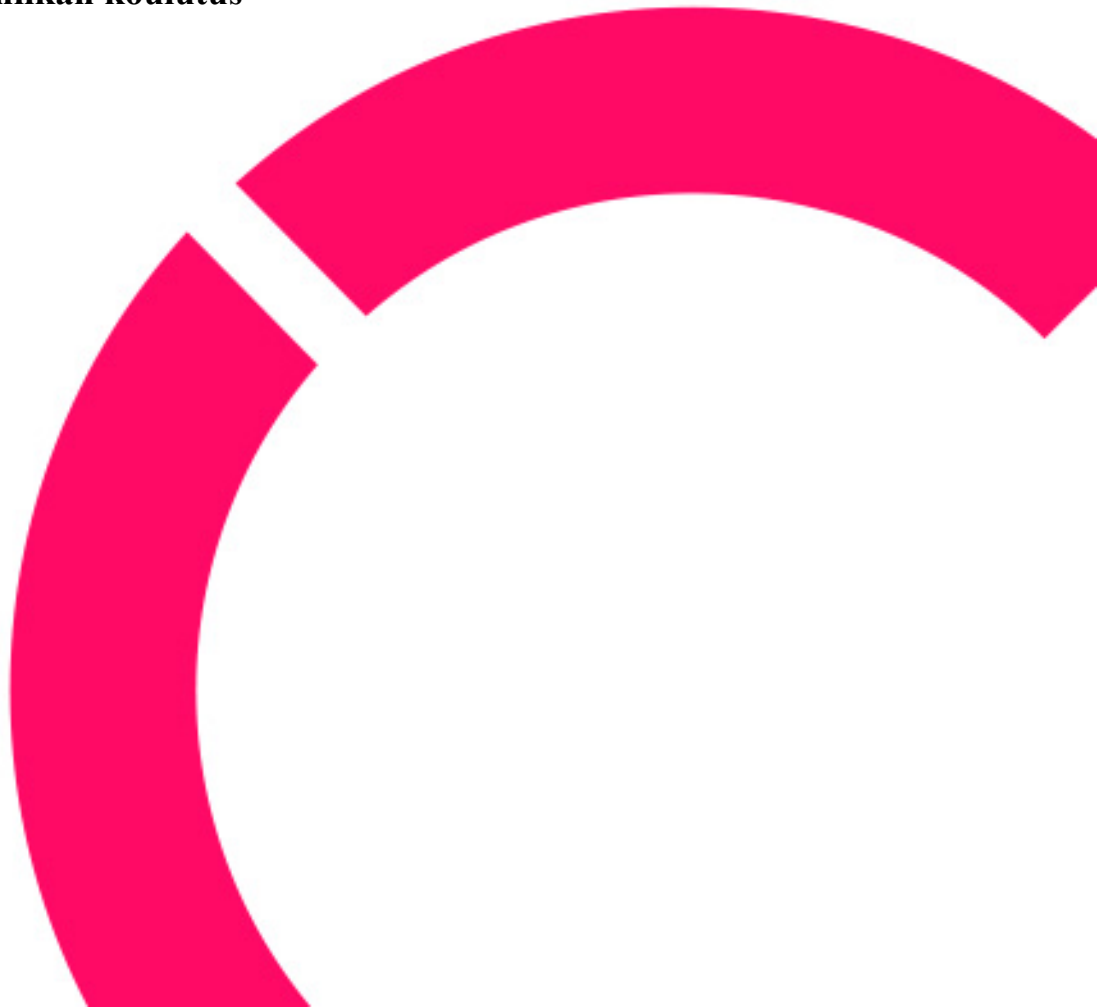


**Karri Kahelin**

# **ELÄMÄÄ IT:N JUOKSUHAUDOISSA**

**Ohjelmistokehittäjän päiväkirja**

**Opinnäytetyö  
CENTRIA-AMMATTIKORKEAKOULU  
Tieto- ja viestintätekniikan koulutus  
Joulukuu 2022**



**TIIVISTELMÄ OPINNÄYTETYÖSTÄ**

<b>Centria-ammattikorkeakoulu</b>	<b>Aika</b> Joulukuu 2022	<b>Tekijä/tekijät</b> Karri Kahelin
<b>Koulutus</b> Tieto- ja viestintäteknikka		<input checked="" type="checkbox"/> AMK <input type="checkbox"/> YAMK
<b>Työn nimi</b> ELÄMÄÄ IT:N JUOKSUHAUDOISSA. Ohjelmistokehittäjän päiväkirja		
<b>Työn ohjaaja</b> Jari Isohanni		<b>Sivumäärä</b> 64 + 0
<b>Työelämäohjaaja</b> Jaana Anglé		
<p>Opinnäytetyö on päiväkirjamuotoinen kuvaus erään ohjelmistokehittäjän arjesta. Opinnäytetyö seurasi opiskelijan työtä päivittäin yhdentoista viikon ajan keväällä 2021 globaalin koronaviruspandemian ja etätyösuositusten hallitessa maailmaa. Viikoittaiset yhteenvedot kertovat, kuinka tuona aikana opiskelija oppi jotakin työstään, maailmasta ja itsestään.</p> <p>Opiskelija työskenteli itsenäisessä yhden hengen tiimissä noin sadan hengen suomalaisessa IT-palveluyrityksessä ohjelmistokehittäjänä. Pääasiallisena työtehtävänään hänellä oli provisiointiskriptien ylläpito ja jatkokehitys Ansible-provisiointityökalun avulla.</p> <p>Päiväkirjaseurannan aikana opiskelija oppi asioita erityisesti Ansiblesta, mentoroinnista ja laadukkaiden tuntikirjausten tekemisestä. Tässä opinnäytetyössä on kerrottu seikkaperäisesti tästä prosessista ja sen eri vaiheista opiskelijan näkökulmasta. Opinnäytetyö on rehellinen kuvaus työelämän iloista, suruista, onnistumisista ja vastoinkäymisistä</p>		
<b>Asiasanat</b> Ansible, devops, ESB, mentorointi, Mule, provisiointi		

## ABSTRACT

<b>Centria University of Applied Sciences</b>	<b>Date</b> December 2022	<b>Author</b> Karri Kahelin
<b>Degree programme</b> Information and communication technology		
<b>Name of thesis</b> LIFE IN THE TRENCHES OF IT. Diary of a software developer		
<b>Centria supervisor</b> Jari Isohanni	<b>Pages</b> 64 + 0	
<b>Instructor representing commissioning institution or company</b> Jaana Anglé		
<p>This thesis is a depiction of everyday life of a software developer in diary format. The thesis followed the author's tasks daily for eleven weeks in the spring of 2021 as the global coronavirus pandemic and telecommuting recommendation ruled the world. Weekly summaries tell how the author learned something about his job, the world and himself during that time.</p> <p>The author worked in an independent one-person team in a Finnish IT services company of about one hundred employees as a software developer. His principal task was maintenance and further development of provisioning scripts using the Ansible provisioning tool.</p> <p>During the diary period the author learned things especially about Ansible, mentoring and creating quality work time reports. This thesis details this process and its different steps from the point of view of the author. The thesis is an honest depiction of the joys, sorrows, successes and adversities of working life.</p>		
<b>Key words</b> Ansible, devops, ESB, mentoring, Mule, provisioning		

## **KÄSITTEIDEN MÄÄRITTELY**

### **Ansible**

Ansible on avoimen lähdekoodin provisiointityökalu

### **Artifactory**

Avoimen lähdekoodin pakettivarasto

### **Daily**

Ketterän ohjelmistokehityksen mallin mukainen päivittäinen tilannekatsauspalaveri

### **ESB**

(Enterprise Service Bus) - Yrityspalveluväylä toteuttaa viestintäjärjestelmän keskenään vuorovaikutuksessa olevien ohjelmistojen välille

### **Inventory**

Kokoelma parametrien arvoja, joita Ansible käyttää provisioinnissa

### **Liquibase**

Avoimen lähdekoodin työkalu tietokantaskeeman ylläpitoon

### **JDK**

(Java Development Kit) - Oraclen Java-kehittäjille tuottama ohjelmistokehityspaketti

### **Jira**

Atlassian-yhtiön tikkettienhallintajärjestelmä

### **Maven**

Apache-järjestön Java-koodin kääntämiseen käytetty, pom.xml-tiedostosta konfiguraationsa lukeva, työkalu

### **Mule**

MuleSoft-yhtiön Java-pohjainen ESB

## **MySQL**

Avoimen lähdekoodin tietokantaohjelmisto

## **Nginx**

Avoimen lähdekoodin WWW- ja välityspalvelin

## **Playbook**

Kokoelma Ansible-komentoja, jotka kertovat, miten jokin ympäristö provisoidaan

## **Provisiointi**

Infrastruktuuri koodina: ohjelmistojen ohjelmallinen valmistelu, kokoonpanon hallinta ja sovellusten ohjelmallinen käyttöönotto.

## **Repositorio**

Pakettivarasto on tietovarasto, jossa ohjelmistojen pakettitiedostoja voidaan hakea verkon yli ja asentaa tietokoneelle

## **Rooli**

Yleiskäyttöinen, parametrisoitava kokoelma Ansible-komentoja

## **Slack**

Organisaatioiden sisäiseen viestintään suunnattu pikaviestinohjelma

## **Tomcat**

Apache-järjestön avoimen lähdekoodin Java-verkkopalvelinympäristö

**TIIVISTELMÄ**  
**ABSTRACT**  
**KÄSITTEIDEN MÄÄRITTELY**  
**SISÄLLYS**

<b>1 JOHDANTO.....</b>	<b>1</b>
<b>2 LÄHTÖTILANTEEN KUVAUS.....</b>	<b>2</b>
2.1 Oman nykyisen työn analyysi.....	2
2.2 Sidosryhmät työpaikalla.....	3
2.3 Vuorovaikutustaidot työpaikalla.....	4
<b>3 PÄIVÄKIRJARAPORTOINTI.....</b>	<b>5</b>
3.1 Viikko 1.....	5
3.2 Viikko 2.....	9
3.3 Viikko 3.....	13
3.4 Viikko 4.....	19
3.5 Viikko 5.....	24
3.6 Viikko 6.....	28
3.7 Viikko 7.....	33
3.8 Viikko 8.....	37
3.9 Viikko 9.....	42
3.10 Viikko 10.....	47
3.11 Viikko 11.....	50
<b>4 POHDINTA JA PÄÄTELMÄT.....</b>	<b>55</b>
<b>LÄHTEET.....</b>	<b>65</b>
<b>KUVIOT</b>	
KUVIO 1 Alfamen organisaatorakenne.....	10

# 1 JOHDANTO

Tämä opinnäytetyö käsittelee päiväkirjamuodossa IT-alan yrityksessä tehtäviä ohjelmistokehittäjän töitä, niissä kohdattavia haasteita ja niiden ratkaisuja. Opinnäytetyö kirjoitettiin yhdentoista viikon ajanjaksolla keväällä 2021, ja se sisältää päiväkirjamerkinnot sekä viikoittaisen yhteenvedon kunkin viikon keskeisimmistä tapahtumista ja kehittämiskohteista. Työn tavoitteena on kehittää omaa toimintaa ja työmenetelmiä ja tunnistaa oman osaamisen kasvua.

Ohjelmistokehittäjänä työskentelen Alfame Systems Oy:lle, jolle olen työskennellyt urani alusta alkaen vuodesta 2008. Urani aikana olen tehnyt monipuolisesti ohjelmistokehittäjän työtehtäviä mitä erilaisimmissa projekteissa.

Alfame Systems Oy on vuonna 2004 perustettu IT-palveluyritys, joka suunnittelee ja toteuttaa liiketoiminnalle kriittisiä integraatio- ja API-ratkaisuja, prosesseja automatisoivia digitalisaatoratkaisuja sekä liiketoimintasovellusten ohjelmistokehitystä. Alfamella on noin sata työntekijää neljällä paikkakunnalla Suomessa. Alfamen asiakkaina on erityisesti julkisen sektorin toimijoita, mutta myös yrityksiä yksityiseltä sektorilta. Pääsääntöisesti työntekijöillä on vähintään ammattikorkeakoulutasoinen tutkinto. (Alfame. GPTW Culture Audit 2022.)

## 2 LÄHTÖTILANTEEN KUVAUS

### 2.1 Oman nykyisen työn analyysi

Työtehtäviini kuuluu kaikenlainen ohjelmistokehitys ja sen ympärillä sijaitsevat tukitoimet. Urani aikana olen käyttänyt monipuolisesti erilaisia ohjelmointikieliä ja teknologioita aina Visual Basicista JavaScriptin kautta C++-kieleen. Viimeaikaisissa projekteissa, joissa olen ollut mukana, pääteknologiat ovat kuitenkin olleet Java-kieli ja Mule-integraatioalusta. Hiukan ennen päiväkirjaseurannan alkamista siirryin pääasiallisesti parantelemaan aiemman projektini provisiointiskriptejä Ansible-provisiointityökalun avulla.

Ansiblen playbookien kirjoittaminen ei teknisesti ottaen ole ohjelmointia, sillä YAML-merkkikieli, jolla playbookit kirjoitetaan, ei ole ohjelmointikieli. Käytännön tasolla työ muistuttaa kuitenkin hyvin paljon ihan perinteistä ohjelmointia: muokkaan tekstieditorilla tekstimuotoisia YAML-tiedostoja, jotka voivat sisältää muuttujia sekä ehto- ja toistorakenteita jotka "tulkki", eli Ansiblen ajoympäristö sitten suorittaa. Työssä tarvitaan samoja loogisen ajattelun kykyjä kuin "perinteisessäkin" ohjelmoinnissa.

Töiden seurantaan ja organisointiin käytössämme on Jira-tiketöintijärjestelmä. Järjestelmään syötettävä tiketti on yksi tehtävä työ. Tiketin tila kertoo työn tilan: suunniteltu, työn alla, testauksessa, valmis. Tiketille pyritään kirjaamaan kaikki työn suorittamisen kannalta olennainen lähtötieto, tiketin ratkaisemiseksi tehdyt toimenpiteet sekä tieto siitä, milloin tiketti on valmis. Isoja tikettejä voidaan pilkkoa alitiketeiksi, jolloin tiketit muodostavat loogisen puurakenteen.

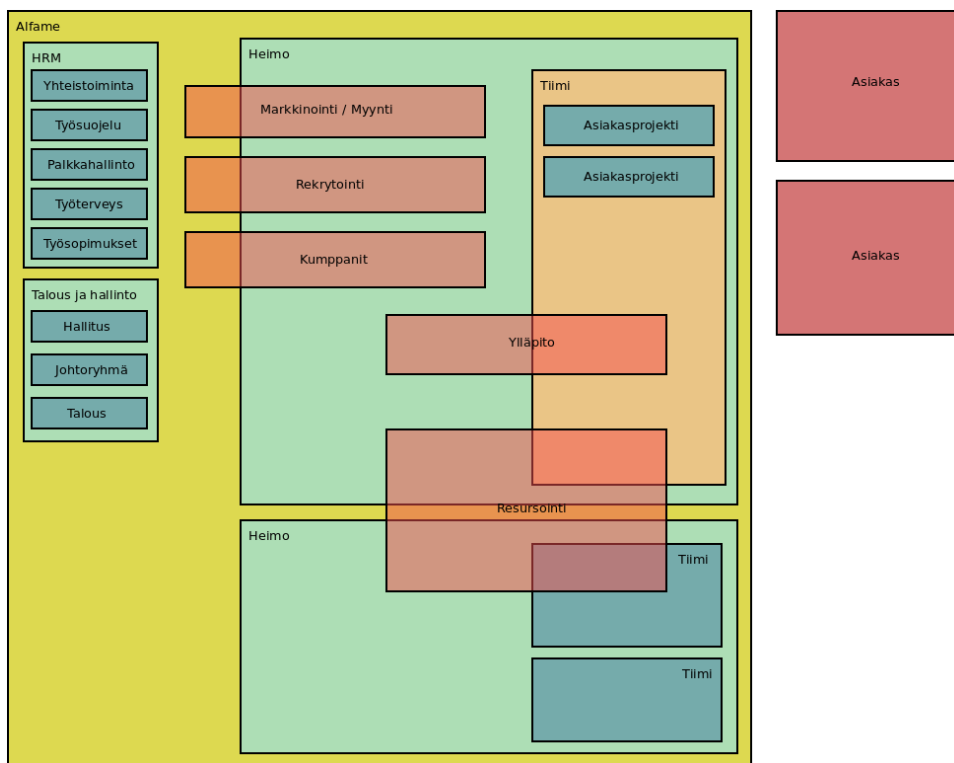
Teoriatasolla teen työtä tiimissä. Tuon tiimin koko provisiointityökalujen paranteluprojektissa vain satuu olemaan yksi työntekijä. Tukenani on kuitenkin koko ajan arkkitehti, jolla on korkeamman abstraktiotason näkemys asiakkaan lukuisista ympäristöistä ja sovelluksista. Arkkitehdin kanssa pidämme päivittäin Scrum-henkisen palaverin, jossa seurataan työn edistymistä ja suunnitellaan tulevia töitä tiketöintijärjestelmän avulla. Lisäksi koordinoin tuotantopäivityksien kaltaisia asioita muiden samalle asiakkaalle töitä tekevien tiimien kanssa.

Tyypillinen tiketti on lisätä toiminnallisuus johonkin jaettuun Ansible-komponenttiin, esimerkiksi Mule-integraatioalustan asentavaan komponenttiin, eli Ansiblen terminologialla rooliin. Joskus tiedän suoraan, miten muutos kannattaa tehdä, joskus kerään sitä varten tietoa kollegoiltani. Toteutan muu-



tokset ja testaan niitä. Tätä varten omalla työasemallani on useita virtuaalikoneita, joilla voin nopeasti testata tekemiäni muutoksia, eivätkä mahdolliset virheet vaikuta muiden työhön millään tavalla. Lopullinen testaus tehdään kuitenkin aina aidossa asiakkaan ympäristössä.

## 2.2 Sidosryhmät työpaikalla



KUVIO 1. Alfamen organisaatiorakenne

Kuviossa 1 on kuvattu Alfamen sidosryhmät, josta ilmenee myös työssäni esiintyvät sidosryhmät. Yrityksellä on toimisto neljällä paikkakunnalla Suomessa. Projektitiimit voivat koostua, ja yleensä koostuvatkin, fyysisesti useammalla paikkakunnalla olevista ihmisistä. Yhden hengen toteutustiimissäni on siis lisäksi tukena arkkitehti sekä asiakasvastaava. Myös asiakkaan edustajia on suoraan tavoitettavissa Slackissä. Samalle asiakkaalle tekee töitä kuitenkin useampikin tiimi, ja työn koordinoitua heidän kanssaan tarvitaan. Kaikista tiiveintä viestintä on saman heimon sisällä ja seuraavaksi tiiveintä asiakkaan suuntaan. Viestintä muihin tahoihin on aktiivisinta erikoistilanteissa, kuten uuden projektin aloituksessa.

Tätä kirjoittaessa yritys on jaettu kolmeen heimoon: integraatiot, .NET ja frontend. Heimojen vastuulla ovat myynnintuki, rekrytointi, vasteajallisen ylläpidon koordinoitua ja asiakastiimien resursointi.

### 2.3 Vuorovaikutustaidot työpaikalla

Opinnäytetyön seurantajaksolla vallitsivat globaalin koronaviruspandemian johdosta poikkeusolot, jolloin etätyöskentelyä suositeltiin kaikille, joille se vain on mahdollista. Ennen pandemiaa työskentelin yleensä toimistolla, mutta koska tiimi ja asiakkaat voivat olla muilla paikkakunnilla, oli videoneuvottelu- ja pikaviestinohjelmien käyttö jokapäiväistä jo ennen kokoontumisrajoituksia.

Käytämme sekä Slack-pikaviestinohjelmaa, joka sisältää pikaviestimen lisäksi ruudunjako- ja videopuheluominaisuudet, että Google Meet -videopuheluohjelmaa. Ensimmäinen on luonteva spontaaneihin yhteydenottoihin, jälkimmäinen sisältää hyvän tuen kalenteroiduille palaverille, ja siihen on helpompi kutsua organisaation ulkopuolisia ihmisiä (Praecom. Mikä on paras etäkokous- ja kollaboraatioalusta? Vertailussa: Microsoft Teams, Webex, Zoom, Google Meet ja Slack.) Jira-tiketöintijärjestelmään muodostuva dokumentaatio on myös osa vuorovaikutusta asiakkaan suuntaan.

Mikäli vapaamuotoista, ei suoraan työtehtäviin liittyvää, rupattelua kollegoiden kanssa ei oteta huomioon, pääkontaktini on arkkitehti, jonka kanssa pidämme päivittäisiä palavereja. Jos tiimissäni olisi useampi henkilö, tulisi heidän kanssaan juteltua enemmän. Mitään tiukkaa hierarkiaa ei ole, ja vaikka asiakasvastaava ja arkkitehti hoitavatkin suurimman osan suorasta asiakaskommunikoinnista, voin tarvittaessa "vetää asiakasta hihasta" milloin vain Slackissa.

## 3 PÄIVÄKIRJARAPORTOINTI

### 3.1 Viikko 1

Maanantai

Tänään oli tarkoitus jatkaa tulostuspalvelun provisioinnin päivittämistä. Aloitin päivittämällä JDK-version provisiointiin ja testaamalla muutokset. Itse tulostuspalvelu-sovellus vaikuttaa toimivan edelleen, mutta samalla palvelimella pyörii muitakin sovelluksia, joiden yhteensopivuus pitää testata myös. Toisin kuin Mule-ohjelmakehyksen kanssa, emme provisioi useita rinnakkaisia versioita JDK:sta. Se olisi mahdollista, mutta ottaen huomioon JDK:n yleisen laadukkaan taaksepäin sopivuuden, vaikuttaa se liioittelulta ja ylimääräiseltä työltä. Kyse on kuitenkin julkiseen Internetiin asiakkaille näkyvistä sovelluksista, joten täysi testaamattomuus olisi naiivia.

Olemassa olevassa kehitysympäristössäni oli muutamia sovelluksia jo valmiiksi, mutta ne tuntuivat kaatuvan siihen, ettei kantaan saada yhteyttä, todennäköisesti koska joku virtuaalikone joko ei ole päällä, tai edes olemassa. Lisäksi nämä ovat erittäin todennäköisesti vanhoja versioita. Dailyssä päätimme arkkitehdin kanssa, että työnnän muutokset ja testaamme niitä dev-ympäristössä, jossa on sekä ajantasaisempia versioita ajossa, että kantayhteydet ja vastaavat todennäköisesti kunnossa.

Dailyssä minua muistutetaan myös, että Founder Monitorin testaaminen testiympäristössä on samassa tilanteessa kuin se oli perjantaina. Teen tiketin palomuuuriavauksista PRH:n Helpdeskillle, jotta se on valmiina, kun etelän hiihtolomat loppuvat.

Tiistai

Päivän suunnitelma on jatkaa playbookin testaamista sekä lokaalisti, että dev-ympäristössä.

Ajan playbookin dev-ympäristöön ja positiivisesti se menee läpi. Testailen sovelluksia hetken aikaa, mutta arkkitehti mainitsee, että vähemmällä testaamisella pääsee, kun työlistalla oleva Tomcatin päivitys tehdään ja testataan samalla. Hän on luonnollisesti oikeassa. Toistaiseksi olen paikallisesti provisioinut vain Mule-sovelluksia, mutta minulla on jäänteinä jostain vanhasta projektista kehitysympäristö sellaisellekin projektille, joka käyttää Tomcatia. Päivitan Tomcatin itselleni ja Artifactoryyn, ajan

playbookin ja huomaan, että se yrittää yhdistää joihinkin ihan muunnimisiin koneisiin kuin mitä minulla on. Päätän, että provisoinnin hosts-tiedoston muokkaaminen on nopeampaa kuin pystyttää uusi kehitysympäristö ja käytän hetken aikaa koneiden nimeämiseen. Onnistun myös jumittamaan koneeni käynnistämällä yhden virtuaalikoneen liikaa. 32 gigatavua keskusmuistia täyttyy yllättävän nopeasti.

Ruokatunnilla on tiedossa ohjelmaa: Palvelumuotoilun jatkokurssin luennosta poimin menetelmiä käyttäjien kokemusten ja mielipiteiden kartoittamiseksi.

Playbookia ajaessa huomaan, ettei symbolinen linkki Tomcatin uuteen versioon vaikuta päivittyvän, mutta seuraavalla ajolla se kuitenkin päivittyy. Pistän asian muistiin ja keskityn suurempaan ongelmaan: playbook ei myöskään deployaa itse sovellusta, vaikka näyttää menevän virheittä läpi. Selvitän ongelmaa hetken aikaa mutta tehtaan pilli viheltää ennen kuin keksin ratkaisua.

## Keskiviikko

Tänään yritän saada korjattua playbookin ongelmat ja mahdollisuuksien mukaan edistää seuraavaa tikkettä. Päivä alkaa palaverilla, jossa speksaan parin arkkitehdin kanssa tulevaisuuden tarpeita lokituksen ja valvonnan suhteen. Olen mukana, koska on tarkoitus, että teen tutkimusta aiheesta tätä opinnäytetyötäkin silmällä pitäen. On ihailtavaa seurata asiansa osaavien ihmisten työskentelyä, ja samaan aikaan omat huomioni sivusta vaikuttavat vähäisiltä. Haluan kuitenkin kasvaa arkkitehdin rooliin jollain aikajänteellä, jolloin töiden seuraaminen on hyvinkin mielenkiintoista ja opettavaista.

Palaverin aikana kollega kysyy apuja omaan tikkettiin liittyen. Tyhjähköstä kalenterista huolimatta jo toisen kerran peräkkäin joudun toteamaan olevani juuri palaverissa. Palaverin jälkeen katsomme videoneuvottelussa, miten tehdään yksikkötesti Mule-flow, jota varten populoidaan h2-kanta ja tehdään sinne kysely. En muista juuri tällaista tehneeni itse aikaisemmin, mutta onnistun googlettamaan sopivan sivun Mule:n dokumentaatiosta (Mulesoft. Database Connector)

Pääsen lopultakin oman tikkettini pariin, jossa pureudun apache\_tomcat\_deploy -rooliin syvemmin. Huomaan, että rooli muistuttaa Mule-sovelluksen asentavaa roolia, jota olen ihmetellyt jo aiemmin. Täälläkin, kuten sielläkin, on lippumuuttuja, joka pitää asettaa todeksi, jotta rooli edes yrittää asentaa sovelluksen, joka ei ole muuttunut. Lisäksi muistan, että tähän liittyen mule\_esb\_deploy -roolissa oli bugi. Kaivan versiohallinnasta muutoksen, jossa se korjattiin, teen saman korjauksen

apache\_tomcat\_deploy-rooliin ja playbookin ajettuani saan todeta, ettei playbook näistä lupaavista muutoksista huolimatta vieläkään asenna sovellusta.

Koodin lukemisen ja muutaman debug-printin jälkeen havaitsen, että playbook hakee pakettia, jossa versionumeron 2.1.1 sijaan esiintyy sana "latest". Muistan yhtäkkiä kollegan maininneen Slackissa jostain "latestiin" liittyvää ja etsin viestin. Siinä kollega kertoo, miten hän löysi toisen projektin provisionista bugin, jonka aikaisemmin tekemäni Ansiblen päivitys laukaisi. Ansiblen 3.10-versio korjasi Ansiblessa olleen bugin "latest"-version käyttöön liittyen, ja meidän playbookeissa oleva erityiskäsittely "latest"-versiolle piti poistaa. Kaivan esiin tuon muutoksen versiohallinnasta, mutta työtunnit alkavat tulla täyteen, joten toteutus saa jäädä huomiseksi.

Slackin lukemista voisi pitää ajan haaskauksena, tai ainakin vitkutteluna, mitä se tietysti joskus onkin, mutta on hyödyllistä tiedostaa, että ohimennen luettu Slack-kommentti saattaa joskus tulevaisuudessa säästää paljonkin aikaa virheenselvityksessä. Itselläni myös on tapana dokumentoida tämänkaltaiset asiat ainoastaan tiketin kommentteihin. Voisi olla hyödyllistä huudella niistä enemmänkin julkisemalla foorumilla. Haasteena on toki identifioida ne asiat, joista tämänkaltaista hyötyä voisi olla muille.

Torstai

Tänään jatkan Tomcat-päivityksen ongelmien selvittämistä. Aloitan vertailemalla oman projektini roolia ja kollegani aiemmin korjaamaa toista projektia. Nämä eivät ole täysin samalla tavalla toteutettuja, enkä suoraan näe, miten voisin siirtää hänen ratkaisunsa omaan projektiin. Olen kuitenkin hyvin varma siitä, että pohjalla oleva ongelma on sama, joten keskityn tutkimaan sitä. Lopuksi toteutus on hyvinkin yksinkertainen ja pääsen käyttämään siinä Ansiblen filttareita, joita olen muutamaan otteeseen nähnyt käytettävän, mutta joihin en itse ole tarkemmin perehtynyt ennen tätä. (Using filters to manipulate data)

Käyn palaverissa koskien tulevan teknologia-alustamme kuormanjakoa. Palaverissa alustan kanssa tähän asti enemmän tekemisissä olleet arkkitehdit pohtivat layer 7 -tason kuormanjaon mahdollisuutta on premise -ympäristössä. Minun tehtäväkseni tiketöidään selvittää, miten Rancherin lokit saa ulos Elasticsearchiin. Palaverin jälkeen saan seurata Slackista yhden arkkitehdin ääneen ajattelua, jossa hän toteaa palaverissa pohditun ratkaisun mahdottomaksi ja että tiimin tulisi keskittyä vararatkaisuna olleeseen MetalLB:en. Itse priorisoin tässä kohtaa kesken jääneen tikettini kuitenkin edelle.

Kun nyt applikaatioiden asentaminen oikeasti toimii, palautan kehitysympäristööni vanhan version Tomcatista ja yritän päivittää sen. Playbook asentaa uuden Tomcatin, mutta ei asenna sovelluksia siihen, eikä käynnistä Tomcatia uudestaan niin, että ajoon tulisi uusi versio. Samaan aikaan chattailen Slackissa erään toisen kollegani kanssa hänen Ansible-ongelmistaan. Muutamat ongelmista johtuvat 2.10-päivityksestä, muutamat siitä, että joku ei ole testannut asioita loppuun asti.

Tulevan teknologia-alustan osalta pääsenkin näköjään sukeltamaan heti syvään päähän, mikä ei ole optimaalista, mutta toki myöskään ensimmäinen kerta työhistoriani aikana. Olisi erinomaista, jos minulla olisi aikaa perehtyä ainakin Kubernetesin sielunelämään ihan teoriatasolla, mutta tällä kertaa muiden opiskelujen kanssa se ei vaikuta hirveän todennäköiseltä.

Perjantai

Päivän suunnitelma on saada Tomcat-tiketti suljettua ja edettyä dev-ympäristössä testaamiseen. Saadakseni selvyyttä siihen, miten tikettini kanssa etenen chattailen toisen, asiakkaalla pitkään olleen ja monia projekteja suunnitelleen ja tämän roolin alun perin toteuttaneen, arkkitehdin kanssa siitä, miten Tomcat-päivityksen tulisi toimia. Pitkällisen väännön jälkeen saan puristettua hänestä ulos, että nykyinen toiminta ja se, että Tomcat pitää käynnistää manuaalisesti uudestaan, on suunniteltua toimintaa, eikä bugi roolissa. Hän toki myöntää, että voisi se automaattisemminkin toimia. Suunnittelen eteneväni tällä ja tekeväni tulevaisuutta varten jatkokehitystiketin lisäautomatisoinnista, sillä kyseessä ei ihan kuitenkaan ihan pieni työ.

Dailyssä oman projektini arkkitehti on kuitenkin sitä mieltä, että ei asiat todellakaan näin saa toimia, nykytoteutuksessa on ihan liikaa käsityötä.

Vaikka hiukan tiukkaa tekeekin heidän kalenteriensa yhteensovittamisessa, varaan palaverin molempien arkkitehtien kanssa heti maanantaille. Omien töideni sujumisen takia on tärkeä saada yhteinen selkeä näkemys mahdollisimman pian.

Päivän loppuksi testaan vielä Mule-roolin ja yllätyksekseni päivityksen suhteen se toimii eri tavalla kuin Tomcat-rooli, vaikka molemmat päällisin puolin vaikuttavat kopioilta toisistaan. Tiedämpä ainakin, mitä maanantaina tulen tekemään ennen palaveria.

Ei tietenkään ole mitenkään harvinaista, että arkkitehdeillä on jonkin verran eriävät käsitykset asioista, mutta varsin harvoin ne ihan näin selkeästi näkyvät omassa työssäni.

## Viikkoyhteenveto

Viikko oli kohtalaisen monipuolinen. Kävin useammassakin palaverissa suunnittelemassa tulevaa, pääsin debuggaamaan niin Mule- kuin Tomcat-roolejakin, toteuttamaan h2-tietokantaa käyttävän Mule-yksikkötestin ja vielä oppimaan palvelumuotoilun menetelmiä. Mielenkiintoisinta ehkä oli havaita tunnistettujen Slack-kommenttien yllättävä arvo, jota en aiemmin ole tullut pohtineeksi. Arkkitehtien kutsuminen samaan palaveriin selvittämään teknisiä yksityiskohtia keskenään on myös jotain, mitä en aikaisemmin ole joutunut tekemään. Suurin osa aiemmista projekteistani on toki ollut niin pieniä, ettei niissä ole kuin yksi arkkitehti, tai mikäli arkkitehtejä onkin sattunut olemaan useampi, on heillä ollut selvät erilliset vastualueet.

En täysin ole tyytyväinen tapaan, jolla Mule- ja Tomcat-roolit tekevät asioita. Minusta ne rikkovat ”principle of least surprisea” (Lomio 2021), mutta suurempi refaktorointi ei nyt tässä kohtaa ole asianmukaista. Työssä on välillä vain pakko sietää keskinkertaisia, tai jopa kehoja ohjelmistopalikoita. Jos refaktoraisin jokaisen komponentin aina mieleisekseni, en koskaan saisi oikeita töitä tehtyä. Tärkeintä on vain tuntea työkalujensa oikut.

Viikon aikana uutta oli ainakin Mulen, H2-kannan ja yksikkötestien yhdistäminen. Olen aikaisemmin tehnyt yksikkötestejä, jotka käyttävät Mulea ja yksikkötestejä, jotka käyttävät H2-kantaa, mutta en yksikkötestejä, jotka käyttäisivät molempia yhtä aikaa. (Mulesoft. Database Connector) Seuraavan viikon tavoite on saada Tomcatin automaattisesti päivittävä playbook valmiiksi.

## 3.2 Viikko 2

### Maanantai

Päivän suunnitelma on saada selvyys siihen, miten Tomcatin päivityksessä oikein pitää menetellä. Aamusta ennen palaveria vertailen Tomcat ja Mule-rooleja keskenään, ja todellakin toinen on selkeästi muokattu kopio toisesta. En nyt kuitenkaan löydä ratkaisevaa eroa, miksi toinen toimii päivityksen suhteen ratkaisevasti eri tavalla.

Videoneuvottelu arkkitehtien kanssa sujui sopuisasti, ja roolit alun perin toteuttanut arkkitehtikin on sitä mieltä, että päivityksen voisi hoitaa automaattisesti. Koitamme kehitysvirtuaalikoneessa pikaisesti muutamia ideoita käsin komentoriviltä. Ensimmäinen idea on provisioida uusi Tomcat, poistaa provisioitava sovellus vanhalta – tai vanhoilta – instansseilta ja käynnistää uusi Tomcat-instanssi. Lopuksi mikäli vanha Tomcat instanssi ei sisältäisi yhtään sovellusta se poistettaisiin serviceineen. Näin sovellukset voisi siirtää yksi kerrallaan ajamalla niiden playbook. Toinen arkkitehdeistä kuitenkin hoksaa, että Systemd-service osoittaisi tämän jälkeen uuteen Tomcatiin ja vanha jäisi vähän ikään kuin limboon. Muutamia ideoita esitetään, mutta toteamme yhteistuumin ne liian monimutkaiseksi ja epävaikaksi ja lopulta päätetään, että Tomcatin servicet versioidaan, playbook poistaa vanhat versiot sovelluksesta ja tyhjien Tomcatien ja niiden servicejen siivous hoidetaan sitten lopuksi käsin. Tämäkin ratkaisu on huomattavasti automaattisempi kuin mitä vanha oli.

Palaverin jälkeen alan toteuttaa tarvittavia muutoksia. On paras käytäntö, että käytettäessä arvo ohjelmakoodissa, Ansible-playbookissa tai muussa vastaavassa luodaan vakio tai muuttuja, jolle tuo arvo annetaan ja käytetään koodissa sitten tätä nimettyä muuttujaa tai vakiota. Idea on, että arvon muuttuessa sen voi muuttaa yhteen paikkaan ja muutos tulee sitten kerralla voimaan kaikkialle. Tämän nimenomaisen projektin kohdalla en kuitenkaan voi välttyä ajatukselta, että mahdollisimman kovakoodatuilla arvoilla olisin tehnyt muutokset nopeammin. Huomaan, että minun on vaikea pitää mielessäni, mikä nyt on juuri tietyn muuttujan arvo, etenkin kun ne voivat koostua toisista muuttujista, eikä tätä auta se, että Ansiblen toimintatavan takia kaikissa on pitkä `apache_tomcat` tai `apache_tomcat_deploy`-prefiksi. Näin jälkeempäin ajatellen olisi pitänyt olla järjestelmällisempi ja hyödyntää muistiinpanoja paremmin.

Tiistai

Tänään suunnittelen jatkavani Tomcatin automaattisen päivityksen toteuttamista. Jatkan siitä, mihin eilen jäin ja saan muuttujanimet ja niiden arvot lopulta järjestykseen. Provisiointi menee läpi, mutta huomaan, että sovellus ei käynnisty, vaan valittaa ettei saa luotua indeksiä tietokantaan. Poistan kannan, ja ajan provisioinnin uudestaan, mutta ongelma pysyy. Tämä on hiukan outoa, sillä sovellus on tuotannossa ja minulla pitäisi olla siitä ajantasainen versio. Ajattelen kuitenkin testata sitten toisella sovelluksella, joka tuotannossa tulee pyörimään samalla palvelimella. Playbookia tarkastellessani huomaan kuitenkin, että `apache_tomcat` ja `apache_tomcat_deploy` -roolien sijaan tämä toteutus käyttää vanhempia rooleja. Nostan tämän esiin dailyssä ja arkkitehti on kanssani samaa mieltä siitä, että koko työni on turhaa, mikäli tämän sovelluksen provisiointia ei päivitetä käyttämään uusia rooleja. Teen päivityksestä tiketin ja aloitan työn, mutta se ei ehdi valmiiksi ennen kuin päivän työaika tulee täyteen.



Ruokatauolla oli jälleen palvelumuotoilun jatkokurssin tilannekatsaus. Olen hiukan huolissani tästä, sillä ryhmäni projekti on edistynyt hitaasti, kun tarvittavia ihmisiä on ollut hankala saada kiinni. Onneksi sovimme kuitenkin viikon lisäajasta, sillä muutkin ryhmät ovat olleet kiireisiä töissä.

## Keskiviikko

Päivän suunnitelma on jatkaa Tomcatin automaattisen päivityksen toteuttamista. Aamusta teen loppuun eilen kesken jättämäni playbookin päivityksen käyttämään uusia rooleja. Samalla vaihtelen Slack-chatissa ideoita kollegani kanssa hänen päivittäessään Ansiblea jollekin palvelimelle ja taistelllessaan Python-kirjastojen epäyhteensopivuuksien kanssa.

Seuraavaksi jatkan edellistä tikettiäni toteuttamalla ominaisuuden, jossa Ansible listaa palvelimella jo olevat tällä playbookilla asennettavat sovellukset, ja poistaa niistä vanhoissa Tomcateissa olevat. Kokemukseni ansiosta toteutan ominaisuuden ensin niin, että oikeasti poistamisen sijaan playbook tulostaa terminaaliin hakemistojen nimet, jotka se poistaisi. Se oli jälleen kerran hyvä idea.

Dailyssä käy ilmi, että kollegani on toteuttanut kantaläpikäsitteilyä hiukan erikoisella tavalla ja tämä vaikuttaisi mahdollistavan SQL-injektion. En itse olisi toteuttanut ominaisuutta noin, mutta onhan tämä kalastelupuhelun lisäksi tervetullut muistutus siitä, että tietoturva-asiat on jatkuvasti pidettävä mielessä.

## Torstai

Tänään aion jatkaa Tomcatin automaattipäivityksen toteutusta. Kalenteriini on joskus myöhään eilen mystisesti ilmestynyt aloituspalaveri erälle vanhalle asiakkaallemme. En tiedä aiheesta mitään, mutta menen palaveriin. Palaverissa käy ilmi, että eräs vanha asiakkaamme, jolle on viimeksi tehty töitä noin kaksi vuotta sitten, haluaa nyt yhden integraation lisää, ja projektiin tuleva tiimi on viime hetkellä muuttunut sisältämään minut. Arkkitehdiksi projektiin lupautunut kollegani kertoo, että esimerkiksi kehitysympäristön tila on ilmeisesti hiukan epämääräinen, ja lisäksi jotkin projektissa käytetyt teknologiat, kuten Mule, on tässä ajassa ehtinyt kerätä päivitysvelkaa. Noita hommia kaksi vuotta sitten tehnyt insinööri onneksi on vielä meillä töissä ja sovimmekin, että katsotaan homma alkuun hänen kanssaan. Jatko sovitaan joko samalle iltapäivälle, mutta minulle tuntemattomasta syystä se siirtyy huomiseksi.

Tomcatin päivittämisen suhteen olen tilanteessa, jossa minulla on n-kappaletta sisäkkäisiä listoja, joista pitäisi muodostaa yksi lista. Luen pariin otteeseen läpi filttareista kertovan sivun Ansiblen dokumentaatiosta, mutta mikään niistä ei tunnu tekevän sitä, mitä haluan. (Ansible project contributors. Using filters to manipulate data) Vaihdan lähestymiskulmaa ja tietorakenteen muokkaamisen sijaan alan miettiä, miten voisin käydä tuon olemassa olevan tietorakenteen läpi. Muistan hiljattain toteuttaneeni kaksiulotteisen silmukan Ansiblella ja etsin tuon toteutukseni käsiini. Vanhan toteutuksen pohjalta teen vastaavan tähän ja se toimii juuri niin kuin pitikin.

## Perjantai

Päivän suunnitelmana on saada valmiiksi Tomcatin automaattinen päivitys Ansiblen playbookilla. Osallistun videoneuvotteluun, jossa projektia pari vuotta sitten tehnyt kollegani, sekä uusi tiimikaverini, pari kuukautta sitten taloon tullut juniorikehittäjä ovat jo ennen palaverin alkamista päivittäneet sovelluksen uusimman Mule 3.9-version päälle. Taisi mennä kivuttomasti. Kauhulla katselen, kuinka kollegani Mac-koneelle asentuu Ansiblen versio 3.0. 2.10-päivitys tuotti ihan riittävästi ongelmia. Kollegani ajaa provisiointia ja aina virheen ilmaantuessa minä tai projektia aikaisemmin toteuttanut kollegani annamme hänelle ohjeita siitä, miten selvitetään, missä ongelma on, ja miten se lopulta ratkaistaan. Kuinkas ollakaan, yksi ensimmäisistä vastaantulevista ongelmista on se, että Ansiblen uudet versiot haluavat käyttää firewall-moduulista nimeä `ansible.posix.firewalld`. Tämä ongelma onneksi oli tullut jo ratkaistua aiemmissa Slack-chateissa kolmannen kollegani kanssa.

Tällainen juniorikehittäjien perehdyttäminen on yrityksessämme nykyään tavallista, ja olenkin hiukan kateellinen, etteivät asiat olleet yhtä hyvin minun aikanani tullessani taloon. Saamme sovelluksen käynnistymään ja päätämme palaverin tokaisuun "nonii, eiku hommiin!"

En kuitenkaan ala "hommiin" vaan vaihdan takaisin edellisen projektini pariin. Viimeinen ongelmani on se, että nykyinen Tomcatin versio pitää jättää tämän käsittelyn ulkopuolelle, sillä siitä huolehtii olemassa oleva koodi myöhemmin. Ansiblen dokumentaatioissa on hyviä esimerkkejä säännöllisillä lausekkeilla mätsäämisestä (Ansible project contributors. Tests, kappale Testing Strings). Kuitenkin dokumentaation esimerkeissä kaikki patternit ovat kovakoodattuja, mikä harvemmin tosielämän tapauksissa tulee kyseeseen. Yrittäessäni muodostaa patternin ketjutetuista muuttujista saankin eteeni virheilmoituksen. Yritän lukea dokumentaatiota, mutta siellä ei puhuta mitään tästä tapauksesta. Testaamalla erilaisia vaihtoehtoja havaitseen, että voin mätsätä muuttujaa toista muuttujaa vasten, mutta että en voi tehdä operaatioita tuolle muuttujalle tässä yhteydessä. Tämä kuitenkin riittää, määrittelen inventoryyn

muuttujan, jossa on haluamani patterni ja playbook lopultakin tekee mitä haluan. Aika kuitenkin loppuu kesken ja siirrän osan testaamisesta, sekä dokumentoinnin, maanantaille.

## Viikkoyhteenveto

Olen kohtuullisen tyytyväinen viikon aikaansaannoksiin. Ihan maaliin asti Tomcatin automaattisesti päivittävää playbookia en saanut, mutta kaikki suuret ongelmat on ratkaistu, ja vain testaaminen ja dokumentointi jää seuraavalle viikolle. Uusina asioina tällä viikolla tuli vastaan erityisesti Ansiblen filteerit, ja kaksiulotteisten listojen kanssa pelaaminen.

Muutaman kerran uusien kehittäjien perehdyttämistä nyt kokeilleena olen havainnut sen ihan mukavaksi ja hupaiseksi työtehtäväksi. Aionkin mainita kiinnostukseni aihetta kohtaan, kun tulevista työtehtävistä seuraavan kerran tulee puhetta. On mukavaa nähdä ihmisten oppivan uusia asioita ja omakin itsetunto vahvistuu muita opettaessa.

Tämän viikon aikana löysin uusia puolia itsestäni ja osaamisestani. Olen pitänyt itseäni mieluiten pimeässä palvelinhuoneessa nyhjöttävänä erakkona, mutta mentorin roolissa itsestäni näyttäisi löytyvän piileviä sosiaalisia kykyjä.

### 3.3 Viikko 3

#### Maanantai

Tänään testaan Tomcatin automaattisen päivityksen loppuun. Aloitan päivän jatkamalla testausta. Aluksi toiminta näyttää lupaavalta, mutta kun olemassa olevan Tomcatin rinnalle käynnistetään toinen Tomcat, ei se käynnisty, koska se yrittää bindata samoihin portteihin olemassa olevan Tomcatin kanssa. Roolin tekijä on ottanut tämänkaltaiset tilanteet huomioon tarjoamalla helpon tavan muuttaa kaikkia kerralla. Tämä on kuitenkin käsin tehtävä konfiguraatio. Kirjaan ongelman ylös ja jatkan testaamista.

Päivitys itsessään vaikuttaa toimivan hyvin, joten päätän testata sitä tapausta, joka näin siirtymätilanteessa tulee heti ensimmäisenä vastaan. Asetan käsin testiympäristöni vastaamaan sitä, mitä testi ja tuotanto tällä hetkellä ovat ja ajan playbookin. Playbookin suoritus keskeytyy hetken kuluttua siihen,

ettei vanhemman version Tomcatia saada käynnistettyä. Lokeja tarkasteltuani havaitsen, että playbook yritti käynnistää saman Tomcat-instanssin kahdesti, ensin nimellä tomcat, ja sitten nimellä tomcat-8.5.51. Ymmärrän heti, että edessä on suuri arkkitehtoninen ongelma. Delegoin tämän ongelman ratkaisemisen arkkitehdille ja jatkan itse toisen havaitsemani ongelman parissa: näyttää siltä, että playbookini asentaa sovelluksen myös vanhaan Tomcat-instanssiin uuden lisäksi.

Olisihan se pitänyt tällä kokemuksella arvata, että testauksessa tulee vielä vastaan jotain isoa. Oikein tarkkaan miettimällä molemmat ongelmat olisi myös voinut havaita etukäteen. Ärsyttää, etten vielä saakaan tikettiä päätökseen.

Tiistai

Päivän suunnitelma on edistää Tomcatin automaattista päivitystä. Ennen kuin saan aamun ensimmäisen kahvikupillisen eteeni, kollega kysyy, jos ehdin auttaa häntä myöhemmin päivällä. Mitään kiireellistä minulla ei ole, joten suostun toki.

Kahvikuppi tyhjenee ja täyttyy ja tyhjenee uudelleen pohtiessani, mistä ihmeestä sovellus ilmestyy vanhaan Tomcatiin. Playbookin ja roolin koodit on luettu jo pariin kertaan läpi ja muuttujien arvot on verifioitu oikeiksi. Yritän aikaansaada virheen muuttamalla Tomcatin webapps-hakemiston oikeuksia. Ei apua. Jossakin kohtaa sammutan Tomcatin puolivahingossa ja käynnistäessäni sen uudestaan teen ratkaisevan havainnon: sovellus ilmestyi Tomcatiin ilman, että ajoin playbookia! Syyllinen ei siis ollutkaan playbookini, vaan Tomcatin konfiguraatio. On vain yksi paikka, josta sovellus saattoi tulla. Muistan, että sovellus kopioidaan Tomcatin alle ensin webappsBase-hakemistoon, josta Tomcat itse asentaa sen webapps-hakemistoon ajoa varten. Olin luullut, että sovellus poistetaan webappsBase-hakemistosta sen jälkeen, kun se on saatu käyntiin. Testaan toimintaa ja havaitsen, että Tomcat poistaa sovelluksen molemmista paikoista, mikäli poistan sen webappsBase-hakemistosta. Korjaus tulee todennäköisesti olemaan yksinkertainen.

Kollegallani on vähintään yhtä erikoinen ongelma. Hän on toteuttanut playbookin Postinumero-nimisen sovelluksen asentamiselle. Playbook toki pohjautuu Tulostuspalvelu-nimisen sovelluksen playbookiin, mutta on olennaisilta osiltaan muutettu niin, että kaiken järjen ja ymmärryksen mukaan sen pitäisi asentaa Postinumero-sovellus. Seuraan ruudunjaon välityksellä, miten playbook kuitenkin yrittää asentaa Tulostuspalvelu-sovelluksen. Heitän kollegalleni muutamia ideoita siitä, missä vika voisi olla mutta kaikki on niiltä osin kunnossa. Playbookin group\_vars-hakemistossa on tiedostoja sekä .yaml-päät-

teellä, että ilman. Parempien ideoiden puutteessa googletan Ansiblen dokumentaation ja etsin sieltä tietoa, onko .yaml-päätteen puuttumisella jotain vaikutuksia (How to build your inventory, kappale Organizing host and group variables). Tuota tietoa etsiessä huomioni kiinnittyi kuitenkin johonkin ihan muuhun seikkaan.

Tässä kohtaa kalenteri hälyttää daily-palaveria ja hyppäämme kollegan kanssa lennosta Slackista Google Meettiin jakamaan ajatuksia arkkitehdin kanssa. Kerron löytäneeni ensinnäkin ratkaisun omaan ongelmaani ja erittäin lupaavan teorian kollegani ongelmaan. Toisin kuin minä, kollegani, ja arkkitehti olimme luulleet, Ansible ei mätsää group\_vars-hakemistossa olevia muuttujakokoelmia ja hosts-tiedostossa määriteltyjä palvelinryhmiä keskenään, vaan jokaista yksittäistä palvelinta kohden Ansible etsii kaikki muuttujakokoelmat group\_vars-hakemistoista, yhdistää niitten muuttujat, ja käyttää tätä yhdistettyä settiä muuttujien lähteenä.

Analyysini osoittautuu oikeaksi ja keskustelemme roolin alun perin toteuttaneen henkilön kanssa, miten tästä edetään. Konsensus löytyy siitä, että group\_varsissa määrittelyn sijasta tarjoamme muuttujat vars-parametreina roolille. Projektini arkkitehti on sitä mieltä, että mielenrauhan kannalta muutos kannattaa tehdä myös muihin samaa roolia käyttäviin playbookeihin. Olen samaa mieltä.

Iltapäivällä katsomme vielä kollegani kanssa hiukan tietokannan provisiointia kehitysympäristöön hänen tikettinsä pohjalta.

Tänään oli kaksi tapausta, jossa ongelma oli jossain ihan muualla, kun kuvittelin sen olevan. Minulle on tyypillistä, että oletan aina ensin ongelman olevan siinä osassa järjestelmää, jonka minä olen toteuttanut, mutta Ansiblea olen käyttänyt niin paljon, että on todella yllättävää löytää sen toiminnasta näin perustavanlaatuinen ero verrattuna siihen, mitä luulin sen tekevän.

## Keskiviikko

Tänään on joka kuukautinen henkilöstöpäivä, joten päätän lähteä toimistolle tapaamaan ihmisiä. Aloitan korjaamalla kuormantasaajan konfiguraation, joka on unohtunut saattaa ajan tasalle Tomcat-muutoksiin nähden. Tutkin myös hiukan, mistä johtuu, että Tomcatin parametrit pitää asettaa kahteen paikkaan. Päivittäminen olisi huomattavasti helpompaa, jos ne tarvitsisi tehdä vain yhteen. Yhtäkkiä arkkitehti kyselee minulta Slack-viestillä asioita, joiden merkitystä en oikein ymmärrä. Lyhyen viestinvaihdon jälkeen tulkitsen, että hän on yrittänyt kollegani kanssa testata erästä kirjautumista, eikä se ole toiminut. Kokeilen itse samaa, eikä se toimi minullakaan. Pallottelemme hetken "yritä tätä" - "ei toimi",

"koita tuota" - "sinne pääsen" sananvaihtoa, kunnes vaihdamme Slack-puheluun ja ruudunjakoon. Kokeilemme ruudunjaolla muutamia asioita, mutta kirjautuminen lokaalissa ympäristössä ei ota toimiakseen. Arkkitehdin arvaus on, että sovellus tarvitsee Tomcatin versiossa 8 jotain konfiguraatiota, jota se ei versiossa 7 tarvitse, sillä testi- ja kehitysympäristöissä kaikissa on käytössä versio 7, ja minä ja kollegani yritimme ajaa sitä versiolla 8.

Tässä kohtaa joudun siirtymään toimiston alakertaan työnantajan tarjoamaa lounasta varten, jonka jälkeen yritän provisoida muokatulla playbookillani sovelluksen Tomcatin versioon 7. Tämä on hyvä testi muutoksilleni, sillä se on lähellä reaali maailman skenaariota, jollainen voi tulla vastaan. Playbook toimii varsin hyvin, mutta itse sovellus ei käynnisty. Kyselen ongelmasta toiselta arkkitehdiltä samalla kun aloitan daily-videopuhelun. Dailyn jälkeen arkkitehti ja kollegani jäävät puheluun selvittämään kollegani ongelmaa liittyen ääkkösiä sisältävien kuntanimien hakuun hänen sovelluksessaan. Keskustelun omasta ongelmastani Slack-viestein toisen arkkitehdin kanssa ja jään seuraamaan puhelua toisella korvalla.

Hetken päästä arkkitehdin pitää siirtyä seuraavaan palaveriin ja jään kollegani kanssa kahden puheluun. Pyydän kollegaa tekemään kyselyn suoraan tietokantaan MySQL-komentorivisovelluksella. Näen ruudunjaosta, kuinka hänen kirjoittaessaan ääkkösen terminaali-ikkunaan ilmestyy vain kaksi kysymysmerkkiä. Itselläni on sama ympäristö ja kokeilen huvikseni itsekkin samaa. Minulla ääkköset toimivat normaalisti. Minulla on Linux-käyttöjärjestelmä ja kollegallani MacOS, ilmeisesti MacOSin terminaali käyttäytyy sopivalla tavalla eri tavalla. Päätän asentaa sovelluksen itsellenikin, todennäköisesti minun pitää se tehdä kuitenkin lähitulevaisuudessa.

Sovellus ei käynnisty. Virhelokista löytyy ilmoitus, jossa sovellus väittää, ettei se löydä taulua tietokannasta. Kurkkaan kantaan, ja taulu on kyllä siellä, mutta kirjainkoko eroaa. Kyseessä on ongelma, johon olen törmännyt aikaisemminkin. MySQL-kantoja luodessa pitäisi aina hoksata laittaa ensin päälle asetus `lower_case_table_names` jolloin ne toimivat luotettavammin erilaisissa ympäristöissä. Tuhoan kannan, annan provisioinnin luoda sen uusiksi ja sovelluskin käynnistyy.

Suorat kyselyt kantaan toimivat, mutta kyselyt sovelluksen kautta jättävät toimimatta samalla tavalla kuin kollegallani. Aikaisemmin työt aloittanut kollega päättää luovuttaa tältä päivältä ja jatkaa huomenna, ja itsekkin haluan pohtia vielä hetken omaa tikettiäni. En kuitenkaan pysty keskittymään omaan tikettiin koska saan idean, jota en malta olla kokeilematta: korvaan HTTP-kutsusta ääkköset URL-enkoodatulla versiolla ja saan vastauksena sellaiset tiedot kuin pitääkin. Raportoin havainnostani kolle-

galle Slack-viestillä ja pohdin vielä hetken, miksi ihmeessä omat kutsuni eivät mene perille Tomcatin 7-versiossa olevaan sovellukseen.

Toinen kollegani pingaa minua Slackissa ja kertoo saaneensa kirjautumisen toimimaan omassa ympäristössään. Hauskasti yksi toimenpiteistä oli aikaisemmin mainitun jaetun kannan poisto ja uudelleenluonti `lower_case_table_names`-asetus päällä. Muut toimenpiteet olivat samoja, joita olen itsekin jo kokeillut.

Henkilöstöpäivät ovat päiviä, jolloin normaalisti useimmat ihmiset tulevat toimistolle lähitöihin. En ole nähnyt muutamaan viikkoon kollegoitani livenä, joten ajattelin mennä moikkaamaan. Perustasoi-sesta koronatilanteesta huolimatta ihmiset olivat kuitenkin sen verran ujoja, ettei toimistolla montaa ihmistä ollut.

Torstai

Aamu alkaa tahmeasti kokiessani, että olen jo yrittänyt kaiken, mitä keksin. Analyysini mukaan HTTP-kutsu menee perille Tomcatiin asti, mutta vain sen HTTP-käsittelijään, ei servletille asti. Vertailen curl-työkalulla paikallisesta ympäristöstä saamiani vastauksia niihin, joita saan dev-ympäristöstä. Ainoa ero on, että dev-ympäristöstä tulee Set-Cookie-HTTP-otsake, jota paikallisesta ympäristöstä ei tule.

Raportoidessani havaintoni dailyssä arkkitehdille kypsyy päässä idea: voisihan hakea dev-ympäristöstä sen toimivan ympäristön ja vertailla omaa toimimatontani siihen. Arkkitehti pahoittelee, ettei ehdi täydeltä kalenteriltaan katselmoida ongelmaa ruudunjaolla, mutta ehdottaa tätä samaa ennen kuin ehdin itse tuoda ajatukseni julki. Lataan dev-ympäristöstä toimivan Tomcatin sovelluksineen ja vertailen sen tiedostoja diff-työkalulla. Lukuun ottamatta sitä, että paikallisessa ympäristössäni on muutamia tiedostoja, joita dev-ympäristössä ei ole, mikä havaittiin jo eilen, ei tiedostoissa ole eroja. Päivän loppuksi katsomme kollegan kanssa ruudunjaolla kuormanjakajan provisiointiin liittyviä asioita.

Perjantai

Kokeilen jotain, jota olisi pitänyt kokeilla jo aiemmin: yritän ajaa dev-ympäristöstä kopioitua Tomcat-ympäristöä sellaisenaan paikallisesti. Katsohan, sehän toimii täydellisesti! Kokeilen vielä muita kombinaatioita dev-ympäristön Tomcatista, paikallisesta Tomcatista, dev-ympäristön sovelluksesta ja pai-

kallisesta sovelluksesta, jonka jälkeen alan ymmärtää, ettei vika ole Tomcat-asennuksessa, vaan itse sovelluksessa. Dailyssä arkkitehti kommentoi, että hyvä havainto, ja että olisihan hänenkin se pitänyt hoksata jo aiemmin.

Tässä välissä käyn videopuhelussa toisen projektin aloituspalaverissa, minkä kehitysympäristöä pystytelimme viikko sitten. Palaverissa on mukana asiakkaan edustaja, jolta pyydämme lähtö- ja tulopään kuvaukset ja esimerkkiedostoja. Tämän viikkoiset merkistökoodaussekoilut tuoreena mielessä pyydän asiakkaalta, että hän sisällyttää testidataan ääkkösiä. Kaiken kaikkiaan projekti vaikuttaa varsin pieneltä ja suoraviivaiselta, vaikka sen käynnistyminen onkin ollut vähän pitkissä kantimissa. Ehkä ensi viikolla pääsen tähänkin sisään.

Katson, että lokaaliympäristön provisiointi ottaa sovelluksen paikallisesta Maven-repositorystä. Sovelluksen war-paketti siellä on peräisin helmikuulta 2019 minkälie aikaisemman projektini seurauksena. Kuulostaa ehkä antiikkiselta, mutta tiedän, ettei sovellukseen ole tullut muutoksia viime aikoina. Ajattelen silti, että varmistan, että koitan provisioida samaa versiota kuin mitä dev-ympäristössä on, ja annan Ansiblelle komentoriviltä argumentin `-e alfame_artifactory_host="https://maven25.prh.fi/artifactory"` joka lataa paketin paikallisen Maven-repositoryni sijasta asiakkaan yhteisestä Maven-repositorystä. Sovellus ei kuitenkaan edelleenkään suostu toimimaan ja yritän asentaa sitä niin playbookini asentamalle Tomcatille kuin dev-ympäristöstä kopioidullekin Tomcatille. Jossain vaiheessa hoksaan rullata ylöspäin Ansiblen tulostusta ja huomaan, että Ansiblehan edelleenkin asentaa sen saman vanhan sovellusversion paikallisesta Maven-repositorystäni, vaikka erikseen käskin sitä hakemaan paketin toisaalta. Luen playbookia läpi ja joudun toteamaan, että niinpä tietenkin, tässä projektissa maven-repositoryn osoite on muuttujassa nimeltä `alfame_artifactory_repository_url`, ei suinkaan `alfame_artifactory_host`. Korjaan muuttujan nimen, ajan playbookin ja saan virheen, että toista playbookin asentamaa pakettia ei löydy yleisestä maven-repositorystä. Tässä vaiheessa lyön hanskat naulaan ja siirryn perjantai-illan viettoon. Ainakin tiedän, mikä hyökkäysvektorini maanantaina tulee olemaan.

Tunnistan itsessäni ongelmaa, jossa ikään kuin laput silmillä pureudun juuri siihen kaninkoloon, jota tällä hetkellä olen tutkimassa. Usein olisi hyödyllistä osata irrottautua yksityiskohdista ja katsella ongelmaa vähän kauempaa. Omaksi puolustuksekseni, tällä kertaa sitä ei huomannut kukaan muukaan.



## Viikkoyhteenveto

Koko viikko oli vähän epämääräistä, hapuilevaa säätöä. Paitsi että viikontakainen optimismi tiketin valmiiksi saattamisesta haihtui heti maanantaina, kokonaisuus tuntuu olevan hiukan hukassa niin itseltä kuin tiimiltäkin. Viikolla oli myös paljon keskeytyksiä kollegoilta tulevien kysymysten ja erilaisten palaverien muodossa, mikä luonnollisesti vaikuttaa keskittymistä vaativien ongelmien ratkaisunopeuteen.

Viikon merkittävin opetus oli selkeästi Ansiblen käytös muuttujien osoittamisessa palvelinryhmille, mikä oli täysin päinvastainen siihen, mitä minä tai kukaan tiimiläisistä luuli sen olevan. On suorastaan uskomatonta, että näin suuri projekti on saatu toimimaan ymmärtämällä täysin väärin näin perustavanlaatuinen ominaisuus käytettävästä työkalusta.

Tutkittuani asiaa lisää minulle selvisi, että ongelman saa kierrettyä antamalla kullekin palvelimelle uniikki alias seuraavaan tapaan. Tulen jatkossa antamaan palvelimille aina uniikin aliaksen Ansible-projekteissa, jolloin Ansible osoittaa muuttujat palvelimiin intuitiivisella tavalla.

```
[muleservers]
```

```
muleservers_mule1 mule1
```

```
muleservers_mule2 mule2
```

```
[tulostuspalveluserservers]
```

```
tulostuspalveluserservers_mule1 mule1
```

```
tulostuspalveluserservers_mule2 mule2
```

Seuraavalla viikolla tavoitteena olisi saada työn alla oleva tiketti valmiiksi ja siirtyä seuraavaan projektiin.

### 3.4 Viikko 4

Maanantai

Tänään suunnittelen selvittäväni, miksi sovellus toimii Tomcat seiskassa, mutta ei Tomcat kasissa. Aloitan aamun selvittelemällä Maven repository -sotkuja. Asennettavanani on siis activiti-webapp-rest2-niminen web-sovellus, sekä alfame-processmanager-niminen, selaimessa pyörivä ReactJS-sovellus. Molemmat löytyvät kyllä Artifactorystä, mutta ihan eri repositoryistä. Ihmettelen, miten tämä provisiointi on koskaan toiminut dev- tai testiympäristöissä, mutta luovutan ja kopioin Artifactorysta uusimman activiti-webapp-rest2-sovelluksen omaan paikalliseen Maven-repositoryyni.

Lopultakin sovellus toimii omassa ympäristössäni. Mutta vain Tomcatin versiolla 7. Versiolla 8 se ei suostu edes käynnistymään. Käynnistymättömyyden syy on se, että erästä propertyä ei ole asetettu.

Asioiden olisi tarkoitus toimia siten, että sovelluksen mukana WAR-paketissa tulee sopivia oletusarvoja sisältävä versio asetustiedostosta. Nämä oletukset, ja muutama property joiden arvoa ei vielä kääntämisen aikana voida tietää, tulevat sitten Tomcatin conf-hakemistossa sijaitsevasta asetustiedostosta. Molemmissa Tomcat-ympäristöissä molemmat tiedostot ovat oikealla paikallaan ja sisältävät odotetut arvot, mutta näyttää siltä, että Tomcat 8 ei lue ollenkaan conf-hakemistossaan olevaa asetustiedostoa. Aloitan vertailemalla kaikki Tomcatin omat asetustiedostot versioiden 7 ja 8 välillä. Vertailussa ei kuitenkaan näy mitään sellaisia eroja, jotka viittaisivat asetustiedostojen lukemiseen conf-hakemistosta. Etsin käsiini Apachen viralliset dokumentit koskien siirtymistä versiosta 7 versioon 8 (The Apache Software Foundation. Apache Tomcat® - Migration Guide - Tomcat 8.0.x.) ja versiosta 8 versioon 8.5, mutta mikään niissäkään ei viittaa tähän ongelmaan (The Apache Software Foundation. Apache Tomcat® - Migration Guide - Tomcat 8.5.x.). Testaan vielä lopuksi kopioida conf-hakemiston asetustiedoston WAR-paketin sisälle ja nyt sovellus toimii kiltisti myös Tomcatin versiossa 8. Analyysini oli oikea, mutta ratkaisu odottaa vielä itseään.

Mainitsen dailyssä, että haluaisin saada tämän sotkun selvitettyä valmiiksi, ennen kuin aloitan toista projektia, mutta että saattaa olla, että joudun aloittamaan sitä jo huomenna. Arkkitehti kommentoi, että lienee parasta, että pysyttelenkin tämän projektin parissa ja kollegani, jolla ei tällä hetkellä ole laskuttavaa työtä, hoitaa sen. Resursointi on ehkä hiukan dynaamista, mutta sopii minulle ihan hyvin. Arkkitehti haluaa myös, että kokeilen asentaa muita samassa ympäristössä Tomcatissa pyöriviä sovelluksia. Hän on alkanut epäillä, etteivät nekään toimi Tomcatin versiolla 8.

Kollegani on saanut oman osuutensa siihen vaiheeseen, että hän suorittaa performansitestejä isokokoisilla tiedostoilla. Hän on huolissaan suorituskyvystä, ja lisäksi joutui kasvattamaan MySQL:n max\_allowed\_packet\_size-asetusta saadakseen ylipäättään testit läpi. Arkkitehti joutuu siirtymään seuraavaan pa-

laveriin, mutta sovimme, että toimin virkaatekevänä arkkitehtina ja katson kollegan kanssa koodit läpi ja avustan kannan asetusten säätämisessä. Aloitamme teoretisoimalla, miksi kysely on MySQL:lle liian suuri, vaikka siinä kulkeva tiedosto, jopa base64-enkoodattuna, on huomattavasti pienempi. Täyteen varmuuteen emme pääse, mutta huomaamme ainakin, että Activiti tallentaa sen kantaan *kahdesti*. Seuraavaksi luntaamme `max_allow_packet_size`-testipalvelimelta. Se on yksi gigatavu, suurin, mitä MySQL tukee ja enemmän kuin riittävästi 60-megatavuisille testitiedostoille.

Yllätys on kuitenkin, kun tuo suurin, 60-megatavuinen, ei kuljekaakaan prosessin läpi, vaan vastaukseksi tulee joku HTTP-virhe. Käy ilmi, että nginx-palvelimessakin on oma rajansa kyselyn koolle, ja tuo suurin tiedosto ylitti sen. Tämä oli uutta tietoa minullekin.

Tiistai

Aloitin hankkimalla pari muuta sovellusta, joilla voin testata Tomcatin toimintaa. Käännän niistä itseleni paketit ja juuri kun aion alkaa provisioimaan niitä, kollega kyselee Slackissa, mikä tilanne sovellukseen kirjautumisessa on. Hänen testaustaan helpottaisi suuresti, mikäli pääsisi käsiksi tähän käyttöliittymään. Koska sain eilen kirjautumisen toimimaan itselläni, otan tämän hyvänä testitilanteena ja ehdotan, että testaamme provisiointia hänen ympäristössään, mikä sopii. Yrittäessäni selittää mihin kohtaan provisioinnissa Tomcatin versionumero kannattaa määritellä huomaan, ettei etukäteen ajattelemani tapa välittää versionumero playbookista muuttujana Tomcatin asentavalle Ansible-roolille toimikaan, sillä versionumeroa tarvitaan myöhemmin vielä myös itse sovelluksen asentavassa roolissa. Teen henkisen muistiinpanon selvittää tätä myöhemmin. Saamme asennusprosessin siihen pisteeseen, että itse sovellus pyörii Tomcatissa ja huomaan, että se on sama toimimaton versio, joka minullakin oli. Kopioimme kollegallekin version Artifactorystä mutta kuinkas ollakaan, ei sovellus toimi edelleenkään. Pyydän kopion ympäristöstä itselleni tutkimuksia varten ja ohjeistan, miten kirjautumisen saa toimimaan.

Tutkin kollegani ympäristöä vertaillen sitä omaani ja ainoa löytämäni ero on, että omassa toimivassa versiossani sovelluksen sisällä classpathissa oleva asetustiedosto sisältää puuttuvat arvot. En osaa selittää, mistä se sinne on päätynyt, mutta todennäköisimmin olen epähuomiossa jossain kohtaa testaamista kopioinut sen sinne käsin itse ja sittemmin unohtanut sen. Haastattelen yhtä arkkitehtia siitä, miten Tomcatissa olevat sovellukset asetukset löytävät ja saan vastaukseksi selostuksen, miten se tehdään Spring Boot -sovelluksessa ja eräässä omassa toteutuksessa. Tutkin hetken activiti-webapp-rest2:n koodia, eikä se ole kumpaakaan. Löydän koodista kohdan, jossa asetukset luetaan, eikä siinä mikään

viittaa asetusten lukemiseen mistään classpathin ulkopuolelta. Arkkitehti varmistaa epäilyni. Eiliseen ihmettelyyni siitä, miten tämä koskaan on toiminut, saan nyt siis vastauksen: ei se olekaan.

## Keskiviikko

Eilen havaitun ongelman voisi ratkaista ainakin kahdella tavalla: muokkaamalla provisiointia niin, että se kopioi asetustiedostot sovelluksen classpathiin tai muokkaamalla itse sovellusta niin, että se yrittää lukea niitä Tomcatin alla olevasta conf-hakemistosta. Kyselin Slackissa arkkitehtien mielipidettä ja saankin kohta Slack-puhelun yhdeltä. Hetken se ottaa, että saan selitettyä ongelmani. Arkkitehdin mielipide kallistuu Ansible-toteutuksen suuntaan, mutta toteamme pian, että suoraan asennettuun sovellukseen asetustiedostojen kopiointi ei onnistu, ja että WAR-pakettiin niiden uittaminen olisi vähintäänkin työlästä ja riskialtista. Arkkitehti kuitenkin onneksi huomaa, että Tomcatin kontekstimäärittelyyn voi asettaa niin WAR-paketin kuin ns. “exploded WAR”, eli paketin sisällön hakemistossa. Hyväksytän idean toisella arkkitehdillä, jonka mielestä se on hyvä idea, ja aloitan toteutuksen.

Muutos kuulostaa melko pieneltä, mutta Ansiblen verboosin syntaksin takia siitä tulee kohtalaisen iso. Ansiblessa ei esimerkiksi voi purkaa zip-pakettia uuteen hakemistoon, vaan hakemisto pitää luoda etukäteen erillisellä taskilla. Lisäksi minulle tulee yllätyksenä, että esimerkiksi copy-moduuli kopioi oletuksena kontrollikoneelta kohdekoneelle, ei kohdekoneen sisällä. Näitä remote\_src-asetuksia saankin lisätä useille taskeille testaamisen jälkeen. Saan aikaan toteutuksen, johon olen suhteellisen tyytyväinen ja applikaatio asentuu paikalleen. Kello on kuitenkin sen verran paljon, että tarkempi testaaminen ja testaaminen muilla applikaatioilla jää huomiseksi.

## Torstai

Ennen kuin alan testaamaan muilla applikaatioilla mietin, puuttuuko toteutuksesta vielä jotain. Hoksaa, että eiliset muutokseni vaativat lisämuutoksia siivouslogiikkaan, jossa vanhoista Tomcateista poistetaan asennettava sovellus. Lisäksi keksin, että automaattisesti voidaan siivota myös kontekstikonfiguraation sisältävät tiedostot. Toteutan muutokset, mikä sujuukin mukavasti. Testaan asennusta sekä Tomcatin versiolla 7, että versiolla 8 ja molemmat tuntuvat toimivan.

Refaktoroin parin muun sovelluksen playbookit rooleihin tekemiini muutoksiin ja kokeilen provisiointia. Määrittelyistä puuttuu pari muuttujaa, jotka lisään roolin oletuksiin. Muuttujien lisäämisen jälkeenkään tämä sovellus ei kuitenkaan vaikuta toimivan. Huomaan, että kontekstipolku tulee väärin. Paitsi että siinä ilmenee versionumero "latest", joka pitäisi korvaantua oikealla versionumerolla on koko paketin nimi väärin. Tarkemmat tutkimukset näistä suoritetaan huomenna.

Perjantai

Aamusta iskee katastrofi. Luulin kaapissa olevan täyden paketin kahvin suodatinpusseja, mutta siellä ei olekaan. Vain yksi suodatinpuski jäljellä. Sen voimalla alan selvittämään, mistä väärä kontekstipolku tulee. Hetken muuttujamäärittelysuossa kahlaamisen ja aikaisempiin git-revisioihin vertailun jälkeen huomaan, että olen parissa paikkaa käyttänyt muuttujaa `apache_tomcat_deploy_artifact_name` kun olisi pitänyt käyttää muuttujaa `apache_tomcat_deploy_artifact_id`. Korjaan muuttujanimet, ajan molempien sovellusten provisioinnin ja se toimii. Käyn lähikaupassa hakemassa lisää kahvin suodatinpusseja.

Tuoreen kahvin voimin jatkan testaamista ja kokeilen kolmatta sovellusta. Sen provisiointi menee muuten hyvin, mutta sovellus ei käynnisty. Lokista löytyy jokin Liquibase-virhe, jolla tuskin on yhtään mitään tekemistä provisioinnin kanssa. Päätän jättää sen huomiotta ja keskityn siivoamaan Ansiblen inventorya päällekkäisistä muuttujista. Olen rooleja ja playbookeja refaktoroidessa yrittänyt tehdä asiat niin, että Tomcatin version, sekä porttioffsetin saisi asetettua vain yhdessä paikassa, per sovellus tietysti, mutta historiallisista syistä muuttujamäärittelyjä on vähän missä sattuu. Kommentoin pois kaikki paitsi yhden, ja provisiointi toimii edelleen hienosti. Tässä yhteydessä huomaan, että sen sijaan, että Tomcat pitäisi pakettia webappsBase-hakemistossa ja siitä ajokopiota webapps-hakemistossa se ajaa sovellusta suoraan webappsBase-hakemistosta. Missäköhän kohtaa olen tällaisen konfiguraatiomuutoksen saanut aikaan? Sen jätän maanantaille selvitettäväksi.

Viikkoyhteenveto

Dynaamisen resursoinnin ansiosta jatkan vastoin suunniteltua edelleen samassa projektissa, mitä pidän itse ihan positiivisena asiana. Projektin vaihtaminen nyt olisi tuntunut asioiden kesken jättämiseltä.

Asiat kuitenkin etenevät tuskallisen hitaasti ja tuntuu, että jokaisen ongelman jälkeen seuraa vain uusi, entistä mystisempi, ongelma.

Jokainen ongelma toki on mahdollisuus oppia jälleen uutta. Tiedän nyt, että myös Nginx:ssä on konfiguraatio paketin maksimikoolle ja että Ansiblen copy-moduuli oletuksena kopioi kontrollikoneelta kohdekoneelle. Tiedän nyt myös enemmän Tomcatin deploy-asetuksista kuin olen koskaan halunnut tietää. Toivon vain, että tälle tiedolle tulee käyttöä joskus myöhemminkin.

### 3.5 Viikko 5

Maanantai

Aloitan tutkimalla, miksi Tomcat muutti toimintansa niin, että se ajaa sovellukset webappsBase-hakemistosta webapps-hakemiston sijaan. Tomcatin konfiguraatio ei ole muuttunut, joten suunnistan dokumentaation äärelle. Luettuani hetken Tomcatin appBasesta ja docBasesta törmään referenssisivuun Tomcatin automaattisesta deploysta (The Apache Software Foundation 2018). Sehän juuri on, mitä haluan! Sivun ei mielestäni ole hirvittävän helppolukuinen, mutta tavattuani sen läpi jää minulle se käsitys, ettei Tomcatin automaattista deployta saa toimimaan kuten haluan. Siinä missä unexploded WAR-paketin kanssa Tomcat voi joko ajaa WAR-pakettia sellaisenaan tai purkaa sen toiseen hakemistoon ja ajaa sieltä, exploded WAR-paketin kanssa Tomcat osaa ainoastaan ajaa sen sieltä, missä se on. Selostan tilanteen Slack-viestissä ja pingaan kahta arkkitehtia. Ei vastausta.

Odotellessani palautan paikallisen ympäristöni tilaan, joka vastaa sitä, mikä tuotannossa on tällä hetkellä. Testaan Tomcatin päivitystä ja sehän epäonnistuu. Tomcat-hakemiston symbolinen linkkaaminen lyhyemmälle nimelle epäonnistuu koska playbook on jostain kumman syystä jo luonut sinne hakemiston sillä nimellä.

Dailyssä saan lopultakin kiinni toisen arkkitehdeista, joka on sitä mieltä, että unohdetaan koko webappBase ja asennetaan sovellukset suoraan webapps-hakemiston alle. Tämä vaatii luonnollisesti myös toisen arkkitehdin hyväksynnän. Lisäksi lupaan testata, että kontekstipolun kustomointi, joka oli alkuperäinen syy webappsBase-hakemiston käytölle, toimii. Palaan kuitenkin edellisen ongelmani pariin.

Testaan activiti-webapp-rest2-sovelluksella, jolla Tomcatin päivitys ja symbolisen linkin luominen sujuu ongelmitta. Epäilen, että jossakin muuttujassa on väärä arvo, joten tulostan playbookilla kaikki muuttujat juuri ennen virhekohtaa sekä activiti-webapp-rest2-sovelluksella, että toisella sovelluksella. Diff-työkalulla tarkasteltaessa erot ovat hämmentävän pieniä, eikä niistä ole apua ongelman ratkaisuun. Päätelen, että ongelman täytyy olla itse playbookeissa. Tarkastelen sekä toimivaa, että hajoavaa vierekkäin ja hetken mielijohteesta siirrän Tomcatin asentavan `include_tasks`-käslyn yhden paikan ylöspäin. Kappas, playbookhan toimii! Tutkin tarkemmin, mitä alaspäin siirretty käsky oikein tekee. Kuvaus on "varmistetaan, että paikallinen indeksipaketihakemisto on olemassa", ja se luo hakemiston, jonka arvo luetaan muuttujasta nimeltä `cti_dsview_api_ftp_upload_local_directory`. Arvaan tuloksen jo tässä vaiheessa, mutta tarkistan kuitenkin muuttujan arvon. Kyllä, se on Tomcatin hakemiston alla.

Symbolisen linkin luonnin epäonnistumisen tutkintaan meni kyllä turhankin paljon aikaa, mutta kun mikään sen kuvauksessa eikä muuttujanimesä viitannut siihen, että se loisi hakemiston kohdekoneen Tomcatin sisään, en osannut oikeaa ratkaisua alkuunkaan epäillä.

Tiistai

Viestittelen toisen arkkitehdin kanssa Slackissa webappsBase-muutoksesta. Pääsemme yhteisymmärrykseen siitä, että voin provisioida ajettavan version webapps-hakemistoon, mutta arkkitehti haluaisi edelleen "varmuuskopion" webappsBase-hakemistoon perustelunaan halu nähdä heti, mikä versio sovelluksesta on ajossa ja kyky tarvittaessa uudelleenasettaa sen suoraan palvelimella ajamatta provisiointia. Suostun tähän ja alan muokkaamaan playbookien toiminnallisuutta.

Kohdassa, jossa playbookin pitäisi poistaa webapps-hakemistossa ajossa oleva versio ei tehdä mitään, koska Ansible ei ole edellisessä kohdassa löytänyt poistettavaa hakemistoa. Tarkistan uudelleen kohdehakemiston ja hakukaavan sekä varmistan, että palvelimella oikeasti on haettava hakemisto, noin kymmeneen kertaan. Seuraavaksi kokeilen `find`-moduulin parametreja, kuten haettua tiedostotyyppiä, tai tulkitaanko hakukaava säännöllisenä lausekkeena vai ei. Lisäksi kokeilen erilaisia variaatioita hakukaavasta, mutta Ansible ei edelleenkään löydä hakemistoa. Jossakin kohtaa palvelimella tiedostoja selatessani otan puolivahingossa pitkän tiedostolistauksen Tomcatin hakemistosta, josta havaitsen virheen: webapps-hakemistolla ei ole riittäviä oikeuksia. Ansiblella ei ole oikeutta listata sen sisältöä. Korjaan oikeudet, ja varmistan, että provisiointi asettaa ne oikein ja vanhan ajossa olevan version poistaminen onnistuu.

Illalla on vielä palvelumuotoilun jatkokurssin viimeinen luento, jossa ryhmät esittelevät, mitä on saatu aikaan.

Ainakin opin, että Ansiblen virheilmoituksiin, tai ehkä pikemminkin niiden puuttumiseen, ei voi luottaa.

## Keskiviikko

Jatkan roolin refaktorointia tehtävä kerrallaan. Testaan suunnilleen jokaisen muokkauksen jälkeen, mikä vaikuttaa hyvältä idealta. Jossakin vaiheessa huomaan, että webapps-hakemistossa sovellus on yhden ylimääräisen hakemistotason sisällä. Tämä korjataan Ansiblen copy-moduulissa lisäämällä yksi kauttaviiva kopiolähteen perään. Lopulta olen tyytyväinen tulokseen ja siivoan koodin debug-tulosteista ynnä muusta vastaavasta roskasta ennen versiohallintaan laittamista.

Kollega haluaa testata omaa playbookiaan, joka käyttää muokkaamaani roolia, täysin puhtaalta pöydältä. Seuraan ruudunjaon välityksellä edistystä ja olen suorastaan yllättynyt, että ainoa kohta, johon provisiointi keskeytyy, on kun se haluaisi purkaa WAR-paketin unzip-työkalulla, jota puhtaaseen ympäristöön ei ole asennettu. Lisään provisiointiin riippuvuuden paketin asennuksesta ja jatkamme hetken kollegan ympäristön läpikäyntiä. Päivitän dev-ympäristön Ansible inventoryn siihen kuntoon, että kuvittelen sen toimivan, mutta ennen kuin ehdin alkaa asentamaan, kollega mainitsee, että hän olisi utelias näkemään miten dev-ympäristöön asennus oikein tapahtuu. Sovimme, että teemme sen yhdessä ensimmäiseksi huomenna.

## Torstai

Heti aamusta otan kollegan kanssa videopuhelun, jossa näytän, miten dev-ympäristöön asennus tapahtuu erillisen provisioner-koneen kautta. Prosessi sujuu muuten hyvin, mutta Ansible ei saa ladattua pakettia Artifactorystä, koska sillä ei ole tunnuksia sinne. Paikallisessa ympäristössä olenkin asettanut, että paketti haetaan paikallisesta Maven-repositorystä, mutta ihmettelen, miten Artifactoryyn nyt yhtäkkiä tarvitsee tunnukset, koska tiedän, että sieltä saa ladattua ilmankin. En jaksa alkaa selvittämään syytä, vaan kopioin tunnukset toisen sovelluksen inventorystä ja lisään ne tämän sovelluksen inventoryyn ja prosessi menee loppuun asti.

Kuten tuotantoympäristökin, dev-ympäristö on jaetta kahteen osaan, asiakkaan sisäverkossa olevaan, sekä ulospäin näkyvään. Activiti-webapp-rest2-sovellusta käytetään molemmissa, joten katson seura-



vaksi ulkoverkkopuolen inventoryn kuntoon ja koitan asentaa sovelluksen sinne. Sovellus asentuikin nätisti ja näyttää käynnistyvän virheittä, mutta en jostain syystä pääse kirjautumaan sinne. Dailyssä arkkitehti kertoo, että syy on todennäköisimmin se, ettei ulkoverkon puolella olevasta palvelimesta ole yhteyttä sisäverkossa olevaan Active Directoryyn. Tietysti. Saan sovellukseen paikalliset admin-tunnukset, joilla kirjaudun onnistuneesti sisään.

Dailyn jälkeen oli tarkoitus asentaa kollegani kanssa hänen sovelluksensa myös dev-ympäristöön, mutta kun vilkaisemme palvelimelle, ei siellä ole riittävästi vapaata keskusmuistia sovelluksen vaatimuksiin. Teemme kollegani kanssa tiketin IT-tukeen, josta vastaus tulee käytännössä välittömästi: "milloin saan uudelleen käynnistää palvelimet?" Olisi pitänyt arvata, että muistin lisäys vaatii uudelleenkäynnistyksen. Sovin arkkitehdin kanssa, että viestin tukeen, että dev-ympäristön saa käynnistää heti kun heille sopii, ja muista palvelimista sovitaan sitten erikseen myöhemmin.

Nyt kun rooli on verifioitu toimivaksi, koetan iltapäivästä vielä asentaa toisen sovelluksen dev-ympäristöön. Se taas ei asennu virheittä, ja tutkin, että ilmeisesti ainakin Tomcatin vanhassa versiossa olevan sovelluksen poistamisessa on vielä jokin ongelma.

Perjantai

Aloitan siitä, mihin eilen jäin ja tutkin miksi sovellus ei poistu vanhasta Tomcatista. Havainnoin, että kyllä sovellus poistetaan webapps-hakemistosta, mutta se ilmestyy sinne hetken päästä uudelleen. Hetkinen, tällainen ongelmahan oli jo joskus aikaisemminkin! Kun katson webappsBase-hakemistoon, siellä todellakin on unexploded WAR-paketti, josta Tomcat palauttaa sovelluksen, kun sen webapps-hakemistosta poistaa. Tarkastelen sovelluksen poistavaa Ansible-roolia ja huomaankin bugin sovelluksen poistossa webappsBase-hakemistosta, joka ei paikallisessa ympäristössä exploded war-paketteja poistaessa ollut ilmennyt. Korjaan virheen ja yritän uudelleen. Provisiointi menee onnistuneesti läpi, mutta selaimen kautta sovellusta testattaessa se ei toimi. Tutkin sovelluksen inventoryä ja löydänkin sieltä muutaman virheellisen muuttujan. Korjaan ne, provisiointi menee edelleen läpi, mutta sovellus ei edelleenkään toimi.

Hetken kulmienkurtistelun jälkeen hoksaan katsoa Tomcatin lokiin. Näyttää, että sovellus ei käynnisty Liquibase-virheen takia. Näyttää, että joku on muuttanut Liquibase-changelogia sen kantaan ajamisen jälkeen. Kyselen Slack-kanavalla, jos joku muu on saanut vastaavaa virhettä aikaisemmin. Tässä välissä pidämme muutaman kollegan kanssa noin tunnin mittaisen session palvelumuotoilukoulutukseen liittyen, joka sattuu hyvään väliin, sillä session loputtua syyllinen on ilmoittautunut ja korjannut ma-

nuaalisesti kannan vastaamaan Liquibasen tilaa. Ajan vielä provisioinnin uudelleen ja se, sekä sovellus toimivat nätisti.

## Viikkoyhteenveto

Työntäyteinen viikko, mutta onneksi hiljalleen alkaa näyttämään siltä, että kaikki bugit on saatu korjattua, niin pääsee hiljalleen muihin puuhiin. Mitä enemmän Ansiblea käyttää, sitä enemmän alkaa kaipaamaan täysremonttia sen (virhe)ilmoituksiin. Maanantainenkin ongelma olisi ratkennut heti, jos Ansible suvaitsi kertoa ne absoluuttiset polut ja tiedostonimet, mitä ja minne se yrittää hakemistoa luo- da. Ansiblen muokkaaminen ei tietenkään ole realistinen optio, mutta tilannetta voisi mahdollisesti selkeyttää käyttämällä kuvaavampia muuttujanimiä. Suurin osa muuttujista ei kuitenkaan ole minun nimeämiäni, eikä käytäntö, jossa yksi muuttuja koostuu useasta muuttujasta, jotka koostuvat useasta muuttujasta ja niin edelleen edesautaa asiaa. Välissä on liian monta kerrosta epäsuoria viittauksia.

## 3.6 Viikko 6

### Maanantai

Aloitan testaamalla kolmatta sovellusta ja sekin toimii oikein hyvin. Seuraavaksi katsomme arkkitehdin kanssa ruudunjaolla ylläpitotehtävän. Erääseen sovellukseen lisätään sallittuihin osoitteisiin uusi yhteistyökumppani. Samalla poistamme käytöstä erään toisen sovelluksen vanhasta ympäristöstä, joka on poistumassa käytöstä. Noin kymmenen minuuttia poiston jälkeen asiakasvastaava postaa Slack-kanavalle Nagios-valvontatyökalun virheilmoituksen siitä, että kyseisen sovelluksen tuotantopalvelin on alhaalla. Unohdimme arkkitehdin kanssa, että sovellus on automaattivalvonnan piirissä. Teen pikaisen tiketin helpdeskille valvonnan käytöstä poistamisesta.

Käy ilmi, että minulla ei ole oikeuksia muokata huoltokatkolokia asiakkaan Confluence-wikisovelluksessa. Asiakasvastaava tekee tästä tiketin helpdeskille, ja odottaessani oikeuksia alan katsomaan testin, qa-ympäristön ja tuotannon Ansible-inventoryjä siihen kuntoon, että uskallan ajaa playbookit näissä ympäristöissä. Dailyyn mennessä oikeuksia minulla ei ole, joten arkkitehti lisää puolestani huoltokatkoilmoitukset testille ja qa:lle huomiselle ja tuotantoon keskiviikolle.

Loppupäivän käytän sellaisen Ansible-playbookin tekemiseen, joka sammuttaa Tomcatit, joissa ei ole yhtään sovellusta ajossa.

Tiistai

Tänään päivitetään Tomcatit test- ja qa-ympäristöihin. Käyn ensiksi läpi vielä Ansiblen inventoryt ja kun olen tyytyväinen niihin, kirjaudun provisioner-koneelle ja käynnistän playbookin sisäverkon test-ympäristöä vasten. Provisiointi pysähtyy siihen, ettei Artifactoryn osoitetta ole määritelty. Hyvin tarkistettu... Sen lisättyäni provisiointi menee kuitenkin nätisti läpi ja koetan avata sovellusta selaimen. Ei toimi. No mitäs nyt? Tarkistan Tomcatin lokit, joista käy ilmi, että sovellus on käynnistynyt onnistuneesti. Seuraavaksi tarkistan, mitä porttia Tomcat kuuntelee ja mihin porttiin Nginx liikenteen yrittää ohjata. Oikein nämäkin. Koetan palvelimella ladata sovelluksen juuresta tiedoston suoraan Tomcatilta: HTTP-virhe 404. Tutkin vielä hetken lokeja ja mietin mahdollisia asioita, jotka voisivat olla väärin, mutta ideat loppuvat, joten huutelen Slack-kanavalla ihmisiä apuun.

Arkkitehdillä sattuu kuin sattuu olemaan aikaa, joten esittelen Slackin ruudunjaon välityksellä hänelle ongelman. Näytän sovelluksen kirjautumisruudun, jossa on HTTP 404-virheeseen kaatunut kirjautuminen ja demonstroidakseni, kuinka se ei toimi, painan kirjautumisnappia uudelleen. Sovellus kirjautuu sisään. Tuijotan ruutua suu auki noin kymmenen sekunnin ajan ja valmistaudun antamaan elämäni selityksen siitä, miten sovellus varmasti ei toiminut vielä hetki sitten kun sitä kokeilin. Ennen kuin ehdin aloittaa, arkkitehti kuitenkin kertoo saaneensa idean. Samantapaisia virheitä on kuulemma ollut myös toisten sovellusten kanssa, ja arkkitehti epäilee syyksi kuormanjaon konfiguraatio-ongelmaa. Tarkistamme, että kuormanjaosta tosiaan puuttuu yksi asetus, joka eräässä toisessa, toimivassa, sovelluksessa on.

Asetuksen puuttumisella on suurempikin merkitys kuin jakelua yrittävän ohjelmistokehittäjän harhauttaminen: sovellukset ovat kahdennettuja, ja jakelu on suunniteltu niin, että noodit päivitetään yksi kerrallaan niin, että toinen niistä on aina pystyssä palvelemaan asiakkaita ja näin välttyään huoltokatkolta. Ilman asetusta palvelu on hetken alhaalla ja huoltokatkoton päivitys ei onnistu kuten suunniteltu. Arkkitehti pyytää, että selvitän tämän yhdessä kollegani kanssa, jonka vastuulla toinen viallinen sovellus on, ja sovinkin hänen kanssaan videopalaverin heti lounaan jälkeen.

Ennen palaveria koodaan äkkiä bash-skriptin, joka sekunnin välein tekee kutsun sovellukseen ja tulostaa 1, mikäli palvelu on pystyssä ja 0, mikäli se ei ole. Selostan kollegalleni tilanteen ja konfiguroin Ansible-projektini templateihin Nginx:n proxy\_next\_upstream -asetuksen. Käynnistän provisiointin sekä skriptini. Pitkän ykkösrivin keskelle ilmestyy pari nollaa. Kirjaudun palvelimelle ja tarkistan

nginx:n konffit. Ne ovat juuri sellaiset kuin halusinkin, joten tarkistan nginx:n lokit seuravaaksi. Sieltä ilmenee, että eihän proxy\_next\_upstream-asetusta käytetä alkuunkaan niin kuin olin käyttänyt, joten tarkistan oikean syntaksin nginx:n dokumentaatiosta ja yritän uudelleen (Module ngx\_http\_proxy\_module, kappale proxy\_next\_upstream). Kurtistelen kulmiani, kun ykkösrivin keskelle edelleen ilmestyy muutama nolla. Eikö asetus vieläkään ole oikein? Onneksi minulla pian välähtää: jotta huoltokatkoton päivitys toimii, pitää sen tietenkin olla käytössä, ja niin ajaessani provisioinnin uudelleen, se toimiikin moitteettomasti.

Samaan aikaan arkkitehti on tutkinut, mitä versioita sovelluksista pitäisi asentaa ja käy ilmi, että on olemassa uudempikin kuin se, mitä Artifactorystä olen asentanut. Saan tehtäväksi kääntää uuden pake-tin, laittaa sen Artifactoryyn ja aloittaa testit uudelleen paikallisesta ympäristöstä. Samalla sovimme, että yhdistän muutokseni versiohallinnassa päähaaraan. Aloitan siitä, että yhdistän päähaaran omaan haaraani, ratkaisen lukuisat konfliktit ja testaan provisiointia paikallisesti. Activiti-webapp-rest2-sovel-luksen provisiointi toimii, mutta toisia sovelluksia asentaessani saan kuitenkin vanhan tutun tietokanta-virheen. Se, että poistan tietokannan ja annan provisioinnin luoda uuden, pitäisi ratkaista ongelma, mutta kun teen niin, saan vähän erinäköisen tietokantavirheen. Arvelen, että paketit, joita asennan, ei-vät ole ajan tasalla, mutta jätän ajantasaisien pakettien hankinnan suosiolla seuraavalle päivälle.

Tänään oli taas malliesimerkki päivästä, joka kääntyi täysin pääläelleen kesken päivän ja kasasi ison määrän tangentialisesti edelliseen tehtävään liittyviä uusia tehtäviä pinon päälle.

## Keskiviikko

Aloitan selaamalla versiohallintaa ja etsimällä uusinta versiota sovelluksesta. Jonkin eilisen Slack-viestin perusteella nappaan itselleni haaran nimeltä "release-alfame-1", ja käänän siitä paketit. Ma-ven-kääntö ei kuitenkaan tee pakettia esimerkiksi hiljattain paljolti käyttämästäni activiti-webapp-rest2-sovelluksesta. Syy tähän on yksinkertainen: sovellusta ei mainita Maven-käännön konfiguraatio-tiedostossa. Tämä saa minut epäilemään, että käytössäni on väärä haara. Kysyn arkkitehdiltä asiasta ja saan vastaukseksi kysymyksen, haluaisinko selvittää asiaa ruudunjaon yli.

Katsomme arkkitehdin kanssa hetken sekä versiohallintaa, että Artifactoryssä olevia paketteja. Nykyti-lanteen voisi tiivistää yhteen sanaan: sekasotku. Otammekin sovelluksen master-haaran, ja yhdistämme sen pariin muuhun haaraan ja luomme "release-alfame-2"-haaran, jossa pitäisi olla viimeisimmät koo-dit. Myös sovelluspakettien versioita nostetaan ykkösestä kakkoseen, ja lisäksi -SNAPSHOT-suffiksi

poistetaan. Versionumeromuutosten jälkeen yritämme kääntää sovellusta, mutta saamme virheen toisensa jälkeen. Myös sekä Artifactoryn pakettien jaottelua, että sovelluksen useaan paikkaan konfiguroituja repositoriokonfiguraatioita voisi kutsua sekasotkuksi. Arkkitehdin kanssa katsomme näitä ja konsultoimme toista arkkitehtia ja saamme jonkinlaista selkoa tilanteeseen.

Yrittäessä kääntää sovellusta se kuitenkin yrittää nyt ladata meidän itsemme tekemää paketti julkisesta Maven Central -repositoriosta. Jäljitämme repositoryasetuksia käyttämällä versiohallinnan web-käyttöliittymää, josta näkyy, että käyttämässämme versiossa on määritelty repositoriot juuri niin kuin haluamme niiden olevan. Jossain vaiheessa katsomme kehitysympäristöni paikallisesta Maven-repositoriosta samaa tiedostoa, ja sieltä repositorioasetukset jostain syystä puuttuvat. Lopulta kopioimme listan repositorioista parent-pomiin ja sovellus kääntyy. Kysyn toiselta arkkitehdiltä, tietääkö hän asiasta mitään. Vastaus on, että Artifactory toimii noin tarkoituksella, jottei kesken käännön alettaisi hakemaan eri repositorioista paketteja. Tietämättämme korjasimme siis ongelman samalla tavalla kuin oli tarkoituskin.

Kääntöprosessi etenee, mutta kaatuu hiukan myöhemmin siihen, ettei kehitysympäristönä käyttämässäni virtuaalikoneessa ole asennettuna nodejs-tulkkiä. Pakettimanagerissa on tarjolla vain antiikkinen versio, joten päädyimme siihen, että otan käyttöön virallisesti tämän sovelluksen kehittämiseen tehdyn kehitysympäristön. Pystytän Vagrantilla uuden virtuaalikoneen, johon siirrän työn alla olevan koodin ja yritän jälleen kääntää sitä. Prosessi ei pääse niinkään pitkälle kuin viimeksi, vaan kaatuu siihen, että log4j2-utils-paketin versiota 1.0.6-SNAPSHOT ei löydy Artifactorystä. Tarkistan, ja Artifactoryssä kyllä on paketti tällä nimellä ja versiolla. Hetken aikaa asiaa pohdittuamme päädyimme arkkitehdin kanssa siihen, että kääntöprosessi todennäköisesti etsii SNAPSHOT- ja ei-SNAPSHOT-paketteja eri paikoista. Vaihdan paikalle version 1.0.5 ja prosessi eteneekin. Tällä kertaa pääsemme samaan nodejs:n puuttumisesta johtuvaan virheeseen kuin aiemminkin. Ilmeisesti virallinen kehitysympäristökään ei ole ihan ajan tasalla. Kehitysympäristön provisioinnista löytyy mahdollisuus asentaa nodejs, mutta versio on prekambriakaudelta. Asennamme sen kuitenkin oletuksella, että koodi on samaa ikäluokkaa.

Pitkään kestävän käännön jälkeen nodejs-kääntö kaatuu virheeseen, joka haiskahtaa vahvasti siltä, että käytössä on liian vanha nodejs:n versio. Kyselen Slack-kanavalla, mitä versiota nodejs:stä pitäisi käyttää. Vastausta odotellessa katsomme kollegan kanssa hänen ongelmaansa, jossa JVM:ltä loppuu aina välillä muisti. Tarkempaa selkoa ongelmaan emme saa, mutta pohdiskelun lomassa päivitän kehitys-

ympäristöni nodejs:n aavistuksen vähemmän vanhaan versioon, ja saan käännettyä sovelluksen onnistuneesti.

Jo toinen päivä syvemmälle kaninkoloon sukeltamista. Jokohan huomenna pääsisi takaisin asiaan? Kaikki ei kuitenkaan ollut huonoa, sillä opin tänään varmaan yhtä paljon Mavenin konfiguroinnista kuin edellisenä viitenä vuotena yhteensä. Yleensä projekteissani joku muu on hoitanut kääntöprosessiin liittyvät ongelmat.

Torstai

Kollega halusi lainata USB-SATA -kiintolevykelkan, niin suuntaan aamusta pitkästä aikaa toimistolle. Julkaisuversio on nyt jäässä, joten muutan kehityshaarasta versionumerot snapshoteiksi ja pusken muutokset, ettei kukaan epähuomiossa jyrää yli tekemiäni julkaisupaketteja. Katsomme arkkitehdin kanssa yhdessä, miten Mavenin deploy-toiminto toimii ja kuinka sen pitäisi toimia. En ole vielä konfiguroinut deployn asetuksia loppuun asti, mutta jo ensimmäinen deploy-komento lataa paketteja Artifactoryyn. Hups. Selvitelyämme arkkitehdin kanssa asiaa hetken toteamme kuitenkin, että ne ovat sentään halutussa paikassa, vaikka vahingossa sinne menivätkin. Emme selvitä loppuun asti, mistä Maven repositiorion tiedot haki, vaan lopetan ruudunjaon ja konfiguroin Mavenille toisen, salasanaa vaativan, Artifactory-repositorion tiedot, jotta saan loputkin paketit lähetettyä Artifactoryyn.

Kollegani on kysellyt Slack-viesteillä omaan tikettiinsä liittyviä asioita muutamaan otteeseen aamupäivän aikana, joten ehdotan, että katsomme asiaa yhdessä ruudunjaon välityksellä. Kollegani haluaisi analysoida JVM:n muistinkäyttöä, mutta JVM:stä ei saa otettua heap dumpia jonkinlaisen oikeusongelman takia. Käy ilmi, että käyttäjän, jolla JVM:ää ajetaan, ryhmä on eri kuin ryhmä, joka omistaa JVM:n tiedostot. Analysoin tämän provisiointiongelmaksiksi. Provisioinnissa ei JVM:ää ajavalle käyttäjälle määritellä oletusryhmää, joten käyttäjänluonnissa luodaan uusi ryhmä, josta tulee käyttäjän oletusryhmä. Tiedostojen oikeudet kuitenkin ovat toisella ryhmällä. Teemme niin sanotun quick & dirty -korjauksen provisiointiin, joka vahvistaa hypoteesini ja korjaa ongelman.

Nykyisessä provisioinnissa käyttäjää luodessa sille annetaan lista ryhmiä, joiden jäseneksi käyttäjä liitetään. Ehdotan vähemmän nopeaksi ja likaiseksi korjaukseksi sellaista, että listan ensimmäinen ryhmä asetetaan aina oletusryhmäksi, ja loppuihin käyttäjä lisätään normaalisti. Konsultoimme provisioinnin alunperin toteuttanutta arkkitehtiä, joka hyväksyy suunnitelman ja päivän päätteeksi vielä toteutamme sen.

Kohtalaisen opettavainen päivä tänäänkin, joskin omia töitani pääsin edistämään harmillisen vähän. Maven deploy-toimintoa en ole aikaisemmin itse konfiguroinut, ja olen jo pitkään halunnut ottaa selvää, miten Javan muistinkäyttöä analysoidaan.

## Viikkoyhteenveto

Kulunut viikko oli varsin opettavainen erityisesti Mavenin konfiguroinnin saralla. Vaikka kyseinen kääntötyökalu on käytössä melkein jokaisessa projektissa, jossa olen ollut mukana, on tarkempi tutustuminen siihen aikaisemmin jäänyt varsin vähälle. Aikaisemmissa projekteissa on joko käytetty vain perusominaisuuksia tai joku muu on konfiguroinut tarvittavat asetukset, eikä minun ole tarvinnut koskea niihin.

Kehitysympäristöjen hallinta on siitä haastavaa, että mikäli projektiin ei tule uutta ohjelmoijaa, sellaisia tarvitsee yleensä pystyttää noin kolmen vuoden välein. Tästä seuraa, että useammin tarvittavat toiminnot ajavat tärkeysjärjestyksessä kehitysympäristöjen hallinnan edelle, ja niiden dokumentointi ja automatisointi laahaa perässä työajan ja resurssien rajallisuuden vuoksi. Tässä projektissa sentään yritettiin, mistä kunnia edeltäjilleni, vaikkei ympäristö ajan tasalla ollutkaan. En odota, että asiat tulevat tällä saralla muuttumaan miksikään lähitulevaisuudessa, vaikka yleisesti laajempi dokumentaatio ja automatiikka työntekoa helpottaakin. On hölmöä, että jokaisen tarvitsee ratkaista samat ongelmat aina uusiksi, kun joku muu on jo keksinyt ratkaisun, ja parhaassa tapauksessa automatisoinut sen.

## 3.7 Viikko 7

### Tiistai

Muistellessani, mihin oikein jäinkään ennen pitkää viikonloppua, hoksaan, että en koskaan tehnyt valmiiksi playbookia, joka sammuttaa Tomcatit kun niissä ei ole enää sovelluksia ajossa, joten otan sen työn alle. Testattuani paikallisesti kommitoin muutokset ja huomaan, että olinkin vielä vanhassa haaraassa. Vaihdan develop-haaraan, yhdistän vanhan tikettikohtaisen haaran sinne, pusken muutokset eteenpäin ja valmistaudun provisioimaan muutokset dev-ympäristöön. Kirjaudun sisään provisionerkoneelle, ja vedän uusimmat muutokseni ja päivitän alimoduulit ja saan Gitiltä virheilmoituksen, ettei viittauksen kohteena olevaa objektia ole olemassa palvelimella. Hoksaan, että edellisellä viikolla tekemäni develop-haaraan yhdistämiset olivatkin itse sovelluksen repositoriossa, eikä provisioinnin reposi-

toriossa. Yhdistän tikettihaaran muutokset develop-haaraan, eikä se tietenkään mene ilman konflikteja. Huokaus.

Tässä välissä käyn dailyssä, jossa arkkitehti huomauttaa, että Tomcatit sammuttavan playbookin pitäisi myös poistaa Tomcatiin liittyvä service niin, ettei se käynnisty enää koskaan uudelleen. Hyvä huomio. Jatkan kuitenkin provisioinnin testaamista, jossa ilmeneekin ongelma: yksi testaamistani playbookeista ei mene läpi. Analyysini mukaan se yrittää hankkia pääkäyttäjän oikeudet ja asentaa paketteja työasemalleni, eikä suinkaan kohdekoneelle. Tällaiseen ongelmaan en ole ennen törmännyt.

Samaan aikaan kollega kyselee, miksei hänen sovelluksensa provisiointi mene läpi. Provisiointi riippuu paketeista, jotka luulin julkaisseeni torstaina, mutta eipä noita näytä Artifactorystä löytyvän. Ajattelen julkaisevani ne pikaisesti ja ajankin Maven deployn paketeille. Maven ei edes yritä julkaista niitä Artifactoryyn, josta alkaa pitkälinen selvitys syystä. Erilaisten repositorioasetusten vääntelyn jälkeen hoksaan, että Maven ei ylipäätään aja maven-deploy-pluginia, jonka tuo julkaisu pitäisi tehdä. Saan lopulta asetukset kohdalleen ja paketit julkaistua, mutta kollegani yrittäessä kääntää sovellusta kaatuu se virheeseen, jossa junit-kirjastoa ei löydy. Kääntäessäni saman koodin itse se kääntyy ongelmitta. Kokeilemme molemmat uutta puhdasta työtilaa ja edelleen kollegallani sovellus ei käänny, mutta minulla kääntyy. Tässä vaiheessa kello näyttää kuitenkin sen verran paljon, että loput ongelman selvityksestä jätämme huomiselle.

Päivän töistä voi oppia ainakin sen, että poistaessa palvelimilta jotain, pitää siivota jälkensä loppuun asti ja poistaa myös poistettavaan asiaan liittyvät muut, turhat jäänteet.

Keskiviikko

Jatkan eilisen versiohallinta-aharojen yhdistämisen aiheuttaman ongelmani selvittelyä. Vertailen playbookia, joka ei toimi, toiseen, joka toimii. Paketteja asentavan roolin kutsumisessa ei ole mitään sellaisia eroja, jotka voisivat selittää eroja. Huomioni kiinnittyy siihen, että toimimattomassa playbookissa ennen pakettien asentamista on tehtävä, joka suoritetaan paikallisesti. Itse pakettien asennustehtävä suoritetaan kokonaan toiselle host-ryhmälle, mutta kokeilen kommentoida paikallisen tehtävän pois ja playbook menee virheettä läpi. Paikallisesti suoritettavan tehtävän tarkoitus on kopioida SSH-avain kohdepalvelimelle, mitä ei ole pakko tehdä playbookilla, sillä saman voi tehdä käsinkin, ja se tarvitsee tehdä vain kerran. Päädyn poistamaan tehtävän.



Toteutan vanhat Tomcatit sammuttavaan playbookiin ominaisuuden, että se poistaa käytöstä palvelun käyttöjärjestelmätasolla niin, ettei se käynnisty palvelimen uudelleenkäynnistyksen yhteydessä. Testaan toiminnallisuutta paikallisesti ja se toimii. Nyt uskallan suorittaa asennuksen sekä dev- että test-ympäristöihin, niin sisä- kuin ulkoverkonkin puolelle. Dev-ympäristöön asennus sujuu ongelmitta, mutta test-ympäristössä törmään siihen, ettei siellä ole ylipäättäen asennettuna kaikkia sovelluksia. Kurkkaan myös qa-ympäristöön ja siellä ei ole asennettuna yhtään ainoaa sovellusta, jolloin sovelluksien päivittämisen testaaminen siellä on hiukan hyödytöntä. Tässä kohtaa kello on jo varsin paljon ja kalenterissani on huomiseksi arkkitehdin kanssa palaveri, joten jätän tarkemmat selvittelyt sinne.

Torstai

Aamun aluksi kurkkaan myös tuotantopalvelimelle. Mielenkiintoisesti ulkoverkon puolella palvelua ei ole kahdennettu ja se pyörii vain ykköskoneella. Asennukseni jälkeen palvelu tulee olemaan kahdennettu, mikä nyt ei ainakaan haittaa. Ehdin myös kokeilla kääntää kollegallani työn alla olevaa sovellusta niin, että poistan Mavenin paikallisen pakettivälimuistin. Nyt saan saman virheen kuin kollegani, mutta en ehdi alkaa ratkoa sitä ongelmaa vaan otan videopuhelun arkkitehdin kanssa. Samassa videopuhelussa on toinenkin arkkitehti, joka tulee siirtymään tähän projektiin tulevaisuudessa. Annan lyhyen selvityksen siitä, millainen arkkitehtuuri tässä projektissa on ja mitkä Ansible-käytännöt ovat. Selostan myös havaintoni qa- ja tuotantoympäristöistä.

Ensimmäinen arkkitehti huomauttaa, että tuotantoon pitäisi provisoida master-haarasta, joten päättämme yhdistää muutokset develop-haarasta vielä master-haaraan. Tämä ei kuitenkaan suju alkuunkaan. Näyttää, että master-haara on hurjan paljon jäljessä develop-haaraa, ja lisäksi yhdistämisestä tulee runsaasti konflikteja. Koetamme yhdessä ratkoa näitä hetken aikaa, mutta konflikteissa on niin paljon sellaisten projektien asioita, joiden kanssa kumpikaan meistä ei ole ollut koskaan tekemisissä, että luovutamme, teemme tiketin ja annamme sen eteenpäin.

Seuraava asiallinen kysymys kuuluu "mikä versio sovelluksesta provisoidaan"? Vastaus on "latest", josta käy kuitenkin ilmi, että Artifactoryn mielestä -alfame-2 versio ei ole "latest". Käyn eksplisiittisesti asettamassa asennettavan version 5.22.0-alfame-2:ksi ja ajan playbookin ja se epäonnistuu. Löysin playbookista bugin versionumeron käsittelyssä.

Arkkitehdeillä alkaa tässä kohtaa toinen palaveri, joten sanomme näkemiset ja jatkan provisiointiongelman selvittämistä ja testaamista paikallisessa ympäristössä. Päivän lopuksi asennan sovelluksen vielä tuotantoympäristöön, sekä sisä- että ulkoverkon puolelle. Lopultakin.

Perjantai

Otamme heti aamusta kollegan kanssa videopuhelun ja parikoodaamme kuntoon puuttuvat riippuvuudet esb-utils-kirjastosta. Lukuisten muutosten jälkeen arkkitehti toteaa, että aiemmin julkaistu alfame-2-versio on susi, ja kannattaa julkaista uusi alfame-3 -versio, jossa riippuvuudet toivottavasti ovat kohdallaan. Parikoodaamme ja testaamme tarvittavat muutokset. Sovellus toimii mainiosti, mutta tulostaa lokiin tietoturvaongelmaan viittaavia tulosteita. Tähän kollegani on jo aiemmin saanut arkkitehdiltä vinkin, että kyseessä lienee Mule-alustan bugi, joka on korjattu myöhempään 3.9.5-versioon. Opastan kollegalleni, miten uusi Mule-versio otetaan käyttöön ja teen ennakoivasti pienen muutoksen playbookiin.

Emme kuitenkaan pääse playbookin ajamiseen asti, sillä kääntäminen uudemmilla Mule-riippuvuuksilla epäonnistuu. Vaadittavia riippuvuuksia ei löydy repositoriosta. Käytössämme on peilattu kopio Mulesoftin repositoriosta, ja pohdin, mahtaakohan peilauksessa olla tapahtunut jokin virhe. Kyselen aiheesta projektin Slack-kanavalla, ja arkkitehti tarttuu kommenttiini ja liittyy puheluun. Katselemme asiaa hetken aikaa kolmestaan ja arkkitehti tekee päätöksen hylätä tuo 3.9.5-versio. Kyseessä on ilmeisesti jonkinlainen puolikas julkaisu tai virhejulkaisu, niin tai näin, käyttöön sitä ei selkeästi ollut tarkoitettu.

Aloitan myös omaa tikettiäni, joka on samanlaisen automaattipäivityksen tekeminen Mule-versioille kuin mitä aikaisemmin tein Tomcat-versioille.

Viikkoyhteenvedo

Kulunut viikko oli monipuolinen ja sisälsi paljon parikoodausta ja videopuheluita arkkitehtien kanssa, mikä on hyväksi havaittu tapa saada vaikeisiinkin ongelmiin ratkaisuja. Valitettavasti vain arkkitehdit tупpaavat kovin usein olemaan hyvin kiireisiä.

Viime viikolla esiin nostamani kehitysympäristöasia nosti taas päätään. Kollegan näennäisesti identtissä ympäristössä sovellus ei kääntynyt, koska sen riippuvuuksista puuttui paketti. Minulta, ja arva-

tenkin sovelluksen alkuperäiseltä kehittäjältä, puuttuva paketti kuitenkin löytyi jo valmiiksi välimuistista, eikä sen puuttumista näin huomannut.

Lisäksi tuotantoon provisioiminen master-haaran provisioinnilla on hyvä ja kannatettava ajatus. Valittavasti vain tuotantoon provisioidaan niin harvoin, että master-haara ehtii hapantua, mikäli siihen ei säännöllisesti yhdistetä muutoksia develop-haarasta. Tähän auttaisi selkeämmät käytännöt, milloin minkin commit viedään master-haaraan, mitkä jonkin kiireisen arkkitehdin pitäisi määritellä.

Toinen ongelma on, että joihinkin sovelluksiin kosketaan niin harvoin, ettei niiden develop-haaran provisioinninkaan kehitys pysy matkassa. Yritän jatkossa aktiivisesti puhua sen puolesta, että sovelluksia päivitetäisiin useammin, edes JDK- ym. tietoturvapäivitysten vuoksi. Useammin päivittäminen auttaisi pitämään provisioinnin lisäksi sovelluksen koodikannan ja sen riippuvuudet enemmän ajan tasalla, jolloin jonkin muutoksen tekeminen ei vaatisi niin valtavaa pohjatyötä.

### 3.8 Viikko 8

Maanantai

Jatkan Mulen automaattipäivityksen parissa ja tarkoitukseni on etsiä listasta Mule-hakemistoja provisioitava olevan sovelluksen instansseja. Huomaan, että playbookissa on myöhemmin käytetty varsin verboosisti nimettyä `mule_esb_deploy_artifact_pattern_filename_base`-muuttujaa, joka nimen perusteella vaikuttaisi olevan juuri se, mitä haluan. Etsin `grep`-työkalulla, missä muuttujan arvo on määritetty, mutta yllätyksekseni muuttujalle ei löydy määrittelyä Ansiblen vars tai defaults-tiedostoista, sitä vain käytetään muutamassa paikassa. Katson niistä yhtä ja se on `include_role`-blokki, joka kuvauksensa perusteella vaikuttaisi lataavan paketin Artifactorystä kontrollikoneelle. Hetken ihmeteltyäni asiaa päätän kokeilla tulostaa muuttujan arvon ennen ja jälkeen tätä `include_role`-blokkia ja kuinkas olla, blokin suorittamisen jälkeen muuttuja on olemassa ja sillä on fiksunoloinen arvo. Ilmeisesti kyseinen `maven_artifact`-rooli asettaa muitakin vastaavan kaltaisia muuttujia myöhempää käyttöä varten. En varsinaisesti haluaisi ladata pakettia vielä tässä vaiheessa playbookin suorittamista, mutta ei siitä haittaakaan ole, joten siirrän `maven_artifact`-roolin suorituksen pikkuisen aikaisemmaksi ja jatkan töitä.

Kollegani kyselee Slack-viesteillä vinkkejä oman tikettinsä ratkaisemiseksi. Hän käyttää myös aiemmin tekemääni Tomcat-roolia, jolla provisiointi epäonnistuu tutunolaiseen virheeseen. Kehotan häntä

yhdistämään master-haaran kehityshaaraansa ja yrittämään uudelleen. Hetken kuluttua provisiointi onnistuu, mutta hänen sovelluksensa kaatuu jonkinlaiseen signaalintivirheeseen. Katson tätä hetken aikaa kollegani kanssa, mutta hänen sovelluksensa ei ole minulle tuttu, enkä ole Activitin signaalintien kanssa ollut itse aikaisemmin tekemisissä. Kutsummekin arkkitehdin apuun, mutta hänen pitää jatkaa seuraavaan palaveriin ennen kuin ratkaisu löytyy. Jatkan vielä hetken oman tikettini suorittamista.

Tiistai

Jatkan Mulen automaattipäivityksen toteuttamista. Huomaan, että itseasiassa sovelluksen playbook, jota olen testimielessä ajanut, asentaa eri version Mulesta kuin pitäisi. Asetan muuttujat kohdalleen, jolloin Mulesta asentaa oikea versio. Tämä ei kuitenkaan käynnisty, sillä käyttöjärjestelmän Systemd-service osoittaa edelleen Muleen, joka kohdekoneella oli jo aikaisemmin. Muutan playbookin asentamaan Mulen Systemd-servicen versioituna, samalla tavalla kuin Tomcatin Systemd-servicet aikanaan. Lisäksi lisään tuen verkkoporttioffsetille, jotta Muleja voisi olla käynnissä useampi yhtä aikaa.

Arkkitehti mainitsee dailyssä sovelluksen, jota voisi käyttää testaamisessa. Ajan playbookin ja se epäonnistuu. Korjaan pari pikkuvikaa ja ajan playbookin uudestaan. Tämä playbook riippuu toisesta playbookista, jota aiemmin käytin Tomcat-roolin kehittämiseen ja testaamiseen ja joka nyt on jo tuotannossa ajossa. Nyt saan kuitenkin virheen, jonka muistan nähneeni kollegani saaneen joskus aikaisemmin. Hän siivosi silloin jotakin kohdekoneelta ja yritti uudelleen, jonka jälkeen provisiointi onnistui. Teen tästä henkisen muistiinpanon ja kierrän ongelman kommentoimalla pois kaksi ehtoa. Seuraavaksi huomaan kuitenkin, että eihän tämä playbook käytä edes roolia, jota olen muokkaamassa, vaan vanhempaa versiota. Alan porttaamaan playbookia uudelle roolille, mutta kello näyttää sen verran paljon, että siirrän loput töistä huomiseksi.

Keskiviikko

Hoidan sovelluksen playbookin siihen kuntoon, että voin ajaa sen. Provisiointi pysähtyy hyvin äkkiä siihen, että se ei löydä itse sovellusta paikallisesta Maven-repositoriostani. Lataan koodit itselleni ja yritän kääntää niitä. Teknistä velkaa on näköjään tässäkin projektissa, eikä se käänny ensimmäisellä yrityksellä. Virheilmoitusta googlaamalla ratkaisu vaikuttaa olevan mule-toolsin version päivitys. Uusi kääntöyritys, ja nyt saan Apache CXF:ään liittyvän virheilmoituksen. Etsin jälleen virheilmoituksella tietoa Googlesta, mutta tällä kertaa mitään ratkaisevaa vinkkiä ei löydy. Kokeilen huvikseni päivittää projektin käyttämän maven-cxf-pluginin antiikkisen version uusimpaan. Tämä yritys kaatuu siihen,

että uusinta versiota ei löydy käyttämästämme Maven-repositoryn peilistä. Vaihdan version uusimpaan, joka sieltä löytyy ja nyt kääntö onnistuu.

Provisiointi etenee siihen pisteeseen asti, että Mulea yritetään käynnistää. Se epäonnistuu. Keksinkin pian, että syy on tunnettu bugi, joka on läsnä myös Tomcat-versiossa. Provisioitaessa uudella playbookilla sitä samaa versiota Mulesta, tai Tomcatista, joka jo on ajossa palvelimella vanhalla playbookilla asennettuna muodostuu tilanne, jossa samaa Mulea yritetään käynnistää kahteen kertaan. Korjaus olisi työläs ja bugiin ei ainakaan vielä ole törmätty oikeassa käytössä. Pelkään kuitenkin, että Mulen kanssa mahdollisuus saattaa olla suurempi kuin Tomcatin. Käynnistän Mulen käsin ja tsekkaan lokin: Mule on käynnistynyt onnistuneesti, mutta sovellus sen sisällä ei.

Sovellus valittaa epäkelvosta XML-syntaksista. Tiedostoa, josta sovellus valittaa ei kuitenkaan löydy sovelluksen sisältä. Jossain vaiheessa hoksaan, että tiedosto on alfameesb-utils-paketin sisällä. Päivitän senkin uusimpaan versioon ja nyt sovellus käynnistyy.

Torstai

Jatkan playbookini debuggaamista. Mulen päivittäminen ei nyt onnistu, sillä playbook ei löydä muutujaa, jolla määritän Mulen version, vaan käyttää aina roolin oletusasetuksista löytyvää. Varoittamatta ruudulle ilmestyy ilmoitus Slack-puhelusta. Linjan toisessa päässä ovat kollegani ja arkkitehti, jotka ovat asentamassa erästä sovellusta test-ympäristöön Tomcatiin aiemmin tekemääni roolia käyttämällä. Ongelma on, ettei Systemd-service saa käynnistettyä Tomcat-binääriä. Tarkistamme, että tiedoston oikeudet ovat kunnossa ja että Systemd-servicessä on määritetty oikea käyttäjä. Hetken ihmettelyn jälkeen hoksaan, että tällä palvelimella on käytössä SELinux. Tarkistamme tiedostojen SELinux-kontekstit ja niistä löytyy vika. Koodaamme ja testaamme Ansible-toteutuksen SELinux-kontekstien asettamiseen.

Provisiointia ajaessa törmäämme samaan ongelmaan, johon törmäsin itse tiistaina: joissain tilanteissa sovellus ei kopioidu kohdekoneelle ja kun playbook sitten yrittää kopioida konfiguraatitiedostoja sen sisään, se epäonnistuu. Saamme jäljitettyä virheen paketteihin, joiden versiossa on kirjaimia. Keksimme kiertotavan ja teen sen korjaamisesta tiketin. Refaktoroimme hiukan playbookeja ja tuplavarmistamme, että tunnukset ja muut vastaavat ovat oikein. Tässä vaiheessa arkkitehdin pitää poistua seuraavaan palaveriin.

Jään kollegani tueksi ja laitamme provisioinnin käyntiin. Se pysähtyy hetken päästä virheeseen, että tietokannan taulua ei voi luoda, koska se on jo olemassa. Tarkistamme kannan ja kyllä siellä tosiaan on muutama rivi parin kuukauden takaa. Olimme kaikki luulleet, että sovellusta ei ole koskaan provisioitu tähän ympäristöön. Kannan luomisellekin playbookissa on kyllä tarkistus, ettei sitä yritetä luoda, mikäli se on jo olemassa, mutta selvästikään se ei tällä kertaa toiminut. Selvitämme, että vika on siinä, että kannan nimi on kovakoodattuna yhdessä kohtaa ja se on eri test-ympäristössä kuin paikallisesti.

Ajamme playbookin uudestaan ja se menee onnistuneesti läpi. Yritämme testata sovellusta, mutta se ei vastaa. Lokista selviää, että sovellus ei saa yhteyttä omaan tietokantaansa. Tässäkin löytyy provisioinnista virhe, jossa tietokannan osoite asetetaan tavalla, joka toimii vain lokaalisti. Uuden provisiointierroksen jälkeen sovellukseen ei edelleenkään saa yhteyttä, mutta lokista seuraamalla se vaikuttaisi toimivan. Kokeilemme ladata etusivua curl-työkalulla suoraan palvelimelta ja sieltä tulee hyvännäköinen vastaus. Arkkitehti muistelee, että palomuriavaukset on pyydetty aikaisemmin, mutta pyytää että tekisin aiheesta tiketin helpdeskillle. Tikettiä luodessani löydän oman pari kuukautta vanhan tikettini, jossa pyydetään ihan samaa asiaa.

Päivän opetus: tarkista aina SELinuxin päälläolo.

Perjantai

Testaillessani playbookia huomaan, että aina välillä kohdekoneeltani poistuu applikaation sijaan koko Mule. Tämähän on selkeä bugi, joka vaatii perehtymistä. Ei kestä kauaa löytää, että yhdessä kohtaa käytetään vahingossa ylemmän tason polkua, eikä sen alta etsittyä itse sovelluksen polkua. Samalla kiinnitän kuitenkin huomiota siihen, että poistamista ei itseasiassa tehdä oikein. Oikea tapa poistaa Mule-sovellus on poistaa sen "ankkuri", tekstitiedosto, jonka poistamisen Mule havaitsee ja käynnistää oman sisäisen undeploy-prosessinsa. Korjaan playbookini tekemään näin ja testaan muutokset.

Seuraavaksi taklaamani asia liittyy siihen, että olen havainnut, että Mule-sovellus käynnistyy oikein itseasiassa vain joka toinen kerta. Testimetodini on jokaisen ajon jälkeen vaihtaa käytettävän Muleen versiota 3.8.1:n ja 3.9.0-hf2:n välillä, jolloin joka toinen kerta sovellus deployataan 3.8.1-version Muleen ja joka toinen kerta 3.9.0-hf2-version Muleen. Näyttää siltä, että aina kun on version 3.9.0-hf2 vuoro, sovellus ei käynnisty. Provisioin sovelluksen tähän versioon ja avaan lokitiedoston. Siellä oleva virhe kertoo, että parametrisedulle kyselylle annetaan parametri, jota ei käytetä itse kyselyssä. Ilmeisesti vanhemman version sallittu käytös on uudemmassa tehty virheeksi. Poistan käyttämättömän paramet-

rin, käännän itselleni uuden version sovelluksesta ja kokeilen uudestaan provisioida uudempaan Mule-versioon. Sovellus ei edelleenkään käynnisty, ja tällä kertaa virheilmoitus näyttää siltä, että se on huomattavasti työläämpi korjattava. Itse provisiointi vaikuttaa siltä, että se kyllä toimii, joten jätän sovelluksen korjaamisen muille ja otan työn alle toisen sovelluksen playbookit.

Sovelluksen provisiointi käyttää jo valmiiksi uusia rooleja, joten yritän vain suoraan ajaa sitä. Tämä ei onnistu ja hetken aikaa asiaa tutkittuani totean, että syy on siinä, että samaan tapaan kuin Tomcat-playbookeja joskus aikaisemmin refaktoroidessani, tässäkin on samanlainen logiikka, jossa käytetään Ansiblen `import_playbook`-toiminnallisuutta, joka vaatii, että määrittelen kaksi settiä kohdekoneita, joille pitää antaa samat asetukset kahteen kertaan. Alan refaktoroimaan playbookeja käyttämään `import_tasks`-toiminnallisuutta `import_playbookin` sijaan, kunnes kello alkaa osoittaa viikonloppua alkavaksi.

## Viikkoyhteenveto

Ansiblen `include_role`-toiminnallisuus oli minulle uutta. Havaitsin vasta myöhemmin tutkiessani asiaa uudestaan, ettei epäselvä asia ollut `include_rolen` toiminnallisuus, vaan Ansiblen muuttujien määrittely ja näkyvyysalue. Kun koodinpätkää tarkastelee tarkemmin huomaa, että "vars"-osiohan ei ole `include_rolen` alla laisinkaan, vaan suoraan Ansiblen taskin alla. Vars-osio tässä toimii käteväenä oikotienä, joka kuvaa playbookissa käytetyt muuttujanimet roolin sisällä käytettyihin muuttujanimiin, jotka normaalisti tulisivat inventorystä.

```
- name: "Download Maven artifact"
  include_role:
    name: shared-roles/maven_artifact
  vars:
    maven_artifact_id: '{{ mule_esb_deploy_package_artifact_id }}'
    maven_artifact_group_id: '{{ mule_esb_deploy_package_group_id }}'
    maven_artifact_package_extension: '{{ mule_esb_deploy_package_extension }}'
    maven_artifact_output_pattern: mule_esb_deploy_artifact_pattern
    maven_artifact_output_pattern_filename_base:
mule_esb_deploy_artifact_pattern_filename_base
    maven_artifact_output_result: mule_esb_deploy_artifact
    maven_artifact_output_result_filename_base: mule_esb_deploy_app
    maven_artifact_force: '{{ mule_esb_deploy_force }}'
    maven_artifact_optional_params:
```

```
version: '{{ mule_esb_deploy_package_version | default( "" )}}'  
source_repository_url: '{{ mule_esb_deploy_source_repository_url |  
default( "" )}}'  
source_repository_username: '{{ mule_esb_deploy_source_repository_username |  
default( "" )}}'  
source_repository_password: '{{ mule_esb_deploy_source_repository_password |  
default( "" )}}'
```

SELinux on jo klassikoksi muodostunut mystisten ongelmien aiheuttaja. SELinuxin päällä olon voi tarkistaa ajamalla palvelimella yhden komennon, mutta se pitää hoksata erikseen tarkistaa. Onneksi SELinux-ongelmat kuitenkin yleensä ovat varsin suoraviivaisia ratkaista, kunhan niitä hoksaa etsiä.

### 3.9 Viikko 9

#### Maanantai

Tällä viikolla kohdalleni osuu ylläpitoviikko, joka tarkoittaa, että tehtävänäni on monitoroida ylläpidon sähköpostilaatikkoa ja reagoida sinne mahdollisesti tuleviin viesteihin. Aamusta on lyhyt palaveri viime viikolla vuorossa olleen kollegan kanssa. Edellinen viikko on ollut rauhallinen eikä keskeneräisiä tapauksia jää minun vuorolleni. Tarkistan laatikon ja siellä on viesti erään asiakkaan edustajalta, että heidän valvontansa näyttää, että muutaman palvelimen levytila on käymässä vähiin. Käyn etsimässä palvelimilta lokitiedostoja, jotka voisin pakata vapauttaakseni levytilaa. Kahdella niistä ei ole merkittäviä määriä lokeja, ja kahdella suurin osa lokeista on jo valmiiksi pakattuna. Kyselen projektin Slack-kanavalla, mitä asialle pitäisi tehdä.

Jatkan playbookin refaktorointia ja saan sen siihen kuntoon, että sovelluksen asennus näyttäisi toimivan. Vaihdan muutaman kerran Mullen versiota ja sovellus asentuu paikalleen, kunnes huomaan, ettei se itseasiassa käynnisty. Lyhyen lokin tutkimisen jälkeen totean, että pohjasyy on, ettei sovellus löydä erästä asetusta asetustiedostostaan. Käyn läpi Ansiblen templatetiedostot, mutta siellä kyseistä asetusta ei mainita ollenkaan. Syyppää on siis itse paketti. Löydänkin asetustiedoston, mutta en mitään, mikä asettaisi puuttuvan asetuksen. Kokemukseni mukaan tämä yleensä asetetaan käynnön aikana pom.xml:ssä, mutta tämän projektin pom.xml:ssä ei ole mitään siihen liittyen. Tarkistan toisen haaran, jonka pom.xml on selkeästi erinäköinen. Tämä haara ei kuitenkaan näytä kääntyvän. Kopioin tämän haaran pom.xml:stä mielestäni oleelliset osat, liitän ne viimeksi kääntyneen haaran pom.xml:ään, ja tulos on käännösvirhe. Haluan tässä vaiheessa vain sovelluksen käyntiin, joten puran sovelluksen käännösvirheen.



netyn paketin, muokkaan asetustiedostoon oleelliset osat käsin, ja asennan sen provisiointiskriptilläni. Lopultakin sovellus käynnistyy.

Samaan aikaan kollegani on reagoinut kyselyyn Slack-kanavalla ja kertoo, että kahden ensimmäisen palvelimen tapauksessa tulemme tarvitsemaan lisää tilaa, kahden jälkimmäisen tapauksessa kuukautta vanhemmat lokit saa poistaa. Kollegani tarjoutuu hoitamaan levytilauksen: sitä kun tarvitaan muillekin palvelimille. Itse käyn poistelemassa vanhoja lokitiedostoja jälkimmäisiltä palvelimilta.

Tiistai

Tänään on tarkoitus saada valmiiksi Muleen päivittävä Ansible playbook. Kun sovellus nyt toimii, testaan eilen esillenoussutta ongelmaa, jossa joskus playbook näyttää menevän onnistuneesti läpi, mutta oikeasti sovellus ei käynnisty. Käytän eilen korjaamani sovelluksen toimimatonta versiota testissä apuna. Playbookin pitäisi toimia niin, että mikäli käyttäjä yrittää provisioida rikkiäistä sovellusta toimivan päälle, playbook palauttaa automaattisesti ottamansa varmuuskopion. Nyt kuitenkin näyttää, että näin ei käy. Käyn läpi playbookin logiikkaa, jossa ei vaikuta olevan mitään ongelmaa, mutta monet tehtävät ovat ehdollisia, ja mikäli yksikin askel jää jostain syystä välistä, saattaisi se selittää ongelman. Analysoin ongelmaa ja näyttää siltä, että playbook kuvittelee, ettei Muleen ole asennettu sovellusta lainkaan, vaikka se siellä selvästi on. Lisään muutamaan kohtaan debug-tehtävän, joka ajaa ls-komennon sovelluksen ankkuritiedostolle, jotta näen, missä kohtaa playbookin suoritusta se on olemassa, ja missä ei.

Tässä välissä käymme arkkitehdin kanssa läpi dev-, test-, qa- ja tuotantoympäristöissä olevat Mule-sovellukset ja sovimme järjestyksen, jossa niitä aletaan päivittää. Jatkan playbookin ongelman selvittämistä, ja alkaa vaikuttaa siltä, että sovelluksen ankkuri on olemassa vain hetken playbookin suorittamisen aloittamisen jälkeen. Käyn läpi playbookin tehtävät, eikä siellä ole mitään, mikä koskisi ankkuriin. Tutkin myös Muleen lokia, ja kiinnitän huomioni siihen, että tietyssä kohtaa sovellus ajetaan alas ja käynnistetään uudelleen. Löydänkin playbookista kohdan, jossa Mule käynnistetään uudelleen ja alan hahmottaa, mitä oikein tapahtuu: ilmeisesti Muleen uudelleenkäynnistys poistaa sovellusten ankkuritiedostot, ennenkuin uudelleenkäynnistys on edennyt tarpeeksi pitkälle, jolloin ne luodaan uudestaan. Tästä prosessista tulee race condition playbookin kanssa, joka haastelee ankkuritiedostoja Mulesta. Playbook etsii ankkuritiedostoja, ja mikäli niitä löytyy, ottaa se varmuuskopion sovelluksesta mahdollista myöhempää palauttamista varten. Nyt ilmeisesti käy niin, että Muleen käynnistyminen on vielä sellaisessa vaiheessa, ettei ankkuritiedostoja ole, ja näin varmuuskopioitakaan ei oteta. Suunnittelen ark-

kitehdin kanssa korjauksen, jossa ennen Mulen uudelleenkäynnistystä otetaan talteen käynnissä olleet sovellukset ja uudelleenkäynnistytksen jälkeen odotetaan, että ne kaikki ovat käynnistyneet.

## Keskiviikko

Testaan Mulen sovellusten käynnistysten odottamista ja heti huomaa, ettei se ainakaan korjaa ongelmaa. Tarkempi analyysi paljastaa, että käynnissä olevien sovellusten haku, ja niiden uudelleenkäynnistymisen odottaminen tapahtuvat playbookin lopuksi, ja sitten vasta Ansible laukaisee Mulen uudelleenkäynnistystapahtuman. Siirrän siis tehtävät Ansiblen handleriin, ja yritän uudelleen. Nyt kyseisiä tehtäviä ei suoriteta ollenkaan. Tutkin hieman Ansiblen handlereita, ja dokumentaation mukaan ne tosiaankin ovat yksittäisiä tehtäviä, ei tehtävien sarjoja (Ansible project contributors. Handlers: Running operations on change). Onneksi saatavilla on kuitenkin listen-toiminnallisuus, jolla saan samasta signaalista laukeamaan useamman tehtävän. Ajan playbookin ja nyt sovelluksen ankkuri on aina paikalla, kun sen olettaakin olevan. Vaan playbook ei siltikään mene läpi.

Tässä välissä kollega kyselee Alfamen Gitlabin järjestelmänvalvojan perään, millaista lähinnä minä kai olen. Hänen tunnuksensa on estetty. Kirjaudun itse Google-tunnuksillani Gitlabiin ja saan itsekin ilmoituksen, että tunnukseni on estetty. Kirjaudun paikallisella järjestelmänvalvojan tunnuksella eston poistamiseksi, mutta tunnuksemme on tilassa "ldap blocked", jonka muuttaminen ei käyttöliittymältä onnistu. Muistan törmänneeni vastaavaan aikaisemminkin, ja varmistan nopealla Google-haulla, että tunnuksien estot poistetaan muuttamalla niiden tila suoraan tietokantaan. Muutan tilan sekä itseltäni, että muutamalta kollegalta, joiden tunnuksukset näyttävät estotilassa olevan.

Pian sama kollega kuitenkin kertoo, että hänen tunnuksensa on jälleen estotilassa. Mitäs ihmettä? Minulla sattuu olemaan selaimessa vielä auki välilehti, jossa Gitlab on auki ja kun päivitän sen, saan virheilmoituksen "Access denied for your LDAP account." Tutkin hiukan Gitlabin lokeja ja sielläkin muutamassa kohtaa mainitaan LDAP. Tämä on hiukan mielenkiintoista, sillä en kirjaudu käyttämällä LDAP:ia vaan OAuth2:sta. Gitlabiimme on kyllä konfiguroituna muutaman vuoden takaa LDAP-autentikointikin, mutta vaihdoimme OAuth2:een, sillä LDAP ei toiminut luotettavasti. Etsin tuon konfiguraation, poistan sen käytöstä, poistan käyttäjien eston ja yritän uudelleen. Nyt pääsemme kollegan kanssa sisään, eikä estoa ole ilmestynyt uudelleen ainakaan vielä päivän loppuun mennessä.

Jatkan Ansible playbookin parissa ja tutkimalla sen tulostetta huomaa, että jaellessani rikkinäistä sovellusta, toimivasta versiosta otetaan varmuuskopio, kuten pitääkin, sovelluksen toimimattomuus tun-

nistetaan oikein, mutta varmuuskopiota palauttaessa se ei käynnisty. Jakelen käsin varmuuskopioidun paketin Muleen ja se käynnistyy ongelmitta. Tarkistan kaikki polut ja oikeudet, mutta ne ovat oikein. Seuraavaksi koitan toistaa käsin sen, mitä playbook tekisi, ja kopioin varmuuskopioidun zip-paketin Mulen apps-hakemistoon. Mule purkaa sovelluksen ja yrittää käynnistää sovellusta ja se epäonnistuu. Näin tehtyäni hoksaan, että itse purin paketin Mulen apps-hakemistoon, sen kopioimisen sijaan. Koikeilen molempia vaihtoehtoja käsin ja todellakin, saman paketin purkamalla sovellus käynnistyy, kopioimalla ja antamalla Mulen purkaa se itse käynnistys epäonnistuu. Kyselen Slack-kanavalla, onko kukaan muu havainnut vastaavaa käytöstä, ja vastaus on negatiivinen. Muutan kuitenkin playbookin kopioimisen sijasta purkamaan paketin Muleen ja lopultakin playbook vaikuttaa toimivan, kuten haluttu. Viimeiset testit jätän vielä huomiseksi.

Torstai

Vien aamusta auton huoltoon ja huoltamolta on lyhyempi matka toimistolle kuin kotiin, joten käyn pitkää aikaa tekemässä lähipäivän. Testailen vielä, että eilen viimeistelemäni playbook oikeasti toimii useissa tilanteissa ja siivoan koodista pois erinäköiset debuggiprintit, joita sinne on matkan varrella kertynyt. Pukkaan koodin versiohallintaan ja valmistaudun siirtymään seuraavaan tehtävään. Sähköpostilaatikkooni kolahtaa kauan odotettu posti: erään ylläpitoasiakkaan VPN-tunnukset. Lisäksi, kuin tilauksesta, lähes samaan aikaan samalta asiakkaalta kolahtaa laatikkoon myös ylläpitopyyntö. Asennankin VPN:n ja testaan, että se toimii ja että pääsen erinäisiin ympäristöihin, joihin minun pitää päästä. En ole aikanaan ollut itse tekemässä tämän asiakkaan järjestelmiä, ja VPN-tunnusten puuttumisen takia en ole niihin tutustunut aiemmillaakaan ylläpitovuoroilla, joten kyselen projektin Slack-kanavalla etenemisohjeita.

Toimistollamme on InfoTV. RaspberryPi pyörittämässä Google Slides -esitystä TV:llä yleisessä tilassa. Jokin aika sitten Google muutti Sheetsissä jotain niin, että esitys ei enää pyöri kokonäytössä. Asiasta on ollut aiemmin puhetta niin vastausta odotellessa, kun nyt kerran olen toimistolla, päätän hoitaa tämän kuntoon. Toteutus toimii ehkä vähän naiivistikin, simuloiden hiiren liikettä ja koko näyttö -nappin painamista. Nyt tuo nappi on kuitenkin siirtynyt piiloon valikon taakse. Haen RaspberryPin työpöydälleni ja kytken siihen omaan näyttöni, näppäimistöni ja hiireni. Havaitseen, että nappi on tosiaan siirtynyt, mutta samalla napille on lisätty pikanäppäin. Muokkaan käynnistyskriptiä niin, että hiiren klikkaamisen simuloinnin sijaan se simuloi pikanäppäimen painamista näppäimistöltä. Ratkaisu vaikuttaisi toimivan. Projektin työläin osuus on MicroSD-kortin irrottaminen RaspberryPista. Lisäksi tarvitsen MicroSD:stä normaali-SD:ksi -adapterin, jollaisen onneksi löydän toimiston varastosta nopeasti.

Lataan muutokset kortilta omalle koneelleni, siirrän ne versiohallintaan muiden toimistojen saataville ja vien RaspberryPin takaisin omalle paikalleen.

Seuraavaksi pitäisi asentaa ennalta sovittu sovellus dev-ympäristöön päivittäen siellä olevan Mulen samalla. Ehdin edistää tätä sen verran, että hankin päivitettyt Mulen versiot, sekä 3.9-, että 3.8-sarjoista, ja laitan ne saataville Artifactoryyn.

Perjantai

Katson sovelluksen, jolla minun pitäisi testata uutta roolia, playbookia, ja sehän ei käytä kyseistä roolia ollenkaan. Joudun aloittamaan siis porttaamalla playbookin käyttämään tätä uudempaa roolia. Kyselen myös Slack-kanavalla uusiksi eilisestä ylläpitotiketistä. Tällä kertaa saan jopa muutaman vastauksen ja käyn palvelimella ihmettelemässä lokitiedostoja. Samalla huomaan, että ylläpidon wikissä oleva komento on väärin. Palvelimella olevat lokitiedostot ovat pakattuja, joten niistä etsimiseen pitää käyttää zgrep-komentoa grep-komennon sijaan. Mitään selvyyttä asiaan en kuitenkaan saa, ja alan jälleen työstää sovelluksen playbookia.

Iltapäivästä ylläpitolaatikkoon kolahtaa vielä pyyntö, jossa asiakkaan asiakkaalle halutaan toimittaa testiympäristön HTTPS-sertifikaatti. Käyn noutamassa sen palvelimelta ja lähetän asiakkaan edustajalle. Onneksi hoksaan lukea tiedoston läpi ennen lähettämistä: siinä kun on tallennettuna sertifikaatin lisäksi epäsymmetrisen salauksen salainen avain, jonka siivoan pois ennen lähettämistä.

Tänään tarpeellinen muistutus oli ainakin käydä kaikki kryptoon liittyvät tiedostot läpi ennen julkituomista.

Viikkoyhteenveto

Ylläpitoviikot ovat usein erittäin sekavia ja pirstaleisia ja saavat koodaripolon tuntemaan kuin häntä revittäisiin noin kuuteen suuntaan yhtä aikaa. Ylläpitotoimien suhteen kulunut viikko oli suorastaan poikkeuksellisen rauhallinen. Ylläpitoviikolla varsinaiselle laskutettavalle kehitystyölle ei ole asetettu tavoitteita, mikä mahdollisti minulle toimiston InfoTV:n korjaamisen. Gitlab-ongelman olisin todennäköisesti joutunut selvittämään itse joka tapauksessa, mutta mukavaa, että sekin sattui tälle viikolle, jolloin se häiritsi normaalia laskutettavaa kehitystyötä mahdollisimman vähän.

Maanantain automaattivalvonnan levytilailmoitus on tyypillisin ylläpitoviikon tehtävä. Tällä viikolla uutuutena oli uusi ylläpitoasiakas, joita tulee ehkä kolme vuodessa. Kyseisen asiakkaan ympäristö on erittäin korkean tietoturvatasonsa takia varsin epätyypillinen, ja siksi tulee viemään useita ylläpitoviikkoja ennen kuin saan itsenäisesti ratkaistua kyseisen asiakkaan tikettejä.

Pääprojektini puolella Ansiblen handlerit olivat jälleen uusi ominaisuus, johon tuli tutustuttua. Dokumentaatio oli onneksi kerrankin hyvää ja haluamani asian sain tehtyä nopeasti sen jälkeen, kun vain keksin, mikä tuo haluamani asia oli.

### 3.10 Viikko 10

Maanantai

Aamusta päivitän seuraavan ylläpitäjän ylläpidon tikettitilanteeseen. Viikkohan oli omalta osaltani varsin rauhallinen. Jatkan testaamalla perjantaina uudelle roolille porttaamaani Notificationcache-sovelluksen playbookia, ja havaitsen, ettei se toimi kuten pitäisi. Jäljitän virheen siihen, että sovelluksen versiota ei tunnisteta oikein, koska sen versionumerossa on kirjaimia. Samanlainen virrehän esiintyi aikaisemmin toisessa kohdassa prosessia. Tällä kertaa ongelma on fataali, niin saan hyvän tekosyyntä ottaa tiketin työn alle ja korjata virheellisen säännöllisen lausekkeen.

Notificationcache on sovellus, joka vastaanottaa muilta sovelluksilta (virhe)viestejä, lisää ne jonoon, ja lähettää ne sitten eteenpäin erissä. Jotta oikeasti voin testata sitä, pitää saada generoitua virheviestejä jostain toisesta sovelluksesta. Kokeilen muutamaa sovellusta, jotka paikallisessa ympäristössäni on asennettuna, mutta virheviestien generointi osoittautuu odotettua hankalammaksi. Onnistun rikkomaan sovellukset kyllä niin, että ne eivät käynnisty lainkaan, mutta en niin, että ne käynnistyisivät, mutta sitten hajoaisivat johonkin.

Tiistai

Kyselen arkkitehdiltä vinkkiä sovelluksen testaamiseen ja katsommekin yhdessä sopivan sovelluksen, ja kuinka se rikotaan sopivasti. Samalla havaitsemme, että notifiaktion saajalistalla on hirveä määrä ihmisiä, joille siitä ei ole mitään hyötyä, etenkin testiympäristöistä. Etsin, mistä sen saisi muuttaa, mutta en löydä. Kysyn jälleen arkkitehdiltä, joka osaa kertoa, mihin asetus on piilotettu, ja lisäksi että

se on AD-ryhmä, eikä lista sähköpostiosotteita, kuten voisi olettaa. Vaihdan asetuksen ja konfiguroin paikallisen testipalvelimeni niin, että sieltä voi lähettää sähköpostia. Käynnistän prosessin, jonka tiedän päättyvän virheeseen ja odotan hetken. Sähköpostiohjelmani ilmoittaa uudesta viestistä ja tiedän, että sovellus toimii ainakin paikallisesti.

## Keskiviikko

Aamusta asennan Notificationcache-sovelluksen dev-ympäristöön ja toistan eiliset toimenpiteeni siellä. Saan jälleen sähköpostinotifikaation merkiksi onnistuneesta toiminnasta. Notificationcache on yksi niistä sovelluksista, joista on kopio sekä sisä-, että ulkoverkon puolella. Testaamani instanssi oli se sisäverkon puolella oleva. Asennan sovelluksen playbookilla myös ulkoverkon dev-palvelimelle, mutta testaus keskeytyy, kun testaamiseen käyttämäni virheen generoivaa sovellusta ei siellä ole. Katson, mitä sovelluksia palvelimella on, ja löydänkin hyvän kandidaatin. En kuitenkaan uskalla suoraan rikkoa sen asetuksia, sillä uskon, että sekin lähettäisi sähköpostia yhdelle, jos toiselle turhalle ihmiselle. Lataan sovelluksen koodit ja alan etsiä, missä tämän sovelluksen notifikaatioiden vastaanottajien lista sijaitsee.

Tämä jää kuitenkin kesken, sillä iltapäivästä lopetan päivän lyhyeen ja siirryn sohvan puolelle suureen videokonferenssiin. Tänään on Great Place to Work -tulosten julkistamisgaala ja olemme organisoineet yhteisen etätapahtuman gaalastriimin ajaksi. Alfame sijoittuu kolmanneksi keskisuurten yritysten sarjassa.

## Torstai

Jatkan sovelluksen koodien tutkimista ja teen havainnon, että sovellus ei vaikuta käyttävän Notificationcachea lainkaan. Minulle herää epäily ja tarkistan muutkin palvelimella olevat sovellukset. Nekkään eivät käytä Notificationcachea. Kerron havainnostani ja arkkitehti kehottaa tarkistamaan tuotannossa pyörivät sovellukset. Tarkistan nekin, eikä niistäkään mikään käytä Notificationcachea. Teemme päätöksen, että ulkoverkon puolen palvelimille ei tätä päivitystä ajeta.

Saan kiireellisen tiketin, jossa pyydetään tarkistamaan, kutsutaanko erästä palvelua HTTP:llä HTTPS:n sijasta. Palvelu on siirtymässä toiselle palvelimelle ja verkkoylläpito haluaa tietää, tarvitseeko palomuurista vielä pitää auki porttia 80. Käyn palvelimella, jolla palvelu sijaitsee, tarkistamassa Nginx:n

lokit, ja kuten arvelenkin, lokiin tulostuva rivi ei eroa kutsumalla palvelua selaimesta HTTP:llä tai HTTPS:llä. Olen jo raportoinut tulosta, kun jokin alkaa vaivaamaan minua. Palvelin, jolla olen lokia lukemassa sijaitsee eri osoitteessa kuin osoite, joka kutsumassani URL:ssä on. Löydänkin tunnukset URL:ssä olevalla nimellä, kirjaudun sisään ja kokonaan uusi palvelin tervehtii minua. Näköjään tällä palvelulla on vielä erillinen edustapalvelin, josta en ollut aiemmin tietoinen. Tällä palvelimella näyttäisi pyörivän Apache HTTPD, ja sen lokeja tarkastellessa löytyy erillinen access\_log ja secure\_access\_log. Apacheen on konfiguroitu uudelleenohjaus HTTP:stä HTTPS:ään, ja yrittäessäni selaimella en saa access\_login puolelle ilmestymään mitään, mutta kun lokitiedostoa hetken aikaa tarkastelen, sinne säännöllisin väliajoin ilmestyy rivejä. En tiedä miten, mutta jokin käyttää palvelua suojaamattomana. Lokia tarkastelemalla käy selväksi, että kaikki pyynnöt tulevat jommastakummasta kahdesta IP-osoitteesta. Raportoin havaintoni ja osoitteet eteenpäin.

Yksi kokeneen työntekijän tärkeimpiä työkaluja on vaisto. Jos jokin alkaa epäilyttää pitää aina muistaa pysähtyä ja tarkastella tilannetta. Jos jokin haiskahtaa mädältä, se mitä todennäköisimmin on mätä. Jättämällä sen takaraivossa kutkuttavan epäilyksen huomiotta olisin tänään raportoinut vääriä tietoja eteenpäin.

Perjantai

Aloitin ajamalla Notificationcachen päivityksen sisäverkon test-ympäristöön. Kaikki menee hyvin, muokkaan koekaniinisovellukseni konfiguraatiota niin, että se generoi virheilmoituksia ja lähettää ne oikealle AD-ryhmälle ja käynnistän prosessin, joka laukaisee virheen. Odotan hetken, että välimuisti ehtii käsitellä ilmoitukset ja lähettää ne eteenpäin... ja mitään ei tapahdu.

Lokia lukemalla näyttää kuin Notificationcache olisi lähettänyt viestit onnistuneesti eteenpäin. Tuplata tarkistan SMTP-palvelimen asetukset ja yritän uudelleen. Ei edelleenkään mitään.

Dailyssä saan tietää, että samassa AD-ryhmässä oleva kollegani on saanut viestit. Vielä oudompaa. Konfiguroin Notificationcachen käyttämäksi SMTP-palvelimeksi localhostin, sammutan siellä pyörivän Postfix-sähköpostipalvelinsovelluksen ja asetan Netcat-työkalun kuuntelemaan SMTP-porttia. Näin minun pitäisi saada ruudulle kaikki Notificationcachen sähköpostipalvelimelle lähettämä tavara. Toistan tilanteen, joka liipaisee postin lähetyksen, mutta Netcat-terminaalini pysyy tyhjänä. Sovelluksen lokeista näen, että tällä kertaa sähköpostin lähetys on epäonnistunut, koska sähköpostipalvelimeen

ei saatu yhteyttä. Lokista selviää myös, että yritetty sähköpostipalvelin on localhost:25, joka on täsmälleen se, mitä Netcatini kuuntelee. Mysteeri toisensa päälle kasaantuu.

## Viikkoyhteenveto

Kulunut viikko oli jotenkin hyvin raskas ja hankala. Tuntui kuin mikään ei oikein olisi edennyt ja muutenkin jotenkin suurin osa viikon työtehtävistä oli ihmettelyä tai johti ihmettelyyn. Huonoja viikkoja on aina silloin tällöin ja niistä pitäisi osata ottaa oppia, mutta en kyllä tiedä, opinko kuluneella viikolla muuta kuin että kaikki voi mennä pieleen, minkä valitettavasti kyllä tiesin jo entuudestaan.

Onneksi keskiviikon Great Place to Work -gaala rikkoi vähän epäonnistumisien ketjua onnistumisen lenkillä. On hienoa olla työyhteisön, joka hyväksyy kehojakin viikkoja ja joka kansallisen tason vertailussakin pärjää näin hyvin, jäsen.

## 3.11 Viikko 11

### Maanantai

Kirjoitan projektin Slack-kanavalle ongelmani auki ja vastausta odotellessani koetan muutamia ideoita, jotka päähäni putkahtavat, kuten bindata IP-osoitteeseen 0.0.0.0 localhost:n sijasta ja varmistaa, ettei kyse ole IPv6/IPv4-ongelmasta. Kysymykseeni ei tule vastauksia, eikä Google-hakukaan tuota oleellisia tuloksia. En edelleenkään tiedä miksi Mule ei osaa Netcatiini yhdistää, SMTP-palvelimen simuloiminen tuolla tavalla pitäisi olla mahdollista, mutta keksin vaihtaa lähestymiskulmaa: konfiguroin takaisin käyttöön oikean SMTP-palvelimen ja asetan palvelimelle tcpdump-ohjelman kuuntelemaan porttiin 25 kulkevaa liikennettä. Toistan virheen ja saan kopion Mulen kommunikaatiosta SMTP-palvelimen kanssa.

Analysoin tuota dataa, ja Mule lähettää minulle postin ihan samoin kuin kaikille muillekin, en vaan jostain syystä saa sitä. Näyttöä tuijotellessani jossain sisälläni välähtää ikävä ajatus. Avaan sähköpostini roskapostilaatikon: puuttuva viesti komeilee listan ylimmäisenä. Kämmeneni läpsäyttää punaisen jäljen otsaani. Käytössämme oleva Googlen sähköpostipalvelu on käytössä olonsa aikana tehnyt niin hyvää työtä, etten ole vielä aikaisemmin roskapostilaatikosta joutunut viestejä kaivelemaan. Lisäksi samannäköiset dev-ympäristöstä peräisin olevat viestit tulivat ongelmitta perille. Tätä eroa käytöksessä en osaa selittää.



Oikeassa elämässä on harvoin vain yksi tapa päästä johonkin lopputulokseen. Tänään sain muistutuksen, että kun yksi keino ei toimi, kannattaa ajatella tarkastella laatikkoa ulkopuolelta ja keksiä vaihtoehtoinen tapa samaan tulokseen pääsemiseksi.

Tiistai

Nyt kun tiedän, että Notificationcache toimii test-ympäristössäkin kuten pitää, valmistaudun viemään sen qa-ympäristöön ja testaamaan sielläkin. QA-ympäristö eroaa muista siinä, että se on ilmeisesti varsin uusi lisäys perheeseen, sillä siellä ei tällä hetkellä pyöri yhtään Mulea tai sovellusta. Arkkitehti tietää, että tarvittavat tietokannat on toimitettu, joten työnä pitäisi olla vain ajaa Notificationcachen, sekä testinä käyttämäni emaksut-sovelluksen playbookit QA-ympäristöä vasten. Testaan vielä paikallisesti emaksut-sovelluksen playbookin kun huomaan projektin Slack-kanavalta projektipäällikön ihmettelyn emaksut-sovelluksesta tulleesta virheilmoituksesta. Hänen mukaansa sovellus, kun on korvautunut pois käytöstä jo 2019. Kerron käyttäneeni sovellusta Notificationcachen testaamiseen, ja että sovellus kyllä löytyy edelleen tuotantopalvelimelta

En halua käyttää testaamiseen vanhentunutta sovellusta, enkä asentaakaan sitä QA-ympäristöön, joten selvitan, mikä sen korvaava sovellus on, ja avaan sen playbookin. Tottahan toki sekin käyttää vanhaa roolia, joten aloitan porttaamaan sitä uudelle roolille. Iltapäivästä kollega kysyy Slack-viestillä josko voisin katsoa kuormantasaajan ongelmaa hänen kanssaan yhdessä. Ongelma ratkeaa sillä, että toisen sovelluspalvelimen nginx:ää komennetaan lataamaan asetuksensa uudelleen. Provisiointi on konfiguroinut nginx:n oikein, mutta ilmeisesti jostain syystä palvelun reload on jäänyt tekemättä. Teen päänsisäisen muistiinpanon tarkkailla asiaa jatkossa.

Keskiviikko

Testaan uudelle roolille portattua playbookia ja se näyttää menevän läpi. Olen kuitenkin hyvin hämilläni, sillä kun katson itse palvelimelta, sovellus ei ole käynnissä, eikä sovelluspalvelimen hakemistossa ole sitä ankkuritiedostoa, jonka playbook väittää sieltä löytävänsä. Tutkin hetken mikä voisi poistaa ankkurin sen jälkeen, kun playbook on sen löytänyt, mutta sellaista ei vaikuta olevan. Lisäksi Mulen lokitiedostoa katsellessa näyttää, ettei siihen ole yritetty asentaa sovellusta laisinkaan hetkeen aikaan. Tutkin seuraavaksi Ansiblen tulostetta, ja jokin siinä ei vaikuta täsmäävän. Sovellus on kahdennettu, joten playbookin pitäisi asentaa se ensin mule1-palvelimelle, ja sen jälkeen mule2-palvelimelle. Ansib-

len tuloste on pitkä, mutta näyttäisi, kuin sovellus asennettaisiin vain kerran. Olen tähän mennessä tarkastellut vain mule1-konetta, pitäisihän sen olla identtinen mule2:n kanssa, mutta parempien ideoiden puutteessa käyn kurkkaamassa myös mule2:ta. Siellä sovellus rullaa onnellisesti ja ankkurikin on paikallaan. Ansiblen tulosteesta näkyy kuitenkin, että sovellusta oltaisiin asentamassa mule1:lle. Mitäs ihmettä? Käy ilmi, että Ansible-inventoryssäni on mule1- ja mule2-koneille molemmille vahingossa annettu alias muleservers\_mule10a, jonka seurauksena Ansible ajoi playbookin vain mule2-konetta vasten, mutta käyttäen aliasta muleservers\_mule10a, joka vastaa mule1-konetta.

Torstai

Sovelluksen playbook vaikuttaa toimivan paikallisesti hyvin, mutta olen kehittänyt sitä omassa haarassaan. Projektin politiikka sen suhteen, mistä haarasta saa provisioida millekin palvelimelle ei ole minulle täysin selkeä, mutta ellei nyt jopa master-haaraan, haluan yhdistää muutokseni ainakin develop-haaraan, sillä tiedän, että kollegani on tehnyt sinne jotain muutoksia sen jälkeen kuin olen oman haaran siitä erottanut. Tämä on lisäksi kaksivaiheinen prosessi, sillä roolit ovat omassa repositoriossaan, joka on liitetty alimoduuliksi itse provisiointirepositorioon. Aloitan yhdistämällä roolirepositorion master-haaran omaan haaraani ja testaamalla. Kaikki sujuu hyvin, ja teen seuraavaksi saman itse provisioinnin repositorion develop-haaralle ja omalle haaralleni. Parin pienen konfliktin jälkeen testaan ja jälleen kaikki sujuu hyvin. Yhdistän nyt haarat toisinpäin ja työnnän ne jaettuun git-repositorioon.

Projektipäällikkö huutelee Slack-kanavalle tiketistä, jonka hän juuri loi asiakkaan yhteistyökumppanin vikailmoituksen perusteella. Arkkitehti bongaa sen ja pingaa minua keskusteluketjussa. Luen tiketin läpi ja hoksaan heti, mitä on tapahtunut. Ongelman kuvaus on, että yhteistyökumppani ei enää pysty yhdistämään erääseen testiympäristössämme olevaan palveluun, vaan hän saa virheilmoituksen "unable to find valid certification path to request target". Kyseessä on testipalvelin, jolle olen ajanut päivityksiä kuluneiden viikkojen ajan. Kyseiseen sovellukseen en ole koskenut, mutta muistan, että aiemmin tällä viikolla palvelimella oli huoltokatko, jonka yhteydessä se käynnistettiin uudelleen. Tämän seurauksena sovellus käyttää nyt päivitettyä versiota JDK:sta. Lisäkäänteenä olemme Oraclen lisenssi-politiikan takia vaihtaneet Oraclen JDK:n heidän toimittamistaan lähdekoodeista käännettyyn OpenJDK:n. Jokin on selkeästi eri tavalla Oraclen JDK:ssa ja OpenJDK:ssa, mitä SSL-sertifikaatteihin tulee.

Kirjoitan analyysini Slack-ketjuun ja käyn palvelimella kopioimassa Mullen lokitiedostosta virheen koulutetun arvaukseni tueksi. Arkkitehti on kanssani samoilla linjoilla ja alkaa kaivelemaan sertifikaat-

teja, jotka JDK:n pitää asentaa sovelluksen toimintaan saattamiseksi. Hän löytää yhden, mutta todennäköisesti sertifikaatteja tarvittaisiin useita. Vilkaisen sovelluksen playbookia ja huomaan, että playbookin suorittamisen yhteydessä kaikki tarvittavat sertifikaatit asennetaan JDK:n. Sovimme arkkitehdin kanssa, että ajan välittömästi sovelluksen provisioinnin uusiksi dev- ja test-ympäristöihin sekä, antaakseni hiukan varoaikaa, huomenna tuotantoon.

Kun saan playbookit ajettua ja sertifikaatit paikalleen kollegani kyselee, jos minulla olisi aikaa katsoa hänen kanssaan QA-ympäristön käyttöönottoon liittyviä asioita hänen kanssaan. Hänen pitäisi provisoida sinne oma sovelluksensa, joka vaatii prosessimoottorin tietokannan olemassaoloa. Prosessimoottorin käyttöönotto QA-ympäristöön on minullakin edessä pian, mutta sisäverkon puolelle. Kollegani sovellus pyörii ulkoverkon puolella. Prosessi pitäisi kuitenkin olla täysin sama, joten opastan, mitä Ansiblen inventory-asetuksia tarvitaan. Suoritamme playbookin ja prosessimoottori sujahtaa ongelmitta paikalleen. Se myös luo tarvitsemansa tietokannan ongelmitta. Provisioimme vielä kollegani sovelluksenkin, joka sekin onnistuu, mutta ilmeisesti kaikkia tarvittavia palomuuurivauksia ei ole tilattu, sillä emme pääse siihen käsiksi.

Perjantai

Aamusta provisioin käyttökatoilmoituksen mukaisesti sertifikaattipäivitykset tuotantoon. Se sujuu ongelmitta. Seuraavaksi ajan testisovellukseni playbookin dev-ympäristöön. Tämä keskeytyy virheilmoitukseen, jossa playbook ei löydä Artifactorystä tarvittavaa pakettia asennettavaksi. Hetken ihmettelyn jälkeen hoksaan, että kyse on jälleen Ansiblen ominaisuudesta, jossa eri group\_vars-hakemistoissa olevat muuttujat niputetaan yhteen vain koska sama kone sattuu olemaan useammassa ryhmässä. Luon hosts-tiedostoon aliakset sovelluksen koneille, sekä dev-, että test-ympäristöihin, ja yritän uudestaan.

Tällä kertaa provisiointi pysähtyy siihen, että eräässä templatessa käytettävää muuttujanimeä ei löydy. Muistankin, että refaktoroin tämän kovakoodatusta käyttämään inventoryssä asetettavaa muuttujaa, ja jostain syystä en silloin lisännyt kyseistä muuttujaa kaikkiin inventoryihin. Nyt dev-ympäristön provisiointi onnistuu. Testaan vielä asettamalla sovellukseen rikkinäiset asetukset ja Notificationcache toimittaa virheviestin minulle onnistuneesti. Ajan seuraavaksi provisioinnin test-ympäristöön, ja teen siellä samat temput. Virheviesti saapuu jälleen onnistuneesti perille. Iltapäivästä ehdin aloittaa vielä qa-ympäristön inventoryn tekemistä. Mitään näistä sovelluksista ei ollut aikaisemmin asennettu siihen ympäristöön, eikä niille ollut inventoryä valmiina. Työajan loppuessa tältä päivältä jätän kuitenkin osan työstä seuraavalle viikolle.

## Viikkoyhteenvedo

Viimeinen seurantaviikko oli aika tavanomainen. Jotkin asiat, kuten kollegan sovelluksen provisiointi puhtaaseen ympäristöön, sujuivat ongelmitta, jotkin toiset olivat hiukan hankalampia. Sähköpostin roskapostilaatikon tarkastamattomuus tietysti hävettää, mutta tyhmiä mokia sattuu jossain välissä kaikille, ja toisin kuin edellisellä viikolla nyt tyhmiä mokia tasapainottamassa oli onnistumisia.

Viikon mielenkiintoisin ongelma oli varmasti tuotannon sertifikaatin puuttuminen, jonka pohjasy syy oli huoltokatkon yhteydessä hallitsemattomasti päivittynyt JDK. Tällaiset tilanteet muistuttavat kokonaisuuden hallinnan tärkeydestä, joka monimutkaisessa ympäristössä valitettavan usein unohtuu.

## 4 POHDINTA JA PÄÄTELMÄT

Olen toki tehnyt jo ohjelmistokehittäjän töitä päälle kymmenen vuotta, mutta koska ohjelmistokehittäjänkin roolin sisällä tehtäviä ja erityisesti erilaisia teknologioita on lukematon määrä, ei yksikään ohjelmistokehittäjä pysty omaksumaan kaikkea. Projektit, joissa olen, vaihtuvat välillä ja tuovat mukanaan uusia työtehtäviä ja teknologioita. Näin ollen uusia asioita opittavaksi tulee väistämättäkin koko ajan vastaan.

Seurantajaksolla työtehtäviini kuului pääasiassa provisiointiskriptien tekoa Ansible-työkalulla. Ansible ei ennestään ollut minulle erityisen tuttu ja sen mahdollisuuksista ja sudenkuopista opinkin seurantajakson aikana paljon. Ansiblen ominaisuudet, ja omituisuudet, ovat monen viikkoyhteenvedon aiheena. Uusina ominaisuuksina seurantajakson aikana tulivat vastaan ainakin filterit ja handlerit, jotka erityisesti olivat syntaksiltaan epäintuitiivisia.

Tapa, jolla Ansible kuvaa muuttujat palvelimiin oli todennäköisesti merkittävin seurantajaksolla oppimani yksittäinen asia. Konfiguraatiolla

```
[tulostuspalveluservers]
mule1
mule2

[postinumeroservers]
mule1
mule2

group_vars/tulostuspalveluservers.yml
group_vars/postinumeroservers.yml
```

intuitiolla jokainen olettaa, että suoritettaessa Ansible playbookia postinumeroservers-palvelinryhmälle muuttujien arvoja haetaan vain ja ainoastaan tiedostosta `group_vars/postinumeroservers.yml`, ei myös tiedostosta `group_vars/tulostuspalveluservers.yml` miten Ansible sen oikeasti tekee. Toiminnan voi kiertää antamalla palvelimille uniikki alias, jonka olen havainnon jälkeen aina tehnyt ja tulen aina tekemään.

```
[tulostuspalveluserservers]  
tulostuspalveluserservers_mule1 ansible_host=mule1  
tulostuspalveluserservers_mule2 ansible_host=mule2
```

```
[postinumeroservers]  
postinumeroservers_mule1 ansible_host=mule1  
postinumeroservers_muel2 ansible_host=mule2
```

En ole tällä hetkellä asemassa, jossa päätän, mitä työkaluja käytetään, mutta omituisuuksistaan huolimatta pidän Ansiblea vähintään tyydyttävänä työkaluna palvelimien provisiointiin. Minulla on kokemusta palvelimien provisioinnista kotikutoisilla bash-skripteillä, joka on sen verran karkea kokemus, että niistä lähes kaikki on jo uudelleenkirjoitettu Ansiblella. Lisäksi Ansible, toisin kuin vaikkapa Puppet, ei tarvitse konfiguroitavaan päähän muuta kuin SSH-palvelimen, joka on mielestäni suuri etu. Olen kyllä alkamassa tulla siihen tulokseen, että tämänkokoinen Ansible-projekti tarvitsisi vähintään selkeät dokumentoidut ohjenuorat, joita kaikki Ansible-skriptejä tekevät, noudattaisivat. Mielelläni refaktoraisin myös muuttujanimiä vähemmän verboosiin suuntaan. Lisäksi, jos pääsisin parantamaan itse Ansible-työkalua, parantaisin sen virheilmoituksia. Olisi suuri etu nähdä suoraan vaikkapa, minkä tiedoston kopiointi mihin kohteeseen epäonnistui sen sijaan, että virheilmoitus on ainoastaan tasoa "tiedoston kopiointi epäonnistui" ja joudun playbookista etsimään, mitä lähde ja kohde olivat.

Ongelmatilanteet ratkaisin pääosin itsekseni Internetin hakukoneiden ja Ansiblen virallisen dokumentaation avulla. Ongelmanratkaisutyylini on aina ollut varsin itsenäinen ja kokonaistehokkuuden kannalta voisikin olla hyödyllistä turvautua kollegoiden apuun hiukan pienemmällä kynnyksellä. Toiseen suuntaan taas olen toki aina pyrkinyt vastaamaan kollegoideni kysymyksiini parhaan osaamisen mukaan, mutta tuoreen kaverin kouluttaminen rooliinsa tuo mielihyvää, mitä ei muista töistä ole vielä saanut. Olin itsekin hiukan yllättynyt siitä, kuinka mukavana pidin mentorin roolia sitä hetken aikaa tehdessäni. On erittäin miellyttävää saada seurata kollegan oppivan ja tarpeen tullen hiukan tökkiä häntä oikeaan suuntaan.

Työssäni joudun laskutusta varten kirjaamaan päivittäin tekemäni työtunnit työaikaseurantajärjestelmään. Nämä kirjaukset menevät sellaisenaan asiakkaalle näkyvälle laskulle. Jokapäiväisten päiväkirjamerkintöjen kirjoittaminen pakotti minut pohtimaan päivän töitä ja tapahtumia erilaiselta kulmalta kuin perinteisesti, mistä seurasi selvä tuntikirjausmerkintöjen laadun paraneminen. Olen joskus saanut palautetta laiskoista tuntikirjauksista, ja pidänkin tätä tämän opinnäytetyöprojektin suurimpana hyötynä.

Pidemmän aikavälin tavoitteenani on edetä arkkitehdin rooliin ja Ansible- ja provisiointiosaaminen on hyvä lisä olemassa olevan devops-osaamiseni ops-päättyyn, ja auttaa varmasti paremmin ymmärtämään jatkossa, miten konfiguraatioiden hallinta ja devops-putket kannattaa suunnitella. Kaiken kaikkiaan olen tyytyväinen nykyiseen osaamiseni tasoon, mutta pyrin toki koko ajan kartuttamaan sitä lisää.

## LÄHTEET

Alfame. GPTW Culture Audit 2022. Saatavissa: [https://docs.google.com/document/d/1wRRzHL7zJVr5ID604\\_aXkG6OP9OY1zIyw9hHyumZckc/edit](https://docs.google.com/document/d/1wRRzHL7zJVr5ID604_aXkG6OP9OY1zIyw9hHyumZckc/edit). Viitattu 16.8.2022

Mulesoft. Database Connector. Saatavissa: <https://docs.mulesoft.com/db-connector/0.3.9/>. Viitattu 24.2.2021

Lomio, Marc Kenneth. Principle of Least Astonishment. Saatavissa: <https://medium.com/@myelmarc/principle-of-least-astonishment-efd5214b44a9>. Viitattu 16.8.2022

Ansible project contributors. Using filters to manipulate data. Saatavissa: [https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_filters.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html). Viitattu 25.2.2021

Ansible project contributors. Tests. Saatavissa: [https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_tests.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_tests.html). Viitattu 5.3.2021

Ansible project contributors. How to build your inventory. Saatavissa: [https://docs.ansible.com/ansible/latest/user\\_guide/intro\\_inventory.html](https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html). Viitattu 9.3.2021

The Apache Software Foundation. Apache Tomcat® - Migration Guide - Tomcat 8.0.x. Saatavissa: <https://tomcat.apache.org/migration-8.html>. Viitattu 15.3.2021

The Apache Software Foundation. Apache Tomcat® - Migration Guide - Tomcat 8.5.x. Saatavissa: <https://tomcat.apache.org/migration-85.html>. Viitattu 15.3.2021

The Apache Software Foundation. 2018. Apache Tomcat 8 Configuration Reference (8.0.53) - Automatic Deployment - Use cases. Saatavissa: <https://tomcat.apache.org/tomcat-8.0-doc/config/automatic-deployment.html>. Viitattu 22.3.2021

Module ngx\_http\_proxy\_module. Saatavissa: [http://nginx.org/en/docs/http/ngx\\_http\\_proxy\\_module.html](http://nginx.org/en/docs/http/ngx_http_proxy_module.html). Viitattu 30.3.2021

Ansible project contributors. Handlers: Running operations on change. Saatavissa: [https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_handlers.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_handlers.html). Viitattu 24.4.20

Praecom. Mikä on paras etäkokous- ja kollaboraatioalusta? Vertailussa: Microsoft Teams, Webex, Zoom, Google Meet ja Slack. Saatavissa: <https://praecom.fi/artikkelit/paras-etakokousalusta>. Viitattu 30.11.2022