmä, kuinka asioita tehdään oikeaoppisesti.

### Konfiguraatio

Infrastruktuurin rakentamiseen tarkoitettujen resurssien toteuttamista varten oleva kooditiedosto, jossa rakennetaan resurssit pilvipalveluun.

### Muuttuja

Ominaisuus, jolle voidaan antaa eri lukuarvoja. Ohjelmoinnissa sitä pidetään tietovarastona, josta tietoa voidaan hakea ja syöttää.

### Pilvipalvelu

Internetin välityksellä toimiva palvelu, joka tarjoaa alustan virtuaalisten koneiden ja sovellusten käyttöönottamiseen.

### Scrum

Projektinhallinnan viitekehys ketterämpään projektityöskentelyyn.

## 2 Lähtötilanteen kuvaus

### 2.1 Oman nykyisen työn analyysi

Työtehtäväni pääsääntöisesti on olla osana toteuttamassa asiakasinfraa Microsoftin pilvipalveluun, kuten:

- Suunnitella, toteuttaa ja ylläpitää Microsoft Azuren päälle rakennettavia resursseja.
- Erilaisten järjestelmäkokonaisuuksien suunnittelu, toteutus ja ylläpito.
- Toimia projektityöskentelijänä osana projektia, auttaa ja konsultoida eri sidosryhmiä vaativissa järjestelmähankkeissa.

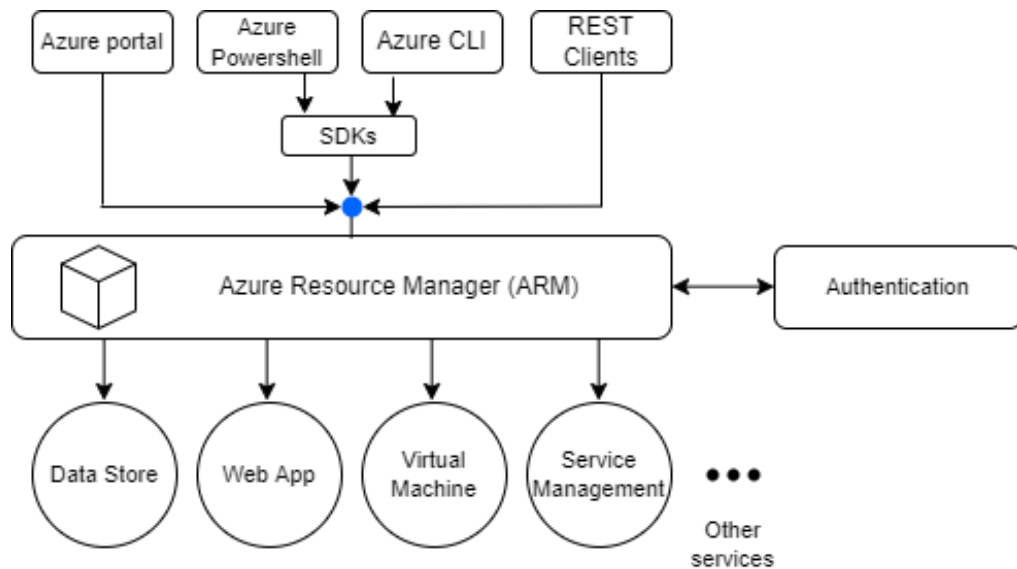
Aiempi kokemukseni on vahvasti IT-alalta palvelutuotannosta tullut kotimaan, kuin myös kansainvälisenkin IT:n palvelutuotannosta, ja on antanut vuosien aikana hyvän lähtökohdan teoreettisen tiedon, kuin myös hankitun osaamisen puolesta soveltaa omaa osaamista ohjelmistokehityksessä. Virallista ohjelmointikieleen perustuvaa syvempää kokemusta tekijällä ei ole, vaan ohjelmistokehitysuralle vaihtaminen alkaa puhtaasti IT-alalta hankittujen taitojen ja teorian pohjalta.

Työnimikkeenäni on Järjestelmäasiantuntija, ja toimin projektityöskentelijänä projektitiimissä. Projekti X toteutetaan Scrum-projektihallinnan viitekehyksen periaatteita noudattaen, joka tarkoittaa, että kyseinen tapa toteuttaa projekti on yleinen ohjelmistokehityksessä. Työn tekemiseen tarkoitettu laitteisto on työnantajalta saatu kannettava tietokone verkkoyhteydellä, näyttö, hiiri sekä näppäimistö.

Päivittäiseen työhöni kuuluu asiakasprojektin kehittäminen eteenpäin. Tarkoitus on toteuttaa Microsoft Azure pilvipalveluun laiteympäristö, minkä varaan toteutetaan tieto-, ja hallintajärjestelmä mihin lähetetään massiivisia määriä erilaista tietoa, jonka jälkeen sitä voidaan ohjata erilaisille muille palveluille esimerkiksi datan visualisointia varten. Järjestelmän alustana toimii Microsoft Azureen rakennettava laiteinfrastruktuuri, joka koostuu järjestelmän toiminnan kannalta erilaisista pilvipalveluresursseista. Nämä ovat osa ympäristöresursseja, joiden avulla rakennetaan järjestelmä itsessään Kubernetes klusterin sisälle.

Infran toteutus, ylläpito ja sen hallinta tapahtuu ketterän kehityksen IaC-työkaluilla jotka ovat ilmaisen lähdekoodin avoimia kehitystyökaluja kehittäjille. Pääsääntöisellä työkalulla koodaan tarpeelliset konfiguraatiot pilviresursseille ja palveluille ensiksi valmiiksi. Sen jälkeen ajan konfiguraatioihin perustuvat mallit Azureen rajapinnan avulla.

Esitän havainnekuvalla kyseisen ohjelmistokehityksen prosessin alusta alkaen kuinka käytännössä saadaan asennettua kehitystyökaluilla resursseja API-rajapinnan kautta pilvipalveluun. Azure Resource Manager (ARM) on käyttöönotto ja hallintapalvelu, jonka avulla voi luoda, päivittää ja poistaa Azuressa olevia resursseja. Kuvasta 1 näkee kuinka Azure Resource Manager toimii API-rajapintasovelluksena.



Kuva1. Azure Resource Manager API (mukaillen Microsoft Corporation 2022)

Käyttäjän lähettäessä pyynnön mistä tahansa kehittäjän tekstieditorista, kuten Microsoft Azuressa olevasta portaalista, PowerShellista, tai Azure CLI työkalulla menee pyyntö suoraan ARM:lle, joka vastaanottaa pyynnön sekä todentaa ja valtuuttaa sen. ARM lähettää pyynnön eteenpäin Azuren palveluun, joka suorittaa pyynnön. Kaikki pyynnot käsitellään saman API-rajapinnan kautta. ARM toimii näin ollen rajapintana toteutetuille pyynnöille tuottaa resursseja Microsoft Azure palveluun.

Kyseiset kehityseditorit voidaan asentaa myös omalle koneelle eikä näin ollen ole pakollista toteuttaa kehitystä ainoastaan pilvityökalujen avulla.

Pilvipalvelun resurssien kuten Azure Kubernetes Servicen varaan toteutamme järjestelmää, joka koostuu avoimen lähdekoodin sovelluksista kuten Kafka, sekä Elasticsearch, Logstash ja Kibana. Kyseistä kokonaisuutta voidaan kutsua nimellä ELK Stack.

Kafka on tietosäilö, joka on optimoitu lokidatan prosessointiin reaaliajassa. Tällaista dataa tuotetaan monista tietolähteistä yhtä aikaa, ja Kafka pystyy prosessoimaan suurenkin määrän dataa samanaikaisesti.

Elasticsearch on datalle tarkoitettu haku-, ja analysointisovellus, joka kykenee vastaanottamaan muunnetun raakadatan Logstashilta. Elasticsearchilla voidaan käsitellä, rikastaa, tallentaa sekä analysoida dataa.

Logstash puolestaan on sovellus, joka vastaanottaa dataa Kafkalta, muuntaa sen sopivaan muotoon sillä raakadatan monipuolisuus voi olla hankalasti analysoitavaa, ja edelleen lähettää sen haluttuihin paikkoihin kuten Elasticsearchille, tai esim. Azure Sentineliin datan hallintaa ja käsittelyä varten.

Kibana on tässä kokonaisuudessa graafinen käyttöliittymä, millä voidaan hallita, ja katsella muunnettua dataa sekä rakentaa visuaalisia näkymiä, ja raportteja.

Järjestelmäkehitys edellyttää hyvää tietoutta Microsoft Azuresta, sekä ohjelmistokehityksen työkaluista ja sovelluksista. Myös hyvä yleistieto ohjelmistokehityksestä ja sen prosesseista on eduksi tässä projektissa, sillä projekti toteutetaan Scrum-viitekehyksen periaatteita noudattaen, joten tuntemus kyseisestä viitekehyksestä on myös hyvä olla. Scrum on yksi yleisimmin kehittyvistä ketterän kehityksen työmalleista.

Goforen Niemi ja Hietaniemi mukaan se on prosessi sellaiseen kehitystyöhön, missä rakennetaan tiimin kanssa jotakin uutta. Scrum edustaa työntöohjausta, missä suunnitellaan etukäteen mitä seuraavan etapin eli sprintin aikana tehdään. Sprintille valitaan ylätasoinen visio ja käyttötapa-ukset, jotka tullaan toteuttamaan sen aikana. Sprint huipentuu sen lopussa pidettävään katselmointiin, jossa käytännössä käydään läpi, kuinka käyttötapa-ukset ympäristössä toimivat. Lopuksi pidetään retrospektiivi, missä pohditaan mikä etapin aikana meni erityisen hyvin ja missä jäi parannettavaa. Parannusideoista valitaan ne ehdotukset, joissa nähtiin eniten parannettavaa ja joita kokeillaan seuraavan sprintin aikana. Scrum prosessissa kehitystiimi muodostuu noin viidestä yhdeksään henkilöä, jonka tarkoituksena ei suinkaan ole tehdä yhtä isoa asiaa kerrallaan, vaan pienempiä asioita lyhyen aikaa. Scrum prosessin elinkaareen kuuluvat siis jatkuvat etapit eli sprintit, jotka ovat noin yhdestä neljän viikon mittaisia. Jokaisen sprintin alussa määritellään tavoitteet, ja sen päättyessä esitetään saavutettuja tavoitteita aidossa ympäristössä. Scrum sisältää kolme roolia, joista ensimmäisenä on tuoteomistaja. Tuoteomistaja vastaa tuotteen visiosta ja työjonon hallinnasta. Tällä roolilla on paras ymmärrys siitä, mikä reitti vie tiimiä nopeimmin kohti maalia. Tuoteomistajan rooliin kuuluu huolehtia siitä, että tiimi tuottaa mahdollisimman paljon lisäarvoa sidosryhmille. Toisena roolina on Scrum Master, jonka tehtävänä on auttaa tuoteomistajaa ja tiimiä siinä, että esteitä olisi mahdollisimman vähän. Scrum Master määrittää projektin etenemisvauhdin ja vastaa yleisen projektinkulun juoksevista asioista. Kolmas rooli on kehitystiimi, joka on projektin moottori. Kehitystiimi muodostuu

yleensä projektin kannalta tärkeistä asiantuntijoista, jotka vastaavat projektin kehityksestä ja sen rakentamisesta. (Niemi & Hietaniemi 2020, 39–43)

Olen saanut tutustua työni kautta tämän projektin puitteissa uusiin IaC-kehittäjätyökaluihin kuten Terraform, Kubernetes, sekä Helm. Kuvaan näitä työkaluja jäljempänä, mitä ne ovat ja mitä niillä tehdään.

Työnantajani tukee oman osaamisen kehittämistä ja antaa minun kouluttaa itseäni kuuden prosentin verran työajastani. Tämä on mahdollistanut paremman tutustumisen projektinlaisiin työkaluihin. Olen mielestäni kehittynyt paljon lähtötilanteesta työkalujen käytössä tässä kehittämistyössä. Työn tekemiseen tarvittava tietoperusta pääsääntöisesti saadaan Internetistä kehittäjätyökalujen valmistajien sivuilta, sekä kansainvälisiltä kehittäjille suunnatulta sivustolta kuten GitHub, joka on yksi tärkeimmistä kehittäjän lähteistä saada nopea apu vikaan tai ongelmaan. Käytän myös e-kirjoja lähteinä työn tekemiseen.

Tarvittava tietoperusta mitä tarvitsen työssäni, liittyy pääsääntöisesti projektissa käytettäviiin kehittäjätyökaluihin. Koodaan Terraformilla ympäristön infraa, joka ajetaan valmiilta pohjalta tuotantovaiheessa Azure DevOps-putkien avulla. Alussa kuitenkin Hiekkalaatikototeutuksessa koodit yleensä ajetaan pilveen Microsoft Visual Studio Code nimisellä ohjelmalla.

Työkalujen toimivuus kehitystyössä on tärkeää, ja sen edellytyksenä työkalujen tulee olla asennettuna koneelle ennen kehitystyön aloittamista. Työkaluja voi olla useita riippuen projektityön tarpeesta. Kuvaan muutamia työni kannalta oleellisimpia työkaluja ja niiden toimintaa mitä ne ovat, ja mitä niillä tehdään.

### **2.1.1 Terraform**

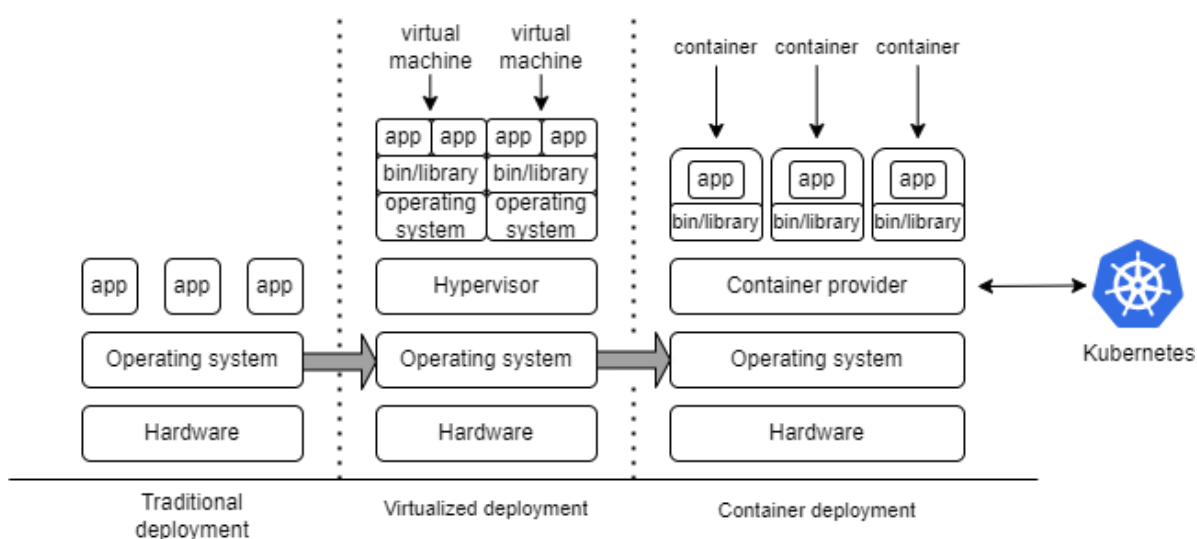
Terraformilla rakennetaan ympäristön koodisto, joka toteuttaa pilvessä erilaisia tehtäviä. Usean mallin kokoelmalla saadaan Azureen rakennettua kokonaisen asiakasyrityksen infra yhdellä Terraform ajolla. Käyttö perustuu pääsääntöisesti neljään oletuksena kuuluvaan komentoon, jotka ovat:

- `init`, joka käyttöönottaa projektin taustatiedoston ja muut työkaluihin perustuvat lisäosat.
- `plan`, joka rakentaa koodeista perustuvan suunnitelman pilveen ajoa varten.
- `apply`, joka hyväksyy ja toteuttaa muutokset pilveen.
- `destroy`, joka poistaa konfiguraatiosta halutun resurssin.

## 2.1.2 Kubernetes

IT-arkkitehti Niklas Wallenius yrityksestä Wallenius Consulting toteaa että ”Kubernetes on avoimeen lähdekoodiin perustuva järjestelmä”, jolla voidaan palvelinklusterin sisällä ajaa kontteja ja muita resursseja. (Wallenius Consulting 2022)

Itse miellän Kubernetesen avoimen lähdekoodin alustaksi, johon voidaan luoda automaattisesti sovelluksia eli mikropalveluita. Käytännössä miellän sen yhdeksi laatikoksi, jonka sisällä on virtuaalikoneita eli noodeja. Nooiden sisällä ajetaan podoja ja podien sisällä on kontteja. Kuvassa 2 erotellaan ratkaisumallit perinteisen ja nykypäivän välillä.



Kuva 2. Ratkaisumallien erot toteutuksissa (mukaillen The Linux Foundation 2022)

The Linux Foundationin mukaan perinteisessä toteutusmallissa on fyysinen laite, ja sille käyttöliittymä. Sitten käyttöliittymässä on monia sovelluksia. Sovelluksia suoritetaan yhdellä fyysisellä palvelimella, joista yksittäinen sovellus vie suurimman osan laitteen resursseista, sen seurauksena on se että muut sovellukset eivät toimi oikein tai ovat todella hitaita.

Virtualisoidussa toteutusmallissa on pääroolissa virtualisointitekniikka, jonka avulla voidaan käyttää useita virtuaalikoneita fyysisen palvelimen suorittimessa. Virtualisointi mahdollistaa resurssien paremman hyödyntämisen ja tarjoaa skaalautuvamman ympäristön joka vähentää laitteistokustannuksia. Virtualisoinnin avulla voidaan esittää joukko fyysisiä resursseja kertakäyttöisen virtuaalikoneen klusterina.

Nykypäivän konttitoteutuksessa virtualisointitekniikka poistuu, vaikka kontit ovatkin samanlaisia kuin virtuaalikoneet. Niillä on kevennetyt ominaisuudet käyttöjärjestelmän jakamiseksi sovellusten kesken. Konttitoteutuksissakin on oma tiedostojärjestelmä,

prosessori, sekä muisti. Erona muihin toteutuksiin, konttitekniologia eroaa niin että kontit ovat irrotettu fyysisestä laitteesta ja ovat siirrettävissä pilvessä ja käyttöjärjestelmäjakeiluissa. (The Linux Foundation 2022)

Työssäni konttien hallintaan käytän pääsääntöisesti kubectl-työkalua, joka on Kubernetesin hallintaan tarkoitettu työkalu. Sillä hallitaan Kubernetesin klusterin sisällä olevia resursseja. Työkalulla voidaan luoda, muokata ja poistaa erilaisia resursseja kuten esimerkiksi nodeja, serviceita, secreteja, namespaceja, ja noodeja.

### **2.1.3 Helm pakettienhallinnan työkalu**

Myös Helm on työkaluna työssäni oleellisessa roolissa. Helm on Kubernetesin hallintaan käytettävä sisäinen pakettienhallinnan työkalu, jolla voidaan käyttöönottaa erilaisia Kubernetesin paketteja eli chartteja. Ne ovat erilaisten kehittäjien tuottamia valmiiksi määriteltyjä asennettavia malleja, jolloin niitä voi hyödyntää omassa työssänsä. Mallit ladataan omalle koneelle, ja oman koneen lokaalista hakemistosta ne voidaan asentaa esimerkiksi klustereille.

### **2.1.4 Microsoft Visual Studio Code**

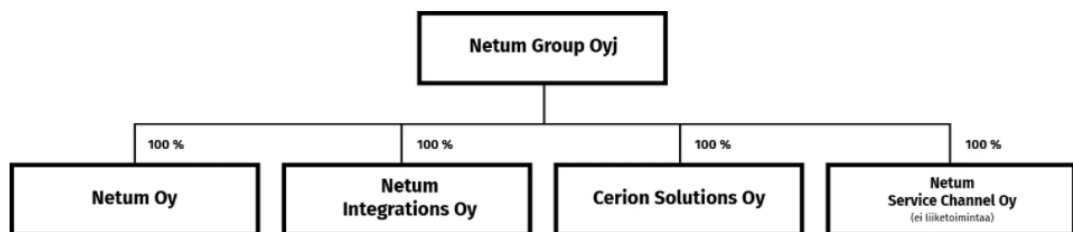
Microsoft Visual Studio Code on ohjelmistokehitykseen tarkoitettu Microsoft Corporationin valmistama ilmainen kehittäjille suunnattu sovellus ohjelmistokehitykseen. Tämä on pääsääntöinen työkaluni millä käytän kaikkia ohjelmistokehityksen mainitsemiani työkaluja. Kehitystyössä käytän editorina pääsääntöisesti Windows-koneeseeni asennetun lisäosan Windows Subsystem for Linux eli WSL editoria. WSL editori on integroitu Microsoft Visual Studioon ja sallii näin ollen käyttää kaikkia kehitystyössä tarvitsemiani työkaluja.

Oman työni tekeminen riippuu näistä mainituista työkaluista kokonaan, sillä ilman esitettyjä työkaluja ohjelmistokehitys ei toteudu, eikä niin ollen tue ammatillista kasvuani ohjelmistokehityksen urallani. Ne ovat siksi tärkeässä roolissa kehittämistyössäni. Käytän myös paljon erilaisia lähteitä päivittäisessä työssäni. Selviydyn työtehtävistä taitavana suoriutujana, ja kykenen yleensä itsenäisesti etsimään tietoa ongelmanratkaisun tueksi.

Aloittelevana ohjelmistokehittäjänä alle vuoden kokemuksella olen kehittynyt todella paljon, ja uskon olevani arvokas työntekijänä tiimilleni osana projektityön toteutusta. Ammatillisessa mielessä tavoitteeni on kehittyä hyväksi ohjelmistokehittäjäksi, joka kykenee loogisesti ja syvällä ammattitaidolla toteuttamaan erilaisia ympäristöjä sekä vaativia järjestelmäkokonaisuuksia, jolla on vankka kokemus puolellaan.

## 2.2 Sidosryhmät työpaikalla

Koska työnantaja ei anna lupaa kuvata työpaikkani sidosryhmistä julkisesti luettavaksi, voin kuitenkin kuvata konsernitasolla Netum Oy:stä. Konsernirakenteeseen kuuluvat Netum Oy, joka on yksi pääyhtiö Netum Group Oyj:n alainen tytäryhtiö, missä työskentele. Netum Group Oyj:llä on muitakin tytäryhtiöitä, jotka ovat Netum Integrations Oy, Cerion Solutions Oy, sekä Netum Service Channel Oy. Kaikki mainitut konsernin yksiköt ovat Netum Group Oyj:n omistuksessa. Kuvassa 3 näkyy Netumin konsernirakenne omistussuhteiden puolesta.



Kuva 3. Netum Group Oyj:n konsernirakenne (Kuvakaappaus sivulta <https://www.netum.fi/sijoittajille/konsernirakenne/>)

Työni koskettaa Netum Oy:n alaisia ulkoisia sidosryhmiä, kuin sisäisiäkin. Sidosryhmässä ja projektissa, missä työskentele, olen päivittäin yhteistyössä niin projektitiimin kanssa, kuin myös sisäisten ja ulkoistenkin sidosryhmien kanssa. Sidosryhmien kanssa päivittäinen yhteydenpito on tärkeätä, sillä projektityöskentelyssä projektia tehdään yleensä tiiminä, mihin voi kuulua myös ulkoisiakin jäseniä. Projektipäällikön lisäksi tiimin muodostavat varsinaiset jäsenet sisäisistä ja ulkoisista sidosryhmistä. Projektitiimi vastaa projektin toteutuksesta, sekä sen kokonaisuunnistumisesta.

Aktiivinen yhteydenpito sidosryhmien välillä takaa tiedon ajantasaisuuden, sekä projektin kannalta oleellisimman tilan missä vaiheessa projekti menee, ja millaisia haasteita sen tekemiseen liittyy. Projektityöskentelyssä sidosryhmät joko tapaavat, tai pitävät palaverin verkossa joitakin kertoja viikossa. Sidosryhmien välillä tapahtuva tiedonkulku tulee pitää ajantasaisena asiakkaan ja palveluntuottajan välillä. Sidosryhmien väliseen viestintään löytyy yhteisiä ohjelmia kuten Microsoft Teams.

### 2.3 Vuorovaikutus työpaikalla

Päivittäinen vuorovaikutus sidosryhmiin, sekä työtovereihin tapahtuu yleensä erilaisilla viestisovelluksilla kuten Microsoft Teamsilla tai Slackilla sekä sähköpostitse.

Maailmalla tapahtuneen koronapandemian seurauksena työpaikallani on suositusten mukainen järjestely etätyöskentelystä. Olen työskennellyt etänä jo pidempään, koska kehitystyön tekeminen onnistuu kodistakin hyvien internetyhteyksien päästä. Kuitenkin pidämme päivittäin projektin Scrum-palaverin, missä jokainen tiimin jäsen kertoo mitä on tehty edellisenä päivänä, ja mitä tullaan tekemään tänään. Käymme yhdessä projektin haasteet läpi, ja statuksen missä mennään.

Projektin tilanteessa missä tarvitsemme ulkoisen sidosryhmän apua projektin eteenpäin viemiseksi, yhteydenpito kontakteihin tapahtuu myös Microsoft Teams sovelluksella sekä mobiilipuhelimella ja sähköpostitse.

Kun suositukset sallivat, on toimistolle mahdollisuus mennä missä yleensä työtovereiden kanssa vuorovaikutus keskittyy lähinnä projektin haasteiden ja ongelmien puhumiseen, sekä juurisyiden etsimiseen. Myös osastopalaverissa, joka järjestetään kerran viikossa, käymme omien osastojen kollegoiden kanssa kuulumiset läpi, missä kuvataan töiden laajuus, niiden haasteet sekä vianselvitysten kuvaukset.

### 3 Päiväkirjaraportointi

#### 3.1 Viikko 7

*Maanantai 14.2.2022*

Aamu alkoi projektin tiimipalaverilla. Projektissa X on tarkoitus tänään työskennellä Terraformilla, ja rakentaa asiakasinfraa monipuolisemmaksi. Kyseisen infran konfiguraatio on muuttunut viime aikoina todella paljon, sillä olemme saaneet ulkopuolisen konsultin apua kyseisen infran rakentamisessa. Koko konfiguraatio on saanut monipuolisemman koodin ympärilleen.

Tavoitteeksi asetin saada aikaan toimiva konfiguraatio ja todeta sen tuomat palvelut eli Kafka, Logstash, Elasticsearch sekä Kibana toimiviksi asiakkaan pilvessä.

Muutaman aamun palaverin jälkeen odotin pääseväni konsultin kanssa jatkamaan työtämme, mutta konsultti perui juuri ennen palaverin alkua. Jatkoin projektia yksin, ja keskityin saamaan konsultin rakentaman koodin ajettua omalla koneellani asiakkaan pilviympäristöön.

Käytän pääsääntöisesti Microsoft Visual Studio Code sekä Terraform nimistä kehittäjäsovellusta työni tekemisessä. Microsoft Visual Studio Code on avoimen lähdekoodin tekstieditori, joka on tarkoitettu pääsääntöisesti ohjelmoijille ja ohjelmistokehitykseen. Terraform on kolmannen osapuolen tuottama työkalu resurssien tuottamiseen pilviympäristöihin kuten Azure.

Sain päivän loppupuolella onnistuneesti asiakkaan hiekkalaatikkoympäristön koodin ajettua. Ajatuksena on saman koodin pohjalta ajaa myös testiympäristö toimintakuntoon. Kuitenkin pienten muutosten jälkeen koodissa oli virhe, jonka selvitystyö jatkuu seuraavana päivänä. Kehityksen puolesta tavoite ei toteutunut täysin, sillä tarkoitus oli saada ympäristön toteutus sellaiseen tilanteeseen, että voimme ympäristöä esitellä asiakkaalle, tänään tuo tavoite ei toteutunut.

*Tiistai 15.2.2022*

Tavoitteeksi asetin jatkaa projektikehitystä eilisestä. Sain virheen korjatuksi, jonka jälkeen tavoitteena on työskennellä edelleen Terraformilla, ja ajaa konfiguraatiot testiympäristössä. Tein hieman muutoksia konfiguraatioon ja ajoin niitä testiympäristöön. Palasin tekemään hiekkalaatikkoympäristön konfiguraatiota vielä ja lähdin ajamaan toteutusta pilveen.

Konfiguraation suunnittelemisen jälkeen näytti vahvasti siltä, että kaikki resurssit ajetaan uudelleen, vaikka ne olivat jo olemassa pilviympäristössä. Konfiguraation plan, eli Terraformin ajaessa komennolla Terraform plan, työkalu tekee eräänlaisen suunnitelman ympäristöön ajettavista resursseista ennen kuin toteutettava muutos eli (apply) tehdään. Ihmetystä herätti se, että kaikki resurssit ajetaan uudelleen ja koska kyseiset resurssit olivat jo olemassa, toteutus tulisi kaatumaan.

Selvittelin asiaa, ja virhekin löytyi nopeasti. Olin tehnyt inhimillisen virheen, ja ajanut vahingossa hiekkalaatikkoympäristön konfiguraation testiympäristöön ja päinvastoin. Olin myös unohtanut vaihtaa ympäristöjen välisen backendin minkä vuoksi projektin päätaustatiedostot olivat sekaisin.

Terraformin ajaessa oleellisessa roolissa on niin sanottu backend, joka on projektin eräänlainen päätaustatiedosto, kun työkalulla tekee jotain niin muutokset tallentuvat tfstate-nimiseen päätaustatiedostoon.

Kehityin erityisesti ymmärtämään backendien merkityksen, ja niiden tärkeyden projektin kehityksen kannalta ja ennen kuin aletaan jatkamaan Terraformilla ajamista pilviympäristöön. Tässä suhteessa niin hiekkalaatikkoympäristön, kuin testiympäristön taustatiedostot olivat sekaisin. Sain korjattua päivän aikana hiekkalaatikkoympäristön taustatiedoston mutta vaati se suuren työn osakseen palauttaa tilanne lähes samankaltaiseksi.

Vaikka oma mieli olikin hieman alla päin tapahtuneen vuoksi, oli päivä todella opettavainen sen suhteen, kuinka äkkiä voidaan koko ympäristö saada sekaisin yhdellä komennolla, joka oli tänään Terraform init, mikä alustaa käyttämään työhakemistoa, joka sisältää määrittystiedostoja. Koska en käyttänyt oikeita muuttujia komennossa, se ajoi sen hetkisen määrittystiedoston ja hajotti ympäristön.

*Keskiviikko 16.2.2022*

Aamupalaverin jälkeen tavoitteeksi asetin jatkaa eilisestä ja päätin yrittää korjata testipuolen infran toimintakuntoon. Samalla pitäisi miettiä Hiekkalaatikkoympäristön klusterin jumiutunutta konttia.

Poistin testipuolen ympäristöstä kaikki resurssit pilvipalvelusta, jotka olivat vialliset. Tiimin kanssa päätettiin yhdessä, että ei ajeta vielä konfiguraatiota testipuolelle, koska se on viallinen. Elasticsearch niminen sovellus kyseisessä klusterissa ei löydä tarvittavia sertifiikaatteja, minkä vuoksi containerit ovat luomistilassa koko ajan.

Ydintiimistämme puuttuu kaksi tärkeätä jäsentä sairastapauksen vuoksi, minkä vuoksi Projekti X ei ole oikein edennyt mihinkään tällä viikolla. Tavoitteet eivät toteutuneet tämän päivän puolesta, sillä projektin kehitys oli niin vähäistä, ja ongelmat jotka esiintyivät olivat mittaluokaltaan sellaisia että oma osaamiseni ei riittänyt, ja ne vaativat asiantuntijoita jotka ovat sairaana.

*Torstai 17.2.2022*

Aamu alkoi tutusti projektialaverilla. Päivän tavoitteeksi asetin saada viimeistelyä hiekkalaatikkoympäristö kokonaan kuntoon. Kun tämän toiminta on varmistettu, voidaan sama ympäristö ajaa testipuolellekin. Konsulttimmekin on paikalla tänään, joten odotan edistystä projektin suhteen.

Aamupäivä meni henkilökohtaisten asioiden hoitamisessa osittain, mutta päivällä pääsimme tiimissä yhdessä tarkastelemaan hiekkalaatikkoympäristön toteutusta. Kollega olikin saanut korjattua varmenteeseen liittyvät virheet, joita esiintyi, mutta puutteita löytyi konfiguraatiosta vielä, jota jouduttiin korjaamaan.

Teimme konfiguraatioon sellaisen muutoksen, että verkkoympäristön aliverkot piti saada assosioitua oikein, jotta pystymme hallitsemaan liikennettä pilvipalvelun Azuren Network Security Groupilla klusterille kokonaisuudessaan.

Julkiverkon yhteys klusterille evättiin ja verkkoyhteys sallittiin jatkossa toimivaksi ainoastaan luotetuista yritysverkoista koska tarkoitus on luoda järjestelmä, joka toimii luotetuissa yritysverkoissa toistaiseksi. Tietyssä vaiheessa projektia tarkoitus on ottaa mukaan myös ulkopuolisia järjestelmiä, jolloin liikennettä sallitaan ulkopuoleltakin tuleville yhteyksille.

Yhteydet niin ulkopuolelta, kuin sisältäkin on määritelty todella tarkkaan, kenellä on oikeus käyttää järjestelmää, sekä toteutuksessa käytetään paljon erilaisia salaisuuksia, joista järjestelmän toiminta riippuu.

Tänään tunsin kehittyväni, sillä tavoitteet täyttyivät kun saimme tiimin kanssa yhdessä rakennettua paremman konfiguraation aikaiseksi. Kehityin myös verkkoliikenteen pääsynhallintasääntöjen asentamisessa.

Ohjelmistokehityksessä askeleet päivän aikana ei välttämättä ole kovinkaan suuria. Jos ongelmia on useita, yhdenkin kriittisen ongelman ratkaiseminen itsessään on jo suuri helppotus kehittäjille.

*Perjantai 18.2.2022*

Aamupalaverin jälkeen tavoitteeksi asetin saada klusterin sovelluksen eli Elasticsearchin käyttöönotettua. Sovellus on koko järjestelmän kannalta tärkeässä roolissa. Sovellus otetaan käyttöön Terraform konfiguraatioissa, ja ajetaan olemassa olevaan ympäristöön. Yritimme yhdessä tiimin kanssa saada konfiguraatiota aikaiseksi, mutta kenelläkään ei ollut sellaista vaadittavaa osaamista kyseisen sovelluksen konfiguraation rakentamisesta tai edes sovelluksen sisäisten komponenttien käyttöönotosta, että olisimme saaneet toteutuksen ajettua pilviympäristöön. Muuttujia ja riippuvuuksia on niin paljon, että osaisimme loogisesti rakentaa konfiguraation kuntoon. Tavoite ei sen puolesta tänään täyttynyt.

Tänään oli myös asiakkaan puolelta palaveri liittyen ympäristön tekniseen tietoturvaan, kuinka sitä olemme ajatellut. Esittelimme toteutuksen havainnekuvan missä laajasti kerroimme ympäristön toteutuksen alkuvaiheista aina tavoitteeseen asti. Kuvasimme asiakkaalle sitä, millaisia komponentteja ympäristön rakentamisessa on käytetty, ja mitä on otettu huomioon tietoturvamielessä.

*Viikko 7 analyysi*

Viikkoon lähdin todella odottavaisin mielin, vaikka tiimin jäsenistä suuri osa oli sairaana ja poissa. Oma työni keskittyi pitkälle Terraform konfiguraatioiden tutkiskeluun ja konfiguraatioiden kehittämiseen.

Projekti X:n kehitys edellisellä viikolla oli edennyt merkittävän askeleen kollegan avustamana, ja konfiguraatio oli kokenut uudelleensyntymisen. Konfiguraatio oli paljon monipuolisempi, loogisesti paremmin rakennettu ja siinä oli otettu monia sellaisia asioita huomioon mihin en itse usko näin lyhyellä kokemuksella pystyväni vielä. Muuttujia ja riippuvuuksia oli todella paljon, jonka avulla ympäristöstä rakentuu turvallisempi, ja monipuolisempi kokonaisuus.

Ympäristön konfiguraatio ei kuitenkaan siitä huolimatta ole vielä sellaisessa validissa tilassa, että se voitaisiin ajaa testiympäristöön. Alun perin oli tavoite saada ympäristö ajettua testipuolelle, mutta sairastapaukset pitkittivät projektin etenemistä eikä mahdollista tavoitettamme.

Itse keskityin enemmän tutkimaan, sekä opettelemaan Terraform konfiguraatioiden riippuvuuksia, sekä tutkimaan mitä ne käytännössä tekevät. Kokemukseni Terraform työkalusta on aika tuore, sillä olen työskennellyt sillä vasta jonkin verran yli puoli vuotta.

Tunsin kuitenkin kehittyväni pikkuisen sen parissa, kuinka loogisella päättelykyvyllä voidaan ohjelmistokehityksessä saada sellaisia oleellisia muutoksia aikaan, jotka parantavat ohjelmiston tietoturvaa, ja pääsynhallintaa.

Olen tuntenut kehittyväni suuresti alusta alkaen projektin aikana, kun se alkoi viime elokuussa, sillä minulla ei ollut yhtään kokemusta ohjelmistokehityksestä, eikä tavasta tuottaa asioita pilveen koodieditorilla. Terraform työkalun käyttö on mielestäni helppoa, ja se on yksi ohjelmistokehityksen työkaluista, jolla tuotetaan resursseja pilvipalveluun.

Työ, jota teen on todella mielekästä. Erityisesti kun saan oppia asioita kokeneemmilta, sekä opetella itse käyttämään erilaisia työkaluja tuottaakseni asiakasprojekteissa pilvi-infraa, voin kuvitella tekeväni ohjelmistokehityksessä työtä vielä vuosia. Olen vasta urani alkutaipaleella mutta toivon kehittyväni hyvää vauhtia kovaksi tekijäksi.

Kun aloitimme projektin, tekijöiksi tiimissä muodostui toisenlaisia osajia. Meillä ei ollut projektin työkaluista paljoakaan osaamista. Kuitenkin asioiden opettelu on ollut mielekästä, ja antoisaa. Se on mahdollistanut projektin aloittamisen, sekä antanut käsityksen kokonaiskuvasta millainen sen tulee olla.

Nyt olemme hyvässä vaiheessa projektin kanssa, ja olen itse tyytyväinen työmenetelmiin. Viisaampana tietenkin jälkeenpäin ajattelee, että tiimissä olisi hyvä ollut olla erilaista osaamista monesta osa-alueesta, sillä erilaisesta sovellusosaamisesta olisi ollut apua.

Pohdin projektin muuttumista, ja sen laadukkuutta. Selaillessani teosta Terraform Cookbook ja erityisesti oikeaoppisen tiedostopolun rakentamista lähtiessä toteuttamaan ohjelmistoprojektia, huomasin paljon yhteneväisyyttä oikeaoppisen hakemistorakenteen luonnissa ohjelmistokehityksessä.

Teoksen mukaan pääsääntö on että kolmella konfiguraatitiedostolla saadaan jo toimiva ympäristö aikaiseksi. Pää tiedosto, johon ajettavat resurssit määritellään on nimeltään main.tf. Resurssien määrittelyyn tarvittavat muuttujat määritellään tiedostoon variables.tf, sekä terraform.tfvars nimiseen tiedostoon. (Krief 2020, 50)

Alun perin projektimme hakemistorakenne ei ollut ympäristökohtaisesti looginen, vaan pikemminkin sekava ja yhden pääkansion alla oli kaikki konfiguraatitiedostot. Projektin muuttumisen, ja erityisesti kollegan tultua mukaan on hakemistorakenne muuttunut paremmaksi, ymmärrettävämmäksi ja projektin kehityksen kannalta loogiseksi. Tämä kertoo sen, että olemme oikealla tiellä projektikehityksessä. Loogisesti oikeaoppisen

hakemistorakenteen kannalta kehitystä on parempi jatkaa, kun voidaan hakemistojen perusteella jo arvioida mitä tietoa löytyy mistäkin.

### 3.2 Viikko 8

*Maanantai 21.02.2022*

Päivän tavoitteeksi asetin valmistautua asiakaspalaveriin ja siihen mitä kerron projektin tilasta asiakkaalle. Myös projektin ylimääräisten koodien putsaaminen on tavoite saada tehtyä tälle päivälle. Aamulla sain projektipäälliköltä viestin että hän on sairastunut ja hän pyysi minua paikkaamaan palaverissa. Projektitiimimme on tällä hetkellä todella pieni, sillä meitä on kaksi henkilöä työkykyisenä tekemässä projektia. Tavoitteena on myös suorittaa työnantajan tietoturvakoulutus verkossa sekä harjoitella itsenäisesti lisää Terraformin käyttöä.

Pidin asiakaspalaverin, missä kuvasin projektin tilanteen, haasteet ja tavoiteaikataulut saada hiekkalaatikkoympäristö toimintavalmiiksi. Päivän aikana kartoitin kehityspuolen konfiguraatioita, ja ajaessani suunnitelman, oli yksi uusi lisäys tulossa. En ajanut toteutusta pilveen, sillä varmuutta ei ole onko kollega tehnyt muutoksia vai miksi Terraform yrittää ajaa olemassa olevaa resurssia taas pilviympäristöön. Konfiguraatio on monimutkainen, ja yksikin ylimääräinen resurssin ajo, joka voisi korvata esimerkiksi olemassa olevan resurssin, saattaa lisätä virheiden määrää.

Viime viikon virheen vuoksi totesin, että en tee muutoksia, joista en ole varma mitä tapahtuu, joten päätin odottaa kollegaa ja katsoa tilanteen hänen kanssaan yhdessä. Sain varattua kollegalta konsultointiaikaa huomiseksi jo, joten odotan että saamme konfiguraation sellaiseen kuntoon, että projekti edistyy.

Suoritin työnantajan tietoturvakoulutuksen, sekä osallistuin oman osastoni tiimipalaveriin ja koko yrityksen viikkopalaveriin.

Päivän tavoitteet toteutuivat ja kehityin hieman projektipäällikön roolissa, sekä projekti X:n konfiguraatioiden putsamisessakin.

*Tiistai 22.02.2022*

Tavoitteeksi asetetaan jatkaa tutkimista ja kehitystä sekä saada Hiekkalaatikkoympäristön toimintaan. Aion myös pitää projektitiimin aamupalaverin tänään. Saamme myös kollegan

tänään avuksi kehittämään projektia kanssamme, joten odotukset ovat korkealla, että saisimme projektin kehitystä vietyä eteenpäin.

Toteutimme tiimin kanssa yhdessä onnistuneesti konfiguraation, missä Terraform hakee Service prinipaalit nyt automaattisesti pilvipalvelun avainsäilöstä eli Keyvaultista. Keyvault on pilvessä oleva asennettu resurssi mihin voidaan tallentaa erilaisia salaisuuksia, avaimia ja sertifikaatteja. Service principal on identiteetti, joka on tarkoitettu sovellusten, palvelujen ja resurssien käyttöä varten.

Tätä ennen objektin salaiset avaimet olivat omassa muuttujat-tiedostossa mistä Terraform luki ne. Oleellista ei ole, että käyttäjä tarvitsee niitä millään muotoa, vaan se että Terraform osaa hakea kyseiset arvot säilöstä automaattisesti ja huomioi ne ympäristössä.

Koska Terraform resurssit käyttävät palvelun klusterin Service principal avaimia, ovat ne salaisuuksia, joita ei haluta muiden näkevän. Salaisuuksien hallinta tapahtuu Keyvaultissa.

Asetetut tavoitteet saavutettiin. Itse kehityin tänään Keyvaultin ja erityisesti sen avaimien määrittelyissä. Myös kollega oli korjannut sertifikaatteihin liittyvän vian, sekä tänään saimme kehitettyä yhdessä myös projektia eteenpäin nimenomaan sertifikaattien puolesta, mikä oli tärkeää saada tänään ratkaistuksi. Kuitenkin vielä on joitain puutteita konfiguraatiossa, ja kehitystä jatketaan.

*Keskiviikko 23.02.2022*

Tavoitteeksi asetan tänään pitää vieläkin projektitiimin aamupalaverin. Eilen saatiin kehitystä eteenpäin, ja tavoitteeksi tänään asetetaan tiimin kanssa Logstash sovelluksen asentaminen hiekkalaatikkoympäristöön, joka tutkii järjestelmädataa ja välittää sitä eteenpäin muille sovelluksille.

Toinen kollegani tutkii myös ongelmaa, missä pitäisi järjestelmässä olevalle Kafkalle saada lokia menemään. Tällä hetkellä se ei saa dataa ollenkaan, vaikka pitäisi.

Jos aikaa jää, yritän tutkia myös Terraform työkaluni vikaa missä se ajon aikana antaa tulokseksi edelleen yhden resurssin asentamista, sekä muutaman salaisuuden päivittämistä vaikka kyseiset resurssit ovat jo aiemmin ajettu pilvipalveluun. Kollegan mukaan tuota tilannetta ei pitäisi syntyä, eli jotain ongelmaa vielä on.

Ongelmaksi muodostuu työkalulla se, että ajaessani muutoksia pilvipalveluun, antaa työkalu tulokseksi että ne ajetaan uusina resursseina pilveen. Koska ne ovat jo aiemmin ajettu sinne, en voinut ajaa ollenkaan konfiguraatiota tämän vuoksi ympäristöön, syystä että salaisuuksien päivitystä ei voida ajaa niiden sekaantumisen vuoksi.

Kollega oli saanut lisäkomponentin tänään asennettua, ja pääsemme jatkamaan huomenna hänen kanssaan kehitystä eteenpäin.

Tänään myös toisen kollegan kanssa selvittelimme Logstash asentamista. Jatkoimme konfiguraation määrittelyä, mutta koska emme voinut ajaa lisäystä ympäristöön, tullaan se tekemään huomenna.

Kehitys tältä päivältä oli vähäistä. Oma kehittyminen kasvoi sen suhteen, että aloin ymmärtämään Terraform lokaalien ja ympäristömuuttujien konfiguraatioita enemmän. Silti kyseinen ympäristö on nyt Terraform konfiguraatioiden puolesta paljon monimutkaisempi, ja siinä on paljon osia, joita joudun opiskelemaan ymmärtääkseni paremmin sen, miten asiat siinä toimivat.

*Torstai 24.02.2022*

Tänään saimme kollegan sairauslomalta takaisin tiimiin. Tavoitteeksi asetan tänään jatkaa eilisestä eli yrittää Terraform työkaluni korjausta, sitä miksi se yrittää päivittää vieläkin projektin salaisuuksia. Myös lisäkomponenttien asennus tämän päivän aikana olisi hyvä saada. Lisäkomponentteja on kolme, jotka pitäisi asentaa, mutta yhdenkin saaminen tänään on askel eteenpäin.

Terraform perustuu siihen, että ajan tasalla olevat konfiguraatiot otetaan hakemistosta Visual Studio Codeen ja näin ollen voidaan muokata sitä. Omat muutokset konfiguraatioon hyväksytään ja ladataan kyseiseen hakemistoon takaisin, jolloin aina muutosten jälkeinen konfiguraatio on saatavilla kaikille.

Testasimme plania kahden koneen välillä, eli minun ja kollegani. Koska projektin backend eli päätaustatiedostoja on vain yksi, niin resurssien ajo käyttää tuota päätaustatiedostoa. Se on käytännössä kaikille yhtenäinen, ja kun käyttäjä ajaa komennon plan omalta lokaalilta koneeltaan, voi lokaalissa koneessa olla jotain erilaista mikä sekoittaa sen eikä näin ollen tunnista jo ajettuja resursseja. Sen vuoksi työkaluni ehdotti resurssien uudeleenaamista pilveen. Kuitenkin tällaisessa tilanteessa ajossa tulee muutosehdotukset ja sitä ei pitäisi tulla. Mystinen vika, emmekä saaneet tuota kollegani kanssa poistettua. Ajoin applyn kuitenkin, eikä stateen tullut ainakaan muutoksia secretien uusista avaimista

eli avaimet olivat ennallaan, joten vaikka plan ehdottikin uusien salaisuuksien asentamista, ei se sitä kuitenkaan tehnyt eli nyt menee taas plan läpi. Kehitystä tapahtui tänään Terraform työkalun käytöstä koko päivän ajalta, sillä työkalussa esiintynyt ongelma vei aikaa paljon.

*Perjantai 25.02.2022*

Tavoitteena tänään on jatkaa iltapäivän sessiossa eilistä. Tiimiin on palannut yksi tekijä, joka on myös asiantuntevana mukana iltapäivällä, joten odotukseni ovat sen mukaiset, että saisimme tänään edistettyä projektia.

Päivän kehityspalaveri tiimin kanssa jatkui Terraform työkalun ihmettelemisellä sekä ongelman ratkaisulla. Ongelmaa ei saatu korjattua, mutta kyseinen ongelma ei kuitenkaan ole este projektin eteenpäin viemiseksi. Kävimme läpi seuraavien askelten eteenpäin viemiseksi mitä tullaan toteuttamaan. Lähdetään peruskonfiguraatiolla liikkeelle lisäpalvelun asentamiseksi, ja muokataan siitä Terraformille sopiva. Komponentin peruskonfiguraatio saatiin asennettua mutta toimintaa ei päästy vielä testaamaan.

*Viikko 8 analyysi*

Viikko on ollut kehityksen puolesta vähäistä, sillä tiimin sisäisiä sairastumisia on todella paljon, ja täysipäiväisiä kehittäjiä on lisäksi ainoastaan yksi koko viikolle, ja osa-aikaisia kehittäjiä kaksi. Myös kollegamme mitä olemme käyttäneet konsulttina ei ole ollut saatavissa projektille niin usein kuin ehkä olisimme toivoneet.

Olen kehittynyt eniten siinä tällä viikolla, että projektityöskentelyssä ei voi pitää itsestään selvyytenä kehityksen kulkua, sillä kehitys riippuu tekijöiden määrästä, sekä osaamisesta. Kun osaamista ei ole, projektin kehitys hidastuu huomattavan paljon kuten tällä viikolla kävi.

Viikon sairastapaukset ottivat tiimistä työtehoa huomattavan paljon pois kun projektipäällikkö sairastui myös. Tiimiltä puuttui johtaja, joten otin roolia johtajuudessa ja paikkasin projektipäällikköä.

Erkka-Pekka Sjöblomin pro-gradussa ulkoisen johtajan vaikutus luovuuteen ja innovatiivisuuden jaetun johtajuuden tiimissä olevan teorian mukaan jaetun johtamisen tiimin johtaminen eroaa perinteisestä johtamisesta siten, että hierarkkisessa tiimissä vaikuttaminen tapahtuu ylhäältä alaspäin ja jaetun johtamisen tiimissä alhaalta ylöspäin. Ulkoinen johtaja

ei suoranaisesti osallistu päivittäiseen tekemiseen, vaan konsultoi, kysyy kysymyksiä, tarjoaa informaatiota ja antaa tiimille neuvoja. (Sjöblom 2021, 16–17)

Lukiessani teosta tunsin olevani vastuussa jaetusta johtamisesta tänä aikana, ja otin paikkaavan roolin osaksi kehitystyötäni. Pyrin päivittäin korostamaan tiimille projektihallinnan alaisten töiden merkitystä, ja käymään ongelmia heidän kanssaan yhdessä läpi. Konsultoin ja kannustin heitä sekä neuvoin tarvittaessa. Kehityin myös tässä roolissa, sillä projektipäällikön tehtävät olivat aikaa vieviä oman päätyön ohella palaverien, erilaisten raportointien ja viestinnän puolesta.

Jälkiviisaana voisi todeta, että projektille olisi ollut saatavissa lisätekiäjiä sairastumisia paikkaamaan, mutta jos osaajia ei ole niin ei se kehitys etene lisätekiäjiilläkään sen enempää. Osaaminen, tieto ja taito ovat pääroolissa ohjelmistokehityksen puolella sillä looginen päättelykyky arvioida asioiden toimivuutta, sekä jäsentää opittua tietoa esimerkiksi ongelmanratkaisussa on arvokas taito, ja sellaista ei kaikilla ole. Erilaisten palvelujen riippuvuudet toisista palveluista, pääsynhallinta, tietoturva, ja järjestelmän sisäisten palveluiden toiminta vaatii todella syvällistä tietoutta ja taitoa ymmärtää tällaisia kokonaisuuksia.

Itse olen urani alkuvaiheilla matkallani, eikä minulle vielä ole kehittynyt niin syvällistä kokemusta ohjelmistokehityksestä, että taitavasti kykenen ratkaisemaan erilaisten järjestelmien vikoja, sekä loogisesti päättämään mitä jokin palvelun asennus tarvitsee toimiakseen järjestelmäkehityksen edellyttämällä tavalla. Enemmänkin tunnen olevani hyvässä vauhdissa kohti ammattitaitoa, jota toivon saavuttavani, ja osaamista erilaisten järjestelmien toteutuksesta, että pystyn loogisesti rakentamaan erilaisia järjestelmäkokonaisuuksia. Olen ajatellut työni ohessa suorittavani myös joitain ohjelmistokehityksen valmiskursseja, tukemaan omaa osaamistani ja jo opittua, sekä myös antamaan minulle laajemman ammattitaidon työkalupakkiini.

### **3.3 Viikko 9**

*Maanantai 28.2.2022*

Sairaana.

*Tiistai 1.3.2022*

Sairaana.

### *Keskiviikko 2.3.2022*

Tavoitteena oli palata iltapäiväksi työhön tänään koska projekti tarvitsee minua ainoana tiimin Terraform osaajana. Olen seurailut projektin kehitystä sairaana ollessani sivusta ja todennut että panostani kaivataan. Olin kuitenkin lyhyemmän päivän töissä, sillä oma vointi oli sellainen, että päätin jatkaa työntekoa vasta iltapäivällä.

Kollega oli tehnyt aika paljon parin päivän aikana lisää, ja projekti on edennyt hienosti koska muut poissaolijat olivat nyt palanneet jo takaisin.

Meillä on nyt hiekkalaatikkototeutus, sekä testiympäristön toteutus aika pitkälle toimintavalmiina, ja konfiguraatiot järjestyksessä. Kehitysprojektin järjestelmätoteutus on sellainen, että kyseinen infra asennetaan Terraformilla kokonaisuudessaan kuntoon. Samalla asentuu myös itse pääjärjestelmä, sillä järjestelmän osien käyttöönotto on huomioitu myös Terraformissa. Tarkoituksena on ollut tehdä sellainen kokonaistoteutus, jota hallitaan Terraformilla. Käytännössä järjestelmän ylläpito tullaan tekemään Terraformilla, ja kaikki muutokset tehdään sen konfiguraatitiedostoihin. Terraformilla ei ajeta jatkossa komentoja lainkaan omalla paikallisella koneella, vaan toteutus tehdään Microsoft Devops kehitysalustalla niin sanottujen putkien avulla.

Avaan hieman mainittua kehitysalustaa, joka on kehittäjille tarkoitettu Microsoftin rakentama kehitysalusta ohjelmistokehitykseen. Microsoft Visual Studio Code ohjelmalla koodatut konfiguraatiot tallennetaan Azureen omaan kirjastoon kyseisessä Devops palvelussa. Kehitysympäristön kehitys taas tapahtuu niin sanotulla putkella eli pipeline, jossa putken ajaessa konfiguraatioihin perustuvat muutokset ajautuvat pilveen. Konfiguraatiot putkessa haetaan kirjastosta mihin ne on ovat tallennettu. Tänään kehityin nimenomaan Devops putkien määrittelyssä. Ajossa ei periydy niinkään käyttöoikeudet, ja ympäristön kehittäminen on pikkuisen helpompaa. Putkilla toteutetaan myös muutostenhallinta, sillä voidaan myös varmistaa erinäisellä hyväksynnällä muutosten ajaminen pilveen.

### *Torstai 3.3.2022*

Asetan tänään tavoitteeksi kehitysympäristön putkien testaamisen, sekä palveluiden salaisuuksien rakentamisen testiympäristölle. Tarkoitus on ajaa kaikki kehitysympäristöt putkien avulla Microsoft Azureen.

Testaaminen onnistui, sillä hiekkalaatikkoympäristön build menee hyvin läpi ja Terraformissa olevat asennukset onnistuvat, eikä muutoksia tapahdu. Testipuolen ajossa ei

vielä voida käyttöönottaa resursseja, sillä kehityksestä puuttuu vielä mainitsemani salaisuuksien luominen Azureen.

Tunsin oikeastaan tänään kehittyväni pikkuisen. Yritimme ajaa testipuolen salaisuuksia Azureen, mutta sertifikaattien purkamisessa oli ongelmia, jonka vuoksi teemme huomenna tuon osuuden tiimin kanssa yhdessä.

*Perjantai 4.3.2022*

Tänään tavoitteena on ajaa varmenteet sekä erilaiset salaisuudet turvallisesti Azuren avainsäilöön. Myös jatketaan edelleen järjestelmän Elasticsearch konfiguraation määrittelyä, sekä sen käyttöönottoa. Iltapäivän aikana on tarkoitus osallistua kyseisen palvelun konsultaatiopalaveriin.

Tavoitteet täyttyivät tämän päivän osalta, sillä sain asennettua varmenteet avainsäilöön. Asensimme tiimin kanssa myös Elasticsearch sovelluksen hiekkalaatikkoympäristöön Terraformissa peruskonfiguraatiolla. Päivän saldoon olen tyytyväinen sillä aina kun projekti etenee edes tätä vauhtia, saa olla tyytyväinen.

On tullut selkeästi esille eri komponenttien välisen osaamisen puutteita, ja se näkyy kehityksessäkin. Tänään kuitenkin sain tiimin tuekseni kasaan ja työstimme työpajassa yhdessä eri osa-alueita. Tiimin tuki, ja yhdessä tekeminen korostavat omaa varmuutta myös tekemiseen, minkä vuoksi pidän yhdessä työskentelyä hyvin tärkeänä osa-alueena sellaisessa vaiheessa missä selkeästi on ongelmia.

*Viikko 9 analyysi*

Viikko alkoi sairastamisella ja näin ollen oma kehittymiseni jäi vajaaksi reilun kahden päivän ajalta. Tuona aikana kollegani olivat saaneet Projekti X:n lokilähteiden konfiguroinnit tehtyä, sekä saanut moniympäristön putket rakennettua kehitysympäristöjä varten valmiiksi. Myös ensimmäisen asiakasjärjestelmän lähdeosoitteiden konfiguroinnit järjestelmää varten olivat tehty. Palasin viikon puolen välin kohdalla kuitenkin töihin koska en halunnut menettää enempää ja tunsin vointini hyväksi. Projektissa ollaan pikkuisen myöhässä sillä osajia ei oikein tuntunut olevan kovinkaan montaa.

Olemme projektitiimissä tehneet sellaisen huomion, että alun alkaen kehitystä lähdettiin toteuttamaan väärällä tavalla. Projektitiimimme oli aloittaessa uusi, eikä kokemusta vastaavanlaisten toteutuksien puolesta ollut kenelläkään.

Jälkeenpäin olen ymmärtänyt mitä projektin aloitus vaatii jatkossa. Uudelta kokemattomalta tiimiltä vie aikaa opetella työkalut, ja looginen ajattelu järjestelmän rakentamiseksi ei ollut osaamisen puolesta niin syvä kuin sen olisi pitänyt olla.

Uusina kehittäjinä lähdimme tietenkin tekemään kaiken alusta, vailla kokemusta kuinka toteutus pääsääntöisesti tehdään. Työkalut olivat kaikille uusia, ja jo niiden opettelu vei aikaa todella paljon. Kun olimme saavuttaneet projektin puolenvälin, olimme ymmärtäneet mitä kaikkea tekeminen vaatii ja korjailleet todella paljon asioita.

Projektitiimillämme olisi ollut saatavissa tukea ja kokemusta tekemiseen alusta alkaen, mutta käytimme sitä todella vähän, ja luotimme omaan osaamiseemme. Nyt asia tuntuu olevan toisin. Joudumme usein pyytämään kollegan apua konsultoimaan päivittäiseen tekemiseen, koska tiimin jäsenillä ei ole juuri järjestelmän sisäisistä komponenteista ja niiden konfiguroimisesta ollenkaan kokemusta.

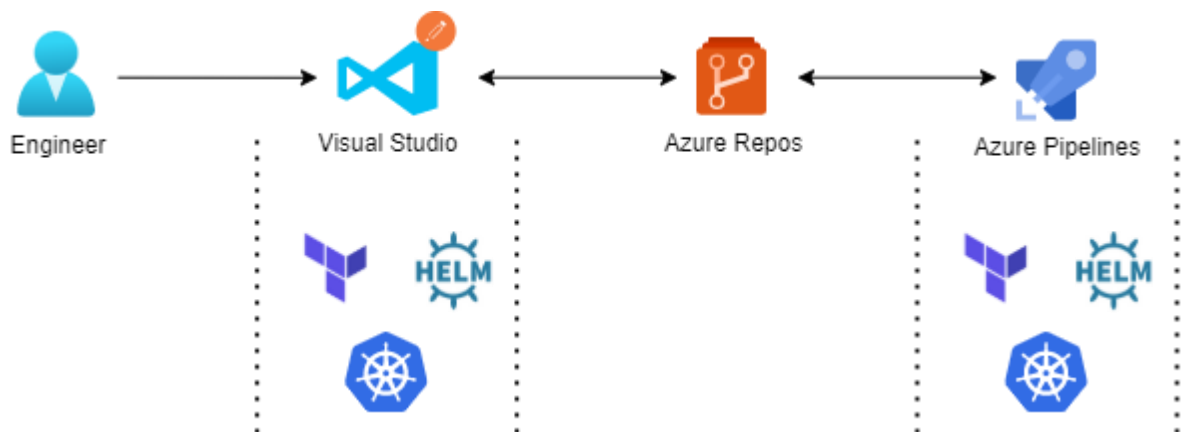
Rakensimme minimitoteutuksen järjestelmästä, mutta nyt kun on tullut mukaan aika paljon erilaisia osatekijöitä, on jouduttu miettimään riippuvuuksia, tietoturvaa, pääsynhallintaa ja koko järjestelmäkehityksen loogisuutta ylipäätään on kehitys ottanut niin suuren askeleen, että olemme ymmärtäneet, kuinka vaativa tämän kaltainen toteutus ylipäätään on.

Edelleen on komponentteja, mitkä eivät ole tuttuja millään tapaa ja niiden ymmärtäminen näin ollen on vajaata, se hidastaa myös kehitystä huomattavan paljon. Järjestelmän kaikkien palveluiden toiminnan ymmärtäminen on hankalaa, ja tuntuu juuri nyt sille, että se veisi vuosien harjoittelun, että ymmärtäisi sen toiminnan ja loogisuuden mutta aika on joka projektilla, niin myös tässäkin.

Itselleni tämä on elämäni ensimmäinen järjestelmäkehityksen projekti, ja ymmärsin kyllä jo aiemminkin, että lisäapua olisi tiimiin tarvittu tekemään. Vaikka mittaluokassaan tämä ei suurimpia projekteja ole, on ollut kuitenkin niin paljon tekemistä, että olisi ollut hyvä saada enemmänkin työntekijöitä mukaan tueksi. On kuitenkin niin monia osatekijöitä missä tarvitsemme apua, ja tekijöitä että projekti on jäänyt hieman jälkeen. Tärkein oppi minulle on se, että kehityksessä mukana olevat henkilöt ymmärtävät työkalut, ja kyllä sen kokemuksen puutteenkin on huomannut tässä projektissa. Kokemus vastaavasta on eduksi, kun on selkeä näkemys siitä, millainen toteutuksen tulee olla.

Pohtiessani kehityksen kannalta työkalujen käyttöä, ja niiden lopputulemaa, miellän itse viikolla käyttämiäni työkaluja Terraformia ja Azure DevOpsia samankaltaiseksi.

Teoksessa Terraform Cookbook, kerrotaan CI / CD-putkien käytöstä, ja pilviressurssien rakentamisesta putkien avulla, missä tullaan kuitenkin samaan lopputulokseen käytännössä suhteessa Terraform työkaluun. Esimerkiksi Azure Devops-sovelluksessa otetaan putkelle käyttöön virtuaalikone, ja määritellään tarpeelliset työkalut ja lisäosat resurssien pilveen ajamista varten. Kun putki ajetaan, se osaa hakea Azuren hakemistosta oikeat pohjat ja rakentaa pilviympäristön resurssit pohjien koodiin perustuen. Tällainen automaatio ohjelmistokehityksessä on valtava harppaus, kun voidaan ajaa yhdellä tai useammalla putkella ympäristöjä pilveen. (Krief 2020, 156–158). Kuvassa 4 esitetään projektitiimini kehitysprosessi käytännössä.



Kuva 4. Kehitysprosessin kulku

Käytännössä siis kehittäjä vasemmalla ottaa käyttöön Visual Studion koodieditorin. Visual Studioon on asennettu kehitystä varten tarpeelliset työkalut kuten Terraform, Kubernetes ja Helm. Kehittäjä vasemmalla rakentaa koodiin perustuen kyseisillä työkaluilla Visual Studiossa valmiit pohjat, joiden avulla voidaan ajaa pilveen resursseja. Koodeja säilytetään Azure kirjastossa eli tätä varten tarkoitettu hakemistossa ja haetaan Visual Studiolle aina muokkausta varten. Myös Azure Pipelines eli putken määrittely tehdään niin, että se määritellään ottamaan ensiksi kaikki tarpeelliset työkalut käyttöönsä kuten Visual Studiosakin on. Sen jälkeen kun putkea ajetaan eteenpäin, se osaa hakea Azuren hakemistosta samat pohjat ja rakentaa pilviressurssit sen avulla.

Terraform sen sijaan kykenee ajamaan komennoilla samat pohjiin perustuvat resurssit suoraan pilveen kulkematta muiden lisäsovellusten läpi, kuten esimerkiksi Devops Pipelines. Terraformille voidaan suoraan kertoa komennolla `apply`, että aja nämä samat konfiguraatiot pilvialustalle heti. Näin ollen kaikki muu turha jää pois. Tämän vuoksi pidän Terraformia myös samankaltaisena putkena `apply` komennon jälkeen, sillä se tekee aivan samat asiat kuin Azure Pipelines.

Azure Pipelines kuitenkin on pikkuisen erilainen kehitysympäristöä rakennettaessa, kuten että sille voidaan määritellä erikseen hyväksyjät, kellä on oikeus ajaa kyseisiä koodeja pilveen ja kenellä ei ole oikeutta. Terraform taas ei huomioi käyttäjiään, vaan komento apply ajaa kaiken mitä koodissa on määriteltynä.

Kehityin erityisesti putkien sekä Terraformin käytössä, ja ymmärtämisessä niiden käyttölogiikan miten kumpikin toimii ajaessaan resursseja pilvipalveluun. Logiikka taustalla on sama, ja lopputulos on sama. Ero niiden välillä on pieni kuten kuvassa 4 nähtiin.

### 3.4 Viikko 10

*Maanantai 7.3.2022*

Päivän tavoitteeksi asetin tänään viime viikolla asennetun varmenesalaisuuksien käyttöönottamisen Azuren avainsäilöstä eli Keyvaultista yhdessä tiimin muiden jäsenten kanssa. Tarkoitus olisi saada järjestelmän sisäinen sovellus eli Elasticsearch myös käyttämään näitä varmenteita ja näin ollen turvata HTTP- yhteys. Jos aikaa riittää, pyrimme myös määrittelemään Logstash konfiguraation Terraformissa.

Päivä alkoi asiakaspalaverilla, jonka jälkeen tiimimme kehitti yhdessä projektia eteenpäin. Saimme otettua salasana-autentikoinnin käyttöön Elasticsearch sovellukselle, ja toteimmme että koska palvelu on klusterin sisäinen, ei klusterin sisällä olevalle sovellukselle tarvita varmenteita käyttöön lainkaan, kun graafista käyttöliittymää ei ole olemassa, eikä palvelu näin ollen tarvitse sertifikaattia käyttöönsä.

Aloitimme myös sisäisen Logstash määrittelyn Terraformissa. Näiden kaikkien komponenttien yhteistoiminnan ymmärrys on osoittautunut todella hankalaksi, minkä vuoksi kehitys on hidastunut merkittävästi. Hidastuminen perustuu tietenkin osaamattomuuteen, mutta myös kokemuksen puutteeseen järjestelmäkehityksestäkin. Toisaalta kaikkien on aloitettava jostakin, ja kehittäjiä on opittava kantapään kautta asiat, eli siinä mielessä järjestelmäkehitys tulee lopussa onnistumaan kyllä. Siihen luotamme, ja lopussa yleensä olemme paljon viisaampia, ja toteamme ehkä yhdessä, kuinka nyt tekisi asiat toisin.

*Tiistai 8.3.2022*

Tänään tavoitteeksi asetetaan kahden toisistaan riippuvien sovellusten eli Logstash ja Kafka konfiguraatioiden määrittely ja niiden yhteistoiminnan varmistaminen

Hiekkalaatikossa. Esittelemme myös tiimin uudelle jäsenelle toteutusta ja perehdymme tiimin kanssa yhteisessä työpajassa projektin kehittämisen parissa.

Lisäapua on oikein mukava saada, sillä uudella tiimin jäsenellä on järjestelmäkehityksestä kokemusta, vaikka ei ihan vastaavanlaisista ympäristöistä kuitenkaan. Kokemus komponenttien toiminnasta, looginen näkemys toteutuksesta ja ymmärrys sovellusten riippuvuuksista on selkeä vahvuus, ja kertoo syvemmästä osaamisesta. Juuri tällaista kaipasimme.

Sain itse tehdä tänään muiden kanssa kehitystyötä ja käyttöönotimme Logstash sovelluksen, sekä määrittelimme sille salaisuudet. Uudesta kollegasta oli suuri apu, ja hänen järjestelmäkehityksensä kokemus tuli esiin aika selkeästi, joka auttoi meitä tänään jatkamaan projektin kehitystä.

### *Keskiviikko 9.3.2022*

Tänään tavoitteena on jatkaa projektin kehitystä tiimin työpajassa. Tavoitteena on saada korjattua Kafkan ja Logstash sovellusten välinen yhteys. Myös käyttöönottaa Logstash sovellus toimivana järjestelmän sisällä.

Yhteydet mainitsemieni instanssien välillä saatiin korjattua. Logstash perusasennus saatiin hienosti käyttöönotettua tiimin kanssa. Olemme perehtyneet nyt tiimin kanssa yhdessä kehittämään projektia jokapäiväisissä työpajoissa, ja tämä on onnistunut hienosti. Projekti etenee, ja olemme tavoitetta paljon lähempänä kuin jokin aika sitten.

Tarkoitus on saada hiekkalaatikkoympäristö ensiksi kuntoon, sitten testiympäristö ja sitten tuotantoympäristö. Kaikki perustuu kehittämiimme Terraform konfiguraatioihin, jotka ajetaan putkella Devopsissa. Putken läpi ajaminen rakentaa ensiksi hiekkalaatikon, ja jos sen rakentamisessa tulee virheitä, saamme tietää jo siinä vaiheessa, että jotakin vikaa on. Voimme ehkäistä näin virheellisten konfiguraatioiden ajamisen tuotantoon, joka olisi kehityksen ja ylläpidon kannalta tuhoisaa. Tätä emme halua, minkä vuoksi olemme päätyneet tällaiseen ratkaisumalliin minkä juuri kuvasin.

Tämä on mielestäni ketterämpää kehitystä, ja erityisesti loistavaa loogista ajattelutapaa rakentaa monivaiheinen kehitysympäristö missä on Hiekkalaatikkoympäristö, Testiympäristö sekä Tuotantoympäristö ja joka voidaan ajaa yhdellä kehityspotkella suoraan kehitysympäristöihin Azureen. Toteutuksen loistokkuus on siinä, että olemme ottaneet huomioon virheellisten asioiden päättymisen tuotannossa olevaan järjestelmään. Mitä ennaltaehkäisevämpää, suunnitelmallisempaa ja asiat huomioon ottavampaa kehitysympäristöjen kehitys on, sen turvallisempi, ja toimivampi toteutuksesta tulee. Tämä myös tukee

osaamistani kehitysteemojen pohjalta, sillä pilviympäristön laadukas suunnittelu ja toteutus pilvityökaluilla kasvattaa ohjelmistokehittäjän ammattitaitoa sekä itseluottamusta omaan tekemiseen.

*Torstai 10.3.2022*

Tänään tavoitteena on käyttöönottaa Azuressa Sentinel niminen palvelu, joka on sekä monitorointi että eräänlainen SIEM ratkaisu tietoturvatietojen hallintaan. Se tarjoaa älykäästä tietoturva-analytiikkaa ja uhkien tiedustelutietoa tarjoten ratkaisun hyökkäyksen havaitsemiseen. Microsoft Sentinel sovelluksella voidaan kerätä dataa niin käyttäjistä, laitteista, sovelluksista ja erilaisista ohjelmista. Sillä voidaan myös tutkia uhkia tekoälyyn perustuvalla lisäominaisuudella.

Myös Azuren Container Registry eli ACR on tavoitteena saada tänään käyttöönotettua, sillä Logstash käyttää Sentinelä, ja näin ollen sille tarvitaan image lisäosa, mikä noudataan ACR:stä. Azure Container Registry on Azuren oma tallennuspaikka lisäosille, joita voidaan ottaa turvallisesti käyttöön ohjelmistokehityksessä.

Sain käyttöönotettua Sentinelin tänään, mutta en määritellyt sitä kummemmin, koska oma osaaminen ei ole tuotteen käytöstä niin vahvaa. Kollegamme omaa vahvan taustan Sentinelin käytöstä, joten uskon saavamme apua häneltä tämän kanssa.

Myös ACR otettiin Terraformissa käyttöön ja resurssi ajettiin pilveen. Ajoimme imagen ACR:ään, ja koitimme saada Logstashin käyttämään kyseistä imagea mutta toteutus ei ollutkaan niin helppo sen osilta, että imagen pull Logstashille toimisi moitteettomasti.

Tänään erityisesti kehityin ACR:n ymmärtämisessä, sekä hieman Docker teknologian ymmärtämisessä, sillä imagen lataaminen Azuren resurssiin eli ACR:ään tehtiin dockerin avulla.

*Perjantai 11.3.2022*

Tavoitteena tänään on jatkaa eilisestä eli Logstash ja Sentinel pluginin selvittelyjä. Logstash toimii järjestelmässä hienosti, mutta ei ota pluginia käyttöön, jolloin järjestelmän lähettämää lokia ei saada tutkittua Microsoftin Sentinelillä.

Päivän aikana yritimme saada työpajassa kollegoiden kanssa ratkaistua ongelmaa, mutta se ei tällä kertaa ratkennut. Päätimme viikon tähän, ja jatkamme ongelman selvittelyä ensi viikolla.

### *Viikko 10 analyysi*

Projektissa oli alkavalla viikolla hieman kaksijakoiset tunnelmat projektin etenemisen suhteen. Keskustelimme tiimin kanssa ja päädyimme hankkimaan lisäapua, joka osoittautui hyväksi ratkaisuksi.

Uuden kollegan kanssa kehitys nytkähtikin pikkuisen taas eteenpäin, ja olemme lähempänä tavoitetta. Kokonaisuudessaan olen huomannut, kuinka paljon kehitys on muuttunut ammattimaisemmaksi, ja järjestelmäkehityksessä otetaan aika paljon asioita huomioon. Itselläni on vahva kokemus ja historia IT:n puolelta, mutta vaihtaessani ohjelmistokehitykseen olen todellakin saanut ihan uutta näkökulmaa järjestelmien toteuttamiseen, niiden tietoturvaan ja yleensäkin projektityöskentelyyn. Olen esimerkiksi tutustunut Scrum projektityöskentelyn viitekehikseen, ja mielestäni alussa kokemattomuus paistoi selkeästi toimintatavoissamme. Olemme ehkä parantaneet kuitenkin siitä, ja päässeet Scrumin ytimeen, kuinka työskennellä niin että saisimme asiakkaalle lisäarvoa tekemillämme asioilla. Ajoittain huomaan vielä kuitenkin puutteita toimintatavoissamme, ja etenemme vaikeimpien asioiden kautta mikä ei olekaan enää kustannustehokasta vaan päinvastoin, aikaa ja rahaa vievää tutkiskelua ja tekemistä.

Scrumin tavoitteena on tuottaa lisäarvoa, ei suinkaan niin että koko tiimi perehtyy yhdessä tekemään jotakin kriittistä ja vaikeaa sekä suurta, vaan se saa ja kuuluu olla priorisoituna niin että tehdään pienempää, ja lisäarvoa tuottavaa, joka saadaan toteutettua nopeammin. Sen tulisi olla niin että tehdään asiat sellaisessa järjestyksessä millä tuotetaan se lisäarvo asiakkaalle. Kehitetään ja toteutetaan niitä asioita, joissa oma osaamisemme ja vahvuutemme on huipussaan. Vaikeammat toteutukset jätetään loppupäähän, missä tutkitaan ja selvitetään kuinka kaikki nivoutuvat yhteen. Uskon että olemme juuri tässä vaiheessa projektia tällä hetkellä. Projektin aloittamisen jälkeen toteutimme tämän vastaavan järjestelmän omassa pilviympäristössämme ja saimme riippuvuudet, sekä sovellusten toiminnan kuntoon. Teimme minimitoteutuksen tästä projektista toimivana.

Kuitenkin on paljon lisäasioita tullut mukaan, konfiguraatiot ovat muuttuneet noista radikaalisti ja paljon asioita pitää miettiä järjestelmän turvallisuuden, toimivuuden ja käytettävyyden kannalta. Tämä on vaikeuttanut osaltaan työskentelyämme ja ainakin itse olen kehittynyt enemmän ajattelemaan tiettyjä asioita esimerkiksi turvallisuuden kannalta. Tämä on avannut paljon asioita minulle ja niiden merkitys on korostunut entisestään.

Kehityin loppuviikosta, sillä saimme käyttöön otettua sovelluksia, sekä testattua järjestelmän erilaisia riippuvuuksia. Erityisesti myös tietoperustan kehitysteemana oleva ketterä

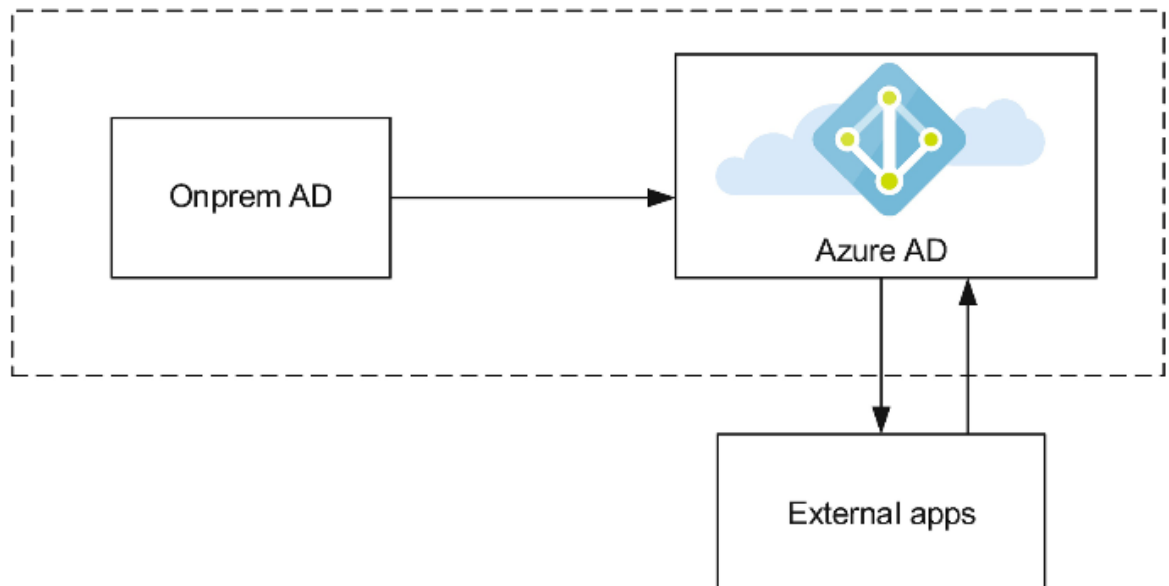
projektityöskentely eli Scrum kehitti minua sillä saimme kehityksemme ansiosta tuotettua asiakkaalle lisäarvoa työskentelytapojemme kautta. Saimme toteutettua tietoturvallisen ratkaisun asiakkaan omien levykuvien säilyttämiseen, sekä sovellusten riippuvuuksia korjattua, mikä edistää tietojärjestelmän parempaa toimintaa.

Tarkoitus on saada projekti mahdollisimman äkkiä siihen pisteeseen, että voimme esitellä sen asiakkaalle ja tehdä myös kyseiselle toteutukselle tietoturva-auditoinnin.

Uusi kollega toi mukanaan paljon osaamista, ja projektilla saimmekin paljon uutta aikaan. Kollega on usean vuoden kehitystyössä ollut ammattilainen, joka hallitsee tietoturvan puolen hyvin. Kehityin erityisesti kollegan näyttämän erilaisten sertifikaattien ja salaisuuksien puolesta, sillä tietoturva on yksi osa-alue mikä on koko projektin aikana päällimmäisenä mielessä.

Teoksessa Azure Security Handbook avataan erityisesti pilvitietoturvan eri osa-alueita, ja sitä kuinka tietoturvamielessä identiteetin ja pääsynhallinta toimii. Hyvin yleinen tapa suur yrityksissä on toteuttaa hybridiympäristöjä missä mukana on yrityksen On-Premise ympäristö sekä Microsoftin Azure Active Directory.

Kuvassa 5 näkyy yleisimmän hybriditoteutuksen käytäntö suurissa asiakasyrityksissä.



Kuva 5. Hybridiympäristö (Azure Security Handbook 2021, 19)

On-Premise Active Directory ympäristö voidaan yhdistää osaksi Azure Active Directorya, koska se tukee useita identiteettilähteitä. Käyttäjät voidaan määrittellä omien järjestelmien kautta synkronoimalla käyttäjät ja muut identiteetit suoraan Azureen. Tätä varten pitää asentaa omaan palvelimeen Azuren Directory Connect palvelu.

Käytännössä yrityksen oma palvelin missä on Active Directory hakemistopalvelut, pitää yhdistää osaksi Azure Active Directorya Directory Connect palvelun avulla, jolloin saadaan Microsoftin parempaa tietoturvaa käyttöön esimerkiksi O365-palveluiden kirjautumisten suhteen. Käyttöön voidaan ottaa tällöin tietoturvallinen Single Sign-On kirjautumismenetelmä joka takaa Multifactor Authentication menetelmän, eli tietoturvallisen kaksivaiheisen kirjautumisen Microsoft palveluihin. Kaksivaiheinen kirjautuminen tarkoittaa sitä, että kirjaututaan yhdellä käyttäjätunnuksella normaalisti palveluun sisään, mutta ennen kuin palveluun pääsee sisään, pyydetään kirjautujaa vielä vahvistamaan kyseinen kirjautuminen koodilla joka lähetetään käyttäjälle joko tekstiviestitse tai sähköpostitse.

### 3.5 Viikko 11

*Maanantai 14.3.2022*

Tänään meillä on täysi porukka taas projektia tekemässä, ja tavoitteena on jatkaa Logstash nimisen sovelluksen parissa sekä Azure Container Registry mihin aiomme ladata Sentinel pluginin, ja käyttöönottaa sen Logstash sovelluksella.

Viime viikon ACR sekä Sentinel päätettiinkin viime hetkellä toteuttaa suoraan tuotantoympäristöön, sillä ACR-palveluun tulee ainoastaan yksi image, ja näin ollen kyseisen resursin ei tarvitsekaan olla Hiekkalaatikkoympäristössä.

Aamun palaverien jälkeen pääsimme projektinkehitykseen. Saimme tiimin kanssa yhdessä sovelluksen asennettua kehitysympäristöön. ACR saatiin asennettua, mutta ACR:n kanssa oli ongelmia, sillä Logstash pod ei löytänyt kyseistä Sentinel lisäosaa, jonka olimme registryyn ladanneet. Tavoitteena on välittää Logstashista suoraan Sentineliin dataa, ja suorittaa Sentinelillä erilaisia hakuja. Tämän ongelman selvittely ei prioriteetissaan ole tärkein, joten aiomme selvittää tätä toisena päivänä.

Itse tykkään henkilökohtaisesti, kuinka tiimimme yhteinen tekeminen on korostunut viime aikoina. Olemme entuudestaan ottaneet Scrumiin liittyvät toimintatavat osaksi työarkeamme, ja kehitämme projektia yhdessä päivittäisissä työpajoissa. Erityisesti projektin vaiheessa, missä olemme lähellä tuotantoon käyttöönottamassa kyseistä järjestelmää, on tärkeää tehdä työpajoissa yhdessä asioita vaihe vaiheelta, missä erilaiset henkilöt ovat vahvimmillaan.

Esimerkiksi joku voi olla pilvipalveluiden kehittämisessä osaavampi kuin toiset, kun taas joku toinen voi olla taitavampi järjestelmän sisäisten sovellusten kehittämisessä ja

käyttöön otossa kuin toinen. Kaikilla on omat vahvuutensa, ja osaamisalueensa minkä perusteella projektitiimi on valittu. Mitä monipuolisempi osaaminen, sen parempi kehitysprojekti ja tuote loppuasiakkaalle saatetaan.

Tavaksi tiimissämme on muodostunut, että aamupalaverien jälkeen minä kutsun projektin työpajaan jokaisen tiimiläisen, ja teemme tiimin kanssa yhdessä töitä. Työpaja on muodostunut äärimmäisen hyväksi ja tehokkaaksi, sillä saamme enemmän asioita aikaan, ja opimme samalla muilta.

Kehitys jäi hieman vajaaksi tänään, sillä emme saaneet paljoa tuotettua kun Logstash ei lukenut imagea ACR:sta. Positiivisena pidän työyhteisön ja projektin jäsenten yhteishenkeä ja tahtoa selvittää asioita, vaikka suoranaista ratkaisuaakaan ei aina ole löydettävissä. Ryhmätyöskentely auttaa kasvamaan niin työntekijänä, kuin myös avaa sosiaalisia yhteyksiä mahdollisesti uusiinkin ihmisiin.

*Tiistai 15.3.2022*

Tänään tavoitteena on saada kehitysympäristöjen putkitus kuntoon, ja jatkossa ajaa muutokset putkien kautta järjestelmälle Azure Devops pipeline avulla. Saamme myös kollegan konsultoimaan tänään tehtyjä muutoksia.

Kollega oli saanut muutokset toteutettua, ja putkituksen kuntoon. Käytännössä kun putkella ajettavat muutokset tehdään Azure Devopsissa, ei Terraform komentoja enää tehdä. Yleinen käytäntö ohjelmistokehitykselle on se, että kaikki järjestelmän rakentamiseksi kuvattu tekeminen kokonaisuudessaan tehdään Terraformilla koodaamalla. Valmiit koodatut tiedostot ajetaan putkien avulla Devopsissa, ja erilaisten muutosten hallinta hoidetaan aina Terraformilla muuttamalla koodia.

Kuitenkaan loppuvaiheessa Terraform komennoilla ei muutoksia enää ajeta kehitysympäristöön, koska vaarana saattaa olla esimerkiksi lokaalin koneen muutokset, jotka menevät koko projektin päätaustatiedostoon, ja voi aiheuttaa suuremman vian koko järjestelmän toiminnassa.

Jos tässä vaiheessa projektin päätaustatiedosto eli backend kokee jonkin muutoksen joutuen yksittäisen paikallisen koneen tiedostoissa olevista vääristä arvoista, saattaa Terraform hyväksyntä eli apply aiheuttaa massiivisen järjestelmän hajoamisen. Tämän vuoksi käytäntö on, että ajot toteutetaan Devopsilla.

Kehityin tänään Azure Devopsin käytössä, ja putkien luomisessa, sekä ajamisessa pilvipalveluympäristöön. Kehitysteeman työkalun Terraformin käytännöt verratessa kehitysalusta Azure Devopsin putkien ajamisessa on tärkeässä roolissa. Paikallisella Terraform työkalulla ei voida ajaa konfiguraatioita pilviympäristöön, vaan toteuttaminen tapahtuu Devops putkilla.

*Keskiviikko 16.3.2022*

Tänään tavoitteena on jatkaa edelleen ACR:n parissa, sekä saada lokia kulkemaan Sentinelin ja käyttöönottaa visuaalinen käyttöliittymä Kibana.

Iltapäivän työpajassa saimme yhdessä tiimin kanssa onnistuneesti ACR:n ja imagen käyttöönotettua Logstash podille. Sentinel on yksi Azureen resursseista, mille ei virallisesti löydy Terraform koodia, eli sitä ei voida käyttöönottaa Terraformilla. Sen vuoksi Sentinel käyttöönotettiin manuaalisena Azureen, ja viittasimme Terraform koodissa kyseiseen olemaan olevaan resurssiin datalähteenä. Sentinelin saatiin tehtyä kysely, missä lokidataa tuli Logstash podilta onnistuneesti.

Asensimme myös kehitysympäristöön Kibana nimisen sovelluksen. Emme määritelleet sitä kuitenkaan vielä loppuun päivän päättymisen vuoksi. Edistys on viime aikoina ollut hyvää uuden tiimin jäsenemme ansiosta. Hänellä on vuosien kokemus järjestelmäkehityksestä ja se on auttanut projektissa meitä kaikkia eteenpäin.

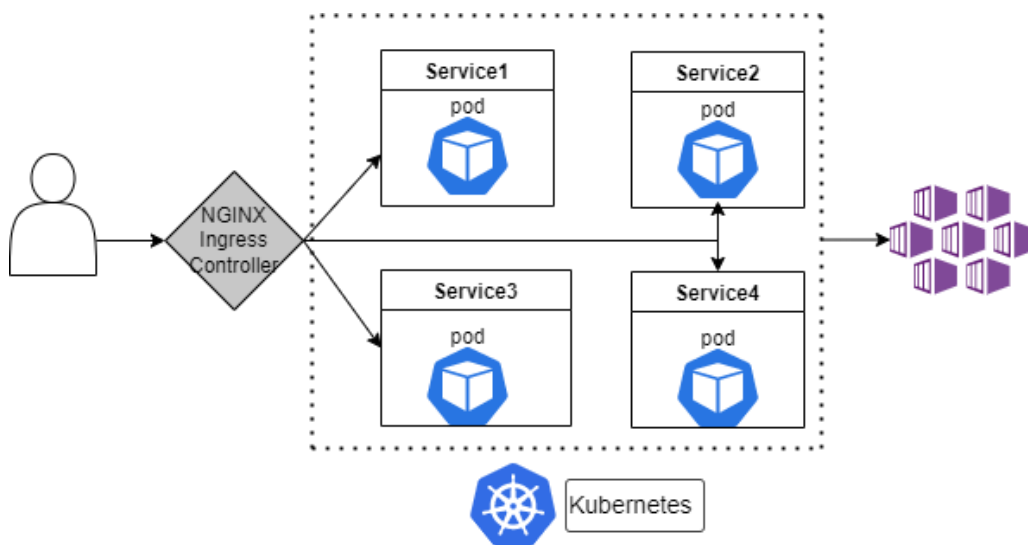
Kehitys on myös itselläni ollut hyvää, sillä olen saanut oppia häneltä paljon. Tunnistan itsessäni aiemmalta ajalta sellaisen varovaisen persoonan erityisesti, kun kehitysympäristömme on muuttunut niin paljon ammattimaisemmaksi, että en ehkä ole uskaltanut toteuttaa niin rohkealla tyylillä ja rakentaa sekä testailla erilaisia kokeiluja, muuttaa koodia radikaalisti tai koskea muiden tekemisiin, mutta olen nyt viime aikana oppinut sen, että kokeilla saa, ja pitää. Olen ollut säikky, sekä pelännyt hajottavani kehitysympäristön vaiheessa, jossa ympäristön tulisi olla jo toiminnassa. Järjestelmän tuotantoon ajaminen lähestyy parin kuukauden sisällä, ja tässä vaiheessa pitäisi olla koko kehitysympäristö jo valmis, sekä kovassa testauksessa. Olemme kuitenkin lähellä tätä tavoitetta, sillä tammamme toteuttaa järjestelmä, on rakentaa Hiekkalaatikkoympäristö niin pitkälle valmiiksi, että voimme ajaa saman konfiguraation testille, että myöskin tuotannolle.

Putkella ajettaessa voimme siis varmistua, että jos hiekkalaatikkoon tehdyt muutokset eivät mene läpi, on vika korjattava ennen kuin toteutus ajetaan testille ja tuotannolle. Jokaiseen ympäristöön tehtävät muutokset vaativat hyväksynnän henkilöiltä, joilla on oikeus hyväksyntä tehdä. Muutoksia ei voi siis ajaa ennen kuin hyväksyntä on tehty. Tämä

varmistaa myös sen, että kuka tahansa ei voi muutoksia järjestelmään tehdä. Myös hyväksynnällä kyseinen henkilö vastaa järjestelmälle aiheutuvista muutoksista.

*Torstai 17.3.2022*

Aamupalaverien jälkeen tavoitteena on saada kehitysympäristöön Kibana määriteltyä toimintakuntoon, sekä käyttöönottaa NGINX Controller, joka ohjaa yhteyksiä ulkoverkosta klusterin sisäisille podeille. Kuvasta 6 näkee, kuinka Nginx Ingress Controller ohjaa yhteyksiä eteenpäin.



Kuva 6. NGINX Ingress Controller yhteydet

NGINX Ingress Controller ohjaa yhteyksiä klusterin sisäisille podeille. Tässä tilanteessa controllerin tarve oli selkeä, sillä Kibana sovellukselle tarvitaan julkiverkosta yhteys käyttöliittymään, minkä vuoksi NGINX Ingress Controller on selkeä vaihtoehto. Kyseinen controlleri toimii välissä liikenteen välittäjänä minkä kautta päästään Kibanan käyttöliittymään selaimella käsiksi. Liikenteen tietoturvan hallinta ja pääsynohjaus hallitaan Network Security Group nimisellä resurssilla, missä määritellään mistä verkoista, tai IP-osoitteista Kibanan käyttöliittymään on pääsy. Kehitystä tapahtui, ja saimme onnistuneesti toteutukset käyttöön. Kibana käyttöliittymä toimii hyvin.

*Perjantai 18.3.2022*

Päivittäisessä Scrum-palaverissa keskustelimme lisäarvon tuottamisesta asiakkaalle visuaalisesti. Tavoitteena tänään on toteuttaa kehitysympäristöön vielä sellainen valvontamillä voidaan nähdä klusterin sisäisten resurssien, ja erilaisten komponenttien, sekä toimintojen suhteen oleellisia tärkeitä ajantasaisia raportteja, sekä viivakaavioita.

Päädyimme toteuttamaan Grafana nimisen valvontajärjestelmän millä kerrottu monitorointi voidaan toteuttaa. Grafana on avoimen lähdekoodin alusta, jolla voidaan tehdä kyselyjä, visualisoida erilaisia näkymiä, sekä luoda hälytyksiin perustuvaa automaatiota. Grafanan avulla saadaan selville minkä tahansa olemassa olevaan dataan perustuvaa tietoa, olipa se sitten Kubernetes klusterista tai jostakin muusta palvelusta. Sen avulla kerätään esimerkiksi Kubernetes klusterin sisäisten resurssien dataa, kuten niiden päällä oloa, muistinkäyttöä, levytilaa, tai monia muita haluttuja kohteita yhdestä hallintapaneelistä. Käyttö perustuu Dashboardiin eli näkymään, mihin voidaan halutessa tuoda klusterin resurssit ja rakennella niistä haluamamme näköisiä näkymiä. Kaikesta Grafanan keräämästä datasta voidaan luoda visuaalisia raportteja tietohallinnon tueksi. Grafanalla myös voidaan rakentaa erilaista hälytysautomaatiota perustuen syötettyihin arvoihin, minkä avulla se hälyttää lähettämällä esimerkiksi sähköpostia haluttuun kohteeseen.

Grafana otettiin käyttöön Helm chartin avulla, ja määriteltiin toimimaan yhdessä Prometheusin ja Alertmanagerin kanssa. Prometheus on myös avoimen lähdekoodin valvontasovellus, ja hälytystyökalu tässä järjestelmäkokonaisuudessa. Se kerää ja tallentaa metriikkansa aikasarjatietoina, eli metriikkatiedot tallennetaan aikaleiman kanssa, jolloin ne tallennettiin sekä valinnaiset avainarvoparit, joita kutsutaan nimiöiksi.

The Linux Foundationin mukaan metriikka on numeerinen mittausarvo, joka kirjataan ajan myötä. Metriikoilla on tärkeä rooli sen ymmärtämisessä, miksi sovellus toimii tietyllä tavalla. Metriikan avulla päästään ymmärtämään paremmin sovelluksen käyttöä, ja sen toimintaan liittyviä tapauksia ja voidaan jopa ennaltaehkäistä tulevia mahdollisia kaatumisia, tai muita sovelluksessa tapahtuvia virheitä. Esimerkkinä voidaan käyttää sellaista, että kun sovelluksessa olevien pyyntöjen määrä lisääntyy eksponentiaalisesti, voi sovellus hidastua tai kaatua kokonaan. Näiden pyyntöjen määrän havaittuamme voimme toteuttaa sovelluksen kuormitusta vähentäviä toimenpiteitä kuten lisätä sovellukselle muistia ja lisätä suorittimien määrää, joka vähentää kuormaa ja tasaa tilannetta. (The Linux Foundation 2022)

Tavoitteet toteutuivat tänään, ja kehitys oli hyvää. Saimme lisäarvoa tuottavan sovelluksen käyttöönotettua järjestelmän tueksi, ja pääsemme esittelemään asiakkaalle sitä ensi viikolla.

### *Viikko 11 analyysi*

Viikko oli mielestäni erittäin onnistunut ja hyvä etenkin projektin kehitysaskelien puolesta. Saimme pientä lisäarvoa tuottavaa toteutettua joka edistää tietojärjestelmän käyttöönottoa entisestään.

Kuten käytäntö on sen osoittanut, on projektitiimissäni maanantaipäivät kehityksellisesti hieman heikompia kuin toiset, sillä suuri osa viikon palavereista on aina maanantaisin. Useat palaverit estävät isomman kehityksen päivän ajalta.

Kuitenkin aina loppuviikon aikana projekti edistyy, ja nyt periaatteessa olemme tilanteessa missä järjestelmän sisäiset sovellukset kaikki on saatu onnistuneesti käyttöön otettua, ja kehitys lähinnä on sellaista resurssien säätelyä, eli varmistetaan että resurssit toimivat, yhteydet toisten sovellusten kanssa toimivat, sekä liikenne ja tietoturva on kunnossa.

Päärooliin on selkeästi noussut tiimin yhteiset työpajat, missä olemme saaneet jakaa toistemme kanssa näkemyksiä kehityksestä, ja saanut tukea kollegoilta tekemiseen. Olen huomannut sen, että oma vahvuus ei karise, vaikka en osakaan kaikkea. Tiimissämme ollaan avoimia toisillemme, ja tuemme toisiamme, vaikka osaamistaso olisikin lähtökohtaisesti heikompi. Tämä antaa itselleni voimaa työn tekemiseen, kun tiedän että tiimi ei paheksu, vaikka tekisinkin virheitä. Virheistä kuitenkin kuuluu ottaa opiksi, ja tämä kehittää itse yksilöä ammattimaisemmaksi työntekijäksi.

Olen mielestäni täysin oikealla tiellä ohjelmistokehittäjänä, ja opettelen sekä kehitän muiden kanssa ahkerasti projektia. Vaikka tämä on ensimmäinen projektini, uskon että muitakin tulee vielä paljon ja saan kehittyä ihan rauhassa. Yhdeksi tärkeimmäksi osa-alueeksi nostaisin kuitenkin kehitysteemasta oman kasvullisen kehityksen ohjelmistokehityksessä. Myös oman ympäristön ja oman viihtyvyyden, sillä Itselläni ei ole mahdollisuutta esimerkiksi lähteä kodista työpöydän äärestä pois, koska kannettavan tietokoneen näyttö on aika pieni.

Uskoisin viihtyvänä esimerkiksi jossakin kahvilassa tai muussa sellaisessa rauhallisessa paikassa missä saa keskittyä. Yleensä sama tuttu ympäristö alkaa tylsistyttämään, ja aika ajoin saattaa huomata tekevänsä jotakin muuta kuin itse projektia tai projektinosaa. Myös uusi virkeämpi ympäristö saattaa aiheuttaa keskittymiselle paremman puitteen, ja kehitystyö voi olla runsaampaa. Itsestä huolehtiminen vapaa-aikana on tärkeä osa jaksamisen kannalta, sillä riittävä uni, terveellinen ruoka ja säännöllinen liikunta auttaa jaksamiseen.

Päätetyössä tärkeää on myös itselleni ajoittainen irtautuminen näytöstä esimerkiksi viideksi minuutiksi ja kävellä hieman. Olen kuitenkin huomannut sen, että kahdeksan tunnin työpäivä sisällä alkaa olemaan aika haasteellista, joten ajattelin laajentaa käyntiä viiden minuutin happihyppelyksi ulkona. Runsas veden juonti pitää nesteytettynä hyvin, ja virkeämpänä mielen. Pidän erittäin tärkeänä osana omaa jaksamista päätetyössä työskenteleville ihmisille.

Pohtiessani työympäristöä, ja itsestä huolehtimista korostuu oma hyvinvointi entisestään.

Teoksen Just sopivasti ketterä mukaan kaikki voi näyttää todellisuudessa hyvältä. Sinulla voi olla hyvä työ, rakastava puoliso, menestyneet ystävät, paljon omaisuutta ja lomareisuja, mutta silti jostain syystä olo on tyhjä. Tuohon tulisi etsiä vastausta kysymykselle miksi? Heidän mukaansa määränpää on kadonnut tai jotenkin hukassa jos tilanne on tällainen. Pitäisi pohtia oman työn merkityksellisyyttä, ja sitä onko oma elämä tarpeeksi laadukasta. (Niemi & Hietaniemi 2020, 158)

Olen vertaillut Niemen ja Hietaniemen teorioita, ja todennut että minunkin tulisi määritellä se määränpää, jota tavoitella. Elämäni on vaihtelevasti laadukasta ja toisinaan taas ei. Onko siinä se syy, miksi välillä tuntuu sille että olo on tyhjä. Sellaisina hetkinä kuin on tuollainen tunne, vaikuttaa se myös vahvasti työn mielekkyyteen ja laatuun. Tunnistan kuitenkin oman elämäni vaikuttavat tekijät, ja yhdeksi suurimmaksi tekijäksi miellän stressin ja kokonaiskuorman joka tällä hetkellä on elämässäni. Monelle saattaa nimenomaan se harteilla oleva kuorma vaikuttaa työntekemiseenkin, ja olenkin priorisoinut oman työkuormani purkamisen osiin pikkuhiljaa. Tämä antaa minulle hiljalleen parempaa oloa, ja mielekkyyttä suhtautua asioihin positiivisemmalla tavalla. Aion tavoitella laadukasta elämää, missä harteillani ei paina kuorma, vaan saan nauttia elämisestä ja itsestäni huolehtimisesta.

### **3.6 Viikko 12**

*Maanantai 21.3.2022*

Päivän tavoitteena on esitellä projektia asiakkaalle tänään. Myös tarkoitus on ottaa ympäristökohtaiset Azure Sentinelit käyttöön. Sentinel otetaan jokaiseen kehitysympäristöön käyttöön, koska sillä voidaan kysellä esimerkiksi klusterista ja sen pödeista oleellista tietoa.

Asiakkaalle esiteltiin projektia lähinnä monitoroinnin puolesta, eli näytimme hieman valvontasovellus Grafanan toimintoja, ja graafeja, jotka tulevat klusterin resursseista. Avasimme asiakkaalle hieman sitä, millaiseen käyttöön sovellus on, sekä kuinka sillä voidaan käyttöönottaa esimerkiksi hälytyksiä kyseisistä resursseista. Demo oli onnistunut ja sovellus selkeästi auttaa kehittäjiä, ja muita oikeutettuja asiahenkilöitä ylläpidossa jatkossa.

Iltapäivällä aloin määrittelemään Azure Sentinelä Terraformissa. Rakensin koodin siten, että se ajetaan resurssina pilveen jokaiseen kehitysympäristöön putkella. Määrittely perustuu erilaisiin muuttujiin mistä haetaan asennukselle oma kehitysympäristö. Putken ajetaessa pilveen luodaan resurssi, joka hakee nimet muuttujista. Esimerkiksi hiekkalaatikko-ympäristöön ajetaan la-sentinel-sandbox, kun taas vastaavasti testiympäristöön ajetaan la-sentinel-test niminen resurssi. Muuttujia voidaan hyväksikäyttää multiajossa ja se helpottaa työskentelyä todella paljon. Muuttujia käyttämällä vältetään monen ympäristön rakentaminen yksitellen. Tämän jälkeen määrittelin Sentinelin käyttämään ympäristökohtaisia Log Analytics Workspace resursseja, koska Sentinel tarvitsee kyseisen resurssin toimiakseen.

Log Analytics Workspace on työkalu, jolla voidaan suorittaa erilaisia kyselyjä halutusta kohteesta. Sillä voidaan analysoida kyselyjen tuloksia ja määritellä vaikkapa ehtoihin perustuvia kyselyjä, jolloin saat juuri sen ajantasaisen tiedon mitä haet.

Yritin myös saada Logstashin käyttöönottamaan keyvaultista LAW:n Workspace ID:tä ja Keytä, mutta en saanut tätä toimimaan. Resurssin salaisuudet eivät oikein hakeudu oikealla tavalla, minkä vuoksi tietoa Logstashista ei nyt mene Sentineliin. Tuohon palaamme myöhemmin uudelleen. Kehityin tänään etenkin Sentinelin käytössä. Sain määriteltyä resurssin kivuttomasti Terraform koodiin, ja ajettua sen putkella kehitysympäristöihin.

*Tiistai 22.3.2022*

Tänään tavoitteeksi asetan itselleni jatkavani eilisen Sentinel ongelman parissa. Myös tarkoitus on saada Kafkan monitorointi Grafanaan toteutettua tiimin kanssa työpajassa.

Sain määritettyä Terraformissa LAW:n pilviresurssin asentamisen ympäristökohtaisesti. Nyt kyseinen resurssi asentuu ajossa Hiekkalaatikkoympäristöön, sekä Testiympäristöön. Määrittelin uudet salaisuudet Logstash secreteille, ja näin ollen Logstash alkoi välittää dataansa myös Sentineliin omiin kehitysympäristöihin.

Grafanaan teimme tiimin kanssa Kafkan monitoroinnin kuntoon Terraformissa. Nyt Grafanaan tulee Kafkan podilta metriikkaa, jota voidaan tarkastella sovelluksella tarkemmin.

Kehityin tänään Terraformin käytössä, sillä sain ympäristökohtaisen konfiguraation käyttöönotettua Log Analytics Workspacelle.

*Keskiviikko 23.3.2022*

Tänään tavoitteena oli tehdä hieman siivoustöitä Terraform konfiguraatioista, ja toteuttaa muutospyyntöjä tiimiläisiltä.

Kollegani halusi rakentaa Kafkalle Zabbix valvonnan joka on monitorointiin ja valvontaan tarkoitettu ohjelmisto millä voidaan valvoa pilviresursseja, ja tarvitsi siihen client id:n ja salasanan. Tein Terraformilla käyttäjän, ja sille salasanan Keyvaultiin. Yhdessä tiimin muiden jäsenten kanssa parantelimme myös hieman Grafanaa, sekä järjestelmän muiden sovelusten toimintaa.

Kun projektin tila on siinä kunnossa, että hieman paineetkin hellittävät, on mukava tehdä välillä sellaisia pieniäkin muutostöitä ja ehkä hengähtää samalla, sillä viime aikoina on ollut tiimillä hieman paineita saada järjestelmä sellaiseen tilaan, että voidaan olla varmoja sen toiminnasta.

Lueskeltuani teosta Just Sopivasti Ketterä, havahtuin siihen, että työympäristöni ei ole millään tavalla houkutteleva. Päätin siltä istuimelta alkaa pienin muutoksin toteuttamaan oman työympäristön päivittämistä ja kävin hankkimassa itselleni toisen näytön helpottamaan työntekoa. Olen tiedostanut, että hyvä ja mielenkiintoinen työympäristö yhdessä oman elämäni kanssa muodostavat ratkaisevan tekijän työn mielekkyydelle, ja siitä palautumiselle.

Goforen Niemi & Hietaniemi mukaan kaikki lähtee keskittymisestä. Oleellista on se, mitä asioita teen ja millä tavalla. Vievätkö teot minua oikeaan suuntaan, ja antavatko tekoni minulle sellaisen riittävän rauhan, jossa voin keskittyä paremmin työn tekemiseen. (Niemi & Hietaniemi 2020, 50–51)

Selvittäessäni Niemen ja Hietaniemen teoriaa, totesin että työympäristöni ei ole kovin houkutteleva, sillä kotonani kuitenkin on rajalliset mahdollisuudet tehdä töitä. Minulla on aika pieni toimisto, talvisin se on vähän kylmäkin ja toimistoni on täynnä tavaraa ympärilläni. Uskon sen olevan yksi osatekijä siinä, että aina töitä ei houkuttele tehdä ympäristössä missä on paljon haittoja. Myös houkuttelevuuteen liittyy aika moni muukin osatekijä, sillä vapaa-ajalla tekemäni asiat liittyvät myös vahvasti työn tekemiseen ja sen mielekkyyteen. Tulisi olla terveellinen ruokavalio, harrastaa liikuntaa sopivissa määrin ja myöskin levätä. Nämä ovat tärkeitä elementtejä siihen, miltä seuraava työpäivä tuntuu ja kuinka keskittyminen onnistuu.

Tiedostan että esimerkiksi väsyneenä työn tekemisen mielekkyys laskee ja kiinnostus katoaa. Siksi minun tulisi nukkua pidempiä öitä. Ruokavalioni kaipaisi kohennusta, sekä lepoa olisi hyvä saada sopivissa määrin. Kaikki nämä ovat niitä asioita millä voisin kohentaa oman työn mielekkyyttä ja pystyisin keskittymään paremmin.

Yhtä kaikki, on kuitenkin olemassa houkutuksia. Perheellisenä miehenä arkeni määrätty aika paljon lasten mukaan. Lapsen harrastukset, koululäksyt ja muu huomionanto on aikaa vaativaa, sekä niin fyysisesti kuin psyykkisestikin mieltä ja kehoa kuluttavaa. Työpäivän päätteeksi lepoa ei oikeastaan kerkeä saamaan, vaan rasitus jatkuu pitkälle iltaan erilaisissa riennoissa.

Kun oman rauhan saa illalla, ihmismieli kaipaa siinä vaiheessa omanlaistansa lepoa, ja yleensä itselläni aukeaa jokin suoratoistopalvelu iltaisin. Olen mieltänyt sen aina omaksi ajaksi, mutta nyt ymmärrän sen olevan juuri se houkutus mikä vaikuttaa keskittymiseeni.

Minun tulisi panostaa enemmän itseeni, ja omaan hyvinvointiini työn tekemisen ja koulunkäynnin sekä perhe-elämän lisäksi. Tämän tueksi tarvitsen aikataulun minkä mukaan rakennan arkeni. Jämerällä ajanhallinnalla tulen hallitsemaan tulevaa keskittymistä, mutta myös tasapainottamaan tarpeellista lepoa.

Toinen asia, minkä tulen korjaamaan on työympäristö, ja sen haittojen poistaminen toimistosta. Pienet muutokset saavat positiivista kaikua aikaan, ja antavat enemmän mielekkyyttä työn tekemiselle.

Kolmas asia on saada tasapainoinen liikunta osaksi elämää ja arkea. Harrastan kylläkin liikuntaa noin 5–6 tuntia viikossa, mutta sekin saattaa olla liikaa sen kuorman lisäksi mitä minulla on entuudestaan. Tehokas ja ajallisesti sopiva suunnitelma liikkumisen tueksi tulee parantamaan jaksamistani työpäivinä.

Neljäntenä tulen korjaamaan sähköisten viestimien ja laitteiden käyttöä. Tiedostan että pitkään yöhön ei kuuluisi katsella suoratoistopalveluita, joten aion karsia sen nyt pois, ja panostaa enemmän itseeni, sekä nukkumiseeni. Some, suoratoistot, podcastit, musiikit tulee karsia minimiin sillä nämäkin ovat palveluita, jotka kuormittavat iltaisin aivoja, eikä uni välttämättä tule yhtä tehokkaasti.

*Torstai 24.3.2022*

Tänään tavoitteena on saada AKS:lle Azuressa tagit kuntoon. Yleensä resurssien tagaus Azuressa helpottaa yrityksen laskutuksen suhteen sillä tagien perusteella voidaan ohjata laskut oikeaan paikkaan.

Työstin tagien konfiguraation kanssa jonkin aikaa, sillä vaikka putki menikin läpi Hiekka-laatikkoympäristölle, ei tagit tulleet resurssille näkyviin. Tämän ihmettelyyn ja erilaiseen kokeiluun meni jonkin aikaa, kunnes huomasin kollegani tehneen kyseiseen konfiguraatioon lyhyehkön koodipätkän, mikä ei huomioi kaikkia tulevia tageja. Tämän poistaminen auttoi välittömästi ja sain asetettua oikeat tagit järjestelmään laskutusta varten.

Iltapäivän aikana oli työnantajani Kyberturvallisuudesta kertova tietoisuus, joka liittyy myös ohjelmistokehitykseen vahvasti. Tietoisuudessa kävimme läpi tämän päivän uhkia, sekä mahdollisuuksia. Kehitystä tapahtui ongelmanratkaisussa, sillä sain selvitettyä vian tagien toimimattomuuden suhteen, sekä kehityin Terraform työkalun käytössä.

*Perjantai 25.3.2022*

Aamun päivittäisen projektipalaverin päätteeksi tavoitteeksi valikoitui tänään asiakasjärjestelmän dataohjauksen käyttöönoton Kibanalalle sekä tehdä tarpeelliset muutospyyntö Terraformilla.

Teimme tiimissä käyttöönoton onnistuneesti, ja testidataa alkoi tulemaan Kibanaan. Teimme Kibanan visualisointien määrittelyä, ja erilaista testaamista. Samalla työstimme Elasticsearch ja Logstash pödeja tuottamaan tarpeellista dataa Kibanaan.

Päivä oli pikkuisen helpompi tänään minun osaltani, ja sain rauhassa työstää omia tehtäviäni. Välillä tehtävät myös jakautuvat niin että tällä kertaa minulla oli aikaa tarkistella kehityksen koodeja, ja putsailla roskat pois koodista. Virallista kehitystä ei tänään tapahtunut suuremmin.

*Viikko 12 analyysi*

Mielenkiintoinen viikko takana. Kuten jo mainitsin, helpompi viikko kun työkuorma on alkanut pikkuhiljaa helpottamaan. Oikeastaan pienehköjä muutostöitä tehtiin koko viikon aikana. Projekti kuitenkin edistyi tiimin yhteisellä tekemisellä ja projektin tila näyttää valoisalta tällä hetkellä.

Kehitykseni kaiken kaikkiaan on tämän projektin puitteissa ollut huimaa. Olen saanut oppia parhailta, sekä oppinut myös itsekin tekemään ohjelmistokehitystä tietyllä tavalla. Tiedostan, että kukaan ei ole mestari syntyessään, eikä kaikkea voi osata vuosienkaan kokemuksella, mutta ohjelmistokehityksen houkutus itselleni on nimenomaan siinä, että asioita pitää etsiä ja tutkiskella itse.

Olen pohtinut tällä viikolla itseni kehittämistä entuudestaan ja verrannut sitä tietoperustojen kehitysteemoihin kuten oman ammattitaidon kehittämiseen ja ohjelmistokehittäjänä kasvamiseen. Perehtyessäni Niemi ja Hietaniemen teokseen Just Sopivasti Ketterä, olen enemmän alkanut välittämään omasta työympäristöstäni sekä hyvinvoinnistani. Ympäriällä olevat houkutukset saattavat ajoittain saada mielenkiinnon pois kehitystyöstä, ja olen vertaillut kehitysteeman puolesta omaa elämääni, ja erityisesti sitä miten parannan työtehokkuuttani, keskittymistäni ja millaisena näen vapaa-aikani ja tarpeellisen levon. Teoksessa avataan syvällisesti keskeisiä asioita ja ne ovat vaikuttaneet minuun sillä pohjalta että aion reflektoida omaa elämääni ja tekemistäni tarkemmin tulevaisuudessa.

Olen myös oppinut että on olemassa erilaisia ratkaisumalleja, esimerkkikoodeja, valmispohjia, videoita ja lukuisia websivuja, joiden avulla periaatteessa kuka tahansa voi ohjelmistokehityksen parissa työskennellä, ja käyttää näitä arvokkaita lähteitä hyödykseen.

Itselläni käytännössä ei ole koodaustaustaa ollenkaan, mutta olen joutunut projektin puitteissa opettelemaan erilaisia koodieditoreita kuten wsl, bash, powershell, ja azurecli ja kehittynyt niiden käytössä huimasti. Näiden asioiden opettelu on antanut minulle tietynlaisen ymmärryksen ohjelmistokehityksestä, ja ohjelmistokehityksen prosesseista. Tiedän, että minulla on vielä paljon opeteltavaa, mutta minulla on vahva usko siihen, että pystyn ammattimaisesti jonakin päivänä olemaan osana suuriakin projekteja.

Viikon aikana keskityin kuitenkin pääosin Microsoft Azuren puoleen, sillä olen ensisijaisesti Azuren asiantuntija. Osallistun kyllä myös kehitystyöhön sisäisten resurssien kehityksessä, mutta ensisijaisesti vastaan pilven infrastruktuuriin, ja sen rakentamisesta Terraformilla.

Sain kuitenkin Sentinelin asennettua kehitysympäristöihin, ja olin osana tiimin kanssa kehittämässä projektia eteenpäin. Saimme yhdessä klusterin sisäiset resurssit asennettua Hiekkalaatikkoympäristöön ja Testiympäristöön, sekä yhteydet toisiin sovelluksiin toimivat hyvin, ja datan käsittely toimii visuaalisella käyttöliittymällä Kibanalla oikein mallikkaasti.

Työni kannalta yhtenä osaamisalueena on Kubernetesin syvä asiantuntemus. Kubernetesin osaamisesta on hyötyä, sillä suurimmilta osin Kubernetes on yksi yleisin tapa toteuttaa ympäristöjä ohjelmistokehityksessä.

Container Journal niminen sivusto kertoo, että Docker sekä Kubernetes ovat avoimia pilvi-pohjaisia alustoja. Mutta näillä kuitenkin on hiuksenhieno ero. Näiden ero on, että Docker pakkaa konttisovellukset vain yhteen noodiin kerrallaan, kun taas Kubernetes ajaa niitä klusterissa. (Container Journal. 2022.)

Kehitysprojektin aikana olen huomannut kuinka myös IaC-työkalu kubectl joka on Kubernetesin sisäinen hallintatyökalu, miten se toimii ja olenkin saanut kehittyä sen käyttäjänä tämän projektin aikana. Kubernetes ehdottomasti on tietoperustan kaltainen tärkeä kehitystyökalu, ja sen laajempi osaaminen kuuluu kehittäjätyöhön sillä olen saanut omakohtaista kokemusta Kubernetesin käytöstä, ja esimerkiksi Azure Kubernetes Servicen älykkyys esiintyy nimenomaan siinä, että se osaa ajaa sovellukset eri noodeille mikä tasaa kuormaa ja pitää näin ollen klusterin toiminnan eheämpänä. Uskon että tämä puoltaa osakseen ohjelmistokehittäjien valinnassa, jos pitää luotettava ja toimiva ympäristö rakentaa. Käytännössä Kubernetes on hieman suurempien projektien ympäristöihin sopiva valinta, kun taas Docker löytää tiensä pienempiin projekteihin.

### 3.7 Viikko 13

*Maanantai 28.3.2022*

Viikko alkoi tehokkaasti projektin päivittäisellä aamupalaverilla. Tänään ei sinällään erikoisempia tavoitteita tullut asetettua, sillä päivä on aika paljon täynnä palavereita.

Kuitenkin otin projektitiimin aamupalaverissa puheeksi klusterin päivityskäytäntöjen ja huoltoikkunoiden määrittelyn, ja sen kuinka sen aiomme toteuttaa. Tarkoitus on ottaa käyttöön noodien uudelleenkäynnistämistä huolehtiva sovellus nimeltä Kured Reboot Daemon.

Kured on sovellus, joka helpottaa uudelleenkäynnistämään automaattisesti noodit klusterin päivityksen jälkeen. Käytännössä kun halutaan suojata klusteri, ovat tietoturvapäivitykset, OS, sekä kernelin päivitykset automaattisesti voimassa palveluntarjoajan puolesta, josta tässä tilanteessa huolehtii Microsoft. Kuitenkin päivitysten jälkeen noodit vaativat uudelleenkäynnistystyksen saadakseen päivitykset voimaan, ja tätä ei Microsoft tarjoa ja siihen on ihan looginen syy. Microsoft ei voi mitenkään tietää mihin aikaan voidaan uudelleenkäynnistää yksittäinen klusteri. Siksi apuun otetaan Kured.

Kuredilla voidaan noodien uudelleenkäynnistäminen automatisoida haluttuun huoltoikkunaan, ja varmistua näin ollen, että virallisena toimistoaikana ei ympäristö tule kaatumaan päivitysten vuoksi.

Päädyimme yhdessä tiimin kanssa kyseiseen valintaan, ja aloin selvittämään hieman enemmän tästä sovelluksesta. Päädyin testaamaan Kuredin hiekkalaatikkototeutusta omaan ympäristöön aluksi. Tänään sain asennettua ympäristöresurssit valmiiksi toteutusta varten. Kuredin testaus kuitenkin jäi vielä vähemmälle.

Kehityksen puolesta sain taas oppia jotakin uutta, kuten tämä Kured joka tuntuu olevan oikein hyvä ja validi tapa ottaa käyttöön päivitysten jälkeiset uudelleenkäynnistämiset noodeille.

*Tiistai 29.3.2022*

Aamun palaverissa päätimme tiimin kanssa käyttöönottaa tänään ulkoisen Nginx Ingress Controllerin käyttöön Kibanalle. Meillä on olemassa jo sisäiselle liikenteelle Ingress Controller, mutta Kibana tarvitsee ulkoisia yhteyksiä varten oman. Myös jos aikaa riittää, katsomme varmenteita parempaan kuntoon, sillä varmenteiden ketjutuksessa on jotakin ongelmaa. Ingress controllerin asennus saatiin otettua käyttöön, mutta edelleen varmenteissa on jotakin vikaa. Kibana löytää hienosti varmenteen ja käyttää sitä, mutta koska ketjussa on vikaa se ei näytä sivustoa täysin luotettuna. Varmenteen ketjua ei saatu tänään toimimaan niin että Kibana sen olisi oikealla tavalla tunnistanut. Ongelman selvitys jatkuu huomenna.

Kehitys oli ihan hyvä tänään, sillä tiimissä työskennellessä, kun joku tekee, ja muut ohjaa pystyy välillä itsekkin rauhasa perehtymään oikeasti siihen, miten asioita tehdään. Käytännössä asennamme sovellukset Helm charttien avulla kehitysympäristöön. Muokkaamme chartteja ympäristöämme vastaavan muotoon ja ajamme chartin putkella.

*Keskiviikko 30.3.2022*

Aamun palaverissa otettiin tähtäimeen tänään saada Kafkan sertifikaatit sekä Nginx default SSL Certificate käyttöönotettua. Kollega työsti sitä, ja muut seurasimme hänen tekevänsä, sillä muilla ei ollut osaamistaso tuohon tehtävään kovin hyvä.

Sertifikaatit saatiin onnistuneesti toimimaan kollegan toimesta, ja yhteydet ovat taas kunnossa. Itse perehdyin vielä Kuredin dokumentaatioihin, ja sen käyttöönottamisen suunnittelemiseen.

Myös etappi eli sprintti on päättymässä, ja kävimme sprintin aikaiset tehtävät läpi.

Olemme sprintin tehtyjen tehtävien perusteella hyvässä vauhdissa, ja olemme kirineet

projektin aikataulun takaisin kiinni. Voimme ottaa ensimmäiset tuotantojärjestelmät mukaan jo toukokuun alussa, ja näin ollen minä voin alkaa toteuttamaan tuotantoympäristön infraa pilveen.

Niemi A. ja Hietaniemi J. kirjan *Just Sopivasti Ketterä* pohjalta mietin erityisesti kehityksen kannalta koko Scrum mallia, ja totesin mallin sopivan minulle. Eihän minulla ole kokemusta muista kehitysmalleista, mutta Scrumissa on hyvä puoli sen suoraviivaisuudessa. Tarkoitus on tehdä pieniä, ja arvoa tuottavia asioita vähän kerrallaan, eikä heti rakentaa kerralla suurta.

Tällainen tekeminen sopii minulle ja on antanut ymmärtää sen, että osa kokonaisuutta ovat useat kehittäjät. Ja sen ajatusmalli rakentaa pieniä asioita kerralla, ei tuota niin suurta kuormaa kehittäjillekään, että välillä voi itse määritellä oman työajan puitteissa, miten töitä tekee.

Tämä malli on suonut minulle hieman lepoaikaa, joka taas heijastuu siihen omaan keskittymiseen ja hyvinvointiin työntekijänä mitä pohdinkin aiemmin. Olen alkanut ottamaan oman hyvinvoinnin tärkeäksi osa-alueeksi omaa jaksamistani ja pyrin siihen, että huomioin myös itseäni kesken päivän enemmän. Saatan käydä lenkillä ulkona, tai salilla ruokatauolla. Pidän myös päätteen äärestä sellaisia muutaman minuutin pituisia taukoja ja käyn jaloittelemassa, ja hakemassa lasin raikastavaa vettä. Olen nykyään pikkuisen jo hieman energisempi, ja keskittyminen ja työnteko alkaa tuntumaan paremmalta. Tämä on se suuri muutos, joka on nyt käynnissä itsessäni.

*Torstai 31.3.2022*

Tänään aamun tavoitteeksi asetin omaan Hiekkalaatikkoympäristööni Kuredin asentamisen, testaamisen sekä dokumentoinnin. Sain asennettua ohjelman, ja testasin sen toimintaa tekemällä AKS versiopäivityksen klusterille. Versiopäivityksessä AKS resurssilla on tietty versionumero, joka yleensä tulee oletuksena Microsoftin puolesta resurssia asentaessa. Kuitenkin versio on jo vanhempi versio, sillä uudempia versioita löytyy kyllä muutamia. Version päivityksen sain onnistuneesti läpi `az-cli` työkalulla, joka on siis Azuren oma työkalu kyseisten resurssien hallintaan. Päivitystä varten piti olla yhteys ensinnäkin Subscriptioniin, joka onnistuu kirjautumalla komennolla `az login`, jonka jälkeen klusterin sen hetkinen versionumero selvitettiin komennolla `az aks get-upgrades --resource-group myResourceGroup --name myAKSCluster --output table` jolloin tulokseksi saadaan sen hetkinen versionumero, sekä päivitettävät versionumerot mihin voidaan kyseisestä versiosta päivittää.

Lähdin päivittämään versiota komennolla `az aks upgrade --resource-group myResourceGroup --name myAKSCluster --kubernetes-version KUBERNETES_VERSION`.

Päivitys kesti noin kymmenen minuuttia, jonka jälkeen version numero oli vaihtunut uudempaan versioon. Kuitenkaan kured ei ollut toistaiseksi huomionnut päivitystä eikä näin ollen uudelleenkäynnistänyt noodeja. Hetken selvittämisen jälkeen selvisi, että sille oli myös looginen syy sillä en ollut kertonut konfiguraatioon käynnistymisen ajankohtaa.

Päivän kehitys tapahtui nimenomaan automaattisten sovellusten käyttöönoton helppoudesta ja siitä, miten ne asennetaan. Myös AKS:n manuaalisten päivitysten toteuttamisesta, ja sen vaiheiden oppimisesta. Hyvä päivä tänään.

### *Perjantai 1.4.2022*

Päivän tavoite on saada Kured toimimaan, ja selkeästi kertomaan noodien uudelleenkäynnistämisestä päivitysten jälkeen. Otin askeleen taaksepäin ja poistin Kuredin klusterilta, sillä halusin toteuttaa sen toisella tavalla. Testailin eri asennusvaihtoehtoja mutta Kuredin testaamista on sinällään hankala toteuttaa, sillä päivityksiä tulee yleensä iltaisin Microsoftin toimesta, ja jos Kured ei tunnista, että päivityksiä on olemassa, se ei myöskään käynnistä noodeja. Pitää odottaa siis sellaiseen määritettyyn ajankohtaan mihin se konfiguroidessa asetetaan.

Asetin Kuredille uudelleenkäynnistämistä varten ajan ja katson ensi viikolla kuinka sovellus toimii, ja onko uudelleenkäynnistämisiä tapahtunut.

### *Viikko 13 analyysi*

Vaikka viikko 13 ei tehtävällisesti ollut mitenkään raskas, mutta se oli kuitenkin sitäkin antoisampi. Pääsääntöisesti osaltani työviikon aikaa kului enemmän AKS-klusterin versiopäivityksien parissa, kuin myös Kubernetes Reboot Daemonin käyttöönottamisessa ja testauksissa.

Projektissa emme ole tiimin kanssa kovinkaan syvällisesti keskustelleet järjestelmän päivittämisestä, sekä huoltoikkunoiden suunnittelemisesta. Kävimme kuitenkin alustavaa keskustelua "Disaster Recovery" eli toisin sanoen katastrofin varalle tehdystä suunnitelmasta kuinka tilanteessa toimitaan.

Tähän otin avuksi pohdintaan kirjan nimeltä Azure Security Handbook, joka kertoo Azureen liittyvistä tietoturvaan vaikuttavista tekijöistä, uhkista, ja erityisesti niihin varautumisesta.

Otsin (2021, 12–13) teoksessa Azure Security Handbook todetaan että pilviturvallisuus, ja pilven tietoturva ei ole niin yksinkertaista nykypäivänä. Tietoturvaan vaikuttaa monia tärkeitä suojaavia osa-alueita kuten identiteetin ja pääsynhallinta, kirjautuminen ja valvonta, sekä tietoverkkojen tietoturva. Myös erilaisten työkuormien, kuten dataan ja pilvipalveluiden kuten esimerkiksi PaaS-palveluun liittyvän tietoturvan huomiointi on ensisijaisen tärkeää siltä osin missä erilaiset salaisuudet ja avaimet ovat säilössä. Kuinka pääsynhallintaa on rajoitettu sellaisiin paikkoihin, jota voidaan valvoa, ja missä on erillinen pääsynhallinnalle olemassa oleva ryhmä, jota voidaan hallita. Kuka saa lukea dataa, kuka saa muokata sitä. Käyttöoikeusryhmiä on useita, joiden avulla hallitaan käyttäjien pääsyjä erilaisiin palveluihin ja resursseihin.

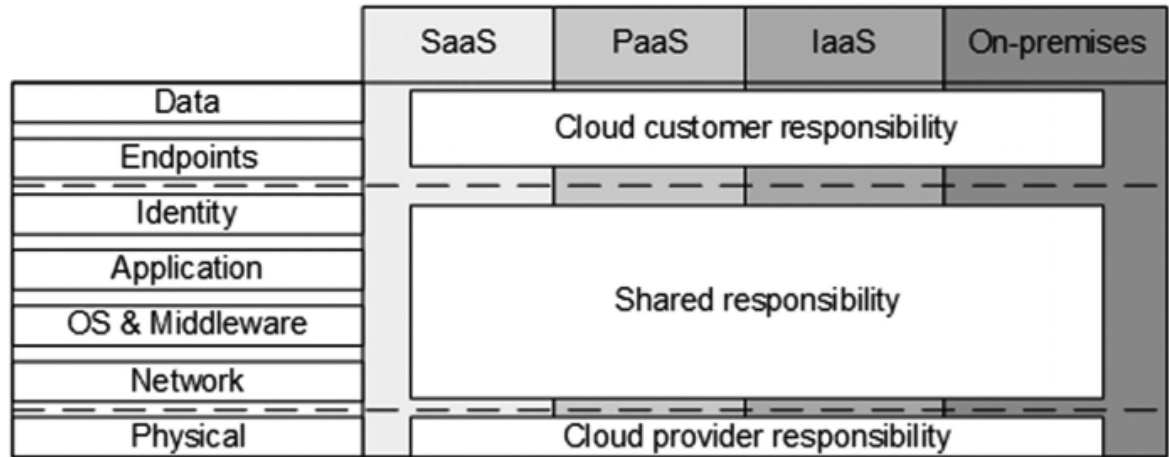
Pohdin tätä kaikkea ja samaistun siihen, että tietoturva on tänä päivänä kaikkien vastuulla, ei ainoastaan tietoturva-asiantuntijoiden vastuulla. Mielestäni korkeimpaan rooliin tietoturvassa nousee organisaation oma tietoturvapolitiikka. Yritysten tietoturvapolitiikka on yksi merkittävimpiä seikkoja sen tietoturvassa ja siinä, kuinka se sovitetaan pilvipalveluiden tietoturvan kanssa yhteen. Pilvipalvelu ei välttämättä tarjoa yhteensopivaa mallia yrityksen oman tietoturvapolitiikan kanssa, mutta se että kykeneekö yritys muuttamaan omaa tietoturvapolitiikkaa pilvipalvelun kanssa yhteensopivaksi, on haasteellista.

Käytettäköön esimerkkinä vaikkapa sellaista missä yritykseltä löytyy On-Premise ympäristö, eli yrityksellä on oma palvelinsali, omat palvelimet ja tietoverkot käytössään, ja se haluaa viedä koko IT-ympäristönsä infran Azureen, tulee vastaan erilaisia kysymyksiä resursseihin, tietoverkkoihin, erilaisiin hankintoihin, kuin myös palvelutasosopimuksen ehtoihin liittyvissä kysymyksissä. Usein saattaa olla kysymyksenä esimerkiksi vastuullisuuteen liittyviä kysymyksiä kuten kuka huolehtii laitteiden fyysisestä puolesta kuten rikkoontumisista ja niiden päivityksistä, kuin myös käyttöjärjestelmien päivityksistä, sekä huoltotoimenpiteistä. Tietoverkon osalta ajatellaan, kenen vastuulla tietoverkon toiminta on, kenen vastuulla OS:n eli Operating System toiminta on, erilaisten sovellusten toiminta, identiteettien hallinta, kuin myös datan hallinta ja sen turvallinen varastointi.

Tämä kaikki on saattanut olla yksittäisen yrityksen oman asiantuntevan henkilön tai kokonaisen osaston hallinnoimaa, ja se halutaan pitää sellaisena.

Tällaiseen vastuullisuusmalliin ei monikaan pilvipalveluntarjoaja kuitenkaan suostu, sillä pilvipalvelujen käyttöön on omat ehtonsa ja mallinsa, kuinka niitä asiakkaille tuotetaan ja

kuinka niistä vastataan. Tällaisessa tilanteessa tulee esiin jaettu vastuullisuusmalli, minkä pilvipalveluntarjoaja yleensä käyttäjälleen tarjoaa. Kuvasta 7 näkee pilvipalvelumallien vastuumallin eri pilvipalvelumalleissa.



Kuva 7. Pilvitietoturvan yhteisvastuumalli (Ots 2021, 3)

Fyysisen laitetietoturvan ja vastuun kantaa aina ensisijaisesti pilvipalveluntarjoaja. Pilvipalveluntarjoajan konesaleissa olevien fyysisten palvelimien ja laitteiden huollot, päivitykset ja korjaukset kuuluvat pilvipalveluntarjoajalle.

Tietoverkkojen osalta organisaatio itse, kuin myös pilvipalveluntarjoaja huolehtii tietoturvasta ja resurssien toiminnasta yhteisesti. Pilvipalveluntarjoaja tarjoaa verkkokortin ja julkisen verkkoyhteyden, mutta myös organisaation itse on määriteltävä omat sisäiset verkot sillä pilvipalveluntarjoaja ei tarjoa tätä.

Käyttöliittymän, sovellusten kuin myös identiteetinhallinnan osalta jaettu vastuumalli kuuluu molemmille osapuolille. Azure tarjoaa siis alustan kaikille resursseille, ja mahdollisuuksille toteuttaa laadukasta tietoturvaa kyseisten resurssien puolesta.

Datan osalta pilvipalveluntuottaja ei ota vastuuta, eli on organisaation oma asia huolehtia datasta missä se on. Vastuullisuusmalli pätee myös hybriditoteutuksissa sillä pilvipalvelua ei tarvitse ottaa kokonaisuena, vaan voidaan toteuttaa tällainen malli missä asiakkaalla ja pilvipalvelun tuottajalla on yhteinen ympäristö. Asiakkaalla saa siis olla oma palvelin käytössä, ja omat sovellukset mutta joitain resursseja otetaan käyttöön pilvipalveluntuottajan toimesta, ja yleensä toteutetaan tällaisena hybridiympäristönä. Esimerkiksi O365-palvelut voidaan yhdistää asiakkaan palvelimeen, ja näin ollen asiakas voi hoitaa pääsyn, sekä identiteetinhallintaa edelleen omien palvelimien kautta, mutta kyseiset tiedot synkronoituvat Azure Active Directoryyn eli Azuren käyttäjähakemistoon.

Miettiessäni asiakasprojekti X:n tietoturvaa, olemme todellakin tehneet kehitysjärjestelmästä sellaisen, missä olemme miettineet kaikkea tietoturvaan liittyviä kysymyksiä, kuten yleinen pääsynhallinta, erilaisten salaisuuksien säilytys ja niiden käyttöoikeuksien hallinta. Tietoverkkojen pääsynhallinta siinä, kuka pääsee mistä ja mihinkin ovat äärimmäisen tarkasti määriteltynä, sekä sovellusten tietoturva kuten sertifikaattien hallinta kuin myös pääsynhallinta klusterin sisäisiin podeihin on toteutettu todella tarkasti.

Kehitysjärjestelmään ei ole auki mitään ulkoisesta verkosta, ja hallintaan käytetään niin asiakkaan omaa palomuuria, kuin myös Network Security Group resurssia missä hallitsemme sisäisen verkkoliikenteen pääsynhallintaan liittyviä sääntöjä.

Katastrofin varalle tehtyä suunnitelmaa emme ole vielä tehnyt, mutta käytännössä järjestelmä on rakennettu siten, että sovelluksilla on replikoitu ympäristö olemassa, jos esimerkiksi klusterin pod menee rikki, tai muuten jotenkin jumiin. Voimme sammuttaa podeja huoletta, koska niiden sammuttaminen rakentaa tilalle uuden replikaation, ennen kuin se kytkee vanhan podin pois. Tämä jo itsessään takaa meille hieman turvaa sen suhteen, että toiminta jatkuu siitä huolimatta vaikka podit olisivat sammuneet jostain syystä.

Mutta yhteiseen pohdintaan menee kuitenkin se, että millaisia malleja katastrofin varalta olisi hyvä käydä läpi, ja kuinka kykenemme vastaamaan erilaisiin tapahtumiin ja uhkiin. Katastrofeja ei voi ennustaa, mutta niihin voidaan varautua käymällä tarkkaan tietoturva, ja erilaiset skenaariot läpi.

### **3.8 Viikko 14**

*Maanantai 4.4.2022*

Päivä alkoi totutusti projektin päivittäisellä palaverilla. Nopeasti tuli ilmi kahden kollegamme sairastapaukset, joten kaksi kehittäjää vähemmän tälle viikolle projektissa mukana. Projekti X on sellaisessa tilanteessa, että se on vakaa ja hyvällä mallilla eli sen suhteen ei ole ongelmaa. Olemme saaneet yhden asiakasjärjestelmän jo liitettyä mukaan testiympäristöön, ja kaikki toimii niin kuin pitääkin.

Kuitenkin päätin jatkaa Kuredin parissa tänäänkin, sillä tiimin yhteisiä kehityskohteita ei ollut tälle päivälle. Kured ei ollut viikonlopun aikana uudelleenkäynnistänyt klusterin noodeja, mikä aiheutti ihmetystä. Testailin päivän aikana hieman sen toimintaa mutta tuloksetta.

Päivä koostui myös monesta palaverista tänään, joten kehitystä ei tapahtunut suuremmilta osin ollenkaan.

#### *Tiistai 5.4.2022*

Päivän palaverissa tutustuttiin yhdessä Azuren resurssien monitorointiin liittyvien Azure Advisor palvelun toimintaan. Azure Advisor on Azuren oma valvontapalvelu, joka huomaa resurssiin kohdistuvat tietoturvaluutteen, mahdolliset asentamisessa tapahtuvat virheet tai ongelmat. Se analysoi resurssien tilaa, ja osaa tarjota käyttökelpoisia ratkaisuja puutteellisten resurssien tueksi. Näin ollen se korjaa resurssien tietoturvaa entisestään.

Edelleen jatkoin myös Kuredin testailua, mutta tällä kertaa asensin Kuredin projektin Hiek-kalaatikkoympäristöön. Sain otettua sen käyttöön ongelmitta, sillä sain kollegalta toisen asiakasjärjestelmän vastaavan toteutuksen konfiguraation, minkä muokkasin sen sopivaksi tälle ympäristölle. Konfiguraation pohjalta automatisoin noodien käynnistymisen maanantain sekä torstain aamuyölle mikäli tietoturva, - tai muita ominaisuuspäivityksiä on saatavissa. Kehityin erityisesti tänään Kuredin konfiguraatioissa, sekä sovelluksen testaamisessa kun sain ajastettua noodien uudelleenkäynnistymisen onnistuneesti.

#### *Keskiviikko 6.4.2022*

Tavoitteeksi asetin tänään jatkaa Kuredin parissa, hieman vaihtelevasti kuitenkin. Aloin selvittämään sovelluksen dokumentaatioita ja tarkastelemaan voisinko demota noodien uudelleenkäynnistymistä Kuredin avulla. Periaatteessa tämä on mahdollista, mutta en saanut kuitenkaan toteutusta onnistuneesti tehtyä. Epäily jäi siltä osin siihen, että noodit kuitenkin ovat ajan tasalla eikä päivityksiä vain kerta kaikkiaan ole saatavissa. Tuo syy voisi olla se, miksi uudelleenkäynnistämistä en havainnut ollenkaan.

Tänään myös aloitettiin asiakasprojekti X:n virallinen dokumentointi. Dokumentointi on yksi tärkeimmistä osa-alueista jatkuvan palvelun takaamiseksi. Dokumentaatiot sisältävät kaiken mahdollisen projektista aina alkaen kuvauksista ja toteutustavoista järjestelmän ylläpitoon saakka. Dokumentoitavaa on todella paljon, ja sitä tulen tekemään seuraavilla viikolla paljon suhteessa kehitysohjelmaan, koska järjestelmä on aika pitkälle valmis suurimmilta osin.

Kehityin tänään asiakkaan dokumentaatioalustan käytössä, joka oli hieman tuttu minulle, mutta sen käyttö olikin astetta haastavampaa, sillä dokumentoinnissa olikin haasteita. Dokumentointialustan nimeä en voi paljastaa, mutta dokumenttien kirjoittamisessa kyseisen alustan tekstieditori ei rivittänyt ihan normaaliin tapaan mikä aiheutti hämmennystä.

Alustalle dokumentoinnin oppiminen onkin kehitysvoitto itselleni, ja helpottaa jatkodokumentoinnin työskentelyä huomattavasti.

*Torstai 7.4.2022*

Tavoitteeksi tänään katson Kuredin toimintaa, ja sitä että onko se käynnistänyt noodit päivityksien jälkeen. Virallisesti noodit eivät näyttäneet päivittyneiltä, mutta Kuredin podit antoivat kyllä ymmärtää että jotain on tapahtunut edellisyönä sillä Kured on laittanut resursseja lukkoon, ja vapauttanut ne viiden minuutin sisällä lukitsemisesta. Tämä automaatio oli siis määriteltynä konfiguraatiossa näin toimivaksi.

Erilaiset kehittäjille suunnatut sivustot kertovat Kuredin loistavasta automaatiosta ja siitä kuinka hyvin se osaa huolehtia klusterin noodien uudelleenkäynnistymisestä. Tarkistaessani Kuredin toimintaa klusterissa, jotain tosiaan oli tapahtunut ja se on tulkittavissa kuvan 8 tavalla.

```
time="2022-04-06T22:14:31Z" level=info msg="Binding node-id command flag to environment variable: KURED_NODE_ID"
time="2022-04-06T22:14:31Z" level=info msg="Kubernetes Reboot Daemon: 1.9.1"
time="2022-04-06T22:14:31Z" level=info msg="Node ID: k8s-node-1"
time="2022-04-06T22:14:31Z" level=info msg="Lock Annotation: kured/kured:weave.works/kured-node-lock"
time="2022-04-06T22:14:31Z" level=info msg="Lock TTL not set, lock will remain until being released"
time="2022-04-06T22:14:31Z" level=info msg="Lock release delay set, lock release will be delayed by: 5m0s"
time="2022-04-06T22:14:31Z" level=info msg="PreferNoSchedule taint: "
time="2022-04-06T22:14:31Z" level=info msg="Blocking Pod Selectors: []"
time="2022-04-06T22:14:31Z" level=info msg="Reboot schedule: ---Mon-----Thu----- between 00:00 and 05:00 Europe/Helsinki"
time="2022-04-06T22:14:31Z" level=info msg="Reboot check command: [test -f /var/run/reboot-required] every 1h0m0s"
time="2022-04-06T22:14:31Z" level=info msg="Reboot command: [/bin/systemctl reboot]"
time="2022-04-06T22:15:06Z" level=info msg="Holding lock"
time="2022-04-06T22:15:06Z" level=info msg="Uncordoning node k8s-node-1"
time="2022-04-06T22:15:06Z" level=info msg="Delaying lock release by 5m0s"
time="2022-04-06T22:20:06Z" level=info msg="Releasing lock"
time="2022-04-06T23:04:03Z" level=info msg="Reboot not required"
```

Kuva 8. Kuredin automatisoitu toiminto

Lokien perusteella Kured on lukinnut noodin, ja ilmoittaa että lukko aukeaa vasta kun viisi minuuttia on kulunut. Tämän perusteella Kured toimii niin kuin sen olin asettanut toimivaksi. Kuitenkin ihmetyttää että uudelleenkäynnistämistä ei tarvittu. Yleensä uudelleenkäynnistäminen on suositeltavaa, sillä kaikki liittymät eivät osaa ottaa päivityksiä käyttöön ennen kuin se on käynnistetty uudelleen. Tässä on vielä minulla selvitettävää mitä on tapahtunut.

Myös tänään kirjoittelin kehitysympäristön dokumentaatiota kuntoon pikkuisen, ja kävimme kollegojen kanssa keskustelua kehitysympäristön rakentamiseen liittyvistä asioista kuten muiden tuotantoympäristöjen mukaan ottamisen tähän järjestelmään.

*Perjantai 8.4.2022*

Tänään tavoitteeksi otin asentaa Tuotantoympäristön pystyyn, ja määritellä koodit kuntoon Terraformissa.

Sain koodit kuntoon, ja muutamalla ajolla ympäristön pystyyn. Tein hienosäätöä kuitenkin vielä muuttujiin ja lokaaleihin, jonka jälkeen ympäristö meni onnistuneesti läpi.

Kuitenkin jäi vielä selvitettävää sillä salaisuudet eivät näkyneet ollenkaan klusterille. Loo-ginen syy on käyttöoikeuksissa, sillä klusterilla ei ollut käyttöoikeutta hakea oikeita avaimia keyvaultista. Tämän tulen korjaamaan ensi viikolla, sillä loppupäivä meni dokumentoidessa päivän tekemisiäni.

Oma kehitys parani, sillä tuotantoympäristön ajo pilveen vaatii tarkkaa ymmärtämistä, sekä koodin puolesta mutta myös tietoturvan puolesta. Tuotantoympäristö on sellainen, mihin käytännössä ei saa minkäänlaisia salaisuuksia julkisesti nähtäville mennä, eikä erillisiä IP-osoitteita tai muitakaan henkilötietoja. Tämän vuoksi kaikki piti ottaa huomioon, ja putken Terraform suunnitelmat piti lukea huolellisesti läpi ettei resurssien ajossa pilveen ole mitään salaisuuksiin tai avaimiin viittaavaa tietoa.

*Viikko 14 analyysi*

Vaikka kehitysympäristöt ovat hyvällä mallilla, kahden tekijän poissaolo vaikuttaa projektiin työmäärällisesti. Itse sain rauhassa tehdä ja perehtyä erityisesti Kurediin, joka ehkä on vieläkin hieman avoinna itselleni sen toiminnan suhteen. Myös yksi suurista onnistumisista oli tuotannon kehitysympäristön käyttöönotto vaikka salaisuuksien hakemisessa olikin vielä hieman haasteita. Pääsin konfiguroimaan Terraform koodit kuntoon, ja ajamaan putkella kehitysympäristöön. Onnistumisen tunne oli mahtava kun seurasin kehitysympäristön ajoa.

Pohdin kaikkea koodaamaani, tekemääni, ja näkemääni suhteessa viimeiseen kahteen vuoteen enkä ole ollut näin innoissani sinä aikana ammatillisen kehittymisen, mutta myös tekemisen suhteen lainkaan. Tämä aika ohjelmistokehittäjänä on saanut minut innostumaan erilaisista projekteista, uusista hankkeista ja muista töistä joiden parissa saan työskennellä. Tämä on avannut silmiäni entisestä, sillä aiemmin IT-alan alkuvaiheessa olin sitä mieltä että ohjelmistokehityksen puolelle en tule ikinä vaihtamaan, mutta tässä sitä ollaan. Vaihdos on ollut yksi viisaimpia päätöksiäni alkaa tekemään jotain sellaista mistä innostun, ja mitä haluan tehdä jatkossakin.

Pohdin myös tulevaa dokumentointiurakkaa joka on aloitettu, ja sitä mitä kaikkea dokumentoinnissa tulisi ottaa huomioon.

Lampinen H. pro gradu tutkielmassaan teknisen dokumentaation haasteet ketterässä järjestelmäkehityksessä, toteaa että dokumentaatio on tallenne käyttäjän tarpeista järjestelmään ja vastaa järjestelmän käyttöön ja sen toimintaan liittyviin tarpeisiin. Se koostuu laajasta skaalasta erityyppistä dataa.

Järjestelmädokumentaation tulisi vastata tarpeisiin:

- se on sopimuksenmukainen
- se auttaa kehitystiimiä ymmärtämään toiminnallisuudet jotka kehitetään
- se mahdollistaa kehitystiimin kommunikaation tulevalle ylläpitotiimille

Dokumentaation tarkoitus ei ole ainoastaan auttaa lukijaansa ymmärtämään järjestelmää, vaan myös vähentää koulutuksen ja järjestelmätuen tarvetta sekä osaltaan parantaa tuotteen yleistä mainetta. (Lampinen 2019, 19–20)

Samaistun Lampisen toteamukseen siltä osin, että se vähentää koulutuksen ja järjestelmätuen tarvetta, mutta korostaisin sen olevan myös organisaation järjestelmää käyttävien henkilöiden tietovaranto käytettävässä järjestelmässä toimimiseksi. Dokumentaatiot opettavat myös muita henkilöitä käyttämään järjestelmää oikeaoppisesti, ja auttavat myös häätälanteessa todella pitkälle ottamaan esimerkiksi oikeisiin kohdehenkilöihin yhteyttä.

### **3.9 Viikko 15**

*Maanantai 11.4.2022*

Tänään tavoitteeksi asetettiin asentaa Tuotantoympäristön salaisuudet, sekä varmistaa klusterin noodien toiminta. Jos aikaa jää, jatkan projektin dokumentointia.

Työpäivän kulku eteni palaverimaisesti pitkin päivää, mutta päivän työpajassa tiimin kanssa yhdessä saimme Tuotantoympäristön salaisuudet toimimaan AKS:n kanssa niin että se noutaa salaisuudet Azure Keyvaultista. Nyt klusterin podit saavat oikeat salaisuudet, ja sovellukset toimivat odotetulla tavalla. Pääsin myös dokumentoimaan projektin kehityksen osa-alueita liittyen työkaluihin, ja niiden käyttöön.

Kehityin erityisesti päivän aikana salaisuuksien käsittelyssä, ja niiden käyttöönotossa klusterille. Myös Tuotantoympäristön konfiguraatioiden puolesta sain lisäkehitystä, sillä varmistelin niiden toiminnan, kuin myös kehityspotkien toiminnan Azure Devopsissa.

*Tiistai 12.4.2022*

Päiväpalaverin jälkeen otin tavoitteeksi korjailla Tuotantoympäristön oikeuksia sekä keskittyä korjaamaan hieman podien toimintaa sillä aamulla huomasimme että Elasticsearch podit olivat varoitustilassa, ja sille löytyi looginen virhe. Myös projektin seuraavan vaiheen palaverin läpikäynti ja suunnittelu, kuinka projekti jatkuu tämän kehitysprojektin jälkeen.

Saimme päivän työpajassa kollegojen kanssa yhdessä selvitettyä Elasticsearchin virheen, mikä johtui Ingress Controllerin tunnistamisesta. Elasticsearch ei siis tunnistanut Ingress Controllerin IP-osoitetta minkä vuoksi se oli virhetilassa. Virheen korjauksen, sekä podien uudelleenkäynnistyksen jälkeen liikenne toimi taas normaalisti, ja varmistimme että virhetilanteita ei enempää löytynyt järjestelmästä.

Päivän kehitys tapahtui erityisesti siinä, kuinka loogisesti kokeillaan erilaisia toimenpiteitä virheen korjauksen myötä. Tällaisessa ympäristössä voi sammutella ja poistaa sovelluksen podeja jos tahtoo, koska podit ovat replikoituja ja nousevat aina uudelleen pystyyn.

*Keskiviikko 13.4.2022*

Asetin itselleni päivän tavoitteeksi testata klusterin sekä sen nooidien versiopäivitysten toimivuutta asiakaspilvessä. Nyt päivitys tulee olemaan erilainen koska aiempi päivitys tehtiin omassa pilviympäristössä. Jos aikaa jää, saatan myös toteuttaa päivityskäytäntöjen dokumentoinnin asiakasalustalle.

Otin kohteeksi asiakaspilven Hiekkalaatikkoympäristön sekä Testiympäristön päivitykset ensimmäisenä. Kubernetes versio 1.21.7 ei ole enää Microsoftin puolesta tuettu, ja resurssin asentaminen kuitenkin oletuksena tulee tuolla versiolla, joka käytännössä on tässäkin projektissa tullut jokaiseen kehitysympäristöön.

Työkaluksi valikoitui Powershell, sillä se sisältää tarvittavan Az-työkalun millä päivitys käytännössä tehdään. Komennot päivitykselle ovat samanlaiset mitä aiemmassa vaiheessa jo kuvattiin.

Ensiksi yhdistin editorin asiakkaan pilveen jälleen komennolla az login, ja sitten tarkistin kehitysympäristön mikä oletuksena kirjautumisen jälkeen tulee. Huomasin että kehitysympäristö oli väärä, ja sen vaihtaminen onnistui komennolla az account set -s "Subscriptionin nimi".

Kun kehitysympäristö oli oikea, hain AKS:lle versiopäivitykset Microsoftin suositusten mukaisesti komennolla `az aks get-upgrades --resource-group myResourceGroup --name myAKSCluster --output table` ja sain listattuna klusterin nykyisen version, sekä mahdolliset versiot jotka voidaan päivittää käyttöön. Päivitin klusterin version kerrallaan komennolla `az aks upgrade --resource-group myResourceGroup --name myAKSCluster --kubernetes-version KUBERNETES_VERSION`, sillä projektin alkuvaiheessa päivitin Azuren portaalin kautta suoraan uusimpaan, ja klusteri meni vikatilaan emmekä saaneet palautettua sitä tuolloin.

Nyt päivitin yhden version kerrallaan molempiin kehitysympäristöihin ja päivitykset onnistuivat hyvin. Päivitysten kesto yllätti minut, sillä versioita oli useampia molempien kehitysympäristöjen klustereille, ja se ottikin koko päivän päivitellä oikeat versiot käyttöön.

Oma kehitys parani kun toteutin päivityskäytäntöjä, sillä toteutus on helppo kuvatuilla komennolla, eikä minkäänlaisia ongelmia syntynyt.

*Torstai 14.4.2022*

Päivän tavoitteeksi otimme projektitiimin kanssa yhteisen työpajan, missä kollegani yritti testipuolen kehitysympäristölle saada Kafkan käyttämään omaa Ingress Controlleria. Oma tavoitteeni oli saada oman komentorivin yhdistäminen podille toimintakuntoon, sillä kun menin katsomaan klusterin sisäistä podia, minkä jälkeen yritin kirjautua podiin sisälle, niin komentorivi ilmoitti että yhteys palvelimelle "localhost" hylättiin.

Vianetsintä ei ollut helppoa, ja kesti yli puoli päivää saada selville miksi en pystynyt yhdistämään podiin. Kubernetes tarvitsee Docker Desktopin toimiakseen oikein, ja minun koneellani kyseinen sovellus olikin ollut vikatilassa hetken aikaa. Vanhan version poistaminen tiedostokansioita myöten, sekä uuden sovelluksen asentaminen auttoi ongelmaan ja pääsin kirjautumaan klusterin podille.

Päivän kehitys eteni ongelmanratkaisutyypisest selvitäessäni koneellani ollutta vikaa. Opetus on se, että välillä kehitystyössäkin saattaa työkalut lakata toimimasta, minkä jälkeen sovellusten toiminta on vajaata ja se vaikuttaa myös oman työn tekemiseen.

*Perjantai 15.4.2022*

Pitkäperjantai, eli on vapaapäivä.

### *Viikko 15 analyysi*

Viikko oli normaalista poikkeava, sillä erityisen haasteellisia ongelmia projektilla ei ollut, minkä vuoksi tiimissä teimme yhteisesti joitakin työtehtäviä ja myös saimme keskittyä omien työtehtävien tekemiseen.

Kehityin erityisesti AKS:n päivityskäytännöissä, sillä aiempi epäonnistunut kokemus nosti päätään ja tekijä ei ihan varma ollut päivityksen onnistumisesta. Kuitenkin päivitykset onnistuivat hienosti, ja antoi itsevarmuutta ohjelmistokehityksen urallani. Päivitysten apuna käytin Microsoftin laatimaa ohjetta, minkä avulla sain komennot päivityksen tueksi. (Microsoft Corporation 2022)

Olemme hyvässä vauhdissa kohti tuotantoympäristön käyttöönottoa, ja joitain korjauksia vielä kehitysympäristöön tehdään. Kokonaisuutena olen huomannut omassa tekemisessä parempaa keskittymistä, sillä olen pienillä muutoksilla tehnyt elämänmuutosta.

Teoksen *Just Sopivasti ketterä – ketterällä kehityksellä kohti parempaa elämää mukaan haaveilemalla ja visualisoimalla* niin mennyttä kuin myös tulevaa voidaan saada aikaan tehokkaampaa keskittymistä. Visualisoinnissa, eli mentaaliharjoittelussa käydään läpi tietoisesti jotakin tapahtumaa, tehtävää tai tilannetta. Visualisointia voidaan hyödyntää esimerkiksi tapahtumaa, tai vierasta tilaisuutta valmistautumiseen varten.

Mentaaliharjoittelua käytetään itseluottamuksen ja stressin tukemiseen. Ajatuksena mentaaliharjoittelussa on käydä hidastetusti läpi jokin tapahtuma. Ennen stressaavaa tilannetta voidaan visualisoida kyseinen mennyt, tai tuleva tapahtuma suoritukseksi ja voidaan kuvitella kuinka tapahtuman hoitaisi itse. Se lievittää stressiä, ja antaa kokonaiskuvan siitä tapahtumasta miten itse sen hoitaisi. Mentaaliharjoittelu laukaisee myös jännitystä, ja rentouttaa henkilön tilaan, kuinka hän tapahtuman itsenäisesti näkee, ja kokee. Mentaaliharjoittelun aikana voi motivoida itsensä, sekä auttaa itseään jaksamaan pitkällä polulla.

Voidaan miettiä omaa kokemusta ja kehitystä uran ajalta, sekä ajatella tulevaa matkaa ja kaikkea sitä mielekästä tekemistä mikä on vielä edessä. Visualisointiharjoitteen tavoitteena on kokea tilanne positiivisessa valossa, ja auttaa henkilöä kehittymään paremmaksi versioksi itsestään.

Tarkoituksena on antaa aivojen tehdä työtä keskittymisen parantamista varten. Keskittymisen myötä aivot oppivat parhaiten mielikuvista ja auttavat henkilöä pidemmällä ajalla luonnollisemmin miettimään asioita. Mentaaliharjoittelua voi myös tehostaa esimerkiksi puhumalla itselleen positiivisesti, ja vakuuttamalla itselleen mielikuvansa perusteella

läpikäydystä tapahtumasta kuinka sen näkee, ja kokee. Itsensä tsemppaaminen tehostaa visualisointikokemusta entisestään ja parantaa omaa luonnollista reaktiota stressaavan tilanteen hallintaan. (Niemi & Hietaniemi 2020, 93–96)

Samaistun teoksessa kerrottuun väitteeseen siitä, että mentaaliharjoittelulla voidaan parantaa omia kokemuksia tulevasta, sekä samalla visualisoida menneitä aikoja luonnollisemmiksi itselleen. On mielestäni erittäin tärkeää että oma mielikuva tekemisestä on mieltä rauhoittavaa, ja stressiä poistavaa eikä harteilla oleva taakka painaisi niin paljon. Yleensä stressi kasvaa tulevasta tapahtumasta, kuten esimerkkinä se, kuinka ohjelmistokehityksen projektissa olemme tulossa Tuotannon käyttöönottovaiheeseen. Sellaisessa tilanteessa paineet alkavat kasautumaan, ja se myös näkyy kehittäjien työn tekemisessä.

Päädyn tämän mietinnän ja harkinnan jälkeen ehdottomasti kokeilemaan mentaaliharjoittelua, ja tekemään itsesuggestiota tulevasta. Pyrin rauhoittumaan, ja käyn mielikuvina kyseisen päivän läpi moneen kertaan. Tsemppaan itseäni, ja pyrin tekemään siitä päivästä itselleni mieluisan, sekä käymään erilaiset skenaariot päivän kulusta läpi etukäteen. Uskon että tällaisella visualisoinnilla autan omaa mieltäni rauhoittumaan paremmin, sekä toivon erityisesti oman keskittymisen parantuvan ja näkyvän jatkossakin.

### **3.10 Viikko 16**

*Maanantai 18.4.2022*

Vapaapäivä, eli toinen pääsiäispäivä.

*Tiistai 19.4.2022*

Tavoitteita päiväkirjaraportoinnin osalta en pysty enää tuottamaan paljoakaan, sillä olen työstä vapaalla tämän viikon opiskelujen tähden. Minulla on hyvää aikaa perehtyä opintoihini, sekä saattaa loppuun kaikki opinnot ennen valmistumista.

*Keskiviikko 20.4.2022*

Vapaapäivä.

*Torstai 21.4.2022*

Vapaapäivä.

*Perjantai 22.4.2022*

Opinnäytetyön viimeinen raportoitava päiväkirjamerkintä. Tänään tavoitteeksi asetin itseleni keskittyä Velero nimisen sovelluksen asennukseen Hiekkalaatikkoympäristössä.

Velero on avoimen lähdekoodin työkalu Kubernetes-klusterien ja niiden taltioiden turvalliseen varmuuskopiointiin, palauttamiseen ja siirtämiseen. Velero toimii komentoriviliittymällä ja sillä voidaan määritellä ajoitetut varmuuskopiot, sekä ajastaa varmuuskopioiden ottaminen klusterista esimerkiksi Azuren File Shareen. Käytännössä Velero on puhdas varmuuskopiontien ottamiseen ja palauttamiseen tarkoitettu sovellus.

Azuren File Share on Microsoft Azuren tiedostonjakopalvelu, joka on rinnastettavissa esimerkiksi tietokoneiden väliseen tiedostonjaon siirtoon.

Aiemmasta kokemuksestani Veleron parissa, on työkalun käyttö minulle hyvin tuttua. Sovellus on helppokäyttöinen, ja monipuolinen, sekä sillä voidaan ottaa klusterin sisäisten resurssien varmuuskopiot ihan mistä resurssista tahansa. Sillä voidaan myös tehdä kokonaisen klusterin migraatio esimerkiksi toiselle klusterille tarpeen vaatiessa. Näin ollen voidaan vaikkapa toteuttaa kahdennettu klusteripohjainen järjestelmä, sillä kun toinen klusteri vikaantuu, tulee automaattisesti toinen klusteri hoitamaan ensimmäisen klusterin virkaa. Klusterit ovat synkronisaatiossa toistensa kanssa koko ajan.

Aiemmin olin asentanut Veleron manuaalisena asennuksena klusterille ja se toimi hienosti, mutta haasteita tuli tällä kertaa eteen erityisesti sen puolesta, että sovelluksen käyttöönotto pitää toteuttaa nyt Terraformilla Helm toteutuksena, eikä Helm oikein tykkää asentua kovinkaan hyvin klusterille. Ongelmia myös esiintyi työkalun käyttämisen credentials-tiedoston hakemisessa, ja siinä kuinka se tunnistaa tarpeelliset salaisuudet tiedostosta.

En saanut tällä kertaa vikaa korjattua, ja päätin jatkaa vianselvitystä seuraavalla kehitysviikolla. Kehitys oli pienimuotoista koska vaikka en saanut sitä toimimaan, sain kuitenkin konfiguraatiot määriteltyä.

*Viikko 10 analyysi*

Viikko on ollut poikkeuksellinen, sillä olin koko viikon vapaalla paitsi perjantaina. Tarkoituksena oli edistää opintojani viikon aikana, sekä myös saada itselleni sellaista miettimis-, kuin myös rauhoittumisaikaa tulevaa Tuotantoympäristön käyttöönottoa varten.

Olen toteuttanut Just Sopivasti ketterä – ketterällä kehityksellä kohti parempaa elämää mukaista mentaaliharjoittelua ja käynyt mielessäni kyseisen päivän tapahtumat, jotta osaan avoimemmalla mielellä perehtyä niihin asioihin joita ammattitaitoni edellyttää, sekä kykenen tehostamaan omaa keskittymistäni työn suhteen. Olen käynyt mielessäni aamu-päivän, aina iltapäivään asti kuinka varaudun ongelmatilanteisiin ja ketkä yhteyshenkilöt minulla ovat käytettävissä tuon päivän aikana. Ennaltaehkäisevä visualisointi auttaa minua rauhoittumaan, sekä olemaan ammattimaisempi palvelija vikatilanteissa, jos sellaisia ilmenee.

Olen tuntenut saavani rauhan tuon päivän käsittelemisestä jo etukäteen mielessäni, ja olenkin hyvällä ja iloisella mielellä menossa kohti tuota päivää. Tunnen tiimini, joka on ammattitaitoinen, osaava ja kykenevä rinnallani ja yhdessä toimimme tuona päivänä tuottaen asiakkaan projektin käyttöönottamisen kokonaisuudessaan loppukäyttäjille. Päivästä tulee hyvä, ja mielenkiintoinen.

Moilanen (2012, 15) mukaan suuremmissa yrityksissä yhtenäisen järjestelmän hyöty on niin merkittävä, että vaikka käyttöönotto ja henkilökunnan opastaminen järjestelmän käyttöön koettaisiinkin projektina isoksi, muodostuu erillisten järjestelmien hallinta lähes mahdottomaksi.

Samaistun Moilasen kanssa siihen, että isoissa yrityksissä järjestelmän hyödyt ovat paremmat, mutta koen myös sen että suuremmissa yrityksissä on myös paljon enemmän muita järjestelmiä sekä ongelmia. Mielestäni usean järjestelmän ympäristö ja sen hallinta on mahdotonta, ja yleensä monen järjestelmän ympäristöt ovat vaikeampia hallita, ja niiden kustannusarviot vuosittain ovat todella suuria.

Ohjelmistokehittäjänä olen ottanut huomioon järjestelmän laadun, sekä sen monipuolisuuden tiimini kanssa. Järjestelmän toimintaa on mielikuvaharjoitteluna testattu moneen kertaan, ja erilaisia vikatilanteita ja ongelmakohtia on käyty myös tiimin kanssa päivittäisissä palavereissa läpi. Tarkoituksenamme on ollut toteuttaa alusta alkaen asiakkaalle laadukas, ja monipuolinen ympäristö joka tukee heidän liiketoiminnan tarpeitaan. Tässä mielestäni olemme onnistuneet hyvin.

Olemme toteuttaneet asiakasprojektin järjestelmän pilvipalveluntarjoajan alustalle, mikä antaa entisestään turvaa niin järjestelmän, kuin myös erilaisten koneiden, ja sovellustenkin toiminnalle. Tietoliikenne, koneet, tietoturva, pääsynhallinta ja monia muita pilviresursseja Azuressa hoitaa järjestelmän toimintaa, ja kokonaisuudella on myös Microsoftin takaama palvelutasosopimuksen mukainen sopimus, joka velvoittaa Microsoftia korjaamaan heidän resurssiinsa kohdistuvan vian, uhan tai hajoamisen sopimuksessa määriteltyjen

aikojen puitteissa. Järjestelmäresurssit Azuressa ovat tietoturvalliset, sekä eristetty ulkoverkosta niin, että mahdolliset hyökkäykset järjestelmään on epätodennäköisiä.

Kehityksen kannalta viikko ei tietenkään ollut kovin positiivinen itselleni, mutta kokonaisuutena olen kehittynyt valtavasti tämän projektin aikana, sillä olen saanut kartuttaa ammattitaitoani monien asiantuntijoiden kanssa, ja toteuttaa hienoa projektia monen ihmisen kanssa.

## 4 Pohdinta ja päätelmät

Aloittaessani ohjelmistokehittäjänä minulla oli ennakkoluuloja paljon, siihen nähden että en osannut koodata enkä tiennyt ohjelmistokehityksen prosesseista yhtään mitään. Rohkeasti päätin olla avoin kaikelle, ja oppia käytännöt.

Ohjelmistokehityksen päätarkoitus on toteuttaa asiakasyritykselle arvoa tuottava toimiva järjestelmä. Järjestelmä mikä tukee yrityksen liiketoimintaa, ja minkä avulla yritys voi tehdä rahaa. Ohjelmistokehityksessä tulee käyttää asiantuntijoita, joilla on laajalti erilaista osaamista. Projektin alkuvaiheessa voidaan jo määritellä projektille tekijät sen vaatimusten, tekniikoiden, sekä käytettyjen työkalujen perusteella.

Ohjelmistokehityksessä tulisikin ottaa huomioon tekijöiden määrä heti alkuvaiheessa, sillä tekijöiden määrä, ja laatu korvaa monta konsulttia matkan varrella. Tämä sulkee pois pitempien jaksottaisten työtehtävien ruuhkautumisia, kun tiimissä on osaamista. Kokemukseni tästä tiimistä, ja tästä projektista on hyvä. Vaikka tiimissämme ei ollut alun alkaenkaan syvempää osaamista kyseisestä järjestelmästä, ovat tiimin jäsenet oppineet työkalujen käytön, sekä järjestelmän sisäisten resurssien toimintatavan loogisuuden omakseen hyvin.

Ohjelmistokehitys itsessään on todella laaja-alaista. Se vaatii ammattitaitoisen asiantuntijaryhmän tuekseen, ja sellaisen osajatiimin joka kykenee viemään pitkän kehitysprojektin läpi tuotannon käyttöönottoon saakka. Ohjelmistokehityksessä on hyvä olla perusasiat hallussa, sekä looginen ymmärryksen taso sellainen, että tietää mistä puhutaan.

Ennakkoluulot, kannattaa sulkea pois omasta mielestään, sillä olen todennut omien ennakkoluulojeni olevan täysin turhia. Avoimuus kasvattaa luottamusta, ole siis avoin osaamisestasi ja taidoistasi. Itse olen kokenut oman avoimuuteni olevan tapa kertoa tiimin jäsenille, millainen osaamiseni on, ja missä asiassa minuun voi luottaa. Tiimilleni on muokautunut kuva osaamisestani, vahvuuksistani ja taidoistani joissa voi luottaa tekemiseeni. Pyrin myös olemaan aina luottamuksen arvoinen kehittäjä.

Aiempi kokemukseni IT-alalta on antanut minulle hyvän lähdön kohti ketterämpää projektityöskentelyä ja ammattitaitoisempaa ohjelmistokehitystä. Vaikka en osannutkaan koodata virallisella koodikielellä, olen oppinut käyttämään laC-työkaluja monipuolisesti, sekä kykeneen toteuttamaan niillä monipuolisia tietojärjestelmiä.

IT-alan aiempi kokemukseni kotimaan, ja kansainvälisen palvelutuotannon parista on kehittännyt minusta pilvipalveluasiantuntijan, mikä helpotti erilaisten pilviresurssien tunte-  
musta ja niiden käyttöönottoa.

Uusina asioina minulle on tullut projektityöskentely, sekä kolmannen osapuolen laC-työkalu-  
illa koodaan järjestelmän infrastruktuuria pilvipalveluun. Olen kehittynyt työssäni vuo-  
den aikana todella paljon, ja odotukset ovat sen mukaisia myös tulevaisuudesta. Tulevai-  
suus näyttää valoisalta urakehityksen puolesta, ja projektikehityksen erilaisten prosessien  
oppimisen puolesta.

Kehitykseni kokonaisuudessaan erilaisten työkalujen käyttämisestä, sekä ketterämmästä  
projektinkehityksestä on kasvanut suuresti, sekä Scrum viitekehityksen mukainen projekti-  
työskentely on avannut minulle uusia näkökulmia laadukkaiden ja monipuolisten tietojär-  
jestelmien kehittämiseen. Ohjelmistokehitys on projektina pitkä, ja todella laaja-alainen  
erilaisten resurssien, asioiden ja vaatimustenkin puolesta. Tietojärjestelmän tuottaminen  
ei ole nopea ratkaisu joka ainoastaan rakennetaan, vaan se on laadukas ammattitaitoisten  
kehittäjien suunnittelema, sekä rakentama tietojärjestelmäkokonaisuus asiakkaan liiketo-  
iminnan tueksi. Sen tarkoitus on palvella asiakasta, ja tuottaa samalla lisäarvoa asiakkaan  
liiketoiminnalle, esimerkiksi siten että asiakas voi myös hyödyntää tuotettaan ja myydä jär-  
jestelmää omille kumppaneilleen.

Päiväkirjamuotoinen opinnäytetyön tekeminen on ollut minulle antoisaa, ja hyvin avaavaa  
etenkin kasvunhaluisena kehittäjänä. Koko projektin edetessä on vastassa ollut haasteita,  
ja vaikeuksia joita olen avannut työssäni sekä sellaista itsereflektiota jota olen harjoittanut  
useaan kertaan. Olen saanut myös perehtyä tietoperustan alaisiin teoksiin, jotka ovat am-  
mentaneet minulle ensisijaisesti oppia, mutta pääsääntöisesti ammattitaitoani kartuttavaa  
tietoa jota olen voinut hyödyntää koko päiväkirjaraportoinnin jakson aikana.

Opinnäytetyön aikana erityisesti sen loppuvaiheessa minulle on auennut entistä enemmän  
mentaaliharjoittelun merkitys työn tekemisen suhteen. Tärkeässä roolissa kehitystyössä  
olen huomannut oman jaksamisen merkityksen, sekä keskittymisen ja rauhallisuuden  
minkä avulla kykenen tehokkaampaan työskentelyyn. Ympäristön valinnalla on tärkeä  
merkitys keskittymiseen, sekä stressaavien tilanteiden visualisointi etukäteen mielikuva-  
harjoitteina auttavat itseäni rauhoittumaan vaikeissa tilanteissa, kun saan käydä tilanteet  
läpi omassa mielessäni etukäteen. Mielikuvaharjoittelu eri tilanteista kuinka koen ne, ja  
kuinka tulen hoitamaan ne helpottavat vaikeita tilanteita ja auttavat minua tehokkaampaan  
työntekoon tulevaisuudessa. Olen saanut siis aikamoisia hienoja asioita myös osaksi elä-  
mäni, ei ainoastaan työelämää varten.

Se ammattitaito, ja oppi, sekä tietoperustan teokset ovat antaneet minulle paljon opinnäytetyön päiväkirjajakson aikana. Olen saanut elämäni tällä matkalla paljon, ja olen voinut hyödyntää lukemaani, sekä ammentaa omaa tietämystäni ohjelmistokehityksen prosesseista, projektityöskentelystä sekä erityisesti olen saanut henkilökohtaisesti kasvaa ihmisenä joka on minulle hyvin tärkeätä. Jotenkin tuntuu hienolta jatkaa elämää tästä eteenpäin, ja odottaa innolla mielenkiintoisia projekteja tulevaisuudessa, sekä hienoa tulevaisuutta alan huipulla olevien hyvien ystävien parissa.

Tulevaisuus sen näyttää, sillä haaveita tietenkin on minullakin, mutta nyt haaveena on tästä eteenpäin kasvaa ihmisenä erinomaiseksi ohjelmistokehittäjäksi, sekä olla monessa mukana. Toivon saavani olla osa erilaisia hienoja projekteja, sekä saavani mahtavia kokemuksia erilaisesta tekemisestä. Jonain päivänä haaveena on toteuttaa oma yritys, mutta tärkeimpänä tulee aina ensin oma hyvinvointi, motivaatio ja se innokkuus joka ohjelmistokehityksessä tarvitaan. Ei se ole kadonnut vielä.

## Lähteet

Container Journal 2022. What's the difference between Docker and Kubernetes. Docker vs. Kubernetes: What's the Difference?. Luettavissa: <https://containerjournal.com/editorial-calendar/best-of-2021/whats-the-difference-between-docker-and-kubernetes/>.

Luettu: 30.3.2022.

Krief, M. 2020. Terraform Cookbook. Provisioning infrastructure in multiple environments. Packt Publishing Ltd. United Kingdom. E-kirja. Luettu: 28.3.2022.

Krief, M. 2020. Terraform Cookbook. Working with workspaces in CI/CD. Packt Publishing Ltd. United Kingdom. E-kirja. Luettu: 11.4.2022

Lampinen, H. 2019. Teknisen dokumentaation haasteet ketterässä järjestelmäkehityksessä. Pro-gradu -tutkielma. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Luettavissa: <https://jyx.jyu.fi/bitstream/handle/123456789/66910/URN%3aNBN%3afi%3ajyu-201912185384.pdf?sequence=1&isAllowed=y>. Luettu: 10.4.2022.

Microsoft Corporation. 2022. What is Azure Resource Manager. Luettavissa: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview>. Luettu: 17.2.2022.

Microsoft Corporation, 2022. Upgrade an Azure Kubernetes Service (AKS) Cluster. Luettavissa: <https://docs.microsoft.com/en-us/azure/aks/upgrade-cluster>. Luettu: 13.4.2022.

Moilanen, A. 2012. Projektinhallinnan käyttöönotto. Opinnäytetyö. Oulun seudun ammattikorkeakoulu, kone- ja tuotantotekniikan koulutusohjelma. Luettavissa: [https://www.theseus.fi/bitstream/handle/10024/40214/moilanen\\_ari.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/40214/moilanen_ari.pdf?sequence=1). Luettu: 25.4.2022.

Netum Group Oyj, 2022. Konsernirakenne. Luettavissa: <https://www.netum.fi/sijoittajille/konsernirakenne/>. Luettu: 19.2.2022.

Niemi, A. & Hietaniemi, J. 2020. Just sopivasti ketterä – Ketterällä kehityksellä kohti parempaa elämää. Scrum. Art House. Helsinki.

Niemi, A. & Hietaniemi, J. 2020. Just sopivasti ketterä – Ketterällä kehityksellä kohti parempaa elämää. Visualisoi mennyttä ja tulevaa. Art House. Helsinki.

Niemi, A. & Hietaniemi, J. 2020. Just sopivasti ketterä – Ketterällä kehityksellä kohti parempaa elämää. Kompassi kuntoon. Art House. Helsinki.

Niemi, A. & Hietaniemi, J. 2020. Just sopivasti ketterä – Ketterällä kehityksellä kohti parempaa elämää. Kaikki lähtee keskittymisestä. Art House. Helsinki.

Ots, K. 2021. Azure Security Handbook. Identity and Access Management. Apress Media. Sveitsi.

Ots, K. 2021. Azure Security Handbook. Shared Responsibility Model. Apress Media. Sveitsi.

Sjöblom, E-P. 2021. Ulkoisen johtajan vaikutus luovuuteen ja innovatiivisuuteen jaetun johtajuuden tiimissä. Pro-gradu -tutkielma. Itä-Suomen yliopisto, yhteiskuntatieteiden ja kauppatieteiden tiedekunta. Luettavissa: <https://erepo.uef.fi/bitstream/handle/123456789/24703/1616751025937890121.pdf>. Luettu: 21.2.2022.

The Linux Foundation. 2022. What are metrics. Luettavissa: <https://prometheus.io/docs/introduction/overview/>. Luettu: 20.3.2022.

The Linux Foundation. 2022. What is Kubernetes?. Luettavissa: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>. Luettu: 3.3.2022.

Wallenius Consulting 2022. Mikä Kubernetes on. Luettavissa: <https://niklaswallenius.fi/kubernetes/>. Luettu: 20.3.2022.