
Isabella Lumenlehto

# AWS-ympäristön julkisten IP-osoitteiden rajoitus

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tutkinto-ohjelman nimi

Insinöörityö

21.10.2024

# Tiivistelmä


| | |
|---|---|
| Tekijä: | Isabella Lumenlehto |
| Otsikko: | AWS-ympäristön julkisten IP-osoitteiden rajoitus |
| Sivumäärä: | 23 sivua + 1 liite |
| Aika: | 21.10.2024 |
| Tutkinto: | Insinööri (AMK) |
| Tutkinto-ohjelma: | Tieto- ja viestintätekniikka |
| Ohjaaja: | Osaamisaluepäällikkö Janne Salonen |


---


Yksi pilvipalvelujen eduista on resurssien luomisen helppous ja nopeus. Sen varjopuolena etenkin kompleksisissa ympäristöissä on pilven resurssien hallinnan ja elinkaaren seurannan vaikeus. Suomalaisessa IT-alan yrityksessä AWS-ympäristöön oli vuosien saatossa kertynyt satoja julkisia IP-osoitteita vailla nimettyjä omistajia tai IP-osoitteiden elinkaaren hallinnan prosesseja.

Opinnäytetyössä tarkastellaan miten tähän tilanteeseen päädyttiin ja mitä potentiaalisia ongelmia se voi aiheuttaa. Sen jälkeen esitellään suunnitelma ongelman korjaamiselle. Olemassa olevat julkiset IP-osoitteet tapauskohtaisesti joko poistetaan, korvataan muilla ratkaisuilla tai korvataan paremmin tarpeeseen soveltuvilla staattisilla IP-osoitteilla. Julkisille IP-osoitteille vaaditaan myös suojaavia kontrolleja kuten Dos/Ddos-suojaus ja tarvittaessa sovelluspalomuuri.

Sen lisäksi esitellään miten voidaan välttää samaan tilanteeseen päätyminen jatkossa käyttäen teknisiä kontrolleja, jotka rajoittavat uusien julkisten IP-osoitteiden luomista.



Avainsanat: AWS, pilvi, IP, tietoturva

## Abstract


| | |
|---|---|
| Author: | Isabella Lumenlehto |
| Title: | Restriction of public IP addresses in an AWS environment |
| Number of Pages: | 23 pages + 1 appendix |
| Date: | 21 October 2024 |
| Degree: | Bachelor of Engineering |
| Degree Programme: | Information and Communications Technology |
| Supervisors: | Janne Salonen, Head of School (ICT) |


---


One of the key benefits of cloud computing services is the ease and speed of creating new resources. One of the downsides is the difficulty of controlling resources and their lifecycle, in particular in complex environments. One Finnish IT company's AWS environment had become populated with hundreds of public IP addresses without named owners or suitable processes for lifecycle management.

In this thesis we will examine how the company arrived in this situation and the potential problems caused by it. After that we present a plan to fix the problem. The pre-existing public IP addresses are on a case by case basis either removed, replaced with other solutions or replaced with static public IP addresses better suited for the use case. Public IP addresses are also required to have protective controls such as firewalls, Dos/Ddos protection and if needed, web application firewalls.

In addition to this we present a solution for ensuring that similar problems cannot arise in the future by implementing a technical control that restricts users from creating new public IP addresses.



Keywords: AWS, cloud, IP, cyber security

## Sisällys

## Lyhenteet

ARN: Amazon Resource Name. Yksilöivä tunniste, jolla AWS:n resurssit voidaan tunnistaa ja niihin voidaan viitata.

AWS: Amazon Web Services, julkipilven palveluntarjoaja

CSV: Comma separated values. Tiedostomuoto, jossa data on eroteltu pilkuilla.

DoS: Denial of service, palvelunestohyökkäys

DdoS: Distributed denial of service, hajautettu palvelunestohyökkäys

EC2: Elastic Cloud Compute. AWS:n tarjoama PaaS-palvelu, jossa käyttäjä voi ostaa virtuaalikoneiden laskentatehoa internetin yli

EIP: Elastic IP, AWS:n staattinen, julkinen IP-osoite.

ELB: Elastic Load Balancing, AWS:n tarjoama pilvipohjainen kuormanjakaja.

ENI: Elastic Network Interface, AWS:n verkkoliitin.

IaaS: Infrastructure as a service, infrastruktuuri palveluna.

IP-osoite: Internet Protocol -osoite. Lyhennetään usein ”IP”.

NAT: Network address translation, osoitteenmuunnos.

OSI: Open Systems Interconnection. Viitemalli, joka kuvaa tietoliikennejärjestelmien toimintaa.

SCP: Service Control Policy, AWS:n rajoittava kontrolli

VPC: Virtual Private Cloud, AWS:n pilvipohjainen, virtuaalinen tietoliikenneverkko.

WAF: Web application firewall, sovelluspalomuuri. Tuote, joka suojaa internetin yli tarjottavia palveluita OSI-mallin kerroksella 7.

# 1 Johdanto

Yksi pilvipalvelujen eduista on resurssien luomisen helppous ja nopeus. Sen varjopuolena etenkin kompleksisissa ympäristöissä on pilven resurssien hallinnan ja elinkaaren seurannan vaikeus. Tämä opinnäytetyö tehtiin toimeksiantona suomalaiselle, IT-alalla toimivalle yritykselle. Yrityksellä oli haasteena heidän AWS-pilviympäristönsä (Amazon Web Services) suureksi kasvanut tarpeettomien julkisten IP-osoitteiden määrä. Tavoitteena oli laatia suunnitelma toimenpiteille, joilla saadaan sekä poistettua tarpeettomaksi havaitut julkiset IP-osoitteet että varmistettua, että jatkossa turhia resursseja ei enää synny.

Tässä luvussa tarkastellaan ensin miten yrityksessä päädyttiin nykytilanteeseen. Sen jälkeen kuvataan tavoitetila, jota kohti on tarkoitus pyrkiä.

## 1.1 Taustaa

AWS-pilvialusta otettiin yrityksessä käyttöön 2010-luvun puolivälissä. Varsinainen liiketoiminta oli keskittynyt vielä perinteisiin konesaleihin, mutta pilvipalveluja kokeiltiin ja testattiin ”start up” -henkisesti yrityksen sisällä pilven mahdollistaman nopean innovoinnin takia.

Yritys toimii tiukasti reguloidulla alalla. Pilveen rakennettavat palvelut eivät kuitenkaan olleet liiketoiminnalle kriittisiä tai sisältäneet arkaluontoista dataa, joten alustan käyttöön suhtauduttiin alkuvaiheessa hyvin rennosti. Pilven voi jopa ajatella olleen kehittäjien leikkikenttä, jossa he pääsivät kokeilemaan uusia teknologioita tarvitsematta suunnitella projekteja kovin pitkälle tai huolehtia resurssien elinkaaresta tai budjetista yhtä tarkasti kuin varsinaisessa tuotantoympäristössä.

Kehittäjillä oli omiin AWS-tileihinsä laajat oikeudet. Virallista käyttöpolitiikkaa tai kehittäjien toimia rajoittavia teknisiä kontrolleja ei pilviympäristössä alkuvaiheessa ollut juuri lainkaan. Se, noudattivatko arkkitehtoniset ratkaisut alan

parhaita käytäntöjä tai tietoturvallisen kehittämisen periaatteita riippui kehittäjän omasta osaamisesta ja viitseliäisyydestä. Huonojen ratkaisujen rakentamista ei estetty eikä kehittäjien sovelluksia katselmoitu.

## 1.2 Ongelman kuvaus

Pilvialustan rento kulttuuri mahdollisti paitsi innovointia ja uusia kokeiluja, se myös jätti jälkeensä suuren määrän resursseja ilman nimettyä omistajaa tai minkäänlaista elinkaarenhallintaa. Vuosien varrella pilvialustan rooli muuttui: pilveen alettiin kehittää myös loppukäyttäjille tarjottavia palveluja ja alustan käyttöön alettiin suhtautua vakavammin.

Alustan ylläpitämiseen muodostettiin oma tiimi. Tämä tiimi teki pilvialustalle merkittäviä parannuksia ja loi toimivan ympäristön uusille AWS:ää hyödyntäville sovelluksille. Parannuksista huolimatta aiemmin vallinnut vapaa kehittämiskulttuuri jätti jälkeensä runsaasti omistajattomia resursseja ja niiden julkisia IP-osoitteita.

Tässä opinnäytetyössä keskitytään julkisiin IP-osoitteisiin eli sellaisiin IP-osoitteisiin, jotka ovat saavutettavissa julkisen internetin yli. Tietoturvan näkökulmasta kaikki avaukset internetiin tulisi tehdä harkiten, käyttäen tarvittavia suojausmekanismeja kuten palomuureja, sovelluspalomuureja ja DoS/DdoS-hyökkäyksiltä suojaavia kontrolleja. Tärkeillä resursseilla kuten julkisilla IP-osoitteilla tulisi olla nimetty omistaja, joka on vastuussa resurssista. Sen lisäksi resursseilla tulisi olla hyvin määritelty elinkaarenhallinnan prosessi, jotta tarvittavat päivitykset tulee tehtyä ja resurssi varmasti poistetaan kun sitä ei enää tarvita.

Tarpeettomat julkiset IP-osoitteet paitsi lisäävät verkkorikollisten hyökkäyspinta-alaa, ne lisäävät myös turhia kustannuksia. AWS ilmoitti vuonna 2023, että se alkaa veloittaa pienen maksun jokaisesta julkisesta IPv4-osoitteesta helmikuusta 2024 alkaen (Barr 2024). Kustannus per IP on pieni ($0,005 per IP per tunti), mutta jos tarpeettomia IP-osoitteita on satoja tai jopa tuhansia, niin kustannus kasvaa heti usealla kertaluokalla ja voi muodostaa merkittäviä turhia lisäkuluja.

### 1.3 Tavoitetila

Tavoitteena oli päästä tilanteeseen, jossa kaikki yrityksen AWS-ympäristön julkiverkkoon näkyvät IP-osoitteet täyttävät seuraavat ehdot:

- Ne on luotu tarkoituksella ja perustellusta syystä.
- Niillä on nimetty omistaja.
- Niille on määritelty elinkaarenhallinnan prosessi.
- Niiden takana olevat järjestelmät on suojattu sovellustason sekä volumetristen palvelunestohyökkäysten varalta käyttäen sovelluspalomuuria sekä sisääntulevan verkkoliikenteen suodatusta.

## 2 Projektin suunnittelu

Tässä luvussa käydään läpi niitä työvaiheita, jotka on tunnistettu tarpeelliseksi tavoitetilaan pääsemiseksi. Haasteina projektilla on henkilöresurssien rajallinen käytettävyys sekä AWS:ää hyödyntävien tiimien kehittäjien osaamisen heterogeenisuus. Osa on todella kokeneita ja alan sertifikaatteja suorittaneita ihmisiä, kun taas osalla tausta on enemmän perinteisessä koodauksessa ja pilviympäristöt ovat uudempi tuttavuus. Siten ongelman potentiaalisen ratkaisun yhtenä vaatimuksena on, että se voidaan toteuttaa mahdollisimman pienillä resursseilla ja alustan hyödyntäjille annettujen ohjeiden noudattaminen onnistuu myös noviiseilta.

### 2.1 Nykyisten IP-osoitteiden kartoitus

Luonnollisesti ensimmäinen vaihe on nykytilanteen kartoittaminen. AWS-ympäristö kattaa useamman regioonan, satoja AWS-tilejä ja tuhansia resursseja. Pilvialustojen resurssit ovat hyvin lyhytikäisiä, joten niiden viemistä ulkoiseen CMDB-järjestelmään ei olla edes harkittu. Kartoitus on siis tehtävä käyttäen AWS:n natiiveja työkaluja, kuten AWS Config.

## 2.2 Tavoitetilan rajaus

Ottaen huomioon käytössä olevien resurssien vähyyden, päädytään siihen lopputulokseen, että luovumme seuraavasta ehdosta:

- Julkisella IP:llä on elinkaarenhallinnan prosessi.

Totesimme, että elinkaarenhallinnan toteutus ja seuranta olisi vaatinut merkittävästi suunnittelua ja aikaa, kun taas sen tuomat hyödyt eivät olleet niin suuret. Jo se, että jokaiselle IP:lle nimetään omistaja, joka vastaa IP:stä sen koko olemassaolon ajan on merkittävä parannus lähtötilanteeseen verrattuna. Elinkaarenhallinta olisi toki hyvä implementoida, mutta sen voi antaa odottaa tilannetta, jossa alustan ylläpidosta vastaavilla on enemmän aikaa ottaa vastaan lisää tehtäviä.

## 2.3 Nykyisten IP-osoitteiden kategorisointi

Alustavan arvion mukaan julkisia IP-osoitteita on koko AWS-alustalla 500-1000 kappaletta. AWS tarjoaa lukuisia vaihtoehtoja reitittää liikenne eri palvelujen välillä niin, että liikenne pysyy AWS:n sisällä eikä tarvitse julkista IP-osoitetta. Ainoat käyttötapaukset, joissa aidosti vaaditaan julkisia IP-osoitteita ovat sellaiset, joissa palveluja ja dataa halutaan tarjota käyttäjille julkisen internetin yli (Amazon 2024d). Ottaen huomioon kuinka vähän yrityksellä on tällaisia käyttötapauksia ja kuinka monta julkista IP-osoitetta löydettiin on perusteltua olettaa, että suurin osa IP-osoitteista eivät täytä tätä kriteeriä.

AWS-alustaa ylläpitävällä tiimillä ei ole mitään mahdollisuuksia eritellä IP-osoitteita tai ottaa kantaa siihen mikä näistä on oleellinen ja mikä ei. Heidän työtoimenkuvansa rajoittuu alustan perustoimintojen kuten lokituksen, monitoroinnin, uusien tilien provisioinnin ja laskutuksen ylläpitämiseen. Toisaalta AWS:ssä kehittävien tiimien teknisten kykyjen laaja skaala tarkoittaa sitä, että tämän erittelyn vastuuttaminen tiimeille itselleen ei välttämättä sekään tuota luotettavia tuloksia.

Nyt ympäristössä olevat julkiset IP-osoitteet voidaan jakaa karkeasti kolmeen eri luokkaan:

- Aidosti tarpeellinen julkinen IP-osoite, jonka kautta tarjotaan tarkoituksella sisältöä internetin yli.
- Julkinen IP-osoite, joka voidaan korvata joko yksityisellä IP-osoitteella tai jollain toisella ratkaisulla.
- Vanhentunut resurssi, joka voidaan poistaa kokonaan.

Tarkastellaan seuraavaksi näitä kolmea eri tapausta ja mitä niiden kohdalla tulee ottaa huomioon.

### 2.3.1 Tarpeelliset julkiset IP-osoitteet

AWS:n hyödyntäjien tulee tunnistaa sellaiset käyttötapaukset, joissa heidän todella tarvitsee tarjota jotain sisältöä julkisen internetin yli. Näitä käyttötapauksia varten he voivat tilata itselleen staattisen julkisen IP-osoitteen. AWS:ssä staattinen julkinen IP kulkee nimellä elastinen IP, Elastic IP (Amazon, 2024a).

AWS:n elastiset IP-osoitteet olivat aiemmin maksuttomia tiettyjen ehtojen täyttyessä. IPv4-osoitteiden pienen määrän takia AWS on muuttanut myös elastiset IP-osoitteet maksullisiksi käyttäjille. Siten kustannussäästöjä ei tulla saavuttamaan vaihtamalla staattisiin osoitteisiin, mutta niistä on muita hyötyjä. Staattisten IP-osoitteiden avulla voidaan esimerkiksi parantaa palvelujen saatavuutta rakentamalla arkkitehtuuri niin, että yhden laskentaresurssin toiminnan vaarantuessa staattinen IP liitetään automaattisesti toiseen, toimivaan resurssiin eikä mahdollinen häiriö näy loppukäyttäjille millään tavalla.

Toinen staattisten IP-osoitteiden hyöty on se, että IP pysyy sen varanneen AWS-tilin käytössä kunnes se vapautetaan. Tällä on monia etuja tietoturvan kannalta. Erityisesti DNS-palvelimissa on suositeltavampaa käyttää staattisia IP-osoitteita. Dynaamisten IP-osoitteiden käyttäminen DNS-palvelimissa voi aiheuttaa mahdollisia ongelma- ja virhetilanteita, joissa IP-osoitteen muuttuminen ajan kuluessa johtaa liikenteen reitittämisen väärään kohteeseen. Tämä riski

yhdistettynä puutteelliseen elinkaarenhallinnan prosessiin altistaa yrityksen esimerkiksi kalasteluhyökkäykselle tilanteessa, jossa hyökkääjä saa haltuunsa aiemmin yrityksen käytössä olleen dynaamisen IP-osoitteen.

Staattisen IP-osoitteen käyttö siis takaa sen, että kyseinen IP-osoite pysyy sen varanneen AWS-tilin käytössä kunnes se tietoisesti päätetään vapauttaa. Tämä tarjoaa suojaa IP-osoitteen muuttumista vastaan, mutta sen lisäksi vaaditaan suojausmekanismeja myös taustaresursseille.

Jos julkisen IP-osoitteen käyttötapauksena on web-sovellus, joka käsittelee HTTP-kyselyjä niin sen suojaksi vaaditaan AWS:n sovelluspalomuuri eli WAF (Web Application Firewall) (Amazon 2024e). Sopivan sovelluspalomuurin säännöstön suunnittelu ja hienosäätö vaativat syvällistä ymmärrystä paitsi käytettävistä tietoturva-tuotteista myös suojattavasta sovelluksesta ja sen tyypillisesti vastaanottamasta liikenteestä (Amazon 2024f). Tämä on liian suuri haaste projektin resursseja ajatellen, joten staattisen IP:n tilannut taho voi konfiguroida suojausmekanismit kuntoon yhteistyössä AWS:n yritykselle tarjoaman enterprise support -palvelun tukihenkilöiden kanssa.

Kaikille julkisille IP-osoitteille vaaditaan myös DoS/Ddos-suojausta. AWS tarjoaa OSI-mallin tasoilla L3 ja L4 peruslaatuista AWS Shield Basic -suojausta, joka suojaa tyypillisiltä volumetrisiltä hyökkäyksiltä. Sen lisäksi on mahdollista hankkia edistyneempi AWS Advanced -suojaus, joka tarjoaa parempaa suojausta sekä muita lisäominaisuuksia (Amazon, 2024c).

AWS Shieldin käyttöönotto ja konfigurointi on parasta jättää AWS:n hyödyntäjille, jotka tuntevat omat sovelluksena ja voivat yhdessä AWS:n asiantuntijoiden kanssa suunnittella omalle sovellukselleen parhaiten sopivat asetukset.

### 2.3.2 IP-osoitteet, jotka voidaan korvata muilla ratkaisuilla

Iso osa löydetyistä julkisista IP-osoitteista kuulunee tähän kategoriaan. Kehittäjät ovat saattaneet luoda resursseja ymmärtämättä, että niillä on julkinen IP tai

tietämättä, että on olemassa vaihtoehtoja joissa liikenne on mahdollista reitittää Amazonin runkoverkossa kulkematta julkisen internetin yli.

Muutamia esimerkkejä sellaisista AWS:n teknologioista, jotka voivat potentiaalisesti generoida turhia julkisia IP-osoitteita:

- Amazon RDS (Relational Database Service). Kehittäjä saattaa tietämättään luoda RDS-instanssin, jolla on julkinen IP-osoite. Parempi tapa on luoda instanssi ilman julkista IP-osoitetta ja käyttää vaikka siltapalvelinta (bastion host) tietokantayhteyksien muodostamiseen.
- EC2-virtuaalikoneet (Elastic Compute Cloud). Esimerkiksi virtuaaliverkko VPC:ssä voi aliverkossa olla päällä asetus, joka luo jokaiselle EC2-virtuaalikoneelle julkisen IP-osoitteen. Parempi tapa reitittää liikenne virtuaalikoneelle olisi luoda virtuaalikone yksityiseen aliverkkoon ja käyttää AWS:n kuormanjakajaa ELB (Elastic Load Balancer) reitittämään liikenne virtuaalikoneelle.
- AWS LightSail, PaaS eli platform as a service, jolla käyttäjät voivat luoda virtuaalisia yksityisiä palvelimia (VPS, virtual private server).

Näissä tapauksissa kehittäjät ohjataan jälleen AWS:n asiantuntijoiden puoleen. AWS:n avustuksella he voivat suunnitella ratkaisunsa uudelleen käyttäen vaihtoehtoja, jotka korvaavat turhien julkisten IP-osoitteiden tarpeen muilla ratkaisuilla.

### 2.3.3 Tarpeettomat IP-osoitteet

Julkisten IP-osoitteiden joukossa on varmasti myös täysin tarpeettomia IP-osoitteita, joilla ei ole mitään tarvetta. Pilvialustan historian ja julkisten IP-osoitteiden suuren määrän huomioon ottaen on todennäköistä, että suuri osa julkisista IP-osoitteista kuuluu tähän kategoriaan.

Haasteellista näiden IP-osoitteiden poistamisesta tekee dokumentaation ja omistajatietojen puute. Vuosien saatossa AWS-alustalla on ollut tusinoittain kehittäjiä sekä konsultteja, joiden jäljiltä on jäänyt puutteellisesti dokumentoituja

tai täysin dokumentoimattomia resursseja. Se, että mikään taho ei tällä hetkellä tunnista tarvitsevansa tai käyttävänsä jotain resurssia ei vielä ole tae siitä, että resurssin voi poistaa ongelmitta.

Tunnistamattomien resurssien poistaminen on erityisen riskialtista sellaisilla AWS-tileillä, joissa ratkaisut on rakennettu niin sanotusti ClickOpsina. ClickOpsilla tarkoitetaan työskentelytapaa, jossa resursseja luodaan ja hallinnoidaan graafisen web-käyttöliittymän yli manuaalisesti sen sijaan, että niiden luonti ja päivitys tehtäisiin automatisoidusti. ClickOps soveltuu uusien teknologioiden kevyeen testaamiseen, opiskeluun tai sandbox-ympäristössä toimimiseen. Kyseisen työtavan käyttäminen tuotantoympäristöissä tekee niistä haavoittuvia ja hankalia ylläpitää (Preuss 2024).

Jos pilveen rakennetut ratkaisut on rakennettu käyttäen infrastruktuuri koodina -työtapaa (infrastructure as code), niin palautuminen virhetilanteista on yksinkertaista ja luotettavaa. Suurin osa tunnistamattomista julkisista IP-osoitteista on kuitenkin nähtävästi luotu ClickOpsina, joten niiden poistamisen aiheuttamat potentiaaliset virhetilanteet saattavat olla hyvin hankalia selvittää ja ratkaista.

Näiden IP-osoitteiden poistamisessa on otettava huomioon mahdolliset virhetilanteet ja pyrittävä minimoimaan aiheutuvat haitat.

## 2.4 Prosessi elastisen IP-osoitteen tilaamiselle

Ongelmatilanteeseen on aikoinaan päädytty kun kehittäjät ovat luoneet harkitsemattomasti resursseja, joiden elinkaaresta he eivät kuitenkaan ole ottaneet vastuuta. Samalla kun pyritään ratkaisemaan jo syntynyt ongelma, on tärkeää huolehtia siitä, että vastaavaan tilanteeseen ei tulla päätymään uudelleen.

Eräs ratkaisu tälle on seuraavanlainen prosessi: kun alustan hyödyntäjät tunnistavat tarpeen julkiselle IP-osoitteelle, he tilaavat alustan ylläpitäjiltä elastisen IP-osoitteen. Ainoa taho, jolla on oikeus luoda julkisia IP-osoitteita on

alustan ylläpidosta vastaava tiimi. Samalla prosessilla voidaan varmistaa, että luotavat IP-osoitteet täyttävät niille asetetut vaatimukset kuten omistajan nimeäminen sekä mahdollisten tarvittavien suojausmekanismien konfigurointi.

Prosessin on kuitenkin oltava sopivan kevyt ja helppo, jotta uusien IP-osoitteiden tilaamisesta ei muodostu pullonkaulaa eikä se vie jo nyt ylityöllistetyltä ylläpito-tiimiltä enempää kaistaa kuin on pakko.

## 2.5 Poikkeusten hyväksyminen

Alustan hyödyntäjät noudattavat työssään ketteriä toimintatapoja, joihin kuuluu sprinteissä työskentely. Jokaisella kehitystiimillä on siten oma backloginsa täynnä suunniteltua työtä sekä tietty kadenssi, jolla tehtäviä tehdään. Jos kehittäviltä tiimeiltä vaaditaan lisää työtä - kuten omien käyttötapausten selvittäminen ja uusien IP-osoitteiden tilaaminen – täytyy varautua siihen, että työn aloittaminen voi joutua odottamaan sopivaa hetkeä.

Kuten luvussa 2.3.2 mainittiin on tiedossa, että AWS:ssä on käytetty julkisia IP-osoitteita sellaisissakin tapauksissa, joissa parhaiden käytäntöjen mukaan tulisi käyttää yksityisiä IP-osoitteita. Näissä tilanteissa vaatimuksena kehittäjille on, että he muuttavat sovelluksen arkkitehtuuria niin, että turhista julkisista IP-osoitteista päästään eroon. Kehittäjille tulee antaa aikaa ottaa selvää vaihtoehtoisista toimintatavoista, kuten vaikka AWS Private Link (mahdollistaa liikenteen reitityksen niin, että liikenne ei kulje julkisen internetin yli) tai virtuaalikoneiden korvaaminen serverless-ratkaisuilla kuten AWS Lambda.

Sovelluksen toimintalogiikan muuttaminen, uusien teknologioiden käytön opettelu ja työn aikatauluttaminen tulee viemään aikaa. Näitä tilanteita varten pitää olla jokin poikkeusmenettely, jolla kehittäjä voi pyytää lisä- tai siirtymäaikaa ennen muutoksen tekemistä.

## 2.6 Uusien dynaamisten IP-osoitteiden luomisen estäminen

AWS:ssä on potentiaalia luoda hyvin tarkkoja ja granulaarisia käyttöoikeus-säännöstöjä (IAM policy) jokaiselle käyttäjälle erikseen. Näiden säännöstöjen luonti ja ylläpito suuressa organisaatiossa on kuitenkin aikaa vievää. Osittain sen takia AWS-alustan käyttöönoton yhteydessä oltiin lipsuttu tilanteeseen, jossa lähes kaikilla kehittäjillä oli administrator-oikeudet heidän omaan AWS-tiliinsä.

AWS:ssä kuten muissakin alustoissa on eritasoisia ympäristöjä. Kehittäjillä voi aivan perustellusti olla vahvat käyttöoikeudet kehitys- ja testaus-ympäristöihin, vähän rajatummat oikeudet hyväksyntätestaus- tai staging-ympäristöihin ja erittäin rajalliset oikeudet tuotanto-ympäristöihin.

AWS:n ylläpidosta vastaava tiimi on tehnyt paljon töitä käyttöoikeuksien rajaamisen parantamiseksi. Olimme silti tilanteessa, jossa erityisesti alemmissa ympäristöissä kehittäjillä on laajoja oikeuksia ja siten mahdollisuus jatkaa turhien resurssien ja myös uuden teknisen velan luomista.

Tarvitaan ratkaisu, joka takaa, että jatkossa kaikki uudet julkiset IP:t on tilattu sille suunniteltua prosessia käyttämällä. Ratkaisun tulee sisältää jokin tekninen kontrolli, joka estää prosessin kiertämisen.

AWS:ssä on koko ympäristön käsittäviä kontrolleja kuten Service Control Policy (SCP), jolla voi rajata tiettyjen AWS-tilien kaikkien käyttäjien oikeuksia keskitetysti. Käyttämällä SCP-säännöstöä on mahdollista estää kaikkia alustan käyttäjiä luomasta itse uusia julkisia IP-osoitteita, mikäli näin halutaan.

SCP-säännöstö on yksinkertainen tapa rajoittaa kehittäjien oikeuksia niin, että jatkossa kaikki julkiset IP-osoitteet AWS-alustalla on luotu sitä varten suunniteltua prosessia käyttäen.

## 2.7 Viestintä

Kehittäjille on oltava selkeät kirjalliset ohjeet, joissa kerrotaan mitä heiltä odotetaan. On myös hyödyllistä ennakoida yleisimpiä kysymyksiä, jotta niiden vastaukset voidaan koota ja julkaista sen sijaan, että kysymyksiin vastataan toistuvasti jokaiselle erikseen.

Jotta varmistutaan, että viestintä tavoittaa kaikki kohdeyleisöön kuuluvat, viestiä tulee jakaa useammassa kanavassa. Kehittäjien tiiminvetäjät kuten Product Ownerit ja esimiehet ovat myös avainasemassa kun halutaan varmistaa, että tälle työlle varataan riittävästi aikaa ja sille annetaan riittävä prioriteetti.

Demo-tilaisuuksien ja lisäkoulutuksien järjestäminen on myös hyödyllistä. On varmasti kehittäjiä, jotka taitojen ja tiedon puutteessa luulevat jonkun käyttötapauksen vaativan julkisen IP-osoitteen, vaikka sama saavutettaisiinkin paremmin käyttämällä yksityisiä IP-osoitteita tai sellaisia Amazonin teknologioita, jotka eivät vaadi IP-osoitetta (esim. AWS Lambda, funktio palveluna).

## 2.8 Arvio aikataulusta

AWS:ssä kehittäville tiimeille on annettava aikaa tarkastella heidän omia käyttötapauksiaan. Vie myös oman aikansa suunnitella, toteuttaa ja testata uusien staattisten IP-osoitteiden tilausprosessi. Mahdollinen SCP-säännöstö, joka rajoittaa kehittäjien oikeuksia pitää myös testata huolella, jotta se ei aiheuta käyttökatkoja ja ongelmatilanteita. Liian kauas tulevaisuuteen ei kannata kuitenkaan tähdätä, koska se madaltaa omalta osaltaan työn prioriteettia kehittäjien mielessä.

Alustava arvio tekemiselle on seuraava:

- Tilausprosessin suunnittelu: 1 viikko.
- Tilausprosessin toteutus: 1 viikko.
- Tilausprosessin testaus: 1-2 viikkoa (mahdollisia korjaustoimenpiteitä varten).
- SCP:n suunnittelu: 1 viikko.

- SCP:n toteutus: 1 viikko.
- SCP:n testaus: 1-2 viikkoa (mahdollisia korjaustoimenpiteitä varten).
- Viestinnän suunnittelu: 1 viikko.

SCP:n ja tilausprosessin eri vaiheet voivat edetä rinnakkain, joten ne vievät yhteensä noin kuukauden. Viestintään sisältyy kehittäjille tehtävä dokumentaatio sekä ”usein kysytyt kysymykset”, jolla toivottavasti vältetään ad hoc -viestien tulva.

Kun valmistelu on tehty, niin kehittäjille tulee antaa 2-3 kuukautta aikaa valmistautua muutokseen. Sopivien sovelluspalomuurisäännöstöjen ja ddos-suojausmekanismien suunnittelu voi olla todella aikaa vievää, joten aika-arviossa kannattaa mieluummin olla vähän pessimistinen.

# 3 Projektin toteutus

Tässä luvussa kuvataan tarkemmin miten eri työvaiheet tullaan toteuttamaan.

## 3.1 IP-osoitteiden kartoitus

Ennen mitään muita toimenpiteitä tarvitaan tietoa siitä kuinka monta julkista IP-osoitetta ympäristössä on. Haluamme löytää kaikki resurssit, joilla on julkinen IP-osoite, sekä selvittää resurssin ID:n (yksilöllinen tunniste), nimen, millä AWS-tilillä ja missä regioonassa se sijaitsee, mikä resurssin tyyppi on ja mikä sen IP-osoite on. Nämä tiedot voi hakea AWS Configista seuraavalla hakulausekkeella:

```
SELECT
  resourceId,
  resourceName,
  accountId,
  resourceType,
  awsRegion,
  configuration.association.publicIp
WHERE
  resourceType = 'AWS::EC2::NetworkInterface'
  AND configuration.association.publicIp > '0.0.0.0'
```

Esimerkkikoodi 1. Hakulauseke, jolla haetaan AWS Configistä tietoja resursseista, joilla on julkinen IP-osoite.

Kun ollaan alustavasti selvittämässä julkisten IP-osoitteiden lukumäärää dataa tulee kerätä useiden viikkojen ajalta, koska pilviresurssien mahdollisesti hyvin lyhyen eliniän takia lukema saattaa elää viikosta toiseen. Config-hakukyselyn voi automatisoida ajautumaan kerran viikossa. Datan keräämistä tullaan jatkamaan koko muutosprosessin ajan sekä sen jälkeen.

## 3.2 Datan keräämisen automatisointi

Datan seuraaminen automatisoidaan seuraavilla työkaluilla:

- AWS Lambda: palveliton ”funktio palveluna”,
- AWS EventBridge: tapahtumien reitityspalvelu, jolla voi ajastaa Lambda-funktion ajautumaan väliajoin,
- AWS Config: pilven resurssien tilan ja konfiguraation hallintatyökalu,
- AWS S3: tiedontallennuspalvelu.

EventBridgellä voidaan ajastaa Lambda-funktio ajautumaan kerran viikossa. Lambda-funktio suorittaa Configin hakukyselyn ja kirjoittaa tuloksen S3-palvelun bucketiin (”ämpäri”, AWS:n termi S3:n tietosäilölle). Data tallennetaan CSV-muodossa.

Tämä automaatio, kuten lähes kaikki muukin kehittäminen pilvessä on suositeltavaa tehdä käyttäen infrastruktuuri koodina -lähestymistapaa (infrastructure as code). Opinnäytetyön liitteessä 1 on CloudFormation-templaatti, jolla automaation voi rakentaa. Selkeyden vuoksi esitellään kuitenkin automaation kaikki osat erikseen.

### 3.2.1 S3-bucket

Ensimmäisenä tarvitsemme S3-bucketin, jonne tiedostot tullaan lataamaan. S3-bucketien nimet ovat globaalisti uniikkeja, koska bucketin nimi toimii sen DNS-nimenä. Eräs tapa varmistaa, että toivottu bucketin nimi on saatavilla on käyttää sen luomishetken päivämäärää tai aikaleimaa nimessä.

Bucketin luomiseen on monta keinoa. Yksi niistä on käyttäen komentorivipohjaista työkalua AWS CLI.

```
aws s3api create-bucket --bucket BUCKET_NAME
```

Esimerkkikoodi 2. Komento, jolla voi luoda uuden S3-bucketin käyttäen AWS CLI:tä.

### 3.2.2 Lambda-funktio

Lambda-funktion käyttämä IAM-rooli (Identity and access management) tarvitsee toimiakseen seuraavat oikeudet:

- config:SelectAggregateResourceConfig.
- s3:PutObject.

Nuo oikeudet takaavat, että Lambdalla on oikeus hakea dataa Configista sekä kirjoittaa tiedostoja S3:een. Aiemmin luotu S3-bucket välitetään Lambda-funktiolle ympäristömuuttujana.

```
import json
import boto3
import csv
```

```
import io
import os
from datetime import datetime

# Initialize AWS clients
config_client = boto3.client('config')
s3_client = boto3.client('s3')

def lambda_handler(event, context):
    # Set your S3 bucket and config query details
    s3_bucket = os.environ['S3_BUCKET']  # Get S3 bucket name from
environment variable
    s3_key_prefix = os.environ.get('S3_KEY_PREFIX', 'config-results/')

    # AWS Config query to be run
    query = "SELECT resourceId, resourceName, accountId, resourceType,
awsRegion, configuration.association.publicIp WHERE resourceType =
'AWS::EC2::NetworkInterface' AND configuration.association.publicIp >
'0.0.0.0'"

    # Execute AWS Config query
    try:
        response =
config_client.select_resource_config(Expression=query)
    except Exception as e:
        print(f"Error querying AWS Config: {str(e)}")
        raise e

    # Process results
    config_results = response.get('Results', [])

    if not config_results:
        print("No results from AWS Config query")
        return {
            'statusCode': 200,
            'body': json.dumps('No results from AWS Config query')
        }

    # Prepare CSV data
    csv_buffer = io.StringIO()
    csv_writer = csv.writer(csv_buffer)
```

```
    # Write header based on fields in the query
    csv_writer.writerow(['resourceId', 'resourceName', 'accountId',
'resourceType', 'awsRegion', 'configuration.association.publicIp'])

    for result in config_results:
        data = json.loads(result)
        csv_writer.writerow([
            data['resourceId'],
            data['resourceName'],
            data['accountId'],
            data['resourceType'],
            data['awsRegion'],
            data['configuration']['association']['publicIp']
        ])

    # Generate a timestamped filename for the CSV file
    timestamp = datetime.utcnow().strftime('%Y-%m-%d_%H-%M-%S')
    s3_key = f"{s3_key_prefix}config-query-results_{timestamp}.csv"

    # Upload the CSV to S3
    try:
        s3_client.put_object(
            Bucket=s3_bucket,
            Key=s3_key,
            Body=csv_buffer.getvalue(),
            ContentType='text/csv'
        )
        print(f"CSV file uploaded to S3: {s3_key}")
    except Exception as e:
        print(f"Error uploading CSV to S3: {str(e)}")
        raise e

    return {
        'statusCode': 200,
        'body': json.dumps(f'Config results saved to S3 as {s3_key}')
    }
```

Esimerkkikoodi 3. Lambda-funktio, joka hakee dataa Configsta ja kirjoittaa sen S3-bucketiin.

### 3.2.3 Funktion ajastaminen EventBridgellä

Luotu Lambda-funktio voidaan ajastaa ajautumaan joka viikko käyttäen EventBridgeä. Eräs tapa tehdä se on käyttämällä AWS CLI:tä.

```
aws events put-rule \
  --name "WeeklyLambdaTrigger" \
  --schedule-expression "cron(0 12 ? * 2 *)" \
  --description "Run Lambda function every Monday at 12:00 PM UTC"
```

Esimerkkikoodi 4. EventBridge-säännön luonti AWS CLI:tä käyttäen. Esimerkissä funktio ajautuu joka maanantai klo 12:00 UTC.

EventBridge tarvitsee oikeuksia Lambda-funktion invokoimiseen. Ne voidaan antaa seuraavalla komennolla, korvaamalla toiselle riville Lambda-funktiolle annettu nimi.

```
aws lambda add-permission \
  --function-name <lambda name> \
  --statement-id MyEventBridgeInvoke \
  --action "lambda:InvokeFunction" \
  --principal events.amazonaws.com \
  --source-arn arn:aws:events:<region>:<account-
id>:rule/WeeklyLambdaTrigger
```

Esimerkkikoodi 5. EventBridgen tarvitsemien oikeuksien lisääminen.

Lopuksi liitetään EventBridgen sääntö haluttuun Lambda-funktioon. Viimeisellä rivillä annetaan Lambda-funktion ARN eli Amazon Resource Name, yksilöivä tunniste jolla Amazonin resurssit voidaan tunnistaa ja niihin voidaan viitata.

```
aws events put-targets \
  --rule "WeeklyLambdaTrigger" \
  --targets "Id"="1","Arn"="arn:aws:lambda:<region>:<account-
id>:function:<lambda name>"
```

Esimerkkikoodi 6. EventBridge-säännön liittäminen Lambda-funktioon.

### 3.2.4 Datan tulkitseminen

Kun datan kerääminen on saatu automatisoitua, odotetaan muutamia viikkoja. Kun dataa on kertynyt usealta viikolta voidaan tietoja vertailla ja arvioida kuinka paljon IP-osoitteiden määrä vaihtelee viikosta toiseen.

IP-osoitteiden määrän lisäksi saadusta datasta käy ilmi resurssin tyyppi, nimi, Id sekä millä AWS-tilillä ja regioonalla resurssi sijaitsee. Yrityksen AWS-tilien nimissä on aina mainittu kyseisen tilin ympäristö (dev, test, staging tai prod) joten samalla saadaan tietoa siitä miten julkisten IP-osoitteiden määrä on jakautunut eri ympäristöjen välille. On perusteltua olettaa, että validit käyttötapaukset julkisille IP-osoitteille tulisi sijoittaa ensisijaisesti prod-ympäristöihin kun taas alemmissa ympäristöissä tulisi käyttää vain yksityisiä IP-osoitteita.

## 3.3 IP-osoitteiden kategorisointi

IP-osoitteiden käyttötapausten tunnistaminen vastuutetaan niille tiimeille, joiden AWS-tileillä resurssit ovat. Sen sijaan, että kaikkia julkisia IP-osoitteita kohdeltaisiin lähtökohtaisesti tarpeellisina ja mietittäisiin mitkä niistä voidaan poistaa kohdellaankin niitä lähtökohtaisesti tarpeettomina ja velvoitetaan julkista IP-osoitetta tarvitsevat tiimit tilaamaan itselleen käyttötarkoitukseen sopiva staattinen IP seuraavassa kappaleessa esiteltävää prosessia käyttäen.

Kehittäjiä ohjeistetaan myös tunnistamaan sellaiset ratkaisut, jotka hyödyntävät julkisia IP-osoitteita tarpeetta. Näissä tapauksissa kehittäjien tulee korvata julkiset IP:t jollain toisella ratkaisulla, kuten AWS PrivateLink. PrivateLink mahdollistaa liikenteen reitityksen AWS:n sisällä sekä pilven ja lokaalin konesalin välillä ilman, että liikenne kulkee julkisen internetin yli. AWS tarjoaa myös palveluja, jotka eivät käytä IP-osoitteita (kuten esimerkiksi ’serverless’ eli palvelittomat ratkaisut). (Amazon, 2024b).

Kaikki julkiset IP-osoitteet, jotka eivät kuulu kahteen edellä mainittuun kategoriaan oletetaan tarpeettomiksi ja tullaan poistamaan sopivan siirtymäajan kuluttua.

## 3.4 Staattisen IP-osoitteen tilaaminen

Kun AWS:ssä toimiva tiimi tunnistaa käyttötapauksen, jota varten tarvitaan julkinen IP-osoite he tarvitsevat prosessin, jolla tilata tähän tapaukseen soveltuva staattinen, elastinen IP. Yrityksellä on käytössä IT-palvelunhallintaan tiketöintijärjestelmä, joka soveltuu myös tähän tarkoitukseen.

Kehittäjille julkaistaan lomake, jossa he voivat kuvailla käyttötapauksensa ja perustella miksi he tarvitsevat julkisen IP-osoitteen. Lomake tulee sisältämään muun muassa seuraavat tiedot:

- Perustelu julkisen IP-osoitteen tarpeelle.
- IP-osoitteen omistaja.
- Pyynnön hyväksyjä (tyypillisesti kehitettävän sovelluksen tuoteomistaja, Product Owner tai muu soveltuva taho).

Lomake tulee käsiteltäväksi AWS:n ylläpidosta vastaavalle tiimille. He voivat järjestelmänvalvojan oikeuksilla luoda IP-osoitteen. Myöhemmässä kappaleessa esiteltävä SCP-säännöstö, joka rajoittaa kehittäjien kykyä luoda julkisia IP-osoitteita ei tule vaikuttamaan ylläpitäjien käyttöoikeuksiin.

Kun AWS:n ylläpidosta vastaava tiimi vastaanottaa lomakkeen, he ohjeistavat tilannutta tahoa olemaan yhteydessä yrityksen AWS Solutions Architectiin. Yhdessä AWS:n asiantuntijoiden kanssa IP:n tilannut tiimi varmistaa, että julkisella IP-osoitteella on asianmukaiset palomuuri-, DoS/Ddos-suojaus- ja sovelluspalomuuriasetukset.

## 3.5 Poikkeusten salliminen

On odotettavissa, että vastaan tulee tilanteita joissa kehitystiimit tarvitsevat poikkeusluvan vanhan dynaamisen julkisen IP-osoitteen käyttämiselle. Syitä voivat olla esimerkiksi osaamisen puute tiimissä, poissaolot tai ehkä julkisen IP-osoitteen tarve on poistumassa lähiaikoina. Näitä poikkeustilanteita varten tarvitaan prosessi, jolla tiimi voi anoa lupaa jatkaa IP:n käyttöä.

Poikkeuksia varten luodaan oma erillinen lomake samaan tiketöintijärjestelmään, jota käytetään elastisen IP:n tilaamiseen. Lomake sisältää ainakin seuraavat tiedot:

- Perustelu poikkeuksen anomiselle.
- Julkisen IP-osoitteen nimetty omistaja.
- Pyynnön hyväksyjä (tyypillisesti kehitettävän sovelluksen tuoteomistaja, Product Owner tai muu soveltuva taho).
- Poikkeuksen kesto.

Poikkeusluvan saaneet tiimit voivat merkitä heidän dynaamiset julkiset IP-osoitteensa esimerkiksi käyttämällä tageja. Tagit ovat avain-arvo-pareja, joita käytetään AWS:ssä resurssien tunnistamiseen ja luokitteluun. Mahdollinen tag poikkeusluvan saaneille IP-osoitteelle on ”ExemptionGranted”: ”True”. Näiden tagien avulla voidaan varmistaa, että poikkeusluvan saaneita IP-osoitteita ei tulla poistamaan.

## 3.6 Uusien dynaamisten IP-osoitteiden luomisen estäminen

Turhaksi todettujen IP-osoitteiden siivouksen jälkeen halutaan varmistaa, että vastaavaan tilanteeseen ei päädytä uudelleen. Samalla halutaan varmistaa, että elastisen IP-osoitteen tilaamisprosessia ei voi kiertää. Tähän sopiva ratkaisu on koko AWS-ympäristöön sovellettava Service Control Policy, jolla voidaan rajoittaa kaikkien käyttäjien oikeuksia.

### 3.6.1 SCP-säännöstö

Alla esitellään säännöstö, joka rajoittaa kaikkien käyttäjien oikeuksia luoda julkisia IP-osoitteita ja liittää niitä resursseihin. Yrityksellä on käytössä AWS:n hallintaa helpottava AWS Organisation -rakenne, johon voi luoda organisaatioyksikköjä eli organisational uniteja, OU. Näitä yksikköjä käyttämällä säännöstöä voidaan testata ensin pienellä otoksella testitilejä.

Säännöstö ei vaikuta AWS-organisaation ylimmällä tasolla eli ns. management accountissa. Se ei siten vaikuta AWS:n ylläpidosta vastaavan tiimin oikeuksiin, joten he voivat jatkossa luoda ja liittää resursseihin uusia staattisia, julkisia IP-osoitteita.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny", // säännöstön vaikutus on estävä
      "Action": [
        "ec2:AllocateAddress", // Estää elastisen IP:n luomisen
        "ec2:AssociateAddress", // Estää elastisen IP:n liittämisen
EC2-instanssiin
        "ec2:RunInstances", // Estää julkisen IP:n sisältävän EC2-
instanssin luomisen
        "ec2:CreateNetworkInterface", // Estää julkisen IP:n
sisältävän verkkoliittimen luomisen
        "ec2:ModifyInstanceAttribute"   // Estää instanssin
muokkaamisen julkisen IP:n liittämiseksi
      ],
      "Resource": "*",  // pätee kaikille resursseille
      "Condition": {
        "Bool": {
          "ec2:AssociatePublicIpAddress": "true"      // toiminnot on
estetty vain, jos toimintoon liittyy julkisen IP:n liittäminen
instanssiin
        }
      }
    }
  ]
}
```

Esimerkkikoodi 7. Service Control Policy -säännöstö, jolla estetään käyttäjiä luomasta tai liittämästä julkisia IP-osoitteita EC2-instanssiin.

## 3.7 Viestintä ja muutoksen aikataulu

Kun kaikki valmistelevat toimeenpiteet on tehty, eli:

- on kerätty dataa olemassa olevista julkisista IP-osoitteista,
- on luotu prosessi staattisen, elastisen IP-n tilaamiselle
- on luotu prosessi poikkeusluvan anomiselle ja
- on luotu SCP-säännöstö estämään uusien dynaamisten IP-osoitteiden luonti,

voidaan muutoksesta viestiä AWS-alustan kehittäjille.

Kehittäjille tehdään selkeät kirjalliset ohjeet. Heitä ohjeisestaan tunnistamaan omat käyttötapauksensa julkisille IP-osoitteille ja tilaamaan tarkoitusta varten elastinen IP. Tapauksissa, joissa heillä on käytössä julkinen IP-osoite, jonka voi korvata muilla ratkaisuilla heitä ohjeistetaan olemaan yhteydessä AWS:n asiantuntijoihin.

Kehittäjille kerrotaan miten ja missä tilanteissa he voivat pyytää poikkeuslupaa, jolla he voivat jatkaa vanhan, dynaamisen julkisen IP-osoitteen käyttöä. Poikkeuslupa myönnetään niin lyhyeksi aikaa kuin mahdollista, jonka jälkeen kaikkien odotetaan siirtyvän noudattamaan uusia julkisten IP-osoitteiden käytänteitä.

Siitä hetkestä kun kehittäjiä on informoitu muutoksesta heille annetaan kolme kuukautta aikaa tarkastella omia käyttötapauksiaan ja tehdä tarvittavat muutokset. Tämän siirtymäajan jälkeen kaikki sellaiset dynaamiset julkiset IP-osoitteet, jotka eivät ole saaneet poikkeuslupaa tullaan poistamaan. Poisto voidaan tehdä keskitetysti AWS:n ylläpidosta vastaavan tiimin toimesta.

Saman kolmen kuukauden siirtymäajan jälkeen AWS:n ylläpitotiimi asettaa voimaan SCP-säännöstön, joka estää kehittäjiä luomasta omatoimisesti julkisia IP-osoitteita.

# 4 Yhteenveto

Tässä opinnäytetyössä on tarkasteltu AWS-ympäristöä, johon on kertynyt tarpeettomia julkisia IP-osoitteita. Olemme esitelleet miten ongelma on syntynyt,

mitä potentiaalista haittaa se aiheuttaa sekä esitelleet ratkaisun sille miten ongema ratkaistaan ja miten huolehditaan, että samaan tilanteeseen ei tulla päätymään uudelleen.

## Lähteet

Amazon (2024a). *Elastic IP Addresses* https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html Haettu 14.10.2024

Amazon (2024b). *AWS PrivateLink* https://aws.amazon.com/privatelink/ Haettu 1.10.2024

Amazon (2024c). AWS *Shield Managed Ddos protection* https://aws.amazon.com/shield/ Haettu 1.10.2024

Amazon (2024d). *AWS VPC IP addressing* https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html Haettu 10.10.2024

Amazon (2024e). *AWS WAF* https://aws.amazon.com/waf/ Haettu 11.10.2024

Amazon (2024f). *WAF Rules* https://docs.aws.amazon.com/waf/latest/developerguide/waf-rules.html Haettu 11.10.2024

Barr, J. (2024) *New – AWS Public IPv4 Address Charge + Public IP Insights* Amazon Blog. https://aws.amazon.com/blogs/aws/new-aws-public-ipv4-address-charge-public-ip-insights/ Haettu 1.10.2024

Preuss, H. J. (2024). *What is ClickOps and how can you prevent it* https://blog.equinix.com/blog/2022/12/01/what-is-clickops-and-how-can-you-prevent-it/ Haettu 15.10.2024

## CloudFormation-templaatti

Liitteenä CloudFormation-templaatti, jolla voi luoda luvussa 3 esitellyn datan keruun automaation helposti IaCina (infrastructure as code).

S3-bucketin ja Lambda-funktion nimi on korvattu dummy-muuttujilla BUCKET_NAME ja LAMBDA_NAME.

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  S3Bucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: BUCKET_NAME
      AccessControl: Private

  LambdaExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              Service: lambda.amazonaws.com
            Action: sts:AssumeRole
      Policies:
        - PolicyName: LambdaS3ConfigAccess
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              - Effect: Allow
                Action:
                  - config:SelectResourceConfig
                  - s3:PutObject
                Resource:
                  - !Sub "arn:aws:s3:::BUCKET_NAME/*"
```

```
                - "*"

  ConfigQueryLambda:
    Type: AWS::Lambda::Function
    Properties:
      FunctionName: LAMBDA_NAME
      Handler: lambda_function.lambda_handler
      Role: !GetAtt LambdaExecutionRole.Arn
      Code:
        ZipFile: |
          import json
          import boto3
          import csv
          import io
          import os
          from datetime import datetime

          # Initialize AWS clients
          config_client = boto3.client('config')
          s3_client = boto3.client('s3')

          def lambda_handler(event, context):
              # Set your S3 bucket and config query details
              s3_bucket = os.environ['S3_BUCKET']  # Get S3 bucket name
from environment variable
              s3_key_prefix = os.environ.get('S3_KEY_PREFIX', 'config-
results/')

              # AWS Config query to be run
              query = """
              SELECT
                resourceId,
                resourceName,
                accountId,
                resourceType,
                awsRegion,
                configuration.association.publicIp
              WHERE
                resourceType = 'AWS::EC2::NetworkInterface'
                AND configuration.association.publicIp > '0.0.0.0'
```

```
            """

            # Execute AWS Config query
            try:
                response                                                =
config_client.select_resource_config(Expression=query)
            except Exception as e:
                print(f"Error querying AWS Config: {str(e)}")
                raise e

            # Process results
            config_results = response.get('Results', [])

            if not config_results:
                print("No results from AWS Config query")
                return {
                    'statusCode': 200,
                    'body': json.dumps('No results from AWS Config
query')
                }

            # Prepare CSV data
            csv_buffer = io.StringIO()
            csv_writer = csv.writer(csv_buffer)

            # Write header based on fields in the query
            csv_writer.writerow(['resourceId',        'resourceName',
'accountId',               'resourceType',               'awsRegion',
'configuration.association.publicIp'])

            for result in config_results:
                data = json.loads(result)
                csv_writer.writerow([
                    data['resourceId'],
                    data['resourceName'],
                    data['accountId'],
                    data['resourceType'],
                    data['awsRegion'],
                    data['configuration']['association']['publicIp']
                ])
```

```
                # Generate a timestamped filename for the CSV file
                timestamp  =  datetime.utcnow().strftime('%Y-%m-%d_%H-%M-
%S')
                s3_key          =          f"{s3_key_prefix}config-query-
results_{timestamp}.csv"

                # Upload the CSV to S3
                try:
                    s3_client.put_object(
                        Bucket=s3_bucket,
                        Key=s3_key,
                        Body=csv_buffer.getvalue(),
                        ContentType='text/csv'
                    )
                    print(f"CSV file uploaded to S3: {s3_key}")
                except Exception as e:
                    print(f"Error uploading CSV to S3: {str(e)}")
                    raise e

                return {
                    'statusCode': 200,
                    'body': json.dumps(f'Config results saved to S3 as
{s3_key}')
                }
      Runtime: python3.9
      Timeout: 60
      MemorySize: 128

  LambdaInvokePermission:
    Type: AWS::Lambda::Permission
    Properties:
      FunctionName: !Ref ConfigQueryLambda
      Action: lambda:InvokeFunction
      Principal: events.amazonaws.com

  EventBridgeRule:
    Type: AWS::Events::Rule
    Properties:
      Name: MyWeeklyConfigQueryRule
```

```
      ScheduleExpression: cron(0 12 ? * 2 *)  # Every Monday at 12:00
PM UTC
      State: ENABLED
      Targets:
        - Arn: !GetAtt ConfigQueryLambda.Arn
          Id: ConfigQueryLambdaTarget

Outputs:
  S3BucketName:
    Description: Name of the S3 bucket where results are stored.
    Value: !Ref S3Bucket

  LambdaFunctionName:
    Description: Name of the Lambda function running the AWS Config
query.
    Value: !Ref ConfigQueryLambda

  EventBridgeRuleName:
    Description: Name of the EventBridge rule that triggers the Lambda.
    Value: !Ref EventBridgeRule
```