SAVONIA

OPINNÄYTETYÖ - **AMMATTIKORKEAKOULUTUTKINTO**
TEKNIIKAN JA LIIKENTEEN ALA

AWS-YMPÄRISTÖN VARMUUSKOPIOINTI

TEKIJÄ: Niilo Remes

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Niilo Remes	
Työn nimi AWS-ympäristön varmuuskopiointi	
Päiväys 11.8.2019	Sivumäärä/Liitteet 39 / 1
Ohjaaja(t) Pasi Liimatainen, Mikko Pääkkönen	
Toimeksiantaja/Yhteistyökumppani(t) Ropo Capital	

Tiivistelmä

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa AWS-pilvipalvelussa toimiva varmuuskopiointiratkaisu, joka turvaa pilvessä säilytetyn datan selviämisen katastrofitilanteessa. Ratkaisun tuli suojata tärkeimmät palvelut sekä laitteistorikoilta että käyttäjän virheiltä ja raportoida tekemisistään säännöllisesti. Tärkeitä ominaisuuksia näiden lisäksi olivat mahdollisuus saada uudet resurssit automaattisesti varmuuskopioinnin piiriin, varmistua varmuuskopioiden salauksesta sekä mahdollistaa eri ympäristöjen ja sitä kautta erilaiset vaatimukset varmuuskopioiden tiheyteen ja säilytykseen liittyen.

Opinnäytetyö käsitteli eri vaihtoehtoja vaadittujen ominaisuuksien toteutukseen, ja varmuuskopiointi toteutettiin testiympäristössä. Lisäksi työssä käsiteltiin suurpiirteisesti varmuuskopioiden hintoja, lähinnä verratessa ratkaisuja ja niiden kustannustehokkuuksia.

Lopputuloksena oli funktionaalinen varmuuskopiointiratkaisu, joka oli mahdollista ottaa sellaisenaan käyttöön tuotantoympäristössä.

Avainsanat
AWS, Pilvipalvelu, Varmuuskopiointi

SAVONIA UNIVERSITY OF APPLIED SCIENCES

THESIS
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Niilo Remes			
Title of Thesis Backup of AWS Environment			
Date	August 24, 2019	Pages/Appendices	39 / 1
Supervisor(s) Mr. Pasi Liimatainen, Senior Lecturer and Mr. Mikko Pääkkönen, Senior Lecturer			
Client Organisation /Partners Ropo Capital Ltd.			

Abstract

The objective of this thesis was to plan and implement a backup solution, which works within AWS Cloud Service. Aiming to prevent catastrophic data losses, the solution protects data stored inside the cloud from both user errors as well as hardware failures, and creates automatic reports from the backups regularly. Additional important features were an option to get new resources backed up automatically, ensuring that the backups are encrypted and enabling the use of different environments and the use of different requirements for backup frequency and storage time.

The work reviewed different options for implementing required features, and executed the backup in test environment. The cost of data backup was also be briefly evaluated by comparing different solutions and their cost-effectiveness.

The end result was a functional backup solution that could be deployed as such in a production environment.

Keywords
AWS, cloud computing, backup

TERMIT JA LYHENTEET

Aurora on Amazonin lisensoima relaatiotietokantatyyppi, joka eroaa monilta ominaisuuksiltaan muista RDS:stä.

Availability Zone on toisista Availability Zoneista maantieteellisesti erotettu osa Regionia. Yksi Region koostuu yleensä kolmesta Availability Zonesta, ja Availability Zone merkitäänkin usein lisäämällä regionin nimen perään a, b tai c.

AWS eli Amazon Web Services on Amazonin hallinnoima, maailman suurin pilvipalvelu, joka perustettiin 2006.

AWS Backup on Amazonin varmuuskopiointipalvelu, joka tukee suurinta osaa AWS:n tallennuspalveluista. AWS Backup varmuuskopioi resurssit määriteltyjen tägien tai resurssien ID:n perusteella, ja kryptaa inkrementaalisesti tehdyt varmuuskopiot.

CloudFormation on AWS:n palvelu, joka mahdollistaa uusien resurssien käyttöönoton sekunneissa valmiita, muokattavia pohjia käyttämällä.

CloudTrail on AWS:n seurantapalvelu, jonka avulla voidaan selvittää palvelun sisällä tapahtuneet käyttäjien toiminnot. CloudTrail tallentaa tietonsa samalla regionilla sijaitsevaan S3-koriin kryptattuna.

DynamoDB on Amazonin NoSQL-tietokantapalvelu.

EC2 eli Elastic Compute Cloud on palvelu, jolla tarjotaan laskentatehoa pilven kautta virtuaalikoneiden muodossa.

EBS eli Elastic Block Store on tallennustilaa tarjoava palvelu, jolla luodaan virtuaalisia kovalevyjä EC2-instanssien käyttöön.

EFS eli Elastic File System on tiedostopalvelimen tavoin toimiva täysin skaalautuva, lohkotallennustekniikkaa käyttävä tallennuspalvelu.

IAM eli AWS Identity and Access Management on palvelu, jolla hallitaan käyttöoikeuksia, käyttäjiä ja ryhmiä AWS:n sisällä.

KMS eli Key Management Servicen on IAM:n sisällä oleva palvelu, jonka avulla hallitaan salausavaimia, joilla kryptataan AWS:n sisään tallennettua dataa.

Pilvipalvelu mahdollistaa perinteisten tietokoneiden toimintojen, kuten laskentatehon, tietokantojen pitämisen ja tiedostojen säilyttämisen tai siirtämisen suorittamisen netin välityksellä, pilvipalvelun tarjoajan palvelinten avulla.

PITR eli Point in Time Restore on tietokantapalveluiden palautusominaisuus, joka mahdollistaa varmuuskopiosta palauttamisen haluttuun ajankohtaan jopa sekuntin tarkkuudella.

RDS eli Relational Database Service on Amazonin relaatiotietokantapalvelu, joka mahdollistaa tietokantainstanssien pitämisen pilvessä.

Region on maantieteellisesti ja loogisesti muista Regioneista erotettu osa AWS:ää. Palvelut ovat usein jaoteltu Regionien mukaan, niin että yhdellä Regionilla olevaa palvelua ei voida käyttää toisella Regionilla.

RTO eli Recovery Time Objective (toipumisaika) tarkoittaa suurinta hyväksyttyä aikaa, joka varmuuskopiolla voi mennä tietojen palauttamiseen aloitushetkestä ympäristön toimintavalmiuteen (Pietikäinen 2016).

RPO eli Recovery Point Objective (toipumispiste) tarkoittaa suurinta hyväksyttyä aikaa, jolta varmuuskopiosta palautettaessa voidaan menettää tiedot (Pietikäinen 2016).

S3 eli Simple Storage Service on Amazonin pilvitallennuspalvelu, johon käyttäjät voivat tallettaa objekteja.

S3 Bucket on jollakin Regionilla sijaitseva tallennuspaikka, jonka sisällä voi olla kansioita ja objekteja. Bucketin eli korin nimen täytyy olla globaalisti uniikki.

Snapshot on tietokannasta tai tallennustilasta tehty varmuuskopio, jota voidaan käsitellä itsenäisesti. Snapshotteja voidaan usein kopioida tai jakaa muille käyttäjille.

Tag eli tägi on resurssille määriteltyä metadataa eli dataa datasta. Tägit ovat muodossa {Avain:Arvo}, ja arvo voi olla tyhjä kirjainjono muttei Null. Samalla avaimella voi olla vain yksi tägi. (Using Tags, AWS)

SISÄLTÖ

1 JOHDANTO 7

2 VARMUUSKOPIOITAVAT PALVELUT JA VARMUUSKOPIOINTITAVAT 8

2.1 Vaatimukset varmuuskopiointitavoille 8

2.2 Varmistettavat palvelut 9

2.3 Varmuuskopiointitavat 9

2.3.1 S3 9

2.3.2 EBS 10

2.3.3 EFS 10

2.3.4 RDS / Aurora 11

2.3.5 DynamoDB 13

2.4 Valitut palvelut 15

3 RESURSSIEN AUTOMAATTINEN LISÄÄMINEN VARMUUSKOPIOINNIN PIIRIIN 16

3.1 AutoTag 16

3.1.1 Rakenne ja toiminnot 16

3.1.2 Asennus 18

3.1.3 Lisätägien lisääminen 19

3.1.4 Lopputulos 20

4 VARMUUSKOPIOINTI 22

4.1 S3 22

4.1.1 Cross-Region Replication 22

4.1.2 S3 Batch Operations 22

4.2 AWS Backup 24

4.2.1 Rakenne ja toiminnot 25

4.2.2 Varmuuskopiointisuunnitelman luominen 25

4.2.3 Hallinta 27

4.2.4 Palauttaminen 27

4.3 Aurora Snapshot Tool 28

4.3.1 Rakenne ja toiminnot 29

4.3.2 Asennus 29

4.3.3 Palauttaminen 31

5 RAPORTOINTI 33
5.1 AWS Backup 33
5.2 Aurora Snapshot Tool 35
6 YHTEENVETO JA POHDINTA 37
7 LÄHTEET 38

1 JOHDANTO

Pilvipalvelut ovat maailmanlaajuisesti jo 214 miljardin dollarin bisnes (Gartner Forecasts), jonka kasvulle ei näy loppua. Pilvipalveluiden käyttö tarjoaa yritykselle joustavuutta ja skaalautuvuutta, kasvanutta tehokkuutta sekä laitteiston käyttömaksujen että työntekijöiden ajan suhteen, sekä strategista arvoa tarjoamalla mahdollisuutta keskittyä olennaiseen missä tahansa (Benefits of Cloud Computing, IBM).

Ropo Capital on Suomen johtava laskunvälittäjäyritys 180 työntekijällään ja 8000 asiakkaallaan (Ropo tänään, Ropo Capital). Vuonna 2012 perustettu yritys on siirtämässä toimintojaan AWS:n piiriin suuremmassa mittakaavassa, ja varmuuskopiointiratkaisuksi haluttiin nykyaikainen, kustannustehokas ja hallittava ratkaisu toimintavarmuudesta tinkimättä.

Palveluna AWS muuttuu ja kehittyy jatkuvasti. Uusia ominaisuuksia lisätään sekä vanhoja poistetaan ja uudistetaan, ja uusien ominaisuuksien jatkuva analysointi toimivankin järjestelmän tilalle tai rinnalle on tärkeää myös kustannustehokkuuden näkökulmasta. Esimerkiksi mahdollisuus CloudFormationin käyttöön AWS Backupin asentamisessa tuli saataville tämän työn kirjoittamisen aikana, ja se otettiin osaksi työtä.

Tärkeitä ominaisuuksia varmuuskopiointiratkaisulle on uusien resurssien automaattinen lisääminen varmuuskopioinnin pariin, varmuuskopioiden kryptaus, raportointi sekä useiden eri säilytys- ja kopiointitiheyksien mahdollistaminen. Arvioidessa eri varmuuskopiointiratkaisuja huomiota kiinnitettiin ratkaisun käytettävyyteen ja hallittavuuteen, hintaan, varmuuteen ja ominaisuuksiin (Cross-Account, Cross-Region) sekä toipumisaikaan (RTO) ja toipumispisteeseen (RPO).

2 VARMUUSKOPIOITAVAT PALVELUT JA VARMUUSKOPIOINTITAVAT

Täydellistä varmuuskopiointitapaa ei ole olemassa. Mahdollisista tavoista tai niiden yhdistelmistä pitää valita sellainen, joka kattaa kaikki ehdottomasti vaaditut ominaisuudet ja mahdollisimman paljon halutuista ominaisuuksista. Tässä luvussa esitellään varmuuskopioitavat palvelut ja tavat varmistaa ne, sekä varmuuskopiointitavoilta vaaditut ja halutut ominaisuudet.

2.1 Vaatimukset varmuuskopiointitavoille

Yksittäisten ominaisuuksien lisäksi RTO ja RPO ovat tärkeitä lukuja, ja niiden minimointi kustannustehokkuus huomioon ottaen on seikka, johon pyritään kiinnittämään huomiota.

Tietojen säilyvyyden kannalta heikoin lenkki on usein ihminen. AWS:n sisäiset varmistukset ovat huippuluokkaa, ja onkin huomattavasti todennäköisempää, että datan menettäminen johtuu inhimillisestä virheestä kuin siitä, että palveluntarjoajan päässä tapahtuu katastrofi (Rock 2018). Näin ollen virheistä palautuminen on tärkeämpää kuin katastrofiin varautuminen.

Varmuuskopioiden salaus on elintärkeää. AWS kryptaa kaikki kryptatusta datasta tehdyt varmuuskopiot automaattisesti, mutta joissakin tapauksissa – esimerkiksi RDS Snapshot Toolia käytettäessä ja S3-koria käsiteltäessä – salaukseen ja sen kanssa toimimiseen pitää kiinnittää huomiota.

Varmuuskopiot pitää voida palauttaa tyhjän päälle. Jos alkuperäinen ympäristö menee syystä tai toisesta täysin käyttökelvottomaksi, niin uuden ympäristön luominen varmuuskopioinnista on pakollinen ominaisuus.

Uusien resurssien saaminen automaattisesti varmuuskopioinnin piiriin varmistaa, että kaikki halutut resurssit todella tulevat varmistetuksi. Inhimillisiä virheitä tapahtuu, ja jos automatisoinnilla saadaan työvaiheita vähennettyä, se helpottaa työntekijöiden työtä sekä luo toimintavarmuutta.

Varmuuskopioiden jatkuvalla tarkkailulla vältetään ikävät yllätykset. Kun epäonnistuneista varmuuskopioinneista tulee ilmoitus, niihin osataan reagoida niiden vaatimalla vakavuudella. Näiden lisäksi manuaalisesti läpikäytävä raportti esimerkiksi viikon välein olisi hyvä olla, jolloin voidaan varmistua myös hälytyksen toimivuudesta.

Cross-Account suojaa varmuuskopioita tietomurtojen ja tarkoituksellisten tuhoamisten varalta. Toiselle tunnukselle tehdyt varmuuskopiot ovat turvassa, mikäli pääasiallisen varmuuskopiointitunnuksen hallinta menetetään vihamieliselle taholle.

Cross-Region suojaa varmuuskopioita yhden regionin tuhoutumiselta tai käyttökatkokselta. Koska AWS tallentaa yhdellä regionilla olevan datan usealle Availability Zonelle ja palvelinkeskukselle, kokonaisen Regionin kaatuminen on äärimmäisen epätodennäköistä. Tähän mennessä yksikään palvelinkeskus ei ole mennyt kertaakaan täysin alas (Sverdlik 2018).

2.2 Varmistettavat palvelut

Kohdeyritys käyttää viittä palvelua, jotka täytyy varmuuskopioida: S3, EBS, EFS, RDS sekä DynamoDB. Näistä jokainen palvelu toimii itsenäisesti, ja ennen AWS Backupin markkinoille tuloa jokainen piti varmuuskopioida omalla tavallaan.

S3 eli Simple Storage Service on Amazonin tiedostotallennuspalvelu (Object Storage Service), johon käyttäjät voivat tallettaa tiedostoja. S3 on äärimmäisen varma, ja tiedostojen luvataan säilyvän jopa 99.999999999% varmuudella (”Eleven nines”). Palvelu tarjoaa edullisen ja käytännöllisen tallennustilan, ja ylivoimaisesti suurin osa kohdeyrityksen pilvessä olevasta datasta onkin S3:sen sisällä.

EBS:n eli Elastic Block Storagen avulla luodaan virtuaalisia kovalevyjä virtuaalikoneiden käyttöön. EBS-levy voi olla yhteydessä vain yhteen EC2-instanssiin samanaikaisesti.

EFS eli Elastic File System on täysin käytön mukaan skaalautuva tallennustila, jota voidaan käyttää useassa EC2-instanssissa samanaikaisesti. Samassa EC2-instanssissa voi siis olla sekä EBS että EFS-levyjä kiinni samanaikaisesti, ja instanssi voi ottaa yhteyden S3:seen internetin kautta. EFS käyttää lohkotallennustekniikkaa, eli levyt sisältävät vain puhdasta dataa.

RDS eli Relational Database Service on relaatiotietokantapalvelu, joka mahdollistaa tietokantainstanssien pitämisen pilvessä. RDS-kantoja on useita eri tyyppisiä, ja näiden varmuuskopiointitavat eroavat hieman toisistaan. Erityisesti Auroran varmistaminen on huomattavan erilaista muihin verrattuna.

DynamoDB on Amazonin NoSQL-tietokantapalvelu, joka on hyvin horisontaalisesti skaalautuva, sekä soveltuu usein ja samanaikaisesti tapahtuviin luku- ja kirjoitusoperaatioihin. DynamoDB:n erikoisuutena on hinnoittelu, joka tapahtuu käytetyn suoritustehon mukaan, ei tallennustilan.

2.3 Varmuuskopiointitavat

2.3.1 S3

S3:n varmuuskopiointi on kaksivaiheista. Tulevat tiedostot voidaan varmuuskopioida toisella regionilla olevaan koriin Cross-Region Replication (CRR) avulla, mutta tällöin vain uudet, replikoinnin aktivoimisen jälkeen luodut tiedostot kopioituvat kohdekoriin. Olemassa olevien tiedostojen varmuuskopiointiin on monta tapaa, mutta suurien tiedostomäärien kopioimiseen suositellaan käytettäväksi S3 Batch Operations -palvelua.

Muita mahdollisia tiedostojen kopiointikeinoja S3-korien välillä on ”s3 sync”-CLI-komento, Data Pipeline -palvelun käyttö tai olemassa olevien tiedostojen kopiointi korin sisällä, jolloin kopiot olisivat uusia objekteja, jotka näin ollen kopioituisivat CRR:n avulla kohdekoriin. Mikään näistä ei kuitenkaan ole vaihtoehto valtavien tiedostomäärien kanssa toimiessa – S3 sync ja Data Pipeline kuluttaa valtavasti suorittajakoneen muistia ja pullonkaulautuu tästä, kun taas CRR-replikointi tiedostojen kopioilla toisi turhan lisäkustannuksen, eikä nopeuttaisi toimintaa merkittävästi.

S3:n käyttäminen on varsin edullista, ja vaikka CRR:n käyttäminen triplaakin hinnan (datan säilytysmaksu tuplaantuu, ja siirtomaksu korien välillä regionilta toiselle on samaa luokkaa säilytysmaksun kanssa), niin suurienkin datamäärien tallettaminen pilvitallennustilaan ei kukkaroa isommin kevennä. Yhden gigatavun S3:n sisällä pitäminen maksaa parisen senttiä kuukaudessa ja CRR:n pyörittäminen eli datan siirtäminen korien välillä maksaa tasan 0.02$/GB/kk Frankfurtin AWS-regionilla (Amazon S3 Pricing, AWS). Datan säilytyshintaa voidaan alentaa käyttämällä Glacier- tai Infrequent Access (IA)-säilytysmuotoja, joista tietojen palauttaminen on hitaampaa ja kalliimpaa. Varmuuskopioiden tapauksessa dataa ei kuitenkaan kovinkaan usein tarvitse palauttaa, joten näiden käyttäminen alentaa hintaa huomattavasti. Batch Operationsien käytöstä menee kiinteä 0.25$ maksu per operaatio, ja 1$ per miljoona käsiteltyä tiedostoa.

2.3.2 EBS

EBS:n varmuuskopiointi onnistuu perinteiseen tyyliin snapshottien avulla tai AWS Backup -palvelulla. Snapshotit voidaan ottaa ja jakaa GUI:n avulla, mutta automatisointi – snapshottien ottaminen, kopiointi toiselle regionille ja vanhojen snapshottien poistaminen - tapahtuu skriptillä (Labrie, 2016) EC2-instanssin avulla. Toinen vaihtoehto on käyttää AWS Backup -palvelua, jolla varmuuskopiointi tapahtuu instantittomasti ja keskitetysti muiden palveluiden (EFS, RDS ja DynamoDB) kanssa.

Hinnoittelueroa perinteisellä tavalla ja AWS Backupilla tehtyjen varmuuskopioiden välillä ei ole missään palvelussa. Reaalimaailmassa kuitenkin AWS Backupilla tehtyjen varmuuskopioiden hinta on usein huomattavasti alhaisempi, koska palvelu ottaa ensimmäisen, ”täyden” varmuuskopion jälkeen inkrementaalisia varmuuskopioita (AWS Backup Pricing, AWS). Inkrementaalinen varmuuskopio kopioi vain viimeisimmän varmuuskopion jälkeen muuttuneet osa-alueet (Riikonen 2009), mikä vähentää kopioiden viemää tilaa huomattavasti, ja tätä kautta tuo käyttäjälle säästöjä.

2.3.3 EFS

Amazon tarjoaa EFS:n varmuuskopioon kahta palvelua, AWS Backuppia ja EFS-to-EFS:ää. CloudFormationilla luotava EFS-to-EFS käyttää CloudWatchin aktivoimaa Lambdaa luodakseen EC2-instanssin, jolla instanssi varmuuskopioi tiedostot lähteeksi määritellystä EFS:stä kohteeksi määriteltyyn EFS:ään käyttäen Fpsync-komentoa, luoden näin inkrementaalisen varmuuskopion (Architecture Overview, AWS). Toiminnon suoritettuaan EC2-instanssi sammutetaan ja

terminoidaan, joten siitä tulevat kulut jäävät minimaalisiksi. VPC peering -ominaisuuden avulla käsiteltävät EFS-levyt voivat olla eri regioneilla tai accounteilla.

Muita mahdollisia keinoja EFS:n varmuuskopiointiin on kopiointi ”S3 sync”-komennon avulla ja AWS Data Pipelinen käyttäminen. Data Pipeline ei kuitenkaan tue kaikkia regioneita, ja Amazon suosittelee Data Pipelinen EFS-varmuuskopioinnin ohjesivuilla käyttämään ennemmin EFS-to-EFS -ratkaisua, joka ”tukee AWS:n parhaita ratkaisuja turvallisuuden ja saatavuuden osalta” (Efs Backup using Data Pipeline, AWS). AWS CLI:stä löytyvällä ”S3 sync”-komennolla voidaan EFS-levyn sisältö helposti synkronoida S3-koriin, ja automatisointi on helppoa esimerkiksi Crontabin avulla. EFS-to-EFS, S3-koriin tai Data Pipelinellä tapahtuva varmuuskopiointi kuluttaa EFS Burst Creditsejä ja levyn toimintakapasiteettiä (Amazon EFS Performance, AWS), mutta AWS Backupilla tehty varmuuskopiointi ei (Using AWS Backup with Amazon EFS, AWS).

EFS:n varmuuskopiointi AWS Backupilla on huomattavan edullista. Kun EFS-to-EFS -tyylillä varmuuskopiointi maksaa normaalin EFS-levyn maksun verran eli 0.36$/GB/kk (Amazon EFS Pricing, AWS), AWS Backupilla varmuuskopioituna kopioiden viemä tila maksaa seitsemäsosan tästä, 0.05$/GB/kk (AWS Backup Pricing, AWS). Edullisinta varmuuskopiointi olisi kuitenkin S3:sen sisään, jolloin varmuuskopiointitila maksaisi noin puolet verrattuna AWS Backupilla tehtyyn varmuuskopioon, mutta S3:sen laskuttamat maksut jokaisesta objektipyynnöstä tasoittavat tätä eroa.

2.3.4 RDS / Aurora

Vaikka Aurora on osa Amazonin Relaatiotietokantapalvelua, niin se eroaa täysin varmuuskopioinniltaan ja monilta muiltakin ominaisuuksiltaan muista Amazonin tarjoamista tietokantamoottoreista (kutsun jatkossa näitä yhteisellä nimityksellä RDS). Siksi monista työkaluista, kuten esimerkiksi myöhemmin esiteltävistä Snapshot Tooleista on tehty kaksi eri versiota – Auroralle ja RDS:lle.

RDS:n varmuuskopiointiin Amazon tarjoaa oletuksena laajat mahdollisuudet. Oletuksena uutta tietokantaa luodessa valitaan Backup Retention Period, joka mahdollistaa samalla tietokannan ”palauttamisen” Point-in-time Restoren (PITR) avulla kyseisten päivien sisällä sekunnin tarkkuudella haluttuun kohtaan. Automaattiset varmuuskopiot voidaan luoda maksimissaan 35 päivän päähän, ja näiden lisäksi voidaan luoda Snapshotteja instansseista. Nämä Snapshotit voidaan kopioida toisille regioneille ja ne voidaan jakaa muiden käyttäjien kanssa. Lisäksi AWS Backup mahdollistaa RDS:n varmuuskopioinnin.

Aurora-tietokannan varmuuskopiointi eroaa hieman RDS:n varmuuskopioinnista. AWS Backup -mahdollisuutta ei ole Auroraa käytettäessä, jolloin keinoiksi jää käytännössä Snapshottien ja Replikoiden käyttö. Perinteisen PITR:n lisäksi Auroran erityisominaisuutena on PITR:ää muistuttava Backtrack-ominaisuus, jolla voidaan kelata aikaa korkeintaan 72h päähän taaksepäin. Backtrack mahdollistaa myös taaksepäin kelaamisen jälkeen tehtävän ”eteenpäin kelaamisen”, kunhan sallitulla aikaikkunalla pysytään. Toisin kuin PITR, Backtrack tapahtuu saman tietokannan päällä – tietokanta

on siis toimintakykyinen minuuteissa, kun uuden kannan luomisessa varmuuskopiosta saattaa kestää tunteja.

Sekä RDS että Aurora tukevat Replicoita, vain hieman eri tavalla. Replica tarkoittaa itsenäistä instanssia, johon kopioituu kaikki alkuperäiseen tietokantaan tapahtuvat muutokset. Molempien tukema Read Replica mahdollistaa read queryjen jakamisen tasan instanssien kesken, niin että Master-instanssiin ei kohdistu niin kovaa rasitusta. RDS:ssä ja Aurora MySQL:ssä Read Replicat voivat olla toisella regionilla, ja Masterin kaaduttua Read Replica voidaan nostaa manuaalisesti uudeksi Masteriksi, näin varmistaen toiminnan jatkuvuuden jopa kokonaisen Regionin kaatuessa. Auroran Cross-Region replikointia kutsutaan Global Databaseksi, ja siinä on useat toiminnot rajoitettu – esimerkikti Backtrack-toimintoa ei ole lainkaan. Sekä RDS että Aurora tukevat lisäksi usealle Availability Zonelle levittäytyvää Multi-AZ Deploymenttiä. Multi-AZ Deploymentillä luotu Replica nousee automaattisesti Masteriksi, jos alkuperäinen Master kaatuu.

Kaikista tietokannoista voidaan ottaa manuaalisia ja automaattisia snapshotteja, mutta vain RDS:n automaattisia snapshotteja voidaan käsitellään AWS-hallintaikkunassa automaattisten varmuuskopioiden (Automated Backups) alla. Kyseisestä ikkunasta tapahtuu PITR, minkä johdosta on ymmärrettävää että Auroran snapshotit käsitellään perinteisemmällä tavalla. Auroralla tapahtuva automaattinen varmuuskopiointi tapahtuu snapshottien muodossa, kun taas RDS:n tapauksessa snapshotit pitää ottaa käsin tai skriptillä. Auroran automaattista varmuuskopiointia ei saa pois päältä, mutta RDS:llä saa. Automaattisesti tapahtuvat snapshotit tulevat aina samalla regionille, ja snapshotit voidaan kopioida regionilta toiselle – silti esimerkiksi Auroran automaattiset varmuuskopiot tulevat kuutena kappaleena, kolmelle eri Availability Zonelle ja jokaiselle tuplana. Vain manuaalisesti luotuja snapshotteja voidaan jakaa muille käyttäjille, ja automaattisesti luodusta snapshotista saa manuaalisen kopioimalla sen. Snapshotit ovat inkrementaalisia (Backing up Aurora DB Cluster, AWS & Working with Backups, AWS RDS), ja automaattisesti luodut snapshotit poistuvat jos varmuuskopioitava kohde poistuu. Tällöin kuitenkin tarjotaan mahdollisuutta luoda Final Snapshot, joka mahdollistaa tietokannan palauttamisen vielä poiston jälkeenkin.

Snapshottien automaattiseen luomiseen, kopioimiseen toiselle regionille ja tunnukselle sekä vanhojen snapshottien poistamiseen on tehty työkaluja, kuten esimerkiksi Amazonin suosittelemat RDS Snapshot Tool (Snapshot Tool for Amazon RDS, Github) ja Aurora Snapshot Tool (Snapshot Tool for Amazon Aurora, Github.). CloudFormationilla asennettavat työkalut käyttävät CloudWatchia, Lambdaa sekä State Machinea toiminnoissaan.

Snapshottien säilytystila maksaa noin viidesosan tietokantojen säilytystilasta, ja inkrementaalisten snapshottien johdosta säilytyskustannukset pysyvät usein varsin kohtuullisina – vaikka snapshottien kopioiminen myös toiselle regionille tuplaisikin hinnan, tai lisäksi toiselle accountille triplaisi hinnan. Replikoiden pitäminen, Multi-AZ Deployment sekä Global Database moninkertaistaa hinnan instanssien määrällä, ja Global Databaseihin replikoidut operaatiot maksavat 0.22$ per miljoona operaatiota (Amazon Aurora Pricing, AWS).

2.3.5 DynamoDB

DynamoDB voidaan varmuuskopioida joko AWS Backupilla tai DynamoDB:n omilla keinoilla, PITR:llä ja snapshoteilla. Automaattinen, ”jatkuva” varmuuskopiointi (Continuous Backup) luo juoksevan varmuuskopion tietokannasta korkeintaan 35 päiväksi, mahdollistaen samalla PITR:n. Lisäksi DynamoDB Tablesta voidaan ottaa manuaalisia snapshotteja eli ”On-Demand”-varmuuskopioita, jotka säilyvät niin pitkään kuin halutaan. Toisin kuin RDS:n tapauksessa, DynamoDB ei poista automaattisia varmuuskopioita varmuuskopiointikohteen hävitessä (Point-in-Time Recovery, AWS).

Global Table mahdollistaa usealle regionille levittäytyvän Multi-Master -tietokannan luomisen, joka replikoi muutokset tietokantojen välillä ja mahdollistaa muutosten tekemisen jokaiseen kantaan. Global Table mahdollistaa toisten instassin automaattisen käytön mikäli yksi region menee alas, tarjoten näin virhevarmuutta.

Automaattinen PITR-varmuuskopiointitila on tuplasti kalliimpaa (0.2448$/GB/kk) kuin käsin On-Demand -varmuuskopioinnilla tehtyjen snapshottien viemä tila (0.1224$/GB/kk). Lisäksi AWS laskuttaa DynamoDB:n palauttamisesta kutakuinkin saman verran kuin itse varmuuskopiointitilasta kuukaudessa (0.1836$/GB), varmuuskopiointitavasta riippumatta (Amazon DynamoDB Pricing, AWS). AWS Backupilla tehty varmuuskopio maksaa saman verran kuin snapshot-varmuuskopiointi.

TAULUKKO 1. Eri varmuuskopiointitapojen vertailua

Varmuus-kopintitapa	**Plussat**	**Miinukset**	**Huomioitavaa**
AWS Backup	Yksinkertainen hallittavuus, lähes kaikki palvelut samalla käyttöliittymällä	Ei Cross-Region tai Cross-Account	Ei tue Aurora-tietokantojen varmuuskopiointia
S3 CRR	Yksinkertainen ja varma, mahdollista tehdä Cross-Account ja Cross-Region. Mahdollista laittaa varmuuskopiot eri Storage Typeen	Siirtomaksut toiselle regionille	Kopioi vain CRR:n jälkeen luodut tiedostot, olemassa olevat pitää kopioida Batch Operationeilla
S3 Sync / Data Pipeline	Mahdollistaa olemassa olevien tiedostojen kopioimisen ilman Batch Operationeita	Ei yhtä yksinkertainen kuin S3 CRR, vaatii suorittajakoneen ja pullonkaulaantuu herkästi	
EBS Snapshotit skriptillä EC2-instanssin avulla	Mahdollistaa Cross-Regionin	Vaatii EC2-koneen, käyttöönotto vaatii hieman vaivaa	Skriptit 3rd partyn tekemiä, AWS ei tarjoa virallisia
EFS-to-EFS	Mahdollistaa Cross-Regionin ja -Accountin	EFS-tallennustila on kallista	AWS:n virallisesti tukema
EFS:n kopiointi S3 Sync -komennolla	Edullinen tallennustila S3 sisällä	Automatisointi vaatii hieman vaivannäköä ja EC2-instanssin	
EFS:n kopiointi Data Pipelinellä		Ei tue kaikki Regioneita, AWS suosittelee EFS-to-EFS -ratkaisua Data Pipelinen sijaan	S3 Sync ja Data pipeline käyttää EFS:n kapasiteettiä, AWS Backup ei
RDS / Aurora Snapshot Tool	Mahdollistaa Cross-Account ja -Region	Työläähkö asennettava, vanhahtava toimintamalli	AWS:n suosittelema
RDS Automated Backups	Mahdollistaa PITR	Varmuuskopiot tuhoutuvat, jos kanta poistetaan	Oletuksena päällä
Aurora Automated Backups	Mahdollistaa PITR sekä Backtrack (pitää aktivoida erikseen)	Varmuuskopiot tuhoutuvat, jos kanta poistetaan	Ei voida ottaa pois päältä
RDS & Aurora Replica	Mahdollistaa instanssin/clusterin toiminnan jatkuvuuden regionin tai AZ:n kaatuessa	Hintava, erityisesti suurien instanssien kanssa	Ei varsinainen varmuuskopiointikeino, mutta hyvä lisä muiden keinojen rinnalle
DynamoDB Continuous Backup	Mahdollistaa PITR	Kallis (2x On-demandin hintainen), ei Cross-Account tai -Region	Pitää aktivoida erikseen, mahdollistaa 35d päähän palautuksen
DynamoDB On-demand Backup	Voidaan säilyttää niin pitkään kuin halutaan	Ei Cross-Account tai Cross-Region	Otetaan oletuksena kerran päivässä, voidaan ottaa lisäksi manuaalisesti

2.4 Valitut palvelut

AWS Backup mahdollistaa lähes kaikkien vaadittujen palveluiden varmuuskopioinnin ja hallinnan yhden hallintaikkunan kautta. Palvelu mahdollisti kaikki vaaditut ominaisuudet, joten puutteista huolimatta se valittiin pääasialliseksi varmuuskopiointikeinoksi. Tänä vuonna julkaistuun palveluun lisätään ominaisuuksia jatkuvasti, esimerkiksi tuki CloudFormation-asennukselle lisättiin opinnäytetyön kirjoittamisen aikana ja Cross-Region -tuki on tulossa myöhemmin tänä vuonna (AWS Backup, AWS News Blog).

Myös Aurora-tietokantojen varmuuskopiointiin tarvittiin ratkaisu, ja siihen paras palvelu oli Aurora Snapshot Tool, yhdistettynä mahdollisuuteen Multi-AZ Deploymenttiin ja/tai Global Databaseen. Muita vaihtoehtoja Aurora-tietokantojen varmuuskopiointiin on Amazonin omien varmuuskopiointien käyttäminen, mutta vaikka nämä ovat käytettävyydeltään huomattavasti Snapshot Toolia parempia, niin ne eivät kopioi tietoja Cross-Account tai Cross-Region. Lisäksi Amazonin omat varmuuskopiot ovat alttiita vahingossa tapahtuville poistamisille.

S3:n varmuuskopiointikeinoksi valittiin CRR, sekä olemassa olevien tiedostojen kopiointitavaksi Batch Operations. CRR:n ominaisuudet ovat ylivertaiset, ja asennus on yksinkertaista. Batch Operations on käytännössä ainoa toimiva tapa suurien (kymmeniä teratavuja dataa) tiedostomäärien kopioimiseksi toiseen S3-koriin.

3 RESURSSIEN AUTOMAATTINEN LISÄÄMINEN VARMUUSKOPIOINNIN PIIRIIN

AWS Backup varmuuskopioi resurssit määriteltyjen tägien (Tag) tai Resource ID:n perusteella. Yleisesti ottaen tägien käyttö on yksinkertaisempaa ja suurten resurssimäärien hallinta tägeillä helppoa, mutta myös yksittäiset resurssit on mahdollista laittaa varmuuskopioinnin piiriin laittamalla ne yksitellen Resource ID:llä.

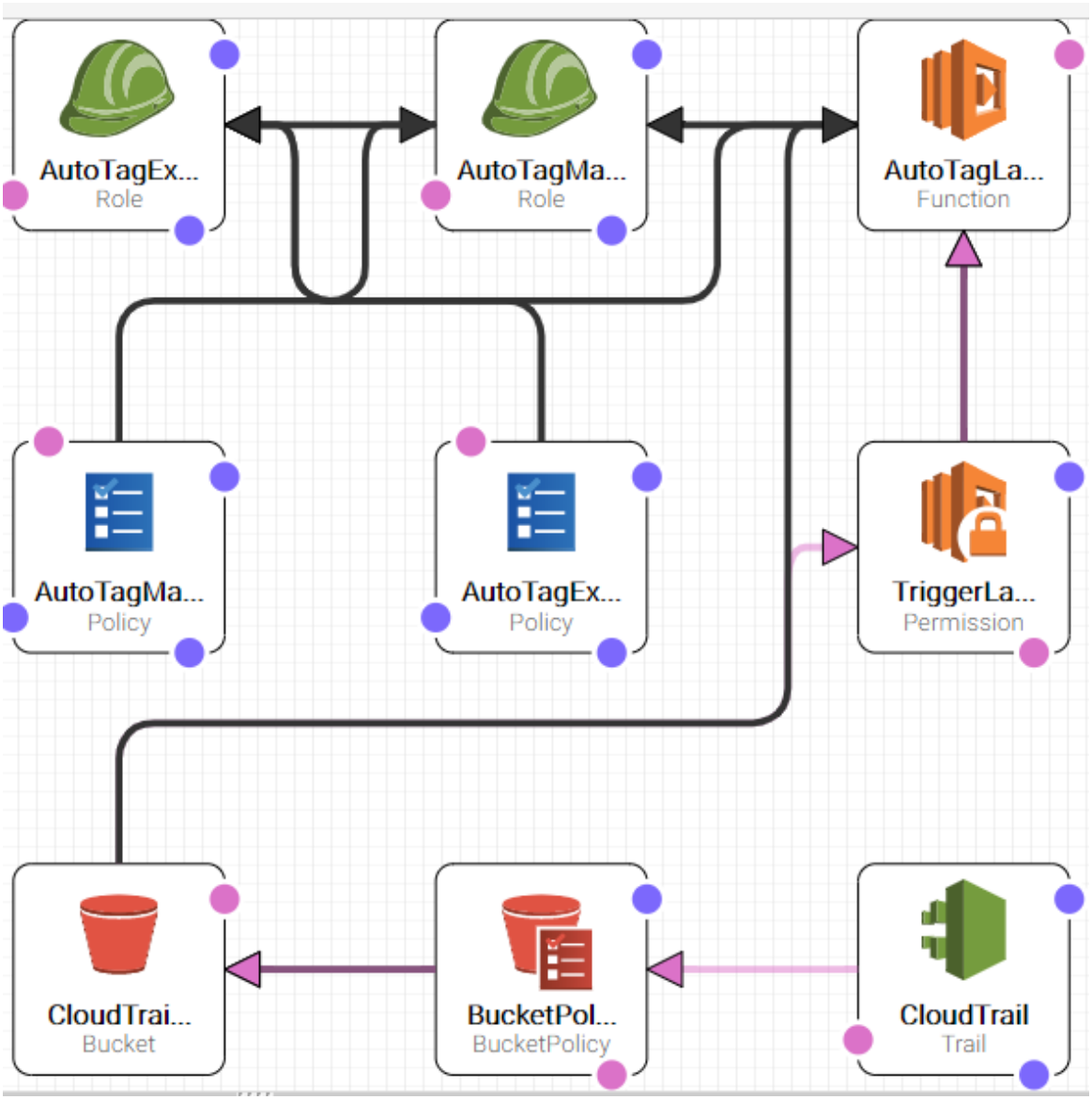
Jotta uudet resurssit saadaan varmuuskopioinnin piiriin, ne tarvitsevat tägin, ja oletuksena vain CloudFormationilla luodut resurssit saavat AWS:n sisällä tägin. Kun uudet resurssit halutaan automaattisesti varmuuskopioinnin piiriin, automaattinen tägäys pitää hoitaa jotenkin. Vielä vuonna 2016 Amazon suositteli (Martini 2016) luomaan itse Lambda-funktion, joka hakee CloudWatch Eventseistä uusien resurssien luomisen aiheuttavat API-kutsut ja tägää kyseiset resurssit halutulla tavalla. Nykyään hommaan on valmiita resursseja, joista yksi on GorillaStack-yrityksen kehittämä AutoTag (Autotag, Github). Autotagin valitsemista puoltaa avoin lähdekoodi ja käytön yksinkertaisuus.

3.1 AutoTag

AutoTag lisää uusiin resursseihin kolme tägiä: Creator, CreateTime sekä InvokedBy, joka kertoo, luotiinko uusi resurssi AWS-konsolin, CloudFormationin vai AutoScaling Groupin kautta. Palvelu mahdollistaa myös jo olemassa olevien resurssien tägäämisen, jos näiden resurssien luomisesta on CloudTrail-logit olemassa. AutoTag on mahdollista asentaa joko yksittäiselle tai usealle regionille, jolloin regionien välisen viestittelyn hoitaa Simple Notification Service (SNS).
Kaikki yleisimmät resurssit EFS:ää lukuunottamatta ovat tuettuja, mutta EFS-levyjä luodaan reaalimaailmassa kohtuullisen harvoin ja näin ollen tarve näiden automaattiselle tägäämiselle on pieni. Lisäksi Githubissa olevissa luomispohjissa (Template) oli käytetty vanhentuneita NodeJS-versioita, minkä takia CloudFormation ei sallinut niillä asentamista ilman pohjan päivittämistä. Pohjat päivitettiin ja päivityksistä kerrottiin GitHubissa, jotta seuraavat asentajat välttyisivät samalta päänvaivalta.

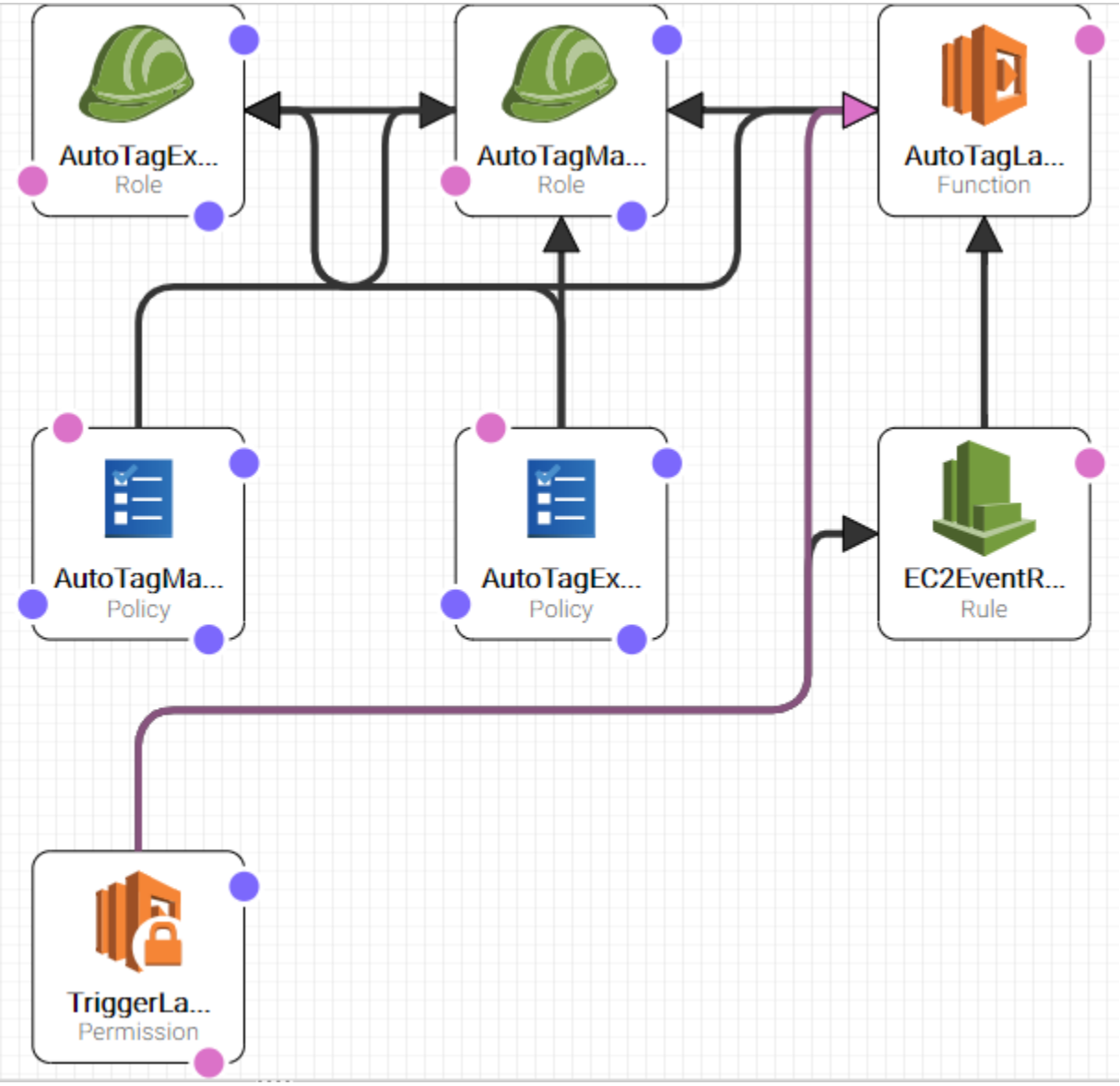
3.1.1 Rakenne ja toiminnot

AutoTag on mahdollista asentaa kahdella tavalla: CloudTrail-korilla ja S3 Put Object Triggerillä (Kuva 1), tai CloudWatch Events Rule Triggerillä (Kuva 2). CloudTrail-korin avulla luotava tyyli luo S3-korin, joka kerää kaikki CloudTrailiin tulevat logit, ja Lambda käynnistyy jokaisesta kyseiseen koriin tulevasta objektista. Näin ollen jokainen koriin tuleva logeista löytyvä tapahtuma prosessoidaan, mikä tuottaa huomattavaa ylimääräistä prosessointityötä. Tämä asennus on kuitenkin nopeampi asentaa ja toimii suoraan kaikilla regioneilla, ja pienellä lisätyöllä myös muilla ympäristössä olevilla tunnuksilla. Muille ympäristössä oleville tunnuksille pitää luoda IAM-rooli, joka sallii päätunnuksella olevan Lambda-funktion ajamisen. Tämä onnistuu helposti CloudFormationin avulla.



KUVA 1. CloudTrailia käyttävän AutoTagin rakenne CloudFormation Designerissä.

CloudWatch Events Rule Trigger filtteröi ympäristössä tapahtuvat tapahtumat, ja oikeanlaiset tapahtumat käynnistävät Lambda-funktion. Tämä on huomattavasti kustannustehokkaampi ja nopeampi ratkaisu, CloudTrail Logs luo huomattavan (5-10 min) viiveen resurssin luomisen ja tägien ilmestymisen välille, ja CloudWatch Eventsillä tehty ratkaisu vähentää Lambda-funktion käynnistämisiä noin 85% verrattuna CloudTrail Logeilla luotuun ratkaisuun. Tällä tavalla asennettu AutoTag vaatii kuitenkin ylimääräistä työtä, mikäli palvelua halutaan käyttää regionien välillä tai muilla tunnuksilla – käytännössä joka tunnukselle tämä täytyy luoda erikseen, ja regionien väliseen viestittelyyn pitää luoda SNS-topic joka vastaanottaa viestit sivuregionin CloudWatch Eventeistä ja lähettää viestit pääregionin Lambda-funktiolle.



KUVA 2. CloudWatch Eventtejä käyttävän AutoTagin rakenne CloudFormation Designerissä.

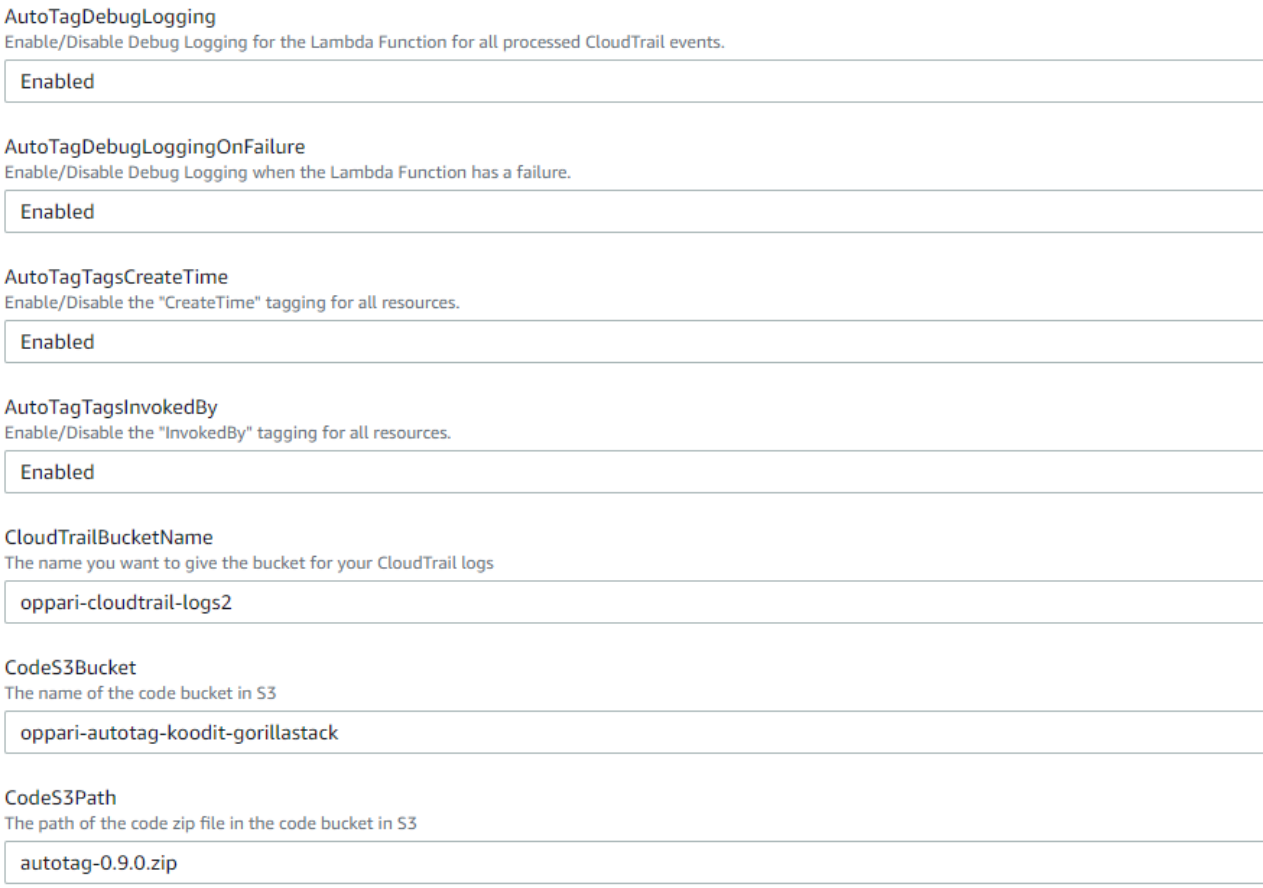
3.1.2 Asennus

Asennus tapahtuu CloudFormation Templaten avulla, ja palvelun käyttämät Lambda-koodit haetaan S3-korista. Oletuksena pohjien parametreissä olevat korit sisälsivät vanhentuneen version palvelusta, joten loin oman Zip-tiedoston, joka ladattiin omaan S3-koriin. Näin varmistettiin, että ajettu Lambda-koodi on varmasti ajantasaista. Tässä esitelty asennuskeino on CloudTrail Logeja käyttävä, mutta CloudWatch Eventtejä käyttävän AutoTagin asennus hoituu kutakuinkin samalla tavalla.

Zip-tiedoston luominen tapahtuu Linux-terminaalin avulla käyttäen hyväksi GCC (GNU Compiler Collection), NPM, Grunt ja AWSCLI -palveluita (Autotag, Github). Käytin tähän Amazonin EC2-instanssia. Tämän jälkeen Zip-tiedosto pitää siirtää S3-koriin, jotta CloudFormation osaa hakea sen Lambda-funktiota varten. Tähän voidaan käyttää joko AWS CLI:tä terminaalissa, tai tiedoston voi siirtää ensin kotikoneelle joko FTP-ohjelmalla (esim. Filezilla) tai SSH-terminaalia ja SCP:tä käyttämällä, minkä jälkeen tiedoston voi siirtää GUI:n kautta S3-koriin. CloudFormation-templatesta

korjattiin Lambda-funktiolle määritellyn NodeJS-versioksi 8.10, jolloin asennus ei tyssää CloudFormation-virheeseen.

CloudFormation-template löytyy GitHubista /cloud_formation/s3object_template -kansiosta, jossa on sekä päätunnukselle tarkoitettu _main-template että ympäristön muille tunnuksille tarkoitettu _role-template. Main-templaten NodeJS-version muokkaamisen jälkeen CloudFormation-konsolista luotiin uusi Stackin ja sille annettiin tarvittavat parametrit (Kuva 3).



KUVA 3. CloudTrail-tyylillä luotavan CloudFormationin vaatimat parametrit. CloudTrailBucketName-parametri luo uuden korin, CodeS3Bucket osoittaa koriin, jossa koodit on ja CodeS3Path on hakemistopolku korin sisällä, joka osoittaa Zip-tiedostoon. Tässä tapauksessa tiedosto on korin juuressa.

3.1.3 Lisätägien lisääminen

AutoTag luo oletuksena kolme tägiä, joilla se ilmoittaa luomisajankohdan, luojan ja luomistavan. Lambda-funktiota muokkaamalla palvelu saadaan luomaan näiden kolmen lisäksi muitakin tägejä. Funktiossa on jokaiselle AWS:n palvelulle oma Javascript-tiedostonsa jota palvelut käyttävät tägien luomiseen, mutta kaikki näistä tiedostoista hakevat tägit src/workers/autotag_default_worker.js -tiedostosta. Default_workeria muokkaamalla onkin mahdollista lisätä uusia tägejä (Kuva 4). Lambda-kyselyillä on mahdollista laittaa tägeihin muuttujia, mutta tässä esimerkissä luon vain kiinteän esimerkkitägin.

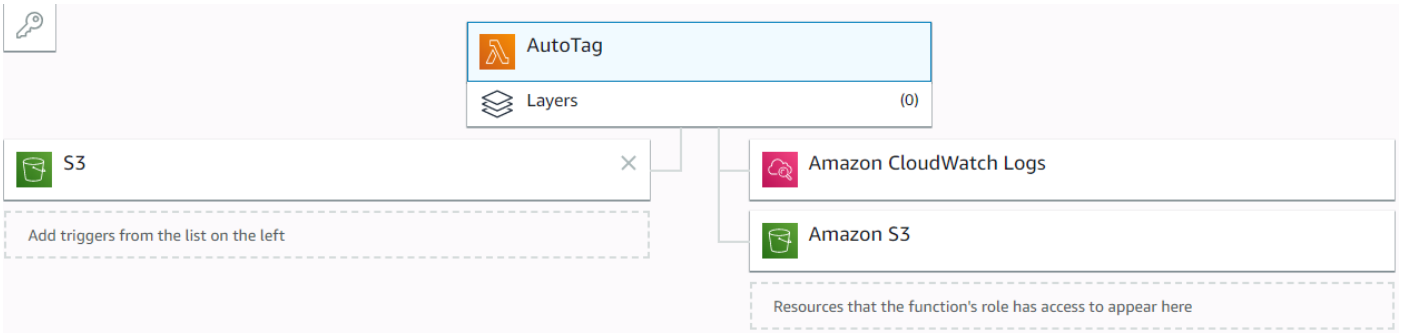
```
key: 'getAutotagTags',
value: function getAutotagTags() {
  return [{ Key: 'kovaKoodattuAvain', Value: 'kovaKoodattuArvo' }, this.getAutotagCreatorTag()].con
  _toConsumableArray(_autotag_settings2.default.AutoTags.CreateTime ? [this.getAutotagCreateTimeTag
  _toConsumableArray(this.getInvokedByTagValue() && _autotag_settings2.default.AutoTags.InvokedBy ?
  getAutotagInvokedByTag()] : []));
}
```

KUVA 4. Uusi tägi pitää lisätä autotag_default_worker.js-tiedoston getAutoTags-funktion sisään return-sulkeisiin, jolloin kaikki suluissa olevat tägit palautetaan funktiota kutsuttaessa. Arvot lisätään muodossa "*{Key: 'täginNimi', Value: 'täginArvo'}*". Kuvassa näkyy myös vakiona tulevat kolme muuttujatägiä.

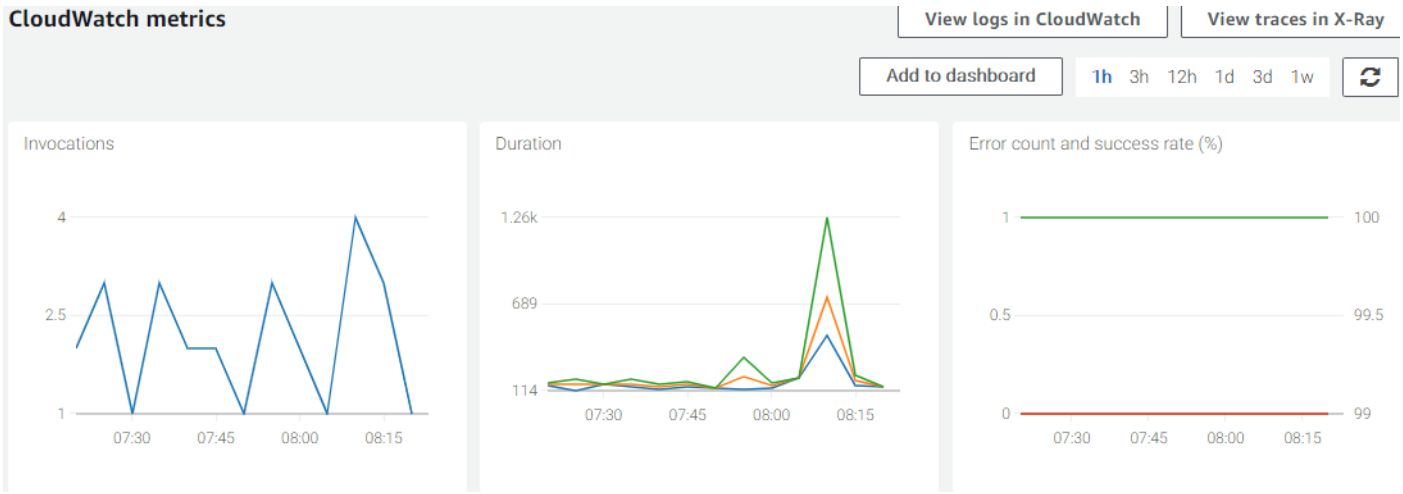
Lambda-funktioon tuodaan koodit Zip-pakettina – pienimuotoisia lambda-koodeja (alle 3MB) voidaan muokata selaimessa, mutta isompia, kuten AutoTag ei – ja koodin muokkaaminen onkin simppeliä. Monet purkuohjelmat mahdollistavat zip-paketin sisäisten tiedostojen muokkaamisen jopa ilman välissä tapahtuvaa purkua. Lopuksi uusi paketti ladataan Lambda-funktion käytettäväksi sen asetuksista, Function Code -kohdasta. Muutokset tulevat voimaan välittömästi tallennuksen jälkeen.

3.1.4 Lopputulos

Itse toiminta on loppukäyttäjälle täysin näkymätön, luotuja tägejä (Kuva 7) lukuunottamatta. Tägit tulevat CloudTrailista aiheutuvalla viiveellä (5-10 min) näkyville resursseihin, ja lokitiedot tulevat CloudWatchiin. Kyseisten lokitietojen määrän näkee myös CloudWatch Metricsistä, helpoiten näihin kahteen pääsee käsiksi kyseisen Lambda-funktion (Kuva 5) Monitoring-välilehdeltä (Kuva 6).



KUVA 5. CloudTrailia käyttävän AutoTagin Lambda -funktion rakenne Lambda Designerissä. Vasemmasta palkista näkyy, että funktio aktivoituu (trigger) S3-tapahtumasta, ja oikealta nähdään että funktiolla on oikeus kirjoittaa logeja sekä CloudWatchiin että S3:seen.



KUVA 6. Lambda-funktion Monitoring-välilehti näyttää lambda-funktion kutsumiskerrat ja mahdollisten virheiden määrän. Tarkkoja logeja pääsee lukemaan ”View logs in CloudWatch”-painikkeella, ja funktion analysoinnissa ja tarkemmassa tutkinnassa voidaan käyttää X-ray -palvelua.

Tags ▲	Value
AutoTag_CreateTime	2019-06-09T12:53:29Z
AutoTag_Creator	arn:aws:iam::[redacted]73:user/[redacted]
AutoTag_InvokedBy	signin.amazonaws.com

KUVA 7. AutoTagin vakiona luomat tägit. Creatorissa nähdään luodan AWS ID sekä näkyvä nimimerkki, ja InvokedBy kertoo, että luotiinko kyseinen resurssi CloudFormationilla, Auto Scalingilla vai käsin AWS Consolen kautta.

AutoTagin käyttäminen on lähes ilmaista. CloudWatch Logeilla luotuna Lambda-funktio tarkistaa uusien resurssien tilanteen keskimäärin kahden minuutin välein, ja jos funktio huomaa, että uusia resursseja ei ole luotu, koko hommasta laskutetaan laskentehoa 200ms ajalta minimimuistimäärällä 128MB, mikä tulisi kuukaudessa maksamaan noin 10 senttiä. Jos uusia tägejä pitää luoda, funktion suorittamiseen menee muutama sekunti. Laskentatehon lisäksi jokaisesta funktion ajopyynnöstä laskutetaan erikseen 0.20$ per miljoona ajopyyntöä, mikä tuo vajaan sentin lisäkustannuksen kuukaudessa. CloudTrail laskuttaa myös 0.10$/100 000 eventtiä, mutta jos kuukaudessa tulee keskimäärin 21 000 eventtiä, tästä tulee noin kahden sentin lisälaskutus.

4 VARMUUSKOPIOINTI

4.1 S3

S3 varmuuskopioidaan Cross-Region Replicationin (CRR) avulla. CRR kopioi koriin tulevat uudet tiedostot – joko kaikki, tai vain tietyllä prefixillä tai tägillä olevat - toisella regionilla olevaan kohdekoriin, jolloin tiedostot ovat turvassa yhden regionin tuhoutumiselta. Kohdekori voi olla myös toisella tunnuksella, jolloin ollaan turvassa tunnuksen joutumiselta vääriin käsiin.

CRR mahdollistaa vain uusien tiedostojen varmuuskopioimisen, ja jo olemassa olevat tiedostot pitää saada kopioitua kohdekoriin. Pienille tiedostomäärille mahdollisia keinoja olisi esimerkiksi tiedostojen suora kopiointi S3:sen sisällä tai tiedostojen kopioiminen kohdekorin sisällä, jolloin tiedostojen kopiot replikoituisivat kohdekoriin. Tämä ei kuitenkaan ole vaihtoehto, kun tiedostoja on miljoonia tai miljardeja, ja tälläisissä tapauksissa joudutaan käyttämään S3 Batch Operations -palvelua.

4.1.1 Cross-Region Replication

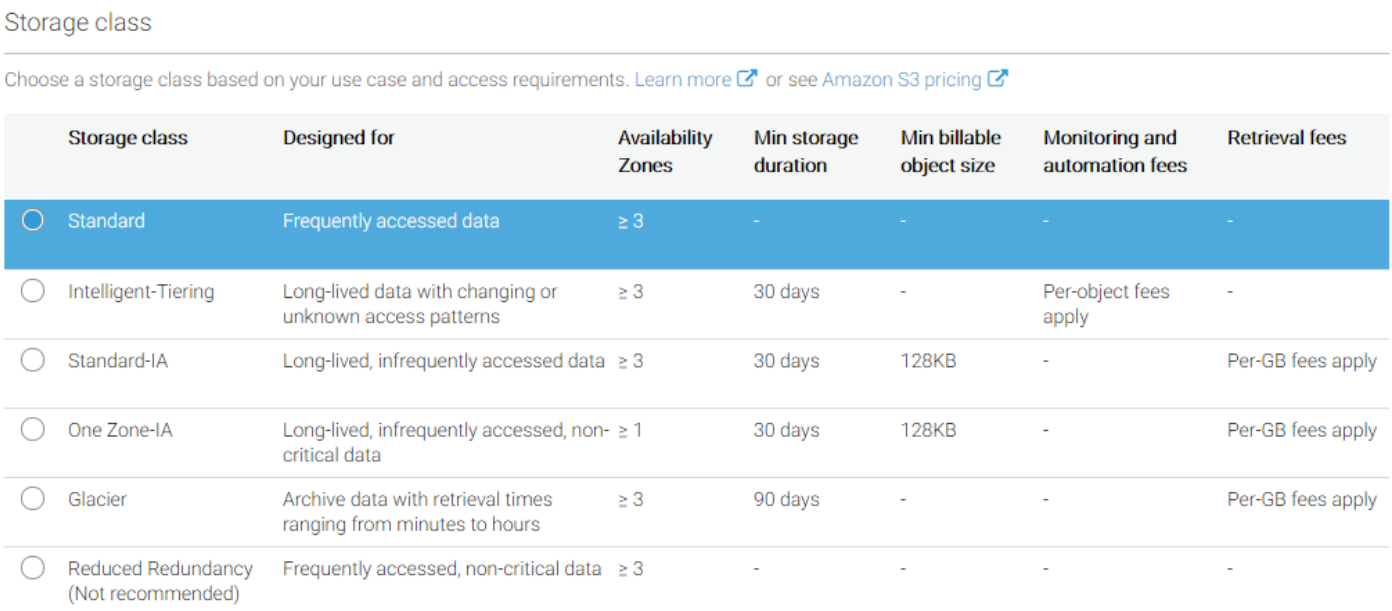
CRR tarvitsee neljä asiaa toimiakseen: lähdekorin (Source Bucket), eri regionilla olevan kohdekorin (Destination Bucket), oikeudet KMS-palvelun käyttämiin avaimiin sekä IAM-roolin, jolla on oikeuksia kopioida tiedostoja S3:sen sisällä. Näistä kolme viimeisintä voidaan luoda tai määritellä replikointia määritellessä, mutta versiointi pitää laittaa etukäteen päälle. Molemmilla koreilla täytyy myös versioinnin olla päällä.

Itse replikoinnin päälle pistäminen on yksinkertaista. Lähdekorin Management-ikkunasta löytyvästä Replication-kohdasta luodaan sääntö, josta muutamalla klikkauksella saadaan replikointi päälle. Välissä voidaan kuitenkin määritellä kopioitavien kohteiden rajaus esimerkiksi prefixillä, replikoinnin käyttämät KMS-avaimet ja IAM-rooli sekä lisäasetukset, joilla voidaan muokata kopioitujen objektien omistussuhdetta tai laittaa ne eri storage classiin kuin alkuperäiset objektit.

Toisen tunnuksen omistamalle kohdekorille tehty CRR vaatii oikeanlaiset oikeudet määriteltynä kohdekorille. Permissions-välilehden alta löytyvään Bucket Policyyn pitää lisätä komento, joka sallii lähdekorin omistajan replikoida objekteja kohdekoriin (CRR Examples, AWS Documentation)

4.1.2 S3 Batch Operations

Olemassa olevien tiedostojen kopiointiin käytetään S3 Batch Operations -palvelua. Kyseinen palvelu mahdollistaa massakopioinnin lisäksi esimerkiksi objektien tägäyksen tai Access Control Listin (ACL) muokkauksen massoittain.



Storage class

Choose a storage class based on your use case and access requirements. Learn more or see Amazon S3 pricing

Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and automation fees	Retrieval fees
Standard	Frequently accessed data	≥ 3	-	-	-	-
Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days	-	Per-object fees apply	-
Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days	128KB	-	Per-GB fees apply
One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
Glacier	Archive data with retrieval times ranging from minutes to hours	≥ 3	90 days	-	-	Per-GB fees apply
Reduced Redundancy (Not recommended)	Frequently accessed, non-critical data	≥ 3	-	-	-	-

KUVA 8. PUT Copy -operaation luomisvaiheessa voidaan kohdetiedostot laittaa eri Storage Classiin tarpeiden mukaan.

Batch Operations tarvitsee Inventory List Filen tai useamman, sekä manifest.json-tiedoston joka osoittaa Inventory List Fileihin. Manifest pitää luoda etukäteen, ja sen luominen kestää kauan, maksimissaan 48h Amazonin mukaan. Testiympäristön pienehkön korin manifesti luotiin noin 15h ajassa. Manifest-tiedosto kannattaa yksinkertaisuuden vuoksi luoda samaan koriin kuin missä kopioitavat tiedostot ovat.

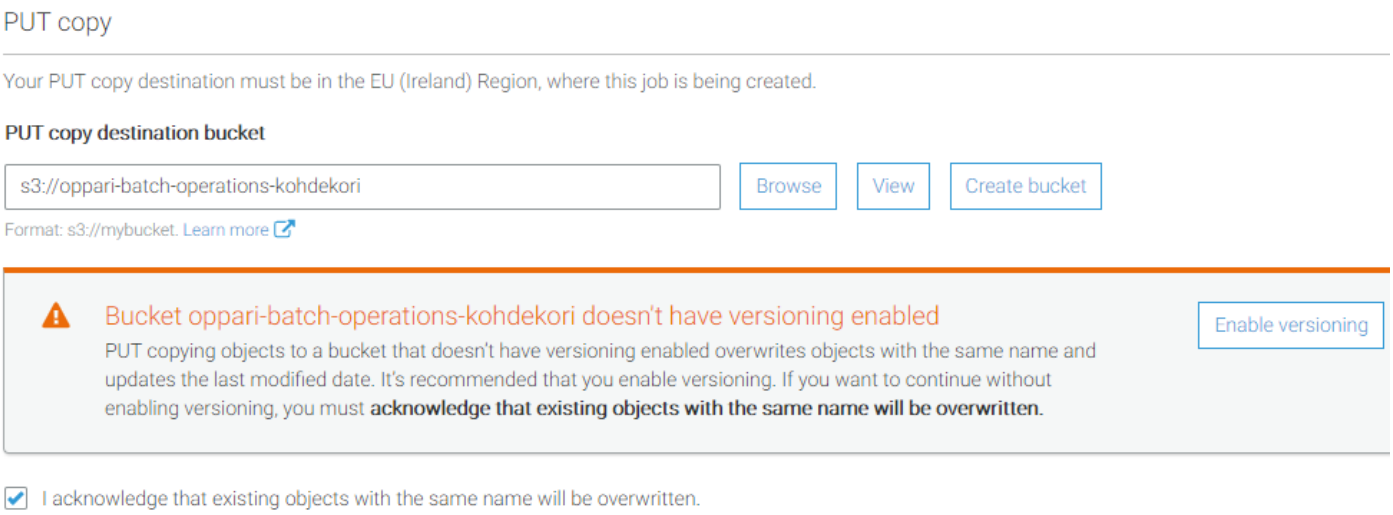
Ennen Manifestin luomista oikeudet täytyy laittaa kohdilleen Bucket Policyssa (Bucket Policy Examples, AWS Documentation), lähdekorin (Source Bucket, kori jossa kopioitavat tiedostot) Permissions-välilehden alta. Kun Bucket Policy on kohdillaan, Manifest luodaan lähdekorin Management-välilehden alta löytyvästä Inventory-kohdasta painamalla ”Add New”. Kohdekoriksi (Destination Bucket) annetaan lähdekori ja laitetaan homma tapahtumaan päivittäin. Lisäksi voidaan määritellä Inventoryn kryptaus, lisäkentät sekä tiedostomuoto. Yli miljoonalle objektille luotavalle inventorylle ei suositella käytettäväksi .CSV-tiedostomuotoa, vaan joko Apache ORC tai Parquet -tiedostomuotoa. Mikäli operaatiossa halutaan käyttää KMS-kryptausta, S3:selle pitää antaa oikeus käyttää kyseistä avainta (Amazon S3 Inventory, AWS Documentation). SSE-S3 -kryptausta käytettäessä lisäoikeuksia ei tarvita.

Batch Operations vaatii IAM-roolin, jolla on tarpeeksi oikeuksia. Uusi rooli luodaan IAM-palvelun avulla, S3 Batch Operations -palvelun kohdalta ja luoden uuden Policyn oikeilla oikeuksilla (Granting Permissions for Amazon S3 Batch Operations, AWS Documentation). Oikeuksien luomista yksinkertaistaa se, että mahdollisia koreja on neljän (Source, Destination, Manifest, Report) sijasta vain kaksi, sillä Manifestin ja Reportin voi ohjata Source-koriin.

Suorittaminen on yksinkertaista. Valitsemalla aiemmin luodusta manifestistä ”Create job from this manifest”-kohdan päästään luomaan Batch Operation tarvitsematta määritellä manifestin sijaintia

erikseen. Jobin luomisessa määriteltäviä asioita on kohde-region ja -kori, haluttu toiminto (tässä tapauksessa PUT Copy), kryptaus, haluttu Storage Class (Kuva 8) prioriteetti sekä aiemmin luotu IAM-rooli. Lopuksi valitaan luotu Job ja ajetaan se painamalla Confirm and Run. Luomisvaiheessa AWS varoittaa tiedon menetyksistä ja muista huomioon otettavista asioista (Kuva 9).

Kohderegionin määrittelyssä pitää muistaa se, että regionin pitää olla sama kuin missä kohteet ovat – siis jos halutaan kopioida esimerkiksi Frankfurtista Irelandiin, valitaan kohde-regioniksi Ireland. Prioriteetilla on merkitystä, jos useita Batch Jobeja ajetaan samanaikaisesti.



KUVA 9. Batch Operationsin luonti-ikkuna muistuttaa tärkeistä asioista, ja tiedon menetyksen mahdollisuuden kohdalla vaaditaan erillinen varmistus käyttäjältä.

4.2 AWS Backup

AWS Backup mahdollistaa yleisimpien AWS-palveluiden varmuuskopioinnin: EBS, EFS, RDS sekä DynamoDB. Automaattisesti varmuuskopioitavat resurssit merkitään varmuuskopiointisuunnitelmaan (Backup Plan) määritellyillä tägeillä (Tag), minkä lisäksi palvelu mahdollistaa yksittäisten, kertaluontoisten varmuuskopioiden tekemisen.

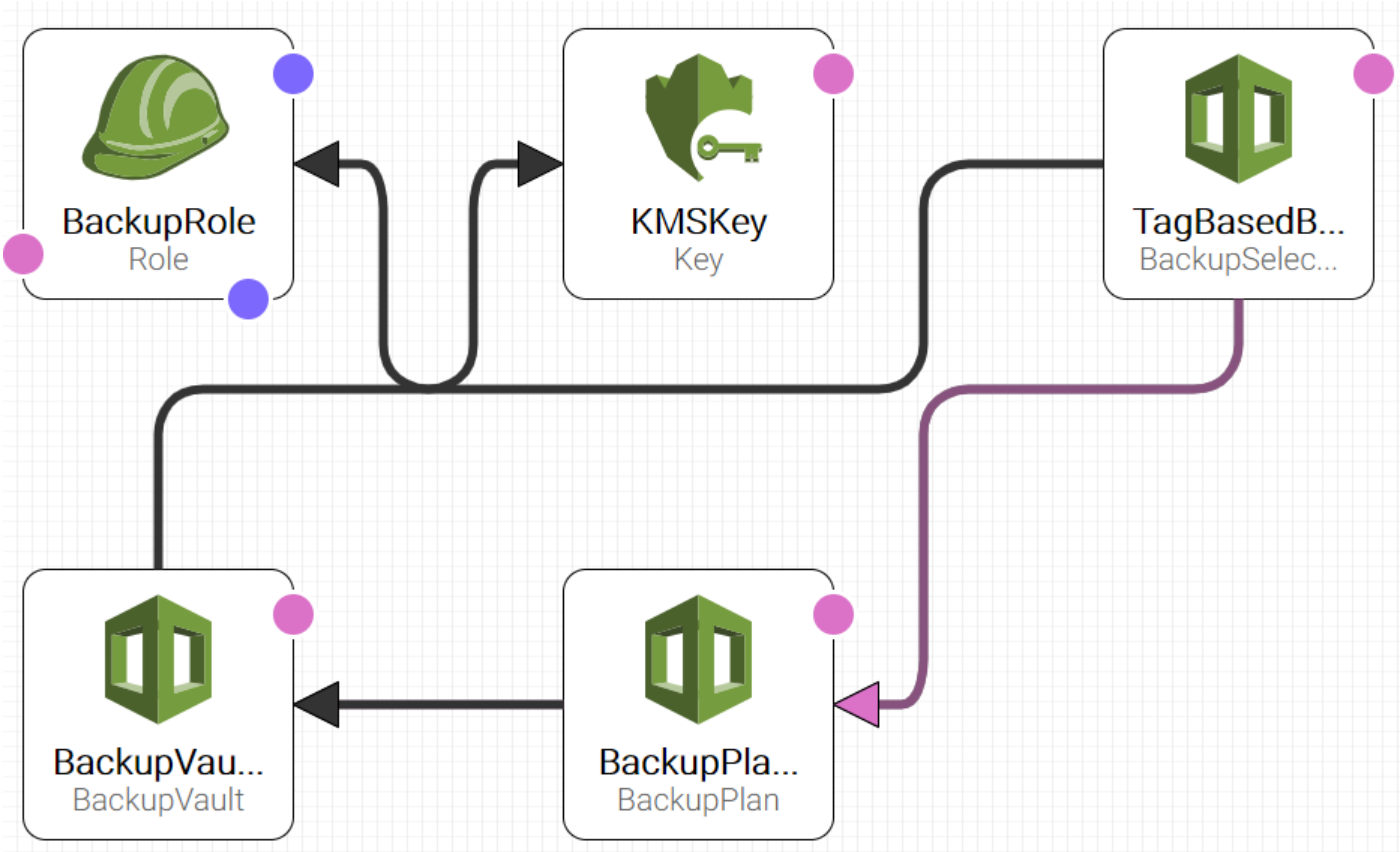
Tänä vuonna julkaistu AWS Backup on vielä varsin tuore palvelu, mistä johtuen sen ominaisuudet ovat vielä kohtuullisen kehittymättömiä - palvelu tuli käyttöön Eurooppalaisille regioneille (vain Ireland ja Frankfurt) vasta tämän vuoden huhtikuussa. Esimerkiksi Cross-Region -tuki on tulossa vasta myöhemmin tänä vuonna (AWS Backup, AWS News Blog) ja Aurora-tietokantojen varmuuskopiointi täytyy hoitaa perinteisin keinoin.

AWS Backup mahdollistaa erilaisten varmuuskopiointisääntöjen luomisen saman suunnitelman sisälle niin, että eri tiheyksillä tehdyt varmuuskopiot voidaan tallentaa eripituisiksi ajoiksi – esimerkiksi päivittäiset varmuuskopiot säilytetään 35 päivää, viikottaiset 12kk ja kuukausittaiset 5v. Näiden lisäksi palvelu mahdollistaa varmuuskopioiden siirtämisen halvempaan säilytysmuotoon lifecyclejen avulla, mutta nykyisellään ominaisuus on tarjolla vain EFS-varmuuskopioille.

4.2.1 Rakenne ja toiminnot

AWS Backup koostuu viidestä palasesta (Kuva 10). Varmuuskopiointisuunnitelman alle luodaan varmuuskopiointisääntö(jä) (Backup Rule), jolla määritellään, kuinka usein varmuuskopiot tehdään ja kuinka kauan niitä säilytetään. Suunnitelmaan pitää myös määritellä resurssit (Resource Assignment) tägeillä, jolloin kyseinen suunnitelma varmuuskopioi kaikki määritellyllä tavalla tägätyt resurssit tai Resource ID:llä yksittäisä resursseja. Varmuuskopiointisuunnitelman voi laittaa käyttämään haluamaansa IAM-roolia, mutta oletuksena Backup luo oman roolinsa suunnitelman luomisvaiheessa. Näiden lisäksi palvelu luo KMS-avaimen, jota käytetään varmuuskopioiden kryptaamiseen, mutta tätä ei GUI:lla luodessa näytetä lainkaan.

Varmuuskopiot (Recovery Point) talletetaan varmuuskopioholviin (Backup Vault). Eri varmuuskopiointisäännöt voivat ohjata varmuuskopiot eri holveihin, ja samaa holvia voidaan käyttää eri varmuuskopiointisuunnitelmissa. AWS Backup kopioi tagit varmuuskopioitavista resursseista varmuuskopioihin (AWS Backup will Automatically Copy Tags from Resource to Recovery Point, About AWS).

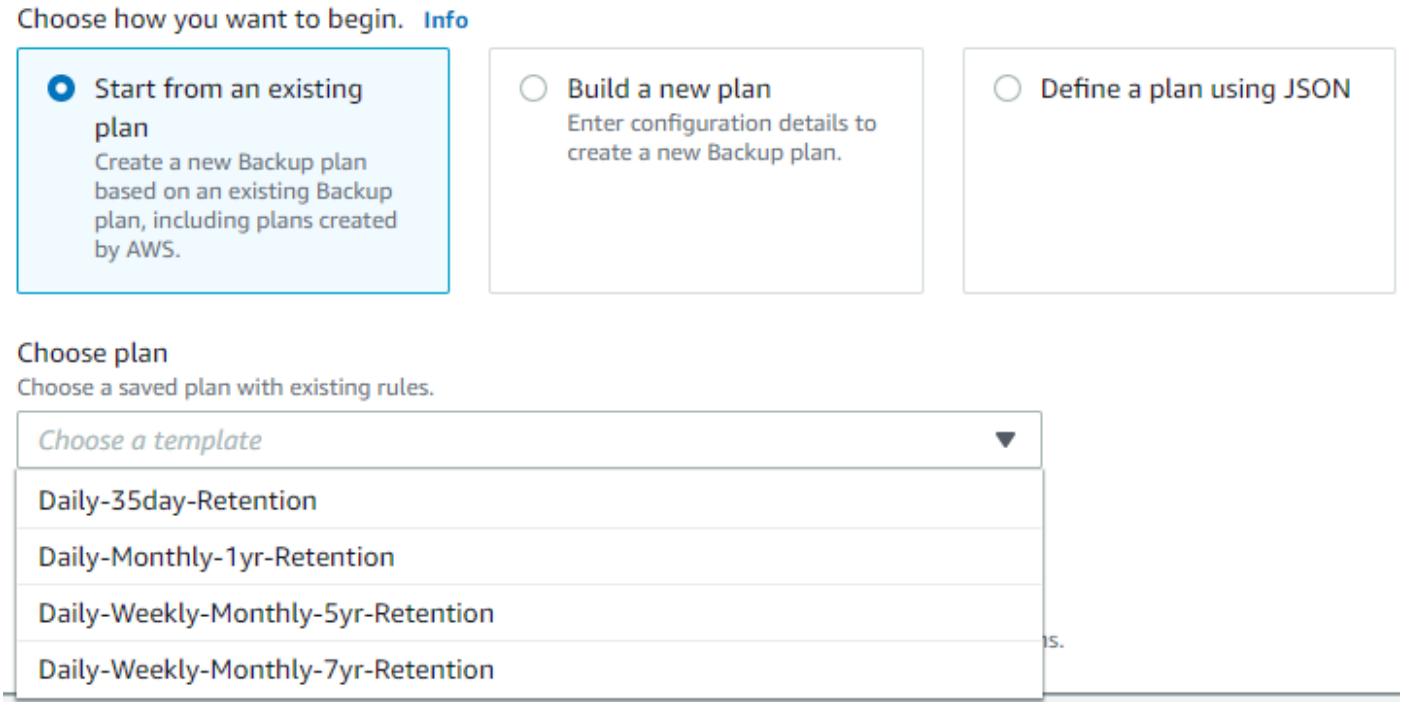


KUVA 10. AWS Backupin rakenne kuvattuna CloudFormation Designerissä. TagBasedB on Resource Assignment, joka kertoo varmuuskopiointisuunnitelmalle varmuuskopioitavat kohteet.

4.2.2 Varmuuskopiointisuunnitelman luominen

AWS Backup voidaan asentaa joko GUI:lla (Kuva 11) tai Cloudformationilla. GUI:lla asentaessa voidaan käyttää joko Amazonin luomia valmispohjia, luoda suunnitelma kokonaan tyhjästä tai luoda suunnitelma JSON:lla. Viimeisin vaihtoehto on lähes sama kuin CloudFormationilla käyttäminen, ja suurinpiirtein samaa JSON-koodia voidaan käyttää sekä CloudFormationissa tai JSON-pohjalla GUI:n avulla rakennettaessa.

Valmispohjilla demotaan palvelun mahdollisuutta käyttää useita varmuuskopiointisääntöjä eri tiheyksillä ja säilytysajoilla tehtyjä kopioita varten. Suunnitelmat ovatkin kumulatiivisia, seuraavissa suunnitelmissa on aina edellisen ominaisuudet mutta myös jotain lisää. Valittavana on päivittäinen varmuuskopio 35 päivän säilytysajalla, sama mutta lisäksi kuukausittainen varmuuskopio vuoden säilytysajalla, sekä lisäksi viikottainen varmuuskopio ja mahdollisuus valita kuukausittaisten varmuuskopioiden säilytysajaksi viisi tai seitsemän vuotta.



KUVA 11. Erilaiset mahdollisuudet luoda AWS Backup GUI:lla.

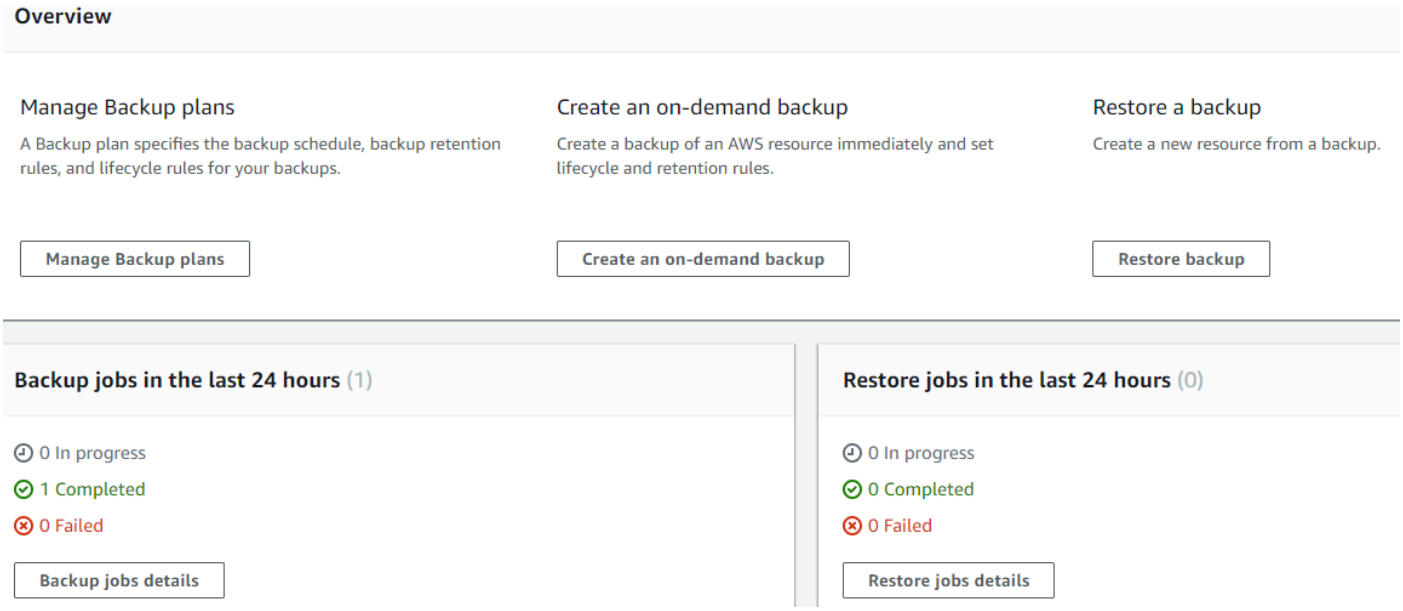
Varmuuskopiointisuunnitelman luomisvaiheessa määritellään yksi varmuuskopiointisääntö ja sitä myötä kopiointitiheys ja säilytysaika, kellonaika jolloin varmuukopiointi suoritetaan, säännön käyttämä varmuuskopiointiholvi sekä tägit, jotka suunnitelmaan tai palautuspisteisiin (Recovery Points) lisätään. Luomisen jälkeen määritellään varmuuskopioitavat resurssit joko tägeillä tai Resource ID:n avulla yksittäisiä resursseja.

CloudFormationin avulla varmuuskopiointisuunnitelman luomista ja hallintaa voidaan yksinkertaistaa. Amazonin tarjoamaa pohjaa (Using AWS CloudFormation Templates with AWS Backup, AWS) muokkaamalla loin CloudFormation-Templaten, joka ottaa parametreinä suunnitelman ja holvin nimen sekä Tägien avaimen ja arvon, joita vastaavat resurssit varmuuskopioidaan.

Yksinkertaistamisen nimissä varmuuskopiointitiheydeksi kovakoodattiin päivittäinen varmuuskopiointi ja säilytysajaksi 35 päivää.

4.2.3 Hallinta

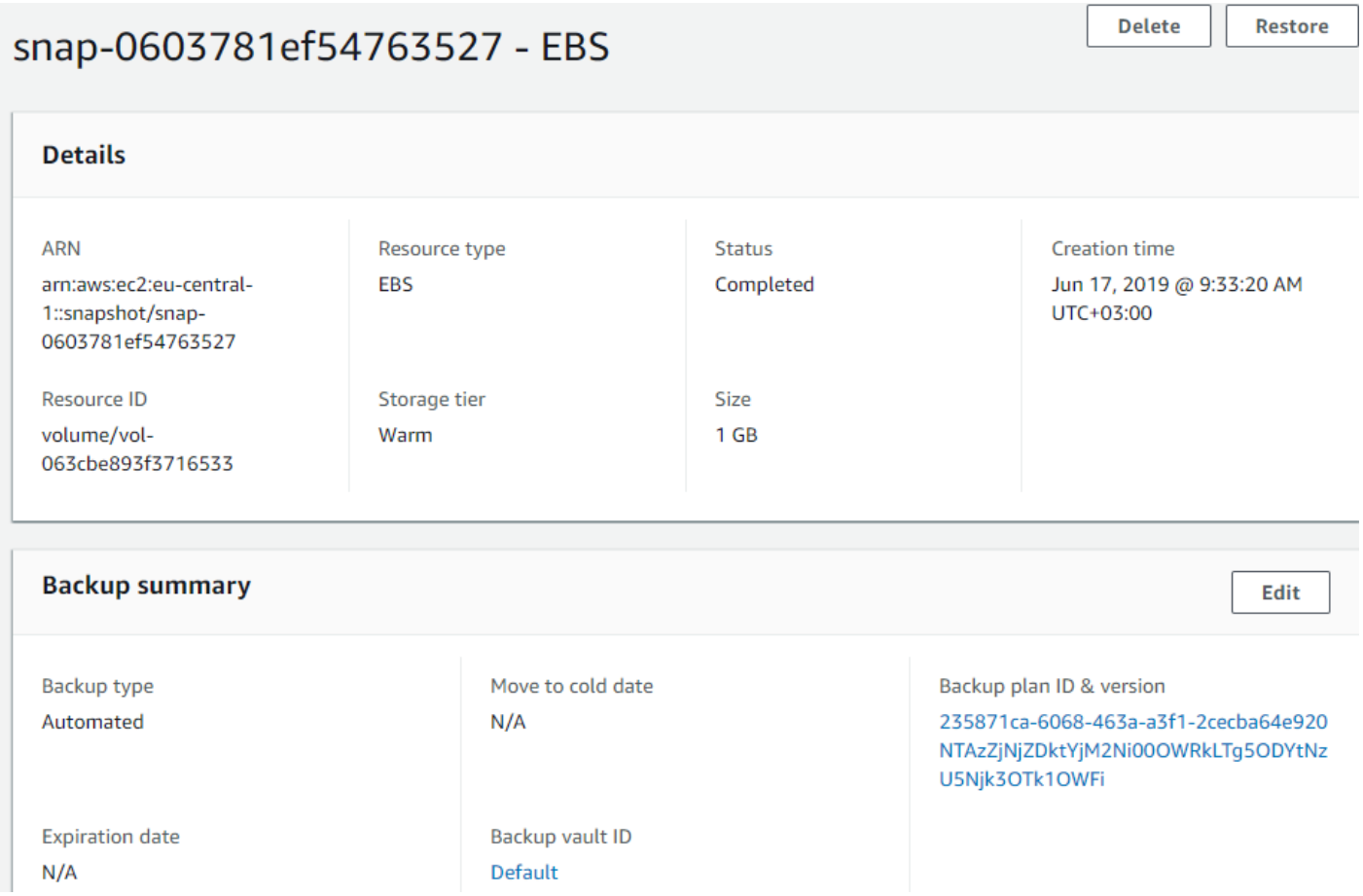
Varmuuskopiointisuunnitelmia voidaan luomisen jälkeen muokata vapaasti Dashboardilla (Kuva 12). Uusia sääntöjä voidaan luoda, vanhoja voidaan poistaa tai muokata ja resurssimäärityksiä voidaan lisätä tai poistaa. Lisäksi yksittäisiä palautuspisteitä voidaan halutessaan poistaa tai niille voidaan lisätä lifecyclepolicy, jolla varmuuskopion vanhentuminen määritellään.



KUVA 12. AWS Backupin ohjausnäkymä (Dashboard)

4.2.4 Palauttaminen

GUI:n avulla palauttaminen on yksinkertaista. Valitaan palautuspiste (Kuva 13), painetaan Restore ja luodaan uusi pohja, johon palautuspisteen data palautetaan. Palauttaminen tapahtuu yleensä uuteen resurssiin, jolloin luodaan joko uudet DNS-endpointit ja EFS:n tapauksessa Mount Targetit täytyy luoda uusiksi. Palautettaessa voidaan määritellä uuden tiedostojärjestelmän, instanssin tai tablen ominaisuudet, mutta DynamoDB:n tapauksessa palautus pitää tehdä samanlaiseen Tableen kuin mistä varmuuskopio on otettu. EFS voidaan palauttaa myös alkuperäiseen tiedostojärjestelmään.



KUVA 13. Palautuspisteestä näkee GUI:n avulla kattavat tiedot. Alalaidan Expiration Datesta nähdään, että tälle varmuuskopiolle ei ole luotu lifecyclejä, joten varmuuskopio säilyy manuaaliseen poistoon asti.

API:n avulla palauttaminen on mahdollista, mutta ainakin yksittäisille palautuksille API:n kautta tehtynä työmäärä olisi huomattavasti suurempi kuin GUI:n kautta hoidettuna. Voidakseen ajaa StartRestoreJob-komennon, tarvitaan palautuspisteen metadata, joka saadaan GetRecoveryPointRestoreMetadata-komennolla, ja tähän komentoon tarvittava RecoveryPointArn-parametri saadaan joko ListRecoveryPointsByBackupVault-komennolla tai ListRecoveryPointsByResource-komennolla. (AWS Backup API Actions, AWS Developer Guide.)

4.3 Aurora Snapshot Tool

AWS Backup ei tue Aurora-tietokantaa, joten Aurora Snapshot Tool hoitaa Auroran varmuuskopioinnin. Avoimen lähdekoodin palvelu kopioi tietokannoista otetut Snapshotit toiselle tunnukselle ja toiselle regionille, samalla poistaen vanhat Snapshotit tilaa viemästä ja raportoiden mahdollisista virheistä SNS:n avulla. Palvelu mahdollistaa kaikkien näiden toimintojen toteuttamisen myös yksittäin, niin että snapshotit tallennetaan joko vain cross-account, cross-region tai samalle regionille. Myös useiden päällekkäisten suunnitelmien samanaikainen päälläpitäminen ja ajaminen on mahdollista.

4.3.1 Rakenne ja toiminnot

Aurora Snapshot Tool käyttää toiminnassaan CloudWatch Eventsejä, Step Functionsia ja Lambdaa. CloudWatch Events aktivoi Amazon Step Functionsin, joka aktivoi Lambdan ja Lambda hoitaa itse työskentelyn. Step Functionsin rooli on varmistaa, että mahdollisen virheen sattuessa Lambda yrittää tehdä toimintonsa uudestaan.

Snapshot Toolin toiminnassa on neljä vaihetta: Snapshotin ottaminen, Snapshotin jakaminen, Snapshotin kopioiminen toiselle accountille ja Snapshotin kopioiminen toiselle regionille. Jos jokin näistä toiminnoista ei onnistu, CloudWatch Events lähettää SNS-topickiin huomautuksen, ja SNS voi lähettää sen eteenpäin esimerkiksi sähköpostiin, tekstiviestillä puhelimeen, HTTP-endpointtiin tai push notificationilla puhelimeen.

Kryptaus on kaksivaiheinen, ja tähän pitää käyttää kahta eri avainta. Ensin Snapshot Tool kopioi snapshotit toiselle tunnukselle, ja kryptaus puretaan alkuperäisellä avaimella. Tämän jälkeen snapshot kopioidaan toiselle regionille ja kryptataan uudella, kohderegionille kuuluvalla avaimella. Amazonin avaintenhallintapalvelu (Key Management Service, KMS) on region-kohtainen, joten samaa avainta ei voida käyttää molemmissa kopioinneissa.

Aurora Snapshot Tool määrittelee varmuuskopioitavat resurssit nimen perusteella, esimerkiksi ADB -parametrin antamalla palvelu kopioi olemassaolevat ADB1, ADB2 jne. resurssit sekä myöhemmin luotavat ADB3 ja ADB10. Snapshot Tool voidaan myös laittaa varmuuskopioimaan kaikki tietokannat.

4.3.2 Asennus

Asennus tapahtuu kahdella CloudFormation-templatella, joista toinen tulee lähdetunnukselle (source account) ja toinen kohdetunnukselle (destination account), **kuitenkin aina samalle regionille**. Mikäli Snapshot Toolia käytetään vain cross-region -kopiointiin ja cross-account -ominaisuutta ei hyödynnetä, asennetaan molemmat templatet saman tunnuksen alle.

CloudFormation-templatet saa ladattua Githubista (Snapshot Tool for Amazon Aurora, Github), ja oletusparametreillä palvelu kopioi kaikki snapshot-clusterit toiselle tunnukselle ja regionille kerran päivässä, poistaen yli viikon vanhat snapshotit. Ainoat pakolliset lisättävät parametrit ovat lähdetunnukselle asennettavan pohjaan asetettava kohdetunnuksen ID sekä kohdetunnukselle asennettavan pohjaan laitettava kohde-region, jonne snapshotit kopioidaan. Kohde-region voi olla sama kuin alkuperäinen region, mikäli toimintoa halutaan käyttää vain cross-account -toimintoon.

Lähdetunnuksen pohjassa tärkeimpiä määriteltäviä asioita on jakamiseen liittyvät ShareSnapshots- ja DestinationAccount-parametrit. Mikäli cross-account -toimintoa käytetään, ShareSnapshots-parametri määritellään TRUE:ksi ja DestinationAccount-parametriin täytetään kohdetunnuksen My Account -osiosta löytyvä ID. Muita oleellisia määrityksiä on varmuuskopiointitiheyden määrittely BackupInterval- ja BackupSchedule-parametreillä, varmuuskopioitavien kohteiden määrittely

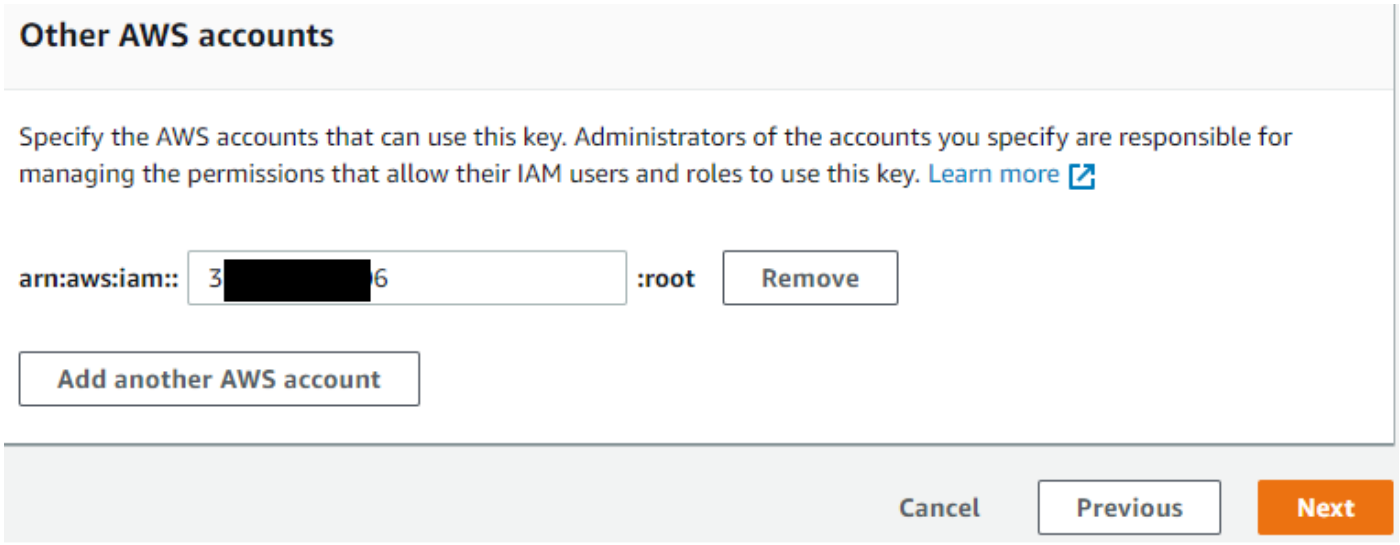
ClusterNamePattern-parametrillä, virheistä raportointi SNS-topickiin sekä vanhojen varmuuskopioiden poistamisen määrittelevät DeleteOldSnapshots- ja RetentionDays-parametrit.

Kohdetunnuksen pohjassa tärkeimpiä parametrejä (Kuva 14) ovat kohde-regionin ja CrossAccountCopy-asetuksen lisäksi kryptaukseen liittyvät KmsKeySource ja KmsKeyDestination, joilla määritellään kryptauksen käyttämät avaimet. Avainten ARN (Amazon Resource Name)-koodit pitää hakea KMS-palvelusta. Lisäksi vanhojen snapshottien mahdollinen poistaminen ja SNS-topic pitää määritellä myös tässä.

CodeBucket	DEFAULT_BUCKET
CrossAccountCopy	FALSE
DeleteOldSnapshots	TRUE
DestinationRegion	eu-west-1
KmsKeyDestination	arn:aws:kms:eu-west-1:7[redacted]3:key/cf0e[redacted]e4d0
KmsKeySource	arn:aws:kms:eu-central-1:7[redacted]3:key/851[redacted]f52b7
LogLevel	ERROR
RetentionDays	7
SNSTopic	-
SnapshotPattern	ALL_SNAPSHOTS
SourceRegionOverride	NO

KUVA 14. Samalle tunnukselle tehtävän varmuuskopioinnin kohdepohjan parametrit. Pohja on asennettu eu-central-1 -regionille, ja se kopioi snapshotit eu-west-1 -regionille. SNSTopiciksi voidaan määritellä jo olemassa oleva, mutta tässä tapauksessa CloudFormation luo uuden topicin johon virheilmoitukset ohjataan.

Mikäli tunnukselta toiselle jaettu snapshot on kryptattu, käytettävä KMS-avain täytyy olla jaettu. Helpoimmillaan jakaminen onnistuu uuden avaimen luomisvaiheessa GUI:n kautta (Kuva 15), mutta jo olemassaoleviin avaimiin täytyy jakaminen lisätä avaimen Key Policyyn JSON-koodikielellä. Huomiotavaa on, että oletusavaimen (Default RDS KMS key) tai sillä kryptatun snapshotin jakaminen ei ole mahdollista!

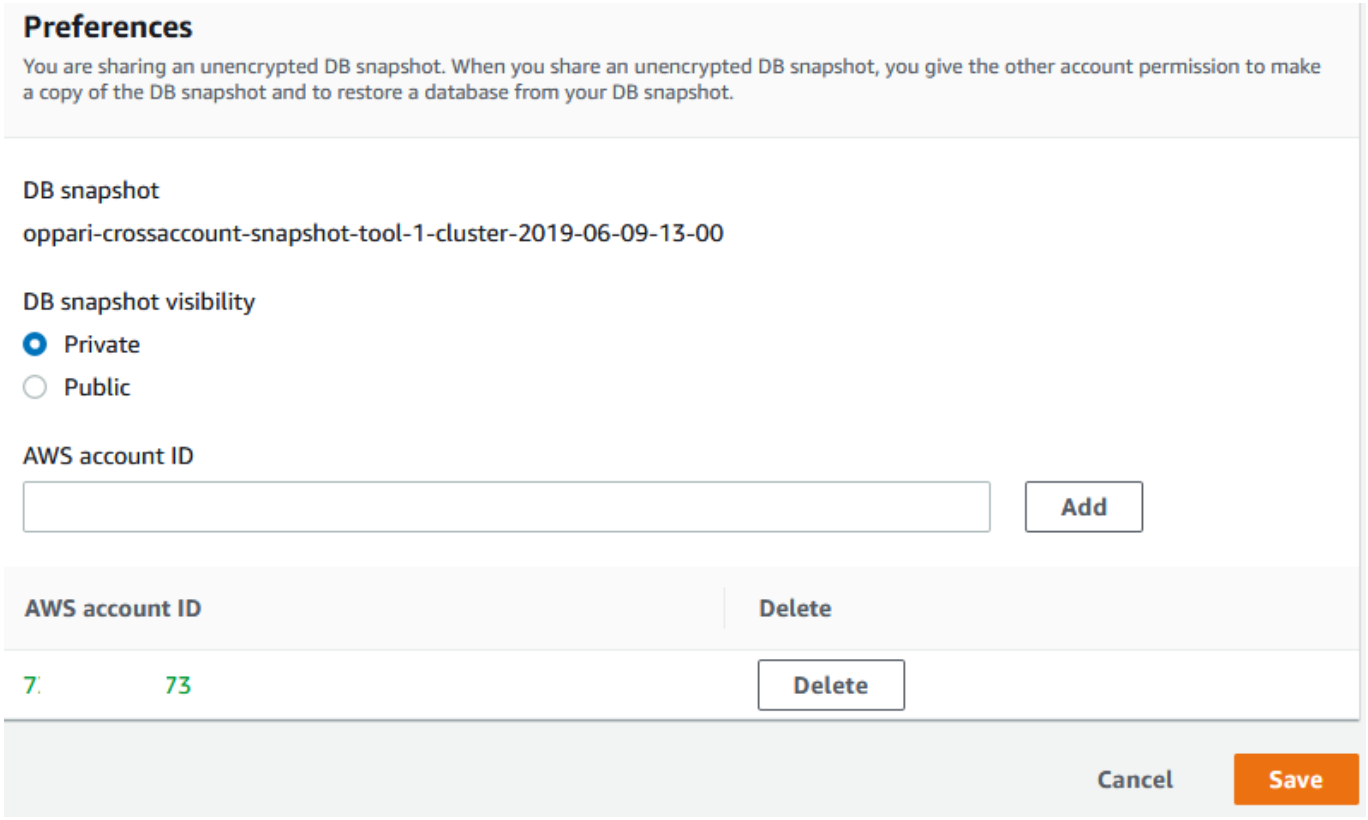


KUVA 15. Avaimen jakaminen muille tunnuksille uuden avaimen luomisen yhteydessä AWS Account ID:n avulla.

4.3.3 Palauttaminen

Snapshotista palauttaminen on yksinkertaista. Luodaan uusi instanssi snapshotilla ja määritellään instanssin asetukset täysin samoin kuin uutta tietokantaa luodessa. Snapshotista luodun instanssin tyyppi voi siis olla eri kuin alkuperäisen instanssin, esimerkiksi Provisioned-tyyppisestä r2.small-kannasta voidaan luoda joko Serverless tai sitten r5.xlarge global database.

Snapshot voidaan kopioida regionilta toiselle GUI:n avulla, Snapshotin asetuksista Actions-valikon alta valitsemalla Copy Snapshot ja valitsemalla kohde-region. Saadakseen snapshotin toiselle tunnukselle, snapshot pitää jakaa toisen käyttäjän kanssa ja vastaanottajalla täytyy olla oikeus käyttää snapshotin kryptaamiseen käytettyä KMS-avainta. Jaetusta snapshotista voidaan suoraan palauttaa tietokanta, sitä ei siis tarvitse ensin kopioida ”omaksi” snapshotiksi. Aurora Snapshot Tool luo suoraan manuaalisen snapshotin jonka voi jakaa suoraan, mutta mikäli automaattisen snapshotin haluaa jakaa (Kuva 16), siitä täytyy tehdä manuaalinen. Tämä onnistuu yksinkertaisesti kopioimalla snapshot.



KUVA 16. Snapshotin jakaminen toiselle käyttäjälle. Kryptaamattoman Snapshotin voi myös tehdä julkiseksi.

5 RAPORTOINTI

5.1 AWS Backup

AWS Backup on todella tuore palvelu, ja se ei itsessään sisällä vielä minkäänlaista raportointitoimintoa. Jonkinlainen ratkaisu varmuuskopioinnin seurantaan piti kehittää, jotta ei oltaisi täysin manuaalisen tarkkailun varassa. Ratkaisuksi valikoituikin Cloudwatch Metricsin ja Alarmsin hyödyntäminen niin, että tarkkaillaan reaaliajassa Backup Vaulteissa olevien Recovery Pointtien määrää, ja luodaan alarmi joka hälyttää (Kuva 19), jos Recovery Pointtien määrä laskee alle tietyn raja-arvon. Tällöin uusia palautuspisteitä ei ole tullut samaan tahtiin, kun vanhat ovat vanhentuneet, eli joko palautettavien resurssien määrä on huomattavasti pienentynyt tai jotain on pielessä. Custom Datan lataaminen Cloudwatch Metricsiin AWS:n avulla on mahdollista kahdella tavalla: EC2-instanssilla esimerkiksi Crontabin ja Bash-skriptin avulla (Kuva 17), tai Lambda-funktiolla ja Cloudwatch Eventsin Cron Expressionia hyödyntämällä. EC2-instanssia hyödyntämällä homma on todella yksinkertainen, mutta siihen vaaditaan aina päällä oleva instanssi, joka suorittaa skriptiä. Näin ollen Lambda-funktio on huomattavasti käytännöllisempi.

```
1  #!/bin/bash
2  maara=$(aws backup list-recovery-points-by-backup-vault --backup-vault-name Default | grep "COMPLETED" | wc -l)
3  aws cloudwatch put-metric-data --namespace "CustomNameSpace" --metric-name
   "AmountOfRecoveryPointsInDefaultBUVault" --value $maara --region eu-central-1
4
5  */15 * * * * /home/ec2-user/skripti.sh
```

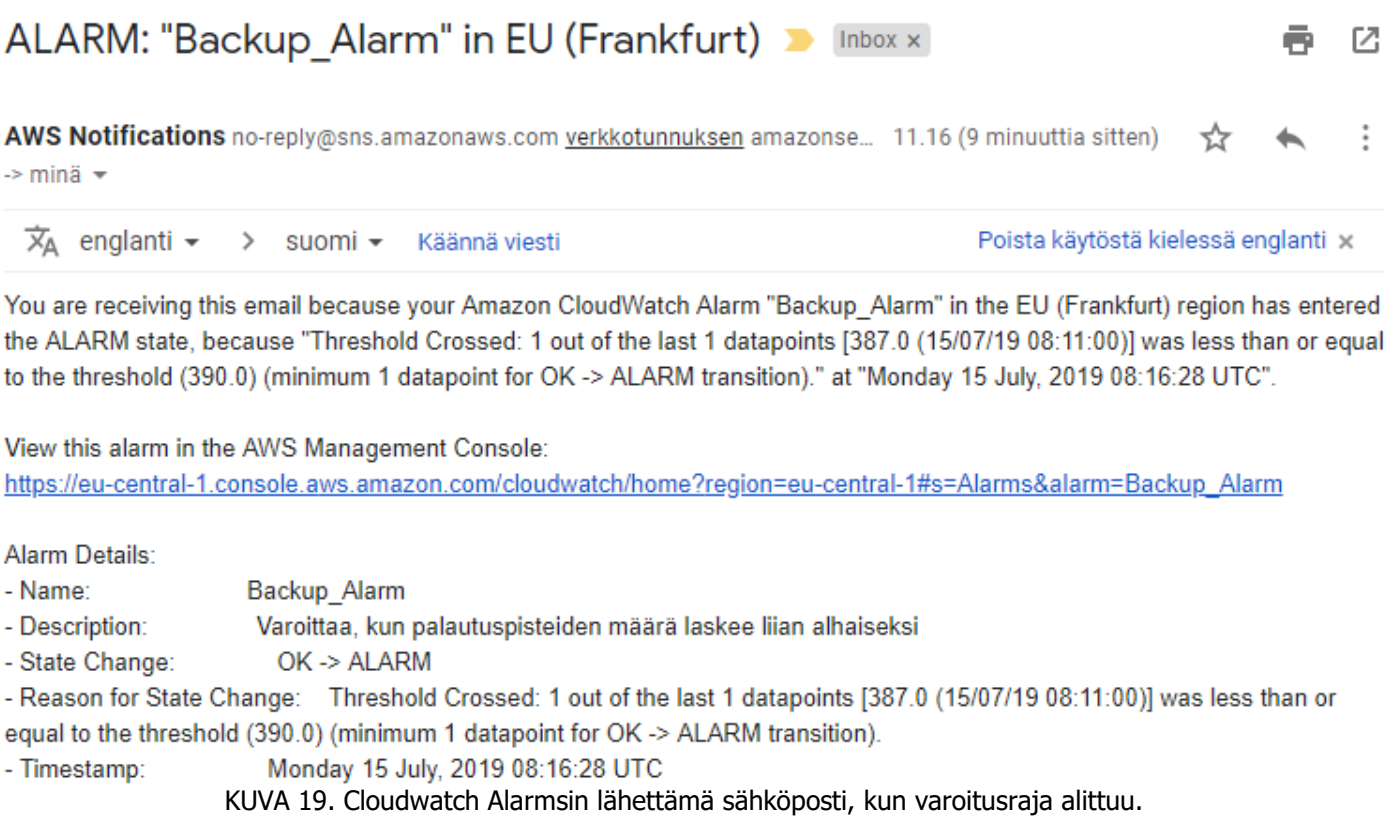
Kuva 17. Bash-kielinen skripti, joka hakee Default-nimisestä Backup Vaultista onnistuneiden varmuuskopioiden määrän, ja tallettaa kyseisen määrän Cloudwatch Metricsin Custom Namespaceen. Alin rivi lisätään osaksi Crontab-funktiota, jolloin 15 minuutin välein ajetaan skripti.sh-tiedosto.

Lambda-funktio luotiin Pythonilla (Kuva 18) käyttäen Pythonin Boto-pakettia (Boto, Github), joka toimii rajapintana Pythonin ja AWS:n välillä. Perusperiaate on sama kuin Bashilla ajettavalla skriptillä: ensin haetaan palautuspisteiden määrä, minkä jälkeen talletetaan se Cloudwatch Metricsin sisään.

```
import boto3
def lambda_handler(event, context):
    backup = boto3.client("backup")
    cloudwatch = boto3.client("cloudwatch")
    response2 = backup.list_recovery_points_by_backup_vault(
        BackupVaultName="Default"
    )
    backupmaara = str(response2)
    maara = backupmaara.count("COMPLETED")
    response = cloudwatch.put_metric_data(
        MetricData=[
            {
                'MetricName': 'AmountOfRecoveryPoints',
                'Dimensions': [
                    {
                        'Name': 'BackUpVault',
                        'Value': 'Default'
                    },
                ],
                'Unit': 'None',
                'Value': maara
            },
        ],
        Namespace='CustomNameSpace'
)
```

KUVA 18. Lambdaan laitettava Python-skripti. Ajettavat koodit laitetaan lambda_handlerin sisään, ja Lambda-funktiossa määritellään handlerin nimeksi [tiedoston nimi].lambda_handler .

AWS Backupin tuoreudesta johtuen sitä ei ole oletuksena AWS:n sisällä olevassa Boto-paketissa, ja jouduinkin luomaan uuden ZIP-paketin, joka sisältää uusimman version kyseisestä paketista. Asensin siis uusimman version Botosta (Boto3) Python-terminaalin avulla paikalliseen kansioon, laitoin saman kansion sisälle haluamani Lambda-skriptin ja lopulta pakkasin kaikki nämä tiedostoksi, jonka latasin Lambdaan.



KUVA 19. Cloudwatch Alarmsin lähettämä sähköposti, kun varoitusraja alittuu.

5.2 Aurora Snapshot Tool

Aurora Snapshot Tool käyttää CloudWatch Alarm -palvelua (Kuva 20), joka raportoi epäonnistuneista palautuksista SNS:n avulla. CloudWatch Alarms tarkkailee Step Functionsin toimintaa, ja varoittaa, jos State Machine (Step Functionsin käyttämät komennot) ei onnistu tehtävässään syystä tai toisesta. Varoitus ohjataan luomisvaiheessa määriteltyyn SNS-topiciin, ja mikäli luomisvaiheessa CloudFormationiin ei määritellä mitään SNS-topicia, luodaan uusi.

CloudWatch Alarmsiin luodaan kaksi alarmia jokaista Snapshot Toolin pohjaa kohti. Lähdetunnukselle epäonnistuneesta varmuuskopioinnista ja snapshottien poistosta kertova, sekä kohdetunnukselle snapshottien epäonnistuneesta kopioinnista tai vanhojen snapshottien poistosta kertova. Nämä Alarmit laittavat määriteltyyn Endpointtiin – HTTP(S), SMS, sähköposti tms. – viestiä, mikäli tila muuttuu. Mahdollisia tiloja on kolme, OK, Insufficient Data sekä Alarm, ja jokaisesta muutoksesta tulee hälytys. Eli myös siitä, kun status menee Alarmista OK:hon. Luomisvaiheessa on mahdollista, että tulee ylimääräinen alarm, kun kaikki palaset eivät ole vielä kerinneet toimintaan mukaan.

Name	State	Conditions	Actions status
snapshot-tool-source-samalleTunnukselle-alarmcwDeleteOldFailed-1XW4OWYJ0SCM8	OK	ExecutionsFailed >= 2 for 2 datapoints within 2 hours	No notifications
snapshot-tool-dest-samalleTunnukselle-euWest1-alarmcwDeleteOldFailedDest-1H3Z9YXOHJY46	OK	ExecutionsFailed >= 2 for 2 datapoints within 2 hours	No notifications
snapshot-tool-dest-samalleTunnukselle-euWest1-alarmcwCopyFailedDest-1WYBNTTRUNQDW	OK	ExecutionsFailed >= 2 for 1 datapoints within 5 minutes	No notifications
snapshot-tool-source-samalleTunnukselle-alarmcwBackupsFailed-5TTBL7KI6IYM	OK	ExecutionsFailed >= 1 for 1 datapoints within 5 minutes	No notifications

KUVA 20. Aurora Snapshot Toolin luomat Cloudwatch Alarmit, kun varmuukopiointi on laitettu samalle tunnukselle. CloudWatch on region-eroteltu palvelu, joten joka regionille on omat alarminsa. CloudWatchin käyttäminen on varsin edullista. Custom Metriikan luominen maksaa 0.30$/kk, ja alarmin pitäminen 0.10$/kk/alarm. Snapshot Toolin käyttämät neljä alarmia ja AWS Backupin Custom Metric alarmineen maksaisivat yhteensä siis 0.80$/kk.

6 YHTEENVETO JA POHDINTA

Tarkoituksena oli luoda käyttökelpoinen, helposti käyttöön otettava ja testattu varmuuskopiointiratkaisu, joka täyttää vaaditut kriteerit hallittavuuden, automatisoinnin ja raportoinnin suhteen. Vaatimuksiin pystyttiin vastaamaan hyvin, vaikka jatkuvassa murroksessa olevan tekniikan takia joitakin kompromissejä jouduttiin tekemään. Tulevaisuudessa, AWS Backupin kehityttyä ja sopivien ominaisuuksien saapuessa esimerkiksi Aurora Snapshot Toolin käytöstä voidaan varmastikin luopua, jolloin saadaan yksi iso muuttuja poistettua yhtälöstä ja hallinta on yksinkertaisempaa ja helpompaa. Tämän lisäksi Cross-Account- ja Cross-Region -varmuuskopiointi on nykyisellään mahdoton toteuttaa AWS Backupilla, mutta ainakin Cross-Region -varmuuskopiointimahdollisuuden AWS on luvannut toteuttaa jo kuluvan vuoden aikana.

Työssä kohdatut ongelmat olivat varsin vähäisiä, ja niistä selvisittiin kevyellä ohjauksella. Autotagin kovakoodaus-ominaisuuden kehittäminen oli ainut isompi ongelma, jonka pohtimisessa meni pidemmän aikaa. Vähäinen kokemus Javascriptin sekä nodejs:n kanssa yhdistettynä työlääseen käyttöönottoon ja testaukseen oli aikaavievä yhdistelmä. AWS Backupin raportointiratkaisussa Ropon ohjaajan tuki tuli tarpeeseen.

Työ oli mielenkiintoinen ja antoisa. Opintojeni aikana suoritettu AWS Certified Solutions Architect -sertifikaatti ei mennyt hukkaan, ja oli hienoa päästä toteuttamaan aidosti käyttöön tuleva ratkaisu. Ropo Capital tarjosi erinomaiset puitteet asian hoitamiseen, ja ympäristö sekä dokumentaatio tekemisistäni jää heidän haltuunsa.

7 LÄHTEET

Amazon EC2 User Guide, Using Tags. [Viitattu 6.8.2019.] Saatavissa: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html

PIETIKÄINEN, Suvi 2016. Jatkuvuussuunnittelun käsitteet ja määritelmät. [Viitattu 6.8.2019.] Saatavissa: https://www.vahtiohje.fi/web/guest/3-jatkuvuussuunnittelun-kasitteet-ja-maaritelmat

Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.5 Percent in 2019, Gartner Press Releases. [Viitattu 11.8.2019] Saatavissa: https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g

Benefits of Cloud Computing, IBM. [Viitattu 11.8.2019] Saatavissa: https://www.ibm.com/cloud/learn/benefits-of-cloud-computing

Ropo tänään, Ropo Capital [Viitattu 11.8.2019] Saatavissa: https://www.ropocapital.fi/fi/yritys/

ROCK, Tracy. Oops! 75% of data loss from human error. Here's how to deal with it. [Viitattu 11.8.2019] Saatavissa: https://invenioit.com/continuity/data-loss-from-human-error/

SVERDLIK Yevgeniy, 2018. AWS Says It's Never Seen a Whole Data Center Go Down. [Viitattu 11.8.2019] Saatavissa: https://www.datacenterknowledge.com/amazon/aws-says-it-s-never-seen-whole-data-center-go-down

Amazon S3 Pricing, AWS. [Viitattu 6.8.2019] Saatavissa: https://aws.amazon.com/s3/pricing/

LABRIE Sean, 2016. Automatic AWS Snapshots with Replication to another Region. [Viitattu 11.8.2019] Saatavissa: http://www.seanlabrie.com/2016/automatic-aws-snapshots-with-replication-to-another-region/

AWS Backup Pricing, AWS. [Viitattu 11.8.2019] Saatavissa: https://aws.amazon.com/backup/pricing/

RIIKONEN, Juha 2009. Tietojärjestelmän varmuuskopioinnin suunnittelu, testaus ja toteutus. [Viitattu 11.8.2019] Saatavissa: https://www.theseus.fi/bitstream/handle/10024/2069/Riikonen_Juha.pdf?sequence=1&isAllowed=y

EFS-to-EFS Architecture Overview, AWS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/solutions/latest/efs-to-efs-backup/architecture.html

Backing Up Amazon EFS File Systems Using AWS Data Pipeline, AWS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/efs/latest/ug/alternative-efs-backup.html

Amazon EFS Performance, AWS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/efs/latest/ug/performance.html

Using AWS Backup with Amazon EFS, AWS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/efs/latest/ug/awsbackup.html

Amazon EFS Pricing, AWS. [Viitattu 11.8.2019] Saatavissa: https://aws.amazon.com/efs/pricing/

Backing up and Restoring an Aurora DB Cluster, AWS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html

Working with Backups, AWS RDS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_WorkingWithAutomatedBackups.html

Snapshot Tool for Amazon RDS, Github. [Viitattu 11.8.2019] Saatavissa: https://github.com/awslabs/rds-snapshot-tool

Snapshot Tool for Amazon Aurora, Github. [Viitattu 11.8.2019] Saatavissa: https://github.com/awslabs/aurora-snapshot-tool

Amazon Aurora Pricing, AWS. [Viitattu 11.8.2019] https://aws.amazon.com/rds/aurora/pricing/

Point-in-Time Recovery, AWS. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/pointintimerecovery_beforeyoubegin.html

Amazon DynamoDB Pricing, AWS. [Viitattu 11.8.2019] Saatavissa: https://aws.amazon.com/dynamodb/pricing/

AWS Backup – Automate and Centrally Manage Your Backups, AWS News Blog. [Viitattu 11.8.2019] Saatavissa: https://aws.amazon.com/blogs/aws/aws-backup-automate-and-centrally-manage-your-backups/

MARTINI Alessandro, 2016. How to Automatically Tag Amazon EC2 Resources in Response to API Events, AWS Security Blog. [Viitattu 11.8.2019] Saatavissa: https://aws.amazon.com/blogs/security/how-to-automatically-tag-amazon-ec2-resources-in-response-to-api-events/

Autotag, Github. [Viitattu 11.8.2019] https://github.com/GorillaStack/auto-tag

CRR Examples, AWS Documentation. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/AmazonS3/latest/dev/crr-walkthrough-2.html

Bucket Policy Examples, AWS Documentation. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-9

Granting Permissions for Amazon S3 Batch Operations, AWS Documentation. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/AmazonS3/latest/dev/batch-ops-iam-role-policies.html

Amazon S3 Inventory, AWS Documentation. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/AmazonS3/latest/dev/storage-inventory.html

AWS Backup will Automatically Copy Tags from Resource to Recovery Point, About AWS. [Viitattu 11.8.2019] Saatavissa: https://aws.amazon.com/about-aws/whats-new/2019/07/aws-backup-will-automatically-copy-tags-from-resource-to-recovery-point/

Using AWS CloudFormation Templates with AWS Backup, AWS Developer Guide. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/aws-backup/latest/devguide/integrate-cloudformation-with-aws-backup.html

AWS Backup API Actions, AWS Developer Guide. [Viitattu 11.8.2019] Saatavissa: https://docs.aws.amazon.com/aws-backup/latest/devguide/API_Operations.html

Boto, Github. [Viitattu 11.8.2019] Saatavissa: https://github.com/boto/boto

LIITTEET

LIITE 1: CLOUDFORMATION-TEMPLATE AWS BACKUPILLE

```
{
    "Description":"Backup Plan template to back up all appropriately tagged resources daily at 5AM Finnish
time. Backups are kept for 35d, and a new IAM-role will be created",

             "Parameters": {
        "BackUpVaultNameParameter": {
            "Description": "Backup Vaultin nimi",
            "Type": "String",
            "Default": "Default"
        },
        "BackUpPlanNameParameter": {
            "Description": "Backup Planin nimi",
            "Type": "String",
            "Default": "New Backup Plan"
        },
        "TagKey": {
            "Description": "Tag Key, joka varmuuskopioidaan",
            "Type": "String",
            "Default": "Key"
        },
        "TagValue": {
            "Description": "Tag Value, joka varmuuskopioidaan",
            "Type": "String",
            "Default": "Value"
        }

    },
             "Resources": {


        "KMSKey": {
            "Type": "AWS::KMS::Key",
            "Properties": {
                "Description": "Encryption key for daily",
                "EnableKeyRotation": true,
                "Enabled": true,
                "KeyPolicy": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "AWS": {
                                    "Fn::Sub": "arn:${AWS::Partition}:iam::${AWS::AccountId}:root"
                                }
                            },
                            "Action": [
                                "kms:*"
                            ],
                            "Resource": "*"
                        }
                    ]
                }
            }
        },
        "BackUpVault": {
            "Type": "AWS::Backup::BackupVault",
            "Properties": {
                "BackupVaultName": {
                    "Ref": "BackUpVaultNameParameter"
                },
                "EncryptionKeyArn": {
                    "Fn::GetAtt": [
                        "KMSKey",
                        "Arn"
                    ]
                }
            }
        },
        "BackupPlan": {
            "Type": "AWS::Backup::BackupPlan",
            "Properties": {
                "BackupPlan": {
                    "BackupPlanName":  {
                    "Ref": "BackUpPlanNameParameter"
```

```
                },
                    "BackupPlanRule": [
                        {
                            "RuleName": "RuleForBackups",
                            "TargetBackupVault": {
                                "Ref": "BackUpVault"
                            },
                                                                    "Lifecycle": {"DeleteAfterDays": 35},
                            "ScheduleExpression": "cron(0 2 ? * * *)"
                        }
                    ]
                }
            },
            "DependsOn": "BackUpVault"
        },
        "BackupRole": {
            "Type": "AWS::IAM::Role",
            "Properties": {
                "AssumeRolePolicyDocument": {
                    "Version": "2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": [
                                    "backup.amazonaws.com"
                                ]
                            },
                            "Action": [
                                "sts:AssumeRole"
                            ]
                        }
                    ]
                },
                "ManagedPolicyArns": [
                    "arn:aws:iam::aws:policy/service-role/AWSBackupServiceRolePolicyForBackup"
                ]
            }
        },
        "TagBasedBackupSelection": {
            "Type": "AWS::Backup::BackupSelection",
            "Properties": {
                "BackupSelection": {
                    "SelectionName": "TagBasedBackupSelection",
                    "IamRoleArn": {
                        "Fn::GetAtt": [
                            "BackupRole",
                            "Arn"
                        ]
                    },
                    "ListOfTags": [
                        {
                            "ConditionType": "STRINGEQUALS",
                            "ConditionKey": {
                                "Ref": "TagKey"
                            },
                            "ConditionValue": {
                                "Ref": "TagValue"
                            }
                        }
                    ]
                },
                "BackupPlanId": {
                    "Ref": "BackupPlan"
                }
            },
            "DependsOn": "BackupPlan"
        }
    }
}
```