Fuwad Kalhori

Googlen pilvialustan käyttöönotto isossa organisaatiossa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinöörityö

10.10.2023

Tiivistelmä

Tekijä:	Fuwad Kalhori
Otsikko:	Googlen pilvialustan käyttöönotto isossa organisaatiossa
Sivumäärä:	40 sivua
Aika:	10.10.2023
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Mobile Solutions
Ohjaajat:	Osaamisaluepäällikkö Janne Salonen Yliopettaja Erkki Räsänen

---

Opinnäytetyön tavoitteena oli tutkia Metropolia Ammattikorkeakoulun ympäristöä Googlen pilvialustalla ja luoda dokumentaatio toteutetusta ympäristöstä ja siihen liittyvästä Terraform-lähdekoodista sekä ehdottaa kehittämis- ja parannuskohteita.

Opinnäytetyössä tutkittiin pilviympäristön teknistä toteutusta tarkastelemalla lähdekoodia, joka oli saatavilla Google Cloud Source Repository -palvelusta ja hyödyntämällä pilvialustan tarjoamia komentorivityökaluja ja kyselykieliä.

Lisäksi opinnäytetyössä tutkittiin ja kuvattiin Googlen pilvialustan käyttöönottoprosessi sekä siihen liittyvät tärkeät palvelut.

Opinnäytetyön tuloksena syntyi selkeä ja yksityiskohtainen dokumentaatio sekä ehdotuksia muun muassa kustannusten hallinnan toiminnallisuuksien parantamiseksi ja keskittämiseksi, tietoturvan lisäämiseksi, Terraform-projektitehtaan tehostamiseksi, Terraform-version ylläpitämiseksi ja pilviperustan luomiseksi.

Tutkimuksen tuloksena saadut tiedot ja dokumentaatio ovat tärkeitä, jotta saatuja oppeja ja teknisen toteutuksen kokonaisuutta voidaan jakaa ja hyödyntää laajemmin Metropolian sisällä sekä tarjota arvokasta tietoa myös muille organisaatioille, jotka harkitsevat tai suunnittelevat Googlen pilvialustan käyttöönottoa.

Avainsanat: Google Cloud Platform, Googlen pilvialusta, pilviympäristö, Terraform, pilvialustan käyttöönotto, Metropolia

---

Abstract

Author:	Fuwad Kalhori
Title:	Deployment of Google Cloud Platform in a large organization
Number of Pages:	40 pages
Date:	10 October 2023
Degree:	Bachelor of Engineering
Degree Programme:	Information and communication technology
Professional Major:	Mobile Solutions
Supervisors:	Janne Salonen, Head of School Erkki Räsänen, Principal Lecturer

---

The aim of this thesis was to investigate the environment of Metropolia University of Applied Sciences on the Google Cloud Platform and create documentation of the implemented environment and related Terraform source code, as well as to suggest areas for development and improvement.

In the thesis, the technical implementation of the cloud environment was examined by analyzing the source code available from the Google Cloud Source Repository, utilizing the command-line tools and query languages provided by the platform.

Additionally, the thesis investigaed and described the deployment process of Google Cloud Platform and its related essential services.

The result of the thesis was a clear and detailed documentation along with proposals, including improving and centralizing const control functionalities, enhancing cybersecurity, streamlining the project Terraform factory, maintaining Terraform versions, and drafting a cloud foundation plan.

The information and documentation obtained as a result of the research are essential for sharing and utilizing the knowledge and technical implementation within Metropolia and also for providing valuable information to other organizations considering or planning to adopt the Google Cloud platform.

Keywords:	Google Cloud Platform, Googlen pilvialusta, pilviympäristö, Terraform, deployment of cloud platform, Metropolia

Sisällys

Lyhenteet

GCP: Google Cloud Platform, Googlen pilvialusta.

VPC: Virtual Private Cloud, Virtuaalinen yksityinen pilvi.

CI/CD: Continuous integration / Continious delivery, Jatkuva integrointi / jatkuva käyttöönotto.

IAM: Identity And Access Management, Identiteetin- ja pääsynhallinta.

VM: Virtual Machine, virtuaalikone.

GKR: Google Kubernetes Engine.

IDE: Integrated development environment, integroitu kehitysympäristö.

IoT: Internet of things, esineiden internet.

TKI Tutkimus-, kehitys- ja innovaatiotoiminta.

1 Johdanto

Opinnäytetyön tavoitteena oli tutkia Metropolia Ammattikorkeakoulun ympäristöä Googlen pilvialustalla ja luoda dokumentaatio toteutetusta ympäristöstä ja siihen liittyvästä Terraform-lähdekoodista sekä ehdottaa kehittämis- ja parannuskohteita.

Informaatioteknologian nopeasti muuttuvassa maailmassa organisaatiot kuten korkeakoulut kohtaavat lukuisia haasteita, jotka liittyvät esimerkiksi tietoturvaan, massiivisen datan hallintaan, TKI-projektien tukemiseen ja toteuttamiseen sekä innovatiivisten palveluiden tarjoamiseen opiskelijoille ja henkilökunnalle. Nämä moninaiset tarpeet johtavat organisaatioita kääntymään Googlen pilvialustan (GCP) puoleen houkuttelevana ratkaisuna.

Opiskelijoiden opettaminen pilvialustan käyttöön ja sen avulla innovatiivisten palveluiden kehittäminen auttaa heitä ymmärtämään digitaalisten ratkaisujen merkityksen ja mahdollisuudet liiketoiminnassa. Pilvialustojen hallintaan liittyvät taidot ovat arvokkaita työelämässä, ja niiden oppiminen voi parantaa opiskelijoiden työllistymismahdollisuuksia ja antaa heille etulyöntiaseman uran alkuvaiheessa.

On tärkeää, että korkeakoulut valmistavat opiskelijoita työelämään tarjoamalla heille kattavan koulutuksen ja kokemuksen pilvialustojen käytöstä. Google Cloud Platform (GCP) voi olla houkutteleva ratkaisu myös tästä näkökulmasta, sillä se tarjoaa monipuolisen joukon palveluita ja resursseja, jotka ovat laajalti käytössä nykypäivän työelämässä, erityisesti tekoälyvetoisissa startup-yrityksissä (1).

Opinnäytetyön tilaajana toimi Metropolia Ammattikorkeakoulu. Päätös siirtyä käyttämään Google Cloud -alustaa Metropoliassa syntyi ensisijaisesti Metropolian innovaatiokeskittymien tarpeista, ja tämä päätös tehtiin laajan ja ulkoistetun niin kutsutun "kansainvälisten tiedonsiirtojen riskiarvioinnin” tuloksena.

Metropolia Ammattikorkeakoulu aikoo ottaa GCP:tä käyttöön vaiheittain tutkimus-, kehitys- ja innovaatioprojekteissa, oppilaitos-yhteistyöprojekteissa, yrityshankkeissa sekä koulutuksessa. Käyttöönottoprosessi kesti noin kaksi kuukautta, ja varsinaisen teknisen toteutuksen suoritti Google Cloud -partneri, GAPPS-niminen yritys.

GAPPS järjesti käyttöönottoprosessin aikana useita työpajoja, joissa keskityttiin tärkeisiin GCP:n tarjoamiin palveluihin ja työkaluihin, kuten esimerkiksi identiteettien ja resurssien hallinta (Identity and Access Management, IAM), virtuaaliset yksityiset verkot (Virtual private cloud, VPC), tietoturva ja monitorointi (Security and Monitoring). Käyttöönottoprosessissa opinnäytetyöntekijä toimi projektiassistenttina ja osallistui aktiivisesti koko prosessiin, kirjoitti laajan tutkielman pilvipalvelun IAM-työkalusta ja lopulta sai vastuulleen pilviympäristön ylläpidon.

Metropolia ja Google Cloud Finland ovat aloittaneet tiiviin yhteistyön tunnistaen molempien osapuolten huomattavat hyödyt. Metropolialle teknologiajätin kumppanuus on arvokas, sillä se tarjoaa pääsyn huippuluokan resursseihin ja asiantuntemukseen sekä edistää kestävien ja älykkäiden ratkaisujen innovointia ja kehittämistä. Toisaalta Googlen kannalta yhteistyö Suomen suurimman ammattikorkeakoulun kanssa avaa merkittäviä mahdollisuuksia liiketoiminnan edistämisessä.

2 Google Cloud -pilvialusta

Google Cloud Platform (GCP) on kattava joukko pilvipalveluita, jotka on suunniteltu tarjoamaan organisaatioille ja kehittäjille tehokas ja skaalautuva infrastruktuuri sovellusten ja datan rakentamiseen, käyttöön ja hallintaan pilvessä. GCP tarjoaa laajan valikoiman palveluita erilaisiin käyttötarkoituksiin, kuten esimerkiksi laskentaan, tiedon tallennukseen, koneoppimiseen ja analytiikkaan, mikä tekee siitä monipuolisen alustan erilaisiin IT-tarpeisiin. Se mahdollistaa organisaatioille Googlen maailmanlaajuisen verkkoinfrastruktuurin ja datakeskusten ja edistyksellisten teknologioiden hyödyntämisen digitaalisten toimintojen tehostamiseksi, joustavuuden parantamiseksi ja kustannusten vähentämiseksi.

GCP:ssä käyttäjät voivat hyödyntää erilaisia työkaluja ja resursseja sovellusten kehittämiseen, käyttöönottoon ja suorittamiseen turvallisessa, luotettavassa ja joustavassa pilviympäristössä.

GCP tunnetaan käyttäjäystävällisestä käyttöliittymästään, vahvoista turvallisuusominaisuuksistaan ja laajasta datakeskusverkostostaan; yksi sen datakeskuksista sijaitsee Suomen Haminassa. Käytä ja maksa -hinnoittelumallin avulla organisaatiot voivat skaalata infrastruktuuriaan ja maksaa vain käyttämistään resursseista. Olipa aloittava yritys, joka haluaa julkaista verkkosovelluksen, tai suuri yritys, joka tarvitsee yritystason pilviratkaisuja, GCP tarjoaa yli 200 palvelua ja toiminnallisuutta, joilla organisaatiot voivat täyttää tarpeensa.

Resurssit ja käyttöoikeudet

Resurssi viittaa mihin tahansa laskentayksikköön tai palveluun, jota voidaan hallita, varata tai käyttää GCP-ympäristössä. Tähän kuuluvat virtuaalikoneet, tietokannat, tallennussäiliöt (storage buckets) ja muut vastaavat. Myös esimerkiksi kansiot, projektit, käyttäjäryhmät, roolit ja pääsynoikeudet ovat GCP:n näkökulmasta resursseja. Resurssit ovat GCP-infrastruktuurin rakennuspalikoita, ja niitä hallitaan GCP:n konsolin, asiakaskirjastojen (client library), rajapintojen (API) ja komentorivityökalujen, kuten esimerkiksi ”gcloud”, avulla. (2.)

Identity and Access Management -työkalu (IAM) mahdollistaa yksityiskohtaisen pääsynhallinnan resursseihin, ja se mahdollistaa vähimmäisoikeuksien turvallisuusperiaatteen (principle of least privilege), mikä̈ tarkoittaa sitä, että̈ kukaan ei saa enempää̈ käyttöoikeuksia kuin todella tarvitsee. IAM:n avulla voidaan sekä myöntää että evätä käyttöoikeuksia resurssien usealla tasolla. (3; 4.)

IAM:n avulla käyttöoikeudet määritetään vastaamalla kysymykseen kenellä (kuka) on minkälaista pääsyä (rooli) mihin resurssiin (resurssi) (3).

Käyttäjäryhmät mahdollistavat suositun käytännön, joka helpottaa käyttöoikeuksien hallintaa ja organisoimista. Ryhmien jäsenten ja heidän pääsynoikeuksiensa määrittely ja hallinta tapahtuu keskitetysti. Tämä keskitetty hallinta

mahdollistaa tehokkaan oikeuksien hallinnan ja varmistaa, että käyttäjät saavat tarvittavat oikeudet ja resurssit helposti ja yhtenäisesti. Käyttäjäryhmät ovat olennainen osa pilvialustan hallintaa, joka parantaa turvallisuutta, hallinnan yksinkertaisuutta. (4.)

Verkot

Virtuaalinen yksityinen pilvi (VPC) -verkko on fyysinen verkon digitaalinen versio, joka on toteutettu Googlen tuotantoverkossa käyttäen Andromeda-teknologiaa. Tämä virtuaaliverkko suorittaa useita keskeisiä tehtäviä: se muun muassa mahdollistaa yhteydet Compute Engine -virtuaalikoneille, tarjoaa sisäisiä verkon kuormajakajia (network load balancer), mahdollistaa yhteydet paikan päällä (on-premises) oleviin verkkoihin käyttämällä Cloud VPN -tunneleita ja Cloud Interconnect -palvelun VLAN-liitteitä sekä ohjaa liikennettä GCP:n ulkoisista kuormanjakajista taustaresursseihin. Google Cloudissa projektit voivat sisältää useita VPC-verkkoja, ja uudet projektit alkavat yleensä oletusverkolla, joka on automaattinen VPC-verkko ja sisältää yhden aliverkon kussakin alueessa (region). Tämä ominaisuus voidaan kytkeä pois päältä organisaation käytännöllä (organization policy).

Compute Engine -virtuaalikoneinstanssit ovat virtualisoituja laskentaresursseja, jotka on isännöity Googlen infrastruktuurissa. Nämä instanssit voivat toimia perustana erilaisille palveluille ja tuotteille, kuten Google Kubernetes Engine (GKE) -klustereille, App Engine -joustaville (Flexible) ympäristöinstansseille ja muille Compute Engine -instansseihin perustuville ratkaisuille. Vaikka näitä instansseja voidaan kutsua eri nimillä, kuten Compute Engine -instansseiksi, VM-instansseiksi tai yksinkertaisesti VM:iksi, ne edustavat kaikki samaa peruskäsitettä. Tämä yhdenmukainen terminologia Google Cloud -ekosysteemissä helpottaa viestintää ja varmistaa selkeyden, kun käsitellään ja hallitaan virtuaalikoneresursseja. (5.)

Jaettu VPC (Shared Virtual Private Cloud) on virtuaaliverkon ominaisuus, joka mahdollistaa useiden GCP-projektien jakaa yhden VPC-verkon, jota kutsutaan

isäntäprojektiksi (host project). Tämä mahdollistaa keskitetyn verkon hallinnan ja resurssien hallinnan samalla, kun yksittäiset projektit, joita kutsutaan palveluprojekteiksi (service project), voivat luoda ja käyttää omia resurssejaan jaettuun verkkoon. Jaetun verkon avulla voidaan yksinkertaistaa verkon hallintaa, parantaa tietoturvaa ja virtaviivaistaa resurssien jakamista projektien välillä, mikä tekee siitä tehokkaan ratkaisun organisaatioille, joilla on monimutkaiset pilvi-infrastruktuuritarpeet. (6.)

Asset Inventory

Isossa organisaatiossa on mahdollisesti satoja työtekijöitä ja projekteja. Projektit sisältävät lukuisia erilaisia palveluja, ja työntekijöillä on rooliinsa sopivat käyttöoikeudet eri tasolla. Työtekijällä voi olla esimerkiksi oikeudet nähdä lista projektin sisältämistä kaikista resursseista mutta oikeus hallita vain tiettyjä projektin resursseja.

Muun muassa pilvialustan ja järjestelmien ylläpidon, oikeintoimivuuden ja tietoturvallisuuden näkökulmasta on elintärkeää, että alusta tarjoaa tehokkaan analysointityökalun, joka mahdollistaa kokonaisvaltaisen näkymän ympäristössä käytössä oleviin resursseihin ja käyttöoikeuksiin keskitetysti yhdestä paikasta.

Turvallisuuden varmistamisen ja tehokkuuden maksimoinnin näkökulmasta on elintärkeää, että ylläpitäjät saavat reaaliaikaisia ilmoituksia, kun ympäristössä tapahtuu kriittisiä ja/tai ei-toivottuja muutoksia, kuten esimerkiksi käyttöoikeuksiin ja organisaation käytäntöihin tehdyt muutokset. On huomioitava, että tällaiset muutokset eivät aina tapahdu pahaa tarkoittavien toimesta, vaan ne voivat aiheutua esimerkiksi työntekijän inhimillisestä virheestä. Näissä tilanteissa ennakointi ja nopea reagointi ovat ratkaisevassa asemassa ympäristön ja palveluiden toimivuuden sekä turvallisuuden takaamisessa.

Usein organisaation eri yksiköillä on tarvetta viedä tiedot muualle, esimerkiksi kolmannen osapuolen palveluihin tai omiin palvelimiin analysoitavaksi ja/tai varastoitavaksi.

Asset Inventory on täysin hallittu (Fully managed) aikasarjatietokantaan perustuva palvelu, joka mahdollistaa pilviympäristössä olevien resurssien, kuten esimerkiksi virtuaalikoneiden, pilvifunktioiden ja Cloud Run -palveluinstanssien, metatietojen haun, reaaliaikaisten ilmoitusten lähettämisen, käyttöoikeuksien ja käytäntöjen analysoinnin sekä viennin esimerkiksi BigQuery-tietokantaan. Analysointia voidaan tehdä eri tasoilla, kuten organisaatio-, hakemisto- tai -projektitasolla. On huomioitava, että palvelu säilyttää historiatietoja kuluneelta 35 päivän ajalta ja pidempiaikainen tietojen säilytys edellyttää tietojen viennin ja tallentamisen johonkin toiseen tietovarastoon. (7; 8; 9.)

Palvelusta voidaan tilata reaaliaikaisia ilmoituksia resurssi- tai käytäntömuutoksista. Reaaliaikaisten ilmoitusten vastaanottaminen edellyttää uuden syötteen (feed) luomista ja sen tilaamista (subscribe). Ilmoitus lähetetään Pub/Sub-aiheeseen, ja syötteen konfiguraatiossa määritetään resurssin tyyppi ja mahdolliset ehdot rajoittamaan ilmoitukset koskemaan vain tietyntyyppisiä muutoksia. (10.)

Asset Inventoryn historiatiedot ovat erittäin arvokkaita. Esimerkiksi voidaan tarkastella, milloin tietylle käyttäjälle on myönnetty tai häneltä on evätty oikeuksia tiettyyn resurssiin. Aikaleimatietokanta mahdollistaa eri aikapisteiden vertailun, ja helppokäyttöinen käyttöliittymä antaa selkeän käsityksen tehdyistä muutoksista.

Asset Inventory mahdollistaa resurssien metadatan keräämisen myös ajonaikaisista ympäristöistä. Esimerkiksi ”Os inventory management” voidaan hyödyntää virtuaalikoneiden metadatan, kuten käyttöjärjestelmätietojen, keräämiseen ja tutkimiseen. Nämä käyttöjärjestelmän tiedot sisältävät tietoja, kuten isäntänimi, käyttöjärjestelmä ja ytimen versio. Kerätty metadata sisältää tietoa asennetuista käyttöjärjestelmän paketeista, saatavilla olevista päivityksistä, Windows-sovelluksista ja käyttöjärjestelmän haavoittuvuuksista. (7; 11.)

Asset Inventory on tehokas työkalu tilannekuvan luomiseen (Asset snapshot). Tilannekuva voidaan luoda tietystä projektista, hakemistosta tai organisaatiosta kuvaamaan niiden sisältämät resurssit tietyssä aikaleimassa. (7.)

Cloud Source Repository

Versionhallintajärjestelmät tarjoavat lukuisia etuja, jotka mullistavat ohjelmistokehityksen prosessin. Ne tarjoavat keskitetyn alustan kehittäjille muutosten seurantaan ja yhteistyöhön sekä koodinmuutosten historian, mikä vähentää tietojen menetyksen riskiä ja mahdollistaa helpon palautumisen aiempiin versioihin.

Git-versionhallintajärjestelmän kaltaiset järjestelmät edistävät tehokasta tiimityötä sallimalla useiden kehittäjien työskennellä samanaikaisesti samassa projektissa, ja ne mahdollistavat jatkuvan integroinnin sekä käyttöönoton (CI/CD) automatisoimalla ohjelmistokehityksen putken ja tehostamalla julkaisuprosessia.

Cloud Source Repository on täysin hallittu Git-versionhallintajärjestelmään perustuva pilvipalvelu, joka mahdollistaa ohjelmakoodin varastoinnin. Projekteissa voidaan luoda useita lähdekoodivarastoja ja yhdistää niitä GitHub- tai Bitbucket-varastoihin. Yhdistämisen jälkeen lähdekoodivarastot synkronisoidaan automaattisesti. (12.)

Lisäksi lähdekoodivarastot voidaan yhdistää laajasti kehittäjien käytössä oleviin kehitysympäristöihin (Integrated development environments IDE), kuten suositut Visual Studio, IntelliJ ja Android Studio (12).

On tilanteita, joissa kehittäjän on tarkasteltava lähdekoodia tai tehtävä pieniä muutoksia koodiin esimerkiksi haarakonfliktien (branch conflicts) aikana päätelaitteella, jossa kehitysympäristö tai koodieditori ei ole valmiiksi asennettu eikä ole järkevää asentaa. Lisäksi on tilanteita, jossa ei ole järkeä ensin ladata koko lähdekoodi omalle tietokoneelle, tehdä tarvittavat muutokset ja lähettää muutokset lähdekoodivarastoon.

Edellä mainituissa tilanteissa on hyvin tyypillistä, että kehittäjä haluaa mieluummin käyttää verkkoselainta tarvittavien toimenpiteiden suorittamiseen ja suorittaa toimenpiteet siellä, missä lähdekoodi on tallennettu. Cloud Source Repository mahdollistaa lähdekoodin selaamisen, ja tarvittavat koodimuutokset voidaan tehdä Cloud Shell Editorissa, joka tarjoaa sekä versionhallintalisäosion että integroidun terminaalin, jossa voidaan suorittaa tuttuja Git-komentoja, kuten esimerkiksi git add, git commit ja git push. Myös Cloud-terminaalia voidaan käyttää lähdekoodin Git-komentojen suorittamiseen. (12.)

Myös tietoturva on huomioitu: Cloud Source Repository estää git push-komentojen suorituksen, jos lähdekoodiin on sisällytetty turva-avaimia (esimerkkikoodi 1). Palvelu huomaa JSON-muotoiset palvelutilien tunnisteavaimet ja PEM-koodatut yksityiset avaimet. (12; 13.)

```
The push has been rejected because we detect that it contains a pri-
vate key.
Please check the following commands and confirm that it's intentional:

git show [COMMIT]

You can use `git rev-list --objects --all` to find the files.

To push these files, please run `git push -o nokeycheck`.
```

Esimerkkikoodi 1. Cloud Source Repository-palvelu estää turva-avainten lataamisen lähdekoodivarastoon ja ilmoittaa siitä käyttäjälle.

Turva-avainten havaitseminen lähdekoodista voidaan poistaa käytöstä (esimerkkikoodi 2) tai kytkeä päälle (esimerkkikoodi 3) terminaalissa käyttämällä ”gcloud” -komentorivityökalua (13).

```
gcloud source project-configs update --disable-pushblock
```

Esimerkkikoodi 2. Turva-avainten havaitsemisen poistaminen käytöstä gcloud-komentorivityökalulla.

```
gcloud source project-configs update --enable-pushblock
```

Esimerkkikoodi 3. Turva-avainten havaitsemisen päällekytkeminen gcloud-komentorivityökalulla (13).

Toisin kuin esimerkiksi GitHub-lähdekoodivarastoissa, Cloud Source Repository ei mahdollista haarojen (branch) suojausta eikä siten voi suojata ”master/main/head” -haaraa vahingossa tapahtuvilta muutosten lähettämistä. Sen sijaan käyttöoikeuksia voidaan hallita samalla tavalla kuin muissakin pilvialustan palveluiden tapauksissa käyttämällä IAM-työkalua. (14.)

On huomioitava, että Cloud Source Repository -palvelun lähdekoodivarastot ovat yksityisiä ja siten käyttöoikeuksien myöntäminen ”allAuthenticatedUsers” ja ”allUsers” käyttäjätyypeille ei ole sallittu (14).

Yhdistämällä Source Repository- ja Cloud Pub/Sub -palvelut voidaan luoda reaaliaikainen ja tehokas viestintäjärjestelmä, jossa projektia työstävät eri käyttäjät voivat saada ilmoituksia reaaliajassa lähdekoodivaraston muutoksista, kuten uuden koodivaraston luonnista, koodin lähettämisestä (push) ja koodivaraston poistamisesta. Perinteiset käyttäjien ilmoitustavat, kuten sähköposti-ilmoitukset tai manuaaliset tarkastukset, voivat aiheuttaa viivettä. Hyödyntämällä näitä organisaatiot voivat ilmoittaa suurelle käyttäjämäärälle samanaikaisesti olipa kyseessä pieni kehitystiimi tai globaali organisaatio, jossa on tuhansia kehittäjiä. Lisäksi lähdekoodivarastot luovat pilvilokeja, mikä helpottaa palvelun käytön seurantaa ja ongelmien nopeaa ratkaisua. (15; 12.)

Cloud Build

Cloud Build -palvelu on niin sanottu palvelimeton (serverless) ja täysin hallittu jatkuvan integroinnin ja jatkuvan toimituksen (CI/CD) alusta. Se tarjoaa yhteyden yksityisverkkoihin ja helpottaa erilaisten ohjelmistotyyppien, kuten esimerkiksi Node.js- ja Go-ohjelmistojen rakentamista, testaamista ja käyttöönottoa. Käyttäjät hyötyvät päivittäisestä 120 koontiminuutin (build minutes) kiintiöstä ilman kustannuksia.

Lisäksi Cloud Build integroituu saumattomasti yritystason lähdekoodinhallintajärjestelmiin, kuten esimerkiksi GitLab Enterprise. Valittavana on 15 erilaista koneen tyyppiä, joiden avulla voidaan käsitellä samanaikaisia ohjelmiston kokoamisia (software builds) tehokkaasti. Se tukee käyttöönottoa useissa

ympäristöissä, mukaan lukien Kubernetes ja virtuaalikoneet (VM), ja yksityisessä verkossa, jolla on pääsy CI/CD-työnkulkuun (CI/CD workflow).

Ydintoimintojensa lisäksi Cloud Build parantaa turvallisuutta tarjoamalla konttikuvien (container images) haavoittuvuuksien skannausta, mikä mahdollistaa tarkastusten ja toimitusten hallinnan alkuperäisen jäljitettävyyden kautta sekä tukemalla SLSA-tason 3 suojausta ohjelmistotoimitusketjuhyökkäyksiä vastaan. (16.)

Palvelu mahdollistaa lähdekoodin tuomisen eri lähdekoodivarastoista, kuten GitHubista tai Bitbucketista. Se suorittaa koontiprosessin (build process) ja luo artefakteja, mukaan lukien Docker-kontit (Docker containers) ja Java-arkistot. Jokainen kokoaminen (build) jaetaan sarjaan vaiheita, joista jokainen suoritetaan Docker-kontissa, joka on yhteydessä paikalliseen Docker-verkkoon nimeltä "cloudbuild". Näillä koontivaiheilla (build steps) on joustavuus suorittaa mitä tahansa toimintoja, jotka ovat mahdollisia konttiympäristössä.

Cloud Build tarjoaa kirjaston tuettuja koontivaiheita, ja käyttäjillä on mahdollisuus luoda omia. Lisäksi se tarjoaa valikoiman avoimen lähdekoodin koontivaiheita käyttäjäyhteisöstään. Koontivaiheiden suorittaminen on samanlaista kuin komentojen suorittaminen skriptissä.

Koontikonfiguraatiotiedosto määrittelee suoritettavat tehtävät, joilla ohjataan Coud Buildin toimia. Tähän konfiguraatioon voi sisältyä riippuvuuksien hakeminen, staattisten analyysien suorittaminen, testien suorittaminen ja artefaktien luominen käyttäen työkaluja kuten Docker, Gradle tai Maven. Se tarjoaa mahdollisuuden liittää jokaisen vaiheen kontti (container) paikalliseen Docker-verkkoon (Cloudbuild-verkko) mikä helpottaa kommunikointia ja tietojen jakamista koontivaiheiden välillä. Cloud Build tukee myös Docker Hub -kuvien käyttöä.

Yksi Cloud Buildin huomattavista ominaisuuksista on sen saumaton integrointi CI/CD-työnkulkuun. Käyttäjät voivat helposti luoda automatisoituja työnkulkuja asettamalla laukaisimia (triggers), jotka reagoivat koodin muutoksiin. Nämä

laukaisimet voidaan integroida kolmannen osapuolen lähdekoodivarastoihin, kuten GitHub ja Bitbucket, mikä helpottaa kehitys- ja julkaisuprosessia. (17.)

Cloud Function

Pilvifunktioista on tarjolla kaksi versiota, ensimmäinen ja toinen sukupolvi (1st gen, 2nd gen). Niin kutsuttu palvelimeton palvelu tarjoaa täysin hallitun suoritusympäristön, joka tukee useita ohjelmointikieliä, kuten Ruby, PHP, C++ ja Node.js. Toisin kuin ensimmäisen sukupolven, toisen sukupolven ympäristö on rakennettu Cloud Run- ja Eventarc-pilvipalveluiden päälle ja mahdollistaa eri pilvipalveluiden yhdistämisen ja toiminnallisuuden laajentamisen. Esimerkiksi kustannusten hallinta on mahdollista saavuttaa yhdistämällä Billing-, Pub/Sub- ja Cloud Function -palvelut ja hyödyntämällä Billing-palvelun kirjastoa funktiossa.

Funktiot voidaan kutsua tai laukaista suoraan sen verkko-osoitetta käyttäen tai kuuntelemalla ja reagoimalla erilaisiin tapahtumiin, joita ovat esimerkiksi uudet Pub/Sub-viestit ja -lokit, lisäys Firestore-tietokantaan tai uuden virtuaalikoneen luonti. On huomioitava, että nämä tapahtumat tapahtuvat riippumatta siitä, reagoidaanko niihin vai ei.

Pilvifunktiot sisältävät palvelutilin tunnistetiedot, ja ne ovat siten saumattomasti tunnistautuneita useimpiin palveluihin, kuten esimerkiksi Cloud Visioniin ja moniin muihin. Lisäksi lukuisat Google Cloud -asiakaskirjastot, kuten ”asset” ja ”resource-manager”, tukevat pilvifunktioita, mikä edelleen yksinkertaistaa näitä integrointeja. Esimerkiksi ”resource-manager”-kirjasto mahdollistaa projektien luetteloimisen. (18; 19; 20.)

Toisen sukupolven funktiot tuovat useita parannuksia. Niitä ovat esimerkiksi pidemmät pyyntökäsittelyajat suurten tietovirtojen käsittelyyn, tuki suuremmille instanssiko’oille, jotta voidaan hallita muisti-intensiivisiä työkuormia, luontainen integraatio Eventarc-laukaisijoiden kanssa laajemmalle tapahtumalähdetuen kattavuudelle sekä johdonmukainen tuki teollisuusstandardin CloudEvents-tapahtumille kaikissa ohjelmointikielissä (all language runtimes). (21.)

Cloud Pub/Sub

Pub/Sub toimii asynkronisena ja skaalautuvana viestinvälityspalveluna, joka erottaa viestin tuottajat viestin kuluttajista. Tämä arkkitehtuuri mahdollistaa palveluiden vuorovaikutuksen asynkronisesti, tyypillisesti vastausaikojen ollessa yleensä noin 100 millisekunnin luokkaa.

Pub/Subin monipuolinen luonne soveltuu erilaisiin sovelluksiin, kuten tietovirta-analytiikkaan (streaming analytics), tietojen integrointiputkiin (data integration pipelines) ja tapahtumapohjaisiin (event-driven) skenaarioihin.

Pub/Sub helpottaa järjestelmien luomista, jotka koostuvat tapahtumien tuottajista (julkaisijat/publishers) ja kuluttajista (tilaajat/subscribers). Toisin kuin synkroniset etäkutsut (RPC), julkaisijat lähettävät tapahtumia Pub/Sub-palveluun välittämättä itse tapahtumien ajoituksesta tai käsittelystä. Pub/Sub huolehtii sitten näiden tapahtumien toimittamisesta kaikille asiaankuuluville palveluille. Tämä asynkroninen lähestymistapa parantaa järjestelmän joustavuutta ja luotettavuutta verrattuna perinteiseen RPC-pohjaiseen viestintään.

Yleisiä käyttötapauksia Pub/Subille ovat käyttäjien vuorovaikutuksen vastaanottaminen, reaaliaikainen tapahtumien jakelu, tietojen replikointi tietokantojen välillä, rinnakkaisprosessointi, yritystason tapahtumaväylät ja tietovirtojen kerääminen eri lähteistä, kuten sovelluksista, palveluista tai IoT-laitteista. Sitä voidaan myös käyttää hajautettujen välimuistien päivittämiseen ja kuormantasapainon saavuttamiseen.

Palvelu tarjoaa kaksi vaihtoehtoa, standardin Pub/Sub-palvelun ja kustannustehokkaamman Pub/Sub Lite -palvelun. Edellistä suositaan sen luotettavuuden, laajan integraatiomahdollisuuksien ja automaattisen kapasiteetinhallinnan vuoksi, kun taas jälkimmäinen, Pub/Sub Lite, tarjoaa edullisemman vaihtoehdon joillakin luotettavuuden ja tallennuksen hallinnan kompromisseilla. Lisäksi Pub/Sub integroituu saumattomasti eri GCP-palveluihin ja mahdollistaa erilaisia toimintoja, kuten esimerkiksi tietovirran käsittely, valvonta ja hälyttäminen. (22.)

Eventarc

Perinteisesti ohjelmiston taustajärjestelmä (back-end system) rakennettiin yhtenä isona kokonaisuutena, joka sisälsi ohjelmiston tai järjestelmän kaikki toiminnallisuudet eli liiketoimintalogiikan (business logic). Esimerkiksi verkkokauppasovelluksen taustajärjestelmän eri toimintojen koodilogiikka, kuten tuotteiden listaaminen, käyttäjän rekisteröinti ja ostoskorin toiminnot, olivat kaikki samassa paikassa. Tämä niin kutsuttu monoliittijärjestelmä (monoliittiarkkitehtuuri) on tehokas tapa pienen sovelluksen kehittämiseen, mutta ohjelmiston toiminnallisuuksien kasvaessa se osoittautuu tehottomaksi. Esimerkiksi ohjelmiston uuden version julkaisu on hidasta, sillä pienikin muutos lähdekoodiin vaatii koko ohjelmiston uudelleenrakentamista.

Mikropalveluarkkitehtuuria soveltavat järjestelmät, joita kutsutaan myös mikropalveluiksi (microservices), ovat itsenäisesti toimivia taustajärjestelmäpalveluja (back-end service). Kullakin näistä palveluista on oma erillinen liiketoimintalogiikkansa ja tietokantansa, ja ne palvelevat tiettyä tarkoitusta. Päivitykset, testaus, käyttöönotto ja skaalautuvuus tapahtuvat kunkin yksittäisen palvelun sisällä. Mikropalvelut hajauttavat isot järjestelmät erillisiin, eristettyihin kokonaisuuksiin. Sen sijaan, että monimutkaisuutta vähennettäisiin, mikropalvelut tekevät siitä näkyvämpää ja hallittavampaa pilkkomalla toiminnallisuudet pienempiin toimintoihin, jotka toimivat itsenäisesti toisistaan riippumatta ja yhdessä muodostaen kokonaisjärjestelmän. Esimerkiksi sen sijaan, että verkkokauppasovelluksen taustajärjestelmä sisältäisi sekä tuoteluettelon että ostoskorin liiketoimintalogiikan, luodaan jokaiselle oma mikropalvelunsa. Perinteisesti tässä mallissa mikropalvelut ovat keskustelleet keskenään ja vuorovaikuttaneet REST-rajapinnan kautta, jota kutsutaan synkroniseksi viestinnäksi tai tiedonsiirroksi. Lisää tietoa tämän ongelmista löytyy lähteen 23 artikkelista. Yksi merkittävimmistä mahdollisuuksista on, että vaikka useat mikropalvelut yhdessä edistävät yhteistä päämäärää, niitä voidaan kehittää eri teknologioilla: esimerkiksi tuoteluettelon mikropalvelu rakennetaan Pythonilla ja ostoskori kehitetään Node.js:llä. (23.)

Tapahtumavetoisessa mallissa (event-driven pattern) mikropalvelut eivät ole suorassa yhteydessä toisiinsa, vaan ne kommunikoivat välittäjäpalvelun kautta. Tätä kommunikointimallia kutsutaan asynkroniseksi viestinnäksi, jossa eri mikropalvelut ovat täysin irrallaan toisistaan mutta pystyvät kommunikoimaan tai saamaan tietoa toisiltaan. Tässä mallissa mikropalvelut eivät lähetä suoria kutsuja toisilleen. Sen sijaan kutsut eli tapahtumat, jotka mahdollisesti sisältävät viestin tai ilmoituksen, lähetetään välittäjäpalvelulle. Muut mikropalvelut reagoivat tapahtumaan, mikäli ovat kiinnostuneita kyseisestä tapahtumasta tai tapahtumatyypistä. Mallin kolme komponenttia ovat tuottaja, kuluttaja ja reititin. (24.)

Eventarc on täysin hallittu välittäjäpalvelu, joka mahdollistaa tapahtumavetoisen arkkitehtuurimallin, samankaltaisen kuin edellä mainittu. Eventarc vastaanottaa tapahtumia pilvipalvelun sisäisistä ja luotetuista kolmannen osapuolen palveluista sekä valvontalokeista (audit log) ja reitittää niitä määritettyyn kohteeseen Pub/Sub-push-tilauksen kautta. Näitä kohteita ovat toisen sukupolven (2nd gen) pilvifunktiot Cloud Run, GKE ja Workflows. (25.)

Eventarc pystyy vastaanottamaan tapahtumia yli 130 alustan palvelusta, kuten Cloud Billing -palvelusta. Esimerkiksi kun projektista poistetaan laskutustili, tapahtuma voidaan välittää pilvifunktiolle, jossa suoritetaan tarpeellisia jatkotoimenpiteitä. On kuitenkin huomioitava, että vaikka tapahtuma reititetään kohteelle sen sisällön perusteella, itse tapahtuma ei sisällä reititystietoja ja tapahtumat tapahtuvat (pilvialusta luo näitä tapahtumia) riippumatta siitä, reagoidaanko niihin vai ei. Tapahtuma tilataan luomalla laukaisija (trigger), ja se välitetään HTTP-pyynnöllä CloudEvents-spesifikaation mukaisessa muodossa. Tapahtuman varsinainen tieto (payload) on joko JSON- tai Protobuf-muotoinen; JSON-muotoisen version koko on suurempi kuin Protobufin. (25.)

Laukaisijat voivat olla joko paikallisia tai globaaleja. Globaalin triggerin aiheuttamien suorituskyky- ja tietosuojaongelmien välttämiseksi triggerit on luotava samassa sijainnissa kuin lähdekohteen sijainti, joko paikallis- tai globaalitasolla. Tarkastelulokeja (Cloud Audit Logs) käyttävät tapahtumat toimitetaan alle minuutissa (huom. vaikka tarkastelulokien laukaisin luodaankin välittömästi, sen

lisäämiseen (propagate) ja tapahtumien suodattamiseen (filter) voi kulua enintään kaksi minuuttia). Pub/Subin avulla laukaistut tapahtumat toimitetaan sekunneissa. (25.)

Palvelua voidaan helposti integroida omiin järjestelmiin tai muihin palveluihin, jotka ovat yhteensopivia Pub/Sub-palvelun kanssa, ja tilata sieltä ilmoituksia. Järjestelmä voi julkaista pubsub-aiheeseen ja luodaan triggeri, joka tilaa kyseistä aihetta. Tällä tavoin voidaan laajentaa Eventarcin lähteiden määrää.

Cloud Billing

Pilvialustan palveluiden ja Google Maps -alustan kulut maksetaan "Cloud Billing" -tililtä. Tili on pilvialustatason resurssi, ja sen käyttöoikeudet määritetään IAM-rooleilla pilvialustan puolella, joka on yhdistetty Google-maksuprofiiliin (Google Payment Profile). Maksuprofiili on Google-tason resurssi, jota hallinnoidaan osoitteessa payments.google.com. Maksuprofiili säilyttää muun muassa yksityis- ja yritysasiakkaiden tiedot, kuten organisaation nimen, maksutavat, kuitit, maksuhistorian, profiilin käyttäjät ja tilaukset. Cloud Billing -tili pitää kirjaa kuluista yksityiskohtaisesti, määrittelee budjetteja ja budjettihälytyksiä, tuottaa raportteja ja laskuttaa tilikohtaisesti sekä määrittelee maksajan. On huomioitava, että laskutustili käyttää vain yhtä määritettyä valuuttaa ja että projekti on linkitettävä aktiiviseen laskutustiliin, jotta voidaan käyttää pilvialustan palveluja myös tilanteessa, jossa käytetään ilmaisia pilvipalveluja. (26; 27.)

Cloud Billing -palvelu on useista laskutustyökaluista koostuva tehokas hallintatyökalupakki kulujen valvontaan, raportointiin, ennakoimiseen ja optimointiin sekä budjettisuunnitteluun. Laskutustilin keräämää dataa voidaan viedä esimerkiksi BigQuery-tietokantaan analysoitavaksi, josta se voidaan edelleen viedä Looker Studioon tietojen visualisoimiseksi. Google suosittelee tietojen viemistä BigQueryiin mahdollisimman aikaisessa vaiheessa, jotta tiedot heijastavat pilvialustan käyttöä alusta alkaen. (26.)

Budjetit mahdollistavat pilvialustan kustannusten valvonnan ja auttavat välttämään yllätyksiä. Ne ovat pilvilaskutustilikohtaisia ja voidaan määrittää

seuraamaan kuluja eri tasoilla (valvontapiiri), kuten esimerkiksi koko organisaation resurssien tai tietyn projektin resurssien kuluja. Budjetille asetetaan lisäksi budjetin valvontapiiriin kuuluva rahallinen summa, joka voi olla joko tietty summa, kuten 200 euroa, tai viime kuukauden kuluja vastaava summa. (28.)

Lisäksi budjetille voidaan asettaa käyttäjän määrittämiin kynnysarvoihin perustuvia hälytyksiä. Hälytykset laukeavat ehtojen täyttyessä, ja hälytysilmoitus lähetetään sähköpostiviestinä tai Pub/Sub-viestinä. Ilmoitus sisältää tärkeää tietoa, kuten budjetin varsinaisen summan ja nykyisen saldon. Kynnysarvot mahdollistavat ilmoitusten lähettämisen, kun todellinen ennustettu kulutus ylittää prosenttiosuuden budjetista tai tietyn summan. On huomioitava, että budjetin ja sen kynnysarvojen määrittäminen eivät sellaisinaan estä lisäkulujen syntymistä. (28.)

Budjetin hälytysilmoituksen lähettäminen Pub/Sub-palveluun ja sieltä edelleen pilvifunktioon avaa uusia mahdollisuuksia kustannusten hallintaan. Viestin sisältämän datan analysointi pilvifunktiossa voi parhaimmillaan auttaa välttymään suurilta taloudellisilta vahingoilta. Viestistä löytyvän tiedon, kuten asetetun budjettisumman ja nykyisen saldon, perusteella voidaan tehdä tärkeitä ja kriittisiä päätöksiä sekä suorittaa toimenpiteitä. On tilanteita, joissa halutaan ehdottomasti pysyä asetetun budjetin rajoissa. Viestin sisällön perusteella voidaan irrottaa laskutustili projektista, mikä johtaa projektin sammuttamiseen. Tällöin projektin palveluresurssit ja siten kustannusten syntyminen pysäytetään. Laskutustilin irrottaminen projektista tarkoittaa, että projektin palveluresurssien data saatetaan mahdollisesti menettää. Googlen mukaan on mahdollista palauttaa em. dataa 30 päivän sisällä projektin sammuttamisesta, mutta datan palauttaminen ei kuitenkaan ole taattu. Datan menetyksen välttämiseksi asiakaskirjastoja hyödyntämällä pilvifunktiossa voidaan ensin tarkistaa ja listata projektin käytössä olevat palveluresurssit ja mahdollisesti sammuttaa kalliit resurssit tai varmuuskopioida resurssien dataa ja tallentaa muualle ennen laskutustilin irrottamista. (29; 30; 31.)

GCP tarjoaa hinnoittelulaskimen, joka on käyttäjäystävällinen työkalu GCP-palveluiden ja resurssien kustannusten arvioimiseen. Työkalu auttaa käyttäjiä tekemään tietoisia päätöksiä pilvi-infrastruktuurin kustannuksistaan. (32.)

Security Command Center

Security Command Center (SCC) on kattava tietoturvan hallinta- ja valvontaratkaisu. Se toimii keskitettynä keskuksena, jonka avulla organisaatiot voivat saada näkyvyyttä ja hallintaa pilvi-infrastruktuurinsa tietoturvaan. SCC tarjoaa reaaliaikaisia näkymiä mahdollisiin tietoturvariskeihin ja haavoittuvuuksiin GCP-resursseissa, kuten esimerkiksi Google Kubernetes Engine, Cloud Storage ja Compute Engine.

SCC sisältää tärkeitä ominaisuuksia, kuten omaisuuden inventoinnin (asset inventory), haavoittuvuusanalyysin ja uhkien havaitsemisen, mikä auttaa organisaatioita tunnistamaan ja reagoimaan tietoturvahaittoihin proaktiivisesti. SCC integroituu Google Cloudin tietoturvapalveluihin, kuten Cloud Security Scanner ja Cloud Armor, mikä mahdollistaa käyttäjien tietoturvatoimintojen virtaviivaistamisen. SCC tukee myös mukautettuja sääntöjä ja käytäntöjä, jotka mahdollistavat tietoturvan sovittamisen organisaatioiden erityistarpeisiin. Käyttäjäystävällisen käyttöliittymän ja automatisoitujen hälytysten avulla Security Command Center antaa yrityksille mahdollisuuden parantaa pilviratkaisujaan, suojata herkkää tietoa ja varmistaa alan standardien määräysten noudattamista, ja auttaa siten säilyttämään asiakkaiden ja kumppaneiden luottamuksen pilviympäristössä. (33.)

Cloud Logging

Pilvipalvelut, käyttäjät, sovellukset ja järjestelmät tuottavat erilaisia lokeja, jotka kuvaavat tiettyä tapahtumaa tai järjestelmän tilaa: esimerkiksi pilvifunktion suorituksen alkaminen ja lopettaminen kirjataan lokeihin automaattisesti. Cloud Logging -palvelu on reaaliaikainen lokien hallintajärjestelmä, joka on varustettu tallentamis-, haku-, analysointi- ja valvonta-toiminnallisuuksilla. Lokit kerätään pilvialustan palveluista ja vaihtoehtoisesti myös ulkoisista lähteistä. (34.)

Lokit on jaettu viiteen eri pääkategoriaan: pilvialustan palvelujen lokit, jotka muun muassa tarjoavat tärkeitä tietoja virheiden jäljittämiseen virhetilanteissa; komponenttilokit, jotka ovat peräisin pilvipalveluissa suoritettavista ohjelmistoista; kahdentyyppiset suojauslokit, jotka vastaavat kysymykseen ”kuka teki mitä, missä ja milloin”; tarkastuslokit (audit logs), jotka tarjoavat tietoa tietosuojan ja tietoturvan kannalta tärkeistä hallinnollisista toiminnoista ja pääsystä pilviresursseihin; pääsyn läpinäkyvyys -lokit (access transparency) lokit, jotka kuvaavat Googlen henkilöstön suorittamia toimia; käyttäjän kirjoittamat lokit, jotka ovat peräisin sovelluksista ja palveluista sekä monipilvi- ja hybridipilvilokit, jotka ovat peräisin paikallisilta infrastruktuureilta (on-premises infrastructures) ja muilta pilvipalveluilta. (34.)

Lokeja voidaan tarkastella ja analysoida Google Cloud -konsolissa joko Logs Explorer- tai Log Analytics -käyttöliittymissä. Google suosittelee Log Explorerin käyttämistä vianetsintään ja suorituskyvyn analysointiin ja Log Analyticsin käyttämistä, kun suoritetaan kokoavia toimenpiteitä lokitiedoilla, kuten keskimääräisen viiveen laskemista ajan kuluessa lähetettyihin http-pyyntöihin tiettyyn verkko-osoitteeseen. Molemmat käyttöliittymät mahdollistavat lokikyselyjen tekemisen ja selaamisen, mutta niillä on erilaisia kyvykkyyksiä ja ne käyttävät eri kyselykieliä. Log Analytics -kyselyt tehdään SQL-kielellä, mikä helpottaa lokitietojen ymmärtämistä. Lisäksi lokikyselyt voidaan suorittaa ohjelmallisesti rajapintaa (API) tai komentorivityökalua (CLI) hyödyntämällä. (34.)

Lokien kerääjät (log sinks) mahdollistavat lokimerkintöjen ohjaamisen erilaisiin kohteisiin, kuten lokisäiliöihin, Cloud Storage -säiliöihin, BigQuery-tietokantaan, Pub/Subiin tai ulkoisiin palveluihin Pub/Sub-palvelun avulla. Tämä joustavuus mahdollistaa organisaatioiden mukautetun lokidatan hallinnan tiettyihin käyttötarkoituksiin, olipa kyseessä pitkäaikainen tallennus, reaaliaikainen analyysi tai integrointi kolmannen osapuolen työkalujen kanssa. On huomioitava, että lokien tallennuksella lokisäiliössä on säilytysaika. (35; 36.)

Monitorointi

Cloud Monitoring on olennainen valvontatyökalu, joka integroituu saumattomasti eri GCP-palveluihin. Palvelu kerää automaattisesti suorituskykytietoja ja metriikoita jopa kolmannen osapuolen sovelluksista. Tämä palvelu tarjoaa tietojen visualisointi- ja analyysityökalut palveluiden kuormituksen, verkkosivuston reagointikyvyn ja kokonaissuorituskyvyn arviointiin.

Palvelu tarjoaa reaaliaikaisia näkymiä kaavioiden ja kojetaulujen (dashboard) kautta näyttämään GCP-palveluiden, agenttien ja käyttäjän määrittelemiä metriikoita. Se mahdollistaa myös valvontatarkistukset verkkosivuston reagointikyvyn seuraamiseksi ja mukautettujen hälytyskäytäntöjen määrityksen.

Lisäksi palvelu mahdollistaa kollektiivisen resurssienhallinnan dynaamisten resurssiryhmien avulla ja helpottaa useiden projektien metriikoiden seurantaa keskitetysti yhdessä projektissa. (37.)

Tiedon salaaminen

GCP käyttää monitasoista lähestymistapaa tietojen salaukseen varmistaakseen tietojen turvallisuuden ja luottamuksellisuuden. Tietojen lepotilassa (data at rest) GCP käyttää Google-hallinnoituja avaimia salatakseen tiedot eri tallennuspalveluissaan, kuten Cloud Storage ja Bigtable. Näitä avaimia hallitaan Googlen Key Management Service (KMS) -palvelulla, joka on suunniteltu suojaamaan luvattomalta pääsyltä. Lisäksi organisaatiot voivat halutessaan tuoda omat salausavaimensa lisäkontrollin saamiseksi. (38.)

Tietojen siirrossa (data in transit) GCP käyttää vahvoja salausprotokollia, kuten TLS (Transport Layer Security) turvatakseen tiedon siirron käyttäjien ja GCP-palveluiden tai eri GCP-palveluiden välillä verkossa. Tämä varmistaa, että tiedot pysyvät suojattuina siirron aikana. (39.)

Infrastruktuuri koodina

GCP:llä on Terraform-tarjoaja (Terraform provider), joka on suunniteltu erityisesti vuorovaikuttamaan ja hallitsemaan pilvialustan resursseja. Tämä

Terraform-tarjoaja mahdollistaa infrastruktuuriresurssien määrittelyn, kuten esimerkiksi projektien, virtuaalikoneiden, tietokantojen ja virtuaaliverkkojen luomisen. (40.)

Arkaluonteisen tiedon suojaaminen

Cloud Data Loss Prevention (DLP) -palvelu on tehokas ratkaisu, jonka tarkoituksena on suojata arkaluontoista tietoa ja pienentää tietovuotojen riskiä GCP-ympäristössä. Palvelu tarjoaa kattavan joukon työkaluja ja toimintoja, joiden avulla organisaatiot voivat tunnistaa, luokitella ja suojella arkaluontoista tietoa. DLP:n avulla voidaan määritellä käytäntöjä, jotka määrittävät, mikä katsotaan arkaluontoiseksi tiedoksi, kuten esimerkiksi henkilötunnukset ja luottokorttinumerot. Näitä voidaan sitten soveltaa eri tietolähteisiin, kuten esimerkiksi tietokantoihin varmistamaan, että arkaluontoinen tieto tunnistetaan ja käsitellään asianmukaisesti. (41.)

DLP palvelu tarjoaa sekä valmiita havainnoijia että mahdollisuuden luoda mukautettuja havainnoijia, jotka tunnistavat organisaation omat ainutlaatuiset kuvioinnit (patterns) ja mallit (models). Lisäksi DLP integroituu muihin alustan palveluihin ja kolmannen osapuolen työkaluihin, mikä tekee siitä tehokkaan valinnan organisaatioille, jotka haluavat parantaa ja tehostaa tietoturvaansa. (41.)

Tarkastuspalvelu mahdollistaa syvällisen yksittäisen resurssin tarkastelun herkkien tietojen havaitsemiseksi. Käyttäjä määrittelee etsittävän tietotyypin, ja tarkastuspalvelu luo raportin kaikista kyseisen tietotyypin vastaavista tietoilmestymistä. Raportissa ilmoitetaan esimerkiksi, kuinka monta luottokorttinumeroa löytyy Cloud Storage -säiliössä ja missä tarkalleen ottaen nämä tietoilmestymät sijaitsevat. (41.)

3 Metropolian käyttöönottoprosessi ja pilviperusta

Googlen pilvialustan käyttöönottoprosessi ja pilviperusta toteutettiin kahdessa vaiheessa. Ensimmäisen vaiheen toimenpiteet suoritettiin Metropolian

Workspace-konsolissa ja pilvialustan konsolissa Computas AS:n tiimin ohjaamana. Näitä toimenpiteitä kuvataan tarkemmin seuraavaksi. Toisessa vaiheessa Gapps Oy toteutti pilviperustan. Analysoimalla pilviympäristön konsolia ja lähdekoodia opinnäytetyön tekijä totesi, että toimenpiteet voidaan jakaa kolmeen osaan: skriptin suoritus, konsolissa suoritetut toimenpiteet ja Terraform-koodin suoritus. Vaiheet kuvataan tarkemmin seuraavassa.

Vaihe 1

Ensimmäisen vaiheen kaikki toimenpiteet suoritettiin etäyhteyksien välityksellä noin kaksi tuntia kestäneen palaverin aikana. Workspace-konsolissa luotiin ja konfiguroitiin kolme käyttäjäryhmää, ja myöhemmin ryhmille myönnettiin tarvittavat pääsynoikeudet Cloud-konsolissa. Workspace-konsolissa todettiin, että pilvialusta oli aiemmin otettu käyttöön ja se oli avoin kaikille Metropolian organisaation käyttäjille. Lisäksi Cloud-konsolissa todettiin, että kaikille Metropolian organisaation käyttäjille oli aiemmin myönnetty Project Creator- ja Billing Account Creator -roolit organisaatiotasolla, ja todettiin myös, että pilvialustassa oli aiemmin luotu projekteja.

Cloud-konsolissa myönnettiin kaikille Metropolian käyttäjille Organization Viewer- ja Folder Viewer -roolit. Ryhmälle gcp-organization-admins myönnettiin Organization Administrator-, Organization Policy Administrator-, Folder Admin- ja Security Center Admin -roolit organisaatiotasolla.

Cloud-konsolissa otettiin käyttöön Security Command Center (SCC) -palvelun ilmainen versio (free tier) ja palvelun automaattisesti luodulle palvelutilille myönnettiin Cloud Functions Service Agent-, Security Center Service Agent- ja Service Usage Admin -roolit organisaatiotasolla. Lisäksi otettiin käyttöön organisaatiotason käytäntö ”skip default network creation”. Tämä on tärkeä käytäntö, sillä SCC valittaa, jos liian monta verkkoa havaitaan. Oletuspalomuurisäännöt avaisivat esim. RDP:n ja SHH:n koko internetille ja sallivat myös aliverkkojen välisen kommunikaation, mikä on turvallisuusriski.

Vaihe 2

Gapps toteutti tämän teknisen vaiheen luomalla niin sanotun pilviperustan, mikä johti nykyiseen pilviympäristötilanteeseen. Metropolia ei osallistunut varsinaiseen tekniseen toteutukseen. Seuraavaksi kuvataan tässä vaiheessa suoritettujen toimenpiteiden seuraukset, jotka perustuvat konsolin ja lähdekoodin analyysituloksiin.

3.1 Resurssihierarkia

Metropolian organisaation alla on kaksi pääkansiota: Student Sandbox ja Metropolia. Student Sandbox -kansiossa käyttäjät voivat kokeilla pilvialustan palveluja vapaammin. Metropolia-kansio (jota joskus kutsutaan laskeutumisalueeksi) on niin sanottu hallittu alue, jossa käyttäjät oletuksena näkevät ainoastaan ympäristön.

Metropolia-organisaatio

Metropolia-organisaatio on pilviympäristön juurisolmu, joka on avainasemassa organisaation laajuisissa käytännöissä. Tällä tasolla on otettu käyttöön koko organisaatiota koskevia käytäntöjä, jotka heijastuvat kaikkiin aliresursseihin. Kaikilla organisaation käyttäjillä on organisaatiotason Selain (Browser)- ja Laskutustilin luonti (Billing Account Creator) -roolit.

Student Sandbox -kansio

Student Sandbox -kansio on tarkoitettu pääasiassa opiskelijoille, joilla on tarvittavat oikeudet projektien ja laskutustilien luomiseen sekä GCP:n ilmaisten krediittien hyödyntämiseen. Student Sandbox -kansion luominen oli välttämätöntä käyttöönottoprosessin aikana, jotta aiemmin luodut projektit säilytettäisiin. Organisaatiotason käytännöt ovat voimassa myös tässä kansiossa. Aiemmin luodut projektit on siirretty tähän kansioon.

Kaikilla organisaation käyttäjillä on tässä kansiossa Project Creator -rooli lisättynä organisaatiotasolta perittyihin Browser- ja Billing Account -rooleihin. Näiden

roolien yhdistelmät mahdollistavat ympäristön kokeilun ja käytön esimerkiksi opiskelijoille.

Metropolia-kansio (laskeutumisalue)

Metropolia-kansio on niin sanottu hallittu alue, jossa käyttöoikeudet on rajattu siten, että oletuksena käyttäjillä ei ole muita oikeuksia kuin ne, jotka on myönnetty organisaatiotasolla. Alikansioiden tarkoitus on erottaa ja järjestää eri yksiköiden, osaamisalueiden, tutkintojen pääaineiden ja yritysten projektit.

Tämän kansion alla oleva Common-kansio sisältää kolme projektia, joilla on laajempi käyttötarkoitus, kuten laskutusviennin, infrastruktuurikoodin ja keskitetyn seurannan projektit, jotka kuvataan seuraavassa tarkemmin. Käyttöönottoprosessin aikana oli epäselvää, mitkä yksiköt ottaisivat alustan käyttöön. Tästä syystä ICT-osaamisalueelle luotiin Metropolia-ICT-niminen kansio ja alikansio nimeltä Research. Lisäksi luotiin Schools-niminen kansio, joka jakautuu Schools of ICT- ja Schools of Smart and Clean Slt -kansioihin. Nämä kansiot on tarkoitettu nimensä mukaisesti osaamisalueiden projekteille. Smart and Clean Slt -kansio sisältää jaetun VPC-projektin. Opinnäytetyön tekijä toteaa, että Common-kansiota lukuun ottamatta luodun kansiorakenteen ja jaetun VPC-projektin tarkoitus oli demonstroida, miten kansiorakennetta voidaan jatkossa suunnitella ja kuinka jaettua verkkoa voidaan hyödyntää.

3.2 Organisaation käytännöt

Organisaatiokäytännöt on otettu käyttöön kahdella tasolla: koko Metropolian organisaatio (metropolia.fi) ja laskeutumisalue (Metropolia-kansio). On huomioitava, että käytännöt eivät ole taannehtivia tai takautuvia (retroactive), eli ne eivät koske resursseja, jotka on luotu ennen käytännön käyttöönottoa.

Metropolia.fi

Opinnäytetyön tekijä toteaa, että Metropolian organisaatiotasolla on otettu käyttöön yhteensä seitsemän organisaatiokäytäntöä. Nämä käytännöt on valittu huolellisesti vastaamaan tiettyihin toiminnallisiin ja turvallisuuden tarpeisiin.

Näitä käytäntöjä sovelletaan keskeisiin osa-alueisiin pilvihallinnassa, kuten verkkojen turvallisuuden parantamiseen oletusverkkojen luomisen ohittamisella ja yhtenäisten pääsyvalvontalistojen (ACL) noudattamisella Cloud Storage -säiliöiden suojelemiseksi. Oleellisten yhteystietojen toimialueiden (essential contacts domains) lisäksi rajoitetaan myös jaettujen VPC-verkkojen lukkojen (lien) vahingossa tapahtuvaa poistoa sekä estetään palvelutiliavainten lataaminen alustalle turvariskien minimoimiseksi. Lisäksi rajoitetaan tietokeskuksien sijainnit Euroopan unioniin tietosuvereniteetin ja alueellisten määräysten noudattamiseksi ja on otettu käyttöön vmExternalIpAccess-käytäntö, joka määrittelee, mitkä virtuaalikoneet voivat käyttää ulkoisia IP-osoitteita. Nämä organisaatiokäytännöt yhdessä tukevat Metropolian ympäristön turvallisuutta, ja ne ovat linjassa Metropolian sitoumuksen kanssa tarjota turvallinen ja luotettava pilvipalveluinfrastruktuuri (esimerkkikoodi 4).

```
gcloud resource-manager org-policies list --organization=org_id
```

Esimerkkikoodi 4. Organisaatiokäytäntöjen listaaminen organisaatiotasolla gcloud-komentorivityökalulla.

Metropolia-kansio (laskeutumisalue)

Laskeutumisalueella on aktivoitu kahdeksan keskeistä organisaatiokäytäntöä, jotka koskevat myös kaikkia sen alaisia resursseja. Nämä käytännöt palvelevat erilaisia tarkoituksia: virtuaalikoneiden sarjaporttien poistaminen käytöstä paremman tietoturvan takaamiseksi, suojattujen virtuaalikoneiden käytön pakottaminen manipulaatioiden estämiseksi, julkisten IP-osoitteiden käytön rajoittaminen tietokantoihin Cloud SQL -välityksen kautta, lähdekoodin latauksen estäminen App Engine -sovelluksesta lähdekoodien suojelemiseksi, Cloud Storage -resurssien julkisen käytön estäminen, VPN-yhteyksien IP-osoitteiden rajoittaminen ja protokollan välityksen rajoittaminen turvallisuuden parantamiseksi sekä oletuspalvelutili-oikeuksien poistaminen mahdollisten haavoittuvuuksien

minimoimiseksi. Nämä käytännöt yhdessä vahvistavat Metropolian pilviympäristöä noudattamalla parhaita käytäntöjä ja korostamalla sitoutumistaan vahvaan tietoturvaan ja hallintaan (esimerkkikoodi 5).

```
gcloud resource-manager org-policies list --folder=folder_id
```

Esimerkkikoodi 5. Organisaatiokäytäntöjen listaaminen hakemistotasolla gcloud-komentorivityökalulla.

3.3 Järjestelmävalvojien käyttöoikeudet

Käyttäjäryhmällä "gcp-organization-admin" on laaja valikoima (18 kpl) rooleja organisaatiotasolla. Nämä roolit kattavat laajan kirjon oikeuksia, jotka mahdollistavat pilviresurssien tehokkaan ylläpidon ja hallinnan, kuten Organization Administrator ja Billing Account Administrator, jotka antavat oikeudet organisaation rakenteen ja taloudellisten näkökohtien hallintaan. Lisäksi on rooleja, jotka keskittyvät tietoturvaan ja pääsynhallintaan, kuten Security Admin ja Access Context Manager Admin. Lisäksi roolit, kuten Cloud Build Creator ja Compute Admin, antavat ryhmälle mahdollisuuden tehokkaasti valvoa pilviresurssien käyttöönottoa ja hallintaa. Tämän monipuolisen roolien valikoiman avulla käyttäjäryhmä voi suorittaa tehtävänsä tehokkaasti ja varmistaa Metropolian ympäristön sujuvan toiminnan ja turvallisuuden.

3.4 Turvallisuusvalvojien käyttöoikeudet

Käyttäjäryhmälle "Gcp-security-admins" on myönnetty kattava joukko rooleja organisaatiotasolla, yhteensä kymmenen roolia, jotka on suunniteltu helpottamaan ryhmän jäsenten vastuita. Nämä roolit antavat heille mahdollisuuden hallita ja turvata resursseja tehokkaasti.

Näiden roolien joukossa käyttäjäryhmällä on oikeuksia, kuten BigQuery Data Viewer datan analysointiin, Compute Viewer laskentaresurssien seurantaan ja Folder IAM Admin hienojakoiseen pääsynhallintaan. Lisäksi ryhmän jäsenillä on rooleja Kubernetes-resurssien valvontaan, lokien määritykseen ja organisaation

laajuisten käytäntöjen hallintaan. Tämä roolien joukko varustaa käyttäjäryhmän tarvittavilla käyttöoikeuksilla ylläpitääkseen tehokkaasti resurssien eheyttä ja turvallisuutta ympäristössä.

3.5 Verkkovalvojien käyttöoikeudet

Käyttäjäryhmällä Gcp-network-admins on organisaatiotasolla useita tärkeitä rooleja, kuten Compute Network Admin, joka myöntää verkon hallintaoikeuksia, Compute Security Admin turvallisuustehtäviin, Compute Shared VPC Admin VPC-verkkojen hallintaan sekä Folder Viewer kansioihin liittyvien oikeuksien tarkasteluun. Näiden roolien avulla ryhmällä on tarvittavat oikeudet hallinnoida tehokkaasti verkkoresursseja, turvallisuusasetuksia ja jaettuja VPC:itä ja saada näkymä kansioiden rakenteisiin.

3.6 Cloud Build -palvelutilin käyttöoikeudet

GCP-ympäristön hallintaan tarkoitetun “prj-mtp-prd-ict-iac”-projektin Cloud Build -palvelutilille on myönnetty joukko kriittisiä organisaatiotason rooleja ympäristön provisioinnin mahdollistamiseksi Terraform-koodilla. Nämä roolit ovat Access Context Manager Admin, Compute Security Admin, Compute Shared VPC Admin, Essential Contacts Admin, Folder Admin, Logging Admin, Monitoring Admin, Organization Policy Administrator, Organization Role Administrator, Project Creator, Security Admin ja Viewer. Myönnetyt roolit mahdollistavat muun muassa ympäristön rakenteen määrittämisen, käyttöoikeuksien hallinnan ja projektien luomisen.

3.7 Keskitetty monitorointi

Tämä Google Cloud -projekti toimii keskitettynä paikkana projektien resurssien monitoroinnille. Sen ensisijainen tarkoitus on tarjota yhtenäinen ja standardoitu ympäristö organisaation eri projektien valvontaan. Asettamalla "attach_to_monitoring"-parametrin todeksi (true) Terraform-projektitehtaassa uudet projektit

voidaan helposti liittää tähän keskitettyyn monitorointi-infrastruktuuriin, mikä takaa yhtenäiset valvontakäytännöt koko ekosysteemissä.

Projektissa ylläpitäjillä on mahdollisuus luoda räätälöityjä valvontaryhmiä, jotka on suunniteltu erityisille kohdeyleisöille tai käyttötarkoituksille. Tämä mahdollistaa hälytys- ja ilmoitusstrategioiden hienosäätämisen, millä varmistetaan, että asianmukaiset sidosryhmät saavat ajantasaista tietoa resurssiensa tilasta. Keskitetyn monitoroinnin projektissa on luotu sähköposti-ilmoituskanava, joka lähettää ilmoituksia edelleen Microsoft Teams -kanavalle käyttäen kanavan sähköpostiosoitetta (esimerkkikoodi 6).

```
module "prj-mtp-fk-centralized_monitoring" {
  source               = "./project-module"
  project_id           = "uuden-projektin-id"
  folder_id            = google_folder.fuka-folder.id
  budget_amount        = "15"
  owners               = ["user:käyttäjänimi@metropolia.fi"]
  attach_to_monitoring = true
}
```

Esimerkkikoodi 6. Uuden projektin luominen Terraform-projektitehtaalla ja sen liittäminen keskitettyyn monitorointiin.

3.8 Jaettu VPC

Puhtaat ja älykkäät -osaamisalueelle on luotu oma erillinen jaettu verkko, joka sijaitsee prj-mtp-shared-network-prd-nimisessä projektissa. Projekti sijaitsee kansiossa Production, joka löytyy ”School Of Smart and Clean Slt” -kansiosta. Projekteja voidaan liittää tähän verkkoon määrittämällä Terraform- projektitehtaan ”enabled_shared_vpc_service project”- ja ”shared_vpc”-parametrit (esimerkkikoodi 7).

```
module "prj-mtp-fk-use-shared-vpc" {
  source               = "./project-module"
  project_id           = "uuden-projektin-id"
  folder_id            = google_folder.fuka-folder.id
  budget_amount        = "15"
  owners               = ["user:käyttäjänimi@metropolia.fi"]
  enable_shared_vpc_service_project = true
  share_vpc            = jaetun_verkon_projektin_id
}
```

Esimerkkikoodi 7. Uuden projektin luominen Terraform-projektitehtaalla ja sen liittäminen jaettuun verkkoon.

3.9 Kustannusanalyysi

Erillinen prj-mtp-billing-export-gcp-niminen projekti on luotu keskittämään kaikkien projektien laskutustiedot. Projektien laskutustiedot viedään tämän projektin sisältämään BigQuery-tietokantaan. Tämä laskutustilitason vientiprosessi on konfiguroitu Cloud Billing -palvelun Billing exports -osiossa, jossa laskutustilin sekä "Standardikäytön kustannukset" että "Yksityiskohtaisen käytön kustannukset" ovat käytössä. BigQuery-tietojoukot päivitetään päivittäin, jotta voidaan varmistaa tiedon ajantasaisuus. Näitä tietoja voidaan visualisoida Looker Studiossa. On huomioitava, että projektin tunnisteet mahdollistavat tehokkaan kustannusanalyysin.

3.10 Kustannusten hallinta

Opinnäytetyön kontekstissa kustannushallinnalla viitataan mekanismeihin, joilla varmistetaan projektin luonnin yhteydessä sille määritetyssä budjetissa pysyminen. Kun projektille määritetään budjetti sen luonnin yhteydessä, tavoitteena on pysyä budjetin rajoissa ja pysyä tietoisena budjetin käytön kasvusta projektin elinkaareen aikana. GCP tarjoaa kustannushallintaan vaihtoehtoja, joita on kuvattu tarkemmin raportin toisen luvun kohdassa Cloud Billing.

Metropolian pilviympäristössä on käytössä kaksi eri mekanismia, budjettihälytykset ja kustannusten pysäyttäminen. Organisaatiotasolla on käytössä kaksi budjettihälytystä, jotka on määritetty seuraamaan kaikkien projektien kustannuksia (koko laskutustili) ja lähettämään kynnysrajoihin (threshold) perustuvia

ilmoituksia. On huomioitava, että edellä mainitut budjetit ja hälytykset eivät sellaisenaan pysäytä lisäkustannusten syntymistä.

Projektin luonnin yhteydessä nk. Tarraform-projektitehtaalla voidaan vaihtoehtoisesti ottaa käyttöön lisämekanismi lisäkustannusten pysäyttämiseen. Kun projektin kustannukset ylittävät budjetin, projektista poistetaan laskutustili, mikä johtaa palveluinstanssien ja siten kustannusten pysäyttämiseen. Mekanismi kuvataan tarkemmin luvussa 3.11.

On huomioitava, että jokaisessa projektissa on ennalta määritetyt kiintiöt ja allokaatiot useimmille resursseille ja rajapinnoille, ja joissakin tapauksissa ne saattavat tarjota rajoittamatonta käyttöä. On kuitenkin myös mahdollista mukauttaa tiettyjä kiintiöitä paremman pilviresurssien valvonnan ja kustannusten hallinnan toteuttamiseksi. (42.)

3.11 Projektitehdas

Uuden projektin luomiseen on kehitetty Terraform-moduuli nimeltä project-module, joka sijaitsee prj-mtp-production-ict-nimisen projektin lähdekoodivarastossa prj-mtp-production-ict-iac-project-factory-nimellä. Moduulilla on useita pakollisia ja valinnaisia parametreja, joilla konfiguroidaan luotavan projektin ominaisuudet.

Moduulin koodin lähteen märittämisen lisäksi sen pakollisiin kenttiin kuuluu project_id, jolla määritetään projektin tunniste (ID). On huomioitava, että projektin tunnisteen on oltava globaalisti uniikki. Folder_id-parametrilla määritetään projektin kansio, joka voi olla esimerkiksi tietylle kurssille luotu kansio. Budget_amount-parametri luo ja asettaa projektille oman budjetin, jossa on 70 %:n ja 100 %:n kynnysarvoihin perustuvat hälytykset. Attach_to_monitoring-parametrilla voidaan liittää projekti keskitettyyn seurantaan (esimerkkikoodi 8).

```
module "prj-mtp-fk-playground" {
  source               = "./project-module"
  project_id           = "prj-mtp-fk-playground"
  folder_id            = google_folder.fuka-folder.id
  budget_amount        = "10"
  attach_to_monitoring = false
}
```

Esimerkkikoodi 8. Uuden projektin luominen Terraform-projektitehtaan pakollisilla parametreilla.

Valinnaisilla parametreilla voidaan asettaa projektin käyttöoikeuksia kolmella tasolla: näyttäjä (viewer), muokkaaja (editor) ja omistaja (owner). Jokaisella roolilla on oma parametrinsa, joka hyväksyy syötteenä taulukon käyttäjien sähköpostiosoitteista. Virhetilanteiden välttämiseksi Metropolian käyttäjien sähköpostiosoitteiden on oltava muodossa käyttäjänimi@metropolia.fi. Lisäksi voidaan määrittää projektin tunnisteet, ottaa käyttöön tarkastuslokit ja GCP:n rajapinnat, liittää projekti keskitettyyn virtuaaliverkkoon ja antaa projektille oikeus muokata sen palomuuriasetuksia.

Asettamalla parametrin revoke_billing arvoksi "true" luodaan projektille toinen 100 %:n kynnysarvoon perustuva budjettihälytys, pilvifunktio ja Pub/Sub-aihe. Projektin kustannuksia sisältävä hälytys lähetetään Pub/Sub-aiheeseen, joka laukaisee kustannustenhallintaan tarkoitetun funktion. Funktiossa tarkistetaan, ovatko projektin kustannukset ylittäneet sille projektin luomisen yhteydessä määrätyn budjetin, ja se poistaa projektista laskutustilin, mikäli se on ylittänyt sen. On huomioitava, että tämä ylimääräinen mekanismi ei takaa budjetissa pysymistä, sillä budjettihälytykset saatetaan toimittaa viiveellä.

Projektitehtaan oletustoiminnallisuus

Oletuksena uudessa projektissa otetaan käyttöön useita GCP-rajapintoja, kuten appengine, bigquery, compute, cloudbuild, container, containeranalysis, containerregistry, cloudresourcemanager, iap, iamcredentials, iam, logging, monitoring, oslogin, sqladmin, stackdriver, servicenetworking, secretmanager ja cloudbilling ja lisää voidaan ottaa käyttöön projektitehtaan enable_apis- parametrilla, joka ottaa syötteenä rajapintojen taulukon. Lisäksi projektille luodaan suojamekanismi (lienin tai lukon), joka estää projektin vahingossa tai tahattomasti

poistamisen (lien lähde). Tämä on yleinen käytäntö monissa tietojärjestelmissä ja sovelluksissa, jotta tärkeät tiedot säilyvät turvassa, ja se voidaan poistaa gcloud-komentorivityökalulla. Lisäksi projektille luodaan Cloud Storage -säiliö Terraform-tilan tallentamiseksi. (43.)

3.12 Kehityskohteet

Opinnäytetyön yhtenä keskeisenä tavoitteena oli tarjota ehdotuksia Metropolian pilviympäristön kehittämiseksi ja parantamiseksi. Opinnäytetyössä esitetään yhteensä viisitoista konkreettista kehittämiskohdetta ja parannusehdotusta. Näillä ehdotuksilla pyritään edistämään Metropolian pilviympäristön tehokkuutta, käytettävyyttä ja turvallisuutta.

1. Metropolian tulisi laatia pilvistrategia, jossa otetaan huomioon tarvittavat resurssit pilvialustan ylläpitämiseen ja kehittämiseen. Näille tahoille on luotava erillinen sähköpostiosoite, jolla he kirjautuvat suorittamaan ylläpitotehtäviään.

2. Ylläpitoryhmien jäsenluetteloa on pidettävä ajan tasalla, ja ryhmien roolien myöntämisessä on noudatettava vähimmäisoikeuksien periaatetta.

3. Metropolian tulee suunnitella ja laatia tarkoituksenmukainen hakemistorakenne. Asiaankuuluville tahoille on myönnettävä tarvittavat käyttöoikeudet päivittäisten toimenpiteiden suorittamiseen. Esimerkiksi pääainevastaaville voidaan luoda omat hakemistonsa ja myöntää heille hakemistotasossa omistajaoikeudet. Tämä merkittävästi vähentäisi ylläpitäjien työkuormaa.

4. Projektitehdasta on kehitettävä siten, että käyttäjille voidaan myöntää halutut roolit ja yksittäiset käyttöoikeudet, ja mahdollistettava estokäytäntöjen määrittäminen.

5. Terraform- ja Google Cloud -tarjoajan versiot on päivitettävä ja pidettävä aina ajan tasalla.

6. Virtuaalikoneiden ulkoiset IP-osoitteet on sallittava Student Sandbox -kansiossa kokeilumahdollisuuksien lisäämiseksi. Tämä ei aiheuta turvallisuusriskejä eikä vaikuta muihin projekteihin.

7. Kustannusten hallinnan toiminallisuus on keskitettävä erilliseen projektiin, ja projektit liitetään tähän projektitehtaan parametrilla. Muut projektit julkaisevat kustannustietonsa keskitetyn projektin Pub/Sub-aiheeseen, joiden palvelutilille myönnetään tarvittavat käyttöoikeudet keskittävässä projektissa. Tämä johtaa uusien projektien luontinopeuden merkittävään kasvuun. Nykyinen mekanismi on hidas, koska jokaiselle projektille luodaan oma pilvifunktio, jonka luominen voi kestää jopa yli 2,5 minuuttia. Opinnäytetyön tekijä toteaa, että toisen sukupolven pilvifunktio on tarkoitukseen sopiva valinta sen samanaikaisen suorituskyvyn ansiosta.

8. Olemassa olevilta projekteilta poistetaan nyt käytössä oleva kustannusten hallintaan tarkoitettu mekanismi ja samalla otetaan käyttöön edellä mainittu uusi mekanismi. Nykyinen mekanismi poistetaan käytöstä yksinkertaisesti asettamalla luotujen projektien revoke_billing-parametrin arvo epätodeksi (false).

9. Projektitehdasta tulee päivittää siten, että se luo projektien lokisäiliöt EU:n datakeskuksissa. GCP ei salli jo luotujen lokisäiliöiden sijainnin (region) muuttamista niiden luomisen jälkeen. Nykyisille projekteille on luotava uudet säiliöt, ja samalla on päätettävä, halutaanko säilyttää vanhat lokit ennen vanhojen säiliöiden poistamista. Vanhat lokit voidaan viedä uusiin säiliöihin, esimerkiksi tallennettavaksi Cloud Storage -säiliöön. (44)

10. Kustannusten hallintaan tarkoitettujen budjettihälytysten summan aikaväli on muutettavaa koskemaan koko vuotta, sillä nykyisin ne koskevat vain yhtä kuukautta.

11. On luotava hälytys ilmoittaman asiaankuuluville tahoille, kun projektista poistetaan laskutustili. Tämä saavutetaan esimerkiksi lokitukseen perustuvan monitorointihälytyksen avulla tai hyödyntämällä Eventarc-palvelua,

joka vastaanottaa Cloud Billing -palvelun DisableResourceBilling-tapahtuman laskutustilin poiston yhteydessä.

12. Käytössä oleva google_project_service-niminen Terraform-moduuli on korvattava google_project_services-moduulilla (45; 46).

13. Projektitehtaan automaattisesti käyttöönotettavat rajapinnat (GCP-palvelut) on rajoitettava vain niihin, jotka ovat olennaisia ja välttämättömiä luotavan projektin luomista ja toimivuutta varten.

14. On luotava syötteitä reaaliaikaisten ilmoitusten saamista varten, kun pilviympäristössä tapahtuu kriittisiä muutoksia (10).

15. Ylläpitäjien tulee ottaa käyttöön kaksivaiheinen tunnistautuminen käyttämällä fyysistä turva-avainta (47).

4 Yhteenveto

Opinnäytetyön päätavoitteena oli syventyä Metropolia Ammattikorkeakoulun ympäristön tekniseen toteutukseen Google Cloud -pilvialustalla, luoda kattava dokumentaatio toteutetusta ympäristöstä ja siihen liittyvästä Terraform-lähdekoodista sekä ehdottaa kehittämiskohteita ja parannuksia.

Opinnäytetyössä tarkasteltiin yksityiskohtaisesti pilviympäristön teknistä toteutusta analysoimalla saatavilla olevaa lähdekoodia Google Cloud Source Repository -palvelusta. Tutkimusprosessissa käytettiin hyväksi pilvialustan tarjoamia komentorivityökaluja ja kyselykieliä, jotka mahdollistivat tarkemman ymmärryksen ympäristön tilasta. Lisäksi opinnäytetyössä käsiteltiin Google Cloud -pilvialustan käyttöönottoprosessia ja tärkeitä siihen liittyviä palveluita.

Opinnäytetyön tuloksena syntyi selkeä ja yksityiskohtainen dokumentaatio Metropolia Ammattikorkeakoulun pilviympäristöstä sekä siihen liittyvästä Terraform-lähdekoodista. Lisäksi työ tarjosi konkreettisia ehdotuksia kehittämis- ja

parannuskohteista, jotka voivat auttaa optimoimaan ja tehostamaan pilviympäristön toimintaa ja turvallisuutta tulevaisuudessa.

Opinnäytetyö tarjoaa arvokasta tietoa Metropolia Ammattikorkeakoululle jatkokehitystyön tueksi ja auttaa ymmärtämään paremmin Googlen pilvialustan mahdollisuuksia ja haasteita organisaation kontekstissa. Työn tulokset voivat myös toimia pohjana tuleville projekteille ja päätöksentekoon.

Opinnäytetyön myötä saavutettiin merkittäviä etuja, kuten resurssien tehokkaampi hyödyntäminen, skaalautuvuus ja kustannustehokkuus. Google Cloud -pilvialustan avulla voitiin nopeuttaa projektien käynnistämistä ja hallintaa sekä tarjota parempaa palvelua opetus-, TKI- ja yritysprojekteissa. Tämä tehosti organisaation toimintaa ja mahdollisti entistä nopeamman reagoinnin muuttuviin tarpeisiin.

Opinnäytetyön aikana havaittiin, että tehokkain tapa oppia ja ymmärtää pilvialustan palveluita ja toiminnallisuuksia oli käyttää pilvialustan konsolia käytännössä. Ainoastaan dokumentaation lukemalla ei ollut mahdollista sisäistää kaikkia yksityiskohtia ja hienouksia, joita Googlen pilvipalvelu tarjoaa. Jokaisen palvelun dokumentaatio oli laaja, ja sen perusteellinen läpikäynti vei aikaa.

Käytännön kokeilut ja konsolin käyttö mahdollistivat opinnäytetyön tekijälle syvällisen ymmärryksen siitä, miten eri palvelut toimivat ja miten ne voivat parhaiten palvella organisaation tarpeita. Käytännön kokemukset auttoivat havaitsemaan käytännön haasteita ja ratkaisuja, joita dokumentaatiosta oli vaikea hahmottaa.

Opinnäytetyö osoitti, että aktiivinen konsolin käyttö, käytännön harjoitukset ja opitun soveltaminen mukautettuihin projekteihin ovat välttämättömiä oppimisprosessissa. Dokumentaatio on tärkeä resurssi, mutta sen täydentäminen käytännön kokemuksella on avain onnistuneeseen oppimiseen ja tehokkaaseen pilvialustan hyödyntämiseen.

Opinnäytetyön tekijä kehitti osaamistaan merkittävästi Googlen pilvialustan toiminnasta ja parhaista käytännöistä. Tämä opinnäytetyö antoi tarvittavan asiantuntemuksen ja valmiudet toteuttaa itsenäisesti turvallinen Google Cloud -pilvialustan käyttöönottoprosessi. Tämä taito on arvokas, sillä se mahdollistaa Metropolian organisaation resurssien tehokkaan hyödyntämisen. Opinnäytetyön myötä opinnäytetyön tekijä on nyt valmis ottamaan lisää vastuita pilvipalveluiden käyttöönoton ja hallinnan saralla, mikä voi tuottaa merkittäviä etuja organisaatiolle. Tämä osaaminen voi tukea Metropolian tulevia pyrkimyksiä hyödyntää pilvipalveluita entistä tehokkaammin ja innovatiivisemmin erilaisissa toimintaympäristöissä.

Opinnäytetyön tuloksena Metropolia saavutti turvallisen ja kustannustehokkaan pilviympäristön käyttöönoton, mikä paransi sen kilpailukykyä ja tehokkuutta toiminnassa. Opinnäytetyön tulokset ja saavutetut taidot tarjoavat vahvan perustan jatkaa pilvipalveluiden menestyksekästä hyödyntämistä tulevaisuudessa.

Lähteet

1 70% Of Generative AI Startups Rely On Google Cloud, AI Capabilities. 2023. Verkkoaineisto. Forbes. <https://www.forbes.com/sites/johanmoreno/2023/07/25/70-of-generative-ai-startups-rely-on-google-cloud-ai-capabilities-says-alphabet-ceo-sundar-pichai/?sh=7ac153361243>. Päivitetty 25.7.2023. Luettu 31.7.2023.

2 Google Cloud overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/docs/overview> Päivitetty 13.9.2023. Luettu 14.9.2023.

3 Use IAM securely. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/iam/docs/using-iam-securely#least_privilege>. Päivitetty 13.9.2023. Luettu 14.9.2023.

4 IAM overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/iam/docs/overview>. Päivitetty 13.9.2023. Luettu 14.9.2023.

5 VPC networks. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/vpc/docs/vpc>. Päivitetty 31.8.2023. Luettu 5.9.2023.

6 Shared VPC. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/vpc/docs/shared-vpc>. Päivitetty 8.9.2023. Luettu 10.9.2023.

7 Introduction to Cloud Asset Inventory. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/asset-inventory/docs/overview>. Päivitetty 4.8.2023. Luettu 4.8.2023.

8 Supported Asset types. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/asset-inventory/docs/supported-asset-types> Päivitetty 4.8.2023. Luettu 4.8.2023.

9 Cloud Asset Inventory. Verkkoaineisto. Google Cloud. <https://cloud.google.com/asset-inventory>. Luettu 4.8.2023.

10 Monitoring asset changes. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/asset-inventory/docs/monitoring-asset-changes>. Päivitetty 4.8.2023. Luettu 5.8.2023.

11 OS inventory management. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/compute/docs/instances/os-inventory-management >. Päivitetty 4.8.2023. Luettu 5.8.2023.

12 Features. Verkkoaineisto. 2023. Google Cloud. <https://cloud.google.com/source-repositories/docs/features>. Päivitetty 4.8.2023. Luettu 5.8.2023.

13 Detecting security keys. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/source-repositories/docs/detecting-security-keys>. Päivitetty 4.8.2023. Luettu 5.8.2023.

14 Access control with IAM. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/source-repositories/docs/configure-access-control>. Päivitetty 4.8.2023. Luettu 5.8.2023.

15 Notifications for Cloud Source Repositories. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/source-repositories/docs/pubsub-notifications>. Päivitetty 4.8.2023. Luettu 5.8.2023.

16 Notifications for Cloud Source Repositories. Verkkoaineisto. Google Cloud. <https://cloud.google.com/build>. Luettu 6.8.2023.

17 Overview of Cloud Build. 2023. Verkkoaineisto. Google. <https://cloud.google.com/build/docs/overview>. Päivitetty 4.8.2023. Luettu 6.8.2023.

18 Cloud Funtions overview. 2023. Verkkoaineisto. Google. <https://cloud.google.com/functions/docs/concepts/overview>. Päivitetty 4.8.2023. Luettu 7.8.2023.

19 Cloud Functions execution environment. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/functions/docs/concepts/execution-environment>. Päivitetty 4.8.2023. Luettu 7.8.2023.

20 Google APIs. Verkkoaineisto. Google Cloud. <https://github.com/googleapis>. Luettu 7.8.2023.

21 Cloud Functions version comparison. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/functions/docs/concepts/version-comparison>. Päivitetty 4.8.2023. Luettu 7.8.2023.

22 What is Pub/Sub? Verkkoaineisto. 2023. Google Cloud. <(https://cloud.google.com/pubsub/docs/overview>. Päivitetty 11.9.2023. Luettu 12.9.2023.

23 Harris, Chandler. Microservices vs. monolithic architecture. Verkkoaineisto. Atlassian. <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>. Luettu 7.8.2023.

24 Event-driven architectures. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/eventarc/docs/event-driven-architectures>. Päivitetty 4.8.2023. Luettu 7.8.2023.

25 Eventarc overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/eventarc/docs/overview>. Päivitetty 4.8.2023. Luettu 7.8.2023.

26 Cloud Billing overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/billing/docs/concepts>. Päivitetty 4.8.2023. Luettu 9.8.2023.

27 Create and manage your payment profile. Verkkoaineisto. Google. <https://support.google.com/paymentscenter/answer/9028746?hl=en>. Luettu 4.8.2023.

28 Create, edit, or delete budgets and budget alerts. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/billing/docs/how-to/budgets>. Päivitetty 10.8.2023. Luettu 11.8.2023.

29 Manage programatic budget alert notifications. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/billing/docs/how-to/budgets-programmatic-notifications>. Päivitetty 10.8.2023. Luettu 11.8.2023.

30 Create, shut down, and restore projects. Verkkoaineisto. Google. <https://support.google.com/googleapi/answer/6251787?hl=en#zippy=%2Cshut-down-a-project>. Luettu 11.8.2023.

31 Service Usage API. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/service-usage/docs/reference/rest>. Päivitetty 13.8.2023. Luettu 18.8.2023.

32 Google Cloud Pricing Calculator. Verkkoaineisto. Google Cloud. <https://cloud.google.com/products/calculator>. Luettu 11.8.2023.

33 Security Command Center overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/security-command-center/docs/concepts-security-command-center-overview>. Päivitetty 8.9.2023. Luettu 10.9.2023.

34 Cloud Logging overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/logging/docs/overview>. Päivitetty 10.8.2023. Luettu 11.8.2023.

35 Routing and storage overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/logging/docs/routing/overview> Päivitetty 10.8.2023. Luettu 11.8.2023.

36 Quotas and limits. Verkkoaineisto. 2023. Google Cloud. <https://cloud.google.com/logging/quotas#logs_retention_periods>. Päivitetty 8.9.2023. Luettu 9.9.2023.

37 Cloud Monitoring overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/monitoring/docs/monitoring-overview>. Päivitetty 8.9.2023. Luettu 11.9.2023.

38 Default encryption at rest. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/docs/security/encryption/default-encryption>. Päivitetty 8.9.2023. Luettu 12.9.2023.

39 Encryption in transit. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/docs/security/encryption-in-transit>. Päivitetty 8.9.2023. Luettu 12.9.2023

40 Terraform-google modules. Verkkoaineisto. Google Cloud and HashiCorp. <https://github.com/terraform-google-modules>. Luettu 3.7.2023.

41 Sensitive Data Protection overview. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/dlp/docs/sensitive-data-protection-overview>. Päivitetty 8.7.2023. Luettu 21.7.2023.

42 Work with quotas. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/docs/quota>. Päivitetty 8.9.2023. Luettu 13.9.2023.

43 Protecting projects with liens. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/resource-manager/docs/project-liens>. Päivitetty 8.9.2023. Luettu 12.9.2023.

44 Configure log buckets. 2023. Verkkoaineisto. Google Cloud. <https://cloud.google.com/logging/docs/buckets>. Päivitetty 12.9.2023. Luettu 13.9.2023.

45 User Guide – google_project_service. Verkkoaineisto. HashiCorp. <https://registry.terraform.io/providers/hashicorp/google-beta/latest/docs/guides/google_project_service>. Luettu 6.7.2023.

46 Project API Activation. Verkkoaineisto. Google Cloud. <https://github.com/terraform-google-modules/terraform-google-project-factory/tree/master/modules/project_services>. Luettu 6.7.2023.

47 Titan Security Key. Verkkoaineisto. Google Cloud. <https://cloud.google.com/titan-security-key>. Luettu 14.9.2023.