Tiedon kopiointi koululaitoksen paikalliselta palvelimelta pilveen Azuren palveluita hyödyntäen

Nita Corrêa

Haaga-Helia ammattikorkeakoulu

Amk-opinnäytetyö

2021

Tietojenkäsittelyn tutkinto

Tiivistelmä

Tekijä(t)
Nita Corrêa

Tutkinto
Tradenomi

Raportin/Opinnäytetyön nimi
Tiedon kopiointi koululaitoksen paikalliselta palvelimelta pilveen Azuren palveluita hyödyntäen

Sivu- ja liitesivumäärä
46 + 4

Tässä opinnäytetyössä otetaan valitut Microsoft Azuren palvelut käyttöön ja kopioidaan tietoa paikallisesta varmuuskopiotietokannasta pilveen. Opinnäytetyö on tehty Haaga-Helia ammattikorkeakoulun tietohallinnon toimeksiantona ja se on osa ammattikorkeakoulussa toteutettua Oppimisanalytiikan alustat -projektia, jonka tavoitteena on kasvattaa organisaation osaamista valituista työkaluista, hyödyntää oppilaitoksen tietoa sekä lisätä valmiutta siirtyä laajemmin hyödyntämään pilvipalveluita. Projektin tarkoituksena on toimia pilottina ja testinä vastaavanlaisille projekteille Haaga-Helia ammattikorkeakoulun tietohallinnossa.

Opinnäytetyö on jaettu kahteen osaan, josta ensimmäinen keskittyy kuvaamaan yleisellä tasolla tietoon ja tiedon analysointiin liittyviä perusasioita, tiedonsiirrossa- ja tallennuksessa käytettäviä pilvipalveluita sekä pilviympäristöä paikallisen konesalin jatkeena. Käytännön osuudessa kuvataan vaihe vaiheelta palveluiden käyttöönotto sekä tiedon kopioinnissa käytettyjen tietolinjastojen toiminta.

Projektissa luodaan Microsoft Azuren portaalissa käyttöönotettaville palveluille resurssiryhmä, jonka yhteyteen Data Factory V2, Data Lake Gen2, Key Vault ja Private Endpoint -palvelut otetaan käyttöön, muodostetaan yhteys paikallisen palvelimen ja pilvestä löytyvän resurssin välille sekä kuvataan tiedon kopioinnissa käytettyjen tietolinjastojen toiminta.

Asiasanat
Pilvipalvelut, Microsoft Azure, tieto, kopiointi, SQL.

Sisällys

1 Johdanto

Opinnäytetyö on tehty Haaga-Helia ammattikorkeakoulun tietohallinnon toimeksiantona. Se on osa ammattikorkeakoulussa toteutettua projektia, jonka tehtävänä on kartoittaa tiedon kopiointia analysointitarkoituksissa paikalliselta palvelimelta Microsoftin julkiseen pilvialustaan Azureen. Projekti on toteutettu Azuren palveluita hyödyntäen ja tiedon kopioinnissa käytetään Data Factory V2 -palvelua ja tiedon tallennuspaikkana pilvessä Data Lake Gen2 -tietojärveä.

Opinnäytetyö on osa Oppimisanalytiikan alustat -projektia, joka taas on osa laajempaa HAUKKA-ohjelman kokonaisuutta. Oppimisanalytiikan alustat -projektilla tavoitellaan tietoa ja osaamista organisaation sisällä tiedon kopioinnista paikalliselta palvelimelta Azuren Data Lake Gen2 -ympäristöön Data Factory V2-integraatiopalvelulla sekä valmiutta hyödyntää näitä palveluita tulevaisuudessa. Lisäksi ymmärrys ja osaaminen lisääntyvät koko pilviympäristöstä, mikä lisää organisaation valmiutta siirtyä laajemmin hyödyntämään Azuren pilvipalveluita. Opinnäytetyö ei kuitenkaan kuvaa koko Haaga-Heliassa toteutettavaa projektia, vaan vain sen ensimmäistä osaa, jossa tarvittavat palvelut otetaan käyttöön, yhteys luodaan paikallisen palvelimen ja pilvessä sijaitsevan resurssin välille sekä tiedon kopiointiprosessi käynnistetään. Projekti toteutettiin keväällä 2021 ja opinnäytetyön tekijä on osallistunut projektiin projektityöntekijänä.

Pilviympäristössä tietoa halutaan tulevaisuudessa yhdistää tarpeen tullen muuhun tietoon organisaation muista tiedonlähteistä ja analysoida sitä hyödyntäen Azuren analytiikkatyökaluja. Opinnäytetyö keskittyy kuvaamaan tiedon kopioinnissa käytettävät palvelut, ympäristön käyttöönottoa sekä tiedon kopioinnin prosessia siten, että organisaatio voi hyödyntää valmista opinnäytetyötä vastaavien projektien suunnittelussa ja toteutuksessa.

Ensimmäisessä osassa projektia työvaiheisiin kuuluivat ympäristöön tutustuminen, ympäristön valmistaminen sekä siirrettävän tiedon määrittäminen. Tämän lisäksi suunniteltiin tiedon kopioinnissa tarvittavat toiminnot Data Factory -palvelussa ja Data Lake -tietojärvessä organisaation tarpeita palveleva tiedostojärjestelmä. Kun tiedon kopiointi paikalliselta palvelimelta oli saatu polkaistua käyntiin, seurattiin prosessia sekä tehtiin tarvittavia muutoksia.

Raportti on jaettu kahteen osaan, joista ensimmäisessä osassa kuvataan tiedonsiirrossa ja tallennuksessa käytettävät palvelut yleisellä tasolla, perusasioita tiedosta ja tiedon analysoinnista sekä havainnollistetaan toimintaympäristöä. Toisessa osassa keskitytään varsinaisen toteutuksen kuvaamiseen vaihe vaiheelta.

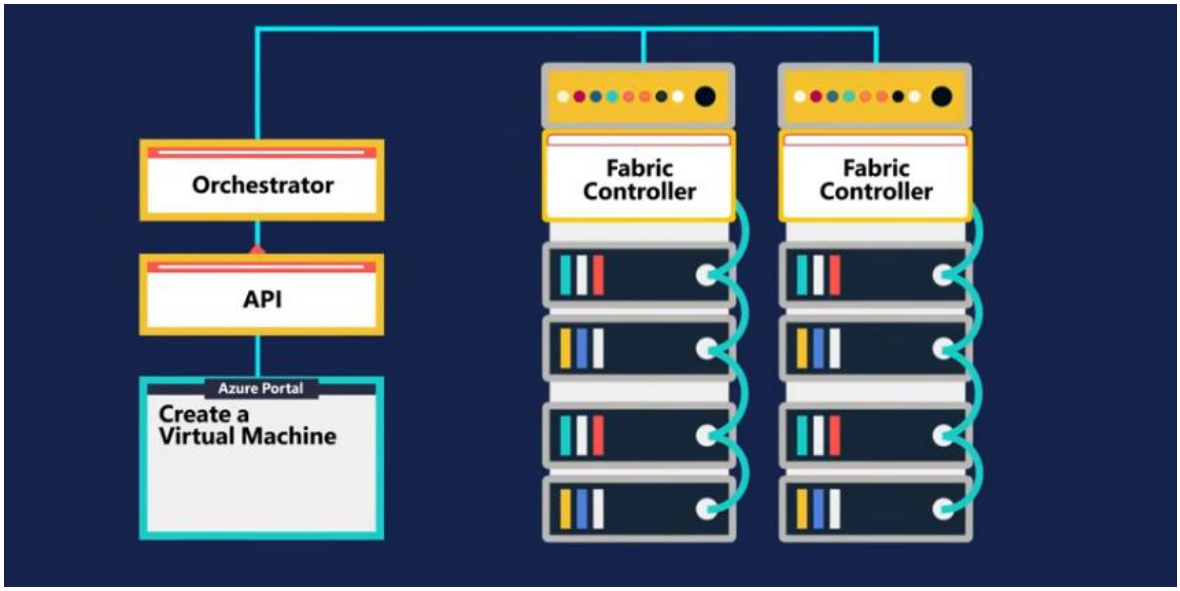
Opinnäytetyössä käytettävät termit ja niiden selitykset ovat mukana liitteessä (Liite 1).

2 Tiedon kopioinnissa käytettävät palvelut ja teoria

Tässä opinnäytetyössä tietoa kopioidaan oppilaitoksen paikalliselta palvelimelta pilveen käyttämällä Microsoft Azuren palveluita. Tiedon kopioinnissa käytetään Data Factory V2 -palvelua, jonka avulla luodaan käytettävät tietolinjastot ja seurataan tiedonsiirronprosessia paikallisen palvelimen sekä Data Lake Gen2 -tietojärvipalvelun välillä. Teoriaosuudessa avataan tietoon ja tiedon analysointiin liittyviä perusasioita, kuvataan käytettyjen palveluiden ominaisuudet sekä toimintaympäristö, jotta varsinaisen testiprojektin seuraaminen, opinnäytetyön toisessa osiossa luvussa 3, kävisi helposti.

2.1 Microsoft Azure

Microsoft Azure on Microsoftin julkinen, alati kasvava pilvialusta. Pilvellä tarkoitetaan yleisesti tietojenkäsittelyresurssien keskitettyä sijaintia, josta palveluita ostetaan julkisen Internet-verkon yli ja hinta muodostuu käytön mukaan. (Microsoft Learn). Azuressa tällaisia resursseja ovat muun muassa palvelimet, tallennustila, tietokannat ja verkot, mutta myös erilaiset avoimen lähdekoodin teknologiat sekä Microsoftin omat ratkaisut, kuten Windows-toimialueen käyttäjätietokanta ja hakemistopalvelu Active Directory (TechRepublic). Azure on julkaistu helmikuussa 2010 (w3schools).



Kuva 1. Azuren toiminta pähkinänkuoressa (Microsoft Azure 31.8.2021, 01:38 min)

Azure käyttää virtualisointitekniikkaa, jossa tietokoneen prosessori ja käyttöjärjestelmä on abstrahoitu Hypervisor-ohjelmistolla. Hypervisor emuloi oikean, fyysisen tietokoneen, kaikkia toimintoja ja prosessoria virtuaalikoneessa. Virtuaalikoneita voidaan ajaa isäntäkoneesta riippumatta millä tahansa yhteensopivalla käyttöjärjestelmällä, kuten Windowsilla tai Linuxilla. Microsoft hyödyntää tätä tekniikkaa omissa lukuisissa datakeskuksissaan ympäri maailman. (Microsoft 18.6.2018, 0:14-0:41min.).

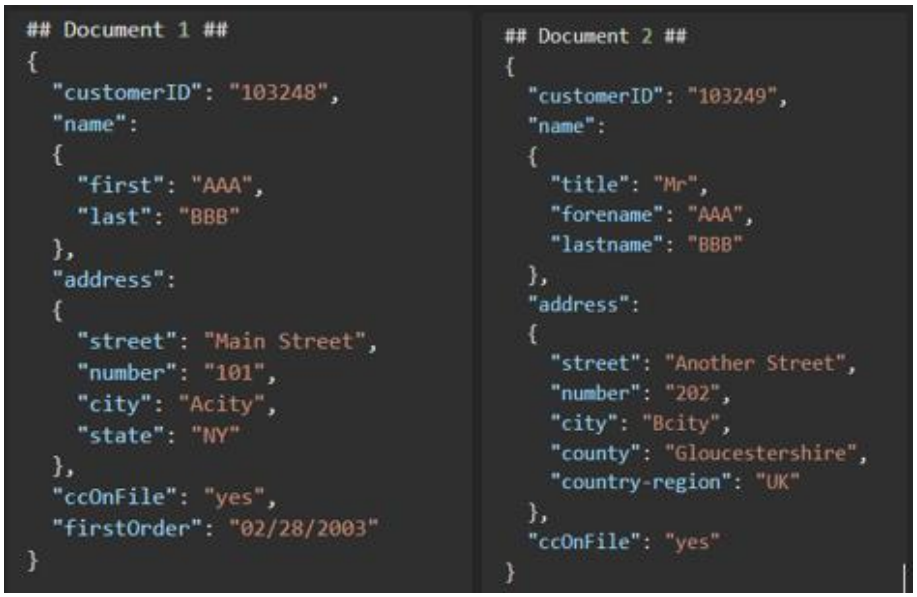
Jokainen Azuren datakeskus koostuu monista palvelinkeskuksesta, räkistä, ja jokainen palvelin sisältää Hypervisor-ohjelmiston pyörittääkseen taas lukemattoman määrän virtuaalikoneita. Verkkokytkin yhdistää jokaisen palvelimen verkkoon, ja jokaisessa räkissä yksi palvelin pyörittää Fabric Controller -ohjelmistoa. Fabric Controller on yhdistynyt orkestroijaan, toiseen ohjelmistoon, joka vastaa kaikesta, mitä Azuressa tapahtuu, kuten vastaa käyttäjäpyyntöihin. Yllä oleva kuva (Kuva 1.) havainnollistaa miten pyynnöt tehdään käyttämällä orkestroijan API-ohjelmointirajapintaa. Käyttäjä voi lähestyä rajapintaa käyttämällä esimerkiksi Azuren portaalia selaimessa. Kun käyttäjä tekee pyynnön luodakseen virtuaalikoneen, orkestroija paketoi kaiken tarvittavan, valitsee itselleen parhaimman palvelimen ja lähettää käyttäjäpyynnön sekä paketin Fabric Controllerille. Kun Fabric Controller on luonut virtuaalikoneen, käyttäjä voi ottaa etäyhteyden siihen. (Microsoft Azure 18.6.2018, 0:42-1:49 min.).

2.2 Tiedon kategoriat

CustomerID	Title	FirstName	MiddleName	LastName	Suffix	CompanyName	Phone
1	Mr.	Orlando	N.	Gee	NULL	A Bike Store	245-555-0173
2	Mr.	Keith	NULL	Harris	NULL	Progressive Sports	170-555-0127
3	Ms.	Donna	F.	Carreras	NULL	Advanced Bike Components	279-555-0130
4	Ms.	Janet	M.	Gates	NULL	Modular Cycle Systems	710-555-0173
5	Mr.	Lucy	NULL	Harrington	NULL	Metropolitan Sports Supply	828-555-0186
6	Ms.	Rosmarie	J.	Carroll	NULL	Aerobic Exercise Company	244-555-0112
7	Mr.	Dominic	P.	Gash	NULL	Associated Bikes	192-555-0173
10	Ms.	Kathleen	M.	Garza	NULL	Rural Cycle Emporium	150-555-0127
11	Ms.	Katherine	NULL	Harding	NULL	Sharp Bikes	926-555-0159
12	Mr.	Johnny	A.	Caprio	Jr.	Bikes and Motorbikes	112-555-0191
16	Mr.	Christopher	R.	Beck	Jr.	Bulk Discount Store	1 (11) 500 555-0132
18	Mr.	David	J.	Liu	NULL	Catalog Store	440-555-0132

Kuva 2. Esimerkkidataa jäsentyneestä tiedosta: jokaisella tietueella on samat attribuutit (Microsoft Learn)

Tieto voidaan karkeasti jakaa kolmeen kategoriaan: jäsentyneeseen, jäsentymättömään ja lähes jäsentyneeseen. Jäsentynyt tieto on sellaista tietoa, jolla on tietty malli ja se on helposti sijoitettavissa relaatiotietokannan tauluun ja samat attribuutit koskevat jokaista entiteettiä tai erilaisia attribuutteja entiteettien välillä ei sallita, esimerkiksi yritysten asiakastietokannat ovat usein tällaisia (Kuva 2.). (Microsoft Learn). Jäsentyneestä tiedosta käytetään usein myös nimitystä SQL-tieto. Jäsentymätön tieto taasen on nimensä mukaisesti sellaista, jolla ei ole valmista mallia eikä se ole jäsentyneen tiedon kaltaisesti sijoitettavissa tauluun. Jokaisella entiteetillä on oma muotonsa. Tällainen tieto on esimerkiksi videokuvaa, valokuvia, ääntä, sää- tai sensoritietoa. (Microsoft Learn).



Kuva 3. Lähes järjestäytyneen tiedon asiakasoliot voivat olla erilaisia: molemmista asiakasdokumenteista löytyvät samat lapsidokumentit: nimi ja osoite, mutta attribuutit näiden sisällä vaihtelevat (Microsoft Learn)

Lähes jäsentynyt tieto puolestaan on sellaista tietoa, joka ei jäsentymättömän tiedon kaltaisesti mene valmiiseen muottiin – tauluun, mutta on kuitenkin jokseenkin rakentunut säännönmukaisesti. Tällainen tieto voi olla esimerkiksi jäsentyneen tiedon tapaisesti asiakastietoa, mutta oliot voivat vaihdella ominaisuuksiltaan (Kuva 3.). (Microsoft Learn).

2.3 Tiedon prosessointi

Tiedon prosessoinnilla tarkoitetaan raakatiedon muuttamista mielekkääksi tiedoksi. Tieto voidaan käsitellä reaaliaikaisesti reaaliaikaisena tietovirtana tai eräajoina, riippuen siitä kuinka se saapuu sitä käsittelevään järjestelmään. Reaaliaikaisesti prosessoitu tieto käsitellään sananmukaisesti heti kun tieto on saapunut. Reaaliaikaisen tiedon prosessointi on hyödyllistä tapauksissa, joissa uutta tietoa syntyy jatkuvasti ja tuoreimman tiedon saatavuus on kriittistä. Tällaista tietoa on esimerkiksi osakemarkkinoiden synnyttämä tieto. Joukkoprosessointi sen sijaan on tiedon käsittelyä tietoryppäissä. Tieto käsitellään vasta kun kaikki saatavilla oleva tieto on saapunut järjestelmään. Prosessointi voidaan määrittää tapahtuvan ajastetusti tai jonkun tapahtuman perusteella, esimerkiksi kun oikea määrä tietoa on saapunut. (Microsoft Learn).

2.4 Tiedon analysoinnin kategoriat

Microsoft Azurella on useita tiedon analysointiin tarkoitettuja palveluita. Palvelut eroavat toisistaan siinä, minkälaista analysointia ne tiedolle tekevät. Tiedon analysointi onkin eräänlainen kattotermi koko prosessille. Analysoinnin tapahtumaketjuun kuuluvat tiedon tunnistaminen, tiedon puhdistus sekä muokkaus ja lopulta mallinnus uuden tiedon löytämiseksi. (Microsoft). Analysoinnin kategoriat ovat kuvaileva, diagnostinen, ennustava,

preskriptiivinen eli ohjaileva ja kognitiivinen (Microsoft Learn). Tiedon puhdistamisella tarkoitetaan prosessia, jossa raakatiedosta poistetaan virheellisiä, korruptoituneita tai väärin muotoutuneita tietoja sekä kaksoiskappaleita. Muokkaus sen sijaan tarkoittaa tietojen rakenteen tai muodon muuttamista toisenlaiseksi. (Tableau).

Kuvaileva analyysi pyrkii vastaamaan kysymykseen, mitä on tapahtunut menneiden tapahtumatietojen perusteella. Diagnostinen taas vastaa kysymykseen miksi jokin on tapahtunut etsimällä menneiden tapahtumien joukosta poikkeavuuksia, yhdistämällä poikkeavuudet ja lopulta hyödyntäen tilastollisia tekniikoita, jotta poikkeavuuksien yhteys ja siten selitys löytyisi. (Microsoft Learn).

Ennustava analyysi nimensä mukaisesti pyrkii vastaamaan kysymykseen, mitä tulevaisuudessa tapahtuu tunnistamalla menneiden tapahtumien tietojen joukosta trendejä sekä määrittelemällä tapahtumien toistuvuuden todennäköisyys. Preskriptiivinen analyysi puolestaan auttaa hahmottamaan, mitä toimenpiteitä on tehtävä tavoitteiden saavuttamiseksi. Preskriptiivisen analyysin tekniikat perustuvat koneoppimisstrategioihin, jossa suurien tietojoukkojen seasta pyritään hahmottamaan tapahtumien kaava. Analysoimalla menneiden tapahtumien ja päätöksien yhteyttä, voidaan todennäköisyys erilaisille tuloksille määritellä. (Microsoft Learn).

Kognitiivinen analytiikka sen sijaan pyrkii löytämään yhteyden olemassa olevasta tiedosta ja trendeistä, tekemään niistä johtopäätöksiä perustuen tämänhetkiseen tietopohjaan ja sitten lisäämällä niiden perusteella tehdyt löydökset tietopohjaan tulevaisuuden varalle luoden siten oppia kerryttävän kehän. Kognitiivinen analytiikka pyrkii vastaamaan kysymyksiin, mitä saattaa tapahtua, jos olosuhteet muuttuvat ja kuinka tilanteita voidaan hallita. (Microsoft Learn).

Huolimatta siitä, minkä tyylistä analysointia tiedolle halutaan tehdä, tieto on ensin kerättävä yhdestä tai useammasta lähteestä. Tietoa mahdollisesti halutaan myös puhdistaa tai muuttaa sen muotoa, lisätä siihen jotakin tai muuten valmistella. Lopulta useista lähteistä yhdistetty tai vain yhdestä lähteestä tuotu tieto ladataan halutulle alustalle, esimerkiksi tietokantaan tai tietovarastoon. Azuressa tämän tapahtumaketjun hoitaa Data Factory -palvelu. (Microsoft Learn).

2.5 Data Factory V2 -palvelu

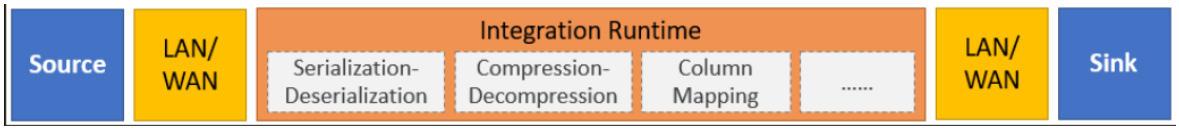
Kuva 4. Data Factory orkestroi tiedonkäsittelyä kokonaisvaltaisesti (Microsoft)

Data Factory V2 on palvelimeton Microsoft Azuren pilviympäristöissä toimiva, skaalautuva monimuotoinen koodivapaa analytiikkapalvelu. Sen avulla orkestroidaan tiedonsiirtoa ja muutosta eri tietolähteiden sekä pilvessä olevien palveluiden välillä, ja julkaistaan tieto lopulta hyödynnettäväksi (Kuva 4.). Yksinkertaistetusti Data Factory V2 yhdistää erilaisia järjestelmiä, joilla tietoa voidaan käsitellä kokonaisvaltaisesti tiedonkäsittelyprosessin jokaisessa vaiheessa. Prosessit ovat monimutkaisia yhdistelmiä ETL:stä ja ELT:stä. (Microsoft).

ETL tulee englannin kielen sanoista Extract, Transfrom ja Load. Vapaasti suomennettuna lyhenteellä tarkoitetaan tiedon hakemista sen alkulähteeltä, muokkausta käsiteltävään muotoon ja lopulta latausta tietovarastoon. Toinen malli on ELT, jossa Transform- ja Load-vaiheet tulevat eri järjestyksessä. Tässä tapauksessa raakatieto, ladataan tietovarastoon tai -järveen ennen sen muokkausta. (A Cloud Guru). Raakatieto on tietoa, jota ei olla vielä käsitelty tiettyä tarkoitusta varten (Red Hat). Tiedon integroiminen sen sijaan tarkoittaa tiedon hakemista ja yhdistämistä useasta eri lähteestä. (A Cloud Guru). Palvelimettomalla palvelulla puolestaan tarkoitetaan, ettei käyttäjän tarvitse määritellä tarvittavaa infrastruktuuria käyttääkseen palvelua. Palvelu skaalautuu automaattisesti tarvittavalla tavalla. Tämä vapauttaa kehittäjän keskittymään omien sovellustensa kehittämiseen. (Microsoft).

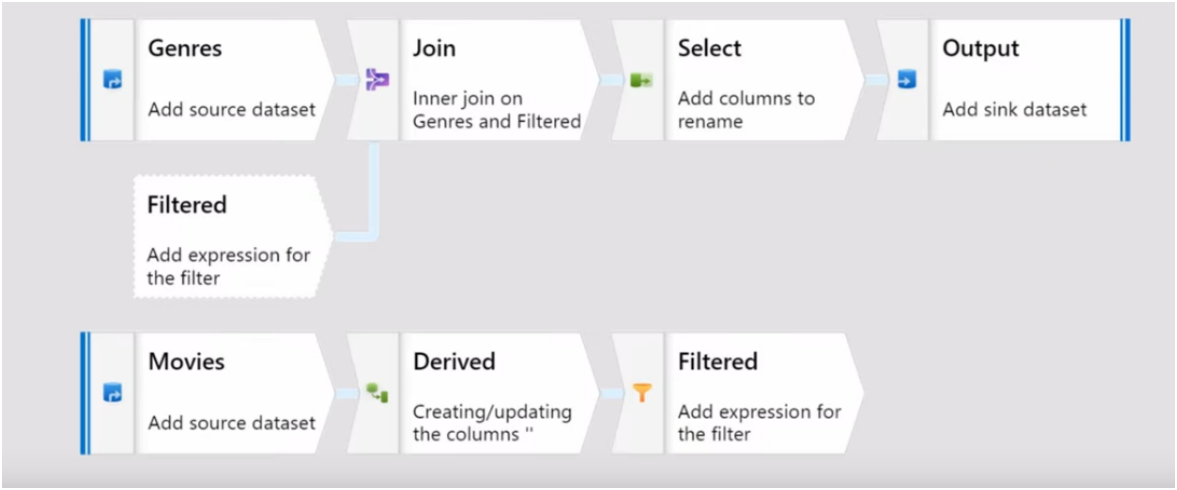
Data Factory -palvelua voidaan käyttää Microsoft Azuren portaalissa graafisella käyttöliittymällä, mutta myös kirjoittamalla JSON-koodia paikallisesti ja käyttöönottamalla se Data Factoryssa PowerShellilla (How 2020, luku 3). JSON tulee sanoista JavaScript Object Notation. Se on kevyt tekstipohjainen tietojenvaihtoformaatti, jonka perusominaisuuksiin kuuluvat helppolukuisuus ja -kirjoitus. (JSON). PowerShell puolestaan on eri alustojen välinen automaatioratkaisu. Se koostuu komentorivityökalusta, komentosarjakielestä sekä kokoonpanohallintatyökalusta. Se on yhteensopiva useiden käyttöjärjestelmien kanssa. (Microsoft). Tämä opinnäytetyö on toteutettu Data Factory V2:lla, joka on päivitetty versio Data Factory V1:stä.

2.5.1 Data Factoryn keskeiset toiminnot



Kuva 5. Copy data -toiminnon tapahtumasarja (Microsoft Learn)

Data Factorylla tietoa voidaan kopioida lukuisista eri lähteistä, niin pilvestä kuin paikallisista tietovarastoista copy data -toiminnolla eli kopiointitoiminnolla. (Kuva 5.) Kopiointitoiminto lukee tiedon sen alkulähteeltä, suorittaa sarjoittamisen eli objektin muuttamisen tavuvirraksi tai käänteisesti muuttaa tavuvirran objektiksi, pakkaa tai purkaa tiedon sekä määrittää jokaisen tietokentän oikealle sarakkeelle. Lopulta tieto kirjoitetaan kohdesijaintiinsa. (Microsoft Learn).



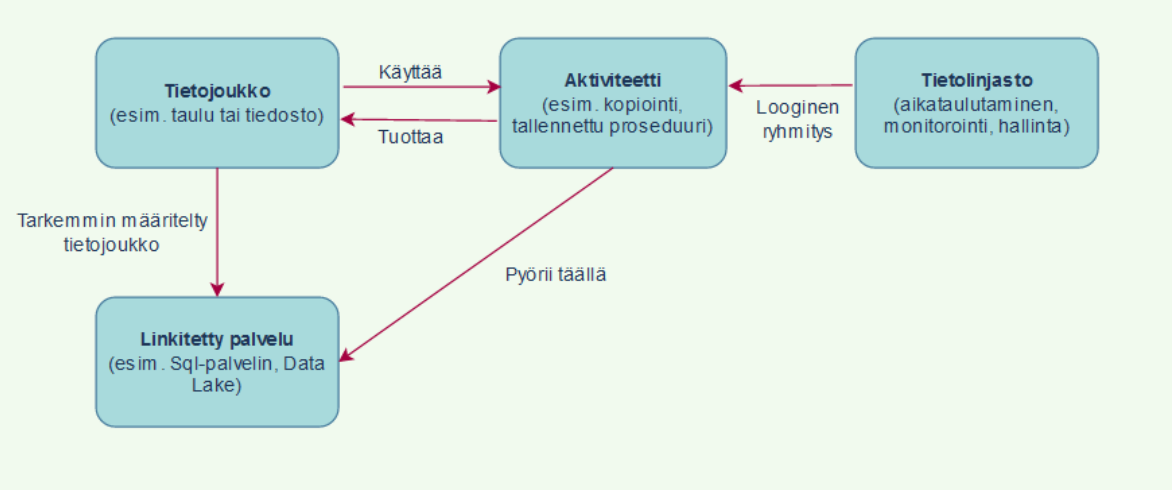
Kuva 6. Esimerkki Mapping Data Flow -toiminnosta, jossa tietoa käsitellään yksityiskohtaisesti piirtoalueella (Marczak marraskuu 2019, 0:53 min.)

Data Factoryn keskeisiin toimintoihin kuuluu myös muun muassa Mapping Data Flow, jonka avulla tiedon muokkauksen logiikan määrittäminen voidaan tehdä yksityiskohtaisesti vaihe vaiheelta UI-pohjaisella avustajalla raahaamalla toimintoja piirtoalueelle ilman ohjelmointitaitoja (Kuva 6.). Ilman tätä ominaisuutta tiedon tallennusmuotoa voitaisiin kyllä muuttaa, mutta varsinainen käsittely olisi suoritettava jonkin ulkoisen laskentapalvelun avulla. Mapping Data Flow -ominaisuudella laskenta tapahtuu automaattisesti Data Factoryn hallinnoimassa Databricks-klusterissa eikä käyttäjän tarvitse ymmärtää mitään Databrickistä. (How 2020, luku 3). Azure Databricks perustuu avoimen lähdekoodin hajautettuun Apache Spark -käsittelyjärjestelmään. Se on Azure-ympäristöön optimoitu Big Data -tiedon analytiikka-alusta. (Marczak, elokuu 2019, 0:32 min). Big Data -termillä tarkoitetaan tietoa, jota on paljon, sen määrä kasvaa voimakkaasti ja tiedon ominaisuudet voivat vaihdella suuresti (Oracle).

Kun Data Factory on jalostanut raakatiedon, voidaan se seuraavaksi ladata jollekin analytiikkamoottorille. Tällaisia palveluita Microsoft Azurella on esimerkiksi Azure Synapse Analytics, joka on yhdistelmä tietovarastointia sekä Big Data -analytiikkaa (Microsoft Learn). Tietovarasto on eräänlainen tiedonhallintajärjestelmä, jonne organisaatio voi kerätä tuottamaansa tietoa eri lähteistä ja jota hyödyntää BI-prosessissa (Guru99). BI-prosessi taas on yhdistelmä tiedon analysointia, visualisointia sekä raportointia yritystoiminnalle päätöksen teon tueksi (OmniSci).

Lisäksi Data Factory tukee jatkuvaa integraatiota sekä jatkuvaa toimitusta. Jatkuva integraatio tarkoittaa, että jokainen tehty muutos testataan automaattisesti mahdollisimman pian. Jatkuva toimitus puolestaan seuraa tätä testausta ja työntää muutokset joko staging-tilaan tai suoraan tuotantoon. (Microsoft Learn). Staging-tilalla tarkoitetaan eräänlaista luonnostilaa, jossa seuraavan version tiedostot odottavat käyttöönottoa (Javapoint). Jatkuvan integraation sekä jatkuvan toimituksen avulla voidaan ETL-tapahtumasarjojen kehittäminen ja toimittaminen suorittaa asteittain ennen julkaisua Azure DevOps -palvelulla tai GitHub-versionhallintaohjelmalla (Microsoft Learn).

2.6 Data Factoryn komponentit



Kuva 7. Toimintalogiikka Data Factoryssa (mukaillen Microsoft Learn)

Data Factory koostuu viidestä ydinkomponentista. (Kuva 7.) Komponentit työskentelevät yhdessä käyttäjän määrittelemällä tavalla, ja niiden avulla luodaan kokonaisvaltaisia tiedonsiirron ja -käsittelyn tapahtumaketjuja. Tietolinjastot koostuvat tietoaineistoista, joiden avulla määritetään data, jota toiminnassa halutaan käyttää sekä toiminnallisuuksista, jotka puolestaan määrittävät itse toiminnon. Yhteys ulkopuolisiin lähteisiin otetaan määrittelemällä linkitetty palvelu. Tämän lisäksi tietolinjastojen toiminta voidaan automatisoida toimimaan herättäjien avulla. (Microsoft). Jokainen komponentti voidaan konfiguroida Azuren portaalin kautta tai skriptata paikallisesti, eli komentosarjoittaa, ja ottaa käyttöön Data Factoryssa Powershellin tai Git-versionhallintaohjelman avulla (How 2020, luku 3).

2.6.1 Linkitetyt palvelut

Jotta Data Factorylla voidaan ottaa yhteys ulkopuoliseen resurssiin, tulee yhteys määrittää linkitetyn palvelun kautta. Linkitetyt palvelut ovat käyttäjän luomia yhteyksiä eri palveluiden välillä. Luodut yhteydet sisältävät kaiken tarvittavan tiedon yhteyden muodostamiseksi sekä tiedon välittämiseksi, kuten pääsytiedot resursseihin sekä yhteysmerkkijonon. Linkitettyjä palveluita voidaan luoda myös Azuren laskennallisten palveluiden välille tiedon tallennuksen ja haun lisäksi. Tällaisia palveluita ovat esimerkiksi Databricks tai Synapse Analytics. Määrittämällä linkitetty palvelu voidaan tiedonkäsittelynprosessit luoda Azuren toisissa palveluissa ja sitten ottaa ne osaksi Data Factoryssa ajettavaa tietolinjastoa. (How 2020, luku 3.)

Jokainen luotu linkitetty palvelu on yhteys yhteen resurssiin. Jos siis esimerkiksi tietoa halutaan kopioida Azure SQL Database -tietokantapalvelusta Azure Data Lake -tietojärvipalveluun, tulee molemmille määrittää omat linkitetyt palvelut. Sen lisäksi, että linkitettyjä palveluita voidaan määrittää eri Azuren tiedon varastointi- sekä laskentapalveluihin, voidaan linkitetty palvelu luoda myös täysin Microsoftin ulkopuolisiin resursseihin, esimerkiksi AWS S3 -varastointipalveluun tai paikallisiin tiedostojärjestelmiin virtuaalikoneiden sisällä. (How 2020, luku 3). AWS on Azuren kaltainen maailmanlaajuinen pilvipalveluiden tuottaja.

Linkitetty palvelu voidaan luoda myös dynaamisesti, niin että tarvittaville tiedoille annetaan parametrit. Mahdollinen skenaario dynaamisen yhteyden tarpeelle olisi esimerkiksi sellainen, jossa Data Factorylla halutaan yhteys useaan eri tietokantapalvelimeen ja ainoa yhteyksiä erottava asia olisi palvelimen nimi. Sen sijaan, että jokaiselle yhteydelle määritetään oma linkitetty palvelu, määritetään vain yksi ja palvelimen nimi parametrisoidaan. Tällöin hallittavana olisi vain yksi linkitetty palvelu useiden sijaan. (Microsoft).

Linkitetyn palvelun luonnissa tulee käyttäjän syöttää arkaluontoisia tietoja, kuten järjestelmien salasanoja ja käyttäjätunnuksia. Azuren linjaamien parhaiden käytäntöjen mukaan Data Factoryn kanssa suositetaankin ottamaan käyttöön Azuren Key Vault -avaintenhallintapalvelu. Key Vault -palvelun avulla Data Factoryssa tarvittavia pääsytietoja hallitaan siten yhdessä paikassa, jolloin tietoturvan hallinta tehostuu. Lisäksi yhteydenottoprotokollat voivat vaihdella Microsoftin ulkopuolella oleviin palveluihin, jolloin jokaiselle linkitetylle palvelulle tarvitaan eri tiedon palaset yhteyden määrittämiseksi. Key Vault -palvelulla tällaisten tietopalasten hallinnointi helpottuu ja tekee Data Factoryn käytöstä sujuvampaa. (How 2020, luku 3).

Jos käytössä on dynaamisesti toteutettu linkitetty palvelu, on hyvä muistaa, ettei salasanojen parametrisointi ole parhaiden käytäntöjen mukaista. Dynaamisen yhteyden tapauksessa hyvä käytäntö on tallentaa salasana Key Vault -palveluun ja parametrisoida käytössä oleva salaisuuden nimi. (Sqlitybi).

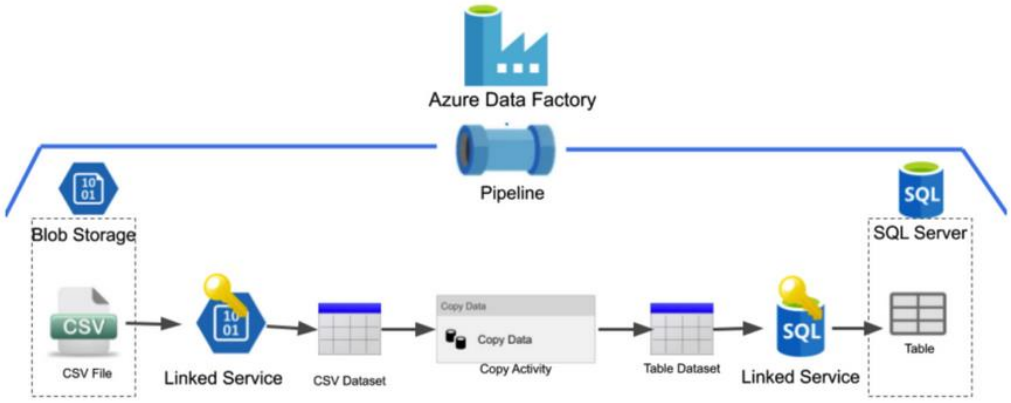
2.6.2 Tietokokoelmat

Tietokokoelmat kuvaavat tiedon rakennetta, jota käsitellään joko syötteinä tai tulosteina (Microsoft Learn). Linkitetyn palvelun avulla saavutetaan yhteys tiedon lähteeseen ja kohteeseen, mutta sen lisäksi yhteyttä on vielä tarkennettava. Tietokokoelmat lisäävät yhteyteen logiikan, jonka avulla määritetään tiedon sijainti yksityiskohtaisemmin, esimerkiksi tiedoston tai tietokantataulun tarkkuudella (How 2020, luku 3). Toisin sanoen siinä missä linkitetty palvelu on yhteys tietokantaan, on tietokokoelma yhteys tiettyyn hakemistoon tai tauluun. Tiedon integroinnissa tietokokoelma on käytössä silloin, kun tietoa haetaan lähteestä. Tämän jälkeen se siirtyy linkitetylle palvelulle, joka ohjaa sen taas toiselle tietokokoelmalle, kohdekokoelmalle.

Tietokokoelmia voidaan käyttää myös tiedostojen pakkaamiseen ja purkamiseen kopiointiaktiviteetin jälkeen. Lisäksi niiden avulla voidaan lukea tiedostojen metatietoja. (How 2020, luku 3). Metatiedot ovat tietoja itse käsiteltävästä tiedosta. Tällaisia tietoja ovat esimerkiksi, milloin tiedosto on luotu, kuka sen on luonut, milloin sitä on muokattu, tiedoston nimi ja niin edelleen. (Dataedo). Metatietojen lukuominaisuuden avulla voidaan siis lähdetiedostoa luettaessa määrittää käytettävät tiedoston ominaisuudet, kuten tiedostomuodon tai sarakkeenerottimen, ja kirjoitettaessa tietoja kohdetietokokoelmaan määrittää käytettävät ominaisuudet uudelleen. Ominaisuutta hyväksikäyttäen voidaan esimerkiksi lukea tekstitiedostoja tiedon alkulähteeltä ja kopioida ne sitten pilvivarastoon optimoidussa muodossa, esimerkiksi Parquet-tietotyyppinä. (How 2020, luku 3).

Data Factoryssa myös tietokokoelmat voidaan toteuttaa dynaamisesti. Tämä tarkoittaa sitä, ettei tietokokoelmalla tarvitse olla valmista mallia eikä sen ominaisuuksia tarvitse määritellä ennalta. (Sqlkover). Sen sijaan dynaamisella tietokokoelmalla on parametrit. Data Factoryssa parametreillä välitetään ulkoisia arvoja. Hyödyntämällä parametrejä tietokokoelmien kovakoodauksen sijaan voidaan samaa kokoelmaa käyttää useasti. Parametrisointia voidaan esimerkiksi käyttää tiedostojen nimissä, yhteyttä määrittävissä parametreissä, kuten tietokannan nimessä sekä lisäämällä päivämäärä tiedostonimen loppuun sekä monessa muussa skenaariossa. (Marczak). Mahdollinen skenaario olisi esimerkiksi sellainen, että yhdestä paikasta halutaan kopioida monta taulua. Sen sijaan että jokaiselle kopioitavalle taululle määritettäisiin oma lähdetietokokoelma sekä kohdekokoelma, toteutetaan tietokokoelmat dynaamisesti, jolloin ei tarvita kuin kaksi kokoelmaa mahdollisten lukuisten kokoelmien sijaan.

2.6.3 Tietolinjastot



Kuva 8. Tietolinjastot ovat ajettavien toimintojen looginen ryhmittymä (Productive/edge)

Tietolinjastot ovat Data Factoryn toiminnan ydin ja niitä voi olla yhdessä Data Factory -yksikössä useampia. Linjastot muodostavat ajettavien toimintojen loogisen ryhmittymän ja yhdessä tietolinjastossa useampi toiminto muodostavat yhden tehtävän. Esimerkiksi (Kuva 8.) yksi tietolinjasto voisi sisältää yhteydenoton linkitetyllä palvelulla tietokantaan ja valittuun tauluun, tiedon lukemisen lähdetietokokoelmasta ja kirjoittamisen kohdetietokokoelmaan sekä lopulta tiedon latauksen kohteeseen linkitetyn palvelun kautta. Toiminnot tietolinjaston sisällä voidaan määrittää suoritettavaksi samanaikaisesti tai peräkkäin. (Microsoft).

2.6.4 Toiminnot

Data Factoryn toiminnot ovat korkeasti määritettyjä JSON-objekteja ja jokainen tällainen objekti muodostaa yhden toiminnon. Toimintotyyppejä on Data Factoryssa monenlaisia ja ne voidaan laajasti jakaa neljään kategoriaan: 1) ulkoiset laskentatoiminnot, 3) sisäiset toiminnot, 3) iterointi- ja ehtotoiminnot sekä 4) verkkotoiminnot. (How 2020, luku 3).

Ulkoisia laskentatoimintoja ovat toiminnot, jotka tapahtuvat Data Factoryn ulkopuolella. Tällaisia ovat muun muassa laskennat Azure Databricks tai HDInsight -palveluissa, mutta myös esimerkiksi tallennetut proseduurit yhteydessä olevan SQL-palvelimen kautta. (How 2020, luku 3). Tallennetut proseduurit ovat tietokantaan talletettuja SQL-kyselyitä, joita voidaan käyttää loputtomasti uudelleen (W3schools). Yhteyden muodostus ulkoisiin laskentapalveluihin muodostetaan määrittämällä linkitetty palvelu. (How 2020, luku 3).

Sisäiset toiminnot nimensä mukaisesti tapahtuvat Data Factoryn sisällä. Mahdollisesti eniten käytetty sisäinen toiminto on aikaisemmin esitetty kopiointitoiminto. Muita sisäisiä toimintoja ovat esimerkiksi poistotoiminto, jossa tietokokoelman avulla määritetään poistettavat tiedostot, sekä myös aiemmin esitetty Mapping Data Flow -toiminto. (How 2020, luku 3).

Iterointi- ja ehtotoiminnot ovat yleisiä ohjelmointitoimintoja, joiden avulla aineistoa on mahdollista käsitellä kokonaisvaltaisesti. Tällaisia ovat muun muassa ForEach-iterointitoiminto, jota käytetään aineiston läpikäymiseen sekä if-ehtolause, jonka avulla voidaan määrittää jokin tapahtuma tapahtuvaksi, jos määritetty ehto täyttyy tai on täyttymättä. Mutta myös esimerkiksi Get Metadata -toiminto, joka palauttaa muunneltavissa olevan listan tiedoston tai hakemiston metatiedoista, ja Lookup-toiminto, jonka avulla tiedostoa voidaan lukea sen lähteeltä ja välittää luetut arvot seuraavalle toiminnolle. (How 2020, luku 3).

Verkkotoiminnot ovat ulkoisia tapahtumia, mutta ne luokitellaan omaksi kategoriakseen koska niiden avulla ei suoriteta raskasta laskentaa eikä suurten tietomäärien siirtoa. Ne ovat tapa kutsua REST API -rajapintaa. (How 2020, luku 3). REST API on sovellusohjelmointirajapinta, jonka välityksellä tietoja voidaan välittää JSON-muodossa ja se noudattaa REST-arkkitehtuuria (Red Hat).

2.6.5 Toiminnan herättäjät

Herättäjät ovat tapa, joilla tietolinjastojen ajo voidaan määrittää tapahtuvaksi automaattisesti jonkin tapahtuman, kellonajan tai säädettävän aikaikkunan perusteella. Herättäjiä voidaan käyttää myös manuaalisesti. Manuaalinen herättäjän käyttö on hyvä vaihtoehto silloin, kun jo valmista tietolinjastoa halutaan testata tuotanto-olosuhteissa. Kun tietolinjasto toimii toivotulla tavalla, voidaan herättäjä määrittää toimimaan automaattisesti. (How 2020, luku 3).

2.6.6 Pilvilaskennat

Integration runtime (IR) on Azuren laskentainfrastruktuuri. Se on skaalautuva pilvilaskentaresurssi, joka hoitaa raskaan työn, kun tietoja kopioidaan paikasta toiseen Azuressa tai, kun toimintoja Data Factorysta ohjataan muille Azuren laskentaresursseille. Azure Integration Runtime on Azuren oletusarvoinen IR ja on useimmissa tapauksissa oikea vaihtoehto. Ollessaan valittuna, jää tarvittava laskentamäärä Azuren määritettäväksi. Kaksi muuta laskentaresurssia ovat Self-hosted Integration runtime sekä laskentaresurssi SSIS-paketeille. (How 2020, luku 3). SSIS tulee sanoista SQL Server Integration Service ja se on alusta yritystasoiselle tiedon integraatiolle sekä tiedon muokkaukselle (Microsoft).

Kun tietoa halutaan kopioida eri verkkojen ja pilvessä olevan resurssin välillä käytetään Self-Hosted Integration Runtime (SHIR) -laskentaresurssia. SIHR konfiguroidaan virtuaalikoneelle ja yhteys paikallisen sekä pilvessä olevan resurssin välillä reitittyy sen kautta. Data Factory -palvelu on altis useille julkisille IP-osoitteille ja SHIR-laskentaresurssia käytetään tietoturvallisista syistä. Ilman SHIR:n käyttöä tulisi paikalliseen verkkoon määrittää saapuva yhteys, jolloin tulisi samalla heikennettyä tietoturvan parhaita käytäntöjä. (How 2020, luku 3).

2.7 Milloin käyttää Data Factorya?

Data Factorylla voidaan tuoda suuria tietomääriä eri lähteistä ja yhdistää se tavalla, jonka avulla voidaan saada uutta arvokasta tietoa organisaation ja sen sidosryhmien kannalta, ja joka muuten saattaisi jäädä huomaamatta. Esimerkiksi koululaitos ja sen opiskelijat tuottavat jatkuvasti kaikenlaista tietoa koulun eri järjestelmiin usealla eri tavalla, kuten navigoimalla opintosivustoilla, klikkailemalla, laatimalla henkilökohtaisia lukujärjestyksiään tai osallistumalla Moodle-tentteihin. Data, jota näistä toiminnoista syntyy, on mahdollisesti hyvin erilaista keskenään ja usein hyödytöntä sellaisenaan. Mutta yhdistämällä hajallaan oleva tieto oikealla tavalla voidaan opiskelijoista ja heidän käyttäytymisestään saada selville jotain mikä hyödyttää kokonaisvaltaisesti koko koululaitosta ja sen jäseniä.

Organisaation arvioidessa olisiko Data Factoryn käytöstä hyötyä, voidaan päätöksen teon tueksi miettiä neljää perustetta: 1) onko organisaatiolla paljon niin kutsuttua Big Dataa, eli suuria määriä eri rakenteista tietoa, kuten klikkaustietovirtaa, lokitietoja tai kuvia? 2) Löytyykö talon sisältä ohjelmointitaitoisia työntekijöitä? 3) Sijaitseeko organisaation tieto useassa järjestelmässä: pilvessä sekä paikallisissa palvelimissa? 4) Pystyykö organisaatio hallitsemaan erillään olevia komponentteja tiedon integroimiseksi? (Microsoft Learn).

Ensimmäinen kohta kysyy, onko tiedon integrointipalvelulle lainkaan tarvetta? Jos organisaatiolla ei ole useita tietolähteitä ei Data Factoryn käytölle ole välttämättä perustetta. Mutta, jos tietoa on paljon ja organisaatio on kiinnostunut hyötymään tuottamastaan datasta erityisesti muiden Azuren palveluiden avulla, voi Data Factoryn käytölle löytyä peruste. Tällaisia palveluita ovat esimerkiksi Azure Machine Learning, joka keskittyy hallitsemaan koneoppimisprojektien elinkaarta, tai Azure Power BI, jonka avulla voidaan luoda interaktiivisia analyysiraportteja organisaation tuottamasta tiedosta. (Microsoft Learn).

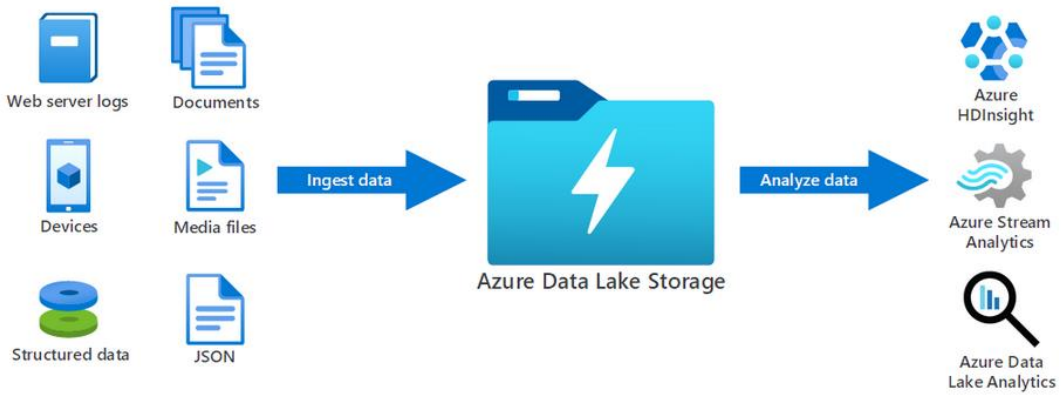
Toinen arviointiperuste on ohjelmistotaitoinen henkilökunta. Data Factoryn käyttö ei edellytä ohjelmointitaitoja. Data Factoryn laatimis- ja valvontatyökalulla voidaan tietolinjastoja luoda graafisesti raahaamalla ja pudottamalla aktiviteettielementtejä suunnittelualustalle. Lisäksi kopiointityökalun avulla tietolinjastot voidaan luoda täysin avusteisesti eikä niiden laatiminen vaadi ymmärrystä edes aktiviteettielementeistä tai niiden toimintaperiaatteista. (Microsoft Learn).

Kolmantena perusteena on organisaation tarve käyttää tietoja useasta eri lähteestä ja sijainneista. Data Factoryssa on yli 90 vakioliitintä, jotka helpottavat yhteyden muodostamisen sekä tiedon integroimisen eri lähteistä. (Microsoft Learn). Tällaisia lähteitä ovat esimerkiksi toisten pilvialustayritysten palvelut, kuten AWS ja Google, sekä lukuisat muut järjestelmät.

Neljäs kohta kysyy, onko organisaatiolla voimavaroja luoda, hallita tai ylläpitää erillisiä tiedon integrointikomponentteja? Jos ei, on Data Factory harkitsemisen arvoinen palvelu sen

loputtoman skaalautuvuuden sekä palvelittoman luonteensa vuoksi. (Microsoft). Kaiken voi aina rakentaa itse, mutta liiketoiminnassa kulujen optimointi on oleellista ja pilviteknologioiden tarjoamat hyödyt sen lukuisten myönteisten ominaisuuksien vuoksi, kuten toimintavarmuuden, joustavuuden sekä maksa vain käytöstä -menettelyn takia, saattaa organisaatio hyötyä suuresti niin kuluissa kuin tietotaidon hallitsemisessa valitessaan pilvipalvelun itse rakentamisen sijaan.

2.8 Data Lake Gen2 -palvelu



Kuva 9. Data Lake Gen2 -palveluun voidaan syöttää erilaisia tietotyyppejä ja tietoja voidaan välittää sen kautta muille analytiikkapalveluille (Microsoft Learn)

Data Lake Gen2 on Azuren pilvipalvelu suurten tietomäärien tallennukseen ja se on suunniteltu tukemaan Big Data -analytiikkaa, eli suurten jäsentymättömien ja jatkuvasti lisääntyvän tiedon analysointia. Se on paikka kaikenlaiselle tiedolle, jäsentyneelle, lähes jäsentyneelle ja jäsentymättömälle, pienille ja isoille, joukkoprosessoidulle ja reaaliaikaiselle tietovirralle. Koneoppiminen ja keinoäly hyödyntävät juuri tällaista yhdistelmätietoa, joista osa on tuotu joukkoina ja osa reaaliaikaisena, yhdessä pilvestä löytyvien loputtomasti skaalautuvien laskentaresurssien kanssa. Tiedon tallentaminen Azuren Data Lake Gen2 -palveluun mahdollistaa tiedon hyödyntämisen myös muista pilvipalveluista käsin, kuten Azuren Synapse Analytics -analysointialustan kautta. (Kuva 9.). Data Lake Gen2 -palveluun tieto tallentuu sen alkuperäisessä muodossaan, yleensä blob-objekteina tai tiedostoina. (Microsoft Learn). Blob-tiedostot ovat suuria binääritiedostoja, joita käytetään erityisesti kuvien, musiikin tai multimediatiedon tallennusmuotona (GeegsforGeegs). Tietoa Data Lake -palvelussa pidetään sen alkuperäisessä muodossa odottamassa, kunnes niitä tarvitaan johonkin käyttöön (Microsoft Learn). Tietoa voidaan hyödyntää analysointitarkoituksiin, milloin vain tai ei koskaan. Tietoja voidaan käyttää myös useasti, mutta jos tietoa on jo jollakin tavoin jalostettu vaikeuttaa se sen uudelleen käyttöä tulevaisuudessa. (Red Hat).

Data Lake eroaa tavallisista tiedon varastointijärjestelmistä siten, että sinne voi tallentaa kaikenlaista tietoa eikä sitä tarvitse käsitellä mitenkään sopivaksi etukäteen (Microsoft Learn). Data Lake Gen2 -palvelun käyttö ei juuri vaadi ylläpitoa, koska käyttäjän ei tarvitse huolehtia tarvittavista palvelimista. Käyttäjä vastaa ainoastaan tiedon laadusta, sen rakenteesta ja hallinnosta. Tietoa ei myöskään koskaan tarvitse Data Lakesta poistaa, ellei joku painava syy niin vaadi, esimerkiksi GDPR-tietosuoja-asetus. (How 2020, luku 5). Tässä opinnäytetyössä käytetään Data Lake Gen2 -palvelua, joka on yhdistelmä edeltävää Data Lake Gen1 -palvelua sekä Azure Blob Storage -palvelua.

2.8.1 Teknologia Data Lake Gen2 -palvelun takana

Data Lake Gen2 on ominaisuuksiltaan yhdistelmä Data Lake Gen1 ja Blob Storage -palveluita. Toiminnallisesta näkökulmasta kaikki kolme palvelua ovat melko saman kaltaisia, mutta erovaisuuksia kuitenkin on. Data Lake Gen2 on päivitetty versio Data Lake Gen1 -palvelusta, mutta enimmäkseen Data Lake Gen2 -palvelun teknologia kuitenkin perustuu Blob Storage -palveluun. Tämä näkyy esimerkiksi alhaisissa hintakustannuksissa ja muun muassa oletusarvoisesti tieto hajautetaan maantieteellisesti eri palvelimien välille. (How 2020, luku 5).

Data Lake Gen1 on Azuren ensimmäinen tietojärvipalvelu ja se on rakennettu käyttämään Apache Hadoop -tiedostojärjestelmää (HDFS) sekä WebHDFS REST API -rajapintaa (How 2020, luku 5). Apache Hadoop on avoimen lähdekoodin ohjelmisto suurten tietojoukkojen hajautettuun käsittelyyn (Apache Hadoop). Apache Hadoop -tiedostojärjestelmän ja sen REST API -rajapinnan käyttäminen mahdollistavat sen, että Data Lake Gen1 -palvelu on helposti integroitavissa myös muihin näitä teknologioita ymmärtäviin palveluihin, kuten Apache Spark tai Hive -palveluihin (How 2020, luku 5). Apache Spark on hajautettu Big Data -kehys, joka auttaa suurten tietomäärien erottamisessa ja käsittelyssä analyyttisia tarkoituksia varten. Apache Hive taasen on avoimen lähdekoodin hajautettu tietokanta, joka toimii Hadoop-tiedostojärjestelmässä. (Logz). HDFS-tiedostojärjestelmän hienoutena on se, että se pystyy tallentamaan minkä tahansa tyyppisiä tai kokoisia tiedostoja eikä Data Lake Gen1 -palvelulla siitä syystä ole ongelmia minkään kokoisten tai laisten tiedostojen käsittelyssä. Lisäksi tiedot tallennetaan hajautetusti usealle eri tallennuspalvelimelle, jolloin tietoja voidaan lukea useasta paikasta saman aikaisesti ja Data Lake -palvelun päällä toimivat erilaiset laskentaresurssit, kuten Hive tai Spark, pystyvät toimimaan mahdollisimman tehokkaasti. (How 2020, luku 5).

Azure Blob Storage -palvelu on pilviratkaisu suurten, jäsentymättömien tietomäärien tallennukseen (Microsoft). Blob Storage -palvelu käyttää myös HDFS-pohjaista tiedostojärjestelmää niin kuin Data Lake Gen1 -palvelu, mutta yleisen WebHDFS Rest API -rajapinnan sijaan se käyttää omaa Azure Storage API -rajapintaa. Erona näiden kahden teknolo-

gian välillä on myös tapa, kuinka tiedostot ja kansiot on toteutettu. Data Lake Gen1 -palvelussa kansiot ovat oikeita kansioita, siten että ne ovat itsenäisiä objekteja järjestelmässä hierarkkisen tiedostojärjestelmän tapaan. (How 2020, luku 5). Hierarkkinen tiedostojärjestelmä on menetelmä, miten levyt, kansiot, tiedostot sekä muut tallennuslaitteet on järjestetty ja näytetty käyttöjärjestelmässä (Computer Hope). Blob Storage -palvelussa sen sijaan tiedostot tallennetaan objekteina säilöön, jolla on tasainen nimiavaruus (How 2020, luku 5). Nimiavaruus on konsepti, jossa kaikkien objektien nimet on oltava yksiselitteisesti ratkaistavissa, jolloin yksi nimi voi esiintyä vain yhden kerran yhdessä tilassa (Microsoft).

Vaikkakin kansiot voidaan toteuttaa Blob Storage -palveluun virtuaalisesti käyttämällä osaa objektin nimestä, ei kansioiden konseptia ole oikeasti palvelussa olemassa. Tämä näkyy siinä, että jos käyttäjä luo tyhjän kansion ja navigoi pois siitä, ei kansiota enää löydy. Näin siksi, että ei ole olemassa objektia, jonka nimi olisi osa kansion nimeä. Hyötynä Blob Storage -palvelulla Data Lake Gen1 -palveluun nähden on myös tietojen varmuuskopiointi usean eri Azure-käytettävyysalueen välillä, esimerkiksi maantieteellisesti toiselle mantereelle. Data Lake Gen1 -palvelu sen sijaan tallentaa varmuuskopiot samalle käyttöalueelle, minne palvelu on otettu käyttöön. (How 2020, luku 5).

Data Lake Gen2 -palvelu onkin eräänlainen Blob Storage ja Data Lake Gen1 -palveluiden liitto. Siinä yhdistyy hierarkkinen tiedostojärjestelmä tiedon varastointialustalla, edullisilla kustannuksilla ja tiedon varmuuskopioinnilla oletusarvoisesti toiselle käytettävyysalueelle. Hierakkinen tiedostojärjestelmä mahdollistaa hakemistorakenteen toteuttamisen fyysisesti sen sijaan, että sitä vain matkittaisiin Blob Storage -palvelun tapaan, jolloin kaikki muutokset hakemistorakenteeseen edellyttävät jokaisen tallennetun objektin uudelleenkäsittelyä ja päivitystä. Data Lake Gen2 -palvelun suojaustoteutus on myös hyvin samankaltainen Gen1 -palvelun tapaan, kun kansioita ei virtualisoida. Azure Active Directory -palvelu on myös integroitu ja kullekin tiedostolle sekä kansiolle voidaan määrittää käyttöoikeudet. Lisäksi hierarkkisen nimiavaruuden myötä hakemistopäivityksestä tulee yksinkertainen muuttamalla vain metatietoja, jolloin varsinaisten tietojen käyttö yksinkertaistuu huomattavasti, mikä parantaa kyselyiden suorituskykyä. Toinen lisäys palveluun on ABFS-ohjain, joka on saatavilla kaikissa Apache Hadoop -ympäristöissä, kuten Azure Databricks ja Azure Synapse Analytic -palveluissa, ja se on erityisesti suunniteltu Big Data -analytiikkaan. (How 2020, luku 5).

2.8.2 Organisaation Data Lake

Data Lake Gen2 -palvelun käyttöönotossa olennaista on suunnitella organisaatiota palveleva hakemistorakenne. Windows-käyttöjärjestelmässä tiedostot on järjestetty siististi kansioihin, siten että latauksille, musiikille, kuville ja muille tiedostoille on jo valmiiksi osoitettu

oma kansio, jotta käyttäjä löytäisi oikean tiedon luo helposti. Lisäksi osa kansioista on tarkoituksellisesti lukittu, jotta käyttäjä ei vahingossa poistaisi jotakin järjestelmälle kriittistä. Näitä samoja periaatteita käytetään myös pilvessä, sillä ilman toimivaa tiedostorakennetta Data Lake -ympäristö on vaarana muuttua tietojärvestä tietosuoksi ja silloin siitä on käyttäjälleen hyvin vähän hyötyä. (How 2020, luku 5).

Tietojärven ensisijaisena tarkoituksena tulisi olla säilyttää tietoa sen raakamuodossa. Raakatiedolle tulisi määrittää oma hakemisto ja tietoa siellä ei koskaan tulisi ylikirjoittaa tai muuttaa. Siten käyttäjä voi aina palata alkuperäisen tiedon luo tarvittaessa. Hyvä käytäntö on myös ryhmitellä tiedot lähdejärjestelmän ja ajankohdan mukaan. Raakatiedolle osoitettu hakemisto tulisi olla myös hyvin suojattu pelkillä lukuoikeuksilla ja jokaisella käsittelyprosessilla tulisi olla kirjotusoikeus vain niihin liittyvin kansioihin, jotta tietoa ei vahingossa poistettaisi tai ylikirjoitettaisi. Muu tiedon lajitteleminen riippuu tiedon tai Data Lake -ympäristön käyttötarkoituksesta. Jos tietoa käsitellään jotenkin, tulisi myös käsitellylle tiedolle olla osoitettu oma kansio. (How 2020, luku 5). Hyvä tapa on nimetä kansiot siten, että niistä käy ilmi minkä laatuista tietoa se sisältää, esimerkiksi puhdistettuja Moodle-tietoja sisältävän kansion nimi voisi olla ”Clean Moodle Data”.

2.9 Key Vault -palvelu

Kuva 10. Azure Key Vault -palvelu säilyttää kokoonpanosalaisuudet pilvessä (Microsoft Learn, 00:07 min.)

Key Vault on Azuren palvelu salaisuuksien pilvisäilytystä varten. Salaisuudella tarkoitetaan, mitä vain minkä saavutettavuutta halutaan hallita, esimerkiksi salasanoja, API-rajapintojen avaimia, varmenteita, yhteysmerkkijonoja tai salausavaimia. (Microsoft). Key Vault -palvelu auttaa hallitsemaan sovellusten salaisuuksia säilyttämällä niitä yhdessä paikassa sekä tarjoamalla suojatun pääsyn, käyttöoikeuksien hallinnan ja käytön seuraamisen. Palvelua ei ole tarkoitettu sovellusten loppukäyttäjien tietojen tallentamiseen, vaan se

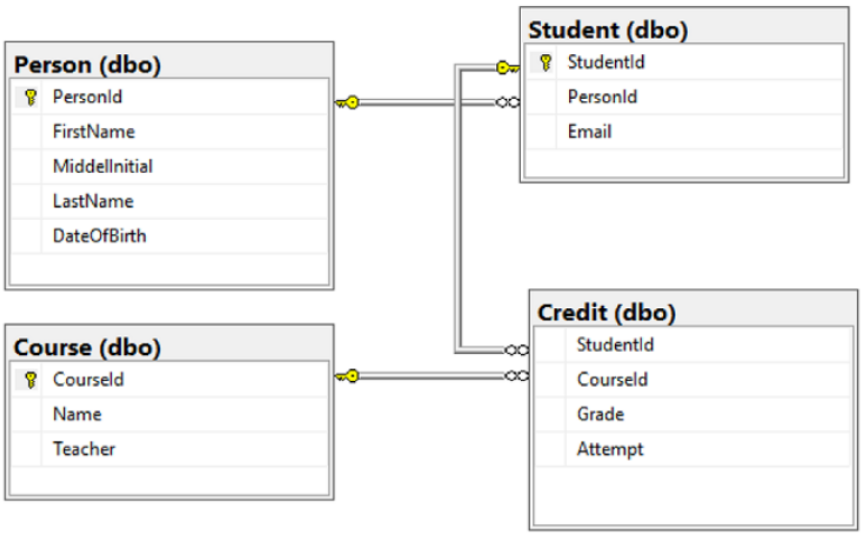
on suunniteltu tallentamaan palvelinsovellusten kokoonpanosalaisuudet. (Microsoft Learn).

Key Vault -palvelu tukee kahdenlaista säilöntätapaa: ohjelmistosuojattuja holveja sekä laitteistosuojattuja varantoja. Oletusarvoisesti Key Vault -palvelu käyttää holveja, mutta tilanteissa, joissa suojausta tarvitaan enemmän, voidaan palvelu ottaa käyttöön laitteistosuojauksella. (Microsoft). Key Vault -palvelussa salaisuus on nimi-arvo-merkkijono. Salaisuuden nimi on oltava 1–127 merkin pituinen, sisältäen ainoastaan aakkosnumeerisia merkkejä ja viivoja, ja sen on oltava ainutlaatuinen Key Vault -holvin sisällä. (Microsoft Learn).

Key Vault -palvelu käyttää Azuren Active Directory -palvelua käyttäjien ja sovellusten autentikointiin. Key Vault -resurssin käyttöoikeuskäytännöt perustuvat toimintoihin ja ne ovat aina voimassa koko Key Vault -holvissa. Toiminnot ovat Get, List ja Set. Vapaasti suomennettuna niillä tarkoitetaan salaisuuksien arvojen lukemista, salaisuuksien nimien listausta sekä salaisuuksien arvojen luomista ja päivittämistä. (Microsoft Learn). Käyttöoikeuksia määrittäessä on hyvä muistaa vähimpien oikeuksien periaate ja antaa käyttäjille ainoastaan tarvittavat oikeudet.

On Azuren parhaiden käytäntöjen mukaista luoda jokaiselle käyttöympäristölle sekä sovellukselle oma Key Vault -holvi, kuten kehitys-, testaus- ja tuotantoympäristöille. Kun holvit ovat erotettu toisistaan, myös mahdollisen hyökkääjän mahdollisuudet salaisuuksien lukemiselle pienentyvät ja siten eri käyttöympäristöt pysyvät turvattuina. Vastaavasti salaisuuksien nimien ollessa samoja sovelluksen käyttöympäristöistä riippumatta, on silloin ainoa muutettava asia holvin URL-osoite, jolloin kokoonpanoasetusten säätö vähenee. (Microsoft Learn).

2.10 Kopioitava tieto

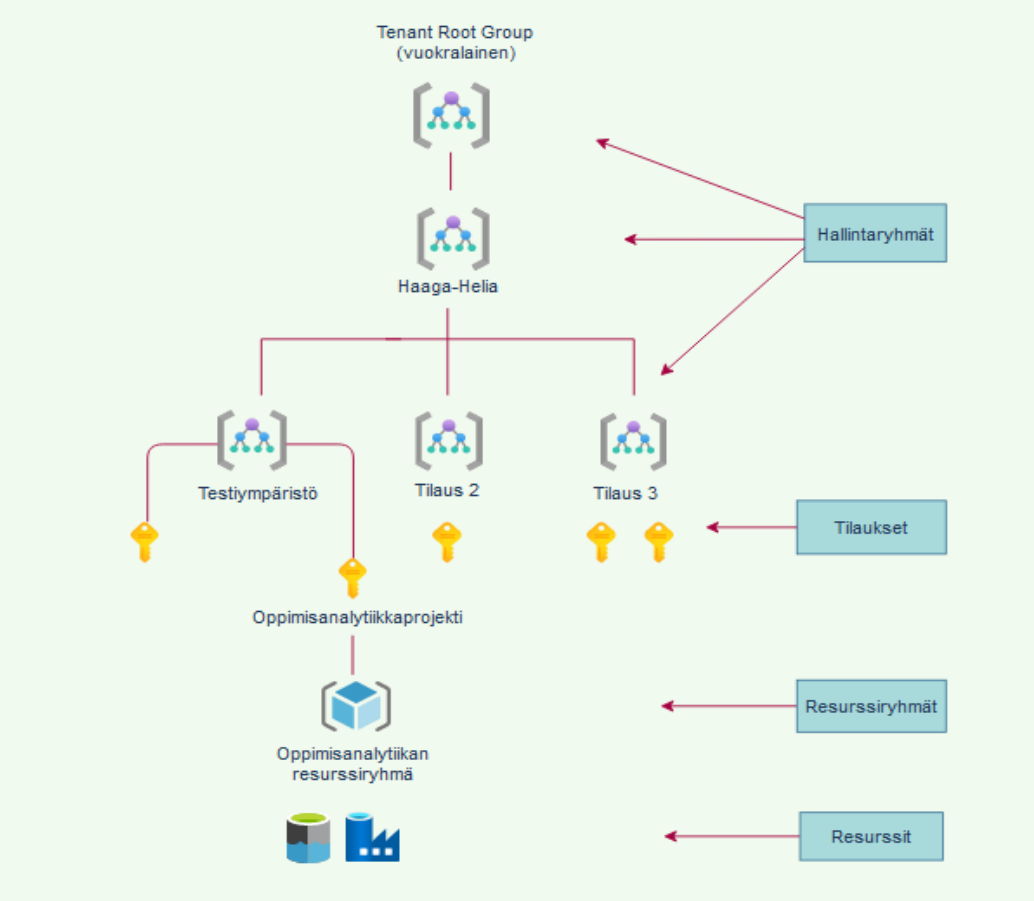


Kuva 11. SQL-oppilastietokantaesimerkki (Microsoft)

Tämä opinnäytetyö keskittyy oppilaitoksen Moodle-varmuuskopiokannan valittujen taulujen siirtoon. Tieto on SQL-tietoa, eli jäsentynyttä tietoa relaatiotietokannasta. Tietoa kopioidaan tietokannan peilikannasta, joka on varmuuskopio varsinaisesta opiskelijatietokannasta. SQL on standardisoitu kieli tietojen tallennukseen, manipulointiin ja hakemiseen tietokannasta (w3schools).

Relaatiotietokannoissa tieto on järjestetty tauluihin ja tauluissa tiedot esitetään riveillä ja sarakkeissa. Taulut ovat yhteydessä toisiinsa viiteavaimilla. Tällainen jäsentynyt tieto on tietoa, jolla on ennalta määritelty kaava, esimerkiksi Oppilas-taulu opiskelijatietojärjestelmässä voisi pitää sisällään opiskelijan yksilöivän tunnuksen lisäksi nimen, sähköpostiosoitteen, opiskelun alkamispäivämäärän ja niin edelleen. Jokaisen opiskelijan tiedot talletetaan Oppilas-tauluun siten, että aina kun uusi opiskelija kirjataan järjestelmään, syntyy Oppilas-tauluun uusi tietue pitäen sisällään juuri tämän opiskelijan tiedot. Taulut taas yhdistyvät toisiinsa viiteavaimien avulla, niin että esimerkiksi opiskelijan henkilökohtaiset tiedot löytyvät toisesta, Henkilö-taulusta, ja Oppilas-taulussa on vain henkilöä identifioiva tunnus (PersonId), joka toimii viiteavaimena viitaten toisen relaation pääavaimeen, eli Henkilö-taulun identifioivaan tunnukseen (Kuva 11.).

2.11 Azure Haaga-Heliassa



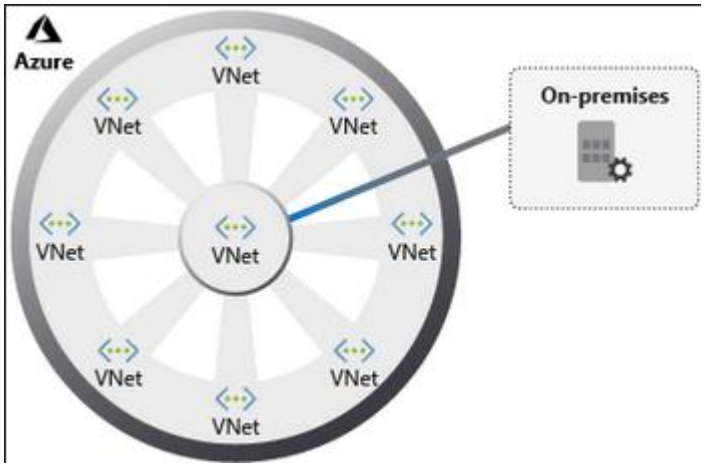
Kuva 12. Organisaation tilaushierarkia Azuressa (mukaillen Microsoft)

Azuressa organisaation ympäristön hallinta tapahtuu neljässä osassa: hallintaryhmät, tilaukset, resurssiryhmät ja resurssit (Haaga-Helia 2021) (Kuva 12.). Ylimpänä on vuokralainen (tenant), joka omistaa ja hallinnoi tiettyä Microsoftin pilvipalveluiden esiintymää. Sitä käytetään usein viittaamaan organisaation Azuren palveluihin. (Microsoft).

Hallintaryhmät ovat ikään kuin säilöjä, joiden avulla voidaan hallita useiden tilausten käyttöoikeuksia, käytäntöjä ja vaatimustenmukaisuutta. Kaikki tilaukset hallintaryhmän alla perivät automaattisesti hallintaryhmään sovellettavat säännöt. (Microsoft). Tilaukset sen sijaan kuuluvat laajemman kokonaisuuden, hallintaryhmän alle. Organisaatiolla voi olla useita hallintaryhmiä jaoteltuna esimerkiksi osastoittain. Tilaukset perivät automaattisesti hallintaryhmään sovellettavat säännöt. (Haaga-Helia 2021). Tilaukset yhdistävät loogisesti käyttäjätilit ja käyttäjätilien luomat resurssit. Resurssiryhmät puolestaan ovat loogisia säilöjä Azuren resursseille, esimerkiksi SQL-tietokannalle, Data Factory -palvelulle tai virtuaalikoneelle, tilauksen sisällä. (Microsoft).

Halutut palvelut luodaan valitun resurssiryhmän alle ja jokainen resurssiryhmä kuuluu valitun tilauksen alle. Organisaatiolla voi olla useita tilauksia omassa Azure-virtuaaliverkossaan ja tilauksia voidaan jaotella esimerkiksi projekteittain. (Haaga-Helia 2021).

2.12 Paikallinen konesali yhteydessä Azuren pilveen

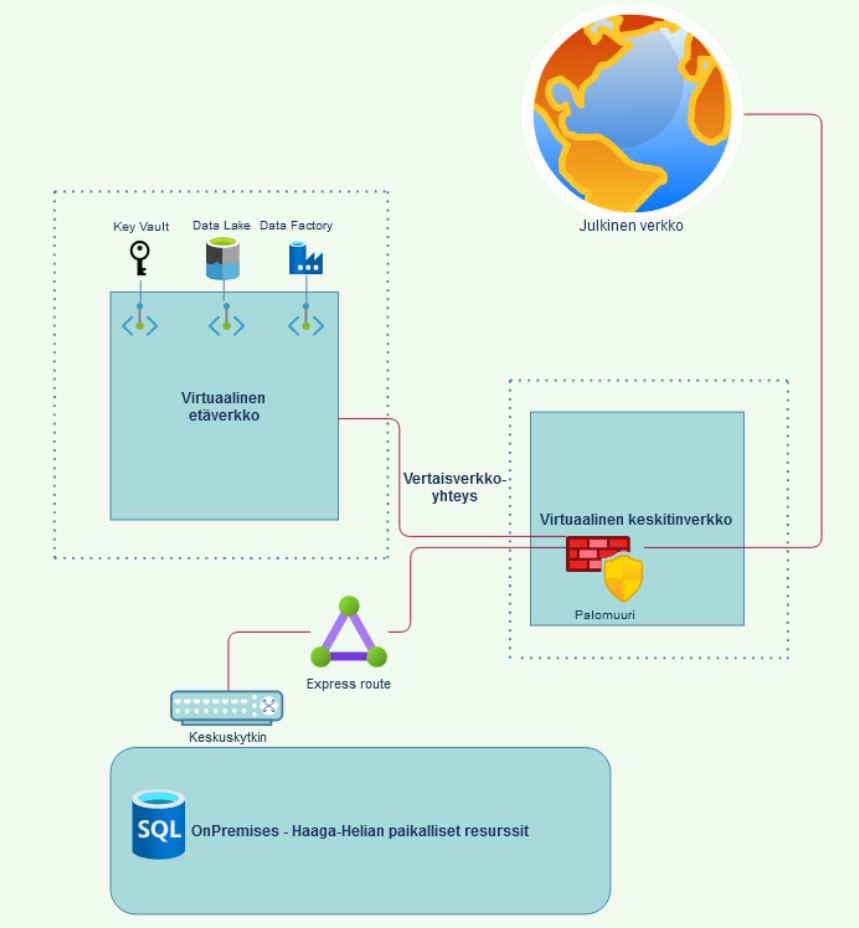


Kuva 13. Keskitin-etäverkko-malli on pyöränmuotoinen verkkoarkkitehtuuri (Microsoft)

Hub-Spoke-verkkoarkkitehtuuri, vapaasti suomennettuna virtuaalinen keskitin-etäverkko-malli, on pyöränmuotoinen verkkoarkkitehtuuri, jonka keskeltä löytyy keskitinverkko ja ympäriltä siihen liittyvät virtuaaliset etäverkot (Kuva 13.). Keskitin-etäverkko-mallin hyödyt ovat muun muassa keskitetysti hallinnoitu yhteys paikalliseen konesaliin, erillisten työympäristöjen yhdistäminen yhteiselle alueelle jaettujen palveluiden hyödyntämiseksi sekä verkkoliikenteen hallinta keskitetysti yhdessä paikassa. (Microsoft Learn). Keskitin-etäverkko-malli on yleisesti käytetty hybridipilviarkkitehtuureissa, sillä sen käyttö ja ylläpito voi

olla vaivattomampaa pitkässä juoksussa (Microsoft Learn). Hybridipilviarkkitehtuurilla tarkoitetaan ympäristöä, jossa organisaation tietoteknisetresurssit on jaettu paikallisen konesalin, julkisen ja yksityisen pilven välillä (NetApp).

Keskitin on virtuaalinen verkko, joka toimii keskuspaikkana ulkoisten yhteyksien hallinnassa sekä isännöidessä palveluita useiden työkuormien välillä. Etäverkot ovat myös virtuaalisia verkkoja ja ne isännöivät työkuormia eristetysti. Keskitin ja etäverkot ovat yhteydessä toisiinsa virtuaalisella vertaisverkkoyhteydellä. (Microsoft Learn). Virtuaalinen vertaisverkkoyhteys on yhteys, jossa kaksi tai useampi virtuaalinen verkko ovat yhteydessä toisiinsa saumattomasti siten, että ne näyttäytyvät kuin olisivat vain yksi verkko. Azuressa vertaisverkkoyhteydellä yhdistettyjen virtuaaliverkkojen liikenne kulkee Azuren runkoverkko pitkin eikä poikkea lainkaan julkiseen verkkoon. (Microsoft). Yhteys keskitinverkon ja organisaation paikallisen verkon välillä kulkee Azuren Express Route -palvelua pitkin. Azure Express Route on palvelu, jonka avulla organisaation paikalliset verkot voidaan laajentaa yksityisellä yhteydellä Azuren pilveen. (Microsoft Learn).



Kuva 14. Organisaation käyttämät Azuren palvelut sijaitsevat yksityisessä etävirtuaaliverkossa. Yhteys keskitinverkon ja organisaation paikallisen verkon välillä kulkee Azure Express Route -palvelua pitkin, keskitinverkko ja etäverkko ovat yhteydessä toisiinsa virtuaalisella vertaisverkkoyhteydellä

Tässä projektissa käyttöönotettavat Azuren palvelut sijaitsevat yksityisessä etävirtuaaliverkossa. (Kuva 14.) Etävirtuaaliverkko on yhdistetty vertaisverkkoyhteydellä keskitinvirtuaaliverkkoon, joka toimii yhteyspisteenä monien, tässä tapauksessa yhden, virtuaalisen etäverkon sekä paikallisen konesalin välillä. Keskitinverkossa on palomuuri, joka sallii liikenteen, kun osoite tunnistetaan sisäiseksi, eli paikalliseen konesaliin meneväksi. Oletuksena palomuuri estää kaiken liikenteen julkisen verkon välillä. Jos ulkoverkkoon halutaan yhteys, tehdään tarvittavat verkkoavaukset tapauskohtaisesti ja siten, että yhteyden avaava pää on Azuressa sijaitseva resurssi. (Ketonen 7.3.2021).

2.13 Private Endpoint -palvelu

Private Endpoint on Azuren verkkorajapintapalvelu. Private Endpoint käyttää käyttäjän oman virtuaaliverkon yksityistä IP-osoitetta. Verkkorajapinta yhdistää käyttäjän Azuren muihin palveluihin Azure Private Link -palvelun kautta. Private Endpoint -palvelulla tuodaan toinen Azuren palvelu ikään kuin sisälle omaan sisäverkkoon. (Microsoft). Azure Private Link -palvelun avulla käyttäjä voi yhdistyä omasta yksityisestä virtuaalisesta verkosta käsin muihin Azuren PaaS-palveluihin. Liikenne yksityisen verkon ja palveluiden välillä kulkee Azuren runkoverkkoa pitkin eikä poikkea julkiseen verkkoon. (Microsoft).

3 Tiedon kopioinnin testiprojekti

Työ toteutetaan Microsoft Azuren portaalissa. Työn vaiheet pyritään toteuttamaan Azuren yleisten parhaiden käytäntöjen mukaan. Lisäksi työhön sovelletaan Haaga-Helian hallintamallia, joka on organisaatiolle luotu, oma, parhaiden käytäntöjen opas. Työn vaiheet raportoidaan yksityiskohtaisesti, mutta niin ettei organisaation yksityisiä tietoja tule julki. Tällaista tietoa ovat esimerkiksi resurssien nimet tai IP-osoitteet. Myös Data Factoryssa luotujen tietolinjastojen, tietokokoelmien ja toimintojen nimet on muutettu.

Opinnäytetyössä kuvailtu prosessi alkaa resurssiryhmän ja resurssien luomisella Azuren portaalissa. Tämän jälkeen luodaan linkitetty palvelu Data Factoryssa, eli muodostetaan yhteys paikalliseen palvelimeen. Yhteyden muodostamisen jälkeen kuvataan Data Factoryssa luotujen tietolinjastojen logiikka sekä kopioidaan valittu tieto Data Laken kansioihin. Tiedon kopiointi on suunniteltu siten, että ensin omasta konesalista tuodaan valittu tietomäärä kokonaan. Sen jälkeen tiedonsiirto tehdään vain päivitetylle tiedolle. Data Lake -palvelussa tiedon talletukseen on loputtomat resurssit ja tieto siellä voidaan tallettaa hierarkkisen nimiavaruuden vuoksi samankaltaiseen tiedostojärjestelmään kuten tavallisesti tietokoneissa on. Ennen varsinaista testiä, eli tiedon kopiointia paikalliselta palvelimelta Azuren pilveen, tiedonsiirtoa on mallinnettu testiympäristössä. Testiympäristöön on tuotu Azuren tarjoama SQL-testiaineisto ja tiedon kopiointia on testattu Azure SQL Database -tietokantapalvelun ja Azure Data Lake Gen2 -tietojärven välillä.

Sijantina resurssiryhmälle ja kaikille sen alle luoduille resursseille valitaan kustannus- ja tietosuojasyistä maantieteellisesti lähellä oleva konesali. Käyttöönotetut resurssit ovat myös loogista sijoittaa lähelle käyttäjiään. Käyttöönotettavat resurssit ovat Data Lake Gen2, Data Factory V2, Key Vault. Lisäksi myöhemmin projektin edetessä huomattiin, että myös Private Endpoint -resurssi tuli ottaa käyttöön. Resurssit luodaan projektin puitteissa luodun resurssiryhmän alle ja ne löytyvät esimerkiksi hakemalla hakusanalla Azuren portaalissa.

3.1 Resurssiryhmän luonti tilauksen alle

Työskentely aloitettiin luomalla resurssiryhmä Azuren portaalissa halutun tilauksen alle. Resurssiryhmä nimettiin Haaga-Helian nimikäytänteiden mukaisesti, niin että nimestä selviää palvelu ja palveluiden omistajuus. Lisäksi resurssiryhmälle valittiin maantieteellinen sijainti. Resurssiryhmät voidaan jaotella toisistaan merkintöjen avulla. Merkinnät helpottavat resurssiryhmien kategorisointia ja niitä voidaan hyödyntää esimerkiksi laskutuksen seurannassa. Merkinnät ovat nimi-arvo-pareja, ja tässä tapauksessa nimi on vastuuhenkilö ja arvo on vastuuhenkilön nimi. Luodulle resurssille ei kuitenkaan annettu merkintöjä. Valinnat tarkastettiin vielä ennen julkaisua Review + create -välilehdellä. Valmis käyttöönotettu resurssiryhmä listautui sen tilauksen alle, jonne se luotiin.

3.2 Data Lake Gen2 -instanssin luonti

Data Lake Gen2 -resurssin löytää hakusanalla ’storage account’. Tiliin sovellettavat asetukset on jaettu viidelle välilehdelle. Tili liitetään oikean tilauksen ja aikaisemmin luodun resurssiryhmän alle. Instanssin luontihetkellä asetukset kulkivat järjestystä, jossa ensin määritetään instanssin perustiedot, sitten edistyneet asetukset, verkkoasetukset ja lopulta määritetään tiedon suojaus ja annetaan instanssille merkinnät.

3.2.1 Instanssin perustiedot

Resurssin luonti aloitettiin antamalle resurssille nimi, sijainti sekä suorituskyvyn ja varmuuskopioinnin taso. Data Laken nimen on oltava 3–24 merkin välillä, se voi sisältää numeroita, mutta pelkästään pieniä kirjaimia. Sen on oltava myös ainutlaatuinen. Kahta saman nimistä Storage account -instanssia ei siis Azuren sisällä voi olla. Tämä mahdollistaa sen, että instanssi on saavutettavissa mistä vain HTTP- tai HTTPS-protokollien kautta. Myös jokainen Data Lakeen tallennettu objekti saa osoitteen, joka sisältää uniikin Data Lake -instanssin nimen. Tilin sijainniksi valittiin lähellä oleva konesali ja nimeksi annettiin Haaga-Helian nimikäytänteiden mukainen nimi, jolla resurssin tunnistaa muiden resurssien joukosta.

Suorituskyvyn valintaan vaikuttaa se mihin tarkoitukseen resurssi luodaan. Tämän projektin tarkoituksena on testata ja saada kokemusta Azuresta, joten resurssit myös luodaan vastaamaan näitä tarpeita. Suorituskyvyksi valittiin ’Standard’. Standard-tason tilit pyörivät magneettista muistia käyttävien levyjen päällä Azuren konesalissa ja Premium-tason tilit tukevat SSD-asemia, eli puolijohdemassamuisteja. Azure suosittaa Standard-tason valintaa useimpiin käyttötarkoituksiin, kuten suurten tietomäärien talletukseen ja tilanteissa, joissa tilin sisältöä haetaan harvoin. Premium-taso tulee kyseeseen silloin kun Data Lake -instanssilta odotetaan alhaista viiveaikaa kaikissa tilanteissa ja tilin sisältö on usein haettu. (Microsoft).

Azure varmuuskopioi aina Data Lake -instanssien sisällön. Käyttäjä voi valita neljästä varmuuskopiointivaihtoehdoista yhden vastaamaan parhaiten instanssinsa käyttötarkoitusta. Valitsimme LRS-varmuuskopioinnin (Locally redundant storage), jolloin varmuuskopiointi tapahtuu paikallisesti saman käytettävyysalueen sisällä kuin missä itse instanssi on otettu käyttöön. Tämä on edullisin vaihtoehto ja takaa perustasoisen suojan palvelinräkkien tai palvelimien hajoamisen varalta. Tarvittaessa replikoinnin tasoa voidaan nostaa, jolloin levyn sisältö varmuuskopioidaan valitun käytettävyysalueen parina toimivan käytettävyysalueen kanssa. (Microsoft).

3.2.2 Data Lake -instanssiin sovellettavat edistyneet asetukset

Instanssin luonnissa voidaan Advanced-välilehdellä vaikuttaa instanssin turva-asetuksiin, hierarkkisen nimiavaruuden käyttöönottoon, instanssin sisällön pääsytasoon, suurten tiedostojen jaon sallimiseen sekä siihen sallitaanko levyn sisällön kryptaus jaetulla avaimella (Shared Key). Tämä instanssi luotiin oletusasetuksilla, eli siten että sallitaan turvallinen siirto, jolloin HTTP-pyynnöt evätään. Myös ’blob public access’ on oletusarvoisesti sallittu. Asetus mahdollistaa sen, että tili voidaan avata kenen tahansa saavutettavaksi. Oletuksena Azure ei tietenkään luo instansseja julkisiksi, mutta asetuksen ollessa päällä se on muutettavissa julkiseksi.

Oletusarvoisesti myös ’storage account key access’ on sallittu. Tämä tarkoittaa sitä, että pyynnöt tilille voidaan valtuuttaa Azure AD-pääsytietojen lisäksi jaetulla avaimella. Asetus jätettiin päälle. Päälle jätettiin myös TLS-asetus, jossa määritetään TLS:n (Transport Layer Security) minimitaso, joka on 1.2. Käyttöönotimme myös hierarkkisen nimiavaruuden, joka nopeuttaa Big Data -analytiikan kuormitusta sekä mahdollistaa ACL-käyttöoikeusluettelon käyttöönoton tilillä tiedostotasoisesti.

Tilin sisällön pääsytasoista on valittavissa kaksi vaihtoehtoa: kuuma ja kylmä. Kuumatason pääsy valitaan silloin, kun tilin sisältö on usein haettu ja päivittäisessä käytössä. Kylmätaso silloin, kun tili on käytössä varmuuskopiointitarkoituksissa tai tilin sisältö ei ole usein haettua. (Microsoft). Instanssille valittiin oletusarvo eli kuumataso. Oletusasetukset jätettiin myös suurien tiedostojen jakamiseen, eli sitä ei sallita.

3.2.3 Data Lake -instanssiin sovellettavat verkkoasetukset

Networking-välilehdellä määritetään instanssin yhteystavat. Oletuksena instanssi luodaan julkisella yhteyspisteellä, jolloin yhteydenotto sallitaan kaikista verkoista. Valittavissa on myös julkinen yhteyspiste, mutta niin että käyttäjä itse määrittelee sallitut verkot, ja yksityinen yhteyspiste, jolloin pääsy rajataan vain tiettyyn verkkoon. Instanssille valikoituu julkinen yhteyspiste valituista verkoista.

Kun julkinen yhteyspiste valituista verkoista on otettu käyttöön, tulee tilille määrittää verkot, joista yhteys halutaan saada. Määritimme instanssille pääsyn ainoastaan Haaga-Helian it-verkosta, joka on opiskelijoiden tai muun henkilönkunnan käytettävästä verkosta erillinen verkko. Lopuksi valitaan, kuinka liikenne reititetään Data Lake -instanssiin. Vaihtoehtoina on joko reititys Microsoftin omaa runkoverkon välityksellä tai julkista verkkoa pitkin. Instanssille valittiin Microsoftin reititys.

3.2.4 Tiedon suojaus, merkinnät ja Data Lake -instanssin käyttöönotto

Asetukset tiedon suojaamiseen valitaan Data Protection -välilehdeltä. Asetukset liittyvät tiedon palautumiseen ja versionhallintaan. Palautumisesta oletusarvoisesti päällä ovat valinnat, joissa sallitaan blob-objektien, säilöjen ja jaettujen tiedostojen palautus 7 päivää poiston jälkeen. Politiikat voidaan ottaa päältä pois kokonaan tai päivien lukumäärä, jonka aikana palautus on mahdollista tehdä, voidaan säätää halutunlaiseksi. Oletuksena päällä ovat myös versionhallinta blob-objekteille sekä niiden muutoshistoria. Luodulle instanssille valittiin oletusasetukset.

Instanssille voidaan luoda merkintä, joka on nimi-arvo-pari. Resurssien merkintä auttaa organisaatiota kategorisoimaan resursseja esimerkiksi laskutuksen avuksi. Luodulle instanssille ei annettu merkintöjä. Lopulta viimeisellä välilehdellä tarkastettiin instanssille sovellettavat asetukset ja painettiin ”create”. Kun instanssi tuli valmiiksi, listautui se sen resurssiryhmän alle, jonne se luotiin.

3.3 Data Factory V2 -resurssin luonti

Data Factory -instanssin luonnissa käydään samankaltaisia asetusvälilehtiä läpi kuin Data Lake -tilin luonnissa. Ensin käydään läpi perustiedot, sitten verkkoasetukset, kehittyneet asetukset ja merkinnät. Lopulta valinnat tarkastetaan viimeisellä välilehdellä ja jos kaikki ovat niin kuin pitää, luodaan instanssi.

Data Factory -instanssi luotiin projektin aikana useasti, sillä palvelulla ei saatu yhteyttä paikalliseen tietokantaan eikä tiedetty mistä se johtui. Selvää oli, että verkkoasetuksissa olisi jotain säädettävää. Yhteyttä yritettiin muodostaa useissa eri yhteistyötilaisuuksissa Teams-viestintäohjelman välityksellä. Lopulta monen Haaga-Helian asiantuntijoiden voimin yhteys saatiin muodostettua välityspalvelimena toimivan virtuaalikoneen kautta, jonne asennettiin SHIR-laskentaresurssi. SHIR-laskentaresurssia tarvitaan, kun tietoa kopioidaan paikallisen palvelimen ja pilvessä sijaitsevan resurssin välillä.

3.3.1 Data Factory -instanssin perusasetukset

Ensimmäisellä välilehdellä instanssille valitaan tilaus, resurssiryhmä, maantieteellinen sijainti, instanssin nimi ja versio. Tilaus ja resurssiryhmä ovat samoja kuin Data Lake -tilillä. Myös maantieteellinen sijainti on sama, eli maantieteellisesti lähellä oleva konesali. Versioksi valittiin V2, joka on päivitetty version Data Factory V1:stä. Nimeksi annettiin nimi, joka on Haaga-Helian ja Azuren nimikäytänteiden mukainen. Toisella välilehdellä Data Factory -instanssi voidaan liittää olemassa olevaan Git-tiliin tai valita Git-tilin määritys tehtäväksi myöhemmin. Data Factory -instanssi luotiin ilman Git-tiliä.

3.3.2 Data Factory -instanssin verkkoasetukset

Verkkoasetuksissa määritetään, otetaanko yhteys SHIR-resurssista Data Factory -palveluun yksityisen vai julkisen yhteyspisteen kautta. Tämä tulee kyseeseen silloin, kun Data Factory käyttää Self-Hosted Integration Runtime -laskentaa. SHIR-laskentaa käytetään, kun tietoa kopioidaan oman palvelimen ja Azuren pilvessä olevan resurssin välillä. Ymmärrys miten yhteyspisteet tulisi ottaa käyttöön tai millaista laskentaa tarvitaan ei ollut tiedossa, kun instanssia ensimmäistä kertaa määritettiin, mutta puutteet huomattiin pian, kun yhteyttä yritettiin luoda linkitetyn palvelun kautta paikalliseen tietokantaan.

Instanssia luotaessa ei siis ollut vielä tiedossa millainen yhteyspiste sille tulisi antaa tai miksi yhteyspiste tulee määrittää. Data Factory -instanssille luotiin aluksi yksityinen yhteyspiste ja liikenne määritettiin kulkevan sen kautta. Linkitettyä palvelua luotaessa yhteyttä ei kuitenkaan saatu muodostettua ja välillä yhteyspiste muutettiin myös julkiseksi. Lisäksi instanssi luotiin uudestaan varmuuden vuoksi. Lopputulemana yhteyttä ei kuitenkaan saatu luotua, sillä Data Factory -palvelulla on todennäköisesti jo DNS-määritys julkiselle yhteyspisteelle. Data Factoryn DNS-asetus tulisi siis kiertää jotenkin. Lopulta DNS-selvitys päätettiin yrittää kiertää siten, että Data Factory -palvelun ja paikallisen tietokannan väliin otettiin käyttöön virtuaalikone Windows-käyttöjärjestelmällä. Virtuaalikoneen tarkoituksena on toimia välityspalvelimena ja samalla SHIR-laskentaresurssin isäntänä. DNS on hierarkkinen nimijärjestelmä, joka kääntää tietokoneiden, palveluiden ja minkä tahansa verkossa olevan resurssin verkkonimen sitä vastaavaksi IP-osoitteeksi (Infoblox).

Otimme virtuaalikoneen käyttöön Azuressa samassa resurssiryhmässä kuin muut projektissa käytettävät resurssit. Loimme tarvittavan laskentaresurssin Data Factoryssa Manage-välilehdellä Integration Runtimes -asetusten kautta. Kun laskentaresurssi saatiin luotua, rekisteröitiin se autentikointiavaimen avulla juuri käyttöönotettuun virtuaalikoneeseen. Virtuaalikoneen host-tiedostoon lisättiin Data Factory -instanssin FQDN, eli instanssin täydellinen nimi, ja palvelun yksityinen IP-osoite, jolloin host-tiedosto toimii ikään kuin DNS-selvittäjänä Windowsin sisällä ja reitittää liikenteen paikallisen verkon ja pilvessä toimivan Data Factoryn välillä ohittaen Data Factoryn oman DNS-selvityksen. Tämä ratkaisu kuitenkin todettiin väliaikaiseksi ja tulevaisuudessa, kun Haaga-Helian ympäristöä Azuressa kehitetään enemmän, tuodaan oikeaoppisempi ratkaisu osaksi arkkitehtuuria.

3.3.3 Data Factoryn edistyneet asetukset, merkinnät ja käyttöönotto

Edistyneissä asetuksissa voidaan vaikuttaa instanssin käsittelemän tiedon kryptaukseen. Oletusarvoisesti tieto on kryptattu Microsoftin hallinnoimilla avaimilla, mutta lisäturvaa voidaan halutessa tuoda organisaation itsehallinnoimilla avaimilla. Jos itsehallinnoituja avaimia halutaan käyttää, tulee ne tallettaa Azuren Key Vault -palveluun. Emme kuiten-

kaan valitse käyttää itsehallinnoituja avaimia ja jätimme asetuksen oletusarvoksi. Merkinnät-välilehdellä instanssille voidaan määrittää nimi-arvo-pari-merkintä, mutta jätimme myös tämän resurssin merkitsemättä.

Lopulta tarkastimme valinnat, ja koska mitään herjoja ei validoinnissa syntynyt, loimme instanssin. Mutta kuten huomattiin, vaikka herjoja ei tullut Data Factoryn -instanssin luonnin yhteydessä, oli yhteyden luominen paikalliseen tietokantaan ja sitä kautta instanssin käyttäminen täysimääräisesti monen erehdyksen ja kokeilun tulos. Onneksi kuitenkaan valmista instanssia ei tarvitse välttämättä korvata uudella, vaan sen monet asetukset ovat vielä muutettavissa Azuren portaalin kautta. Me kuitenkin loimme instanssin uudelleen, kun epäselvää oli mistä yhteysongelmat johtuivat. Lopulta instanssin asetuksia vain muokattiin.

3.4 Private Endpoint -yhteyspisteen luonti

Niin kuin aikaisemmin osoittautui, tulee Azuren pilvessä pyöriville resursseille luoda Private Endpoint eli yksityinen yhteyspiste, jos palveluita halutaan käyttää yksityisestä virtuaaliverkosta käsin ilman että liikenne poikkeaa julkiseen verkkoon. Yksityinen yhteyspiste on Azure-resurssin verkkorajapinta, joka käyttää IP-osoitetta Haaga-Helian yksityisestä virtuaaliverkosta Azuressa. Yhteyspiste yhdistää Azuren PaaS-palveluihin, joita myös Data Factory ja Data Lake ovat, Azure Private Link -linkin kautta. Azure Private Link siis tuo Azuren julkisessa pilvessä pyörivät resurssit Microsoftin runkoverkkoa pitkin osaksi organisaation Azuressa pyörivää virtuaaliverkkoa. Yhteys ei mene ollenkaan julkiseen verkkoon, jolloin julkisia osoitteita ei tarvita.

Yksityisen yhteyspisteen luonnissa käydään läpi samanlaisia asetusvälilehtiä kuin aikaisempien resurssien luonnissa. Määritettävät ominaisuudet, perustietoja lukuun ottamatta, kuitenkin eroavat hieman aikaisempien resurssien asetuksista.

Loimme jokaiselle palvelulle oman yhteyspisteen, eli Data Factory, Data Lake ja Key Vault -palveluille. Yhteyspisteinstanssi liitetään oikeaan tilaukseen ja resurssiryhmään. Sen lisäksi annoimme jokaiselle kuvaavan nimen, josta käy ilmi mille resurssille yhteyspiste on. Koska kaikki resurssit, joille yksityinen yhteyspiste luodaan, löytyvät samasta Azuren hakemistosta, määritämme yhteyspisteille vain oikean tilauksen, resurssin tyypin ja itse resurssin Resource-välilehdellä. Configuration-välilehdellä yhteyspiste liitettiin oikeaan virtuaaliverkkoon ja aliverkkoon. Lisäksi, jotta yhteyspisteeseen voidaan yhdistyä yksityisesti, tulee se liittää DNS-tietueen. Loimme Azuren ehdottamalla tavalla yksityisen DNS-alueen, johon liitämme yhteyspisteet. Jätimme taas merkinnät välistä ja siirryimme suoraan tarkastamaan valinnat. Koska herjoja ei syntynyt loimme yhteyspisteet näillä asetuksilla.

3.5 Key Vault -resurssin luonti ja käyttö

Key Vault -palvelu on Microsoftin parhaiden käytäntöjen mukainen tapa sertifikaattien, salasanojen, yhteysmerkkijonojen ja salaisuuksien säilyttämiseen. Loimme Key Vault -instanssin, jotta voimme turvallisesti tallettaa paikallisen tietokannan salasanan ja yhteysmerkkijonon. Niin kuin aikaisemmat resurssit, myös Key Vault:n käyttöönotto tapahtuu samanlaisesti kuin muiden palveluiden. Palvelu liitetään osaksi oikeaa tilausta, resurssiryhmää ja sille valitaan maantieteellinen alue. Lisäksi sille annetaan kuvaava nimi. Microsoft suosittelee käyttämään jokaisella ympäristöllä ja tilauksella omaa Key Vault -säilöä, joten myös nimeäminen on hyvä tehdä tästä näkökulmasta.

Key Vault -palvelun tasoa voidaan myös korottaa valitsemalla Premium-vaihtoehto. Premium-vaihtoehto eroaa Standard-vaihtoehdosta siten, että palveluun tallennetut salaisuudet ynnä muut suojataan HSM (Hardware Security Module) laitteistotason suojauksella. Emme kuitenkaan valitse tätä. Vaihtoehto voisi kuitenkin olla paikallaan, jos toteuttaisimme jotain kriittisempää.

Perusasetuksissa myös määritetään, minkälaista politiikkaa poistojen yhteydessä sovelletaan. Key Vault sallii oletusarvoisesti matalankynnyksen poiston (soft delete), mikä tarkoittaa sitä, että poistetut objektit voidaan vielä palauttaa valitun aikaikkunan aikana. Oletuksena on 90 päivää. Lisäksi valitaan, halutaanko puhdistusturva (purge protection) kytkeä päälle. Puhdistusturvan kytkemistä päälle ei voi peruuttaa ja se voi olla päällä ainoastaan siinä tapauksessa, että soft delete on myös kytketty päälle. Kun puhdistusturva on päällä, poistetut objektit puhdistetaan vasta sitten, kunnes valittu säilytysaika on kulunut umpeen. Valitsemme oletusarvoiset asetukset, eli matalankynnyksen poiston 90 päivän aikaikkunalla ja jätämme puhdistusturvan pois päältä.

Perusasetuksien lisäksi tulee Key Vault määrittää millaista pääsypolitiikkaa se käyttää sekä minne kaikkialle Key Vault:n sallitaan pääsy. Jätimme arvot oletuksiksi sekä valitsimme pääsypolitiikan tulevan Azuressa käytettyjen roolien kautta. Lopulta määritetään, miten Key Vault -palveluun otetaan yhteys, merkinnät sekä tarkastetaan valinnat ennen käyttöönottoa. Instanssi määritettiin käyttämään yksityistä yhteyspistettä sekä yhteys Key Vault -instanssiin sallittiin valitusta yksityisestä verkosta. Jätimme merkinnät välistä, tarkastimme valinnat ja otimme palvelun käyttöön.

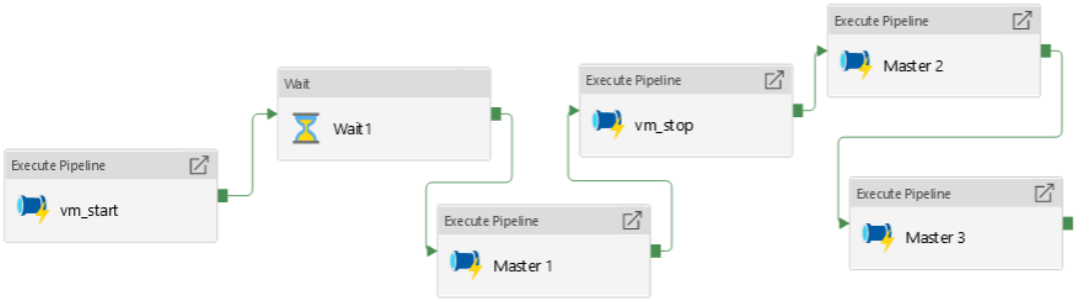
Kun palvelu oli luotu, lisäsimme sinne paikallisen tietokannan yhteysmerkkijonon. Emme kuitenkaan onnistu tekemään tätä oikein, joten luovuimme yhteysmerkkijonon käytöstä Key Vault -palvelun kautta ja lisäsimme sen manuaalisesti paikalliseen tietokantaan yhteyttä ottavaan linkitettyyn palveluun. Tietokannan salasana saatiin kuitenkin onnistuneesti lisättyä ja linkitetty palvelu toimi tällä säädöllä juuri niin kuin pitää.

3.6 Tietolinjastojen toteutus ja testaus

Data Factoryssa työnkulut muodostetaan tietolinjastoista ja tietolinjastot rakennetaan erilaisista toiminnoista. Työnkulkuja voidaan muodostaa raahaamalla haluttuja toimintoelementtejä työtilaan ja yhdistämällä ne toisiinsa nuolilla. Nuolet edustavat aktiviteettien ajojärjestystä, mutta niiden väri kertoo myös ehdosta, minkä perusteella seuraava aktiviteetti tapahtuu. Työtila projektin toteutushetkellä löytyi Data Factoryn Author-välilehdeltä. Työnkulut voidaan myös rakentaa Data Factoryn Copy Data tool -työkalulla, joka on hyvä valinta silloin kun tiedonsiirtoa tehdään ensimmäistä kertaa. Tällöin syvällistä ymmärrystä kaikista Data Factoryn komponenteista ei tarvitse vielä olla. Työnkulkuja voidaan silti muokata perinteisesti työtilassa toiminto kerrallaan, huolimatta siitä tehtiinkö ne Copy Data tool -työkalulla vai perinteisesti raahaamalla aktiviteettielementtejä.

Tämän projektin tiedonsiirtoa varten on luotu kuusi erillistä tietolinjastoa, jotka toimivat järjestyksessä. Tietolinjastot ovat kaikki yhden herättäjälinjaston sisällä: execute_master_pipelines (Kuva 15.). Tietolinjastot ovat laukaisijan takana ja toistaiseksi ne toimivat manuaalisesti, koska kyseessä on testi, mutta tarvittaessa laukaisijat voidaan määrittää toimimaan myös ajastetusti. Ensimmäisen tietolinjaston tehtävänä on polkaista prosessi käyntiin. Toisen käynnistää tiedonsiirtoa varten virtuaalikone, joka tarjoaa laskentatehon, kun tietoa kopioidaan Moodlen varmuuskopiokannasta. Tämän jälkeen, jos virtuaalikone on onnistuneesti käynnistynyt, ajetaan Wait-toiminto, joka odottaa 300 sekuntia. Odotusaika siksi, että self-hosted Integration runtime -laskentaresurssi (shir) vaatii noin 2–5 minuuttia aikaa käynnistyäkseen.

Wait-toiminnon jälkeen ajetaan ensimmäinen kopiointi, Master 1, paikalliselta palvelimelta, ja tämän jälkeen suljetaan virtuaalikone. Virtuaalikoneen onnistuneesti sulkeuduttua siirtyy työnkulku vielä viimeisille tietolinjastoille: Master 2 ja Master 3, joiden tehtävänä on siirtää tietoa kansiosta toiseen Data Laken sisällä.



Kuva 15. Execute _master_pipelines -tietolinjaston sisällä ajettavat tietolinjastot ajojärjestyksessä. Ajo siirtyy aina seuraavaan tietolinjastoon, kun edellinen on suoriutunut onnistuneesti

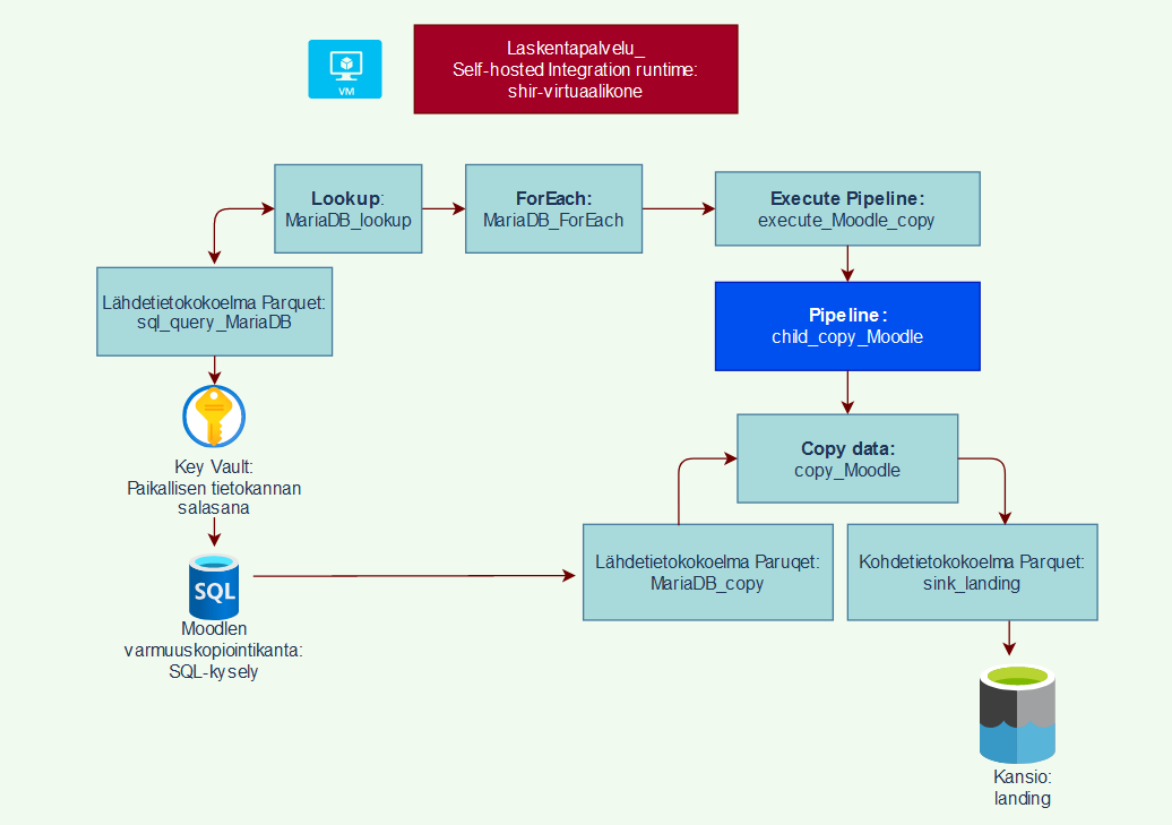
Tässä projektissa tieto on haluttu jakaa kolmeen kansioon. Ensimmäinen kansio, landing, toimii laskeutumistasona tiedon saapuessa Data Lakeen paikalliselta palvelimelta. Toinen kansio, archieve, on tarkoitettu tiedon arkistoimiseen. Ja kolmas, serving, silloin, kun tietoa Data Lakesta halutaan hyödyntää erilaisiin analysointitarpeisiin. Kansiorakenne archieve-kansiossa on toteutettu siten, että tieto järjestyy sinne kopiointiajankohtansa mukaan. Yläkansiona on vuosi, ja alikansioina kuukausi ja päivä. Päiväalikansioissa tiedot on järjestetty tauluittain. Valittu rakenne on yleinen Data Lakeen arkistointiin, mutta sitä voidaan tarvittaessa muuttaa esimerkiksi niin, että kansiot muodostuvat taulun mukaan ja yhdessä kansiossa ovat kaikki yhden taulun tiedot ajasta riippumatta.

Tiedonsiirtoa varten on luotu neljä linkitettyä palvelua, eli yhteyksiä muihin resursseihin. Yhteydet ovat Azure Key Vault, Moodlen varmuuskopiotietokanta paikalliseen konesaliin sekä kaksi yhteyttä Data Lake -tietojärveen. Toinen yhteyksistä tietojärvessä toimii shir-laskentapalvelulla, joka on asennettuna virtuaalikoneeseen nimeltä shir-virtuaalikone. Shir-laskentaa tarvitaan silloin, kun tietoa siirretään paikallisesta konesalista ja tallennetaan aikaisemmin käyttöönotettuun tietojärveen. Toinen yhteys on myös tietojärveen, mutta on käytössä silloin kun kopiointi tapahtuu tietojärven sisällä kansiosta toiseen, eli landing-kansiosta archieve- ja serving-kansioihin, jolloin laskenta tapahtuu Azuren oletus-laskentapalvelulla, eli Azure Auto Resolve Integration Runtime -laskennalla.

Tiedonsiirrossa on käytössä yhdeksän Parquet-tietokokoelmaa. Jokaisella tietokokoelmalla on oma roolinsa tiedonsiirron työnkulussa. Siinä missä linkitetty palvelu on yhteys tietokantaan, on tiedostokokoelmalla tarkemmin määritetty yhteys esimerkiksi tiettyyn tauluun tai kansioon.

Käytetyt tietokokoelmat ovat 1) sql_query_MariaDB, joka suorittaa SQL-kyselyn Moodlen varmuuskopiokantaan, 2) MariaDB_copy, joka toimii lähdetietokokoelmana tietoa kopioitaessa Moodlen varmuuskopiokannasta. Tietokokoelma kopioi taulut perustuen aikaisemmin tehtyyn SQL-kyselyyn ja lisää kaksi saraketta tauluihin lisää: latauspäivämäärän ja kopiointilähteen, joka tässä tapauksessa on Moodle. 3) sink_landing toimii kohdetietokokoelmana Moodlesta kopiotavalle tiedolle. 4) Source_landing toimii lähdetietokokoelmana ja 5) sink_archieve kohdetietokokoelmana kopioitaessa tietoa Data Laken sisällä landing-kansiosta serving-kansioon. Kohdetietokokoelmaan on määritetty kansioinnin tapahtuvan kopiointiajankohdan mukaan. 6) Delete_serving -tietokokoelma poistaa kaiken sisällön serving-kansiosta. 7) Copy_landing kopioi kaiken sisällön landing-kansiosta 8) sink_serving -kokoelmalla serving-kansioon Data Laken sisällä. Ja lopulta 9) delete_landing, joka poistaa kaiken sisällön landing-kansiosta.

3.6.1 Execute Pipeline: Master 1

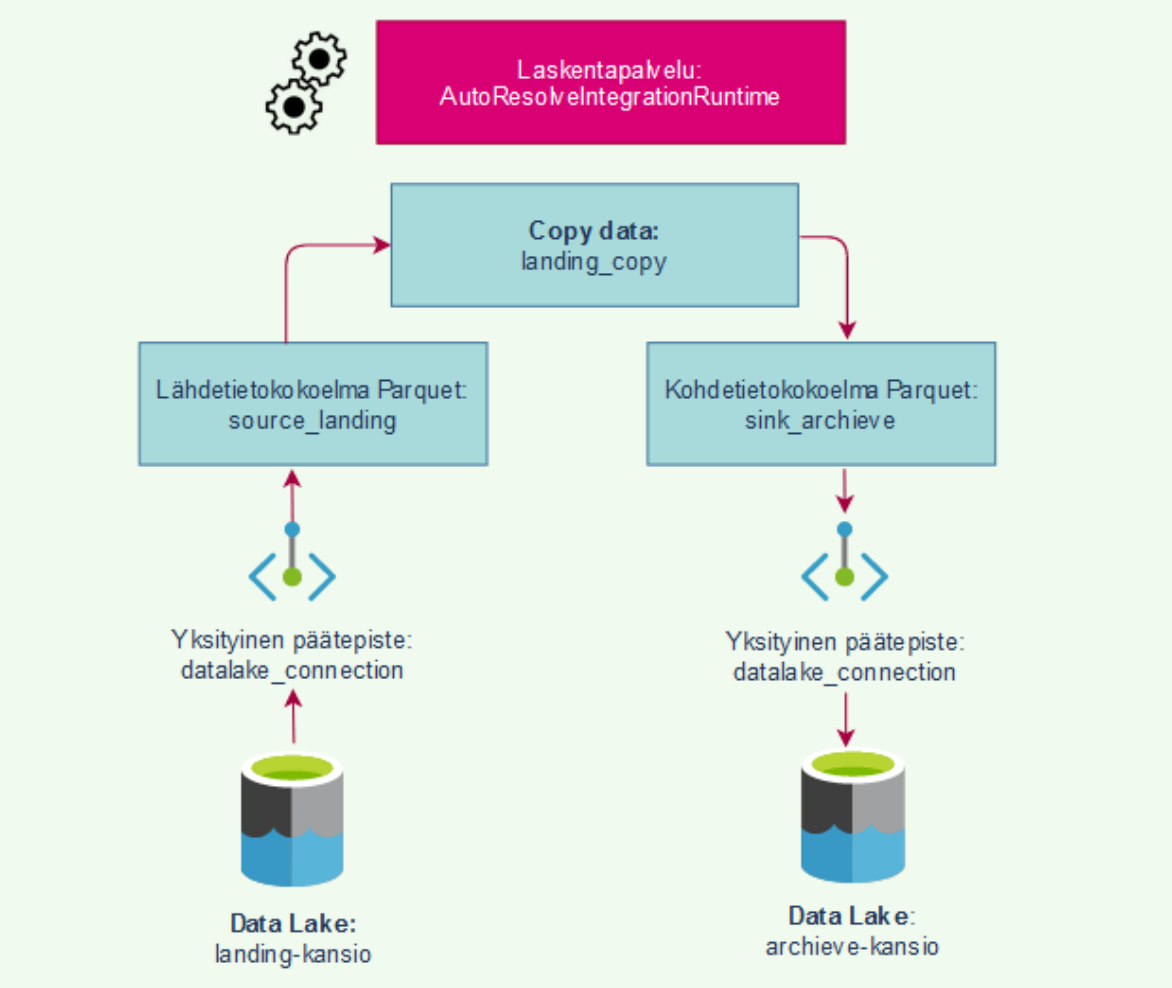


Kuva 16. Master 1 tietolinjasto herättää master_copy_Moodle -tietolinjaston, joka kopioi 12 taulua Moodlen varmuuskopiokannasta Data Lake -palveluun landing-kansioon

Master 1 on Execute Pipeline -aktiviteetti. Sen tehtävänä on siis herättää varsinainen kopiointilinjasto, eli Master_Copy_Moodle (Kuva 15.), jonka tietokokoelmassa on määritetty lookup-funktio, joka suorittaa Moodlen varmuuskopiokannassa SQL-kyselyn hakien kopioitavat taulut. Kopioitavia tauluja on 12. Käytetty tietokokoelma on dynaaminen, joten se on helposti käytettävissä uudelleen, muokattavissa tai kopioitavissa toisiin tarkoituksiin, kun tietoa halutaan Moodlen varmuuskopiokannasta. Yhteys siis pysyy samana, mutta SQL-kyselyä hieman muuttaen voidaan tietokokoelmaa käyttää esimerkiksi samaan kantaan, mutta toisiin tauluihin.

ForEach-funktion sisälle on kirjattu Execute Pipeline -aktiviteetti, joka laukaisee lapsitoimintaputken: child_copy_Moodle. Lapsitietolinjasto sen jälkeen kopioi paikallisesta SQL-varmuuskopiointitietokannasta taulut perustuen aikaisemmin tehtyyn Lookup-funktion SQL-kyselyyn. Laskennan suorittaa virtuaalikoneeseen asennettu Self-hosted Integration Runtime. Kuvaan laskenta on merkitty punaisilla nuolilla. Kun kopiointi on ajettu, siirtyy työnkulku seuraavalle Execute Pipeline -toiminnolle, joka sammuttaa laskennassa käytetyn virtuaalikoneen (Kuva 16.).

3.6.2 Execute Pipeline: Master 2

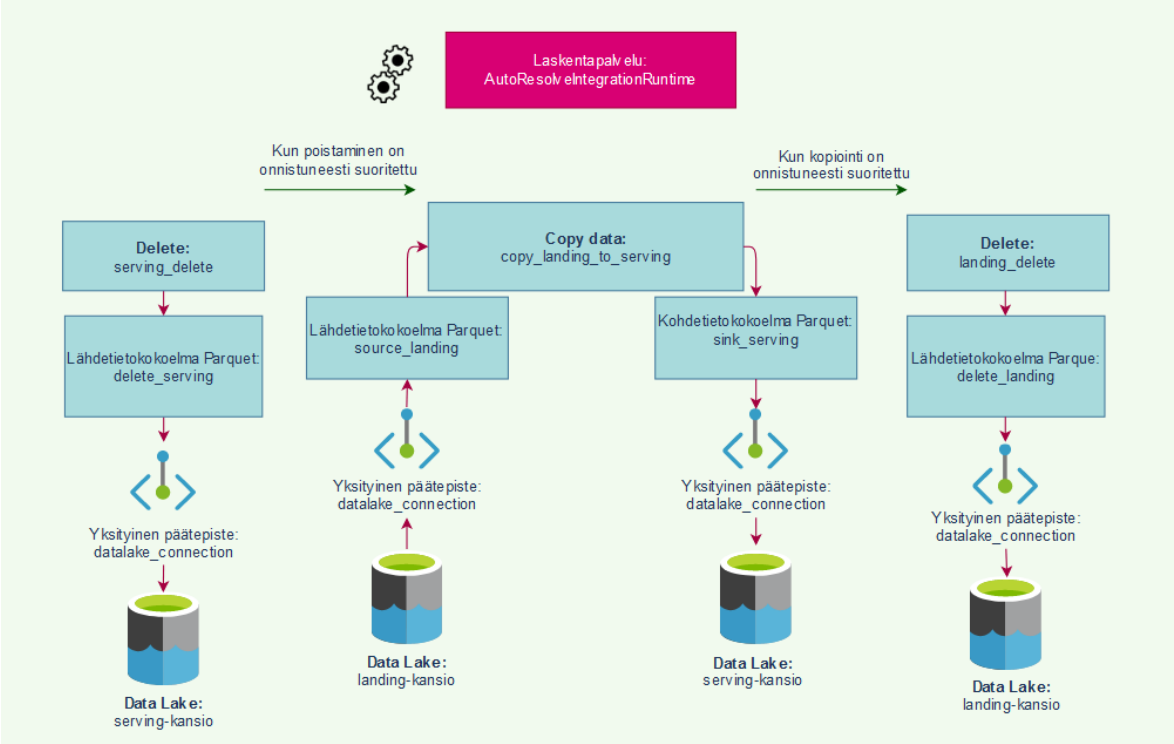


Kuva 17. Master 2 -tietolinjasto herättää landing_copy -tietolinjaston, joka kopioi landing-kansion sisällön archieve-kansioon

Kun virtuaalikone on sammutettu, ajetaan seuraava Execute Pipeline -aktiviteetti: Master 2, joka herättää landing_copy -tietolinjaston. Linjastossa kopioidaan landing-kansion sisältö archieve-kansioon. Laskennan suorittaa Azuren oletuslaskenta, eli AutoResolveInteghrationRuntime. Kuvaan laskenta on korostettu pinkillä nuolilla. Archieve-kansioon kansiointi muodostuu vuoden, kuukauden ja päivän mukaan (Kuva 17.).

3.6.3 Execute Pipeline: Master 3

Viimeisessä, delete_serving-copy_landing-delete_landing -tietolinjastossa, ajetaan ensin delete-aktiviteetti, joka tyhjentää serving-kansion, jotta sieltä löytyisi aina pelkästään tuorein tieto. Kun poistaminen on suoritettu onnistuneesti loppuun, suoritetaan kopiointi landing-kansiosta serving-kansioon. Lopuksi ajetaan vielä delete-aktiviteetti, joka tyhjentää landing-kansion. Laskennan suorittaa Azuren oletuslaskentapalvelu, eli AutoResolveIntegrationRuntime -laskenta. Kuvaan laskenta on korostettu pinkillä nuolilla. (Kuva 18.).



Kuva 18. Master 3 -tietolinjasto herättää delete_serving_copy-landing_delete_landing -linjaston

3.7 Kulujen seuranta ja kustannusten huomioiminen projektissa

Azuressa kulujen monitorointi onnistuu kätevästi portaalista. Kulut on esitetty visuaalisesti erilaisin diagrammein ja siten, että yksittäisten resurssien menot on helppo havainnoida kirkkaiden värien avulla. Tilauksen kulut muiden oleellisten tietojen ohessa on nähtävissä projektin tekohetkellä pääpiirteittäin tilauksen Overview-välilehdellä Azuren portaalissa. Kulujen seurantaan on mahdollista myös laatia Power BI -raportteja, jotka voivat tulla kyseeseen erityisesti silloin, kun kulujen seurantaa halutaan jakaa sellaisille sidosryhmille tai -henkilöille, joilla ei ole pääsyä portaaliin. Azuren kulut voidaan yhdistää Power BI Desktop -ohjelmaan Azure Cost Management -liittimellä.

Azuren kulujenseuranta (Cost Management) on maksuton palvelu. Sen tarkoituksena on ohjata palvelun käyttäjää, organisaatiota, optimoimaan resurssien käyttöä pilvessä ja tekemään siten ratkaisuja, jotka hyödyttävät kuluttajaa, mutta myös itse Azurea. Cost Management löytyy portaalista, oman resurssiryhmänsä alta omalta välilehdeltään. Kuluja voidaan seurata reaaliaikaisesti siten, että edellispäivän kustannukset näkyvät tänään ja ohjelma myös ennustaa käytön perusteella kulut tulevaisuudessa. Kulujen näkyminen visuaalisesti resurssiryhmän tai resurssin Overview-välilehdellä helpottaa kustannusten hahmottamista ja suunnittelua, sekä mahdollistaa nopean reagoinnin tarvittaessa.

Tässä projektissa tavoitellaan tietoa myös siitä, minkälainen kuluerä Azure ja sen palvelut ovat Haaga-Helialle. Onko suunta oikea ja kannattaako laajentumista pilven suuntaan ottaa isommin harppauksin. Valitut resurssit ja niiden ominaisuudet on pyritty toteuttamaan siten, että vain tarvittava on otettu käyttöön. Pilven luonteeseen kuuluu maksa vain käytöstä -periaate, joten tarvittaessa resursseja sekä suorituskykyä voidaan nostaa. Pilvessä resurssit ovat loputtomat ja niiden käyttöönotto on vain muutaman napin painalluksen päässä. Siksi mitään ei tarvitse varata ikään kuin ennakoiden varastoon. Myös ylimääräisistä resursseista on helppo luopua, jos tarvetta niille ei enää ole.

Projektissa käyttöönotetut resurssit on pyritty ottamaan käyttöön mahdollisimman pienillä kustannuksilla. Projektin toteutushetkellä keväällä 2021 siirrettävän tiedon määrät ovat pieniä ja projektin jatkosta tai kestosta ei vielä tiedetä tarkemmin. Tämä on pyritty huomioimaan resursseja varatessa. Esimerkiksi laskentaresurssia tarjoavan virtuaalikoneen ominaisuudet on pyritty huomioimaan siten, että ne vastaavat juuri tässä projektissa tarvittavaa määrää. Tarvittaessa virtuaalikoneen ominaisuuksia voidaan nostaa tai laskea Azuren portaalista resurssin omalta sivulta. Data Factoryn, jonka avuksi virtuaalikone on pystytetty, käyttämää laskentakapasiteettia voidaan seurata reaaliaikaisesti resurssin omalta Overview-välilehdeltä. Virtuaalikoneen kuluja pyritään myös rajoittamaan siten, että se on päällä ainoastaan tarvittaessa, vaikka kulut virtuaalikoneen jatkuvasti päällä ollessa ovat mahdollisesti myös hyvin pienet.

Data Laken tietojen replikointiin on valittu saman käytettävyysalueen replikointi eli LRS, joka on edullisin vaihtoehto tietojen varmuuskopioinnissa Azuressa. Raportin kirjoitushetkellä siirrettävän tiedon määrät ovat pieniä ja projektin kesto auki, joten GRS (Geo Redundant Storage) -replikointia ei tarvita. GRS-replikointi varmuuskopioisisi tiedot myös toiselle käytettävyysalueelle. Tarvittaessa replikoinnin tasoa voidaan nostaa, jolloin tiedot kahdennetaan myös toiseen konesaliin eri maantieteellisellä alueella.

Tiedonsiirrossa käytetään tiedostomuotona Parquet-formaattia, joka on edullinen ja tehokas vaihtoehto suurien tiedostomäärien pakkaamiseen. Vaikka projektin tietomäärä on pientä ja toisellakin tiedostomuodolla siirto olisi edelleen edullista, on Parquet tulevaisuuden kannalta hyvä valinta mahdollisesti tietomäärän kasvaessa.

4 Projektin tulokset

Opinnäytetyö tehtiin Haaga-Helian ammattikorkeakoulun tietohallinnossa keväällä 2021 toteutetusta projektista, jonka tarkoituksena oli kartuttaa oppilaitoksen osaamista ja kokemusta Azuren moderneista työkaluista tiedonsiirrossa sekä -säilytyksessä tietojärvessä.

Projektin vaiheet etenivät aluksi Azuren ympäristöön tutustumisella sekä teoriaa opiskelemalla. Azuren ympäristöön luotiin varsinaisen testin testitila, jonne ladattiin Azuren SQL-testiaineisto, jolla mallinnettiin tiedonsiirtoa sekä harjoiteltiin erilaisia tekniikoita, esimerkiksi metatietojen lukemista ja ehtolauseiden käyttöä. Mallia katsottiin muun muassa Paul Mitchellin YouTube-videoista ja konsultointiapua saatiin Sulava-yritykseltä.

Kun testiaineistolla oli harjoiteltu tarpeeksi, oli vuorossa virallisen testiympäristön pystyttäminen Azureen. Ympäristö ja resurssit otettiin käyttöön yhdessä monien Haaga-Helian tietohallinnon asiantuntijoiden kanssa. Ensin luotiin resurssiryhmä oikean tilauksen alle ja resurssiryhmään tarvittavat resurssit: Azure Data Lake Gen2, Azure Data Factory V2 ja Azure Key Vault -palvelut. Pian huomattiin, ettei yhteyden luominen Data Factorysta käsin käy niin suoraviivaisesti kuin oli ajateltu. Käyttöönotettujen palveluiden lisäksi piti vielä ottaa käyttöön Azure Private Endpoint -palvelu, jotta palveluita voidaan käyttää yksityisestä virtuaaliverkosta käsin ilman, että liikenne poikkeaa julkiseen verkkoon. Kun ympäristö oli toiminnassa, luotiin harjoitteluiden pohjalta tiedonsiirrossa käytettävät tietolinjastot ja seurattiin niiden ajoa. Lopuksi seurattiin myös kustannuksia ja hahmotettiin kulujen muodostumista.

Varsinaisen opinnäytetyön kirjoittaminen alkoi samaan aikaan projektin kanssa, mutta eteni aluksi hitaasti. Projektin ensimmäinen vaihe, josta myös opinnäytetyö on tehty, saatiin kokonaisuudessaan päätökseen keväällä 2021, mutta opinnäytetyöraportin kirjoittaminen jatkui suunnitelmista huolimatta pitkälle syksyyn 2021.

Projektin luonne oli testata ja pilotoida tiedonsiirtoa paikalliselta palvelimelta pilveen käyttämällä Azuren moderneja pilvityökaluja. Projektin vaiheet sujuivat hyvin ja kaikki saatiin lopulta toteutettua suunnitelmien mukaan. Mahdollisesti eniten aikaa vei Data Factorylla erilaisten tekniikoiden harjoitteleminen ja teorian lukeminen. Data Laken käytössä eniten päänvaivaa aiheutti juuri tähän projektiin sopivan tiedostorakenteen suunnittelu. Tietoa ei usein ollut vaikea löytää, mutta tiedon ymmärtäminen ja soveltaminen veivät useasti paljon aikaa. Saman tiedon äärelle piti myös palata toistuvasti. Lopulta projektissa valituksi tulleet ratkaisut vaikuttivat juuri sopivalta tämänkaltaiseen tiedon kopiointiin, jossa tarkoituksena on mallintaa tiedonsiirtoa paikallisen palvelimen ja pilvessä sijaitsevan resurssin välillä.

Projektin eri komponenttien määrittäminen on pyritty tekemään Azuren parhaiden käytäntöjen mukaan ja siten, että ne olisivat helposti muunnettavissa myös toiseen käyttötarkoitukseen. Erityisesti Data Lake -palvelun tiedostorakenne voisi toimia minkälaisen aineiston kanssa tahansa, jossa tiedon arkistoimisen lisäksi halutaan tila, josta löytää tuorein tieto analyyttisiin tarkoituksiin, esimerkiksi BI-raporttien laadintaan. Projekti onnistuikin mielestäni hyvin hahmottamaan mitä tiedonsiirto paikalliselta palvelimelta pilveen vaatii suunnittelultaan ja työkalujen hallinnaltaan. Vaikka kyseessä oli yksinkertainen kopiointi tusinalle tietokantataululle, ei eri vaiheiden hyvää suunnittelua kannata aliarvioida. Siitä huolimatta, että valitut ratkaisut ovat yleensä aina peruutettavissa tai muutettavissa, sujuu pilvityökalujen käyttö tehokkaammin ja hermoja säästävämmin, kun suunnittelutyö on huolellisesti tehty. Moni vaihe voi myös mennä pieleen ja erityisesti luottamuksellisen tiedon käsittelyssä tietoturvan varmistaminen on keskiössä. Lisäksi huolellisesti suunniteltu ja toteutettu tiedonsiirto lisää organisaation valmiutta toteuttaa vastaavanlainen projekti tulevaisuudessa suuremmassa mittakaavassa. Tiedon lisääntyessä myös tarve sen hyödyntämiselle kasvaa, joten oletettavaa on, että Data Factoryn ja Data Laken kaltaisille palveluille ja niiden hallinnalle löytyy myös tulevaisuudessa tarvetta.

4.1 Oman oppimisen arviointi

Opinnäytetyön aiheen valinta syntyi, kun olin yhteydessä Haaga-Helian tietohallintoon loppuvuonna 2020. Toivomuksena oli tehdä jotain pilvestä ja sattumoisin tällainen projekti oli alkamassa. Pilvi aiheena tuntui saattavan opinnot kokonaisvaltaisesti päätökseen, sillä pilvestä löytyy ne kaikki komponentit mitä vuosien aikana on tullut opiskeltua nykyaikaisessa paketissa. Lisäksi se vastasi henkilökohtaista opintopolkuani, jossa on yhdistetty ohjelmistotuotantoa ja ICT-infrastruktuuria. Työhön liittyi monia elementtejä näistä, kuten verkkoja, relaatiotietokantoja ja tiedon eri muotoja, ohjelmoinnin perustoiminnallisuuksia sekä tietysti tietoturvaa. Uutena asiana oli tiedon analysoimiseen tähtäävät eri toimintavaiheet, joita ei opinnoiden aikana ollut käsitelty. Kosketuspintaa tuli tietenkin myös itse projektiin osallistumisesta, sen hallinnasta ja eteenpäin viemisestä, mikä on ollut keskiössä tradenomiopinnoissa. Kaiken kaikkiaan aihe tuntui hyvin sopivalta niin omien mielenkiinnon kohteiden kuin opintojen loppuun saattamisen kannalta.

Opinnäytetyön tekeminen alkoi keväällä 2021, jolloin Suomessa elettiin vielä poikkeusaikoja maailmalaajuisesta tartuntataudista johtuen. Siitä syystä työn kaikki vaiheet on toteutettu fyysisesti etäällä muista projektiin osallistuneista tahoista ja henkilöistä. Onneksi nykyaikaisilla työvälineillä tämä ei ollut ongelma ja lisäksi toimintaympäristön ollessa pilvi, ei juuri muuta tarvittu kuin jotenkuten toimiva päätelaite sekä riittävä tietoverkkoyhteys. Olisi kuitenkin aliarvioimista sanoa, ettei tautitilanteella ja siitä johtuvalla etätyöskentelyllä olisi ollut vaikusta työhön. Toisaalta etätyöskentely on vapauttanut monista voimavaroja vaativista tehtävistä, kuten paikalle saapumisesta fyysisesti toisaalta. Mutta toisaalta viestintä

on saattanut kärsiä ja monia kommunikoinnin kannalta tärkeitä vihjeitä on saattanut jäädä huomaamatta silloin kun niitä ei ole osattu pukea sanoiksi. Epävarmuus onkin ollut usein läsnä työn eri vaiheissa. Onneksi ei kuitenkaan pelkästään epävarmuus, sillä kannustusta on tullut monelta eri suunnalta ja lähestyvä valmistuminen on motivoinut jatkamaan epämukavista ajatuksista huolimatta. Lisäksi työskentelytapa teki myös tutuksi sen mitä se myös sitten oikeassa työelämässä mahdollisesti on.

On ollut etuoikeus saada tehdä lopputyö oikeassa työympäristössä, vaikka se on myös luonut paineita onnistumisesta. Erityisesti raportin kirjoittamisen pitkittyminen horjutti usein uskoa sen loppuun saamisesta. Onneksi projektiin ja lopputyöhön liittyvät eri sidoshenkilöt ovat olleet todella rohkaisevia ja helposti lähestyttäviä: on ollut tunne, että apua voi aina pyytää ja keskustelut ovat välillä rönsyilleet erittäin kotoisasti. Hienoa myös oli, kun projektin eri vaiheisiin osallistui tietohallinnon eri alojen asiantuntijoita ja heidänkin työskentelyänsä pääsi seuraamaan. Projektin aikana olen kasvattanut huomattavasti ymmärrystäni Azuresta ja sen pilvipalveluista, jota myös toivoin opinnäytetyön aihetta valittaessa. Yhteistyö kaikkien osapuolien kanssa on sujunut hyvin ja olen projektin aikana saanut paljon kannustusta ja hyviä neuvoja.

Lähteet

A Cloud Guru. A visual guide to Azure Data Factory. Luettavissa: https://acloud-guru.com/blog/engineering/a-visual-guide-to-azure-data-factory. Luettu: 1.9.2021.

Apache. Apache Hadoop. Luettavissa: https://hadoop.apache.org/. Luettu: 28.10.2021.

Dataedo. What is metadata (with examples). Luettavissa: https://dataedo.com/kb/data-glossary/what-is-metadata. Luettu 14.9.2021.

Geegs for Geegs. BLOB Full Form. Luettavissa: https://www.geeksforgeeks.org/blob-full-form/. Luettu: 26.11.2021.

Guru99. What is Data Warehouse? Luettavissa: https://www.guru99.com/data-ware-housing.html**.** Luettu: 13.9.2021.

Haaga-Helia, 2020. Vuosikertomus 2019. Haaga-Helian ammattikorkeakoulu Oy. Luetta-vissa: https://www.haaga-helia.fi/sites/default/files/file/2020-11/haaga-helia-vuosikerto-mus-2019.pdf. Luettu: 26.3.2021.

Haaga-Helia. Tietoa Haaga-Heliasta. Luettavissa: https://www.haaga-helia.fi/fi/haaga-heli-asta. Luettu: 26.3.2021.

Hallintamalli: Microsoft Azure. Haaga-Helia ammattikorkeakoulu 2021. Luettu: 31.3.2021

How, M. 2020. The Modern Data Warehouse in Azure: Building with Speed and Agility on Microsoft’s Cloud Platform. Apress. Luettavissa: https://learning.oreilly.com/lib-rary/view/the-modern-data/9781484258231/. Luettu: 31.3.2021

Infoblox. What is Domain Name System (DNS)? Luettavissa: https://www.infob-lox.com/glossary/domain-name-system-dns/. Luettu: 26.11.2021.

Javapoint. Git Index. Luettavissa: https://www.javatpoint.com/git-index. Luettu: 9.9.2021

JSON. Introducing JSON. Luettavissa: https://www.json.org/json-en.html. Luettu 22.9.2021.

Logz. Comparing Apache Hive vs. Spark. Luettavissa: https://logz.io/blog/hive-vs-spark/. Luettu: 28.10.2021.

Ketonen, V. 7.3.2021. Järjestelmäsuunnittelija. Haaga-Helia amk. Lausunto. Helsinki.

Marczak, A. Elokuu 2019. Azure Databricks Tutorial. Data transformation on scale. Video. Katsottavissa: https://www.youtube.com/watch?v=M7t1T1Q5MNc. Katsottu: 9.9.2021.

Microsoft 18.6.2018. How does Microsoft Azure work? Video. Katsottavissa: https://www.youtube.com/watch?v=KXkBZCe699A. Katsottu: 26.3.2021

Microsoft. Azure Key Vault basic concepts. Luettavissa: https://docs.microsoft.com/en-us/azure/key-vault/general/basic-concepts. Luettu: 8.10.2021.

Microsoft. Azure Storage redundancy. Luettavissa: https://docs.microsoft.com/en-gb/azure/storage/common/storage-redundancy. Luettu: 26.11.2021.

Microsoft. Identify the need for data solutions. Luettavissa: https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-identify-need-data-solutions. Luettu: 26.5.2021.

Microsoft. Introduction to Azure Blob Storage. Luettavissa: https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction. Luettu: 26.5.2021.

Microsoft. Introduction to Azure Data Factory. Luettavissa: https://docs.microsoft.com/en-us/azure/data-factory/v1/data-factory-introduction. Luettu: 1.8.2021.

Microsoft. Namespace. Luettavissa: https://docs.microsoft.com/en-us/windows/win32/dns/name-space. Luettu: 28.10.2021.

Microsoft. Parameterize linked services in Azure Data Factory and Azure Synapse Analytics. Luettavissa: https://docs.microsoft.com/en-us/azure/data-factory/parameterize-linked-services?tabs=data-factory. Luettu: 22.9.2021.
Powers

Microsoft. Serverless computing. Luettavissa: https://azure.microsoft.com/en-us/overview/serverless-computing/. Luettu: 8.9.2021

Microsoft. SQL Server Integration Services. Luettavissa: https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15. Luettu: 7.10.2021.

Microsoft. Storage account overview. Luettavissa: https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview#performance-tiers. Luettu: 26.11.2021.

Microsoft. Tutorial: Design a relational database in Azure SQL Database using SSMS. Luettavissa: https://docs.microsoft.com/en-us/azure/azure-sql/database/design-first-database-tutorial. Luettu: 8.10.2021.

Microsoft. What is Azure? Luettavissa: https://azure.microsoft.com/en-gb/overview/what-is-azure/. Luettu: 26.3.2021

Microsoft. What is Azure Data Factory? Luettavissa: https://docs.microsoft.com/en-us/azure/data-factory/introduction Luettu: 22.04.2021.

Microsoft. What is Azure Key Vault? Luettavissa: https://docs.microsoft.com/en-us/learn/modules/manage-secrets-with-azure-key-vault/2-what-is-key-vault. Luettu: 20.10.2021.

What is Azure Private Endpoint? Luettavissa: https://docs.microsoft.com/en-us/azure/private-link/private-endpoint-overview. Luettu: 25.11.2021.

What is Azure Private Link? Luettavissa: https://docs.microsoft.com/en-us/azure/private-link/private-link-overview. Luettu: 25.11.2021

Microsoft Learn. Data Integration at scale with Azure Data Factory or Azure Synapse Pipeline. Luettavissa: https://docs.microsoft.com/en-us/learn/paths/data-integration-scale-azure-data-factory/. Luettu: 25.5.2021.

Microsoft Learn. Identify the need for data solutions. Luettavissa: https://docs.microsoft.com/en-us/learn/modules/explore-core-data-concepts/2-identify-need-data-solutions. Luettu: 8.10.2021.

Microsoft Learn. Implement a hub-spoke network topology in Azure. Luettavissa: https://docs.microsoft.com/en-us/learn/modules/hub-and-spoke-network-architecture/2-implement-hub-spoke. Luettu: 22.10.2021.

Microsoft Learn. Understand Azure Data Factory components. Luettavissa: https://docs.microsoft.com/en-us/learn/modules/data-integration-azure-data-factory/5-understand-components. Luettu: 7.10.21.

NetApp. What is hybrid cloud? Luettavissa: https://www.netapp.com/hybrid-cloud/what-is-hybrid-cloud/. Luettu: 21.10.2021.

OmniSci. Business Intelligence. Luettavissa: https://www.omnisci.com/technical-glossary/business-intelligence. Luettu: 13.9.2021.

Oracle. What is Big Data? Luettavissa: https://www.oracle.com/big-data/what-is-big-data/. Luettu: 23.9.2021.

Productive/edge. Azure Data Factory and its dynamic capabilities. Luettavissa: https://www.productiveedge.com/2020/07/27/azure-data-factory-capabilities/. Luettu: 21.9.2021

Red Hat. What is a data lake? Luettavissa: https://www.redhat.com/en/topics/data-storage/what-is-a-data-lake. Luettu: 1.10.2021.
Red Hat. What is REST API? Luettavissa: https://www.redhat.com/en/topics/api/what-is-a-rest-api#overview. Luettu: 24.9.2021

Sisense. What is Data Cleaning? Luettavissa: https://www.sisense.com/glossary/data-cleaning/. Luettu: 31.8.2021.

Sqlitybi. How to Build Dynamic Azure Data Factory Pipelines. Luettavissa: https://sqlitybi.com/how-to-build-dynamic-azure-data-factory-pipelines/. Luettu: 22.9.2021.

Sqlkover. Dynamic Datasets in Azure Data Factory. Luettavissa: https://sqlkover.com/dynamic-datasets-in-azure-data-factory/. Luettu: 14.9.2021.

TechRepublic. Microsoft Azure: A cheat sheet. Luettavissa: https://www.techrepublic.com/article/microsoft-azure-the-smart-persons-guide/. Luettu: 31.8.2021.

W3schools. Microsoft Cloud Services. Luettavissa: https://www.w3schools.in/microsoft-cloud-services/. Luettu: 31.8.2021.

W3schools. SQL Stored Procedures for SQL Server. Luettavissa: https://www.w3schools.com/sql/sql_stored_procedures.asp. Luettu: 23.9.2021

W3schools. SQL Tutorial. Luettavissa: https://www.w3schools.com/sql/. Luettu: 26.11.2021.

Liitteet

Liite 1. Opinnäytetyössä käytetyt käsitteet ja niiden merkitykset

ACL	Käyttöoikeuslista, eng. Access Control List
Active Directory	Windows-toimialueen käyttäjätietokanta ja hakemistopalvelu
Apache Hadoop	Avoimen lähdekoodin ohjelmisto suurten tietojoukkojen hajautettuun käsittelyyn
Apache Hive	Avoimen lähdekoodin hajautettu tietokanta, joka toimii Hadoop-tiedostojärjestelmässä
Apache Parquet	Kolumnipohjainen tiedon varastointiformaatti
Apache Spark	Hajautettu Big Data -kehys suurten tietomäärien erottamiseen ja käsittelyyn
API	Sovellusohjelmointirajapinta
Azure Blob Storage	Azuren varastointipalvelu blob-tiedostojen tallennukseen
Azure Databricks	Azuren alustalle optimoitu tiedon analysointipalvelu
Azure Data Factory	Azuren palvelu tiedon kokonaisvaltaiselle käsittelylle
Azure Data Lake	Azuren tiedon varastointi- ja analytiikkapalvelu
Azure Key Vault	Azuren palvelu kokoonpanosalaisuuksien tallennukseen ja tietoturvalliseen käyttöön
Azure Portal	Verkkopohjainen konsoli, vaihtoehto komentorivityökalulle.
Azure Synapse Analytic	Azuren palvelu, joka on yhdistelmä tiedon varastointia ja Big Data -analytiikkaa

Big Data	Tietoa, jota on paljon, sen määrä kasvaa voimakkaasti ja sen ominaisuudet voivat vaihdella suuresti
Blob	Binääritiedostomuoto, usein kuvaa, musiikkia tai multimediatietoa, eng. BLOB – Binary Large Object
Copy Data tool	Data Factoryn työkalu tiedon kopiointiin avusteisesti
Data Flow	Data Factoryn ominaisuus, jolla luoda graafisia datamuunnostoimintaperiaatteita, jotka voidaan suorittaa toimintoina tietolinjastojen sisällä
DNS	Hierarkkinen nimijärjestelmä, joka kääntää erilaisten verkkolaitteiden, palveluiden ja muiden resurssien ihmisluettavat verkkonimet IP-osoitteiksi
ETL/ELT	Tiedonkäsittelyprosessi, jossa tieto haetaan sen alkulähteeltä, muokataan käsiteltävään muotoon ja ladataan tietovarastoon. Kaksi viimeistä vaihetta voivat mennä eri järjestyksessä, jolloin tieto ladataan tietovarastoon ennen sen muokkausta
Fabric Controller	Hajautettu ohjelma, joka hallinnoi ja valvoo Azuren palvelimia sekä koordinoi sovellusten resursseja
FQDN	Täydellinen osoite, eng. fully qualified domain name
Hallintaryhmä	Yksikkö Azuressa, jonka alle kuuluu useampi tilaus. Hallintaryhmän avulla hallitaan tilausten käyttöoikeuksia, käytäntöjä ja vaatimustenmukaisuutta

Herättäjä	Data Factoryssa tietolinjastojen ajo automatisoidaan herättäjien avulla, eng. trigger
HTTP/HTTPS	Tiedonsiirtoprotokolla
Hypervisor	Ohjelmisto, joka luo ja ajaa virtuaalikoneita
Integration runtime	Azuren skaalautuva pilvilaskentainfrastruktuuri
JSON	Tekstipohjainen tietojenvaihtoformaatti
Keskitinvirtuaaliverkko	Pyöränmuotoinen verkkoarkkitehtuuri, jonka keskeltä löytyy virtuaalinen keskitinverkko (hub) ja ympäriltä siihen liittyvät virtuaaliset etäverkot (spoke), eng. hub-spoke network
Linkitetty palvelu	Data Factoryssa määritettävä yhteys tiedon lähteelle ja kohteelle, sekä tarvittaville lisäpalveluille, eng. linked service
Metatieto	Tietoja tiedostosta, esimerkiksi tiedoston luontipäivämäärä, tiedoston nimi, muokkaushistoria
Microsoft	Ohjelmistoalan yritys
Microsoft Azure	Microsoftin julkinen pilvialusta
Palvelimeton palvelu	Pilvipalvelun muoto, jossa käyttäjän ei tarvitse määritellä tarvittavaa infrastruktuuria käyttääkseen palvelua, eng. serverless service
Pilvi	Tietojenkäsittelyresurssien keskitetty sijainti, jotka saavutetaan Internet-verkon yli
PowerShell	Tietojenkäsittelytehtävien automatisointityökalu
Raakatieto	Käsittelemätöntä tietoa
REST API	Sovellusohjelmointirajapinta, joka noudattaa REST-arkkitehtuuria
Resurssiryhmä	Säilö resursseille Azuressa

SQL	Standardisoitu kieli tiedon tallennukseen, manipulointiin ja hakemiseen tietokannasta. Käytetään usein etuliitteenä kuvaamaan tiedon laatua, eli jäsentynyttä tietoa
Tietokokoelma	Data Factoryssa tietokokoelmat kuvaavat tiedon rakennetta, jota käsitellään joko syötteenä tai tulosteena, eng. dataset
Tietolinjasto	Data Factoryssa toimintojen looginen ryhmittymä, eng. pipeline
Tietosuo	Käytetään tietojärvistä, joiden hallinta on mennyt pieleen ja tieto on muuttunut saavuttamattomaksi, eng. dataswamp
Toiminto	Data Factoryssa toiminnot määrittävät tiedolle suoritettavan toimenpiteen, eng. activity
Vertaisverkkoyhteys	Virtuaalinen yhteys, jossa kaksi tai useampi virtuaalinen verkko ovat yhteydessä toisiinsa saumattomasti, näyttäytyen kuin olisivat vain yksi verkko, eng. vnet peering