

Lauri Saavalainen

# AWS-palveluiden vertailu Ilmatieteen laitoksen tarpeisiin

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinöörityö

20.12.2024

# Tiivistelmä

Tekijä: Lauri Saavalainen  
Otsikko: AWS-palveluiden vertailu Ilmatieteen laitoksen tarpeisiin  
Sivumäärä: 30 sivua  
Aika: 20.12.2024

Tutkinto: Insinööri (AMK)  
Tutkinto-ohjelma: Tieto- ja viestintätekniikka  
Ammatillinen pääaine: Ohjelmistotuotanto  
Ohjaajat: Yliopettaja Janne Salonen

---

Tässä insinööriyössä toteutettiin vertailu Amazonin AWS-pilven eri konttien hallintajärjestelmistä. Työ toteutettiin Ilmatieteen laitokselle, joka halusi kahdentaa omassa konesalissa olevat säännustusjärjestelmänsä AWS-pilveen. Varsinaista vertailua alustettiin yleistiedolla konteista, pilvipalveluista ja AWS:tä. Alkuperäisenä tavoitteena oli myös perustaa konttiympäristö AWS-pilveen hyödyntäen vertailun tuloksena valittua AWS-palvelua. Tämä ei kuitenkaan toteutunut, joten sen sijaan toteutettiin pieni Ilmatieteen laitoksen avointa säädädataa ja Docker-kontteja hyödyntävä ohjelmointiprojekti.

Vertailu keskittyi kolmeen varteenotettavimpaan vaihtoehtoon, jotka olivat AWS:n ECS, EKS ja ROSA. Kutakin vertailtiin Ilmatieteen laitoksen asettamien kriteerien perusteella. Tärkeimmät kriteerit olivat palvelun yhteensopivuus nykyisen järjestelmän kanssa ja se, miten palvelu pystyy skaalautumaan resurssien tarpeen vaihdellessa. Vertailussa todettiin ECS:n ja EKS:n soveltuvan ominaisuuksiltaan ja skaalautuvuudeltaan Ilmatieteen laitoksen tarpeisiin, mutta niiden ei katsottu olevan riittävän yhteensopivia nykyisen järjestelmän kanssa. ROSA-palvelun todettiin olevan paras vaihtoehto Ilmatieteen laitoksen tarpeisiin, koska se ainoana palveluna täytti kaikki asetetut kriteerit. ROSA-palvelun erotti muista se, että se käyttää OpenShift-konttien hallintajärjestelmää, joka on jo käytössä Ilmatieteen laitoksella.

Tähän insinööriyöhön kuuluu myös ohjelmointiprojekti, joka tehtiin täydentämään tässä työssä aiemmin esitettyä teoriaa ja korvaamaan alkuperäisessä suunnitelmassa olleen konttiympäristön perustamisen AWS-pilveen. Ohjelmointiprojekti koostuu paikallisesti kontitetusta ohjelmasta, joka hakee Ilmatieteen laitoksen voimassa olevat varoitukset avoimesta tietokannasta, karsii ja muokkaa haettuja tietoja selkeämpään muotoon ja asettaa ne luettavaksi localhostiin. Tämän lisäksi projektiin toteutettiin ohjelma, joka lukee säävaroitukset localhostista ja asettaa ne luomaansa käyttöliittymään luettavaksi.

Avainsanat: AWS, pilvipalvelu, kontti, ROSA, vertailu, Ilmatieteen laitos

---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

## Abstract

Author:	Lauri Saavalainen
Title:	Comparing AWS Cloud Services for Finnish Meteorological Institute
Number of Pages:	30 pages
Date:	20 December 2024
Degree:	Bachelor of Engineering
Degree Programme:	Information and Communication Technology
Professional Major:	Software Engineering
Supervisors:	Principal Lecturer Janne Salonen

---

In this engineering thesis, a comparison of different container orchestration tools in Amazon's AWS cloud was conducted. The project was carried out for the Finnish Meteorological Institute (FMI) that wanted to duplicate its on-premises weather forecasting systems in the AWS cloud. The comparison was preceded with general information about containers, cloud services, and AWS. The original goal also included setting up a container environment in the AWS cloud using the service selected based on the comparison results. However, this was not realized, and instead, a small programming project utilizing Docker containers and open weather data from the Finnish Meteorological Institute was constructed.

The comparison focused on three viable AWS services: ECS, EKS, and ROSA. Each was evaluated based on criteria set by the FMI. The most important criteria were the compatibility of the service with the existing system and its ability to adapt to fluctuating resource demands. The comparison found that while ECS and EKS were suitable in terms of features and scalability, they were not deemed to be compatible enough with the existing system. ROSA was determined to be the best option for the FMI's needs as it was the only service that met all the established criteria. What set ROSA apart was its use of the OpenShift container management system, which the FMI was already using.

This thesis also includes a programming project that was carried out to complement the theory presented earlier in the thesis and replace the originally planned container environment setup in the AWS cloud. The programming project consists of a locally containerized program that retrieves the active alerts from the FMI's open database, filters and structures the data to more readable format and makes it accessible via localhost. Additionally, a program was implemented to read the weather alerts from localhost and display them in a graphical user interface.

Keywords: AWS, cloud computing, container, ROSA, comparison, Finnish Meteorological Institute

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Pilvipalvelut ja kontit	3
2.1	Pilvipalvelut	3
2.1.1	Mitä ongelmia pilvipalvelut ratkaisevat?	4
2.1.2	Mitä riskejä pilvipalveluihin liittyy?	5
2.2	Kontit	6
2.3	AWS	8
3	AWS konttipalvelun valitseminen	10
3.1	Ilmatieteen laitoksen kriteerit AWS-palvelulle	10
3.2	AWS-konttityökalut	12
3.2.1	ECS	12
3.2.2	EKS	14
3.2.3	ROSA	16
3.3	Muut palvelut/työkalut	18
3.3.1	Batch	18
3.3.2	Lightsail	19
3.3.3	Lambda	20
3.4	Konttipalvelun valinta	20
4	Docker-ohjelmointiprojekti	22
4.1	Projektin kuvaus	22
4.2	Projektin toteutus	23
4.2.1	Käytetyt työkalut	23
4.2.2	Ohjelman toiminta	24
4.3	Projektin itsearviointi	25
5	Yhteenveto	27
	Lähteet	29

## Lyhenteet ja käsitteet

- ECS:** Elastic Container Service. AWS-pilven oma konttien hallintajärjestelmä.
- ECR:** Amazonin pilvipalvelu, joka tarjoaa säiliön konttien luomisen tarvittavien tiedostojen (image) säilömiseen ja hallintaan.
- EC2:** Elastic Compute Cloud. Amazonin virtuaalikoneita tarjoava pilvipalvelu.
- EKS:** Elastic Kubernetes Service. AWS-pilven Kubernetes pohjainen konttien hallintajärjestelmä.
- Image:** Tiedosto, joka sisältää kaiken tarvittavan datan kontin luomiseksi
- Klusteri:** Looginen kokonaisuus, joka sisältää tarvittavat resurssit konttien hallinnoimiseksi.
- Kontti:** Kontti on standardisoitu ympäristö tietokoneohjelmalle, joka sisältää kaikki sen ajamiseen tarvittavat resurssit järjestelmästä riippumatta.
- Localhost:** Localhost viittaa osoitteeseen, jota käytetään kutsuttaessa tietokoneen omaa sisäistä palvelinta.
- OpenShift:** Red Hatin tarjoama konttien hallintajärjestelmä, joka perustuu Kubernetesiin ja laajentaa sen ominaisuuksia.
- ROSA:** Red Hat OpenShift Service on AWS. Amazon ja Red Hat -yriyten yhdessä tarjoama OpenShift-pohjainen konttien hallintajärjestelmä.
- S2I:** Source to Image. Tarkoittaa niitä prosesseja ja työkaluja, joita tarvitaan ohjelmakoodin muuttamiseksi kontin luontitiedostoksi (image).

# 1 Johdanto

Tässä insinööriyössä vertaillaan AWS-pilven eri konttipalveluita ja tarkastellaan niiden soveltuvuutta Ilmatieteen laitoksen käyttämille säännennustusjärjestelmille. Työ toteutetaan osittain Ilmatieteen laitokselle, jonka tämänhetkiset säännennustusjärjestelmät toimivat Ilmatieteen laitoksen omissa konesaleissa. Ilmatieteen laitos on julkinen laitos, joka tutkii säätä, ilmakehää, meriä ja lähiavaruutta. Ilmatieteen laitos tuottaa keräämästään datasta palveluita kansalaisten ja elinkeinoelämän tarpeisiin. Lisäksi Ilmatieteen laitos tekee yhteistyötä muiden maiden vastaavien organisaatioiden ja kansainvälisten yhteisöjen kanssa. (1.)

Ilmatieteen laitos haluaa kahdentaa nykyisen omassa konesalissa olevan laskentaympäristönsä. Pilveen perustettava laskentaympäristö ei korvaa nykyistä järjestelmää, vaan se tukee sitä ja pystyy toimimaan itsenäisesti, jos nykyinen järjestelmä ei syystä tai toisesta toimi. Pilveen perustettava järjestelmä ei ole myöskään yhtä laaja kuin nykyinen järjestelmä. Nykyisestä järjestelmästä vietään pilveen vain ne osat, jotka ovat välttämättömiä ennusteen laatimiseksi. Pilveen vietävä järjestelmä ei siis tuota yhtä tarkkoja tai monipuolisia ennusteita kuin nykyiset järjestelmät.

Tämä insinööriyö jakautuu kolmeen osaan. Ensimmäisen osion on tarkoitus tarjota konteksti muulle työlle. Siinä käydään läpi pilvipalveluita, AWS:ää ja konttien hallintaa yleisellä tasolla. Tarkoitus on käydä läpi sitä, miten ne toimivat, mitä ongelmia ne pyrkivät ratkaisemaan ja mitä haasteita ne tuovat mukanaan. Osio pyrkii selittämään miksi pilvipalvelut ovat niin suosittuja ja samalla auttamaan lukijaa ymmärtämään tämän työn merkitystä. Toisessa osassa suoritetaan varsinainen Ilmatieteen laitokselle tehtävä vertailu eri AWS:n konttien hallintajärjestelmistä ja kolmannessa osassa toteutetaan pieni ohjelmointiprojekti aiheeseen liittyen.

Toisen vaiheen AWS-pilven konttien hallintajärjestelmien vertailu toteutetaan Ilmatieteen laitokselle. Ilmatieteen laitos on asettanut tietyt kriteerit, joiden

pohjalta vertailun voi tehdä. Tässä työssä keskitytään muutamaan varteenotettavimpaan vaihtoehtoon, joista Ilmatieteen laitos valitsee yhden toteutettavaksi tämän vertailun pohjalta. Jokaisen palvelun toimintaperiaate ja ominaisuudet käydään läpi ja jokaisen palvelun soveltuvuutta arvioidaan Ilmatieteen laitoksen asettamien kriteerien pohjalta. Lisäksi tässä vaiheessa käydään läpi lyhyemmin myös muita vain osittain tarkoitukseen sopivia AWS-palveluita.

Työn kolmannessa vaiheessa toteutetaan pieni ohjelmointiprojekti, joka hyödyntää Ilmatieteen laitoksen avointa säädataa ja Docker-sovellusalustaa. Sen on tarkoitus havainnollistaa konttien toimintaa ja näin täydentää tässä työssä esitettyä teoriaa konttien toiminnasta. Alun perin tämän projektin piti olla Ilmatieteen laitokselle toteutettava kokonaisuus, jossa AWS-pilveen perustettaisiin laskentaympäristö valittua AWS-palvelua varten. Tämä ei kuitenkaan toteutunut, joten kolmas osio jouduttiin supistamaan pienemmäksi projektiksi. Tämän supistumisen syihin ja alkuperäisen suunnitelman korvaavan projektiin tutustutaan tarkemmin kolmannessa osiossa.

## 2 Pilvipalvelut ja kontit

### 2.1 Pilvipalvelut

Pilvipalvelu on verkkopalvelu, joka sijaitsee palveluntarjoajan omalla palvelimella. Käyttäjän ei tarvitse asentaa erillistä ohjelmistoa palvelun käyttämiseksi, vaan käyttäjä hyödyntää palvelua internet-yhteyden välityksellä. Pilveen on mahdollista tehdä monenlaisia palveluita, mutta tavallisille käyttäjille tunnetuimpia näistä palveluista lienevät palvelut, johon käyttäjät voivat tallentaa esimerkiksi kuvia, videoita ja tiedostoja. Yksityishenkilöille ja yrityksille on kuitenkin olemassa myös paljon monipuolisempia palveluita. Käyttäjä voi esimerkiksi pyörittää omaa nettisivuaan pilvessä, luoda pilveen tietokannan omaa ohjelmaa varten tai vuokrata laskentatehoa datan prosessointia varten. Käyttäjille on olemassa paljon valmiita palveluita eri tarpeisiin, mutta käyttäjä voi myös itse rakentaa oman palvelunsa virtuaaliseen pilviympäristöön.

Monet eri yritykset tarjoavat erilaisia palveluita. Tavalliselle käyttäjälle Microsoftin Office 365, Applen iCloud ja Dropbox lienevät kaikista tutuimpia palveluita. Office 365 -palvelu mahdollistaa erilaisten tekstinkäsittely- ja taulukkolaskentaohjelmien käytön verkossa ilman asennusta. Käyttäjän tarvitsee vain kirjautua palveluun päästäkseen käsiksi tiedostoihinsa ja luodut tiedostot säilytetään pilvessä, jossa useampi käyttäjä voi muokata niitä samanaikaisesti. Applen iCloud ja Dropbox tarjoavat molemmat varmuuskopioinnin käyttäjän tiedostoille. Lisäksi Applen iCloud mahdollistaa Applen omien mobiililaitteiden varmuuskopioinnin pilvessä. Moni pilvipalvelu on myös täysin näkymätön loppukäyttäjälle. Käyttäjä saattaa käyttää pilvipalveluita tietämättään tehdessään verkko-ostoksia, pelatessaan online-peliä tai katsoessaan elokuvaa suoratoistopalvelusta.

Yrityksille tärkeimmät pilvipalvelut ovat joko IaaS-palveluita tai PaaS-palveluita IaaS-palvelut (Infrastructure as a Service) tarkoittaa palvelua, jossa palvelun tarjoaja ylläpitää virtuaalipalvelimia, joihin käyttäjä asentaa oman ohjelmistonsa. Pakettiin kuuluu yleensä myös tallennustilaa palvelua varten. PaaS-palvelu

(Platform as a Service) on IaaS-palveluita monipuolisempi pilviympäristö, jossa käyttäjälle tarjotaan kehitysalusta ohjelmiston kehitykselle ja ylläpidolle. Käyttäjä voi liittää oman ohjelmansa osaksi joustavaa kehitysympäristöä, jolloin siitä tulee helposti skaalautuva tehotarpeen mukaan. Amazon Web Services (AWS), Google Cloud Platform (GCP) ja Microsoft Azure ovat suurimpia IaaS- ja PaaS - palveluiden tarjoajia. Kolmas pilvipalvelun tyyppi on SaaS (Software as a Service), jossa palveluntarjoaja tarjoaa valmiin pilviohjelmiston käyttäjälle. (2.)

### 2.1.1 Mitä ongelmia pilvipalvelut ratkaisevat?

Pilvipalveluiden tärkein ominaisuus käyttäjän kannalta on palvelimiin liittyvien monimutkaisuusien poistaminen tai yksinkertaistaminen. Käyttäjä voi valikoivasti ja joustavasti siirtää pilveen haluamansa ohjelmistot tai tilata valmiita palveluita korvaamaan ne. Yritykset voivat hyötyä suuresti, kun palvelimien ylläpitoon ei tarvitse käyttää resursseja, vaan ne voi ostaa tai vuorata halvemmalla pilvestä. Pilvipalvelut tarjoavat yritykselle myös enemmän joustavuutta, sillä yritys voi siirtää myös vain osan palveluista pilveen tai kahdentaa olemassa olevat palvelut pilveen. Pilvipalvelut tarjoavat myös pienyrityksille ja yksittäisille ihmisille mahdollisuuden ajaa omia palveluitaan pilvessä. Näillä tahoilla ei välttämättä ole tarvittavia resursseja, osaamista tai aikaa oman palvelimen ylläpitoon, mutta pilvipalveluiden avulla oman nettisivun tai ohjelmiston pyörittäminen onnistuu.

Toinen pilvipalveluiden merkittävä etu on niiden skaalautuvuus. Perinteisesti palvelun kapasiteetin kasvattaminen on vaatinut uusien palvelimien asentamista ja konfigurointia. Pilvipalvelut tarjoavat useita eri vaihtoehtoja palvelun laajentamiseksi. Olemassa olevia virtuaalipalvelimia on helpompi ottaa käyttöön, koska uusia fyysisiä palvelimia ei tarvitse asentaa. Riittää, että uudet virtuaalipalvelimet konfiguroidaan. Jotkin pilvipalvelut kykenevät skaalautumaan automaattisesti käyttäjän tarpeisiin. Tällöin palvelu kasvattaa tai vähentää virtuaalipalvelimien määrää automaattisesti työkuorman määrän mukaan. Tämä

tuo säästöjä pilvipalvelujen käyttäjälle, koska käyttäjän ei tarvitse maksaa kiinteästä palvelimesta, vaan hän vuokraa palvelinta vain silloin kuin sitä tarvitsee.

Kolmas pilvipalveluiden tuoma etu on niiden modulaarisuus. Pilvipalveluun tuodut ohjelmistot on mahdollista liittää osaksi suurempaa kokonaisuutta, jossa on tarjolla pilvipalvelun tarjoajan luomia valmiita palveluita. Pilveen tuotu ohjelmisto voi esimerkiksi hyödyntää pilvipalvelun valmista tietokantaa tai tiedonkäsittelyohjelmaa. Joissain tapauksissa käyttäjän on mahdollista rakentaa oma palvelunsa käyttäen pelkästään pilvipalvelun tarjoajan kehittämiä työkaluja. Pilvipalvelun tarjoajat haluavat mahdollistaa myös mahdollisimman sujuvan palveluiden siirron konesaleista pilveen. Tämän vuoksi pilvipalvelut yleensä tarjoavat mahdollisuuden hyödyntää yleisimpiä konttien hallintajärjestelmiä, kuten Kubernetesiä ja OpenShiftiä myös pilvessä. Tämä mahdollistaa myös osittaisen yhteensopivuuden eri pilvitarjoajien välillä.

### 2.1.2 Mitä riskejä pilvipalveluihin liittyy?

Ohjelmistojen siirtäminen omasta konesalista toisen yrityksen konesaliin luo väistämättä useita ongelmia. Yksi niistä on osittainen hallinnan menettäminen. Monelle yritykselle on tärkeää pystyä hallitsemaan infrastruktuuria, jonka päälle pilvipalvelu on rakennettu. Pilvipalveluita käytettäessä ainakin osa tästä hallinnasta siirtyy toisen yrityksen vastuulle. Pilvipalvelut tarjoavat erilaisia ratkaisuja, joista osa mahdollistaa pääsyn virtuaalipalvelimille verkon välityksellä. Tällöin yritys edelleen itse vastaa palvelimen konfiguroinnista ja asetuksista. Jos yritys on valmis luopumaan virtuaalipalvelimien hallinnasta, voi se ulkoistaa virtuaalipalvelimien hallinnan täysin pilvipalvelun tarjoajalle. Hallinnan menettämiseen liittyy kuitenkin myös muita riskejä, jotka yrityksen täytyy huomioida pilvipalvelua valittaessa.

Tietoturva on yksi tärkeimmistä osa-alueista palvelua suunnitellessa. Pilvipalvelua käytettäessä vastuu tietoturvasta siirtyy kokonaan tai osittain

palveluntarjoajan vastuulle. Useimmat pilvipalvelut väittävät olevansa turvallisia, mutta todellisuudessa käyttäjän on vaikea arvioida pilvipalvelun tietoturvan tasoa. Käyttäjän ohjelman luotettavuuden ja toimivuuden kannalta on kuitenkin tärkeää, että käyttäjän valitsema palvelu on täysin luotettava. Yritykset luottavat yleensä kaikista suurimpiin ja tunnetuimpiin pilvipalveluiden tarjoajiin tietoturvan varmistamiseksi. Suurimmilla palveluntarjoajilla on kaikista eniten resursseja tietoturvan varmistamiseksi ja suurin palveluun luottava asiakaskunta. Jos käyttäjällä on riittävästi resursseja ja tietoturvaosaamista, voi käyttäjä vain vuokrata pilvipalvelimia ja huolehtia itse niiden turvallisuudesta. Tässäkin tapauksessa käyttäjä joutuu kuitenkin turvautumaan osin palveluntarjoajan tietoturvaan palvelujen välisessä tietoliikenteessä.

Jos yritys käyttää laajasti jonkun palveluntarjoajan pilvipalveluita, voi yrityksen järjestelmästä tulla erittäin riippuvainen palveluntarjoajan järjestelmästä. Jos pilvipalveluun tulee häiriö tai jos pilvipalvelun ylläpitäjä ajautuu konkurssiin, voi pilvipalvelussa olevat palvelut yllättäen katketa, jos ne eivät pyöri myös jossain muussa pilvipalvelussa tai konesalissa. Palvelun käyttäjät voivat ehkäistä näitä ongelmia kahdentamalla järjestelmän toiseen pilvipalveluun tai konesaliin, mikä tosin lisää palvelun kustannuksia huomattavasti. Riippuvuus aiheuttaa myös sen, että pilvipalvelun tarjoajan vaihtaminen voi olla vaikeaa, jos palvelussa oleva järjestelmä ei ole yhteensopiva muiden pilvipalveluiden kanssa. Pilvipalveluiden tarjoajat kuitenkin yleensä tarjoavat palveluita, jotka ovat yhteensopivia yleisimpien konttien hallintaympäristöjen, kuten Kubernetesin ja OpenShiftin kanssa. Tällöin palvelun siirtäminen tai kahdentaminen toiseen ympäristöön on helpompaa.

## 2.2 Kontit

Kontti on standardisoitu ympäristö tietokoneohjelmille. Se mahdollistaa niiden ajamisen erilaisissa järjestelmissä ja konttien siirtämisen eri järjestelmien välillä. Kontit sisältävät kaiken tarvittavan ohjelman ajamiseksi. Se sisältää varsinaisen

ohjelman lisäksi kaikki tarvittavat kirjastot, liitännäiset ja konfiguroinnit. Tämä mahdollistaa sen, että ohjelma voidaan ajaa saman tien, kun se on siirretty uuteen ympäristöön. Kontteja on helppo siirtää eri tietokoneiden ja käyttöjärjestelmien välillä, mutta niistä on eniten hyötyä ohjelmiston kehityksessä. Kontteja on helppo siirtää kehittäjän koneelta testi- ja staging-ympäristöön ja lopulta tuotantoon. Kontteja voi ajaa kaikenlaisissa ympäristöissä, kuten konesaleissa ja pilvipalveluissa. Docker on suosituin kontittamiseen käytetty työkalu. Sen avulla voi luoda ja ajaa kontteja. Lisäksi on olemassa paljon erilaisia ohjelmistoja ja palveluja, joiden avulla pystyy hallinnoimaan suuria määriä kontteja. Tällaisia ohjelmistoja ja palveluita ovat muun muassa Kubernetes, Amazonin ECS ja Red Hatin OpenShift.

Mitä hyötyä ohjelmien kontittamisesta sitten on? Edellä mainittiin jo se, miten konttien avulla voi siirtää ohjelmia eri järjestelmien välillä ja miten kontteja voidaan ajaa näissä ympäristöissä ilman, että tarvitsisi tehdä muutoksia ohjelmaan tai kohdejärjestelmään. Tämän mahdollistaa se, että konttiin on asennettu kaikki ohjelman ajoon tarvittavat työkalut. Perinteisesti virtuaalipalvelimia on käytetty konttien tapaan alustana ohjelmien ajamiseksi. Kontit pystyvät jakamaan virtuaalipalvelimen tarjoamat resurssit, kuten käyttöjärjestelmän, mikä mahdollistaa konttien huomattavasti pienemmän koon verrattuna virtuaalipalvelimiin. Kontit jakavat yhden virtuaalipalvelimen ja sisältävät vain ohjelman ajamiseen tarvittavat komponentit. Tämä mahdollistaa myös sen, että konttien ei tarvitse varata resursseja etukäteen, vaan kontteja voidaan luoda, ajaa ja sulkea tarpeen mukaan. Kaiken kaikkiaan kontit säästävät siis kehittäjien aikaa, tietokoneresursseja ja sitä myöten myös kustannuksia. (3.)

Konteista on myös paljon hyötyä pilvisiirtymisessä. Kontit tarjoavat samat ominaisuudet sekä pilvessä että konesalissa. Kontteja on helppo siirtää konesalista pilviympäristöön ilman muutoksia varsinaiseen ohjelmaan. Pilvessä oleva ympäristö on yleensä hyvin samanlainen kuin vastaava konesalin ympäristö, mikä helpottaa järjestelmän siirtämistä konesalista pilviympäristöön. Se mahdollistaa myös järjestelmän, jossa samaa ohjelmaa käytetään sekä konesalissa että pilviympäristössä. Konttien standardisoinnin ansiosta

ohjelmaan tehty muutos on helppo viedä automaattisesti ja samanaikaisesti sekä omaan konesaliin kuin myös ulkoiselle pilvipalvelimelle. Ilman kontteja jouduttaisiin ylläpitämään kahta erillistä järjestelmää konesalissa ja pilvessä, joihin molempiin muutokset pitäisi viedä erikseen.

## 2.3 AWS

Amazon Web Services (AWS) on yksi maailman suurimmista pilvipalveluiden tarjoajista. Se tarjoaa hyvin monipuolisesti erilaisia palveluita. Perinteisesti AWS on mielletty IaaS-palveluksi (Infrastructure as a Service), koska AWS:n vanhimmat palvelut ovat tarjonneet vuokrattavia palvelimia eri tarkoituksiin. Näitä ovat muun muassa EC2 (Amazon Elastic Compute Cloud) ja S3. EC2 on virtuaalipalvelin, jota käyttäjä voi vuokrata oman ohjelman ajamiseen. Sillä voi esimerkiksi ylläpitää nettisivun palvelinta tai käyttää alustana konteille. S3 puolestaan on työkalu, jota käyttäjän luomat ohjelmat ja muut palvelut voivat käyttää tiedon tallentamiseen. AWS:n palvelut ovat monipuolistuneet vuosien saatossa huomattavasti, eikä AWS:ää voi kutsua enää puhtaaksi IaaS-palveluksi. AWS tarjoaa käyttäjilleen monia valmiita SaaS-ohjelmistoja (Software as a Service) sellaisenaan käytettäväksi, mutta erityisesti AWS on kunnostautunut tarjoamaan erilaisia PaaS-kehitysalustoja (Platform as a Service) ohjelmistoille.

AWS tarjoaa nykyään myös paljon kehittyneempiä palveluita ohjelmistojen ajamiseen ja tiedon tallentamiseen pilvessä. ECS, EKS ja ROSA tarjoavat eri tasoisia konttien virtuaalipalvelimella ajamiseen liittyviä palveluita. Lambda, Lightsail ja Fargate -palveluilla voi puolestaan ajaa kontteja ja ohjelmia ilman virtuaalipalvelimia. Näihin palveluihin tutustutaan tarkemmin myöhemmin eri konttipalveluiden vertailun yhteydessä. AWS tarjoaa myös monia kehittyneitä tietokantapalveluita. Nämäkin palvelut on suunniteltu mahdollisimman yhteensopiviksi ja tukemaan yleisimpiä tietokantatyyppejä, kuten MySQL:ää, MariaDB:tä ja MongoDB:tä. AWS-tietokannat voi integroida osaksi käyttäjän

ohjelmistoa tai muita AWS-palveluita. Amazonin tietokantapalveluita ovat muun muassa Aurora, RDS ja DocumentDB.

AWS ei ole enää pelkkä palvelinvuokrauspalvelu, joka se oli vielä 2000-luvun ensimmäisellä vuosikymmenellä. AWS tarjoaa nyt myös monipuolisemman alustan, ei pelkästään ohjelman pyörittämiseksi, vaan myös sen kehittämiseksi. Lisäksi AWS:tä löytyy myös koneoppimiseen, robotiikkaan ja esineiden internetiin (Internet of Things) liittyviä palveluita. Se tarjoaa myös bisnekseen, analytiikkaan ja ryhmätyöhön liittyviä palveluita. Näiden palveluiden yksityiskohtainen läpikäyminen ei kuulu tämän työn skaalaan. Tarkoituksena on vain havainnollistaa AWS:n merkitystä erilaisille toimijoille ja tarjota mielekäs konteksti tämän työn varsinaiselle aiheelle. Seuraavassa osiossa käsitellään eri AWS:n konttiajotyökaluja yksityiskohtaisesti ja selvitetään, mikä niistä sopii parhaiten Ilmatieteen laitoksen tarkoituksiin.

### 3 AWS konttipalvelun valitseminen

Tässä osiossa kuvataan konttipalvelun valitseminen AWS:n tarjoamien pilvipalveluiden joukosta. Valittuun palveluun toteutetaan laskentaympäristö, jota Ilmatieteen laitos käyttää ennusteiden laatimiseen. Laskentaympäristön perustaminen ei kuulu tämän insinööriyön piiriin. Tämä osio jakautuu kahteen vaiheeseen. Ensimmäisessä vaiheessa kuvataan Ilmatieteen laitoksen palvelulle asettamat kriteerit. Ilmatieteen laitos haluaa, että valittava palvelu on mahdollisimman sopiva sen tarkoituksiin, minkä vuoksi on tärkeää asettaa kriteerit palvelun soveltuvuuden mittaamiseksi. Toisessa osiossa vertaillaan AWS:n tarjoamia konttipalveluita. Tarkoitus on käydä läpi sitä, miten ne toimivat ja mitä ominaisuuksia niillä on. Tarkoitus on vertailla palveluita keskenään ja tarkastella täyttävätkö ne Ilmatieteen laitoksen asettamat kriteerit. Osaa palveluista käsitellään vain pintapuolisesti ja varteenotettavimpia palveluita analysoidaan syvemmin.

#### 3.1 Ilmatieteen laitoksen kriteerit AWS-palvelulle

Aluksi tarkastellaan resurssien käytölle asetettuja kriteerejä. Konttiajot, joita Ilmatieteen laitos aikoo ajaa valittavassa palvelussa ovat melko kevyitä. Ajoja on kahdeksan, jotka kaikki ovat enintään muutaman minuutin mittaisia. Ajoja ei ajeta samanaikaisesti, vaan pääsääntöisesti yksitellen. Palvelulta ei siis vaadita valtavasti suorituskykyä. Tärkeämpää on optimoida resurssien käyttö. Ajot ajetaan kerran tunnissa ja Ilmatieteen laitos haluaa, että palvelu käyttää resursseja vain ajojen aikana ja on muulloin pois päältä. Palvelun täytyy siis pystyä luomaan kontit, ajamaan ne palvelimella ja lopuksi sulkemaan ne säämallien valmistuttua. Palvelimet eivät saa olla kiinteitä, vaan niitä täytyy voida ottaa käyttöön vasta tarpeen vaatiessa. Tällöin Ilmatieteen laitosta veloitetaan vain ajojen aikana käytetyistä resursseista.

Valittavan palvelun kuuluisi olla mahdollisimman yhteensopiva olemassa järjestelmien kanssa. Ilmatieteen laitoksella on jo konesalissa käytössään sääennusteen laatimiseen tarvittavat konttiajot, jotka on nyt tarkoitus kahdentaa pilviympäristöön. Ilmatieteen laitoksella on käytössään OpenShiftillä toteutettu laskentaympäristö ja on tärkeää, että uusi laskentaympäristö olisi mahdollisimman yhteensopiva konesalissa olevan ympäristön kanssa. Ideaalissa toteutuksessa järjestelmät olisivat käytännössä identtisiä, jolloin muutokset ja päivitykset voitaisiin viedä yhdellä kertaa molempiin järjestelmiin. OpenShift on Kubernetes pohjainen, joten myös Kubernetesistä tukeva palvelu olisi teoriassa mahdollinen valinta, vaikka se ei olisikaan täysin yhteensopiva nykyisten järjestelmien kanssa.

Palvelun ylläpidettävyys ja helppokäyttöisyys ovat tärkeitä kriteerejä palvelun käytettävyyden kannalta. Valittavan palvelun täytyy olla helppo ottaa haltuun ja helppo opettaa uusille palvelun käyttäjille. Tätä tukee tarjolla oleva dokumentaatio. Mitä dokumentaatiota AWS tarjoaa palvelun tueksi? Löytyykö palvelusta paljon sen käyttäjien tekemää opetusmateriaalia? Palvelun käytettävyyteen vaikuttaa myös sen käyttämiseen vaadittavat työkalut. Voiko palvelua ylläpitää samoilla työkaluilla, jotka ovat jo nyt käytössä Ilmatieteen laitoksella? Jos palvelu tukee OpenShiftiä ja Pythonia, on sen haltuun ottaminen helpompaa nykyisiä Ilmatieteen laitoksen järjestelmiä käyttäville. Tärkeää on myös huomioida palvelun yleinen helppokäyttöisyys. Millainen käyttöliittymä palvelulla on? Ovatko palvelun toiminnot selkeitä ja intuitiivisia?

Palvelun käyttöönotossa kannattaa huomioida myös mahdolliset tulevat laajennukset. Edellä mainittiin, että palvelun suorituskyky ei ole kriittisen tärkeää, koska ajettavat kontit ovat kevyitä ja niitä ajetaan vain muutamien minuuttien ajan tunneittain. Tulevaisuudessa pilvipalvelulta vaadittu suorituskyky voi kuitenkin kasvaa huomattavasti, joten on hyvä arvioida palvelun suorituskykyä myös suuremmilla työkuormilla. Lisäksi on hyvä tarkastella palvelun laajentuvuutta yleisellä tasolla. Kuinka helppo palvelun laajentaminen on ja minkälaisia tehtäviä palvelussa on mahdollista suorittaa? Myös nyt toteutettavan palvelun skaalautuvuus on tärkeää. Nyt ajettavat työkuormat ovat hyvin samanlaisia, mutta tulevaisuudessa on mahdollista, että

työkuormien koot vaihtelevat ajosta toiseen tai päivästä toiseen, joten palvelun täytyy kyetä mukautumaan työkuorman vaihteluihin.

Valittavan palvelun toimintaa olisi hyvä pystyä seuraamaan. Palvelun pitää raportoida käynnissä olevista ja menneistä ajoista ja siitä onko ne ajettu oikein. Jos ajo epäonnistuu, täytyy palvelun antaa selkeitä virheilmoituksia virheiden paikantamiseksi. Lisäksi on tärkeää tietää, kuinka paljon resursseja palvelu on käyttänyt ja kuinka paljon palvelun käyttö on maksanut. Ilmatieteen laitos haluaa lokitietoja sekä teksti että graafisessa muodossa. Kuvaajista saa hyvän käsityksen käytetyistä resursseista ja yleiskuvan ajojen toiminnasta, kun taas lokitiedot ja virheilmoitukset ovat hyödyllisiä ongelman ratkaisussa ja tarkemmassa palvelun toiminnan valvonnassa.

## 3.2 AWS-konttityökalut

Seuraavaksi tarkastellaan AWS:n tarjoamia konttipalveluita. Tarkoituksena on tutkia niiden ominaisuuksia ja vertailla niitä toisiinsa. Vertailun pohjana on Ilmatieteen laitoksen asettamat kriteerit valittavalle palvelulle. Varsinaiseen vertailuun kuuluu kolme ehdokasta. AWS:n palvelut ECS, EKS ja ROSA. Nämä palvelut on todettu Ilmatieteen laitoksen kanssa suullisesti tehdyn arvion perusteella parhaiksi vaihtoehtoiksi ja siksi tässä vertailussa keskitytään näihin kolmeen palveluun. Vertailun lopussa käsitellään kuitenkin myös muutamaa muuta AWS:n palvelua ja niiden ominaisuuksia. Näitä palveluita käsitellään vain pintapuolisesti ja tarkoituksena on lähinnä selvittää sitä miksi ne eivät sovellu Ilmatieteen laitoksen tarkoituksiin.

### 3.2.1 ECS

ECS on AWS-palvelu, jota käytetään konttien ajamiseen AWS-pilvessä. ECS pystyy hallinnoimaan kontteja koko niiden elinkaaren ajan. Se pystyy siis luomaan, ajamaan ja sulkemaan kontteja tarpeen mukaan. ECS pystyy myös

jakamaan kuormitusta eri virtuaalikoneiden välillä. ECS hallinnoi kontteja ECS-klusterin avulla. Tämä klusteri sisältää kaikki konttien hallinnoimiseen tarvittavat työkalut. ECS on kokonaan AWS-pohjainen palvelu ja sen käyttöönotto vaatii tuotavien konttien yhteensopivuutta ECS:n kanssa. ECS:ää voi käyttää kahdella eri tavalla. Sitä voi käyttää yhdessä EC2-instanssien kanssa tai sitä voi käyttää Fargate-palvelun kanssa. (4.)

ECS tarvitsee EC2-instansseja (virtuaalikoneita) konttien ajamiseen. EC2-instansseille asennetaan konttiajoagentti (esimerkiksi Docker), joka ajaa kontteja. Lisäksi EC2-instanssit saavat ECS-agentin, jonka avulla ECS-klusteri kommunikoi EC2-instanssien kanssa. ECS ei pysty itse luomaan uusia virtuaalikoneita ja kontteja eli EC2-instanssien luonti ja hallinnointi kuuluu käyttäjälle. Käyttäjän täytyy siis itse huolehtia virtuaalikoneiden konfiguroinnista, käyttöjärjestelmistä ja tarvittavista resursseista. Toisaalta tämä mahdollistaa virtuaalikoneiden ja konttien täyden hallinnan. ECS-klusteri toimii EC2-instanssien yläpuolella ja jakaa työkuormaa EC2-instansseille.

Toinen vaihtoehto on käyttää ECS:ää Fargate-palvelun kanssa. Fargaten etuna on se, että konteille ei tarvitse asettaa virtuaalikonetta tai resursseja. Fargate-palvelu hoitaa tämän automaattisesti tarpeen mukaan eli konteille asetetaan virtuaalikoneet ja resurssit vasta kun niille on tarve. Tämä mahdollistaa sen, että käyttäjä maksaa ainoastaan käyttämistään resursseista, koska konttien ajamiseen ei ole etukäteen varattu resursseja. Käyttäjän tarvitsee siis huolehtia ainoastaan kirjoittamastaan ohjelmasta. ECS ja Fargate -palvelut hoitavat kokonaan konttien hallinnan. Fargate-palvelua käytettäessä käyttäjällä ei ole kuitenkaan enää mahdollisuutta hallita konttien ajamiseen käytettäviä virtuaalikoneita. (5.)

ECS täyttää ison osan Ilmatieteen laitoksen asettamista kriteereistä. Jos sitä käytetään Fargate-palvelun kanssa, kykenee se suorittamaan Ilmatieteen laitoksen ajastetut ajot niin, ettei se kuluta resursseja ajojen ulkopuolella. ECS kykenee myös skaalautumaan vaihteleviin työkuormiin, mikä on tärkeää, jos Ilmatieteen laitos haluaa kasvattaa olemassa olevaa työkuormaa tai tuoda uusia ohjelmistoja pilviympäristöön. Siihen voi liittää myös AWS:n muita pilvipalveluita,

joilla voi esimerkiksi hoitaa monitorointia CloudWatchin avulla tai luoda tietokantoja AWS:n eri tietokantapalveluiden avulla. ECS:stä sekä EC2-instansseista ja Fargate-palvelusta on tarjolla runsaasti dokumentaatiota ja opetusmateriaalia. Amazon itse tuottaa paljon materiaalia näistä palveluista, minkä lisäksi tarjolla on myös paljon kolmansien osapuolien tuottamaa opetusmateriaalia.

ECS:n käyttöön Ilmatieteen laitoksella liittyy kuitenkin myös monia ongelmia. ECS on kokonaan AWS-pohjainen palvelu eli se ei ole yhteensopiva Ilmatieteen laitoksen nykyisen OpenShiftiä käyttävän ympäristön kanssa. Tämä tarkoittaisi sitä, että Ilmatieteen laitoksen täytyisi ylläpitää kahta erilaista ympäristöä konesalissa ja pilviympäristössä. ECS:n käyttöönottamiseen vaadittaisiin suuria muutoksia pilveen vietävään järjestelmään. Tulevien käyttäjien täytyisi totutella käyttämään täysin uudenlaista järjestelmää ja hallinnoida sitä samanaikaisesti vanhan järjestelmän kanssa. Kaikki muutokset täytyisi viedä erikseen sekä konesaliin että pilviympäristöön. ECS:n käyttö tekisi Ilmatieteen laitoksesta myös riippuvaisen AWS-ympäristöstä, koska siellä luodut klusterit ja kontit eivät ole yhteensopivia muiden pilvipalveluiden tai konttien hallintajärjestelmien kanssa. ECS tarvitsee myös erillisen ECR-säiliön, josta hakea tuoreet konttien luontitiedostot (images) ajettavaksi.

### 3.2.2 EKS

EKS on hyvin samankaltainen ECS:n kanssa. EKS pystyy luomaan, ajamaan ja sulkemaan kontteja ja se jakaa työkuormaa eri konttien välillä. ECS:n tapaan se käyttää klusteria (EKS-klusteri) konttien hallitsemiseen. Nämä klusterit sisältävät kaikki tarpeelliset työkalut konttien hallintaan. EKS on suunniteltu yhteensopivaksi Kubernetes-hallintajärjestelmän kanssa, mikä mahdollistaa Kubernetesella luotujen konttien tuomisen AWS-pilveen. Tämä helpottaa AWS-pilviympäristön rakentamista, jos organisaatio on ennestään käyttänyt Kubernetesistä konttien hallintaan. ECS:n tapaan se voi käyttää joko EC2-

instansseja tai Fargate-palvelua konttien ajamiseen. Lisäksi EKS:ää voi käyttää Nodegroup-palvelun kanssa. (6.)

EKS:n toimintaperiaate on hyvin samankaltainen kuin ECS:ssä. EC2-instansseja käytettäessä käyttäjä huolehtii itse virtuaalikoneiden luonnista, käyttöjärjestelmistä ja resursseista. EC2-instansseille asennetaan konttiajoagentti, joka vastaa konttien ajamisesta. Lisäksi virtuaalikoneille asennetaan tarvittavat Kubernetes-prosessit EKS-klusterin ja EC2-instanssien välistä kommunikointia varten. Fargate-palvelua käytettäessä käyttäjän ei tarvitse huolehtia konttien ajamiseen tai hallintaan tarvittavista resursseista. Samalla tavalla kuin ECS:ssä, Fargate hallinnoi ja sääntelee virtuaalikoneita ja niiden resursseja niin, että käyttäjä maksaa vain ajon aikana käytetyistä resursseista.

EKS tarjoaa myös kolmannen vaihtoehdon konttien alustaksi. EKS tarjoaa mahdollisuuden jakaa EC2-instansseja erityisiksi ryhmiksi (Nodegroups). Nämä ryhmät sisältävät kaikki tarvittavat työkalut ja konfiguroinnit EC2-instanssien luomiseksi. Tämä helpottaa uusien EC2-instanssien luomista, koska kaikki sen tarvitsemat työkalut asentuvat automaattisesti EC2-instanssin oman ryhmän konfiguroinnin mukaan. Nodegroup-ryhmät eivät skaalaudu automaattisesti eli käyttäjän täytyy itse luoda uudet EC2-instanssit. Nodegroup-ryhmät kuitenkin tarjoavat mahdollisuuden ryhmitellä virtuaalikoneet loogisiksi ryhmiksi, mikä helpottaa ja selkeyttää uusien ja olemassa olevien EC2-instanssien hallintaa. (5.)

EKS täyttää suurimman osan Ilmatieteen laitoksen asettamista kriteereistä, joten se on vahva ehdokas alustaksi uudelle laskentaympäristölle. ECS:n tapaan sekin pystyy ajamaan tarvittavat ajot ilman, että se kuluttaisi turhaan resursseja. Lisäksi se pystyy skaalautumaan tulevaisuuden vaihteleviin työkuormiin ja uusiin pilveen tuotaviin palveluihin. ECS:n tapaan EKS:stä löytyy runsaasti dokumentaatiota ja opetusmateriaalia sen käytön helpottamiseksi. Kubernetes maailma tuo mukanaan paljon etuja mitä ECS ei pystynyt tarjoamaan. Se parantaa yhteensopivuutta muiden pilviympäristöjen ja konttien hallintajärjestelmien kanssa, koska Kubernetesistä tuetaan lähes kaikissa

palveluissa. EKS:ssä voi hyödyntää myös Kubernetesen mukanaan tuomia palveluita, joilla yhdessä AWS:n tarjoamien palveluiden kanssa voi hoitaa muun muassa monitorointia, tietokantoja ja sovelluskehitystä.

EKS tarjoaa myös enemmän yhteensopivuutta Ilmatieteen laitoksen nykyisen OpenShiftiä käyttävän järjestelmän kanssa. Sekä EKS että OpenShift perustuvat Kubernetesiin, mikä mahdollistaisi osittaisen yhteensopivuuden konesalin ja pilviympäristön järjestelmien välillä. Tämän ansiosta pilveen vietävään järjestelmään tarvitsisi tehdä vähemmän muutoksia. Palvelu olisi myös osittain jo tuttu Ilmatieteen laitoksen työntekijöille Kubernetes-yhteensopivuuden ansiosta. On kuitenkin muistettava, että EKS ja Kubernetes eivät ole sama asia kuin OpenShift. Järjestelmät olisivat siis edelleen erilaiset, joten Ilmatieteen laitoksen täytyisi edelleen ylläpitää kahta melko erilaista järjestelmää ja niiden käyttäjien täytyisi osata käyttää kahta erilaista järjestelmää. ECS:n tapaan EKS tarvitsee myös ECR-säiliön kontti-imagien säilyttämiseksi.

### 3.2.3 ROSA

ROSA on AWS-palvelu, joka mahdollistaa OpenShift-konttialustan käytön AWS-pilvessä. ROSA-palvelua tukevat sekä Red Hat että Amazon, mutta Red Hat vastaa varsinaisesta ROSA-palvelun kehityksestä. ROSA-klusterin luonti ja säilytys tapahtuu AWS-pilvessä, mutta Red Hat vastaa klusterin hallinnasta ja tuesta. ROSA-palvelu tarjoaa samat palvelut kuin EKS tai ECS yhdistettynä Fargate-palvelun kanssa. ROSA-palveluun on siis sisäänrakennettuna kaikki samat palvelut kuin EKS:ssä tai ECS:ssä ja Fargatessa. Se pystyy luomaan ja hallitsemaan kontteja automaattisesti ja jakamaan työkuormaa niiden välillä. Lisäksi sillä on sisäänrakennettuna muitakin ominaisuuksia, joita EKS, ECS tai Fargate ei tarjoa. (7.)

EKS ja ECS vaativat toimiakseen ECR-palvelun, joka säilyttää konttien luontiin tarvittavat imaget ennen niiden tuomista EKS ja ECS palveluihin. Vastaava

palvelu on OpenShift-palvelussa sisäänrakennettuna. Käyttäjän ei tarvitse huolehtia konttien luontitiedostojen (image) säilytyksestä tai konttien luonnista, sillä OpenShift osaa tehdä tämän automaattisesti. OpenShift tarjoaa tämän lisäksi työkalut kontti-imagen luontiin. Käyttäjän kirjoittama koodi voidaan muuttaa kontti-imageksi OpenShiftissä olevien S2I-työkalujen (Source to Image) avulla. Käyttäjä voi myös halutessaan myös luoda itse tarvittavat kontti-imaget ja tuoda ne OpenShiftiin tai käyttää OpenShiftiä yhdessä ECR-rekisterin kanssa. OpenShiftin vahvimpiin ominaisuuksiin kuuluu nimenomaan suuri joustavuus ja modulaarisuus. OpenShift tarjoaa ison määrän palveluita koodin kontittamiseen, konttien hallintaan ja monitorointiin, mutta se tukee myös monia muita ulkoisia palveluita.

OpenShift tarjoaa monia hyödyllisiä työkaluja monitorointiin. OpenShiftissä on sisäänrakennettuna monitorointiin ja raportointiin liittyviä työkaluja, minkä lisäksi se tukee myös ulkopuolisia työkaluja, kuten Promethiustaja Grafanaa.

OpenShift tarjoaa myös Pipelines nimisen CI/CD-palvelun eli palvelun, jonka avulla voidaan tuoda pieniä jatkuvia muutoksia versionhallintaan ja edelleen julkaistaviksi. Pipelines perustuu avoimen lähdekoodin Tekton järjestelmään eli tämäkin palvelu tukee OpenShiftin joustavuutta ja modulaarisuutta. Merkittävää on myös se, että OpenShiftiä ajetaan AWS-pilvessä ROSA-palvelun avulla. Tämä tarkoittaa sitä, että käyttäjällä on pääsy myös kaikkiin AWS:n tarjoamiin palveluihin.

OpenShiftin konttien hallinta muistuttaa EKS:n ja ECS:n vastaavia palveluita yhdistettynä Fargate-palveluun. Tämä tarkoittaa sitä, että käyttäjän ei tarvitse huolehtia konttien hallinnasta tai niiden luonnista. OpenShift pystyy luomaan, hallitsemaan ja poistamaan kontteja automaattisesti sen mukaan, kuinka paljon laskentatehoa milloinkin tarvitaan. OpenShift pystyy jakamaan työkuormaa tasaisesti eri virtuaalikoneiden kesken (load balancing). Tämä tarkoittaa myös sitä, että käyttäjällä ei ole pääsyä virtuaalikoneiden konfiguraatioon tai sen käyttämiin resursseihin. OpenShift ei siis sovellu tilanteisiin, jossa käyttäjä haluaa itse määritellä tarkasti virtuaalikoneiden toiminnan. OpenShiftin yksinkertaisuus takaa kuitenkin palvelun skaalautuvuuden ja sen käytön

sujuvuuden. OpenShift tarjoaa monipuoliset palvelut uusien virtuaalikoiden ja päivityksien tekemiseksi. (8.)

ROSA soveltuu erinomaisesti Ilmatieteen laitoksen pilviratkaisuksi. Se täyttää kaikki Ilmatieteen laitoksen asettamat kriteerit. Ilmatieteen laitoksen nykyiset järjestelmät toimivat konesaleissa OpenShiftillä. Näiden palveluiden tuominen AWS-pilveen olisi yksinkertaista, koska ROSA on suunniteltu OpenShift-ympäristöksi AWS-pilvessä. Myös palvelun käytön opetteleminen on helppoa, koska tulevat käyttäjät ovat jo oppineet OpenShiftin käytön. ROSA on myös EKS:n ja ECS:n tavoin helppokäyttöinen ja siitä löytyy runsaasti opetusmateriaalia. ROSA on myös yhteensopiva muiden pilvipalveluiden kanssa, koska OpenShift käyttää hyväkseen Kubernetes-järjestelmää. ROSA ja OpenShift sisältävät myös monia muita hyödyllisiä avoimen lähdekoodin sovelluksia, joilla voi tuottaa muun muassa kuvaajia ja lokitietoja.

### 3.3 Muut palvelut/työkalut

#### 3.3.1 Batch

AWS Batch on palvelu, jolla voi ajaa kontitettuja Batch- ja koneoppimisen -ajoja AWS-pilvessä. Se soveltuu parhaiten tilanteisiin, jossa käsitellään ja analysoidaan suuria datamääriä. Batchin avulla voidaan priorisoida eri ajoja niiden ominaisuuksien perusteella. Eri ajoille voi asettaa prioriteetin manuaalisesti tai Batch voi itse priorisoida ajoja niiden tarvitsemien resurssien perusteella. Batchin pystyy myös automaattisesti mukautumaan vaihteleviin työkuormiin vähentämällä tai lisäämällä laskentaresursseja. Batch voi myös hyödyntää muita AWS-pilven palveluita ja toimia itse osana laajempaa laskentaympäristöä. (9.)

Batch-palvelu ei sovellu täysin Ilmatieteen laitoksen tarpeisiin. Ilmatieteen laitoksen suunnittelemat konttiajot käynnistyvät kerran tunnissa ja niitä ei tarvitse priorisoida, koska ne ovat hyvin kevyitä. Ilmatieteen laitoksella on jo

valmiiksi kontitetut ajot, jotka eivät tarvitse Batch-ympäristöä. Ilmatieteen laitoksen tämänhetkiset järjestelmät konesaleissa hyödyntävät OpenShift-järjestelmää. Tämän järjestelmän siirtäminen Batch-palveluun vaatisi suuria muutoksia ja paljon resursseja. Batch-palvelun käyttöönotto vaatisi myös sen käyttäjien kouluttamista täysin uuteen ympäristöön, mikä sekin söisi resursseja.

### 3.3.2 Lightsail

Lightsail on AWS-palvelu, joka keskittyy tarjoamaan mahdollisimman yksinkertaisen ja helppokäyttöisen pilvipalvelun. Se soveltuu pienten ja yksinkertaisten applikaatioiden, tietokantojen ylläpitoon ja erityisesti yksityishenkilöiden käyttöön. Lightsailissa on tarjolla rajallinen määrä valmiiksi konfiguroituja palveluita, joista käyttäjä voi valita tarvitsemansa. Lightsail luo käyttäjälle instanssin, jossa on valmiiksi asennettuna kaikki käyttäjän tarvitsemat palvelut. Myös Lightsailin hinnoittelu on suunniteltu hyvin yksinkertaiseksi. Käyttäjälle on tarjolla muutamia eri paketteja, joiden hinta riippuu tarvittavasta muistin ja laskentatehon määrästä. Lightsail kykenee myös hetkellisesti ylittämään sille varatut resurssit, jos Lightsailin isännöimässä palvelussa on ruuhkaa. (10.)

Lightsail ei sovellu Ilmatieteen laitoksen tarkoitukseen. Vaikka Lightsail tukeekin konttiajoja, ei se silti tarjoa riittävän monipuolista palvelua Ilmatieteen laitoksen tarpeisiin. Suurin ongelma on sen staattisuus. Lightsail tarjoaa vain rajallisen määrän palveluita, jotka itsessään ovat staattisia ja valmiiksi konfiguroituja, joten käyttäjällä ei ole mahdollisuutta hallita käyttämäänsä infrastruktuuria. Palveluiden rajallisuus myös rajoittaa uusien palveluiden tuomista pilveen. Lisäksi Lightsail on kokonaan erillinen kokonaisuus muista AWS-palveluista eli se ei voi suoraan hyödyntää muita AWS-pilven palveluita.

### 3.3.3 Lambda

Lambda on AWS-palvelu, jolla voi ajaa koodia pilvessä ilman kiinteää serveriä. Lambdan perusyksikkö on funktio, joka määrää palvelun käyttäytymisen. Funktiota voi ajatella Lambdan konfiguraationa. Funktioita käytetään hallitsemaan Lambdassa pyörivää varsinaista ohjelmaa. Lambda on erittäin joustava, sillä funktiot tarjoavat mahdollisuuden käyttää Lambdaa melkein mihin tahansa. Lambdan hinnoittelu perustuu käytettyihin resursseihin eli käyttäjän ei tarvitse maksaa kiinteästä serveristä. Lambda kykenee automaattisesti kasvattamaan tai vähentämään laskentaresursseja tarpeen mukaan. Käyttäjän täytyy huolehtia ainoastaan Lambdan funktioista sekä kirjoittamastaan koodista. (11.)

Lambda täyttää osan Ilmatieteen laitoksen kriteereistä. Se on helppokäyttöinen ja siitä löytyy runsaasti dokumentaatiota. Se kykenee skaalautumaan käyttäjän tarpeisiin ja sen toiminnallisuutta on helppo laajentaa. Lambda kykenee myös hyödyntämään AWS:n muita pilvipalveluita, joka mahdollistaa muun muassa monitoroinnin. Lambdan huonona puolena on se, että sitä ei ole rakennettu erityisesti konttiajoympäristöksi. Lambdalla pystyy tekemään monenlaisia ympäristöjä, mutta se ei tarjoa valmiita työkaluja niiden rakentamiseksi, vaan kaikki rakennetaan funktioiden avulla. Ilmatieteen laitoksen nykyiset järjestelmät käyttävät OpenShiftiä, mikä tarkoittaa sitä, että pilveen täytyisi perustaa aivan erilainen ympäristö konttien ajamiseksi.

### 3.4 Konttipalvelun valinta

Valittavasta konttipalvelusta pidettiin kokous tämän vertailun tekijän ja Ilmatieteen laitoksen edustajien kesken. Kokouksessa päädyttiin valitsemaan ROSA-palvelu pilvialustaksi uudelle laskentaympäristölle. ROSA-palvelu oli ainoa, joka täytti kaikki Ilmatieteen laitoksen asettamat kriteerit, mikä teki sen valinnasta loogisen. Seuraavassa osiossa piti käsitellä ROSA-palvelun

käyttöönottoa ja Ilmatieteen laitoksen laskentaympäristön perustamista AWS-pilveen. Sen sijaan seuraavassa osiossa kuvataan pienemmän ohjelmointiprojektin toteutus. Tätä ennen käsitellään kuitenkin vielä lyhyesti sitä, miksi ROSA-palvelu valittiin ja miksi se oli parempi vaihtoehto verrattuna muihin vertailussa mukana olleille AWS-palveluille.

Tärkein ROSA-palvelun ominaisuus Ilmatieteen laitoksen kannalta on sen yhteensopivuus OpenShiftin kanssa. Tämä yhteensopivuus mahdollistaa olemassa olevan OpenShift ympäristön sujuvan siirtämisen pilveen, identtisen käyttökokemuksen nykyisen ja uuden järjestelmän välillä ja samanaikaiset muutokset kumpaankin ympäristöön. Muilla vertailussa mukana olleilla palveluilla ei ole näitä ominaisuuksia. EKS on Kuberernetes pohjainen palvelu, kuten myös OpenShift, mutta niitä ei voi kuitenkaan ylläpitää samanaikaisesti, koska ylätasolla ne ovat erilaisia järjestelmiä. ECS puolestaan on kokonaan AWS-pohjainen järjestelmä, mikä tarkoittaisi kokonaan erilaista käyttökokemusta ja ylläpitoa konesalin ja pilviympäristön välillä.

ROSA-palvelun valintaan vaikutti myös sen tarjoamat palvelut. EKS ja ECS tarvitsevat erilliseen ECR-säiliön kontti-imageja varten. ROSA-palvelu pystyy luomaan ja säilömään kontti-imaget suoraan koodista, minkä lisäksi ROSA:sta löytyy runsaasti eri ohjelmoijia ja ohjelmistonkehitystä tukevia palveluita. ROSA erottuu muista AWS:n palveluista juuri sen modulaarisuudella. EKS ja ECS on mahdollista yhdistää muihin AWS:n tarjoamiin palveluihin. Tämä pätee myös ROSA-palveluun, mutta tämän lisäksi ROSA-palveluun on mahdollista yhdistää monia AWS-pilven ulkopuolisia palveluita OpenShiftin kautta. Kaikki vertailut palvelut tarjoavat laajan kokonaisuuden palvelun ylläpitämiselle ja kehittämiselle, mutta vain ROSA pystyy luomaan kontit suoraan koodista ja vain ROSA:ssa on tarjolla niin paljon AWS-pilven ulkopuolisia palveluita.

## 4 Docker-ohjelmointiprojekti

### 4.1 Projektin kuvaus

Tässä osiossa kuvataan pienen konttiympäristön ja sitä käyttävän ohjelman toteutus. Sen tarkoituksena on täydentää tämän opinnäytetyön aikaisempia osioita antamalla esimerkin konttien toiminnasta ja sen tuomista eduista. Tämän osion piti olla alun perin osa Ilmatieteen laitokselle toteutettavaa projektia, jossa AWS-pilveen perustetaan konttiympäristö aiemmassa osiossa valittua AWS:n ROSA-palvelua hyväksi käyttäen. Tämä osoittautui kuitenkin liian vaikeaksi johtuen siitä, että kirjoittaja ei pystynyt haalimaan riittävää osaamista projektin toteuttamiseksi annetun aikataulun puitteissa. Tämän takia tätä osiota varten toteutettua projektia supistettiin ja sen korvasi pienempi paikallisesti toimiva Ilmatieteen laitoksen avointa säädataa hyödyntävä konttiympäristö. Projekti toimii kuitenkin edelleen hyvänä esimerkkinä konttiympäristön toiminnasta ja sen tarkoituksena onkin täydentää lukijalle kuvaa konttiympäristön toiminnasta ja sitä mihin sitä voi käyttää.

Tässä projektissa hyödynnettiin Ilmatieteen laitoksen avointa dataa. Projektia suunniteltaessa haluttiin hyödyntää Ilmatieteen laitoksen avointa säädataa, koska se sopii hyvin tämän opinnäytetyön teemaan ja muihin osioihin. Valittavan säädatan kuului olla sellaista, josta voisi erotella ohjelman käyttäjän kannalta oleellista tietoa ja jota voisi hyödyntää sellaisenaan ilman muita tietolähteitä. Esimerkiksi pelkän lämpötiladatan hyödyntäminen olisi vaikeaa, koska lämpötilan tuntemukseen vaikuttaa moni muukin seikka, kuten tuulennopeus ja ilmakosteus. Tietolähteeksi valikoitui Ilmatieteen laitoksen voimassa olevat varoitukset. Tämä sopi avoimelle säädatalle asetetut kriteerit. Säävaroitukset ovat käyttäjille oleellisia sellaisenaan ja niistä on helppo erotella tärkeimmät tiedot käyttäjän luettavaksi.

Ohjelma hakee siis Ilmatieteen laitoksen avoimesta tietokannasta voimassa olevat varoitukset. Varoitukset ovat XML-muodossa ja niissä on paljon loppukäyttäjälle tarpeetonta tai vaikeasti luettavaa tietoa, minkä takia saatua tietoa pitää karsia ja muuttaa luettavampaan muotoon loppukäyttäjää varten. Tämän jälkeen

varoitukset voi asettaa luettavaksi loppukäyttäjälle. Tämä tapahtuu FastAPI-työkalun avulla. FastAPI-työkalulla tiedot asetetaan luettavaksi localhostiin, josta loppukäyttäjä voi lukea Ilmatieteen laitoksen voimassa olevat varoitukset ihmisluettavassa muodossa. Ohjelman kontitettu osa on se, joka hakee varoitukset Ilmatieteen laitoksen tietokannasta, muuttaa ja karsii tiedot luettavaan muotoon ja asettaa ne loppukäyttäjälle luettavaksi localhostiin.

## 4.2 Projektin toteutus

### 4.2.1 Käytetyt työkalut

Projekti toteutettiin Python ohjelmointikielellä. Tähän projektiin se sopi yksinkertaisuutensa ansiosta ja koska datan hallinnointi sekä FastAPI-työkalun käyttö sen avulla toimivat mutkattomasti. FastAPI työkalu valikoitui niin ikään yksinkertaisuutensa vuoksi ja koska se on suunniteltu toimimaan nimenomaan Python ohjelmointikielen kanssa. FastAPI on verkkokehys (framework), jonka avulla voi rakentaa ohjelmointirajapintoja. FastAPI on suunniteltu nimenomaan yksinkertaiseksi ja virtaviivaiseksi käyttää, jotta sen oppiminen ja käyttö olisi mahdollisimman virtaviivaista.

Ohjelman kontitus tapahtuu Dockerin avulla. Tässä projektissa ei hyödynnetä mitään konttien hallintajärjestelmää, kuten Kubernetesiä. Alun perin tarkoituksena oli hyödyntää AWS-pilven konttipalveluita, mutta projektin supistuksen takia tämä kävi tarpeettomaksi. Tämä projekti tarvitsi toimiakseen vain yhden Docker tiedoston ja yhden kontin, joten tehokkaammalle konttienhallinnalle ei ollut tarvetta. Projektissa hyödynnettiin myös jonkin verran tekoälyä. Käytettyä koodia ei ole suoraan ohjelmoitu tekoälyn avulla. Tekoälyä käytettiin malliesimerkkien luomiseen, jotka havainnollistivat aina tietyn ohjelmointitekniikan tai kirjaston toimintaa. Lisäksi tekoälyä käytettiin ohjelmointivirheiden etsimiseen. Projektissa käytettiin ChatGPT-tekoälyä, koska se soveltuu hyvin pienten ohjelmointiesimerkkien luontiin, minkä lisäksi se osaa selittää kirjoittamansa koodin, muiden ohjelmistojen ja kirjastojen toimintaa.

## 4.2.2 Ohjelman toiminta

Seuraavaksi käsitellään tämän projektin kontitettavaa osaa hieman tarkemmin. Se koostuu kahdesta eri osasta. Ohjelma ei sisällä luokkia, mutta toiminnallisesti se koostuu kahdesta eri osasta ja ne on jaoteltu omiin tiedostoihinsa. Toinen tiedosto (`dataHandler.py`) sisältää tarvittavat toiminnot datankäsittelyyn. Se pystyy hakemaan Ilmatieteen laitoksen tietokannasta voimassa olevat varoitukset, missä se hyödyntää `requests`-kirjastoa. Ohjelma lukee tarvittavat tiedot ja tallentaa saadun XML-datan omaan XML-tiedostoonsa jatkokäsittelyä varten. Ilmatieteen laitoksen data sisältää paljon metatietoa, joka ei ole loppukäyttäjälle oleellista. Ohjelma erottelee datasta vain varsinaiseen varoitukseen liittyvät tiedot, kuten varoituksen kuvauksen, otsikon sekä aika- ja paikkatiedot. Osa datasta, kuten esimerkiksi aikatiedot eivät ole ihmisluettavassa muodossa, joten ohjelma muuttaa ne ymmärrettävään muotoon.

Ohjelman toinen (`app.py`) osa vastaa tiedon saatavuudesta localhostissa. Tämä on ohjelman varsinainen ajettava tiedosto ja se hakee ja parsii tarvittavat tiedot hyödyntäen ohjelman toista toiminnallista osaa. Tämän tehtyään se käyttää `FastAPI`:n `GET`-metodia asettaen käsitellyt tiedot localhostista saataviksi. Varsinaiseen ohjelman lisäksi tähän projektiin kuuluu myös kaksi muuta tiedostoa, jotka molemmat liittyvät projektin Docker-ympäristöön. Toinen sisältää listan ohjelman Docker-kontissa tarvitsemista kirjastoista. Toinen tiedosto on varsinainen Docker-tiedosto, joka sisältää tarvittavat Docker-komennot ohjelman imagen luomiseksi. Kun kontti luodaan käyttäen tätä imagea, tulevat Ilmatieteen laitoksen voimassa olevat varoitukset saataville localhostiin.

Edellä on kuvattu projektin kontitettava osuus, mutta projektiin kuuluu myös datan käyttöä simuloiva pieni ohjelma, joka hakee valmiit tiedot localhostista. Kyseessä on ohjelma, joka listaa haettujen varoitusten otsikkotiedot, joita painamalla käyttäjä saa klikatun varoituksen tarkemmat tiedot esiin. Nämä tiedot sisältävät tarkemman kuvauksen varoituksesta sekä varoituksen aika- ja paikkatiedot.

Varsinainen ohjelma koostuu yhdestä Python tiedostosta (main.py). Ohjelma käyttää hyväkseen Tkinter-rajapintaa, jonka avulla se rakentaa yksinkertaisen käyttöliittymän säävaroitusten esittämiseksi.

### 4.3 Projektin itsearviointi

Tämän projektin toteutus sujui loppujen lopuksi kohtalaisesti, mutta kohtasin työtä tehdessäni monia vaikeuksia, jotka hidastivat sekä hankaloittivat työn etenemistä ja johtivat lopulta projektin merkittävään supistumiseen. Projektin alkuperäinen tavoite, jossa minun piti kahdentaa osa Ilmatieteen laitoksen sään enustamiseen käytettävistä järjestelmistä AWS-pilveen, muuttui yksinkertaiseksi paikallisesti toimivaksi avointa säädataa ja kontteja hyödyntäväksi ohjelmaksi. Projektin alkuperäinen ongelma olikin sen liian kunnianhimoiset tavoitteet. Lähtiessäni toteuttamaan tätä projektia, en hahmottanut kuinka suuresta kokonaisuudesta on kyse ja kuinka paljon vaaditaan sen oppimiseksi. Minulla oli projektin alussa optimistinen ja kunnianhimoinen käsitys omasta kyvystäni oppia nopeasti kaikki kontittamiseen ja AWS-pilveen liittyvät seikat.

Tämä johtui ainakin osittain siitä, että tämän opinnäytetyön ensimmäinen osuus oli sujunut ilman suurempia ongelmia. Olin oppinut valtavasti teoreettista tietoa siitä, miten kontit toimivat ja miten niitä voi hyödyntää AWS-pilvessä. Minulta puuttui kuitenkin käytännön kokemus konttien ja AWS-pilven käytöstä. Työn alkaessa kävi nopeasti selväksi, että en saisi projektia valmiiksi alkuperäisen suunnitelman ja aikataulun puitteissa. Suunnitelmaa supistettiin useita kertoja, kunnes lopulta päädyttiin nykyiseen projektiin. Projektin venymisestä ja siihen liittyvistä vaikeuksista koitui minulle myös henkisiä paineita, jotka myös vaikuttivat projektin etenemiseen negatiivisesti. Realistisempi tavoite olisikin ollut juuri tämän projektin kaltainen kokonaisuus, jota minun olisi ollut helpompi hallita ja jota tekemällä olisin voinut helpommin oppia kontittamiseen liittyviä asioita.

Näiden vaikeuksien jälkeen lopullisen projektin toteuttaminen sujui melko kivuttomasti. Minun oli helppo oppia kontittamiseen liittyvät asiat lukemalla Docker-

järjestelmään liittyvää dokumentaatiota ja katsomalla siihen liittyviä opetusvideoita. Lopullinen projekti toteutui täysin uuden suunnitelman mukaisesti ja se tuotti täysin toimivan esimerkin konttien toiminnasta. Teemaltaan se sopii hyvin tähän opinnäytetyöhön, koska se hyödyntää Ilmatieteen laitoksen avointa dataa ja havainnollistaa ainakin osaa tämän opinnäytetyön kontteihin liittyvistä sisällöistä. Yksinkertaisuutensa vuoksi projektin kokonaisuus jäi kuitenkin hieman vaillinaiseksi, eikä se kuvaa esimerkiksi pilvipalveluiden toimintaa ollenkaan ja konteistakin se antaa hieman puutteellisen kuvan. Kaiken kaikkiaan pidän projektia kuitenkin onnistuneena ja olen tyytyväinen siitä, miten se täydentää opinnäytetyötäni.

## 5 Yhteenveto

Lopuksi käsitellään vielä tätä insinööriyötä kokonaisuutena ja tarkastellaan saavutettuja ja saavuttamattomia tavoitteita sekä miten tässä projektissa käsitellyjä asioita voidaan hyödyntää ja laajentaa tulevaisuudessa. Projektin kaksi päätaavoitetta olivat sopivan AWS-palvelun valitseminen Ilmatieteen laitoksen tarpeisiin ja laskentaympäristön perustaminen AWS-pilveen valittua palvelua käyttäen. Ensimmäinen tavoite toteutui kokonaisuudessaan. AWS-pilven konttien hallintajärjestelmiä vertailtiin tarkasti ja kolmesta pääehdokkaasta valikoitui Ilmatieteen laitokselle kaikkien mielestä paras vaihtoehto. Valittu ROSA-palvelu valikoitui joustavuutensa ja yhteensopivuutensa ansiosta muiden joukosta. Ilmatieteen laitos otti ROSA:n käyttöön suunnitelman mukaisesti, vaikka sitä ei tässä insinööriyössä hyödynnettykään.

Projektin toinen tavoite jäi lähes kokonaan saavuttamatta. Ilmatieteen laitos otti ROSA-palvelun käyttöönsä ja laskentaympäristön rakentaminen AWS-pilveen on aloitettu, mutta tätä työtä ei ole tehty tämän insinööriyön aikana. Insinööriyön tavoitteet olivat tämän osion osalta liian kunnianhimoiset, johtuen riittämättömästä osaamisesta ja aikataulupaineista. Tätä pystyttiin onneksi osin korvaamaan pienemmällä ohjelmointiprojektilla. Vaikka tämä projekti ei ole hyödyllisyydeltään samaa luokkaa kuin alkuperäinen suunniteltu projekti, on siitä kuitenkin hyötyä tässä insinööriyössä. Se sopii hyvin tämän insinööriyön teemaan ja antaa tyydyttävän esimerkin siitä, millaisissa tilanteissa kontteja voidaan hyödyntää.

Miten tässä insinööriyössä käsitellyjä asioita voidaan hyödyntää tulevaisuudessa? Mitä uusia kysymyksiä ja tutkimuksen aiheita voitaisiin tulevaisuudessa ratkoa? Kuten jo sanottua, Ilmatieteen laitos on jo toteuttamassa laskentaympäristöä AWS-pilveen säädatan mallintamista varten. Tällä hetkellä sinne vain kahdennetaan jo olemassa olevaa järjestelmää, mutta tulevaisuudessa sinne voidaan rakentaa myös täysin uusia ominaisuuksia. AWS-pilvipalvelut tarjoavat valtavan määrän erilaisia valmiita palveluita, joita on ehkä mahdollista hyödyntää. Näitä mahdollisuuksia olisi mahdollista tutkia tulevaisuudessa.

Voitaisiin myös tutkia sitä, että voisiko Ilmatieteen laitoksen säänmallinnus tapahtua kokonaan pilvessä. Tällä hetkellä säänmallinnus tapahtuu pääosin Ilmatieteen laitoksen omissa konesaleissa ja pilvessä oleva järjestelmä on tarkoitettu varajärjestelmäksi nykyisen rinnalle. Tulevaisuudessa pilvessä oleva järjestelmä kuitenkin laajenee ja on mahdollista, että AWS-pilvessä olevien palvelujen ansiosta Ilmatieteen laitoksen pilvijärjestelmästä tulee ensisijainen järjestelmä säädatan mallintamiseen. Useat yritykset ja organisaatiot ovat siirtyneet kokonaan käyttämään pilvipalveluita. Ilmatieteen laitoksella voitaisiin tutkia laajemmin mitä mahdollisuuksia pilveen siirtyminen tarjoaisi ja mitä se tarkoittaisi palvelun luotettavuuden kannalta, jos konesalista luovuttaisiin kokonaan.

## Lähteet

1. Ilmatieteen laitos - turvaa ja sujuvuutta yhteiskunnalle. 2024. Verkkoaineisto. Ilmatieteen laitos. <<https://www.ilmatieteenlaitos.fi/ilmatieteen-laitos>>. Päivitetty 14.5.2024. Luettu 5.12.2024.
2. What are Iaas, Paas and Saas? 2024. Verkkoaineisto. IBM. <<https://www.ibm.com/topics/iaas-paas-saas>>. Päivitetty 28.10.2024 Luettu 5.12.2024.
3. Wallenius, Niklas. 2022. Konttitekнологia – mitä kontit ovat ja mitä hyötyä niistä on? Verkkoaineisto. <<https://niklaswallenius.fi/konttitekнологia-mita-hyotyja/>>. Julkaistu 23.2.2022. Luettu 5.12.2024.
4. Amazon Elastic Container Service. 2024. Verkkoaineisto. Amazon. <[https://aws.amazon.com/ecs/?nc2=h\\_ql\\_prod\\_ct\\_ecs](https://aws.amazon.com/ecs/?nc2=h_ql_prod_ct_ecs)>. Päivitetty 4.12.2024. Luettu 5.12.2024.
5. Containers on AWS Overview: ECS | EKS | Fargate | ECR. 2020. Verkkoaineisto. TechWorld with Nana. <<https://www.youtube.com/watch?v=AYAh6YDXuho&t=454s>>. Julkaistu 29.11.2020. Katsoettu 11.12.2024.
6. Amazon Elastic Kubernetes Service. 2024. Verkkoaineisto. Amazon. <[https://aws.amazon.com/eks/?nc2=h\\_ql\\_prod\\_ct\\_eks](https://aws.amazon.com/eks/?nc2=h_ql_prod_ct_eks)>. Päivitetty 4.12.2024. Luettu 5.12.2024.
7. Red Hat OpenShift Service on AWS. 2024. Verkkoaineisto. Amazon. <[https://aws.amazon.com/rosa/?nc2=h\\_ql\\_prod\\_ct\\_rosa](https://aws.amazon.com/rosa/?nc2=h_ql_prod_ct_rosa)>. Päivitetty 4.12.2024. Luettu 5.12.2024.
8. What is ROSA (Red Hat OpenShift Service on AWS)? 2022. Verkkoaineisto. Containers from the Couch.

<<https://www.youtube.com/watch?v=6W-xDavWgYg&t=165s>>. Julkaistu 9.3.2022. Katsottu 11.12.2024.

9. AWS Batch. Verkkoaineisto. 2024. Amazon. <[https://aws.amazon.com/batch/?nc2=h\\_ql\\_prod\\_cp\\_ba](https://aws.amazon.com/batch/?nc2=h_ql_prod_cp_ba)>. Päivitetty 10.12.2024. Luettu 11.12.2024.
10. Carty, David. 2021. Amazon Lightsail. Verkkoaineisto. <<https://www.techtarget.com/searchaws/definition/Amazon-Lightsail>>. Julkaistu 10.6.2021. Luettu 11.12.2024.
11. AWS Lambda. Verkkoaineisto. David Carty. <<https://www.geeksforgeeks.org/introduction-to-aws-lambda/>>. Päivitetty 7.12.2024. Luettu 11.12.2024.