

Google Cloud Platform & IoT Core



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Kevät 2018

Joni Korhonen

Tietojenkäsittelyn koulutusohjelma
Visamäki, Hämeenlinna

Tekijä	Joni Korhonen	Vuosi 2018
Työn nimi	Google Cloud Platform & IoT Core	
Työn ohjaaja/t	Tommi Lahti, Lauri Salminen	

TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli esitellä yleisesti, mikä on Google Cloud Platformin IoT Core ja miten sen ja IoT-laitteen välille luodaan yhteys. Työn toimeksiantajana oli Hämeen ammattikorkeakoulun Älykkäät palvelut, jonne IoT Core on mahdollisesti tulossa käyttöön. Työ tarjoaa toimeksiantajalle pohjatietoa.

Opinnäytetyön teoriaosuudessa esitellään aluksi IoT lyhyesti käsitteenä ja käydään läpi työssä käytetty IoT-laite Raspberry Pi, sen käyttöjärjestelmä Raspbian ja laitteeseen kuuluvia lisäkomponentteja. Tämän jälkeen esitellään Google Cloud Platform ja sen sisältämiä palveluita. Käytännön osuudessa esitellään demo, jossa toteutetaan yhteyden luominen ja esitellään sen toimivuutta lähettämällä mittaustuloksia Google Cloud Platformiin.

Luodussa demossa luotiin yhteys Raspberry Pin ja Google IoT Coren välille MQTT-protokollaa hyödyntäen. Tätä luotua yhteyttä hyödynnettiin lähettämällä Raspberry Pihin liitetyn lämpötila- ja ilmankosteusanturin mittamaa dataa Cloud Platformin palveluihin jatkokäsittelyä varten.

Avainsanat Internet of Things, Google Cloud Platform, IoT Core, Raspberry Pi

Sivut 25 sivua, joista liitteitä 0 sivua

Degree Programme in Business Information Technology
Campus

Author	Joni Korhonen	Year 2018
Subject	Google Cloud Platform & IoT Core	
Supervisors	Tommi Lahti, Lauri Salminen	

ABSTRACT

The goal of this Bachelor's thesis was to introduce Google Cloud Platform's IoT Core and how to create connection between IoT Core and IoT Device. The topic of the work was commissioned by the Älykkäät palvelut of Häme University of Applied Sciences. The client may use IoT Core in the future and this thesis would be basis for them.

The theoretical part of the thesis presents IoT as a concept, used IoT device Raspberry Pi, its operating system Raspbian and Add-on Components. After these Google Cloud Platform and its services are presented. In the practical part of the thesis it was meant to create demo where the connection is created and used to save measured data.

In the demo connection between Raspberry Pi and IoT core is created with MQTT protocol. This connection is used to save data obtained by the temperature-humidity sensor which is connected to Raspberry Pi. This data is then sent to Google Cloud platform for further processing.

Keywords Internet of Things, Google Cloud Platform, IoT Core, Raspberry Pi.

Pages 25 pages including appendices 0 pages

Sanasto

IoT	Internet of Things eli esineiden internet. Nimitys internettiin yhteydessä olevista laitteista, jotka keräävät ja analysoivat dataa automaattisesti.
API	Application Programming Interface eli ohjelmointirajapinta, jonka avulla sovellukset voivat kommunikoida.
Cloud SDK	Cloud Software Development Kit, kokoelma Cloud -työkaluja.
MQTT	Message Queuing Telemetry Transport, viestintäprotokolla.
JSON Web Token	Standardi, jolla voidaan hallinnoida käyttöoikeuksia
GPIO	General Purpose I/O, portti johon voidaan yhdistää dataa lähettävä tai vastaanottava komponentti
Node.js	Kehitysympäristö JavaScriptille
Skripti	Komentosarja joka voidaan lukea ajon aikana
SSL/TLS	Transport Layer Security, salausprotokolla

SISÄLLYS

1	JOHDANTO.....	1
2	INTERNET OF THINGS	2
3	RASPBERRY PI	3
3.1	Raspberry Pi yleisesti.....	3
3.2	Raspbian	4
3.3	KytKentälevy.....	4
3.4	Komponentit	4
4	GOOGLE CLOUD PLATFORM.....	6
4.1	Cloud SQL	6
4.2	Cloud IoT Core	6
4.3	HTTP & MQTT	7
4.4	Cloud Pub/Sub.....	8
4.5	Google Cloud Functions	9
4.6	Tietoturva	9
5	RASPBERRY PIN KÄYTTÖÖOTTO	10
6	GOOGLE CLOUD PLATFORMIN KÄYTTÖÖNOTTO	11
6.1	Cloud Platform -projektin luonti	11
6.2	Cloud IoT Core API.....	12
6.3	Google Cloud SDK.....	12
6.4	Laiteavainparit.....	14
6.5	Laitteen lisääminen IoT Coreen.....	15
6.6	Pub/Sub API.....	17
7	DEMON RAKENTAMINEN	18
7.1	DHT11 Anturin käyttöönotto	18
7.2	MQTT Kirjastot	19
7.3	Viestin lähettäminen Pub/Sub topiciin	20
7.4	Cloud Functions API.....	21
8	YHTEENVETO	23
	LÄHTEET	24

1 JOHDANTO

Opinnäytetyön tilaajana on Hämeenlinnan ammattikorkeakoulun Älykkäät palvelut -tutkimusyksikkö. Google Cloud Platform ja sen tarjoama IoT Core alusta ovat ehkä tulossa tilaajalle käyttöön myöhemmin, joten opinnäytetyö tulee tarvittaessa toimimaan pohjamateriaalina Älykkäille palveluille.

Aiheen valintaan vaikuttivat henkilökohtainen mielenkiinto IoT-laitteita kohtaan ja erilaiset mahdollisuudet hyödyntää niitä. Erityisen mielenkiintoiseksi aiheen tekee se, että voi toteuttaa IoT-laitteen asennuksen ja yhdistämisen pilvipalveluun käytännössä.

Opinnäytetyön tavoitteena on luoda yhteys IoT-laitteen ja Google Cloud Platform pilvipalvelun välille. Google tarjoaa tällaisen yhteyden luomista varten IoT Core -palvelun. Yhteyden luomisen jälkeen toteutetaan yksinkertainen demo, jolla osoitetaan yhteyden toimivuus.

Teoriaosuudessa esitellään IoT yleisesti käsitteenä ja se miksi se on niin iso ja puhuttu aihe. Teoriaosuudessa käydään läpi mikä on Raspberry Pi ja sen tarjoamat IoT mahdollisuudet, Google Cloud Platform ja IoT Core palvelu.

Demossa päätelaitteena toimii Raspberry Pi -tietokone, jolle on tarkoitus luoda yhteys Google Cloudiin ja hyödyntää luotua yhteyttä tallentamalla päätelaitteen mittaamaa dataa verkon yli Google Cloud Platformiin. Opinnäytetyön käytännön osuudessa käydään läpi demon rakentaminen ja käyttöönotto vaihe vaiheelta mahdollisten kuvien kera.

Opinnäytetyö vastaa seuraaviin tutkimuskysymyksiin:

Mikä on Google Cloud Platform ja sen IoT Core? Miten Google Cloud IoT Coren ja laitteen välille luodaan yhteys? Miten Google Cloudin ja IoT laitteen välille luotua yhteyttä voidaan hyödyntää mittaustulosten X tallentamisessa?

2 INTERNET OF THINGS

Internet of Things (IoT) eli esineiden internet on käsite, jolla tarkoitetaan laitteita jotka ovat yhteydessä internettiin ja tätä myötä älykkäitä. Kaikki internettiin yhdistetyt laitteet eivät kuitenkaan automaattisesti ole älykkäitä, ollakseen älykäs IoT-laite tulee sen kerätä ja analysoida dataa automaattisesti, ilman ihmisen osallistumista. (Lawrence Miller 2017, 3)

IoT-laitteita voidaan kuvailla kolmen ydinosan mukaan: Itse fyysinen laite, laitteen älykkyys ja laitteen yhdistettävyyys. Fyysisellä laitteella tarkoitetaan mekaanisia tai elektronisia päätelaitteita. Laitteen älykkyydellä tarkoitetaan sitä, että laiteessa on esimerkiksi erilaisia sensoreita, joiden myötä laitteen toiminta on pitkälle kehittyntä. Laitteen yhdistettävyydellä tarkoitetaan sitä, että laite on yhdistetty internettiin ja voi vastaanottaa tai lähettää tietoa internetin välityksellä (Michael Porter & James Heppelmann 2014).

Älykkäiden IoT-laitteiden myötä informaation kulku ja töiden automatisointi helpottuu. Esimerkiksi suomalainen yritys Enevo tarjoaa järjestelmää, jossa jäteastioihin kiinnitetään anturit, jotka mittaaavat jäteastian täyttöastetta. Tämä mahdollistaa sen, että jäteauto ei turhaan aja tyhjentämään vajaan tai tyhjää jäteastiaa. Myös erilaisten työvälineiden kuntoa ja huoltotarvetta voitaisiin mitata automaattisesti ja näin ollen tehdä ennakoivaa huoltoa ja välttää töiden katkeaminen (Eskonen 2015).

Monet isot IT-alan yritykset ovat panostaneet IoT teknologiaan, sen tarjotaan ja kehittämiseen. Alan suurimpia toimijoita ovat: Google, Amazon, IBM ja Microsoft. Tässä opinnäytetyössä paneudutaan Googlen tarjoamaan palveluun.

IoT:hen liittyy myös varjopuolia tietoturvan, datan määrän ja ekologisuuden osalta. Datan määrä tulee lisääntymään räjähdysmäisesti IoT-laitteiden myötä ja tähän tulee kulumaan paljon energiaa (Eskonen 2015).

Tietoturva on olennainen kysymys kaiken kohdalla, mikä liitetään internettiin, miten data ja itse laitteet voidaan suojata ulkoisilta hyökkäyksiltä. Tero Jänisvaara on tehnyt hyvän opinnäytetyön IoT:n tietoturvasta HAMK:ille vuonna 2017.

3 RASPBERRY PI

3.1 Raspberry Pi yleisesti

Raspberry Pi on isobritannialaisen Raspberry Pi Foundationin kehittämä noin luottokortin kokoinen minitietokone, jossa kaikki tarvittavat peruskomponentit ovat yhdessä piirilevyssä. Raspberry Pi Foundation on hyväntekeväisyysäätiö, jonka ideana on kehittää tietojenkäsittelyn oppimista kouluissa. Raspberry Pi -tietokoneet ovat tämän takia hinnaltaan mahdollisimman edullisia, mutta tämän takia tulee myös rajoituksia käytettävien komponenttien ja suorituskyvyn suhteen, kuten taulukosta 1. käy ilmi. (Raspberry Pi Foundation n.d.).

Edullisuutensa, kokonsa ja kehitettävyytensä takia Raspberry Pit ovat suosittuja rakentelun kohteita alan harrastajien parissa ja Raspberry Pi Foundation mainostaakin, että vain taivas on rajana laitteen kehitettävyydessä. Luonnollisesti kehitettävyytensä takia Raspberry Pi -tietokoneet ovat erinomaisia IoT-laite-alustoja.

Raspberry Pi:ssä on ulkoisten komponenttien liittämistä varten useita General Purpose Input/Output (GPIO) pinnejä. Nämä pinnit mahdollistavat komponenttien helpon kytkemisen ja käyttöönoton. Laitteessa voi olla useitakin eri komponentteja samaan aikaan kytkettynä (Raspberry Pi Foundation n.d.).

Taulukko 1. Raspberry Pi -tietokoneiden spesifikaatiot ja hinnat. (Raspberry Pi Foundation n.d.)

Tuote	SoC	Nopeus	RAM	USB Portit	Ethernet	Wireless/Bluetooth	Hinta
Raspberry Pi Model A+	BCM2835	700Mhz	512MB	1	No	No	20\$
Raspberry Pi Model B+	BCM2835	700Mhz	512MB	4	Yes	No	25\$
Raspberry Pi 2 Model B	BCM2836 or BCM2837	900Mhz	1GB	4	Yes	No	35\$
Raspberry Pi 3 Model B	BCM2837	1200Mhz	1GB	4	Yes	Yes	35\$
Raspberry Pi Zero	BCM2835	1000Mhz	512MB	1	No	No	5\$
Raspberry Pi Zero W	BCM2835	1000Mhz	512MB	1	No	Yes	10\$

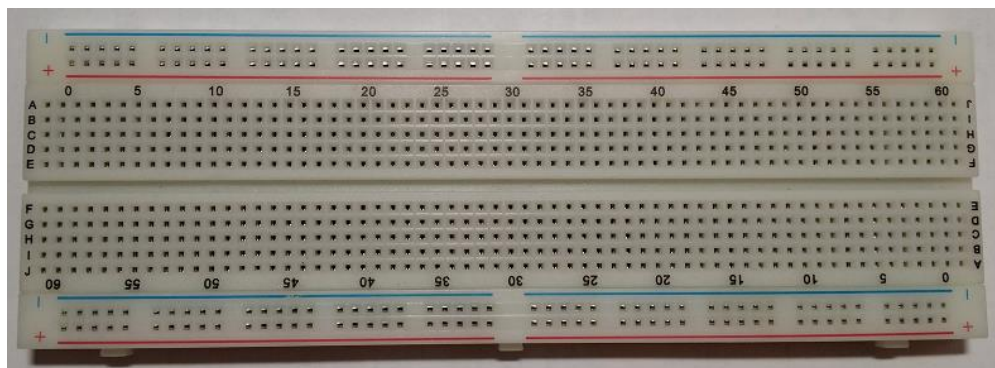
3.2 Raspbian

Raspberry Pi -tietokoneille on olemassa monia eri käyttöjärjestelmä vaihtoehtoja, mutta suosituin ja käytetyin käyttöjärjestelmä on Raspbian. Raspbian on Raspberry Pin virallinen oma käyttöjärjestelmä ja se on täysin ilmainen.

Raspbian on Debian pohjainen, avoimen lähdekoodin käyttöjärjestelmä, joka on täysin optimoitu Rasperry Pille. Käyttöjärjestelmä on jatkuvassa kehityksessä parhaan mahdollisen stabiliteetin ja suorituskyvyn aikaan saamiseksi. Raspbian sisältää myös yli 35 000 koottua asennusvalmista ohjelmistoa koodausta, opetuskäyttöä tai tietokoneen yleiskäyttöä varten (Raspbian n.d.). Tässä opinnäytetyössä käytetään käyttöjärjestelmänä Raspbiania sen mukautuvuuden ja helppokäyttöisyyden takia.

3.3 KytKentälevy

KytKentälevy (Kuva 1) mahdollistaa erilaisten antureiden ja LED-valojen liittämisen Raspberry Pihin ilman komponenttien erillistä juottamista piirilevyyn. KytKentälevyssä on reikiä joihin komponentit, johdot tai vaikkapa vastukset kytketään. KytKentälevy yhdistetään hyppylangoilla Raspberry Pin yleispinneille, jotka ovat tarkoitettu komponenttien liittämiseen.

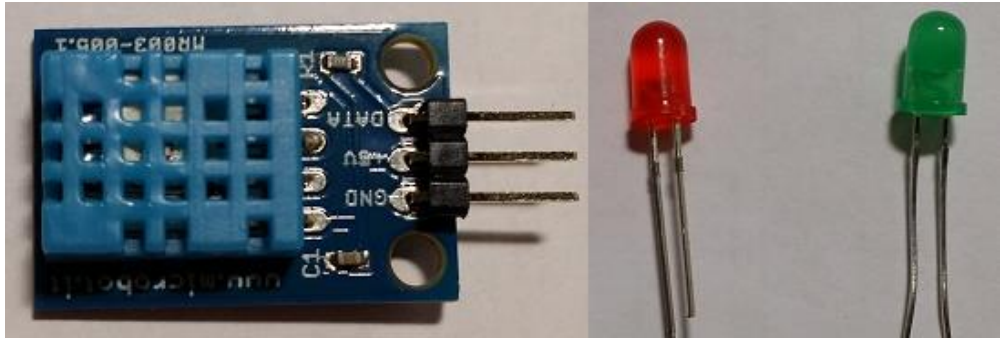


Kuva 1. KytKentälevy

3.4 Komponentit

Raspberry Pille on olemassa monia erilaisia komponentteja, joilla voidaan mitata erilaisia arvoja tai lähettää dataa. Suosittuja komponentteja ovat erilaiset LED-lamput (Kuva 2), näytöt, kamerat ja robotiikkaan liittyvät komponentit.

Tässä opinnäytetyössä hyödynnetään DHT11 nimistä sensoria (Kuva 2), jonka tarkoituksena on mitata lämpötilaa ja ilmankosteutta. Tämän komponentin mittaavat arvot hyödynnetään lähettämällä ne Google Cloud Platformiin IoT Coren välityksellä.



Kuva 2. Vasemmalla DHT11 sensori ja oikealla kaksi eriväristä LED-lamppua.

4 GOOGLE CLOUD PLATFORM

Pilvipalveluilla tarkoitetaan sitä, että perinteiset sovellus- ja laitteisto tuotteet ovat muuttuneet palveluiksi: ne eivät ole omalla tietokoneella vaan ne on ulkoistettu palveluntarjoajan palvelimelle. Pilvipalveluja on kolmea päätyyppiä ja ne ovat: Platform as a Service (PaaS), Infrastructure as a Service (IaaS) ja Software as a Service (SaaS). Pilvipalvelun etuna on ajankäytön ja työvoiman säästö, koska kaikki ylläpito siirtyy palveluntarjoajalle (Amazon n.d.).

Google Cloud Platform on Googlen tarjoama pilvipalvelualusta, johon kuuluu monia erilaisia tuotteita ja palveluita, esimerkiksi tietokantaratkaisuja, tietokannan analytiikkaratkaisuja, erilaisia virtuaalikoneita ja IoT Core. Google tarjoaa hyvän dokumentaation ja erilaisia kurseja, miten hyödyntää sen tarjoamia pilvipalveluita. Useimmat palvelut, dokumentaatiot ja kurssit ovat kuitenkin maksullisia, ja rekisteröityminen vaatii luottokortin tai pankkitiedot (Google n.d.).

Googlega on monia palvelinkeskuksia ympäri maailmaa. Nämä palvelinkeskukset sisältävät joukon erilaisia fyysisiä voimavaroja, esimerkiksi tietokoneita, kovalevyjä ja virtuaalisia resursseja, kuten virtuaalikoneita. Palvelinkeskusten hajauttamisen etuna on redundanssi virhetapauksien varalta sekä se, että tiedonvälityksen viiveet pysyvät pieninä (Google n.d.).

4.1 Cloud SQL

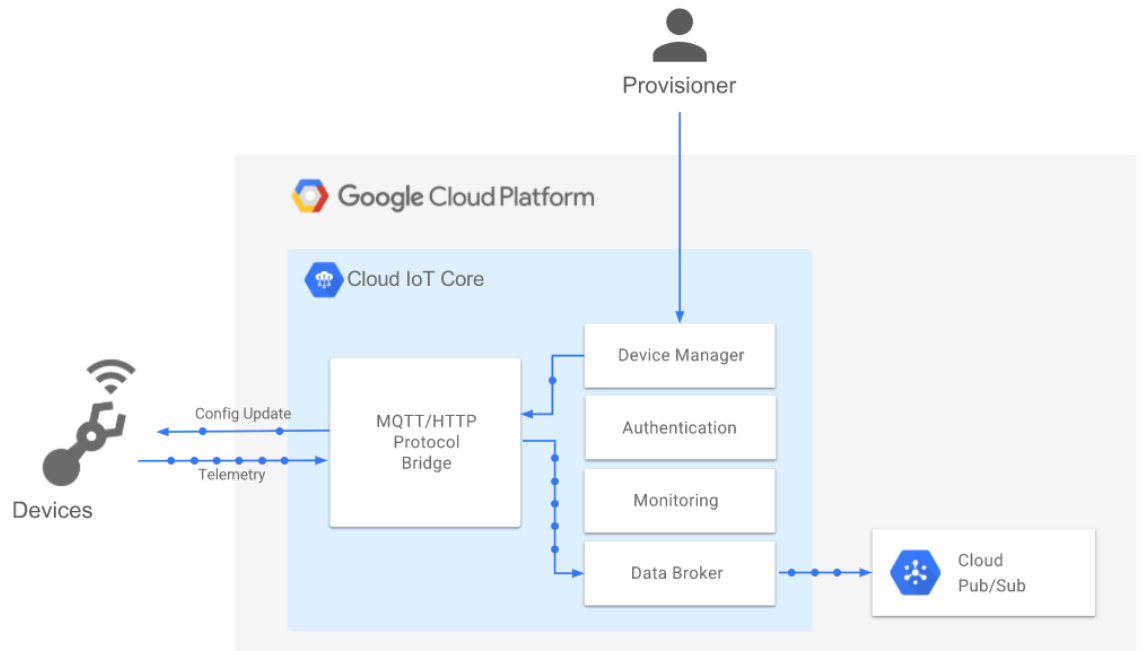
Cloud SQL on yksi Googlen tarjoamista pilvipalveluista. Se tarjoaa helposti toteutettavan tietokantaratkaisun joko PostgreSQL- tai MySQL-tietokantaohjelmistoilla joista PostgreSQL on vielä beta-vaiheessa. Palvelu tarjoaa tietokannan automaattisen varmuuskopiointin ja ajantasaisuuden, eikä palvelu vaadi minkäänlaista ohjelmiston asentamista (Google n.d.).

4.2 Cloud IoT Core

Cloud IoT Core Googlen vielä beta-vaiheessa oleva IoT-laitteille tarjoama palvelu. Palvelu mahdollistaa helpon ja turvallisen tavan yhdistää hallinnoida IoT-laitteita ja ratkaisun datan keräämiseen, prosessointiin ja analytiikkaan reaaliaikaisesti.

IoT Coren kaksi pääkomponenttia ovat laitehallinta ja protokollasillat. Laitehallinnan avulla päätelaitteet rekisteröidään ja niitä voidaan tämän avulla monitoroida ja konfiguroida. IoT Coren käyttämät protokollasillat ovat MQTT ja http. Näiden protokollien avulla päätelaite voidaan yhdistää Google Cloud Platformiin. Protokollasillat välittävät päätelaitteen telemet-

ria datan Cloud Pub/Sub otsakkeelle eli topicille, jonka avulla voidaan käynnistää haluttuja toimintoja eli eventtejä Cloud Platformissa (Cloud Functions) (Google n.d.). Kuvassa 3 käy ilmi, miten päätelaite voi hyödyntää yhteyttä IoT Coreen.



Kuva 3. Kaavio IoT Coren toiminnasta (Google n.d.)

4.3 HTTP & MQTT

HTTP (HyperText Transfer Protocol) on tiedonsiirtoprotokolla, jonka avulla päätelaitteet tai sovellukset kommunikoivat verkon yli palvelimien kanssa. HTTP-protokollan toiminta perustuu päätelaitteen lähettämiin pyyntöihin ja palvelimen lähettämiin vastauksiin. HTTP hyödyntää kommunikoinnissa TCP (Transmission Control Protocol) -yhteyttä (Bradley Mitchell 2017).

MQTT (MQ Telemetry Transport) on kevyt Publish/Subscribe-viestintäprotokolla, joka HTTP:n tavoin hyödyntää TCP:t kommunikoinnissa. MQTT on ns. machine-to-machine protokolla, joka on erittäin hyödyllinen IoT-laitteille. MQTT:n suurimpia etuja ovat jokaisessa sen välittämässä viestissä oleva pieni koodinpätkä ns. sormenjälkenä, viestin tietosisältö on rajoitettu 256 megabittiin ja se sisältää määritellyn Quality Of Service taso. Viestin rajaaminen 256 megabittiin mahdollistaa MQTT protokollan keveyden.

Quality Of Service mahdollistaa MQTT viestien perille pääsemisen, vaikka yhteydet ovat huonoja. Quality Of Service mahdollistaa viestien uudelleen lähetyksen ja takaa sen, että lähetetty viesti pääsee perille. Quality Of Ser-

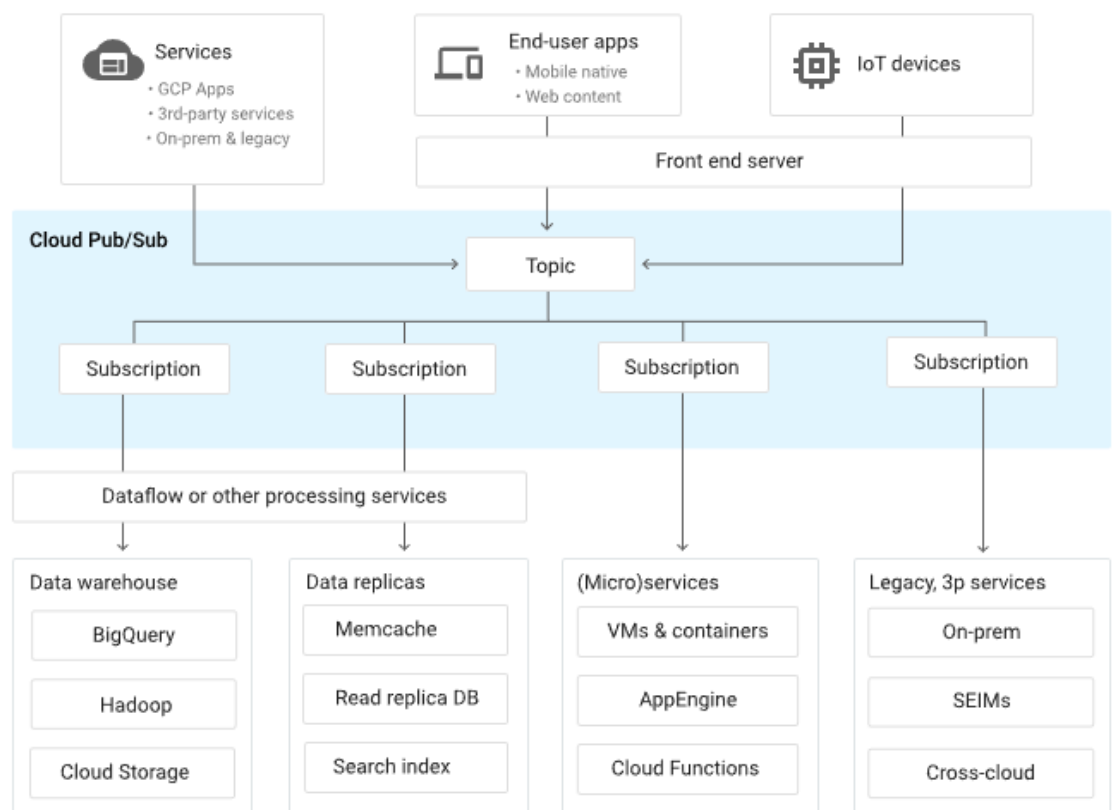
vice pitää sisällään kolme eri tasoa (0–2). Mitä korkeampi taso on, sitä luotettavampi. Toisaalta Quality Of Servicen korkea taso aiheuttaa viivettä ja vaatii enemmän kaistaa yhteydeltä.

MQTT-sessio on jaettu neljään eri osa-alueeseen: yhdistäminen, käyttöoikeuksien tarkistaminen, kommunikointi ja yhteyden päättäminen. MQTT-vakioportit ovat: 1883 kryptaamattomalle kommunikaatiolle ja 8883 kryptatulle kommunikaatiolle (Margaret Rouse 2015).

Tämän opinnäytetyön demossa hyödynnetään MQTT protokollaa viestien lähettämisessä Googlen Pub/Sub API:iin.

4.4 Cloud Pub/Sub

Cloud Pub/Sub, eli Cloud Publisher and Subscriber. Publisherilla tarkoitetaan applikaatioita jotka voivat tehdä HTTPS kyselyitä googleapis.com:ille. Nämä applikaatiot voivat olla esimerkiksi sovelluksia mobiililaitteissa, erilaisia web palveluita tai IoT -laitteita. Nämä applikaatiot lähettävät viestejä topic resurssiin, josta subscriber applikaatiot voivat hakea lähetetyt viestit HTTPS kyselyillä (Kuva 4). Nämä publisherin lähettämät viestit ovat yhdistelmä attribuutteja ja dataa. Näin mahdollistetaan aSynkronoitu monelta-monelle viestintä, jonka avulla itsenäiset applikaatiot voivat turvallisesti kommunikoida toisilleen (Google n.d.).



Kuva 4. Cloud Pub/Sub toimintaperiaate (Google n.d.)

4.5 Google Cloud Functions

Google Cloud Functions on serveritön ajoympäristö, joka on tarkoitettu Cloud-palvelujen rakentamiseen ja yhdistämiseen. Cloud Functionin avulla voidaan luoda yksinkertaisia yhtä tarkoitusta palvelevia funktioita, jotka ovat yhdistetty Cloud-palveluista saapuviin tapahtumiin eli eventteihin. Funktio ajetaan, kun tarkkailtava eventti toteutuu. Tämän on hyödyllistä silloin kun esimerkiksi monet IoT-laitteet lähettävät dataa Cloud Pub/Subiin käynnistäen funktioita, joiden avulla voidaan prosessoida ja säilöä saapunutta dataa (Google n.d.).

4.6 Tietoturva

Google Cloud Platform käyttää laitekohtaista public/private avaimia ja JSON Web tokeneita (JWT) laitteiden ja käyttäjän tunnistamiseen. Laitteelle tulee luoda ensimmäisenä avainpari, joista public avain tulee syöttää laitetta lisätessä Google Cloud Platformiin. Jos laite yhdistetään MQTT protokollalla, Cloud Platformiin tulee laitteen luoda JSON Web Token, joka on allekirjoitettu laitteen private avaimella. Laite esittää yhdistyessään tämän JSON Web Tokenin, jota MQTT silta vertaa syötettyyn public avaimen.

Google Cloud Platform tukee RSA ja Elliptic Curve salausalgoritmeja ja näille voidaan määrittää eräntymispäivät. Luotuja salausavaimia voidaan myös tarvittaessa kierrättää (Google n.d.).

Google tarjoaa myös sertifikaattipaketin, joka toimii lisäturvana public/private avainparien ja JSON Web tokenin ohella. Tämä sertifikaatti tarkistaa, että public/private avainparit kuuluvat oikealle laitteelle. sertifikaattitiedosto tulee olla samassa kansiossa kuin ajettavat python tiedostot (Google n.d.).

5 RASPBERRY PIN KÄYTTÖÖTTO

Raspbian käyttöjärjestelmän asennus Raspberry Pihin oli yksinkertainen prosessi. Raspberryn kotisivuilta löytyi ohjeet asennukseen sekä latauslinkit Raspbianin levykuvaan ja ohjelmaan, jolla voidaan asentaa se muistikortille. Tämän jälkeen muistikortti laitetaan Raspberryyin ja Raspbian on käyttövalmis.

Käyttöjärjestelmän asentamisen jälkeen ensimmäiseksi päivitettiin Raspbian uusimpaan versioon. Tämän jälkeen päivitettiin myös Raspberry Pin firmware yhteensopivuus ongelmien välttämiseksi. Ruudunkaappauksia varten asennettiin Scrot-niminen sovellus, jolla voidaan ottaa terminaali-komennoilla ruudunkaappauksia halutulla tavalla, esimerkiksi määritetyllä viiveellä.

Raspberryn Pin lokalisaatio ja näppäimistön asettelu ovat oletuksena englanniksi, näiden asetuksia voidaan muuttaa terminaalin kautta.

Raspberry Pin konfiguraatio

```
sudo raspi-config
```

Edelle esitelty komento avaa Raspberry Pin konfiguraatiovalikon, jonka alta löytyy asetukset lokalisaatiolle ja näppäimistön asettelulle.

Käyttöönotossa ilmeni kuitenkin ongelma, kun Raspberry rupesi kesken käytön kaatuilemaan ja ruudun oikeaan yläkulmaan ilmestyi virheilmoitus, joka indikoi virran puutteesta. Ongelmaa yritettiin aluksi ratkaista kokeilemalla toista virtalähdettä, toista SD-korttia tuoreella Raspbian asennuksella ja testaamalla laitteen keskusmuistia. Lopulta ongelma saatiin ratkaistua hankkimalla Raspberry Pin virallinen virtalähde.

6 GOOGLE CLOUD PLATFORMIN KÄYTTÖNOTTO

Ensimmäinen vaihe Google Cloud Platformin käyttöönotossa on luoda Google Käyttäjätili. Google tarjoaa vuoden mittaisen kokeilujakson Cloud Platformiin ja 300 dollarin edestä krediittiä, joita voi käyttää Platformin tarjoamiin palveluihin. Kun krediitit tai aikajakso on kulutettu loppuun, tulee käyttäjän päivittää tili maksulliseen versioon, joka veloittaa käyttäjältä ker- ran kuukaudessa käytön mukaan kasautuneen rahasumman. Google ha- luaa käyttäjältä maksukortin tiedot rekisteröityessä varmistaakseen, ettei kyseessä ole robotti. Minkäänlaisia maksuja ei veloiteta käyttäjältä ennen kuin hän itse päivittää tilin maksulliseen versioon (Google n.d.).

6.1 Cloud Platform -projektin luonti

Käyttäjätilin luonnin jälkeen valitaan tai luodaan uusi projekti Cloud Plat- formiin. Google luo käyttäjätilin ohessa myös valmiin oletusprojektin ja an- taa sille valmiin ID:n, tässä opinnäytetyössä käytetään tätä oletusprojektia ja se on uudelleen nimetty lotRaspiksi. Luotuun projektiin pitää myös liit- tää olemassa oleva tai luoda uusi laskutustili. Tälle tilille voidaan määrittää projektikohtainen budjetti, ettei käytön mukaan kasvava lasku muodostu liian suureksi. Tilille voidaan määrittää monta eri maksukorttia, jotka voi- daan yhdistää eri projekteihin tarpeen vaatiessa. Kuvassa 5 on nähtävillä uusi projekti ja sille luotu laskutustili.

The image shows two screenshots of the Google Cloud Platform interface. The top screenshot is the 'Manage resources' page, which displays a table of projects. The bottom screenshot is the 'Billing' page, showing the 'Overview' section with 'Credits remaining' of €241.62 and 'Days remaining' of 364, ending on Feb 6, 2019.

Project name	Project ID
lotRasp	shining-domain-194413

Project name	Project ID
lotRasp	shining-domain-194413

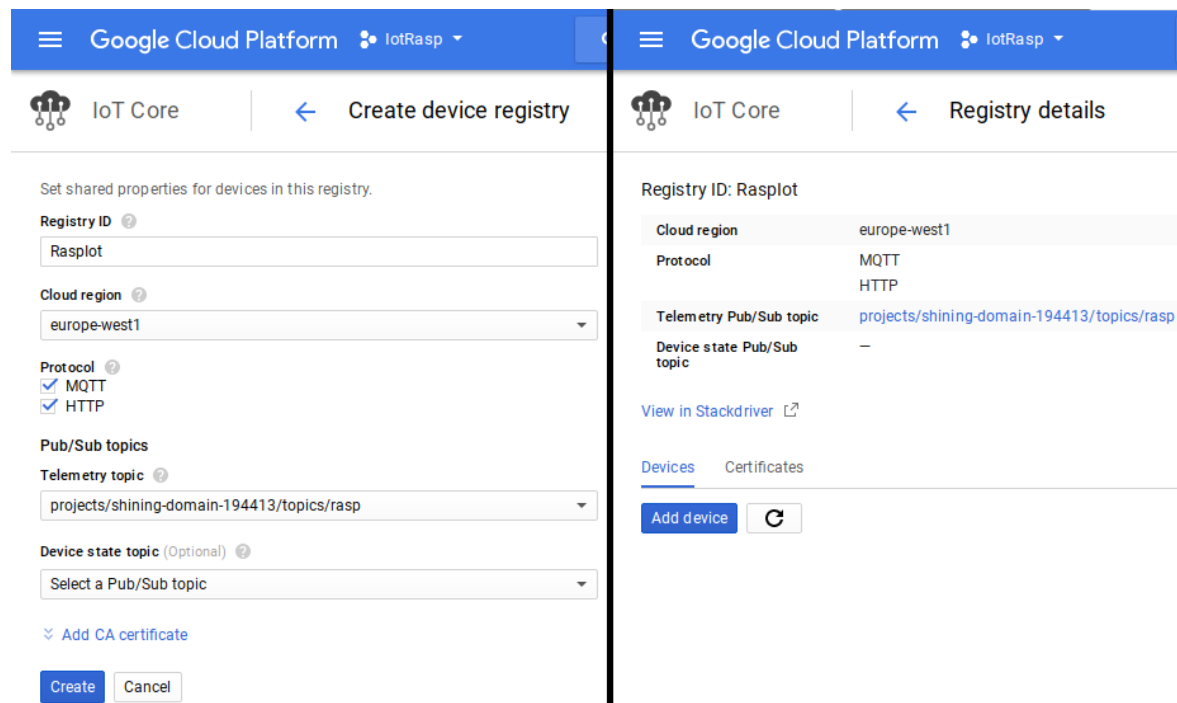
Project name	Project ID
lotRasp	shining-domain-194413

Kuva 5. Cloud Platform -projektin luonti ja siihen liitetty laskutustili

6.2 Cloud IoT Core API

IoT Core API voidaan pistää päälle Cloud Platformin konsolista. Sen luomiseen tarvitaan edellä luodun projektin ID. IoT Core API tarvitsee toimiakseen Device Registryn, johon voidaan lisätä APIa käyttävät laitteet ja tarvittavat konfiguraatiot. Device Registryyn määritellään ID, käytettävä pilvipalvelimen sijainti, käytettävä yhdistämisprotokolla ja topic Pub/Sub moduulille. palvelimen tulee olla mahdollisimman lähellä, jotta viive datan lähetyksessä ja vastaanotossa ei ole liian pitkä.

Projektin ID:ksi määriteltiin Rasplot ja lähin valittavissa oleva palvelin alue on Länsi-Eurooppa. Yhdistämisprotokollaksi valittiin HTTP ja MQTT, jotta on mahdollista testata tarpeen vaatiessa molempia protokollia. Pub/Sub moduulille annettiin aluksi luodun projektin ID ja topicin nimeksi rasp (Kuva 6).



Kuva 6. Vasemmalla Device Registryn luonti ja oikealla valmis Registry

6.3 Google Cloud SDK

Jotta IoT Corea pystytään hallinnoimaan ja konfiguroimaan, on asennettava Google Cloud SDK -paketti, joka on kokoelma erilaisia työkaluja tällaisia toimintoja varten. Asentamalla tämä SDK-paketin saadaan käyttöön gcloud, gsutil ja bq komentorivikomennot.

Jotta Google Cloud SDK saadaan käyttöön, tulee käyttäjän ajaa seuraavat komennot terminaalissa Raspberry Piillä

Luodaan ympäristömuuttuja oikealle jakeluversiolle

```
export CLOUD_SDK_REPO="cloud-sdk-$(lsb_release -c -s)"
```

Lisätään Cloud SDK:n jakelu URI paketin lähteeksi

```
echo "deb http://packages.cloud.google.com/apt $CLOUD_SDK_REPO
main"
```

```
| sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list
```

Tuodaan Google Cloudin julkinen avain

```
curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-
key add -
```

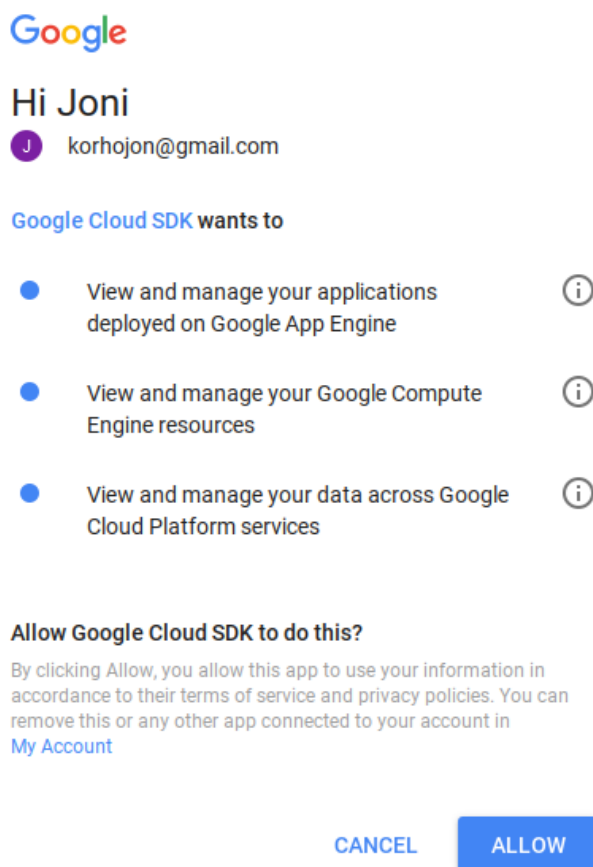
Päivitetään ja asennetaan Cloud SDK

```
sudo apt-get update && sudo apt-get install google-cloud-sdk
```

Käynnistetään Cloud SDK

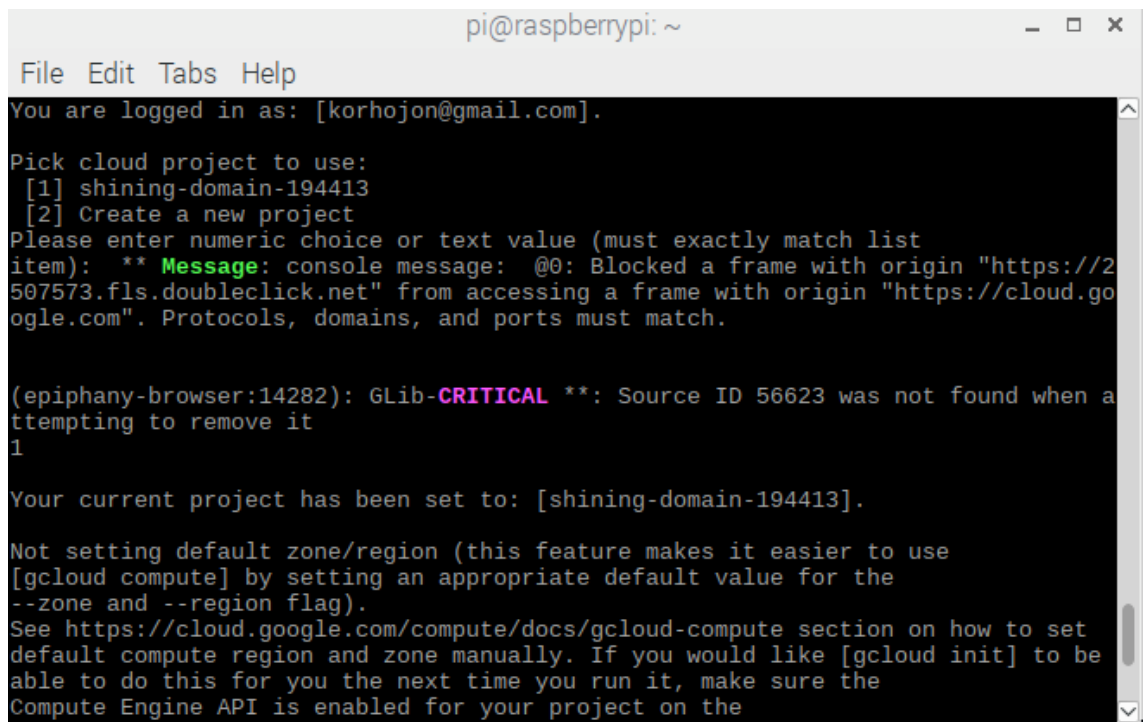
```
gcloud init
```

Näiden komentojen jälkeen laitteeseen asentuu Cloud SDK paketti ja asennus avaa selaimen ohjatakseen käyttäjän kirjautumaan Google tiliinsä sille todentaakseen käyttäjän oikeaksi ja pyytääkseen oikeuksia tiettyihin toimintoihin (Kuva 7).



Kuva 7. Kirjautumisen jälkeen Cloud SDK kysyy oikeuksia toimintoihin

Kun sisäänkirjautuminen ja oikeuksien antaminen on suoritettu, voidaan valita aktiivinen projekti terminaalista. Cloud SDK listaa löydetyt projektit allekkain tai vaihtoehtoisesti voi myös luoda uuden projektin. Haluttu projekti valitaan antamalla terminaaliin sitä vastaava numero (Kuva 8).



```

pi@raspberrypi: ~
File Edit Tabs Help
You are logged in as: [korhojon@gmail.com].
Pick cloud project to use:
  [1] shining-domain-194413
  [2] Create a new project
Please enter numeric choice or text value (must exactly match list
item): ** Message: console message: @0: Blocked a frame with origin "https://2
507573.fl.s.doubleclick.net" from accessing a frame with origin "https://cloud.go
ogle.com". Protocols, domains, and ports must match.

(epiphany-browser:14282): GLib-CRITICAL **: Source ID 56623 was not found when a
ttempting to remove it
1
Your current project has been set to: [shining-domain-194413].
Not setting default zone/region (this feature makes it easier to use
[gcloud compute] by setting an appropriate default value for the
--zone and --region flag).
See https://cloud.google.com/compute/docs/gcloud-compute section on how to set
default compute region and zone manually. If you would like [gcloud init] to be
able to do this for you the next time you run it, make sure the
Compute Engine API is enabled for your project on the

```

Kuva 8. Aktiivisen projektin valinta.

Cloud SDK:lle voidaan asentaa vaihtoehtoisia komponentteja, joita on monia erilaisia eri käyttökohteita varten. Tätä opinnäytetyötä varten asennutettiin seuraavat lisäkomponentit:

- google-cloud-sdk-app-engine-python
App Engine työkalu Python sovelluksien hallintaan
- google-cloud-sdk-app-engine-java
App Engine työkalu Java sovelluksien hallintaan
- google-cloud-sdk-pubsub-emulator
Pub/Sub emulaattori, jonka avulla voidaan testata sovelluksia lokaalisti.

6.4 Laiteavainparit

Ennen kuin laiterekisteriin lisätään laite, tulee tälle luoda ensin public/private avainpari. Jokainen päätelaite luo JSON Web Tokenin, joka on allekirjoitettu private-avaimella. Tämän avaimen IoT Core varmentaa käyttämällä laitteen public-avainta.

Tätä projektia varten luotiin 2048-bittiset RS256-avaimet ilman eräänymispäivää. Avaimet voidaan luoda seuraavilla komennoilla:

Private avain

```
openssl genrsa -out rsa_private.pem 2048
```

Public avain

```
openssl rsa -in rsa_private.pem -pubout -out rsa_cert.pem
```

Nämä komennot luovat kaksi tiedostoa: `rsa_private.pem`, joka pitää sisällään private avaimen ja `rsa_cert.pem`, joka pitää sisällään public avaimen mitä IoT Core käyttää varmentaakseen päätelaitteen. Public avainta tarvitaan, kun IoT Coreen halutaan lisätä uusi laite ja tämän avaimen voi lukea suoraan terminaalista komennolla tai avaamalla kyseinen tiedosto sen tallennussijainnista.

Komento public avaimen lukuun

```
less rsa_cert.pem
```

6.5 Laitteen lisääminen IoT Coreen

Kun edellä esitellyt vaiheet on suoritettu, voidaan haluttu päätelaite lisätä IoT Core laitteeksi sen device registryyn. Ensiksi laite nimetään antamalla sille Device ID, tämän jälkeen voidaan päättää sallitkaanko laitteen kanssa kommunikointi. Yleensä aina kommunikointi sallitaan, mutta jos esimerkiksi laite ei jostain syystä toimi halutulla tavalla voidaan kommunikointi aluksi estää kokonaan. Seuraavaksi syötetään edellä luotu public avain, joka voidaan syöttää manuaallisesti tai tiedosto voidaan ladata suoraan konsoliin. Tässä tapauksessa avain syötetään manuaalisesti valitsemalla ensiksi avaimen tyyppi (RS256) ja syöttämällä luotu public avain sille tarkoitettuun kohtaan. Näiden tarvittavien pakollisten tietojen jälkeen voidaan lisätä päätelaite uudeksi IoT Core laitteeksi. Kuvassa 9 näkyvät syötetyt tiedot Cloud Platformissa ja lisätty laite.

Vaihtoehtoisina asetuksina voidaan määrittää public avaimelle eräänymispäivä ja laitteelle metadataa esimerkiksi sarjanumero, joita ei tässä projektissa määritelty.

IoT Core | Add device

Device ID [?]
Rasp3

Device communication [?]
 Allow
 Block

Authentication (Optional) [?]
Input method
 Enter manually
 Upload

Public key format
 RS256 [?]
 ES256 [?]
 RS256_X509 [?]
 ES256_X509 [?]

Public key value

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtIXQ0wsDzaTug3Xj0/Ur
N2JT6rmuUIJv5BGMIPgRAahC2WIGSPQXZYEfYSW4BfJ4Evs9i f jNwLynh27h7Bw
1e10EBJhpPkPq6w4kGkxFRY0TRAoXR5SRD4DXy/XPCtVXLz55KB17EUC08TsaKgC
/X1GZKqIa6Ycmv4eAodYgGVYpABPjWjNHX6t1YbQ3TZmevy4ovaU9Djmj7xWoQe1
GRcCtmfjg097AMS27uzpAWhs/xBUqLv1FLTvGbWz1CDMchAuLwwCj+37xGvmVo9T9
Kw9Hek/XD06DuyDTmN53/kNqT9ncA0USCpv9kQ7t/yfCNGTI7Nwph1MGEsaxC4a
kQIDAQAB
-----END PUBLIC KEY-----
```

Public key expiration date (Optional)
 Expires on:

Device metadata (Optional) [?]
Key must contain only letters, numbers, hyphens, and underscores, and be no longer than 128 characters

Key	Value
+ Add attribute	

IoT Core | Device details

Device ID: Rasp3
Numeric ID: 2891591985361829 Registry: Rasplot
Device communication: ✔ Allowed

[Details](#) [Configuration & state history](#)

Latest activity

Heartbeat (MQTT only)	—
Telemetry event received	—
Device state event received	—
Config sent	—
Config ACK (MQTT only)	—
Error	—

Device metadata
None

Authentication

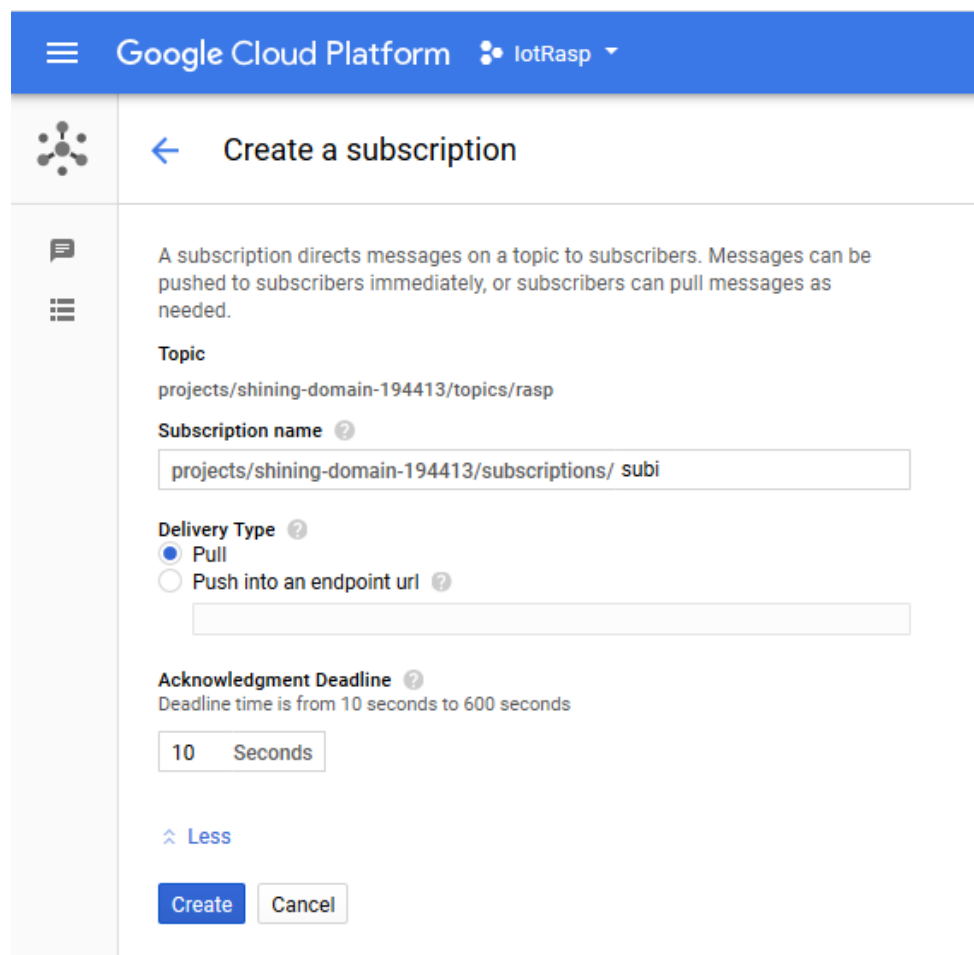
Key format	Key value
<input type="checkbox"/> RS256	**** ..****

Kuva 9. Vasemmalla laitteen lisäys ja oikealla lisätty laite, joka näkyy device registryssä.

6.6 Pub/Sub API

Pub/Sub API voidaan ottaa käyttöön Google Cloud Platformin konsolista ja kuten aiemmissa tapauksissa myös sen käyttöönottoon tarvitaan laskutus-tili ja olemassa oleva projekti johon se voidaan liittää.

Ensimmäiseksi luodaan Pub/Sub topic, joka nimetään halutunlaiseksi, tässä tapauksessa nimeksi annettiin rasp. Topicin luonnin jälkeen tulee sille luoda subscription, joka ohjaa topicin viestit subscribereille joille viestit voidaan välittää välittömästi tai ne voidaan hakea tarvittaessa. Subscriptionia luodessa tulee sille antaa nimi, delivery type ja vaihtoehtoisesti acknowledgment deadline. Jos Delivery type on push, välittyvät viestit heti, kun ne ovat julkaistu. Jos Delivery type taas on pull, tulee viestit erikseen pyytää. Acknowledgment deadline määrittää kuinka kauan Pub/Sub odottaa subscriberin saavan ja kuittaavan viestin ennen kuin lähettää tämän viestin uudelleen. Tähän demoon määritettiin Delivery typeksi: Pull, koska demon viestien lähetystä on turha automatisoida ja Acknowledgment deadline jäi oletusarvoon (Kuva 10).



The screenshot shows the 'Create a subscription' page in the Google Cloud Platform console. The page title is 'Create a subscription' with a back arrow. Below the title, there is a description: 'A subscription directs messages on a topic to subscribers. Messages can be pushed to subscribers immediately, or subscribers can pull messages as needed.' The form fields are: 'Topic' with the value 'projects/shining-domain-194413/topics/rasp'; 'Subscription name' with the value 'projects/shining-domain-194413/subscriptions/ subi'; 'Delivery Type' with 'Pull' selected (radio button) and 'Push into an endpoint url' unselected; and 'Acknowledgment Deadline' with a value of '10' and the unit 'Seconds'. At the bottom, there are 'Create' and 'Cancel' buttons.

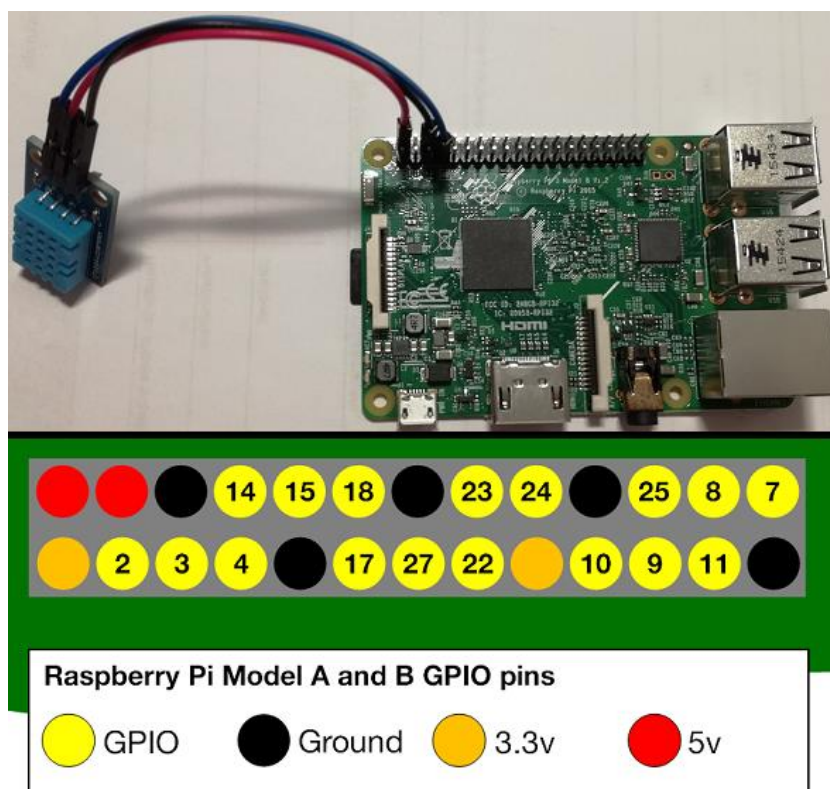
Kuva 10. Subscriptionin luonti projektin Pub/Sub topiciin

7 DEMON RAKENTAMINEN

Kun kaikki tarvittavat Google Cloud Platformin -palvelut ovat otettu käyttöön ja konfiguroitu halutulla tavalla, voidaan aloittaa demona rakentaminen. Demossa on tarkoituksena esitellä, miten mitata dataa Raspberry Pillä ja miten tätä dataa käsitellään Google Cloud Platformin -palveluissa.

7.1 DHT11 Anturin käyttöönotto

DHT11 Anturi asennetaan muiden lisäkomponenttien tavalla liittämällä se hyppylangoilla Raspberry Pin GPIO pinneihin. Anturi voidaan lisätä suoraan Raspberryn pinneihin tai käyttämällä kytkentälevyä ns. välikätenä. Kytkentälevyn käyttö on suotavaa silloin, jos asennettava anturi on paljon virtaa käyttävä ja näin ollen se tarvitsee vastuksen kytkennän väliin, jotta vältetään vahingoittamasta itse Raspberrystä. Tässä demossa käytetty DHT11 anturi voidaan huoletta kytkeä suoraan Raspberryn kiinni. Anturissa on kolme liitäntää: 5V virtaliitäntä, maadoitus ja datan lähetys. Nämä liitännät yhdistetään hyppylangoilla vastaaviin Raspberrissä oleviin liitäntöihin käyttäen datan lukua varten GPIO 4 liitäntää.



Kuva 11. Yllä anturi liitetty Raspberryn ja pinnikaavio (Raspberry Pi Foundation n.d.)

Adafruit elektroniikkavalmistaja tarjoaa valmiin python kirjaston DHT11 anturille, joka on ladattavissa ilmaiseksi yrityksen Githubistaan (Adafruit 2017).

DHT11 Python kirjasto

git clone https://github.com/adafruit/Adafruit_Python_DHT.git

Kirjasto voidaan tuoda python koodiin antamalla: import Adafruit_DHT. Tämän jälkeen määritellään sensori muuttujaan käytettävä sensori, eli DHT11 ja pinnimuuttujaan se Raspberry Pin GPIO pinni mihin DHT11 datan lähetys on yhdistetty. Tämän jälkeen voidaan hakea lämpötila- ja ilman- kosteusmuuttujille arvot ajamalla DHT11 ohjelma, näin arvot voidaan tu- lostaa komentoriviin luettavaksi (Kuva 12).

```
import sys
import Adafruit_DHT

sensori = Adafruit_DHT.DHT11
pinni = 4 # GPIO numero (Pinni 7)

# Ajetaan DHT ohjelma ja tulostetaan lämpötila ja ilmankosteus
lampo, ilma = Adafruit_DHT.read_retry(sensori, pinni)
print 'Lämpötila {}C Ilmankosteus {}'.format(lampo, ilma)
```

Kuva 12. Koodiesimerkki miten lämpötila ja ilmankosteus saadaan mitat- tua ja tulostettua.

7.2 MQTT Kirjastot

Jotta viestejä saataisiin lähetettyä MQTT avulla Google Cloudin Pub/Sub topiciin tulee ensin asentaa muutamia erilaisia kirjastoja seuraavilla ko- mennoilla.

SSL/TLS kirjasto jota MQTT käyttää viestien kryptaamiseen

apt install build-essential libssl-dev libffi-dev python-dev

JSON Web Token ja MQTT kirjastot pythonille

pip install pyjwt paho-mqtt cryptography

Päivitetään tai asennetaan Google Cloudin python ympäristö

pip install --upgrade google-api-python-client

Päivitetään tai asennetaan Google Cloudin IoT Core ympäristö

pip install --upgrade google-cloud-core

Päivitetään tai asennetaan Google Cloudin Pub/Sub ympäristö

pip install --upgrade google-cloud-pubsub

Seuraavaksi haetaan Googlen tarjoama CA sertifikaattipaketti, jolla kappaleessa 6.4 luodut avainparit voidaan tarkistaa. Näin voidaan olla varmoja, että avainparit kuuluvat oikealle laitteelle.

Google CA sertifikaatti

wget <https://pki.google.com/roots.pem>

7.3 Viestin lähettäminen Pub/Sub topiciin

Google tarjoaa valmiin python koodin, jolla voidaan luoda JWT Token, avata MQTT yhteys IoT Coreen ja lähettää viesti Pub/Sub topiciin. Koodissa on myös parset joilla se saa muuttujiin annetut arvot komentorivi komenosta. Tätä kyseistä koodia on hyödynnetty tässä demossa ja siihen on lisätty kappaleessa 7.2 esitelty DHT11 koodi, jotta lämpötila ja ilmankosteus saadaan lähetettyä viestinä Pub/Sub topiciin (ARM Software 2017 ja Google n.d.).

Viesti lähetetään ajamalla komentorivillä python tiedosto missä edellä mainittu koodi on. Tämän tiedoston nimen jälkeen annetaan Google Cloud projektin id, device registryn id, device registryyn lisätyn laitteen nimi, luotu private key ja siinä käytetty salausalgoritmi.

Viestin lähetys

```
python test2.py --project_id=shining-domain-194413 --registry_id=Rasplot --device_id=Rasp3 --private_key_file=rsa_private.pem --algorithm=RS256
```

Lähetetty viesti voidaan lukea seuraavalla gcloud komennolla ja antamalla siihen Google Cloudissa olevan Pub/Sub subscriptionin osoite.

Viestin luku

```
gcloud beta pubsub subscriptions pull --auto-ack projects/shining-domain-194413/subscriptions/subi
```

Kuvassa 13 on nähtävillä viestin lähetys ja sen noutaminen, jos viestin on saapunut perille ja se onnistuneesti noudettu tulostuu kuvan 13 kaltainen viesti komentoriville.

```
root@raspberrypi:/home/pi# python test2.py --project_id=shining-domain-194413 --registry_id=Rasplot --device_id=Rasp3 --private_key_file=rsa_private.pem
Creating JWT using RS256 from private key file rsa_private.pem
Publishing message Rasplot/Rasp3 Time 2018-02-26 17:06:43 temperature 22.0C & humidity 34.0%
('on_connect', '0: No error.')
on_publish
Finished.
root@raspberrypi:/home/pi# gcloud beta pubsub subscriptions pull --auto-ack projects/shining-domain-194413/subscriptions/subi
```

DATA	MESSAGE_ID	ATTRIBUTES
Rasplot/Rasp3 Time 2018-02-26 17:06:43 temperature 22.0C & humidity 34.0%	44575683065604	deviceId=Rasp3 deviceNumId=2891591985361829 deviceRegistryId=Rasplot deviceRegistryLocation=europe-west1 projectId=shining-domain-194413 subFolder=

Kuva 13. Onnistunut viestin lähetys ja sen noutaminen

7.4 Cloud Functions API

Julkaistuja viestejä voidaan hyödyntää esimerkiksi Cloud Functioneissa ja jotta voidaan käyttää Cloud Functioneita pitää API ottaa käyttöön edellisten tapaan Cloud Platformin konsolista. Cloud Platform hyödyntää node.js ympäristöä, joka ajaa JavaScript koodia. Node.js kehitysympäristö pitää asentaa ensimmäisenä, jotta voidaan luoda skriptejä Cloud Platformille.

Ensimmäiseksi asennetaan Node Version Manager, jolla hallitaan Node.js asennuksia. Huomioitavaa on, että Node Version Manager ei tue natiivina Windows käyttöjärjestelmiä. Node Version Manager voidaan asentaa hakemalla wget komennolla valmis asennusskripti.

Asennusskripti

```
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.8/install.sh | bash
```

Asennuksen tarkistus

```
command -v nvm
```

Tämän jälkeen voidaan asentaa itse Node.js ympäristö ja npm, jolla voidaan asentaa paketteja Node.js:lle.

Node.js asennus

```
nvm install stable
```

Määritellään asennettu versio oletukseksi

```
nvm alias default stable
```

npm työkalun asennus

```
npm install --save express
```

Näiden pakettien asennuksen jälkeen on Node.js ympäristö valmis ja voidaan Javascript tiedostoja Cloud Functionille.

Seuraavaksi tulee luoda kansio haluttuun sijaintiin etuliitteellä: gfc_. Tässä demossa luotiin kansio nimeltä gfc_DhtData. Tähän kansioon sijoitetaan kaikki ajettavat koodit mitä Cloud Functions tulee käyttämään. Kansioon luodaan index.js niminen JavaScript tiedosto, johon haluttu koodi sijoitetaan. Google tarjoaa pienen esimerkki koodin, miten noutaa Pub/Sub topiikkiin lähetetty viesti, tätä kyseistä koodinpätkää on hyödynnetty tässä demossa, jotta saadaan mitattu lämpötila ja ilmankosteus printattua Google Cloud Platformin konsolilokiin (Google n.d.).

Luotuja koodeja voidaan tarkistella myös Cloud konsolin kautta (Kuva 14), mutta niihin ei voida tehdä muutoksia. Kuvassa 14 käy ilmi funktion nimi kohdassa exports.Dht ja sille on määritelty, että se ajetaan, kun sen event toteutuu. Event tässä tapauksessa on uuden viestin saapuminen Pub/Sub topiciin. Tämä viesti sijoitetaan koodissa muuttujaan, muuttuja muutetaan string muotoon ja tulostetaan lopulta Cloud Platformin lokiin.

```

1  /**
2   * Background Cloud Function to be triggered by Pub/Sub
3   *
4   * @param {object} event The Cloud Functions event.
5   * @param {function} callback The callback function.
6   */
7  exports.Dht = (event, callback) => {
8    const pubsubMessage = event.data;
9    const msg = pubsubMessage.data ? Buffer.from(pubsubMessage.data) : null;
10
11   console.log(`Topic viesti, ${msg}`);
12   callback();
13 };
14
15

```

Kuva 14. Esimerkkikoodi, jolla saadaan haettua viesti Pub/Sub topicista.

Dht skripti voidaan ajaa lähettämällä sille gcloud trigger komento komentorivillä. trigger-resource kohtaan määritellään Pub/Sub topic mistä viesti käydään noutamassa ja trigger-event on tapahtumatyyppi mikä pitää toteutua, että skripti ajetaan, joka tässä tapauksessa on uuden viestin saapuminen Pub/Sub topiciin.

Skriptin ajaminen.

```
gcloud beta functions deploy Dht --trigger-resource rasp --trigger-event google.pubsub.topic.publish
```

Kuvassa 15 on nähtävillä onnistunut topicista noudettujen tietojen tulostus Cloud Platformin lokiin.

```

▼ 2018-03-02 16:35:01.886 EET Dht 45676105107843 Topic viesti:, RaspIot/Rasp3 Time 2018-03-02 14:35:01 te
  {
    insertId: "000000-aed20e55-9f66-443e-b36a-42113c196972"
    labels: {...}
    logName: "projects/shining-domain-194413/logs/cloudfunctions.googleapis.com%2Fcloud-functions"
    receiveTimestamp: "2018-03-02T14:35:08.077836777Z"
    resource: {...}
    severity: "INFO"
    textPayload: "Topic viesti:, RaspIot/Rasp3 Time 2018-03-02 14:35:01 temperature 22.0C & humidity 34.0%"
    timestamp: "2018-03-02T14:35:01.886Z"
  }

```

Kuva 15. Topicin viesti luettavissa Cloud Platformin lokitiedoissa.

8 YHTEENVETO

Tässä opinnäytetyössä esiteltiin Google Cloud Platformia yleisellä tasolla ja siihen kuuluvia palveluita. Lähemmin näistä palveluista tutustuttiin IoT Coreen, jota hyödyntäen opinnäytetyön käytännönsuudessa rakennettiin demo. Työn tilaajana oli Hämeen ammattikorkea koulun Älykkäät palvelututkimusyksikkö, joka halusi perustavanlaatuisen tietopakettin Googlen Cloud Platformista ja siitä, kuinka yhdistää IoT-laite palvelun IoT Coreen. Mielestäni onnistuin tuomaan halutut asiat hyvin esille ja pystyin myös rakentamaan demoa pidemmälle onnistuneen yhteyden luomisen jälkeen. Koen että tämä opinnäytetyö toimii hyvänä pohjatietona Älykkäille palveluille, jos he itse ottavat työssä käytetyt palvelut käyttöönsä.

Kaikki tässä opinnäytetyössä käytyt asiat olivat itselleni ennaltaan tuntemattomia. IoT käsitteestä ja Raspberry Pi -tietokoneista olin lukenut muutamien artikkelien, mutta minkäänlaista käytännön kokemusta minulla ei ollut, ja Google Cloud Platformista en ollut edes kuullut ennen opinnäytetyön aloittamista. Tämän myötä työssä käsitellyt aiheet olivat kiinnostavia, mutta välillä haastavia varsinkin, kun Googlen IoT Core on vasta beta-vaiheessa, minkä takia välillä tullessiin virheisiin ei aina löytynyt apua. Teoriaosuudessa käytiin läpi kaikki tarvittava, jotta käytännönsuudessa rakennettu demo voitaisiin toteuttaa.

Demossa luotiin Raspberryn Pin ja IoT Coren välille yhteys, jonka avulla Raspberry Pin mittaamia arvoja lähetettiin eteenpäin muille Cloud Platformin palveluille. Demoa rakentaessa oli ongelmia Raspberryn Pin vakauden kanssa, jotka ratkesivat kuitenkin uuden virtalähteen myötä. Demon rakentaminen auttoi minua erittäin paljon ymmärtämään, kuinka IoT Core toimii ja kuinka sen avulla voidaan jatkokäsitellä IoT-laitteen lähettämää dataa. Demoa rakentaessa opin myös paljon Raspberryn Pistä ja sen Raspbian käyttöjärjestelmästä. Demoa voitaisiin kehittää vielä pidemmälle, jotta mitattu data saataisiin tallennettua järkevämpään paikkaan esimerkiksi tietokantaan.

Sain tätä opinnäytetyötä tehdessäni uuden käsityksen IoT-laitteista ja kuinka sitä voidaan tulevaisuudessa hyödyntää. Koen myös rakennetun demon aikana opitut asiat ja tehdyt virheet arvokkaana tietona tulevaisuuttani ajatellen.

LÄHTEET

Adafruit (2017). *Adafruit Python DHT Sensor Library*. Haettu 14.02.2018
https://github.com/adafruit/Adafruit_Python_DHT

Amazon AWS (n.d.). *Types of Cloud Computing*. Haettu 23.01.2018
<https://aws.amazon.com/types-of-cloud-computing/>

ARM Software (2017). *CPUTemp Example*. Haettu 15.02.2018
<https://github.com/ARM-software/Cloud-IoT-Core-Kit-Examples/tree/master/CPUTemp>

Eskonen, H (2015). *"Esineiden internet mullistaa maailmaa yhtä paljon kuin sähkön ja tietokoneiden tulo."* – *Ensin pitää ratkaista tyhjenevien akkujen ongelma*. *Yle uutiset* 9/2015. Haettu 16.01.2018 osoitteesta
<https://yle.fi/uutiset/3-8302027>

Google (2018). *Google Cloud IoT Core API Python Samples*. Haettu 15.02.2018
https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/iot/api-client/mqtt_example

Google (n.d.). *Cloud Pub/Sub Tutorial*. Haettu 28.02.2018
<https://cloud.google.com/functions/docs/tutorials/pubsub>

Google (n.d.). *Using the MQTT Bridge*. Haettu 15.02.2018
<https://cloud.google.com/iot/docs/how-tos/mqtt-bridge#iot-core-mqtt-auth-run-python>

Google Cloud Platform (n.d.). *Device Security*. Haettu 02.03.2018
<https://cloud.google.com/iot/docs/concepts/device-security>

Google Cloud Platform (n.d.). *Frequently Asked Questions*. Haettu 06.02.2018
<https://cloud.google.com/free/docs/frequently-asked-questions>

Google Cloud Platform (n.d.). *Google Cloud Functions*. Haettu 24.01.2018
<https://cloud.google.com/functions/docs/concepts/overview>

Google Cloud Platform (n.d.). *Google Cloud IoT Core*. Haettu 18.01.2018
<https://cloud.google.com/iot-core/>

Google Cloud Platform (n.d.). *Google Cloud Pub/Sub*. Haettu 24.01.2018
<https://cloud.google.com/pubsub/docs/overview#endpoints>

Google Cloud Platform (n.d.). *Google Cloud SQL*. Haettu 18.01.2018
<https://cloud.google.com/sql/docs/>

Google Cloud Platform (n.d.). *Kotisivu*. Haettu 18.01.2018
<https://cloud.google.com/>

Google Cloud Platform (n.d.). *Overview*. Haettu 23.01.2018
<https://cloud.google.com/docs/overview/>

Lawrence, M. (2017). *The Internet of Things for Dummies*, Qorvo. Haettu 16.01.2017 osoitteesta
http://media.wiley.com/assets/7348/78/Vol_1_9781119349891_Internet_of_Things_FD_QorvoSE.pdf

Mitchell, B (2017). "Hypertext Transfer Protocol Explained" *Lifewire* 11/2017. Haettu 31.01.2018
<https://www.lifewire.com/hypertext-transfer-protocol-817944>

Porter, M & Heppelmann, J. (2014). *How Smart, Connected Products Are Transforming Competition*. *Harvard Business Review* 11/2014. Haettu 16.01.2018 osoitteesta
<https://hbr.org/2014/11/how-smart-connected-products-are-transforming-competition>

Raspbian (n.d.). *Kotisivu*. Haettu 17.01.2018 osoitteesta
<https://www.raspbian.org/>

Raspberry Pi Foundation (n.d.). *GPIO: Raspberry Pi Models A and B*. Haettu 17.01.2018
<https://www.raspberrypi.org/documentation/usage/gpio/>

Raspberry Pi Foundation (n.d.). *Kotisivu*. Haettu 17.01.2018 osoitteesta
<https://www.raspberrypi.org/help/faqs/#introWhatIs>

Rouse, M (2015). "MQTT (MQ Telemetry Transport)" *TechTarget IoT Agenda* 12/2015. Haettu 31.01.2018
<http://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport>