

GitOps & API-hallinta

Tila: luonnos

Pvm: 23.6.2022

Tekijä(t): Mika Mähönen



Esityksen rakenne

- Kelan GitOPS –tarina Noin 10 min
- API-Hallintaa ja GitOPS Noin 10 min
- Kysymykset yhdessä Noin 10 min

Lyhyt esittäytyminen

Mika Mähönen

Kela IT-Pääarkkitehti

Vastuualue: Sovellusarkkitehtuuri

Teknologiayksikkö

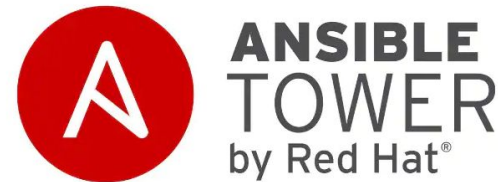
Kehittäjä/Sovellusarkkitehti taustalla

Osaamisia: Java, React, C#, Ansible, OpenShift, OAuth2, JMS, Git, yms.

Kelassa noin reilut 3 vuotta, aikaisempaa taustaa konsulttina

GitOps for newbies

- Menetelmä jossa infrastruktuuriin liittyvät asetukset viedään versiohallintaan. Asetukset synkronoidaan automaation avulla hallittaviin kohteisiin esimerkiksi virtuaalipalvelimiin tai konttiympäristöön.
- GitOPSia voidaan harjoittaa niin perinteisten palvelinten kuin konttien kanssa.
- Muutamia teknologioita aiheeseen liittyen:



Lyhyt historia Kelan GitOpsista

- Ensimmäiset kokeilut automaatiosta Kelassa ponnistettiin 2017 tienoilla. Tuolloin ”mankeloitiin” omia sovelluspalvelimia ja virtuaalikoneita ylläpidon tarpeisiin. Ensimmäisten versioiden osalta automaatio oli ainoastaan käytettävissä keskitettyjen ylläpitotiimien kesken. Varsinaisesta GitOpsista ei voitu vielä puhua.
- 2018 Kelassa käyttöön otettiin ensimmäiset OpenShift –versiot. Ensimmäisten versioiden osalta automaatio oli vielä varsin vähäistä konfiguraation osalta ja eri Kubernetes-objekteja hallinointiin pitkälti OpenShift Consolesta käsin. Tämän koettiin vaativan yleisesti paljon osaamista, koska konttien ylläpidon osuus ei ollut keskitettyä.
- Vuoden 2019 tienoilla siirryimme käyttämään OpenShiftin 4.x versiota, OpenShiftistä alkoi tulla laajemmin käytetty alusta Kelassa. Samalla teimme strategisen päätöksen sovelluskehityksen keskittämiseksi OpenShift-alustalle. Samalla olimme tilanteessa, jossa automaation tasoa oli kasvatettava, koska enemmän ja enemmän tiimejä tuli mukaan kehitykseen.
- 2019 loppuvuodesta kokeiltiin ensimmäisiä HelmChartteja, laajemmin käyttöön ne levisivät vuoden 2020 aikana. Vuoden 2020 fokus oli siirtyä Jenkinsistä Tekton Pipelinejen käyttöön OpenShiftillä.
- Siirryimme käyttämään ArgoCD:tä 2021 vuoden lopussa namespacejen/projektien määrittysten osalta. Vuoden 2022 alusta hallinnoimme niin namespacejen, pipelinejen kuin sovellusten asetuksia ArgoCD:llä.

GitOps – Organisaation muutosajureina

GitOps ja DevSecOps ovat Kelan olleet merkittäviä muutosajureita osana organisaatorakenteiden ketteröittämisestä. Sovellusten asetusmuutoksiin ei tarvita aina Ops –tiimin apua (esim. asennuksien tai päivitysten osalta). Samalla voimme aukottomasti sanoa, mitä muutoksia on tehty ja kenen toimesta, milloin ja kuka ne on hyväksynyt. Samalla säästetään sovelluskehitystiimeiltä merkittävästi aikaa myös muutosten dokumentoinnin osalta. Automatisoimalla ylläpitoon liittyviä toimenpiteitä saadaan aikaan enemmän vähemmällä. Lisäksi ”compliance”-näkökohdat saadaan paremmin haltuun, koska käytetyt parametrit ovat dokumentoitu versiohallintaan ja niiden muutoshistoriaa on helpompi seurata ja raportoida. Vrt. erinäisten Word/Excel –dokumenttien asemasta, joita kertyy vuosien varrelta. ”Kyllähän tämä testissä toimi, mikäs siinä nyt on, kappas tämä parametri on eri”.

Kelan IT:ssä työskentelee noin 800 henkeä. Ylläpidettävien sovellusten määrä on voimakkaassa kasvussa mikropalveluarkkitehtuurin myötä. Hallinnoimme yli 30 etuutta sekä tuotamme palveluita myös Kannalle.

Haasteet

- Muutosvastarinta

"Näin on toimittu aina ennenkin, mikä tämän lisäarvo on? Pystyn ylläpitämään tätä palvelinta ilman automaatiotakin."

- Uuden oppiminen

Ops-käyttäjillä ei välttämättä kokemusta versiohallinnasta ja taas Dev -ihmiset voivat vierastaa esim. sovelluspalvelimiin liittyvien konfiguraatioparametrien haltuunottoa. Automaatiokielet edellyttävät myös opiskelua.

- Useita eri välineitä

Vaikeus valita yhtä ainoaa välinettä. Esim. Kubernetes -alustalla ArgoCD + HelmChartit tarjoavat hyvän tulokulman. Virtualikoneiden kanssa vaihtoehtoja ovat mm. Ansible/Saltstack.

- Valtasuhteet / Tiimien rajat

"Ei me tuota ainakaan oteta, tiedetään itse paremmin." Kulminoituu yleensä kolmen edellisen perusteella. Luovutanko omaa päätösvaltaani pois?

Tulevaisuuden näkymät

- OpenShift on Kelan strateginen valinta sovellusten ajoalustaksi. Uudet palvelut tuotetaan mikropalveluina tai funktiona monoliittien asemasta. Tuolloin ajettavien softakomponenttien määrä kasvaa.
- Sovellusten määrä kasvaa entisestään, mutta koko pienenee, joka mahdollistaa muutostarpeiden viemisen läpi ketterämmin organisaatiossa.
- Sovelluskehityksen kompleksisuus aste kasvaa tulevaisuudessa nopeasti, joka nostaa tarvetta automaatiolle. ”Kaikki eivät voi osata millään kaikkea”, jolloin automatisoimalla konfiguraation hallintaa voidaan poistaa pullonkauloja. Eli ”Ideasta tuotantoon nopeammin”
- Haavoittuvuuksista 3. osapuolen koodeissa raportoidaan yhä tiuhempaa. Lisäksi eri kirjastojen ja kehysten tukiajat lyhenevät entisestään. Tämä tarkoittaa nopeampaa sykliä päivitysten osalta. Tämä korostaa automaation tarvetta entisestään, kehitystiimiä pitää voida palvella entistäkin nopeammin ja yksinkertaisemmin.
- IT:n Prosessit ja vaatimukset kasvattavat vaatimuksia dokumentoinnin ja läpinäkyvyyden suhteen. GitOps tarjoaa työvälineen Kelalle täyttää osan vaatimuksista.

Automatisoinnin saralla

- Kehitämme edelleen käyttäjäkokemusta sovelluskehityksen automaation parissa. Voisiko ympäristön tilata täyttämällä vain lomake?
- Kehitämme nykyistä infran automaatiota entisestään. Pyrimme ulottamaan automaation yhä uusiin käyttökohteisiin. Tästä esimerkkinä myös Mainframe –puolen automatisoinnin jatkokehittäminen. Voidaanko tulevaisuudessa Mainframe –kehityksessä käyttää samoja välineitä Java/NodeJS –sovellusten kanssa?
- Pyrimme siirtymään laajasti kontteihin ja mahdollistamaan tiimien laajemman autonomian, mutta samalla paremman tietoturvan ja nopeammat korjaukset haavoittuvuuksien osalta
- Kehitämme lisää API-hallintaan liittyvää automaatiota GitOps:in avulla.

”Vuodessa on 365 päivää, joista jokaisena pitäisi voida tehdä tarvittaessa käyttöönottoja”

Lyhyt esittäytyminen

Riku Kuusisto

Kela IT-arkkitehti

Vastuualue: Integraatioarkkitehtuuri

Java-kehittäjä/ratkaisuarkkitehti

Kelassa vuoden, aikaisempaa taustaa konsulttina

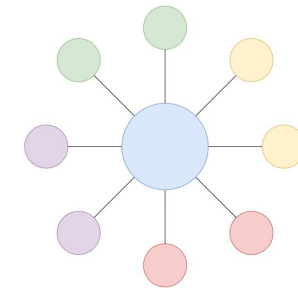
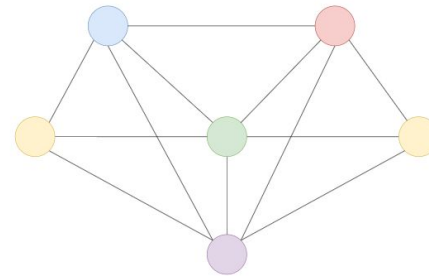
Kelan integraatioarkkitehtuurin historiaa

Yleisesti käytettyjä integraatiomalleja

Point-to-point

Hub-spoke

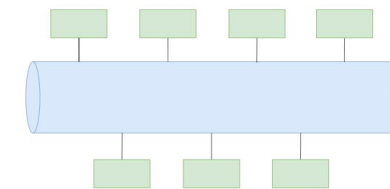
ESB



Haasteita

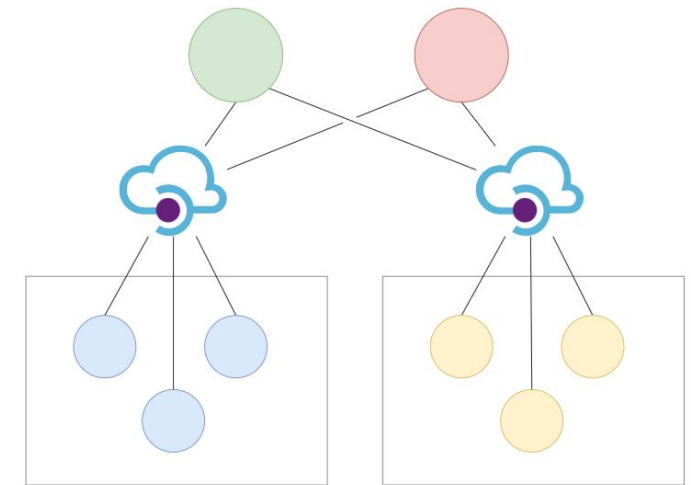
Lukuisien point-to-point integraatioiden hallitseminen kaoottista

Hub-spoke-malli aiheuttaa yksittäisen vikapisteen



API-arkkitehtuuri & Red Hat 3Scale

- Sovellukset tarjotaan REST-rajapintointia
- Hallinta Red Hat 3scale API-hallintajärjestelmällä
- REST-rajapinnat kootaan API-kokonaisuuksiksi
- Sanomaliikenne APIcast yhdyskäytävien kautta
- Vikasietoisuus, kestävyys ja saatavuus



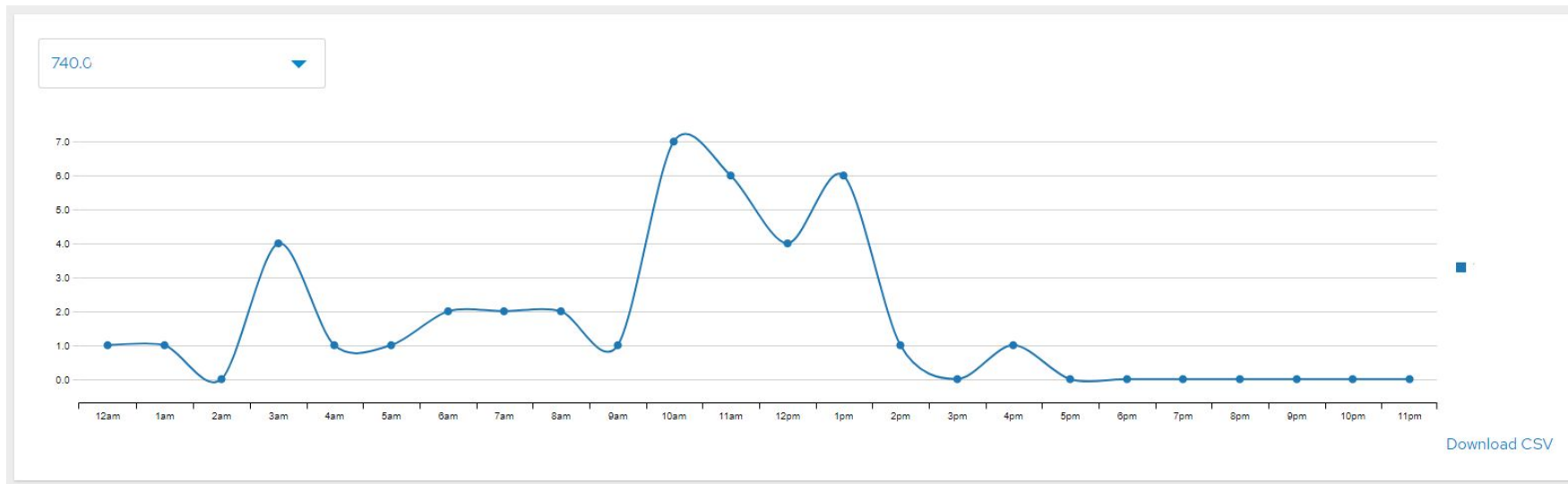
<https://sovellus1.api.kela.fi>



<https://komponentti1.sovellus1.kela.fi>
<https://komponentti2.sovellus1.kela.fi>
<https://komponentti3.sovellus1.kela.fi>

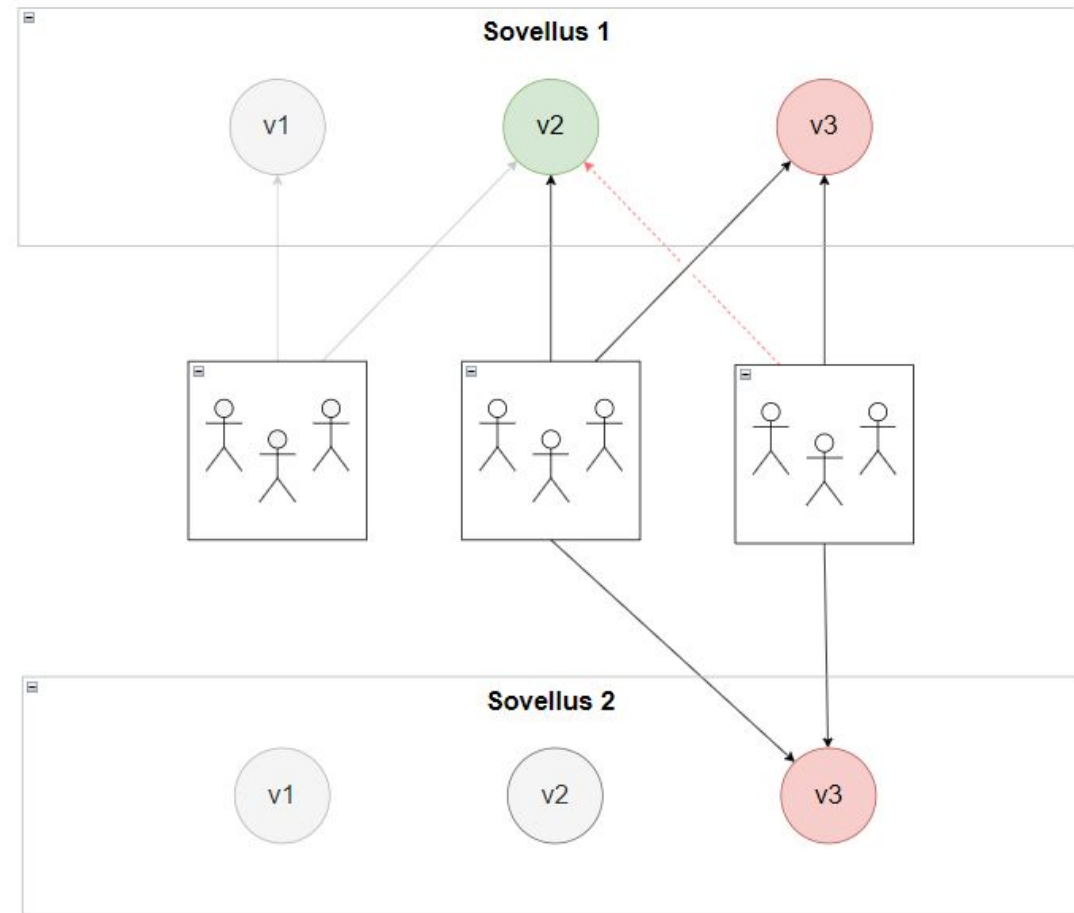
3scale kehittäjän apuna

- Metriikka
- Pääsynhallinta
- Rajapintakuvausten julkaisu



3scale & sovellusten versiointi

- Jokainen sovelluksen versio omana tuotteenaan
- Käyttäjille pääsy tarvittaviin versioihin
- Käyttäjäkunta on hyvin dokumentoitu
- Viestintä sovelluksen muutoksista tehokasta ja kohdennettua



API-hallinta ja automaatio

- 3scalen asennus ja ylläpito Ansible Playbookeilla
- Parametrit versionhallinnassa
- Käyttöönotto nopeaa ja inhimillisen virheen riski pienempi

- API-hallinnan ylläpito myös automaattista
- Sovellusten versiointi skriptattu
- Rajapintakuvausten julkaisu automatisoitu



Haasteita

- API-hallinta on uusi konsepti, vaatii uutta perspektiiviä
 - API-hallinnan käyttöönottoon menevä työ maksaa itsensä nopeasti takaisin
- Sovelluksia API-hallinnan kohteiksi tulee lisää, miten sen hallinta?
 - Sovellustiimit huolehtivat, että API-hallinta on omien sovellusten osalta ajan tasalla
 - 3Scalen tekninen ylläpito on keskitetty, eikä siten rasita sovellustiimejä ja kehittäjiä
 - Ylläpidon automaatiota kehitetään entisestään

Tulevaisuuden näkymiä

- Sovellusten elinkaarisuunnittelu
- API-hallinnan ylläpidon automaatiota parannetaan entisestään
 - Tavoitteena täysi GitOps-automaatio
- API-hallinta mukana kaikessa sovelluskehityksessä
- API-hallinta mukaan myös kumppaniverkoston sovellusintegraatioissa

