



Karelia-ammattikorkeakoulu  
Tradenomi, Tietojenkäsittely (AMK)

# Tietoalusta digitaaliselle kaksoselle

Ville Niemi

Opinnäytetyö, Marraskuu 2021

[www.karelia.fi](http://www.karelia.fi)



**OPINNÄYTETYÖ**  
**Marraskuu 2021**  
**Tietojenkäsittelyn koulutus**

Tikkarinne 9  
80200 JOENSUU  
+358 13 260 600 (vaihde)

Tekijä(t)  
Ville Niemi

Nimeke  
Tietoalusta digitaaliselle kaksoselle

Toimeksiantaja  
Karelia AMK

**Tiivistelmä**

Opinnäytetyön aihe on tietöalustan luominen digitaaliselle kaksoselle. Yksi tärkeimmistä tavoitteista oli sopivan tietokannan valitseminen projektin käyttöön. Tietokannan oli pystyttävä tallentamaan nopeasti tulevaa IoT-dataa ja sitä piti pystyä tarkastelemaan reaaliaikaisesti. Työssä luotiin myös putki, jota pitkin tieto kulkee IoT-sensoreilta tietokantaan ja digitaalisen kaksosen käyttöön.

Opinnäytetyö tehtiin Microsoft Azuren portaalista löytyvillä palveluilla. Työhön valittiin parhaimmat sovellukset testaamalla eri toteutuksia. Testaaminen tapahtui simuloimalla oikeanlaista tietoa ja välittämällä sitä palveluille. Sen perusteella, kuinka palvelut reagoivat tietoon ja tiedonrakenteisiin, käyttöön valikoituivat IoT Hub, Time Series Insights, Azure Functions ja Digital Twins.

Tieto kulki tietokantaan IoT Hubin kautta ja tiedon rakenne on selkeä. Tietokannasta voi tarkastella yksittäisiä sensoreita, mikä oli yksi tärkeimmistä vaatimuksista tietokannalle. Digitaalinen kaksonen -palvelu aktivoitiin käyttöön ja sinne tehtiin harjoitusmalli, joka vastasi lämpötilasensoria. Mallin lämpötila muuttui aina, kun uutta tietoa saapui digitaaliselle kaksoselle.

Kieli  
suomi

Kieli  
suomi

Asiasanat  
Azure, Digitaalinen Kaksonen, Tietokanta



**THESIS**  
**November 2021**  
**Degree Programme in information technology**

Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+ 358 13 260 600 (switchboard)

Author (s)  
Ville Niemi

Title  
Information platform for the Digital Twin

Commissioned by  
Karelia University of applied sciences

**Abstract**

The topic of the thesis is the creation of an information platform for the digital twin. One of the most important objectives was to choose the right database for the project. The database had to be able to store incoming IoT data quickly and be able to view in in real-time.

The thesis was done with the services in Microsoft Azure portal. The best services were selected with testing different implementations. The testing was done by simulating the right kind of information and passing it on to the services. IoT Hub, Time Series Insights, Azure Functions and Digital Twins were selected for use based on how they reacted to data and data structures.

The data is passed on to the database through the IoT Hub. Individual sensors can be viewed from the database, which was one of the most important requirements for it. The digital twin service was activated, and a training model equivalent to the temperature sensor was created there. The temperature of the model changed whenever new data arrived for the digital twin.

Language  
Finnish

Pages 48  
Appendices 0  
Pages of Appendices 0

Keywords  
Azure, Digital Twin, Database

# Sisältö

1	Johdanto .....	5
2	Digitaalinen kaksonen .....	6
2.1	Historia .....	6
2.2	Määritelmä .....	6
3	Azure .....	7
3.1	Azuren määritelmä .....	7
3.2	IoT Hub .....	8
3.3	Stream Analytics .....	8
3.4	CosmosDB .....	9
3.5	Time Series Insights .....	10
3.6	Azure Functions .....	10
3.7	Azure Digital Twins .....	11
4.1	Visual Studio .....	11
4.2	Azure IoT Explorer .....	12
5	Opinnäytetyön suunnittelu ja toteutus .....	12
5.1	Projektin yleiskatsaus .....	12
5.2	Opinnäytetyön alku .....	13
5.3	IoT Hub .....	13
5.3.1	IoT Hub käyttöönotto .....	14
5.3.2	IoT Hub ja oikeudet .....	16
5.3.3	Laitteiden lisääminen IoT Hubiin .....	17
5.4	Ensimmäinen tietokantatesti .....	18
5.4.1	Striimauspalvelu .....	19
5.5	Aikasarjatietokanta .....	22
5.6	Tiedon simulointi .....	26
5.7	Tiedon lähettäminen aikasarjatietokantaan .....	27
5.7.1	Tietojen esittäminen graafien avulla .....	30
5.8	Digitaalisen kaksonen hyödyntäminen .....	31
5.9	DTDl kieli .....	34
5.10	Tiedon lähettämisen ongelmat .....	35
5.11	Simulaattorin vaihto .....	36
6	Tulokset .....	38
7	Pohdinta .....	38
7.1	Työn jatkaminen .....	38
7.2	IFC-malli .....	39
	Lähteet .....	40

## 1 Johdanto

Opinnäytetyön aiheena on tietöalustan luominen digitaaliselle kaksoselle. Digitaalinen kaksonen on varsin uusi termi, joten sen hyödyntäminen olisi varsin edullista Karelian digitaaliselle kehitykselle. Opinnäytetyöhön kuuluu kirjallinen osuus, jossa käsitellään yleisimpiä opinnäytetyöhön liittyviä termejä sekä käytettyjä tekniikoita. Pääpaino on kuitenkin toiminnallisella osuudella, jossa kuvien avulla selostetaan, kuinka ratkaisu on tehty ja millaiseen lopputulokseen on päästy.

Opinnäytetyö on osa Karelian projektia, jossa tavoitteena on rakentaa tietoputki erilaisilta IoT-sensoreilta tietokantaan ja siitä eteenpäin digitaaliselle kaksoselle. Opinnäytetyössä on myös hyvä ottaa huomioon tiedon kylmävarastoinnin mahdollisuus ja kolmiulotteisen mallin (IFC-malli) tuominen järjestelmän saataville esimerkiksi Unreal Enginen avulla.

Opinnäytetyössä käytetään valmiita Azuren palveluita, joista täytyi valita sopivimmat työtä ajatellen. Azuren portaalissa on todella paljon erilaisia palveluita, jotka muistuttavat toisiaan. Työn edetessä palvelut muuttuivat sitä mukaa kun niitä päästiin testaamaan simuloidun tiedon avulla.

Opinnäytetyössä selviää Azuren palveluiden monimuotoisuus ja se, mitä valintoja on tehty, jotta tietöalusta on saatu toimimaan. Opinnäytetyön avulla on mahdollista rakentaa vastaava tietöalusta IoT-sensoreille Azuren palveluita käyttäen. Työtä on myös mahdollista jatkaa eteenpäin ja selvityksessä otetaan esille esimerkkejä mahdollisista jatkotoimenpiteistä.

## 2 Digitaalinen kaksonen

### 2.1 Historia

Digitaalisen kaksosen määrittely on opinnäytetyön kannalta tärkeää, koska kyseessä on suhteellisen uusi, mutta kovaa vauhtia kasvamassa oleva käsite. Vuonna 2020 Gartnerin teknologisten trendien listauksessa hyperautomaatio, johon digitaalinen kaksonen liittyy, sijoittui ensimmäiseksi (Panetta 2019). Ensimmäisenä digitaalisen kaksosen konseptin otti käyttöön Michael Grieves 2002, mutta käsitteenä se sai alkunsa jo vuonna 1991. Silloin David Gelernter ennusti kirjassaan "Mirror Worlds", kuinka koko maailma voidaan matkia tietokoneen sovelluksen avulla. (Devasia 2020).

NASA on pitkään pohdiskellut digitaalisen kaksosen konseptia. NASA on täydellinen kohde digitaalisen kaksosen käyttämiseen, koska insinöörien valvomat laitteet sijaitsevat avaruudessa. Digitaalinen kopio reaaliaikaisella tiedolla olisi kätevä insinööreille. NASA:n John Vickers käytti termiä "Digital twin" vasta vuonna 2010. (Devasia 2020.)

### 2.2 Määritelmä

Digitaalisella kaksosella voidaan tarkoittaa digitaalista kopiota oikeista asioista, kuten ihmisistä, prosesseista, fyysisistä ominaisuuksista, järjestelmistä ja laitteista (Devasia 2020). Kopiointiprosessi voidaan tehdä molemmin päin, joko ensin oikeasta asiasta luodaan digitaalinen malli tai digitaalisesta mallista rakennetaan fyysinen kopio. Pohjimmiltaan digitaalinen kaksonen on tietokoneohjelma, joka ottaa todellista tietoa fyysisestä kohteesta syötteinä ja tuottaa lähtönä ennustuksia tai simulaatioita siitä, kuinka syötteet vaikuttavat kohteeseen. (Shaw & Fruhlinger, 2019.)

## 2.1 Käyttökohteet

Itselleni digitaalinen kaksonen tuli ensimmäistä kertaa vastaan IoT-opintojaksolla. Siellä mallinsimme sensoridataa pilveen, joka sitten ohjasi vastaanotettua dataa eteenpäin. Eteenpäin lähetettyä dataa voidaan ohjata erilaisia käyttökohteita varten. Niiden avulla voidaan esimerkiksi taloautomaatiossa kytkeä lämmitys tarvittaessa pois päältä tai sulkea ilmastointi, jos jotkin sensoreilta saadut raja-arvot ylittyvät. Tätä voidaan myös automatisoida, jolloin raja-arvojen ylittyessä järjestelmä tekee itse päätöksen datan perusteella. Tämä onkin digitaalisen kaksonen eräs käyttökohteista. Tuotantotekniikassa tuottajat etsivät jatkuvasti keinoja, joilla voidaan luoda säästöjä. Digitaalisen kaksonen avulla erilaisten koneiden tuottama reaaliaikainen tieto mahdollistaa häiriöihin puuttumisen nopeasti. Autoteollisuudessa digitaalisella kaksoella voidaan kuvata autojen moottoreita tai osio, joiden avulla luodaan simulaatioita ja data-analyysejä. Tekoäly lisättynä näihin simulaatioihin ja analyysihin parantaa testauksen ja komponenttien suorituskykyä. (Fuller, Fan, Day, Barlow 2020, 3–4.)

Terveystieteiden tutkimus hyötyy digitaalisen kaksonen kehityksestä. IoT-laitteet muuttuvat halvemmiksi ja niitä on entistä helpompaa sijoitella. Tulevaisuudessa onkin mahdollista, että ihmisestä luodaan digitaalinen kopio, joka antaa reaaliaikaista tietoa kehosta. Digitaalista kaksoista hyödynnetään tällä hetkellä esimerkiksi lääkkeiden vaikutuksen simuloinnin hahmottamiseen. (Fuller ym. 2020, 4.)

## 3 Azure

### 3.1 Azuren määritelmä

Azure on Microsoftin julkinen pilvipalvelualue. Sen tarkoituksena on auttaa yrityksiä kohtaamaan liiketoiminta tavoitteensa ja haasteensa. (Bigelow 2020.) Azure tarjoaa software as a service (SaaS), platform as a service (PaaS) ja infrastructure as a service (IaaS) ja tukee useita eri ohjelmointikieliä, työkaluja ja kehyksiä Microsoftilta ja myös kolmansien osapuolien ohjelmistoilta ja

järjestelmiltä. Azure julkistettiin vuonna 2008 nimen ”Project Red Dog” alla ja lopullisesti julkaistiin vuonna 2010 nimellä Windows Azure. 2014 se uudelleen nimettiin Microsoft Azureksi. (Wikipedia 2021a.)

### **3.2 IoT Hub**

IoT Hub on Microsoft Azuren tarjoama pilvipalvelu. Palvelu tarjoaa kommunikaatioväylän laitteelta hubiin ja toisinpäin. IoT Hubiin voidaan kiinnittää miljoonia laitteita eri puolilta maailmaa. Huomioon otettavaa on, että laitteiden määrän kasvaessa palvelun kustannukset lisääntyvät. Azuresta löytyy myös erilaisia viestin välitykseen erikoistuneita palveluita, kuten Azure Event Grid, Azure Event Hub ja Azure Service Bus. IoT Hub valikoitui kuitenkin opinnäytetyöhön, koska sen toiminta on suunniteltu eritoten IoT-laitteiden, kuten sensoreiden käyttöön, sen skaalautuvuuden ja viestin molempiin suuntiin mahdollistavan välityksen vuoksi. (Microsoft documentation 2021a.)

IoT Hub tukee paljon erilaisia ohjelmointikieliä: C, Embedded C, C#, Java, Python ja Node.js. Laitteiden yhdistämiseen voidaan käyttää seuraavia protokollia: HTTPS, AMQP, AMQP over WebSockets, MQTT ja MQTT over WebSockets. Jos kyseisiä protokollia ei voida käyttää, IoT Hub tarjoaa myös Azure IoT Edge ja Azure IoT protocol gateway -palvelut. (Microsoft documentation 2021a.)

### **3.3 Stream Analytics**

Stream Analytics on reaaliaikainen analytiikka- ja monimutkaisten tapahtumien käsittelymoottori, joka on suunniteltu analysoimaan ja käsittelemään suuria määriä nopeaa suoratoistodataa useista lähteistä samanaikaisesti. Mallit ja suhteet voidaan tunnistaa syöttölähteistä. Näitä lähteitä voi olla erilaiset laitteet, anturit ja sosiaalisen median syötteet ja sovellukset. Näiden tietojen perusteella voidaan tiedolle tehdä erilaisia toimenpiteitä, kuten hälytyksiä, tai sitä voidaan tallentaa tietokantaan. (Microsoft documentation 2021b.)



Stream Analytics tarvitsee toimiakseen syötteen, kyselyn ja kohteen minne lähettää tuloksen. Stream Analytics voi ottaa syötteenä esimerkiksi IoT Hubiin tulevia viestejä erilaisilta sensoreilta. Kyselyn avulla voidaan muokata tulevaa tietoa tarvittavaan muotoon. Kyselyt toimivat SQL-kielen avulla. Muokatun tiedon voi lähettää vaikkapa Cosmos DB -tietokantaan. (Microsoft documentation 2021b.)

Opinnäytetyöhön haluttiin palvelu, jonka avulla tietoa voitiin käsitellä ennen kuin se siirtyy tietokantaan. Stream Analytics kuulosti juuri oikealta Microsoftin dokumentaation perusteella. Sillä voitiin muokata sisään tulevaa tietoa ja tehdä erilaisia hälytyksiä, vaikkapa lämpötilan laskiessa, lähettää sähköpostiin viesti. Stream Analyticsin yksi suurimmista vahvuuksista oli se, että se toimii SQL-kielen avulla.

### **3.4 CosmosDB**

Azuren Cosmos DB on NoSQL tietokanta. Cosmos DB on nopea vastaamaan ja sen skaalautuvuus on välitön. Siinä on myös automaattinen hallinta, päivitykset ja paikkaukset. (Microsoft documentation 2021c.) Cosmos DB tallentaa tiedot ”items in containers” muotoon. Items eli kohteet saavat automaattisesti indexin, mutta sen voi muuttaa asetuksista. Automaattinen indeksointi parantaa suorituskykyä. (Wikipedia 2021b.)

Cosmos DB tarjoaa myös kuusi erilaista käyttöliittymää tietokannalleen. SQL API, MongoDB API, Gremlin API, Cassandra API, Azure Table Storage API ja etcd API. (Wikipedia 2021b.) SQL API on automaattisesti käytössä, jos ei halua tai muista vaihtaa valintaa.

Tietokannan rakentaminen oli yksi opinnäytetyön keskeisimmistä tavoitteista. Ensimmäisenä ratkaisuna mietittiin perinteistä SQL tietokantaa, jollaisen Azure tarjoaa. Microsoftin dokumentaation perusteella NoSQL kanta CosmosDB vaikutti kuitenkin juuri oikealta ratkaisulta nopean vasteajan takia. CosmosDB on myöskin suhteellisen tuore, joten se tarjoaisi haasteita tekemiseen.

CosmosDB otettiin käyttöön ja yhteysputki saatiin rakennettua. Sensoreilta tieto kulkeutui Stream Analyticsin välityksellä tietokantaan. Tieto ei kuitenkaan mennyt kantaan niin kuin oli tarkoitus, vaan jokainen uusi sensorilta tullut tapahtuma loi kantaan oman taulunsa. Tarkoituksena olisi ollut, että tietokannassa on yksi päätaulu, jonka alle tiedot kasaantuvat.

Ongelmaa selvittäessäni törmäsin Azuren toiseen palveluun, Time Series Insightsiin. Time Series Insights vaikutti tietokantaratkaisuna vielä paremmalta kuin CosmosDB. Päätimme yhteistuumin projektipäällikön kanssa poistaa CosmosDB:n käytöstä ja siirtyä Time Series Insightsiin.

### **3.5 Time Series Insights**

Time Series Insights on Azuren aikasarjatietokanta ja analytiikkapalvelu. Sitä voidaan verrata jollain asteella tietokantaan, koska tiedot säilyvät siellä seitsemän vuorokautta, jonka aikana niitä voidaan tarkastella. Tätä kutsutaan nimellä Warm Storage. Seitsemän vuorokauden jälkeen tiedot siirtyvät cold storageen, jonka voi tehdä Time Series Insightsin käyttönoton yhteydessä. Data, joka kulkee sensoreilta IoT Hubiin, siirtyy Time Seriesin muutaman yhteyskanavan avulla. Time Series Insights tarjoaa palvelussaan helpon hierarkiarakenteen, jota voidaan muokata ja jokaisen eri tason alle voidaan sijoitella sensoreita. Hierarkioita voi olla useita, joten esimerkiksi usean eri kohteen sisällyttäminen tietokantaan on helppoa.

CosmosDB:n jälkeen siirryttiin käyttämään Time Series Insightsia. Time Series Insights on suunniteltu täysin IoT ratkaisuiden luomiseen rakenteensa vuoksi. Time Series Insightsin määrittämisen yhteydessä annetaan Time Series Id. Time Series Id toimii samalla tavalla kuin primary tai foreign key SQL-tietokannassa. Sen avulla voidaan tehdä tietokannan rakenteesta halutunlainen.

### **3.6 Azure Functions**

Azure Functions on serveritön ratkaisu, jossa pilvi-infrastruktuuri huolehtii sovelluksen ylläpidon tarpeista. Azure Functions tarjoaa mahdollisuuden

kirjoittaa sovelluksen logiikka valmiille koodilohkoille. Lohkot voidaan suorittaa milloin tahansa ja niihin varataan Azuren puolelta niin paljon resursseja, kuin on tarpeellista. Kun lohkojen suoritus loppuu, resurssien käyttökin loppuu, jotta ylimääräisiä kuluja ei syntyisi. (Microsoft documentation 2021d.)

Azure Functions tarjoaa opinnäytetyössä yhteysväylän IoT Hubilta digitaaliseen kaksoseen. Microsoftin dokumentaatio ei tarjonnut mitään muuta Azuren palvelua tähän tarkoitukseen, joten Azure Functionsin valinta oli itsestään selvä. Funktion kirjoittamiseen käytettiin Visual Studiota paikallisella koneella. (Microsoft documentation 2021e.)

### **3.7 Azure Digital Twins**

Digitaalinen kaksonen on Microsoft Azuren tarjoama alusta palveluna -ratkaisu. Sen avulla voidaan luoda kaksosgraafi digitaalisesti vastaamaan oikean elämän vastinetta. Digitaalisen kaksosen graafi luodaan JSON-kieleen perustuvalla Digital Twins Definition kielellä. Sen avulla voidaan kuvata kaksosten ominaisuuksia, telemetriatapahtumia, komentoja, komponentteja ja suhteita. Azure tarjoaa digitaalisen kaksosen hallintaan Azure Digital Twins Explorer - ominaisuuden, jonka avulla voidaan visualisoida ja tehdä kyselyitä digitaalisen kaksosen graafille. (Microsoft documentation 2021f.)

## **4 Työkalut**

### **4.1 Visual Studio**

Visual studio on Microsoftin kehittämä ohjelmointiympäristö käyttöjärjestelmien, konsolien, verkko- ja mobiilisovellusten sekä pilvi- ja verkkopalveluiden kehittämiseen. Alusta ei ole kieli riippuvainen, joten sillä on mahdollista kirjoittaa C#-, C++-, VB- Python- ja muita ohjelmointikieliä. Visual Studio on saatavilla Windowsille ja macOS:lle. Visual Studiosta on kolme erilaista versiota, Community, Professional ja Enterprise. (Aggerwal 2019.) Community on Visual Studion ilmainen versio, jonka kaikki voivat ladata. Ohjelmaa voi käyttää ilman

tunnuksia 30 päivää, jonka jälkeen viimeistään täytyy rekisteröityä tai kirjautua Microsoftin tunnuksilla sisään. Sisään kirjautuminen edesauttaa sovelluskehitystä esimerkiksi Azureen, jos käytössä on sama Microsoft-tunnus, jolloin tunnistautuminen tapahtuu automaattisesti.

## **4.2 Azure IoT Explorer**

Azuren tarjoama IoT Explorer on työkalu, jolla voidaan tarkastella IoT Hubissa sijaitsevia virtuaalisia laitteita ja niiden vastaanottamaa dataa. Explorer asennetaan paikalliseen koneeseen, jolloin sillä voidaan ottaa yhteyttä IoT Hubiin. Yhteydenottaminen vaatii kyseisen Hubin yhteysmerkkijonon. Kun yhteys on luotu, laitteet, jotka sijaitsevat IoT Hubissa näkyvät listana Explorerissa. (Microsoft documentation 2021g.) Työssäni varsin käytännölliseksi Explorer osoittautui viestien sisällön ja lähetettävän laitteen ominaisuuksien tarkastelussa.

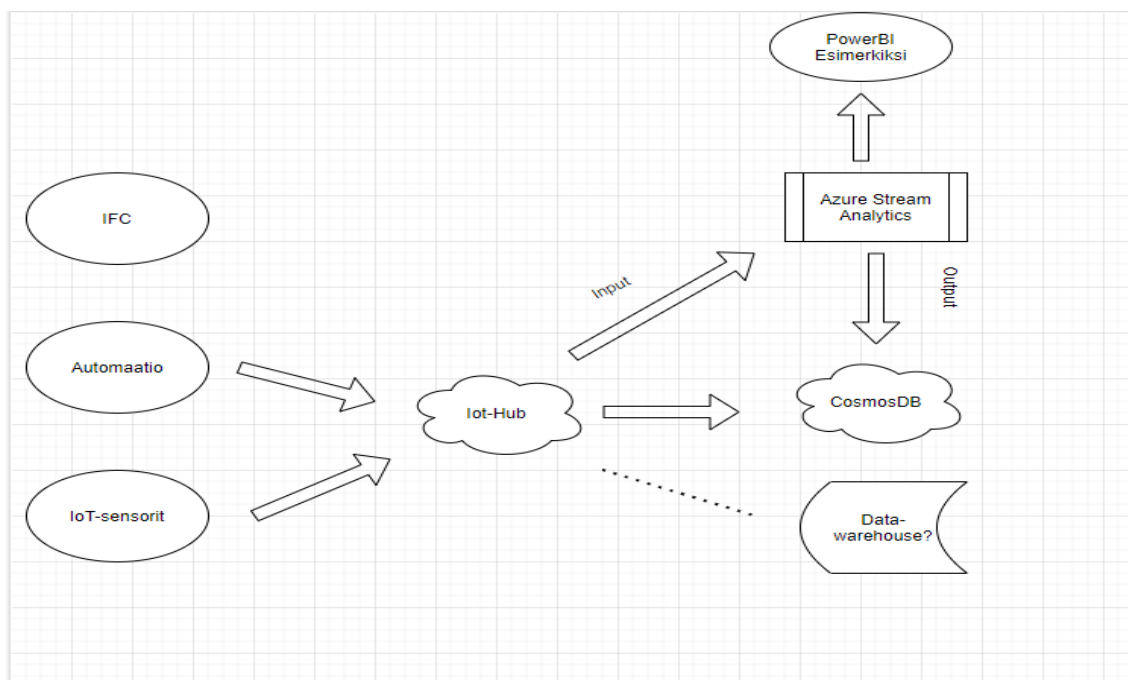
# **5 Opinnäytetyön suunnittelu ja toteutus**

## **5.1 Projektin yleiskatsaus**

Digitaalisen kaksosen tietöalustan luominen on osa isompaa hanketta. Hankkeen nimi on kokonaisuudessaan ”Digital Twin - Kiinteistö- ja talotekniikan kestävät palvelut ja osaaminen”. Hankkeen rahoittajana toimii Etelä-Savon elinkeino-, liikenne- ja ympäristökeskus, Kestävää kasvua ja työtä 2014–2020 Suomen rakennerahasto-ohjelma. Tavoitteena hankkeella on kehittää Karelia-ammattikorkeakoulun rakennustekniikan ja talotekniikan mallintamisosaaamista ja talotekniikan kiinteistöautomaation koulutusta yhteistyössä tietojenkäsittelyn koulutuksen, kiinteistöautomaation erikoistuneiden yritysten ja kiinteistöjen ”edelläkävijäylläpitäjien” kanssa. Hankkeen kumppaneina toimivat taloautomaatiota kehittävät yritykset sekä maakunnan yritykset ja yhteisöt, joilla on merkittävää kiinteistömassaa. Kohderyhmänä ovat myöskin alueen ohjelmistoalan yritykset. Pää toteuttajana hankkeella on Karelia Ammattikorkeakoulu oy. (Laakkonen 2021.)

## 5.2 Opinnäytetyön alku

Opinnäytetyön tekeminen alkoi keväällä, jolloin tavoitteena oli selvittää Azuren erilaisten tekniikoiden soveltuvuus projektin tavoitteisiin. Alusta asti oli selvää, että opinnäytetyössä käytettäisiin Microsoftin Azuresta löytyviä palveluita. Viikoittaisissa yhteisissä Teams -palavereissa tarkasteltiin edistymistä ja mahdollisia ongelmia. Ensimmäinen tavoite oli tehdä kuvaus opinnäytetyöhön tarvittavista palveluista ja datan liikkumisesta niiden välillä.



Kuva 1. Alkuperäinen suunnitelma (Niemi).

## 5.3 IoT Hub

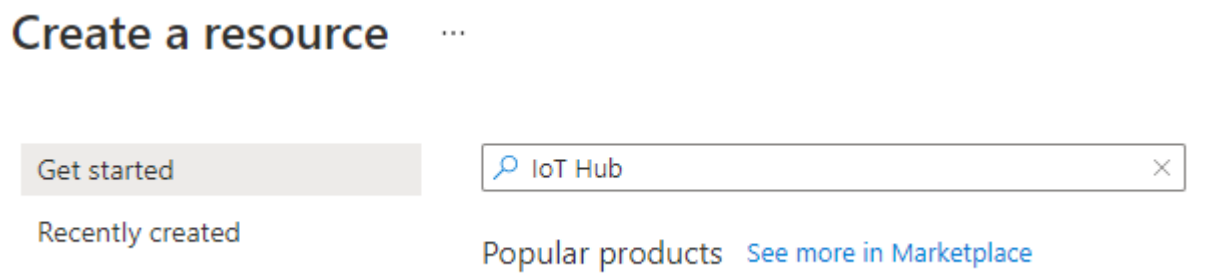
IoT Hub oli ensimmäinen komponentti, joka tuli käyttöön Azuren portaaliin. IoT Hubin tekeminen oli tärkein opinnäytetyön kannalta, jotta muut projektiin osallisena olevat henkilöt, saivat testattua tekemiensä asioiden toimivuuden, kuten toimiiko datan lähettäminen Raspberry Pi:ltä eteenpäin. Oikeuteni eivät vielä riittäneet komponenttien aktivointiin, joten jouduin olemaan yhteydessä projektipäällikköön, joka loi IoT Hubin puolestani ja antoi tarvittavat oikeudet minulle tulevaisuuden kannalta.

### 5.3.1 IoT Hub käyttöönotto

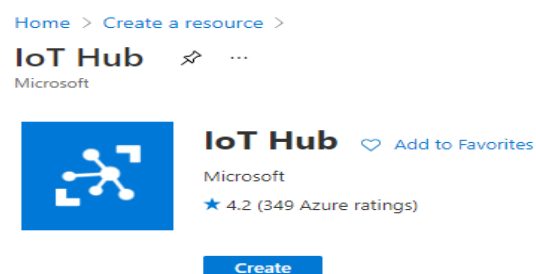
IoT Hubin käyttöönotto tapahtui Azuren portaalin kautta, osoitteesta:

<https://portal.azure.com>. Azuren portaaliin kirjaututtiin koulun tunnuksilla.

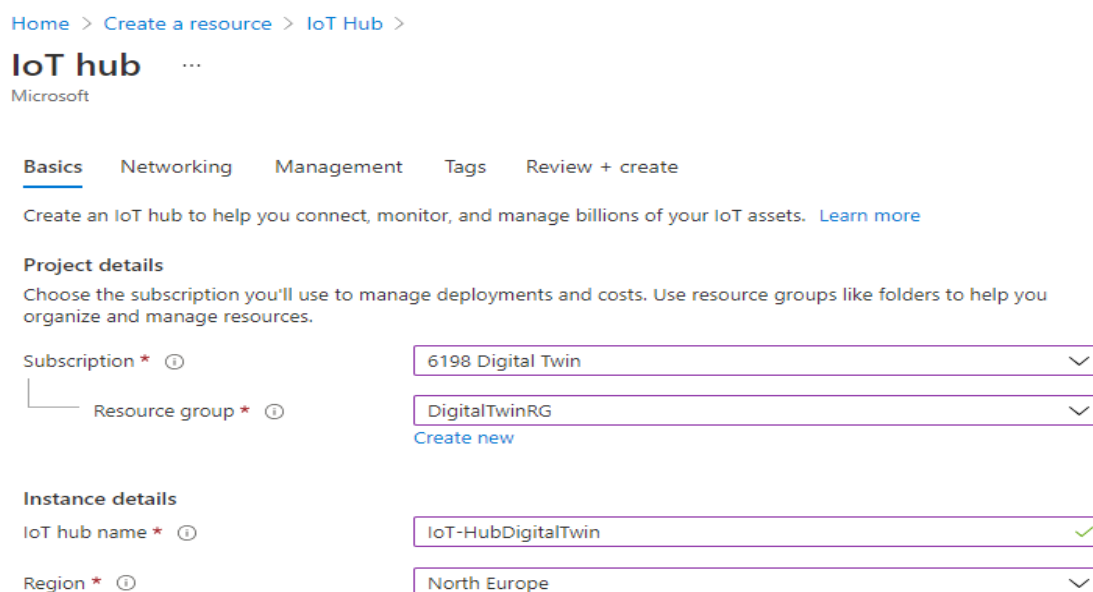
Kirjautumisen onnistuttua, etusivulla olevan hakupalkin avulla etsittiin IoT Hub ja Create a Resource -painikkeen avulla päästiin luomaan sitä.



Kuva 1. Hakupalkki resurssien etsimiseen (Niemi).



Kuva 2. Create -painike luo resurssin (Niemi).



Kuva 3. IoT Hub perusasetukset (Niemi).

Ensimmäisenä avautui perusasetukset, jotka täytyi säätää vastaamaan Azuren tilausta. Opinnäytetyössä tilaus oli projektin Digital Twin -alla. Resource Group

valikosta voi luoda tarvittaessa uuden resurssi ryhmän, minkä alle kustannukset tulevat. Projektia varten oli luotu oma, DigitalTwinRG – resurssi ryhmä, joka mahdollisti palveluiden löytymisen yhdestä paikasta.

Basics **Networking** Management Tags Review + create

You can connect to your IoT hub either publicly via its public hostname or privately using a private endpoint. [Learn more](#)

Connectivity configuration \*

Public access

Private access (Recommended)

**i** You can change this or configure another connectivey method after this resource has been created. [Learn more](#)

Kuva 4. IoT Hub Networking (Niemi).

Networking -välilehden alta valittiin oikea yhteystapa. Opinnäytetyössä käytettiin Public Access -tapaa. Learn More avaa ohjeistuksen, jos valinnan kanssa on ongelmia. Networking asetuksia voitiin muuttaa tarvittaessa myöhemmin.

**IoT hub** ...  
Microsoft

Basics **Networking** Management Tags Review + create

Each IoT hub is provisioned with a certain number of units in a specific tier. The tier and number of units determine the maximum daily quota of messages that you can send. [Learn more](#)

Scale tier and units

Pricing and scale tier \* ⓘ  [Learn how to choose the right IoT hub tier for your solution](#)

Number of S1 IoT hub units ⓘ  Determines how your IoT hub can scale. You can change this later if your needs increase.

Defender for IoT  On  
Turn on Defender for IoT and add an extra layer of threat protection to IoT Hub, IoT Edge, and your devices. [Learn more](#)

Pricing and scale tier ⓘ	S1	Device-to-cloud-messages ⓘ	Enabled
Messages per day ⓘ	400 000	Message routing ⓘ	Enabled
Cost per month	21.51 EUR	Cloud-to-device commands ⓘ	Enabled
Defender for IoT ⓘ	0.00086122435 IoT Edge ⓘ	Device management ⓘ	Enabled

Role-based access control  
Change the permission model to Azure role-based access control (RBAC) only, or to a combination of shared access policies and RBAC. [Learn more](#)

RBAC only

Shared access policy + RBAC

To manage the elements within an instance, a user needs access to IoT Hub data APIs. Select the suggested role below to grant yourself full access to the APIs. You can also use Access Control (IAM) to chose appropriate roles later. [Learn more](#)

Assign me to the IoT Hub Data Contributor role ⓘ

Advanced settings

Scale

Device-to-cloud partitions ⓘ

Kuva 5. IoT Hub Management (Niemi).

Management osassa määriteltiin IoT Hubin kustannukset ja se kuinka paljon laitteita haluttiin mukaan toteutukseen. Perus S1 eli Standard Tier 1, tarjosi 1.5GB/päivä dataa ja 400000 viestiä päivässä. Suuremmissa ratkaisuisissa on suositeltavaa valita S2 tai S3 Taso. IoT Hub units on myöskin suositeltavaa nostaa suuremmaksi kuin 1, jos ratkaisu on suuri. Kustannukset kuitenkin nousevat suhteessa valintoihin ja sen vuoksi opinnäytetyössä oli käytössä perustaso. Microsoftin olisi syytä ajatella myöskin pienempiä ratkaisuja, joille riittää esimerkiksi muutaman sensorin liittäminen IoT Hubiin. Hinta ei sinällään ole korkea, mutta muiden palveluiden liittäminen Azuren Portaaliin kasvattaa menoja merkittävästi. Valinnoissa oli myös tarjolla suojauksen käyttöönotto, mutta opinnäytetyöhön sitä ei aktivoitu. Role-based access control -valinnoissa on mahdollista määritellä resurssin turvallisuutta ja kun hubi on luotu, käyttäjille voidaan antaa oikeuksia esimerkiksi IoT Hubin tarkasteluun. Lopuksi yhteenvetoikkunassa löytyi valitut asetukset ja kustannus. Jos jotain puuttuu, on vielä mahdollista palata taaksepäin muuttamaan asetuksia. Hubin luominen käynnistyi Create -näppäintä painettaessa ja siinä kesti muutamia minutteja. Kun IoT Hub oli valmis, se löytyi joko suoraan etusivulta tai hakupalkin avulla.

### 5.3.2 IoT Hub ja oikeudet

IoT Hubin muokkauksiin ja yhdistämissiin tarvitaan soveltuvat oikeudet. Oikeudet voi pyytää henkilöltä, joka hallinnoi kyseistä Azuren palvelua. Vähimmäis-oikeudeksi riittää Contributor -rooli. IoT Hubin Access controllista päästään muokkaamaan näitä oikeuksia

The screenshot displays the Azure IoT Hub Access control (IAM) interface. The main page shows navigation options like 'Check access', 'Role assignments', 'Roles', 'Deny assignments', and 'Classic administrators'. A 'My access' section is visible on the left. The 'Add role assignment' dialog box is open on the right, showing the following details:

- Role:** Contributor
- Assign access to:** User, group, or service principal
- Select:** ville.niemi
- Member:** Niemi Ville A. (1804213@edu.karelia.fi)

At the bottom of the dialog, there are 'Save' and 'Discard' buttons. The background interface includes a search bar for members and a 'Selected members' section indicating no members are currently selected.



Kuva 6. IoT Hub Access control (Niemi).

Avautuvan valikon vasemmassa yläreunassa on +Add painike. Add role assignment -valikosta valitaan Contributor, etsitään haluttu henkilö, painetaan nimeä ja klikataan Save. Näin mahdollistetaan tarvittavien oikeuksien jakaminen niitä tarvitseville.

### 5.3.3 Laitteiden lisääminen IoT Hubiin

IoT Hubiin lisättävällä laitteella ei ole mitään merkitystä millainen se on. Laitteen ei tarvitse olla sensori, jotta sille voi tehdä virtuaalisen vastakappaleen hubiin. IoT Hubin etusivulla vasemmassa reunassa on Explorers -otsikon alla IoT devices. Napin alta avautuu laitteiden luonti ja valmiiden laitteiden lista. Add Device -painikkeesta voidaan lisätä uusia laitteita.

View, create, delete, and update devices in your IoT Hub.

Device name

**Find devices**

+ Add Device   Refresh   Delete

Device ID	Status
Device2	Enabled
Classroom_214	Enabled

Kuva 7. IoT Hub Find devices ja Add device (Niemi).

Device ID \* ⓘ

Authentication type ⓘ

**Symmetric key**   X.509 Self-Signed   X.509 CA Signed

Auto-generate keys ⓘ

Connect this device to an IoT hub ⓘ

**Enable**   Disable

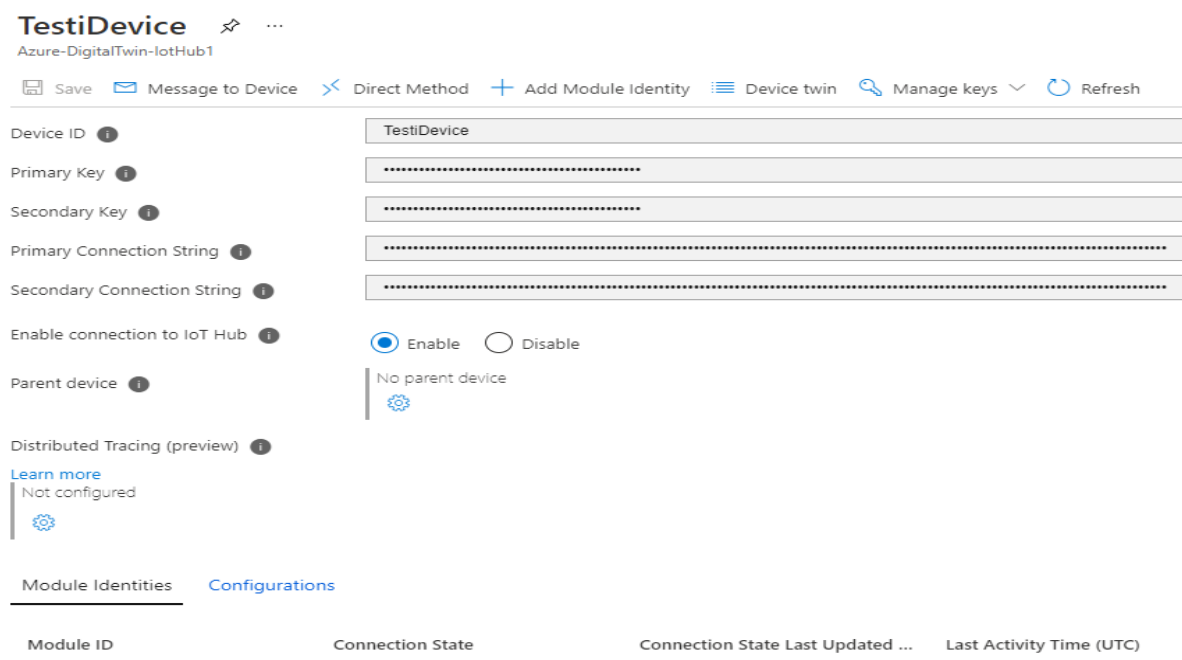
Parent device ⓘ

**No parent device**  
[Set a parent device](#)

Kuva 8. IoT Hub uuden laitteen luominen (Niemi).

Laitteelle valitaan tunniste ja Auto-generate keys -kohdassa oleva rasti takaa sen, että laitteelle generoituu automaattisesti sille yksilöllinen yhteysmerkkijono,

jonka avulla siihen voidaan yhdistää fyysinen laite rajapinnan avulla. Kun virtuaalinen laite on luotiin, se ilmestyi aiempaan valikkoon, jossa sijaitsee kaikki luodut laitteet. Hakutoiminto helpottaa, jos luotuja laitteita on paljon.

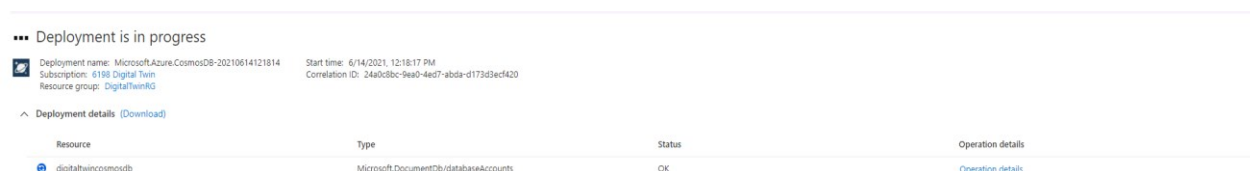


Kuva 9. IoT Hub Laitteen asetuksia ja merkkijonot (Niemi).

Virtuaalinen laite sisältää erilaisia asetuksia ja avaimia, joita voidaan hyödyntää erilaisissa ratkaisuissa. Opinnäytetyössä ei ole tarvetta muulle kuin Primary Connection String -merkkijonolle.

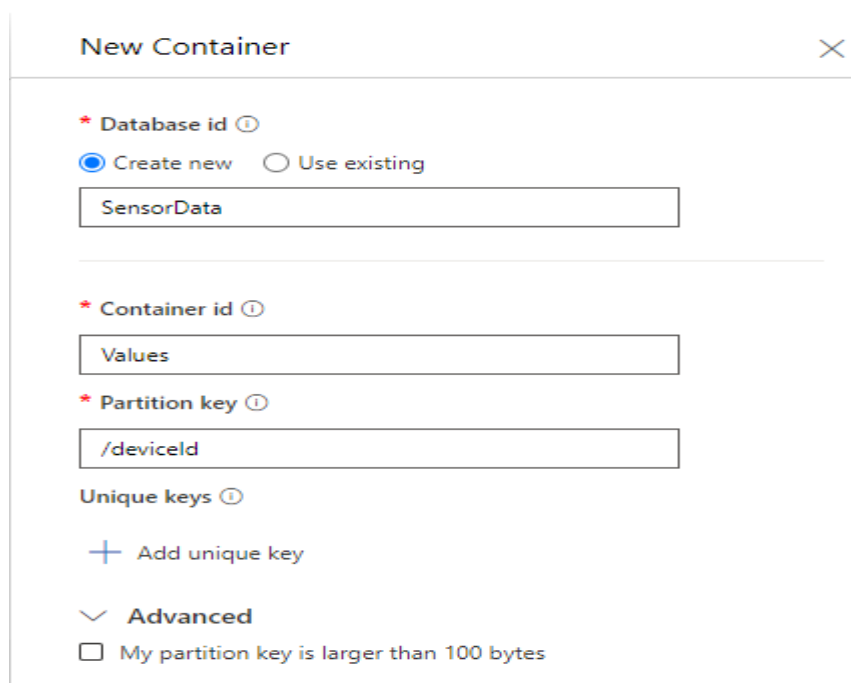
## 5.4 Ensimmäinen tietokantatesti

IoT Hubin luomisen jälkeen alkoi tietokannan testaaminen. Tietokannaksi valikoitui alussa CosmosDB ja tiedon tietokantaan kuljettavaksi palveluksi Stream Analytics. Sovimme projektipäällikön kanssa CosmosDB -tietokannan käyttöönotosta, koska haluttiin tutustua uuteen teknologiaan ja sen kuukausittainen kulutus ei ollut suuri, noin 20e/kk.



Kuva 10. Tietokannan luominen vaiheessa (Niemi).

CosmosDB vaati erilaisia avainkenttiä toimiakseen oikein. Database id, tarkoittaa tietokannan nimeä, container id, niin sanotun taulun tunnusta ja Partition key, tunnusta millä tieto jakaantuu taulun sisällä. Tärkein tunniste oli Partition Key. Luonnolliseksi vaihtoehdoksi valikoitui lähetettävän laitteentunnus, koska jokaisella laitteella se on yksilöllinen.



New Container

\* Database id ⓘ  
 Create new  Use existing  
SensorData

\* Container id ⓘ  
Values

\* Partition key ⓘ  
/deviceid

Unique keys ⓘ  
+ Add unique key

Advanced  
 My partition key is larger than 100 bytes

Kuva 11. Tietokannan luominen (Niemi).

### 5.4.1 Striimauspalvelu

Seuraavaksi täytyi aktivoida Stream Analytics -palvelu. Sen avulla voitiin lähettää dataa sensoreilta tietokantaan. Kun Stream Analytics aktivoitiin sen toiminta vaatii input- ja output -asetusten määrittämisen. Input -asetukseen lisättiin IoT Hub, josta data tulee ja Output -asetukseksi CosmosDB, jonne data menee.

## IoT Hub ✕

New input

**Input alias \***

 ✓

Provide IoT Hub settings manually

Select IoT Hub from your subscriptions

**Subscription**

 ▾

**IoT Hub \* ⓘ**

 ▾

**Consumer group \* ⓘ**

 ▾

**Shared access policy name \* ⓘ**

 ▾

**Shared access policy key ⓘ**

**Endpoint ⓘ**

 ▾

**Partition key ⓘ**

**Event serialization format \* ⓘ**

 ▾

**Encoding ⓘ**

 ▾

**Event compression type ⓘ**

 ▾

Kuva 12. IoT Hub input (Niemi).

Kuva 13. CosmosDB output (Niemi).

Tiedon ohjaamiseksi Stream Analytics vaati SQL-kyselyn. Opinnäytetyössä käytettiin Raspberry Pi Web simulaattoria, jonka avulla simuloitiin sensoridataa. Työssä valittiin simulaattorilta kaikki tuleva tieto ja lähetettiin se tietokantaan.

```

1 SELECT
2   *
3 INTO
4   [cosmosdboutput]
5 FROM
6   [iothubinput]
7 WHERE
8   deviceID = 'Raspberry Pi Web Client'

```

Showing 3 rows from 'cosmosdboutput':

messageid	deviceid	temperature	humidity
3	"Raspberry Pi Web Client"	31.977447715160253	65.68331278561304
2	"Raspberry Pi Web Client"	26.73162659058186	72.53873697233013
1	"Raspberry Pi Web Client"	27.090746104234626	76.925592496537

Kuva 14. Stream Analytics testi (Niemi).

9fb818a8-1bed-4058-a60b-c0fa325dfcfa	Raspberry Pi Web Client
7191afdd-c79c-40ee-b16b-b478a036677b	Raspberry Pi Web Client
c79ec7c4-0413-4860-8376-1335965dfad4	Raspberry Pi Web Client
333987fb-13ce-4d0c-afbd-43448ae52a74	Raspberry Pi Web Client
da6a47af-26f1-4fb8-ab03-790490c6849d	Raspberry Pi Web Client
4daebaa0-de23-4826-aff1-857531aa1d7f	Raspberry Pi Web Client
0452f86d-8354-4ef9-acff-09229ebf2a19	Raspberry Pi Web Client
24ba0ad3-e814-4b4b-8f95-8c49820aebf7	Raspberry Pi Web Client

```

4
5
6
7
8
9
10
11
12
13
"messageId": 54,
"deviceId": "Raspberry Pi Web Client",
"temperature": 28.67,
"humidity": 73.53,
"EventProcessedUtcTime": "2021-06-14T09:56:52.0603201Z",
"id": "7a634d6a-4fbc-4585-a1c9-8d8840e7ecc4",
"_rid": "4gplAK+5r3gtAAAAAAAAAA=",
"_sel": "dbs/4gplAK+5r3gtAAAAAAAAAA=/docs/4gplAK+5r3gtAAAAAAAAAA=/",
"_etag": "\"0000062d7-0000-0c00-0000-60c727e40000\"",
"_attachments": "attachments/",
"_ts": 1623664612

```

Kuva 15. Tieto tietokannassa (Niemi).

Tieto siirtyi tietokantaan, mutta tässä vaiheessa ongelmaksi muodostui se, että laitteella voi olla useita sensoreita, joten ne piti saada näkyviin jollain tavalla. Valitettavasti nyt jokainen sensorilta tuleva tieto luo oman osionsa tietokantaan ja tiedon hakeminen erilaisten kyselyjen avulla on hidasta. Ratkaisua yritettiin löytää Stream Analyticsin kyselyjen muokkaamisen avulla. Tietoa ei kuitenkaan onnistuttu viemään tietokantaan oikeanlaisessa rakenteessa.

## 5.5 Aikasarjatietokanta

CosmosDB tietokannan ongelmien selvityksen yhteydessä keskustelimme vaihtoehtoisesta toteutuksesta Teams -palaverissa projektipäällikön kanssa. Microsoftin Azure tarjosi täysin IoT aikadatalle suunnatun tietokantaratkaisun. Päätimme hylätä CosmosDB tietokannan ja alkaa selvittämään kuinka vaihtoehtoinen ratkaisu toimisi. Time Series Insights eli aikasarjatietokanta tarjoaisi ratkaisun yhden laitteen usean sensorin määrittelyyn ja niiden tietojen näyttämiseen. Tiedon hakeminen ja näyttäminen tapahtuu helposti visuaalisten työkalujen ja graafien avulla

Aikasarjatietokannan käyttöönotto tapahtui Azuren portaalista samalla tavalla kuin IoT Hubin. Haettiin palvelu hakukentän avulla ja palvelu luotiin Create -painikkeella.

[Home](#) > [Create a resource](#) >

# Time Series Insights

Microsoft



## Time Series Insights [Add to Favorites](#)

Microsoft

★ 3.4 (35 Azure ratings)


Create


Kuva 16. Time Series Insights Create (Niemi).

Ensimmäiseksi piti määrittellä tilaus- ja resurssiryhmä, jonka alle Time Series Insights sijoitettiin. Opinnäytetyössä käytettiin digitaalisen kaksosen tilausta. Instanssin yksityiskohdat valittiin vastaamaan maantieteellistä sijaintia. Tier eli taso valittiin GEN2, joka oli aikasarjatietokannan uusin versio.


### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* 

6198 Digital Twin 




Resource group \* 

DigitalTwinRG 

[Create new](#)


### Instance details


Environment name \*

TimeSeriesInsights 

Location \*

North Europe 

Tier \* 

Gen2 (L1) 

Kuva 17. Aikasarjatietokannan asetuksia (Niemi).

Aikasarjatietokantaa määrittäessä täytyi olla tarkkana Time Series Id:n kanssa. Id:n valinnassa piti miettiä tietokannan rakennetta ja tuottaisiko se vaikeuksia esimerkiksi sensoreiden koodin suunnittelijoille. Opinnäytetyössä jouduttiin tekemään kanta uudelleen kaksi kertaa, koska Id:t eivät olleet soveltuvia tietojen luettavuuden kannalta. Opinnäytetyössä oli tärkeä ottaa huomioon jokaisen sensorin tarkastelun mahdollisuus. Työssä oli käytössä DeviceID,

jonka mukaan eri laitteet määrittyivät. Muita yksilöiviä tunnuksia oli Telemetry.SensorID, joka mahdollisti eri sensoreiden tarkastelun yksittäisinä.

Tietovarastointia ei ollut tarkoitus tuoda opinnäytetyöhön mukaan, mutta aikasarjatietokannan myötä se täytyi tehdä. Time Series Insights säilyttää maksimissaan seitsemän päivää tietoja, jonka jälkeen ne siirtyvät kylmävarastointiin. Tämä mahdollistaa sen, että tietoja voidaan tarkastella Time Series Insights Explorerissa pitkältikin aikaväliltä.

#### Time series ID

**i** Time Series ID acts as a partition key for your data and as a primary key for your time series model. It is important that you specify the appropriate Time Series ID during environment creation, since you can't change it later.

\* Property name ⓘ

For example, deviceId, objectId or a tag name

#### Cold store

Creates a new Azure Storage resource in the subscription and region you've chosen for the TSI environment. You will incur data storage and transaction charges for the data that Time Series Insights reads and writes to this storage resource. [Learn more](#) ↗

Storage account name \* ⓘ

Create a new storage account

Storage account kind \* ⓘ

StorageV2 (general purpose V2) ▼

Storage account replication \* ⓘ

Locally redundant storage (LRS) ▼

#### Data Lake Storage Gen2

Hierarchical namespace \* ⓘ

Enabled  Disabled

#### Warm store

Creates a warm store for the TSI environment optimized for higher query performance and unlimited queries. The warm store can be removed from the environment at a later time. By selecting "Yes", you will incur data storage costs.

Enable warm store \* ⓘ

Yes  No

Data retention time (in days) ⓘ

7

Kuva 18. Time Series Id, tiedon kylmä -ja lämmin varasto (Niemi).

Time Series Insights täytyy putkittaa ottamaan vastaan tietoa IoT Hubilta, joten asennuksen yhteydessä on tärkeää määrittää oikea IoT Hub tiedon lähteeksi. Tietolähteen yhteydessä täytyi asettaa IoT Hub access policy name iothubowneriksi ja luoda uusi consumer group, ellei sitä ollut tehty aikaisemmin.



Asennuksen viimeistelyksi tarkastettiin asetusten yhteenveto, jotta välttyttäisiin kannan uudelleen asennukselta.

\* Basics \* **Event Source** \* Networking Tags Summary

An event source is the IoT Hub or Event Hub that feeds data into your Time Series Insights environment. [Learn more](#) ↗

Create an event source? \*  Yes  No

Source type \*  IoT Hub  Event Hub

Name \*

Subscription \*

IoT Hub name \*

IoT Hub access policy name \*

IoT Hub consumer group \*

**Start options**

Choose the data you want us to collect. We can begin now, or we can also include older, preexisting data. Choosing a custom enqueued time (UTC) means we'll go back to when the event was added to your IoT hub or Event hub and collect data from that point forward. Then enqueued time is different from the timestamp property within the body of your event. [Learn more](#) ↗

Start time

**Timestamp**

Create an event source timestamp property name. If you don't enter a value, we'll use the message enqueued time from the event source. [Learn more](#) ↗

Property name

Kuva 19. Tietolähde IoT Hub (Niemi).

### Event Hub Details

Partitions

Event Hub-compatible name

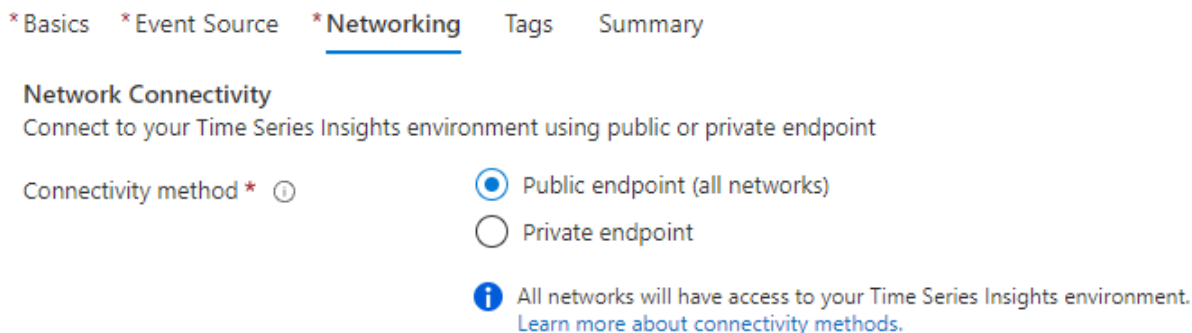
Retain for  Days

Consumer Groups

\$Default

tsiconsumergroup

Kuva 20. Consumer group IoT Hubista (Niemi).



Kuva 21. Verkkoyhteydet asetetaan julkisiksi (Niemi).

## 5.6 Tiedon simulointi

Opinnäytetyöhön piti ottaa huomioon erilaiset sensorit, jotka tuottavat tietoa esimerkiksi lämpötilasta ja paine-eroista. Oikean datan saaminen opinnäytetyöhön ei ollut kuitenkaan mahdollista vielä, joten käytössä oli Raspberry -simulaattori, jonka avulla saatiin simuloitua dataa opinnäytetyön käyttöön.

Kuten oikeankin Raspberryn kanssa, simulaattoriin jouduttiin tekemään rajapinta, jotta tieto kulkisi laitteelta IoT Hubiin. Simulaattorin yhdistämiseksi IoT Hubiin tarvittiin yhteysmerkkijono. Simulaattorista ei tarvinnut muuttaa kuin viestin rakennetta vastaamaan tietokantaa. Opinnäytetyössä oli myös muita osallistujia. Toinen heistä simuloi analogisia- ja binäärisiä -kanava-arvoja ja toinen lähetti fyysisellä Raspberryllä lämpötilatietoja.

```
const connectionString = 'HostName=Azure-DigitalTwin-IotHub1.azure-de
const LEDPin = 4;
```

Kuva 22. Yhteysmerkkijono IoT Hubista (Niemi).

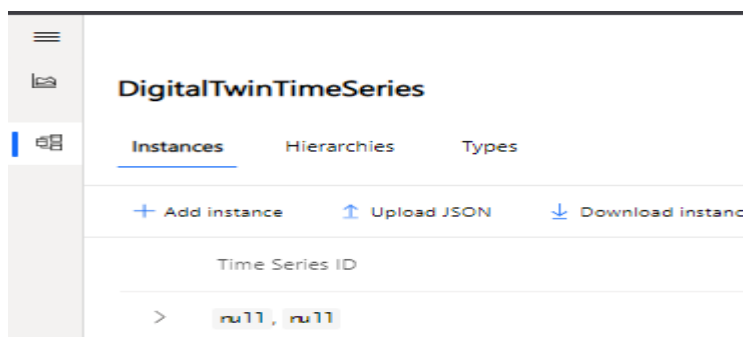
```
function getMessage(cb) {
  messageId++;
  sensor.readSensorData()
    .then(function (data) {
      cb(JSON.stringify({
        temperature: data.temperature_C,
      }), data.temperature_C > 30);
    })
    .catch(function (err) {
      console.error('Failed to read out sensor data: ' + err);
    });
}
```

Kuva 23. Viestin rakenne (Niemi).

## 5.7 Tiedon lähettäminen aikasarjatietokantaan

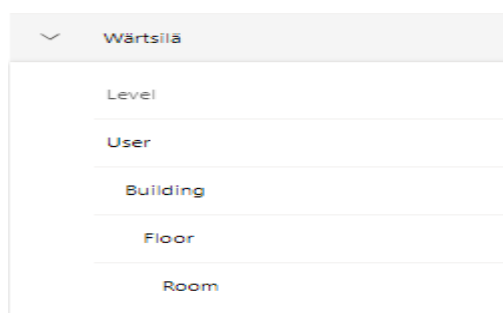
Aikasarjatietokannan luomisen yhteydessä määriteltiin Time Series Id. Sen avulla voitiin luoda oikeanlainen tietokantarakenne. Kokeilujen yhteydessä todettiin, että laitteella voi olla useita sensoreita ja jokaisen sensorin tuottama tieto olisi hyvä olla eriteltyinä. Ongelmaksi nousi varsinkin kanava-arvot, jossa yhdellä laitteella saattoi olla satoja eri mittapisteitä. Simulaattorin lähettämän datan ja siitä tehtyjen päätelmien takia jouduttiin aikasarjatietokanta poistamaan ja luomaan uudelleen, jotta rakenne olisi oikea.

Time Series Insights vaatii toimiakseen tiettyjen asetusten asettamista tiedolle. Kun dataa reititetään aikasarjatietokantaan, laitteet, joilta data on tullut löytyvät Instances välilehdeltä, joka sijaitsee Model -osion alla.



Kuva 24. Instanssit (Niemi).

Ennen kuin instansseja lähdettiin määrittämään, luotiin valmiiksi hierarkiat ja tyypit, jotta ne voitiin lisätä instansseille. Hierarkiat määrittävät datan rakenteen aikasarjatietokannan sisällä, joten sen avulla oli mahdollista luoda esimerkiksi rakennuksesta hierarkia, joka ottaa huomioon kerrokset ja huoneet.



Kuva 25. Hierarkia rakennukselle (Niemi).

Tulevan tiedon tietotyypit määriteltiin Types -välilehdellä. Tietotyypit kannattaa määrittellä, jotta tiedosta saadaan tehtyä visuaalisia graafeja. Opinnäytetyössä

sensoreilta tuleva data oli kaikki numeraalista tietoa, joten niille oli valmiina numeric -määrittys. Tiedoille voitiin tehdä myös erilaisia laskutoimituksia valmiiden operaatioiden avulla. Tätä ominaisuutta ei kuitenkaan hyödynnetty työssä. Operaatioiden avulla saatiin myös muutettua merkkityyppinen tieto numeraaliseksi.

Name	Kind	Value
Humidity	numeric	Sevent.Telemetry:humidity.Double
Temperature	numeric	Sevent.Telemetry:temperature.Double
Laitteentunnus	numeric	Sevent.DeviceID.Long
Min Temperature	numeric	Sevent.Telemetry:temperature.Double

Kuva 26. Numeraalisia tietotyyppiä (Niemi).

### Edit Humidity

**Name**

**Kind**

**Value**

Select from preset  Custom

Advanced options

**Aggregation operation**

Select from preset  Custom

**Interpolation**

*Interpolation is not supported for selected aggregation operation*

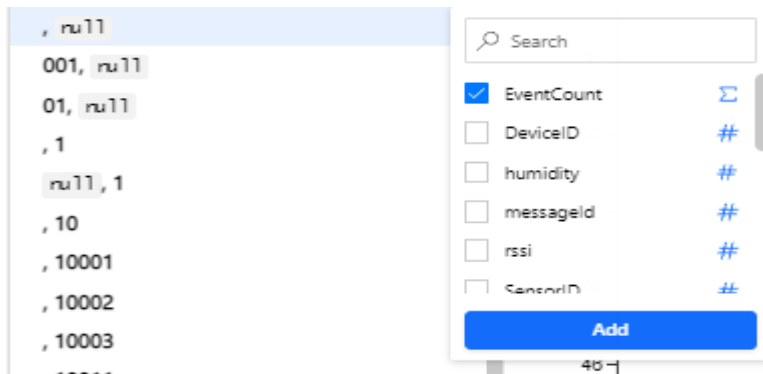
**Filter (optional)**

[Learn more](#)

Kuva 27. Tietotyyppien asettaminen (Niemi).

Jos tietotyyppiä ei ollut määritelty, Time Series Insights tarjoaa default types -asetuksen, joka on automaattisesti päällä jokaisella instanssilla. Default type sisältää Event count eli tapahtumien lukumäärä -tyypin. Default type -ominaisuuden avulla tietoja on mahdollista tarkastella, vaikka määritelty Time

Series Id olisikin väärin ja instanssit eivät tulisi näkyviin. Etusivun instanssi luettelosta löytyy null tai null,null arvot. Tämä tarkoittaa sitä, että Time Series ei ole tunnistanut Id:tä oikein. Valitsemalla null instanssin ja Event count tyyppiin saadaan kuitenkin tarkastettua, millaista tietoa tietokantaan tulee.



Kuva 28. Event Count (Niemi).

Tietotyyppien ja hierarkioiden asettamisen jälkeen, voitiin muokata instansseja. Yleensä instansseja on niin paljon, että kaikki eivät sovi listaan, tällöin voitiin käyttää hakutoimintoa avuksi. Instanssia muokkaamalla sille asetettiin edellä mainitut tietotyypit ja hierarkiarakenne. Hierarkia rakenteeseen voitiin nyt asettaa halutut arvot, jotka mahdollistavat rakennuksen eri kohteet.



Kuva 29. Instanssin tietotyyppien valinta (Niemi).

Edit 34, 1

Properties Instance fields

**Hierarchies**

Wärtsilä  
 Tikkarinne  
 Komminikoiva energia

**Instance fields**

Name	Value	
User (from hierarchy)	Ville	×
Building (from hierarchy)	A1	×
Floor (from hierarchy)	1	×
Room (from hierarchy)	201	×

[+ Add instance field](#)

Kuva 30. Instanssin hierarkia rakenne (Niemi).

Hierarkiaan lisätyt kohteet löytyvät nyt etusivulta. Hierarkia valinnan avulla voitiin valita haluttu hierarkia ja sen alta löytyivät sensorit.



Kuva 31. Sensorit hierarkian alla (Niemi).

### 5.7.1 Tietojen esittäminen graafien avulla

Perinteiseen tietokantaan verrattuna aikasarjatietokanta tarjosi datan esittämisen graafisena, mikä on varsinkin sensoridatalle toivottu ominaisuus. Graafien perusteella on helppo nähdä erilaisia poikkeamia datassa, kuten lämpötilanmuutoksia. Tietoja voitiin vertailla keskenään eri "uimalinjojen" avulla. Näin voitiin verrata korreloiko jokin poikkeama toisen tiedon poikkeaman kanssa.



Kuva 32. Tietoja ”uimalinjoilla” (Niemi).

## 5.8 Digitaalisen kaksosen hyödyntäminen

Tietokannan ollessa kunnossa siirryttiin digitaalisen kaksosen suunnitteluun. Kävi ilmi, ettei tietokannasta tai IoT Hubista voi viedä dataa suoraan digitaaliseen kaksoseen, vaan se tarvitsee väliin Azure Functions -palvelun. Azure Functions toimii rajapintana IoT Hubin ja digitaalisen kaksosen välillä. Azure Functions on tarvittaessa suoritettava ohjelma, joka voidaan koodata paikallisella koneella ja siirtää sen jälkeen Azuren portaaliin. Azure Functionsin koodaamisessa käytettiin Visual Studiota ja se kirjoitettiin C# kielellä. Microsoftin dokumenteista löytyi pohja koodille, joka siirtää dataa IoT Hubista digitaaliseen kaksoseen. Hyödynsin valmista pohjaa, mutta koodiin täytyi tehdä muutoksia sen ollessa vanhentunut. Koodi sisälsi myös osion, jossa IoT Hubiin tulleesta datasta irrotetaan runko, josta löytyvät esimerkiksi lämpötila, kosteus ja sensorin yksilöllinen tunniste. Kun koodi oli saatu valmiiksi, se siirrettiin Deploy -ominaisuudella Azuren portaaliin.

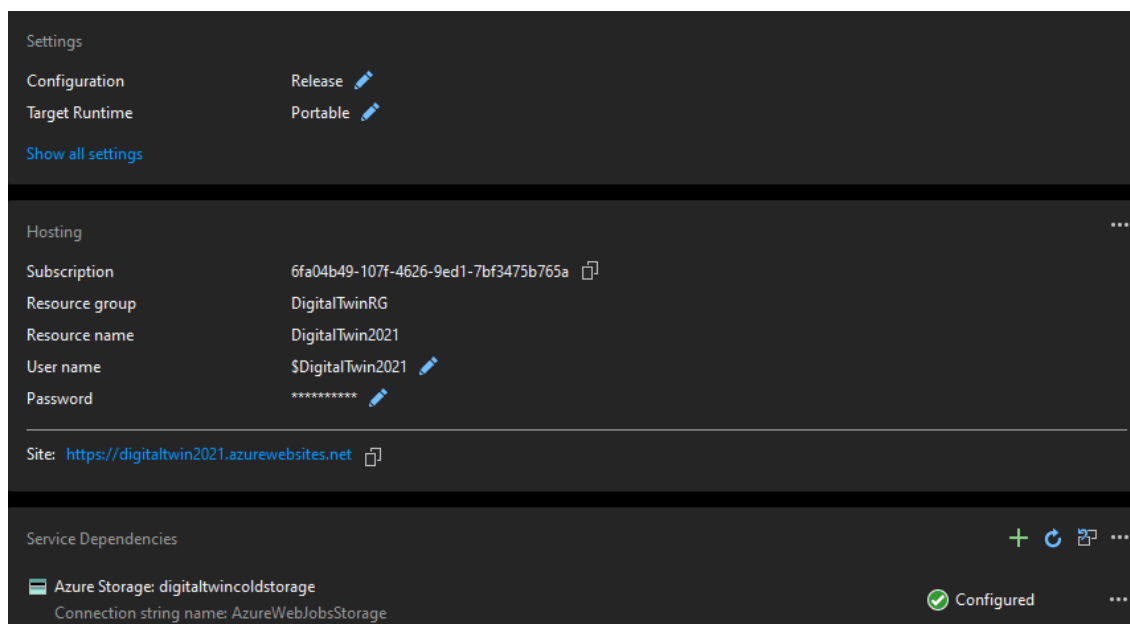
```
0 references
public async void Run([EventGridTrigger] EventGridEvent eventGridEvent, ILogger log)
{
    if (adtInstance
    // Authenti
    var cred = new ManagedIdentityCredential("https://digitaltwins.azure.net");
}
void IoTHubtoTwins.Run(EventGridEvent eventGridEvent, ILogger log)
AZF0001: Async void can lead to unexpected behavior. Return Task instead.
Show potential fixes (Alt+Enter or Ctrl+.)
```

Kuva 33. Vanhentunut osio koodista (Niemi).

```
0 references
public static async Task Run([EventGridTrigger] EventGridEvent eventGridEvent, ILogger log)
{
}
```

Kuva 34. Korjattu koodi (Niemi).

Julkaisun yhteydessä Azureen luotiin Functions App -palvelu. Paikallisella koneella kirjaututtiin Azuren portaaliin, josta haettiin tilauksen tiedot ja valittiin oikea varastointipaikka funktiolle. Azure Functions -palvelu varastoitiin samaan paikkaan kuin aikasarjatietokanta kylmävarastoi tietonsa.



Kuva 35. Functions publish (Niemi).

Azure Functions rakennettiin EventGridTrigger -nimisellä ominaisuudella. Sen vuoksi vaadittiin tilauksella aktiiviseksi EventGrid -resurssintarjoaja. Kyseinen resurssintarjoaja ei ollut aktiivisena tilauksella, joten se jouduttiin aktivoimaan manuaalisesti. Tilauksen nimellä hakuvalikosta etsimällä siirryttiin tilauksen hallintaan. Vasemmasta reunasta löytyi valikko resource providers, jota klikkaamalla päästiin etsimään ja aktivoimaan vaadittu resurssintarjoaja.

Provider	Status
Microsoft.EventGrid	Registered

Kuva 36. EventGrid aktivoituna (Niemi).

IoT Hub vaati event subscriptionin eli tapahtuman tilauksen, jotta se voi lähettää tietoja Azure Functionssiin. Tapahtuman tilaus tehtiin IoT Hubin etusivulla sijaitsevan events -osion alta. Event subscription -painikkeella päästiin luomaan uutta tilausta. Tilauksen määrittelyssä tärkeää oli laittaa lähde eli IoT Hub oikein ja asettaa tapahtuman tyyppiä telemetria. Päätepisteeksi valittiin Azure Functions.



Home > Azure-DigitalTwin-IoTHub1 >

## Create Event Subscription

Event Grid

Basic Filters Additional Features Delivery Properties

Event Subscriptions listen for events emitted by the topic resource and send them to the endpoint resource. [Learn more](#)

### EVENT SUBSCRIPTION DETAILS

Name \*

Event Schema

### TOPIC DETAILS

Pick a topic resource for which events should be pushed to your destination. [Learn more](#)

Topic Type IoT Hub

Source Resource Azure-DigitalTwin-IoTHub1

System Topic Name \*

### EVENT TYPES

Pick which event types get pushed to your destination. [Learn more](#)

Filter to Event Types

### ENDPOINT DETAILS

Pick an event handler to receive your events. [Learn more](#)

Endpoint Type \* Azure Function (change)

Endpoint \* [Select an endpoint](#)  
 Please select an endpoint

Kuva 37. Event Subscription (Niemi).

Digitaalisen kaksosen asentaminen Azuren portaaliin oli yksinkertaista. Ensin palvelu täytyi etsiä kuten aiemmatkin palvelut, joita oli asennettu. Sen jälkeen asetuksista täytyi vain lisäillä oikea tilaus ja resurssiryhmä.

Home > Create a resource > Marketplace > Azure Digital Twins >

## Create Resource

Azure Digital Twins

\* Basics \* Networking Advanced Tags Review + create

### Summary

Subscription	6198 Digital Twin
Resource group	DigitalTwinRG
Name	DigitalTwinKarelia
Region	North Europe

Kuva 38. Digitaalisen kaksosen luonti (Niemi).

## 5.9 DTDL kieli

Azuren digitaalisen kaksosen mallit rakennettiin JSON-kieltä muistuttavan DTDL-kielen avulla. Työssä käytettiin valmista mallia Microsoftin dokumentaation perusteella. Mallin voi rakentaa millä tahansa tekstieditorilla.

```
{
  "@id": "dtmi:contosocom:DigitalTwins:Thermostat;1",
  "@type": "Interface",
  "@context": "dtmi:dtdl:context;2",
  "contents": [
    {
      "@type": "Property",
      "name": "Temperature",
      "schema": "double"
    }
  ]
}
```

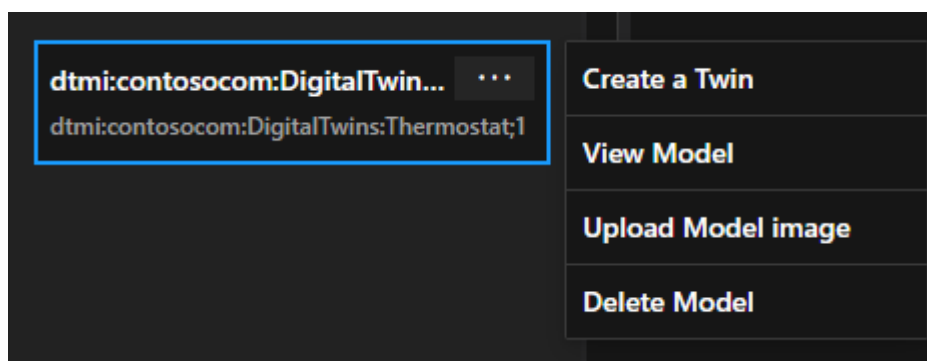
Kuva 39. Harjoitus malli (Niemi).

Mallin rakentamisen jälkeen se täytyi siirtää Azuren digitaaliseen kaksoseen. Siirtäminen voitiin tehdä usealla tapaa, mutta opinnäytetyössä se toteutettiin Digital Twin Explorerin avulla. Se löytyi Digital Twin -palvelusta Azuren portaalista.



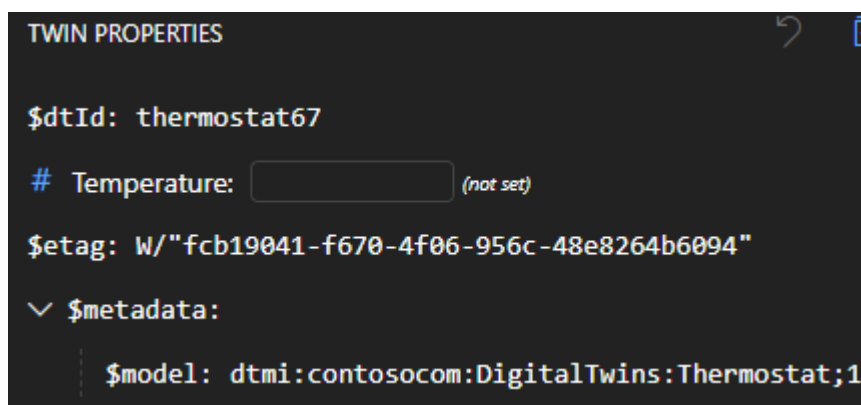
Kuva 40. Digital Twin Explorer (Niemi).

Explorerissa hallitaan digitaalisen kaksosen malleja kätevästi. Mallit ladattiin palveluun paikalliselta koneelta Models-osion alta löytyvän nuolen avulla. Kun malli oli palvelussa, se ilmestyi vasempaan reunaan. Hiiren oikealla näppäimellä luotiin mallista digitaalinen kaksonen Create a Twin -valinnan avulla.



Kuva 41. Mallista kaksosiksi (Niemi).

Digitaaliset kaksoset ilmestyivät ruudun keskelle erivärisiksi palloiksi. Palloja klikkaamalla saatiin kaksosen tarkemmat tiedot näkyville. Tässä vaiheessa kaksosen tiedot olivat vielä tyhjä, koska tietoa ei ollut vielä lähetetty.



Kuva 42. Kaksosen tiedot (Niemi).

## 5.10 Tiedon lähettämisen ongelmat

Digitaalisen kaksosen mukaan tulon myötä aiempi Raspberry simulaattori alkoi aiheuttamaan ongelmia. IoT Hubin ja digitaalisen kaksosen välissä pyörivä Azure Functions ei tunnistanut Raspberyllä tulevaa sanomaa. Ongelmaa alettiin selvittämään koodauksen avulla, koska IoT Hub vaatii telemetrisen sanoman koodauksen olevan UTF-8, 16 tai 32. Paikalliselle koneelle asennettiin Azure IoT Explorerin, jonka avulla pystyi seuraamaan hubiin luotuja laitteita ja niille tulevia viestejä. Vaikka viestin runko oli näkyvässä IoT Explorerissa, se ei silti toiminut Azure Functionssissa.

```
{
  "body": {
    "temperature": 29.517592529254113
  },
  "enqueuedTime": "Wed Oct 20 2021 16:44:16 GMT+0300 (Itä-Euroopan kesäaika)",
  "properties": {
    "temperatureAlert": "false"
  },
  "systemProperties": {
    "iothub-connection-device-id": "thermostat67",
    "iothub-connection-auth-method": "{\"scope\":\"device\", \"type\":\"sas\", \"",
    "iothub-connection-auth-generation-id": "637664313006840927",
    "iothub-enqueuedtime": 1634737456204,
    "iothub-message-source": "Telemetry",
    "x-opt-sequence-number": 231255,
    "x-opt-offset": "38685288392",
    "x-opt-enqueued-time": 1634737456313
  }
}
```

Kuva 43. Viesti Azure IoT Explorerissa (Niemi).

```
getMessage(function (content, temperatureAlert) {
  var message = new Message(content, content_encoding = "utf-8", content_type = "application/json");
  message.properties.add('temperatureAlert', temperatureAlert.toString());
}
```

Kuva 44. Raspberry koodin muokkausta (Niemi).

```
{ "properties": { "temperatureAlert": "false" }, "systemProperties": { "iothub-connection-device-id": "thermostat67", "iothub-connection-auth-method": "
{"scope\":\"device\", \"type\":\"sas\", \"issuer\":\"iothub\", \"acceptingIpFilterRule\":null}", "iothub-connection-auth-generation-id": "637664313006840927",
"iothub-enqueuedtime": "2021-10-20T13:38:56.662Z", "iothub-message-source": "Telemetry" }, "body":
"eyJ0ZW1wZXJhdHVyZS16MjQuNjYyZk40Mjk0NTc1MDg4fQ==" }
```

Kuva 45. Viesti muutosten jälkeen (Niemi).

## 5.11 Simulaattorin vaihto

Microsoftin sivuilta löytyy ohje IoT Hubin ja digitaalisen kaksosen väliselle tiedonsiirrolle. Ohjeesta löytyi C# -kielellä kirjoitettu simulaattori. (Microsoft documentation. 2021h.) Koodiin täytyi tehdä muutamia muokkauksia, jotta se vastasi tietokannan rakennetta. IoT Hubiin yhteyden ottaminen määritettiin koodissa yhteysmerkkijono -muuttujilla. Niihin täytyi määritellä opinnäytetyön IoT Hubin ja luodun virtuaalisen laitteen osoite.

```
private const string iotHubConnectionString =
// <summary>
// Please replace with correct device connect
// The device connect string could be got fro
// </summary>
private const string deviceConnectionString =
```

Kuva 46. Simulaattorin yhteysmerkkijonot (Niemi).

Lähetetty viesti täytyi muokata vastaamaan aikasarjatietokannan rakennetta. Viestin täytyi sisältää DeviceID ja Telemetry.SensorID. C# simulaattorissa viestin koodaus ja Contenttype oli määritelty selkeästi.

```
var Telemetrydatapoint = new
{
    DeviceID = "Classroom_213",
    Telemetry = new object[] {
        new { SensorID = "1", Temperature = currentConsumption},
        new { SensorID = "2", CO2 = CO2 },
    }
};

var messageString = JsonSerializer.Serialize(Telemetrydatapoint);
var message = new Microsoft.Azure.Devices.Client.Message(Encoding.UTF8.GetBytes(messageString))
{
    ContentType = "application/json",
    ContentEncoding = "utf-8"
};
```

Kuva 47. Simulaattorin viesti ja koodaus (Niemi).

Simulaattorin vaihdon jälkeen viestin koodausongelmat ratkesivat ja Azure Functions tunnisti tulevan viestin rungon ja pystyi ottamaan sensorin lähettämät tiedot.

Message	Type
Executing 'IOTHubToTwins' (Reason='EventGrid trigger fired at 2021-11-22T14:38:24.2541650+00:00', Id=71e176a0-4b49-40a4-a821-d26a61d3f753)	Information
ADT service client connection created.	Information
{ "properties": {}, "systemProperties": { "iothub-content-type": "application/json", "iothub-content-encoding": "utf-8", "iothub-connection-device-id": "thermostat67", "iothub-connection-auth-method": "{ \"scope\": \"device\", \"type\": \"sas\", \"issuer\": \"iothub\", \"acceptingIpFilterRule\": null }", "iothub-connection-auth-generation-id": "637664313006840927", "iothub-enqueuedtime": "2021-11-22T14:38:23.94Z", "iothub-message-source": "Telemetry" }, "body": { "Temperature": 67.61254794551644 } }	Information
Device:thermostat67 Temperature is:67.61254794551644	Information

Kuva 48. Viestin runko tunnistettuna (Niemi.)

Tiedot menivät digitaaliseen kaksoseseen onnistuneesti. Digitaalisessa kaksosessa täytyi muistaa antaa jokin arvo esimerkiksi 0 digitaalisen kaksosen mallin aloitusarvoksi. Jos arvoa ei muistanut syöttää Azure Functions ei tunnistanut digitaalisen kaksosen mallin lämpötilakenttää

```
$dtId: thermostat67
# Temperature: 67,3268451 ×
$etag: W/"442d1ca4-f36b-4e2c-a7e2-eb3294c2c779"
$metadata:
  $model: dtmi:contosocom:DigitalTwins:Thermostat;1
  Temperature:
    lastUpdateTime: 2021-11-24T12:57:17.8300636Z
```

Kuva 49. Tiedot päivittävät (Niemi.)

## 6 Tulokset

Tuloksena saatiin onnistunut tietoalusta digitaaliselle kaksoselle. Tiedot siirtyivät mittauspisteiltä erilaisten rajapintojen avulla IoT Hubiin. Opinnäytetyössä käytettiin simulaattoreita, mutta muut projektiin osallistujat käyttivät fyysisiä laitteita, joiden avulla tieto siirtyi samalla tavalla tietokantaan. Projektipäällikkö lisäsi opinnäytetyön loppuvaiheilla Ruuvi Gateway -nimisen rajapinnan ja onnistui liittämään sen tietokantaan. Valittu tietokanta, Time Series Insights, sopi täydellisesti IoT datan käsittelyyn ja sen avulla voidaan seurata reaaliaikaisesti graafien avulla datassa tapahtuvia muutoksia.

Digitaalisesta kaksosesta luotiin mallit ja ne ladattiin Azuren palveluun. Digitaalinen kaksonen vaati toimiakseen Azure Functions -nimisen palvelun aktivoinnin. Palvelun aktivointi onnistui ja paikallisella koneella tehty funktio sovellus ladattiin Azuren portaaliin. Funktio pyörii taustalla ja suoritetaan aina tarvittaessa, kun IoT Hubiin tulee dataa. Funktio lähetti tietoa eteenpäin digitaaliselle kaksoselle onnistuneesti esimerkiksi käytetylle lämpötilasensorille ja sensorin arvo muuttui.

## 7 Pohdinta

### 7.1 Työn jatkaminen

Opinnäytetyötä on mahdollista jatkaa eteenpäin tehdyn toteutuksen pohjalta. Azure Functions -palvelun saattaminen toimivaksi, on yksi tärkeimmistä ominaisuuksista digitaalisen kaksosen toimivuuden kannalta. Functions -koodi, joka on kirjoitettu C# -kielellä, toimii tällä hetkellä vain osittain. Koodin pitäisi osata lukea automaattisesti tulevan sensoridatan runko ja ottaa sen perusteella tarvittavat tiedot ja siirtää ne digitaaliseen kaksoseen.

Digitaalinen kaksonen -palvelu aktivoitiin ja sinne ovat rakennettiin omia malleja, mutta Azure Functionssin toimimattomuuden vuoksi niitä ei voinut testata. On hyvinkin mahdollista, että DTDL-kielen avulla tehdyt mallit sisältävät

joitain asiavirheitä. Mallien tarkastaminen ja sovittaminen yhteen toimivan datan kanssa on tarpeen.

Työhön on hyvä liittää myös automatisoituja toimintoja. Näillä automatisoiduilla toiminnoilla voidaan esimerkiksi sulkea ilmastointi tai lämmitys tiettyjen raja-arvojen puitteissa. Azuresta löytyy esimerkiksi Logic Apps -palvelu, jonka avulla voidaan tehdä automatisoituja toimenpiteitä IoT datan perusteella.

## **7.2 IFC-malli**

Työhön on tarkoitus liittää myös kolmiulotteinen malli Wärtsilän rakennuksesta. Jotta palvelut olisivat yhteydessä keskenään ja vaihtaisivat dataa, malli täytyisi ladata jollain tavalla Azuren portaaliin. Mallin tuominen digitaalisen kaksosen käyttöön vaatii tarkempaa perehtymistä ja tekniikoiden selvittämistä. Unreal Enginen avulla on mahdollista ainakin avata ja tarkastella IFC-mallia. Malliin olisi tarkoitus sijoitella myös sensorit, ja mittauspisteet, jotka sijaitsevat Wärtsilässä. Mallin pitäisi sisältää mahdollisuus tarkastella sensoreiden tuottamaa dataa.

## Lähteet

- Aggerwal, A. 2019. Introduction to Visual Studio.  
<https://www.geeksforgeeks.org/introduction-to-visual-studio/>.  
 7.11.2021.
- Bigelow, S. 2020. Microsoft Azure  
<https://searchcloudcomputing.techtarget.com/definition/Windows-Azure>.7.11.2021.
- Devasia, A. 2020. Digital Twinning: Leaders in the Emerging Technology.  
<https://control.com/technical-articles/digital-twinning-leaders-in-the-emerging-technology/>.3.10.2021.
- Fuller, A. Fan, Z. Day, C. Barlow, C. 2020. Digital Twin: Enabling Technologies, Challenges and Open Research.  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9103025>.7.11.2021.
- Laakkonen, O. 2021.Sähköpostikeskustelu.4.11.2021.
- Microsoft documentation. 2021a. IoT Concepts and Azure IoT Hub  
<https://docs.microsoft.com/en-us/azure/iot-hub/about-iot-hub>  
 10.08.2021.
- Microsoft documentation.2021b. Welcome to Azure Stream Analytics.  
<https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction>.7.11.2021.
- Microsoft documentation.2021c. Welcome to Azure CosmosDB.  
<https://docs.microsoft.com/en-us/azure/cosmos-db/introduction>.  
 7.11.2021.
- Microsoft documentation 2021d. Introduction to Azure Functions.  
<https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview>.7.11.2021.
- Microsoft documentation. 2021e. Ingest IoT Hub telemetry into Azure Digital Twins. <https://docs.microsoft.com/en-us/azure/digital-twins/how-to-ingest-iot-hub-data?tabs=cli>.7.11.2021. 25.11.2021
- Microsoft documentation. 2021f. What is Azure Digital Twins?  
<https://docs.microsoft.com/en-us/azure/digital-twins/overview>.  
 25.11.2021.
- Microsoft documentation. 2021g. Install and use Azure IoT Explorer.  
<https://docs.microsoft.com/en-us/azure/iot-fundamentals/howto-use-iot-explorer> 7.11.2021.
- Microsoft documentation. 2021h. Azure Digital Twins end-to-end samples  
<https://docs.microsoft.com/en-us/samples/azure-samples/digital-twins-samples/digital-twins-samples>.30.11.2021.
- Panetta, K. 2019 Gartner top 10 strategic technology trends for 2020.  
<https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020>.7.11.2021
- Shaw, K. Fruhlinger, J. 2019. What is digital twin and why it's important to IoT  
<https://www.networkworld.com/article/3280225/what-is-digital-twin-technology-and-why-it-matters.html>. 7.11.2021.
- Wikipedia. 2021a. Microsoft Azure.  
[https://en.wikipedia.org/wiki/Microsoft\\_Azure](https://en.wikipedia.org/wiki/Microsoft_Azure). 7.11.2021.
- Wikipedia. 2021b. CosmosDB.  
[https://en.wikipedia.org/wiki/Cosmos\\_DB](https://en.wikipedia.org/wiki/Cosmos_DB).7.11.2021.