

IBM Watson – T0 – MKR1000 – Getting started

Käyttöönotto-ohje

1. IoT –alusta ja mikro-ohjain

IoT –alusta, IoT Platform, on kehitysympäristö, jossa kehitetään järjestelmäylläpitäjälle ja loppukäyttäjälle suunnattuja toimintoja IoT-järjestelmään. Toimintoja voivat olla esim. IoT-laitekannan hallinta tai hälytysten saanti analysoidun datan perusteella. Tässä esimerkissä IoT –alusta on IBM Cloud Watson IoT. Palvelun markkinointinimi oli aiemmin IBM Bluemix Watson IoT.

<https://developer.ibm.com/iotplatform/>

Anturidatan keräämiseen käytämme Arduino MKR1000 –korttia.

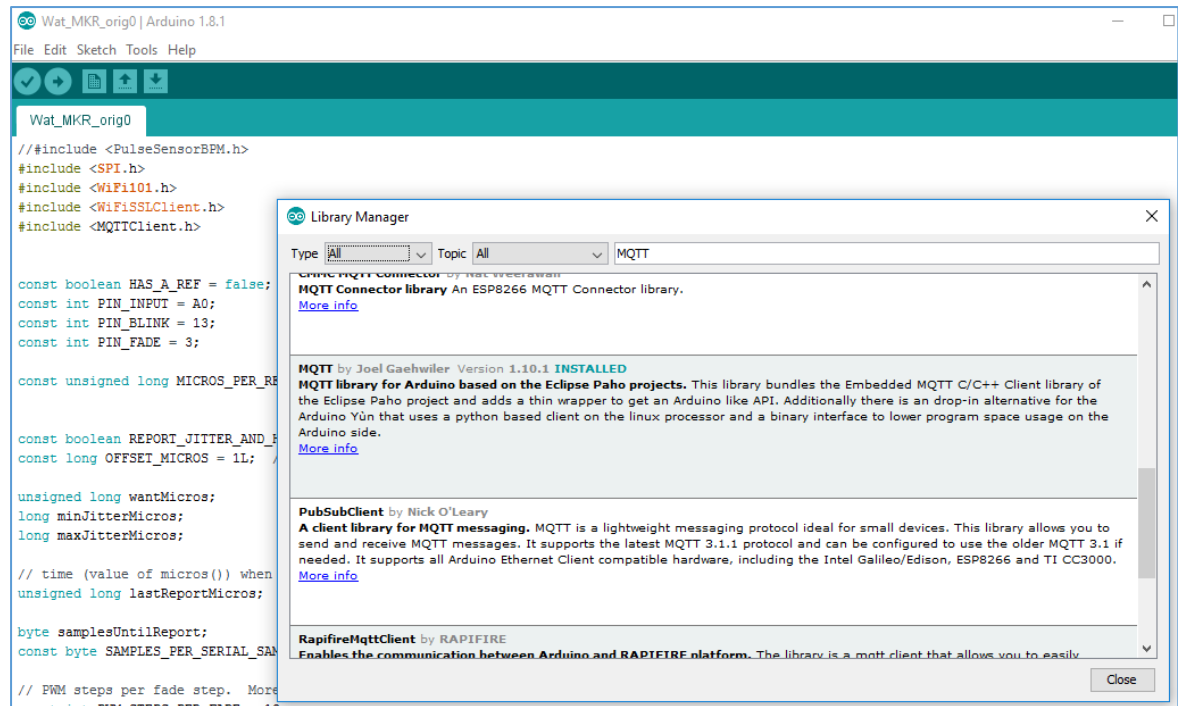
<https://www.arduino.cc/en/Main/Products>

Saat käsityksen IoT -alustan toiminnasta seuraamalla tätä esimerkkiä.

Jos suunnitelmasi on aloittaa työskentely IoT-alustan kanssa, on parasta tehdä jo ensimmäistä kertaa lukiessa kaikki vastaavat toimenpiteet kuin tässä esimerkissä.

2. MKR1000 -koodi

IBM Watson käyttää standardin mukaista MQTT –protokollaa. Arduino-korteille on tätä protokollaan varten useita luokkakirjastoja. Avaa Arduino IDE, alusvetovalikko Sketch, Include Library, Manage Libraries. Etsi hakusanalla MQTT.



Kuva 2.1 MQTT-kirjasto Joel Gähwiler.

Lataa kirjasto, jonka tekijä on Joel Gähwiler.

Tämän koodin lähtökohtana on käytetty eri esimerkkejä IBM Bluemix, Watson IoT verkkosivuilta.

Ahkeran testaamisen jälkeen koodi saadaan alla näkyvään muotoon.

```
/*
MKR1000 connecting to IBM Watson IoT Platform

Based on documentation and "recipes" on IBM Bluemix
https://www.ibm.com/cloud-computing/bluemix/watson
Timo Karppinen 19.2.2017

Modified for testing SPI microphone board Digilent PmodMIC3
Please connect
MKR1000 - PmodMIC3
GND - 5 GND
Vcc - 6 Vcc
9 SCK - 4 SCK
10 MISO - 3 MISO
1 - 1 SS

a sound indicator LED
6 - 220 ohm - LED or the onboard LED
Timo Karppinen 13.9.2017

Added calculated sensor data
Loop timing redone
14.11.2017
*/

#include <SPI.h>
#include <WiFi101.h>
#include <WiFiSSLClient.h>
#include <MQTTClient.h> // The Gähwiler mqtt library

// WLAN
```

```

char ssid[] = "Moto_Z2_TK"; // your network SSID (name)
char pass[] = "xxxxxxxxxxxxxx"; // your network password (use for WPA)

//char ssid[] = "HAMKWlan"; // your network SSID (name)
//char pass[] = "xxxxxxxxxxxxxxxxxxxx"; // your network password (use for WPA)

// IBM Watson
// Your organization and device needs to be registered in IBM Watson IoT Platform.
// Instruction for registering on page
// https://internetofthings.ibmcloud.com/#

//char *client_id = "d:<your Organization ID>:<your Device Type>:<your Device ID>";
char *client_id = "d:v8yyyyy:A_MKR1000:DF48";
char *user_id = "use-token-auth"; // telling that authentication will be done with token
char *authToken = "xxxxxxxxxxx"; // Your IBM Watson Authentication Token

//char *ibm_hostname = "<your-org-id.messaging.internetofthings.ibmcloud.com>";
char *ibm_hostname = "v8yyyyy.messaging.internetofthings.ibmcloud.com";

// sensors and LEDs
const int ainputPin = A0;
const int soundLEDPin = 6; // must be a pin that supports PWM. 0...8 on MKR1000
// PModMIC3
const int mic3CS = 1; // chip select for MIC3 SPI communication
int sound12bit = 0; // 12 bit sound level value [ 0000 nnnn nnnn nnnn ] nnn.. = two's complement!
int soundByte1 = 0; // 8 bit data from mic board
int soundByte2 = 0; // 8 bit data from mic board
int sound32bit = 0; // in MKR1000 board SAMD21 processor the int is 32 bit two's complement
int sound8bit = 0;

const int numSamples = 100;
int sound8bitA[numSamples];
int sampleIndex = 0;
int soundSum = 0;

int soundLevel = 0; // 8 bit positive number from 0 to 255
int blinkState = 0;

/*use this class if you connect using SSL
 * WiFiSSLClient net;
 */
WiFiClient net;
MQTTClient MQTTC;

unsigned long lastSampleMillis = 0;
unsigned long previousWiFiBeginMillis = 0;
unsigned long lastWatsonMillis = 0;
unsigned long lastPrintMillis = 0;

void setup()
{
    pinMode(mic3CS, OUTPUT);
    digitalWrite(mic3CS, HIGH); // for not communicating with MIC3 at the moment
    Serial.begin(9600);
    delay(2000); // Wait for wifi unit to power up
    WiFi.begin(ssid, pass);
    delay(5000); // Wait for WiFi to connect
    Serial.println("Connected to WLAN");
    printWiFiStatus();

    /*
    client.begin("<Address Watson IOT>", 1883, net);
    Address Watson IOT: <WatsonIOTOrganizationID>.messaging.internetofthings.ibmcloud.com
    Example:
    client.begin("iqwckl.messaging.internetofthings.ibmcloud.com", 1883, net);
    */
    MQTTC.begin(ibm_hostname, 1883, net); // Cut for testing without Watson

    connect();

    SPI.begin();
    // Set up the I/O pins

    pinMode(mic3CS, OUTPUT);
    pinMode(soundLEDPin, OUTPUT);

    // Initializing the sound sample array to zero.
    for(int i = 0; i < numSamples; i++)
    {
        sound8bitA[i] = 0;
    }
}

void loop() {
    MQTTC.loop(); // Cut for testing without Watson

```

```

// opening and closing SPI communication for reading MIC3
if(millis() - lastSampleMillis > 1000/numSamples)
{
    lastSampleMillis = millis();
    SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE0));
    digitalWrite(mic3CS, LOW);

    soundByte1 = SPI.transfer(0x00);
    soundByte2 = SPI.transfer(0x00);

    digitalWrite(mic3CS, HIGH);
    SPI.endTransaction();

    soundByte1 = soundByte1 << 8;
    sound12bit = soundByte1 | soundByte2;
    sound32bit = sound12bit << 22; // 22 bits to the left to create 32 bit two's complement
    sound8bit = sound32bit / 16777216; // 2 exp24 = 16 777 216 means shifting 24 bits left without
    shifting the sign!

    soundSum = soundSum - sound8bitA[sampleIndex]; // subtract the oldest sample
    sound8bitA[sampleIndex] = sqrt(sound8bit * sound8bit); // reading the | latest sample |
    soundSum = soundSum + sound8bitA[sampleIndex]; //add the latest sample
    analogWrite(soundLEDPin, sound8bitA[sampleIndex]); // blink the LED with intensity = | sound sample |
    sampleIndex = sampleIndex + 1;
    if(sampleIndex >= numSamples)
    {
        sampleIndex = 0;
    }

    soundLevel = soundSum / numSamples;
}

// Print on serial monitor once in 1000 millisecond
if(millis() - lastPrintMillis > 1000)
{
    Serial.print("Sound32bit ");
    Serial.print(sound32bit);
    Serial.print(" Sound8bit ");
    Serial.print(sound8bit);
    Serial.print(" SoundLevel ");
    Serial.println(soundLevel);
    lastPrintMillis = millis();
}

// publish a message every 30 second.
if(millis() - lastWatsonMillis > 30000)
{
    Serial.println("Publishing to Watson...");
    if(!MQTTc.connected()) { // Cut for testing without Watson
        connect(); // Cut for testing without Watson
    } // Cut for testing without Watson
    lastWatsonMillis = millis();
    //Cut for testing without Watson
    MQTTc.publish("iot-2/evt/SoundTwo/fmt/json", "{\"Sound level sensors\":\"Sounds from field, too
\", \"SoundMean\": \"\" + String(soundLevel) + \"\", \"SoundStreight\": \"\" + String(sound8bit) + \"\"}");

}

delay(1);

// end of loop
}

void connect()
{
    Serial.print("checking WLAN...");
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print("."); // printing a dot every half second
        if ( millis() - previousWiFiBeginMillis > 5000) // reconnecting
        {
            previousWiFiBeginMillis = millis();
            WiFi.begin(ssid, pass);
            delay(5000); // Wait for WiFi to connect
            Serial.println("Connected to WLAN");
            printWiFiStatus();
        }
        delay(500);
    }
}
/*
Example:
MQTTc.connect("d:igwckl:arduino:oxigenarbpn","use-token-auth","90wT2?a*1WAMVJStb1")

```

```
Documentation:
https://console.ng.bluemix.net/docs/services/IoT/iotplatform_task.html#iotplatform_task
*/

Serial.print("\nconnecting Watson with MQTT...");
// Cut for testing without Watson
while (!MQTTc.connect(client_id,user_id,authToken))
{
    Serial.print(".");
    delay(3000);
}
Serial.println("\nconnected!");
}

void messageReceived(String topic, String payload, char * bytes, unsigned int length) {
    Serial.print("incoming: ");
    Serial.print(topic);
    Serial.print(" - ");
    Serial.print(payload);
    Serial.println();
}

void printWiFiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print your WiFi shield's IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");
}
```

Koodiesimerkki 2.1 Sovellus ”A MKR1000 sound”. Koodin versio 4. Laitteet osaavat mm. liittyä uudelleen wlan-verkkoon, jos laitteet joutuivat verkon katveeseen ja palasivat uudelleen verkon kuuluvuusalueelle. Jatkuvasti ajettavan silmukan ajoitus on kirjoitettu uudelleen.

Ohjelmakoodin esimerkki toimii vain, jos organisaatio ja laite on rekisteröity IBM Cloud Watson IoT -platform:iin. Huomaa, jos kopioit koodin pdf-tiedostosta, saat mitä todennäköisimmin mukana erikoismerkkejä. Koodisi ei välttämättä toimi.

Koodiesimerkissä käytetään salaamatonta yhteyttä. Jotta tämä toimisi on Bluemix IoT-alustassa valittava ”Security”, ”Connection Security”, ”TSL Optional” .

Riippumatta siitä, mitä IoT-alustaa tai mqtt-brokeria käytetään, on jokaiselle alustaan tai brokeriin yhteydessä olevalle laitteelle luotava oma tunnus client ID ja salasana authentication token. Jos samoilla tunnuksilla yritetään kirjautua samanaikaisesti useasta laitteesta, estää IoT-alusta tai mqtt-broker yhteyden.

3. IBM Cloud

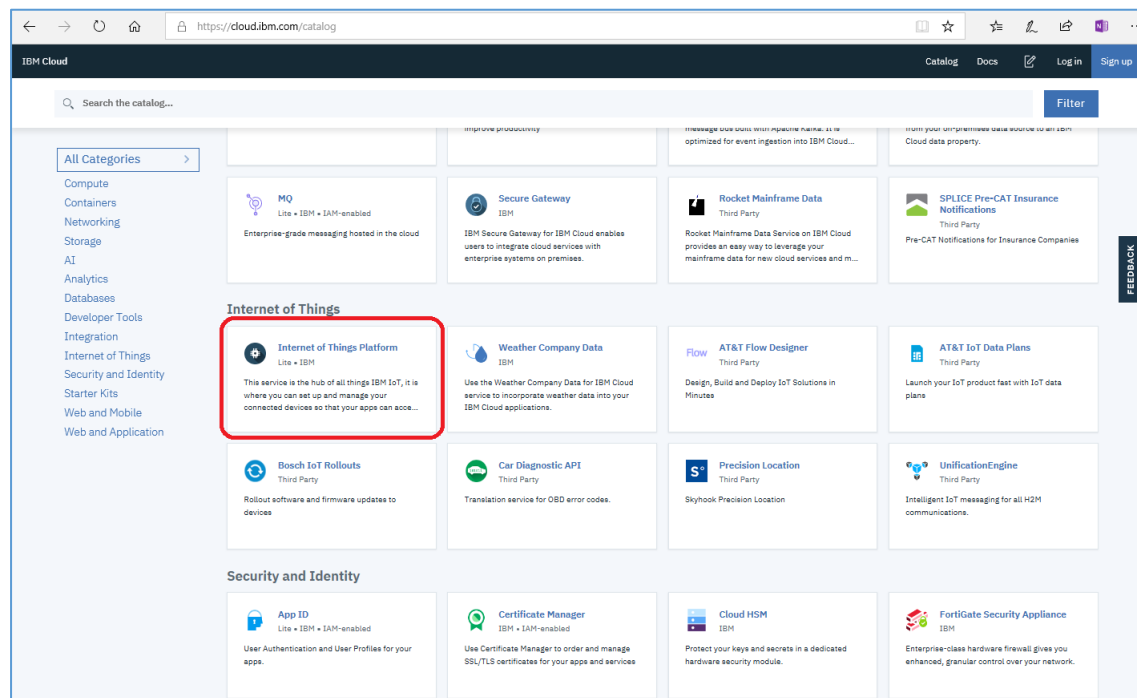
Selaimessa hae hakusanalla ”IBM Bluemix Catalog”. Päädyt todennäköisesti sivulle

<https://console.ng.bluemix.net/catalog/>

Valikosta vasemmalla valitse ”Internet of Things”.

Voit myös hakea hakusanalla ”IBM Cloud Catalog”. Tuotenimi on uudistettu ja aloitussivuna toimivat sekä ”cloud”- että ”bluemix”-nimeä käyttävät sivut. Päädyt todennäköisesti sivulle

<https://cloud.ibm.com/catalog>

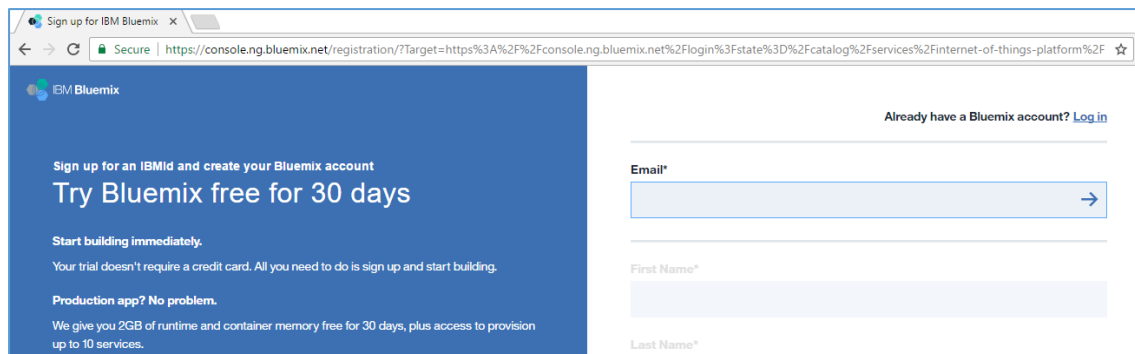


kuva 3.1 IBM Cloud, Internet of Things, Internet of Things Platform / <https://cloud.ibm.com/catalog> 3.6. 2019

Valitse maksuohjelmaksi ilmainen vaihtoehto

Pricing Plans			Monthly prices shown are for country or region: United States
PLAN	FEATURES	PRICING	
✓ Lite	Includes up to 20 registered devices, and a maximum of 100 MB of each data metric Maximum of 20 registered devices Maximum of 10 application bindings Maximum of 100 MB of each of data exchanged, data analyzed and edge data analyzed	Free	
The Lite service plan for Internet of Things Platform includes up to 20 registered devices, and a maximum of 100 MB each of data exchanged, data analyzed, and edge data analyzed per month.			
Standard	The Standard service plan for Internet of Things Platform includes your free tier of 100 MB each of data exchanged, data analyzed and edge data analyzed per month at no cost. Charge per MB of data exchanged (tiered by usage in MB) Charge per MB of data analyzed Charge per MB of edge data analyzed Multi-Tiered	Expand each section to view details	

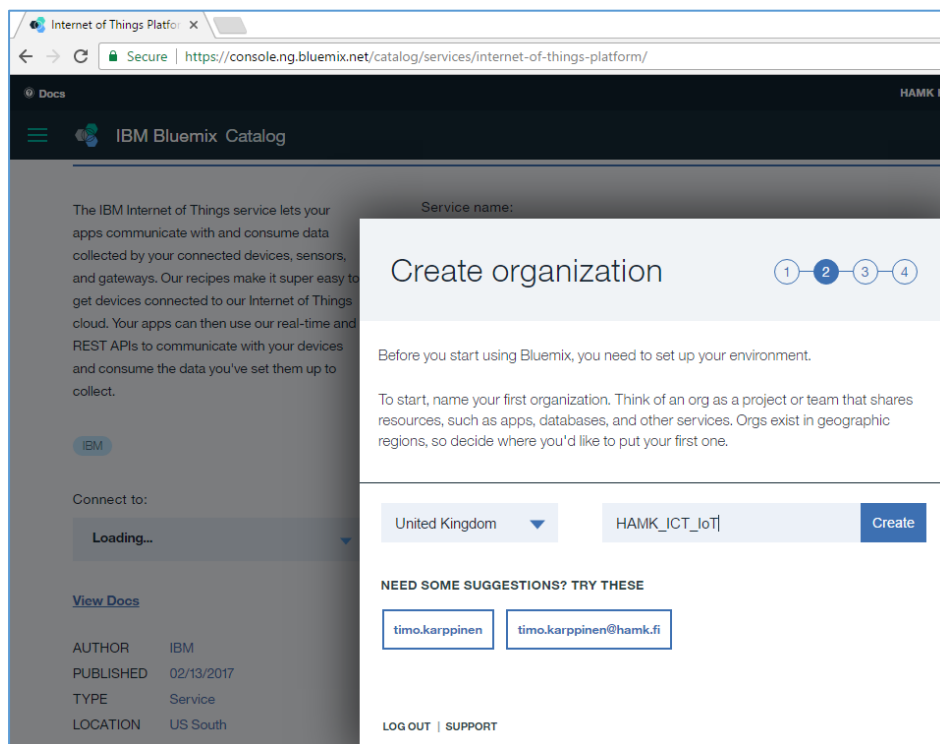
kuva 3.2. Ilmainen käyttösojimus. Tarjouksen ilmainen resurssimäärä kasvaa vuosittain / 2017



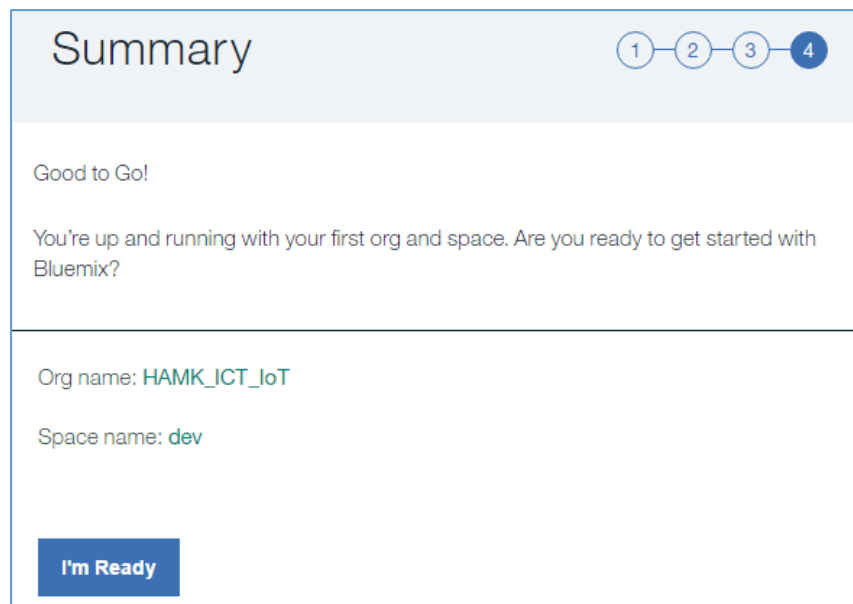
kuva 3.3. Ilmainen kokeilusopimus on voimassa 30 vrk. / 2017

Luo tili ja seuraa kirjautumisohjeita. Huomaa, että luot nyt kehittäjän IBM ID:n. Tämä on eri tili kuin IBM Documentation -alueella luotu tili! Voit myöhemmin – kylläkin vain tämän 30 vrk vielä ollessa voimassa – muuttaa tämän tilin pidempään voimassa olevaksi akateemiseksi opiskelijan tai opettajan tiliksi. Tästä saat tarvittaessa erillisen ohjeen.

Huomaa, että opiskelijan tai opettajan tili on uusittava tilin viimeisen voimassaolokuukauden aikana. Näin saat uuden jakson käyttöösi. Kirjautumissivujen kuvat yllä ovat vuodelta 2017 mutta edelleen vuonna 2019 toiminnot olivat samat.



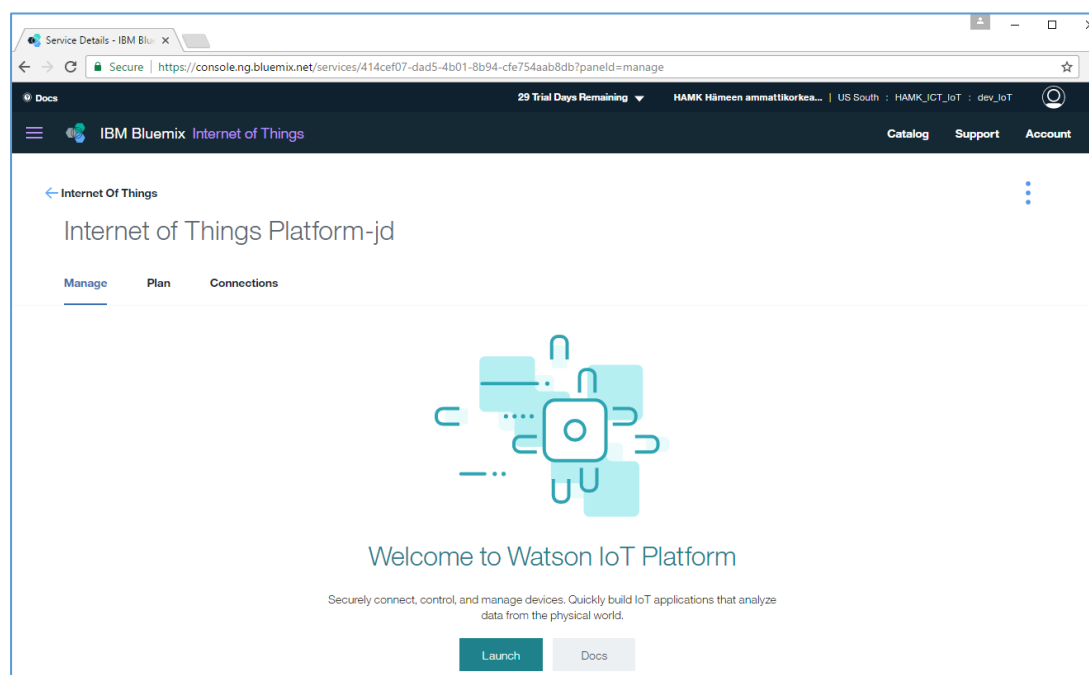
kuva 3.4 Tili sidotaan käyttämään palveluita esim. UK servereiltä.



kuva 3.5 Tilin tiedot

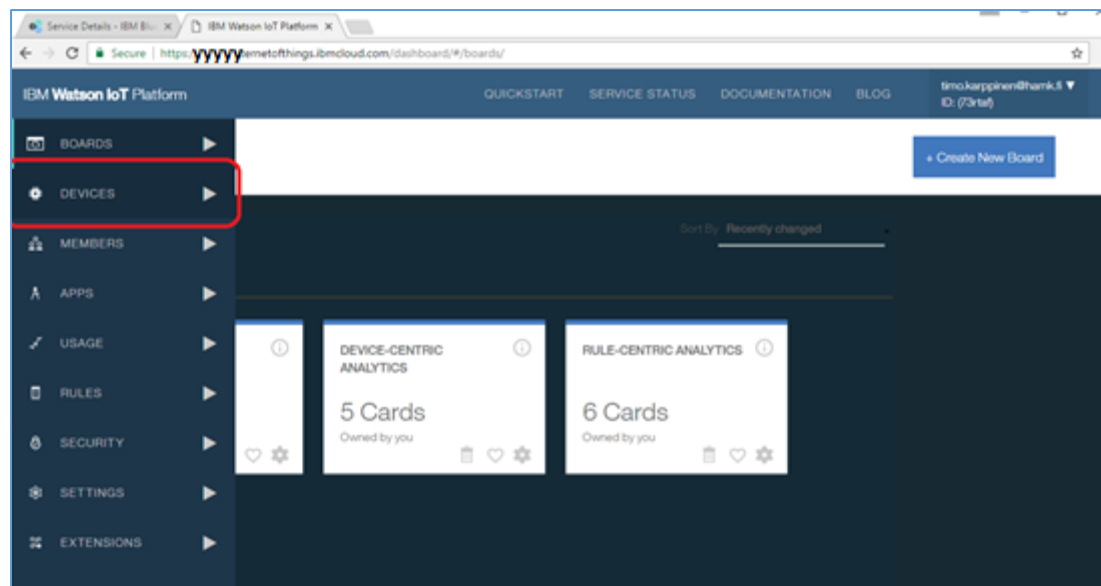
Päätynyt takaisin kuvan 3.2 tilanteeseen. valitse alareunasta sininen painike "Create".

Nyt päädyt uudelle sivulle.



Kuva 3.6 IoT Platform

Valitse "Launch"



Kuva 3.7 IoT platform

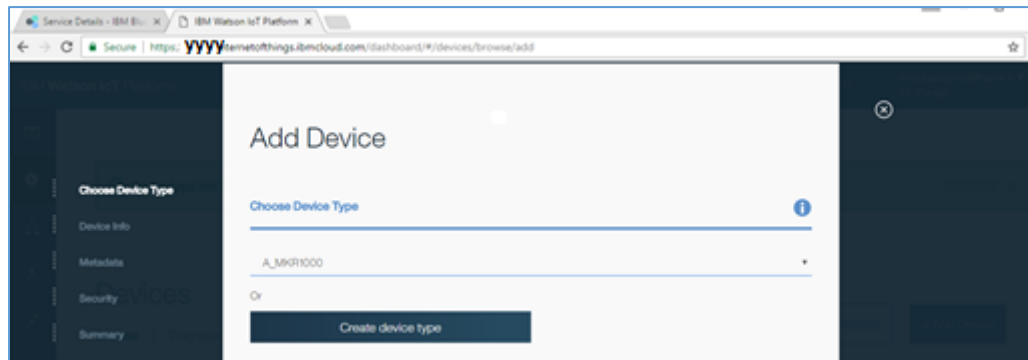
Valitse vasemmalta "Devices" ja edelleen "Add new device"

Täytä laitteelle tunnistetiedot. Etenet aina painikkeella "Next".



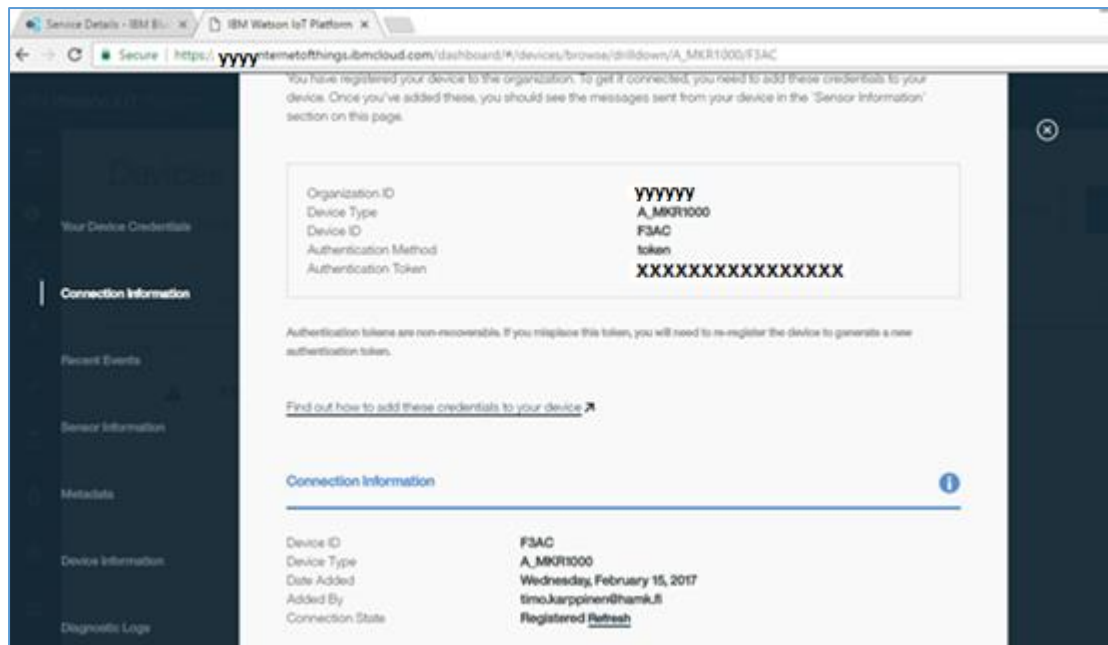
Kuva 3.8 Nimeät laitteellesi laitetyypin Device Type.

Kun olet täyttänyt tiedot laitetyypille, ilmestyy tyyppi valikkoon Choose Device Type.



Kuva 3.9. Äskettäin luotu laite.

Valitse laite ja jatka eteenpäin täyttämällä kysyttyjä tietoja. Anna järjestelmän luoda tunnistautumisen salasana eli Token.

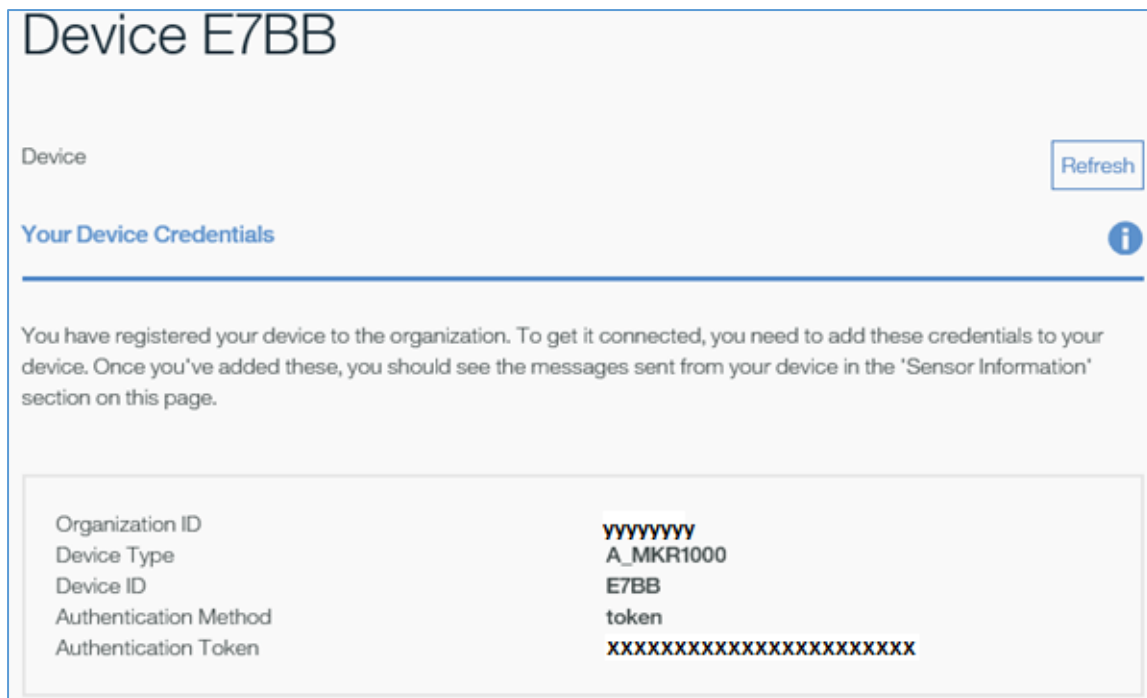


kuva 3.10 Tunnistetiedot.

Kopioi tunnistetiedot tästä näkymästä. Tarvitsemme niitä myöhemmin.

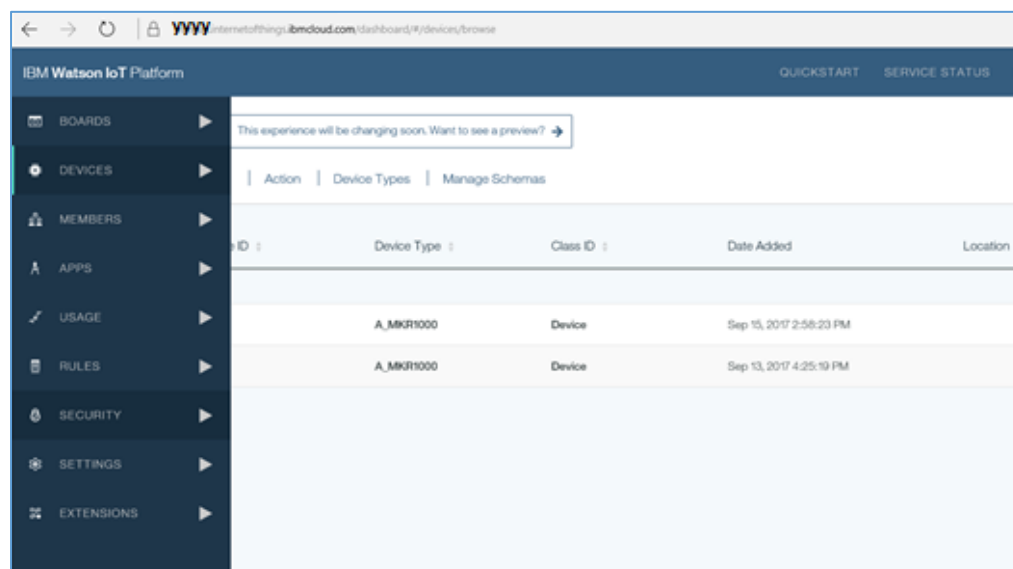
Laitteita voit luoda tarpeen mukaan lisää. Mqtt-client, oli se sitten mikro-ohjain tai esim. Windows:ssa ajettava mqtt client -sovellus, kirjautuu IBM Watson IoT-palveluun laitteena. Jokainen mqtt client tarvitsee oman laitetunnuksen Device ID ja salasanan Authentication Token.

Ilmaisella kokeilutunnuksella ja opiskelijan tunnuksella käytettävissä olevat resurssit IBM Cloud Watson:ssa voivat rajoittaa uusien organisaatioiden määrää. Luot siis uudet laitteet samalle Organization ID:lle.



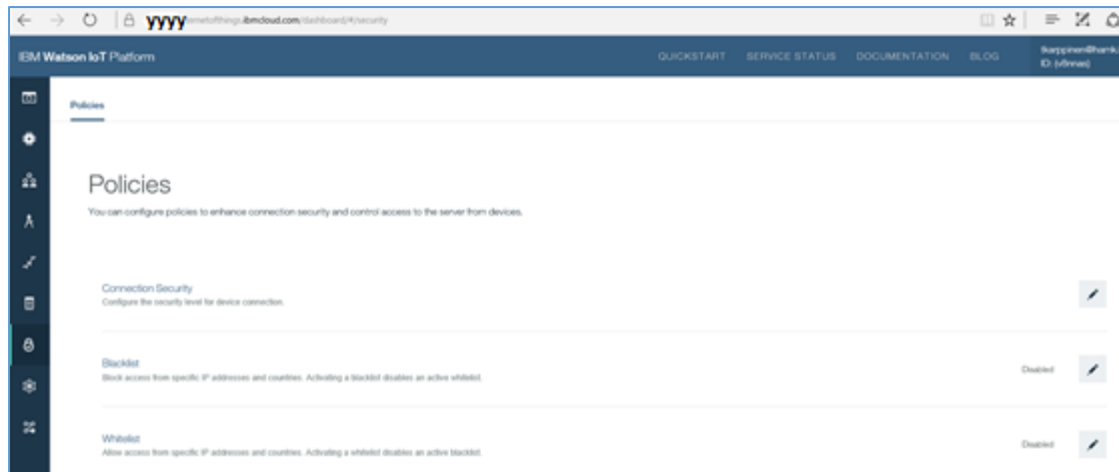
kuva 3.11. Toisen organisaation ja laitteen tunnistetiedot.

Watson IoT:n turvallisuusasetukset määrittävät oletuksena TLS-salatun yhteyden. Yksinkertaisinta on kuitenkin nyt liittyä ilman salausta. IBM Watson IoT Dashboard:ssa on muutettava Security-asetuksia.



kuva 3.12 Turvallisuusasetusten muuttaminen Security-välilehdellä.

Klikkaa vasemmalta valikosta "SECURITY". Pitäisi aueta seuraavanlainen sivu.



Kuva 3.13 Policies, Connection Security

Valitse "Connection Security" klikkaamalla kynän kuvaa.



Kuva 3.14 TLS Optional

Valitse "...CONNECTION SECURITY...", "TLS Optional"

4. Anturitiedot IBM Watson:ssa

Voit nyt liittää mikro-ohjaimella ja antureilla varustetun IoT anturilaitteesi Watson IoT palveluun. Tutki edellä esitettyä koodiesimerkkiä "A MKR1000 sound" ja tee itsellesi vastaava sovellus. Laboratorioharjoitustunneilla saat tehtävään ohjausta.

Kun yhteys lopulta onnistuu, on laitteen lähettämä tieto järjestelmään kirjautuneille käyttäjille nähtävissä sivulla IBM Watson IoT Platform, Dashboard, Devices.

The screenshot displays the IBM Watson IoT Dashboard for a specific device, F3AC. The interface is divided into a left sidebar with navigation options and a main content area. The main content area includes sections for Connection Information, Recent Events, and Sensor Information.

Device F3AC

Device ID: F3AC
Device Type: A_MKR1000
Date Added: Wednesday, February 15, 2017
Added By: timo.karpinen@hamk.fi
Connection State: Connected on Sunday, February 19, 2017 at 7:03:12 PM from 91.145.91.156 with an insecure connection [Refresh](#)

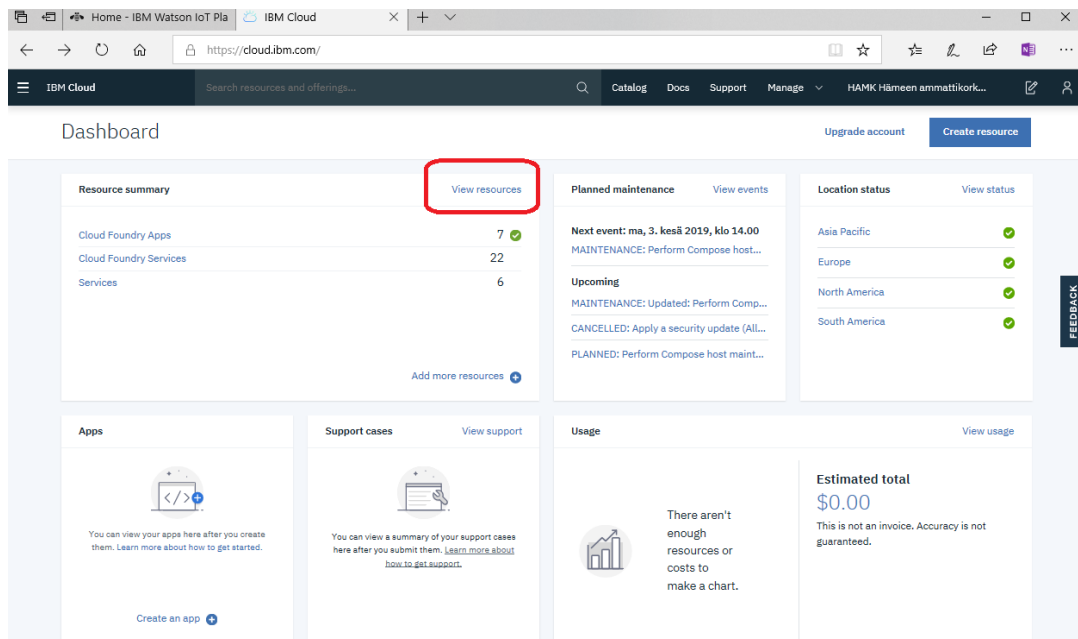
Recent Events

Event	Format	Time Received
tpm	json	Feb 19, 2017 7:07:42 PM
tpm	json	Feb 19, 2017 7:07:52 PM
tpm	json	Feb 19, 2017 7:08:02 PM
tpm	json	Feb 19, 2017 7:08:12 PM
tpm	json	Feb 19, 2017 7:08:22 PM
tpm	json	Feb 19, 2017 7:08:32 PM
tpm	json	Feb 19, 2017 7:08:42 PM
tpm	json	Feb 19, 2017 7:08:52 PM
tpm	json	Feb 19, 2017 7:09:02 PM
tpm	json	Feb 19, 2017 7:09:12 PM

Sensor Information

Event	Datapoint	Value	Time Received
tpm	name	Analog input value	Feb 19, 2017 7:09:12 PM
tpm	tpm	792	Feb 19, 2017 7:09:12 PM

Kuva 4.1 Laitteen lähettämä tieto on näkyvissä IBM Watson IoT, Dashboard, Device -sivulla. / https://yyyyyy.internetofthings.ibmcloud.com/dashboard/#/devices/browse/drilldown/A_MKR1000/F3AC , 2017



Kuva 4.2 Jos Dashboard-näkymää etsiessä päädyit tämän kaltaiselle sivulle, valitse kuvassa punaisella ympyröity View Resources. / <https://cloud.ibm.com/> 3.6.2019

Saatoit Dashboard-näkymää etsiessäsi päätyä näkymään IBM Cloud Dashboard. Tämä tilanne näkyy kuvassa yllä. Valitse

View Resources

Cloud Foundry Services, Internet of Things Platform ...

Launch

Sinun pitäisi päätyä sivulle

<https://yyyyyyyy.internetofthings.ibmcloud.com/dashboard/> jossa yyyyyyyy on sinun edellä luomasi IoT-alustan tunnus.

5. Datan näyttö Dash Board-näkymässä IBM Watsonissa

Datan analysointi on yksi palveluista, joita IoT -alusta, IoT Platform, tarjoaa. Tutustuminen IoT -alustoihin voidaan aloittaa hakemalla selaimella hakusanalla IBM Developer.

Todennäköisesti päädyt IBM Developer -yhteisön pääsivulle. Sivulla

<https://developer.ibm.com/> voit valita Topics, Technologies, IoT. Sivulla

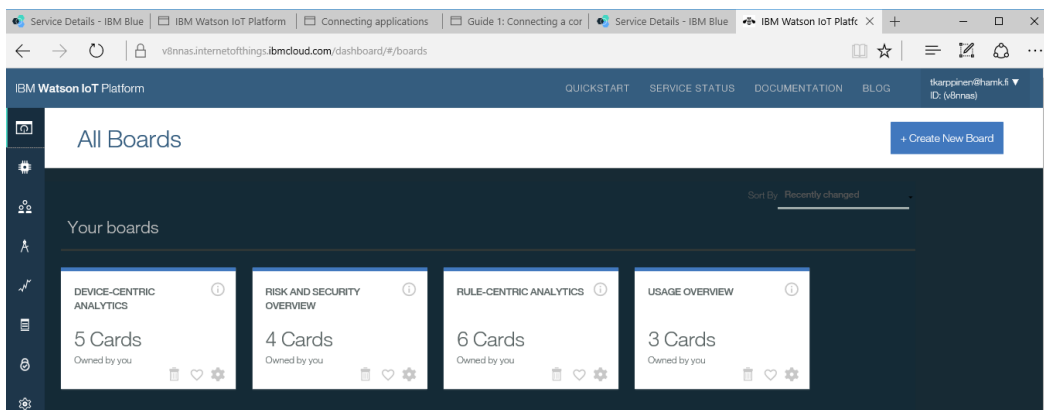
<https://developer.ibm.com/technologies/iot/> on kymmenittäin tekstiartikkeleita, Articles, ja ohjeita, Tutorials.

Johdanto IoT -alustoihin kerrotaan selkeästi artikkelissa "Streamlining the development of your IoT applications by using an IoT platform",

<https://developer.ibm.com/technologies/iot/articles/iot-lp101-why-use-iot-platform> . Lue tämä.

Jatkamme datan analysoinnilla suoraan IoT -alustassa niin sanotussa Dashboard:ssa. Alla olevat kuvat ovat IBM Bluemix:stä siinä muodossa kuin sivut aukesivat 27.9.2017 . Vuonna 2019 toiminnot ovat edelleen samat.

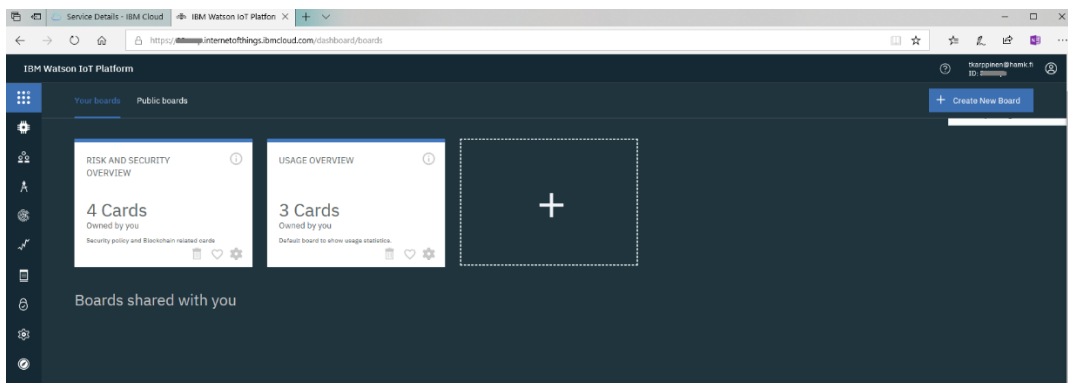
Oletamme, että olet laite-esimerkin ohjeen mukaisesti luonut tunnukset ja kirjautunut IBM Cloud Watson IoT –palveluun. Oletamme myös, että laitteesi lähettää mittausarvoja Watson IoT -alustaan. Valitse IoT Dashboard –näkylässä valikosta vasemmalla Boards.



Kuva 5.1 Dashboard, Boards / 2017

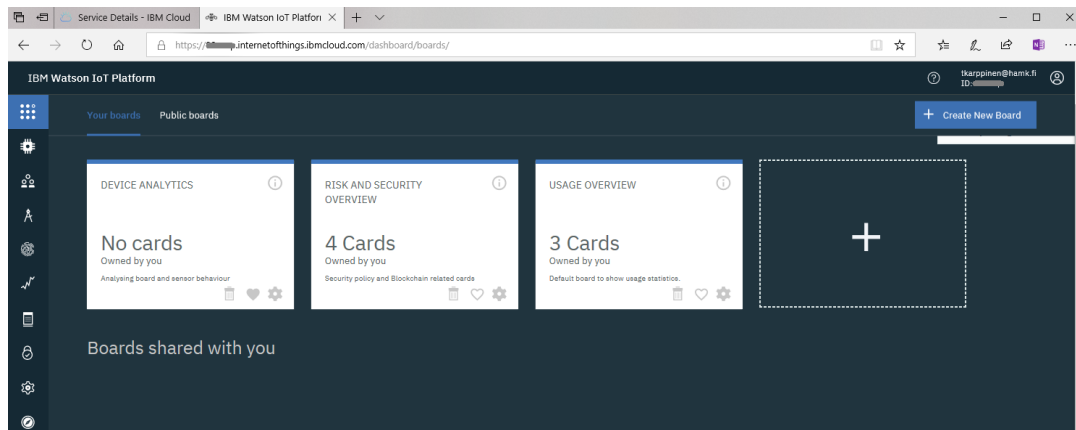
Tästä voit valita Device Centric Analytics.

Jos Device Centric Analytics ei ole valmiina olemassa, voimme itse luoda vastaavan Board – näkymän.



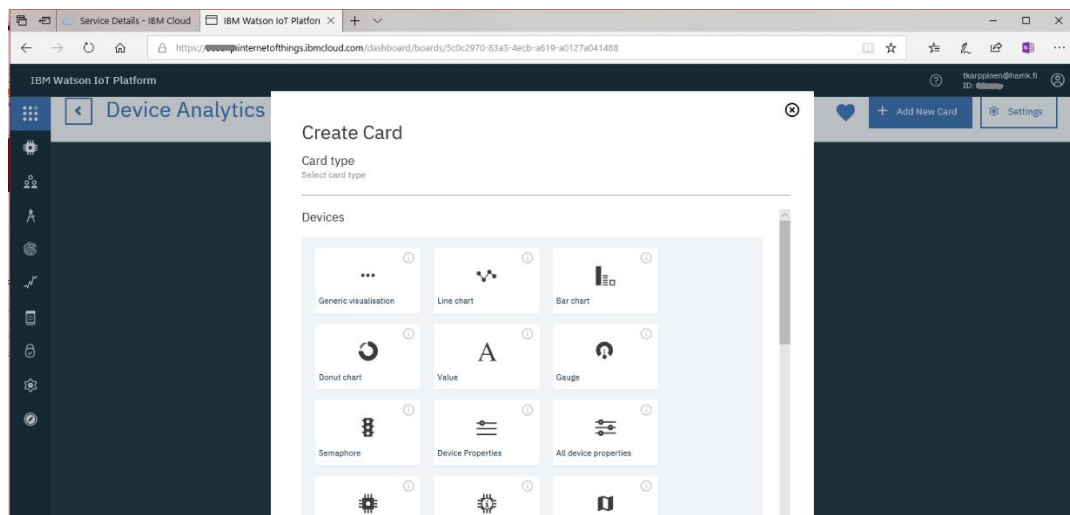
Kuva 5.2 Uusi Board – lisätään +kuvakkeesta / 2019

Lisää tähän näkymään uusi Board-kortti. Anna sille nimeksi vaikka Device Analytics.



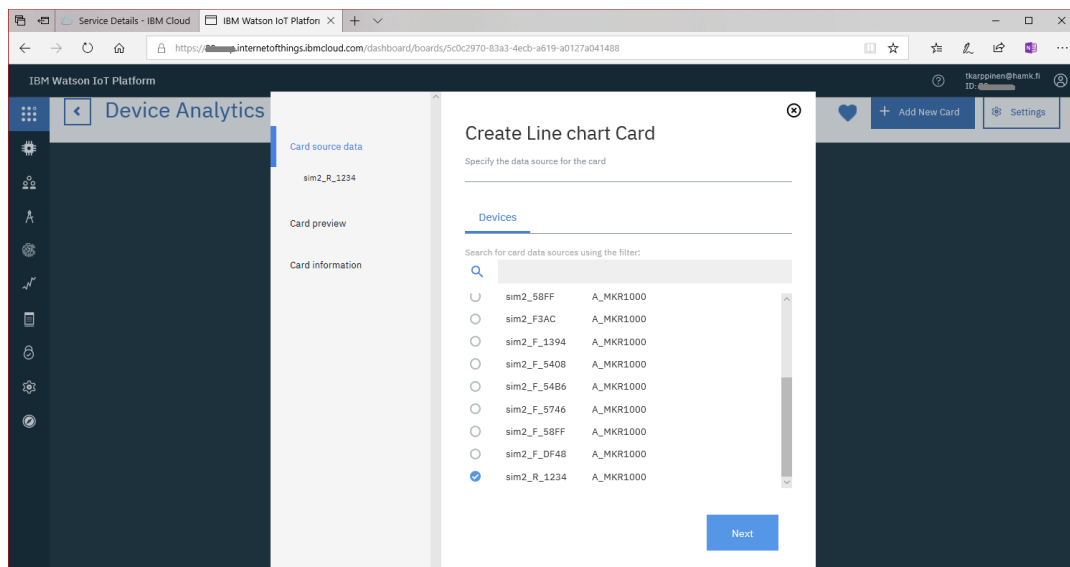
Kuva 5.3 Uusi Board-näkymä on luotu. / 2019

Valitse tämä uusi Board ja edelleen Create Card.



Kuva 5.4 Card -tyyppiä / 2019

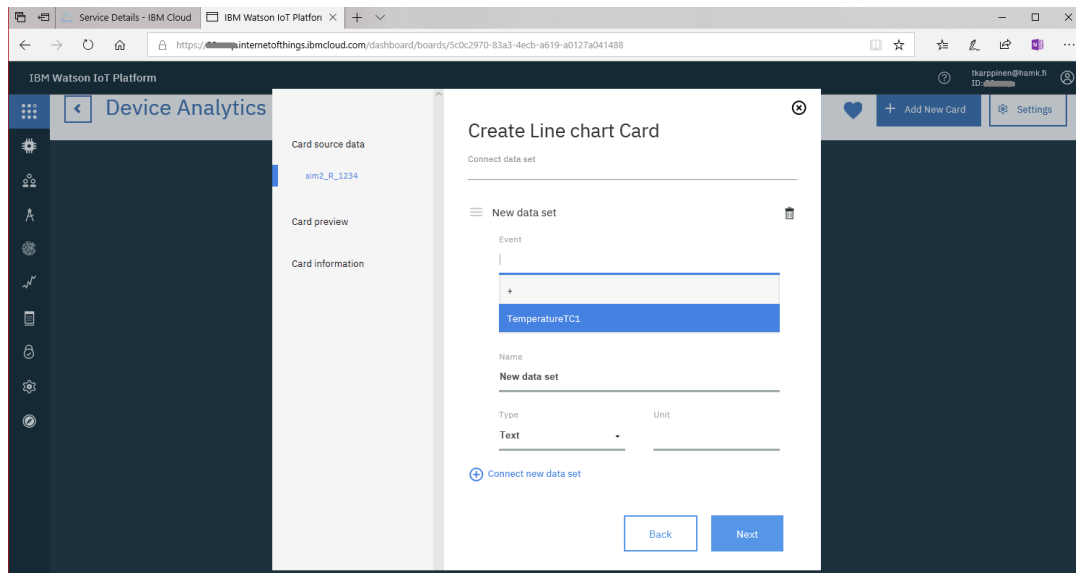
Avautuvasta valikosta valitset Line chart.



Kuva 5.5 Laitteen kirjaaminen Line chart -korttiin. / 2019

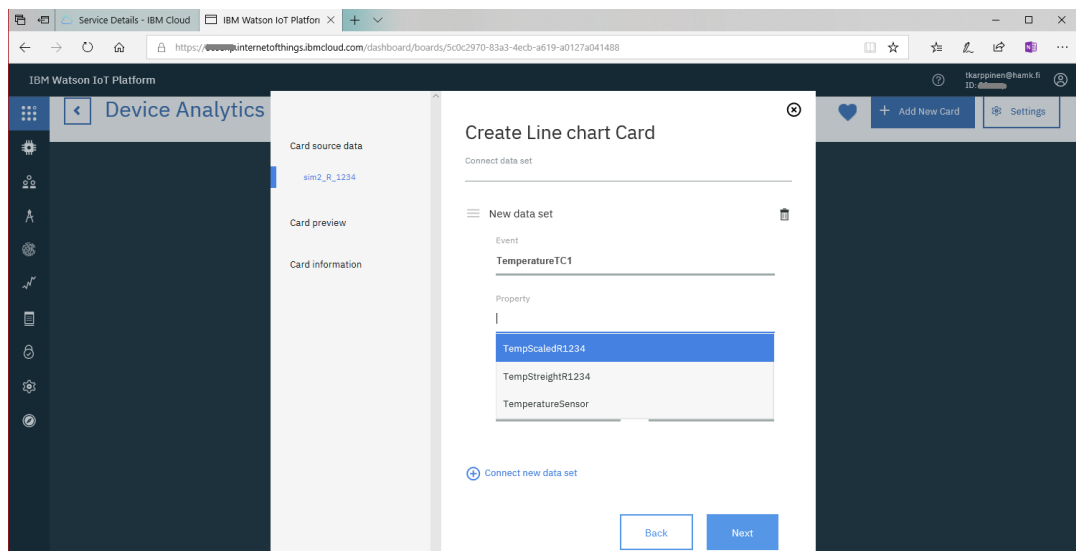
Valitse, minkä laitteesi mittaustietoja haluat tarkastella.

Valitse Connect new dataset. Jos laitteesi lähettää parhaillaan mqtt-protokollan mukaisia laitteesi autentikointitietojen mukaisia viestejä, osaa Watson purkaa sanoman ja ehdottaa sen mukaan oikeita määrittelyjä.



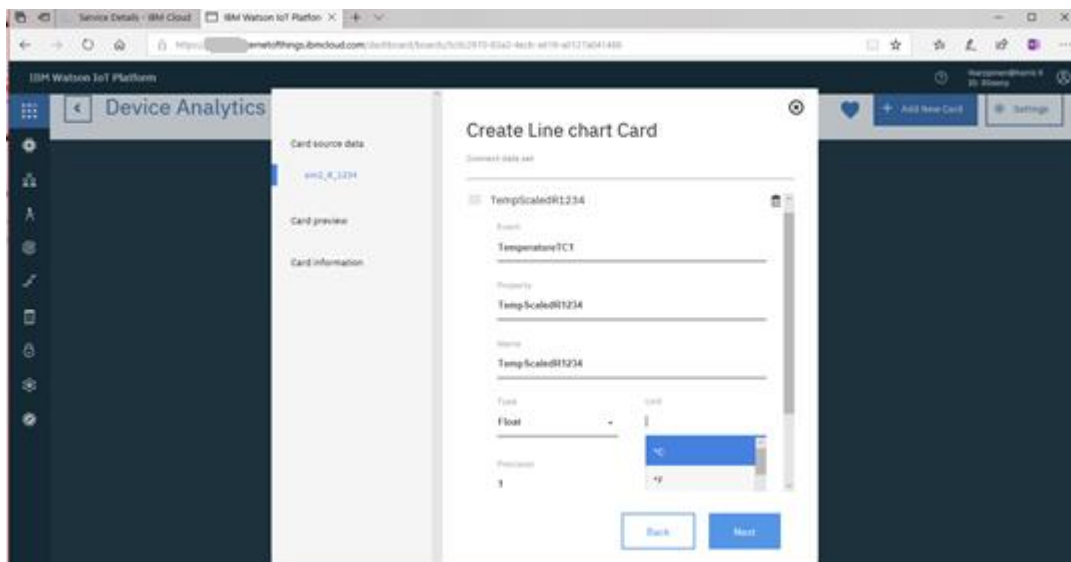
Kuva 5.6 Event-tyyppit saadaan laitteen viimeksi lähettämistä mqtt-viesteistä. / 2019

New Data Set event-tyyppiksi poimitaan itse sanomassa määrittämäsi event-nimi.



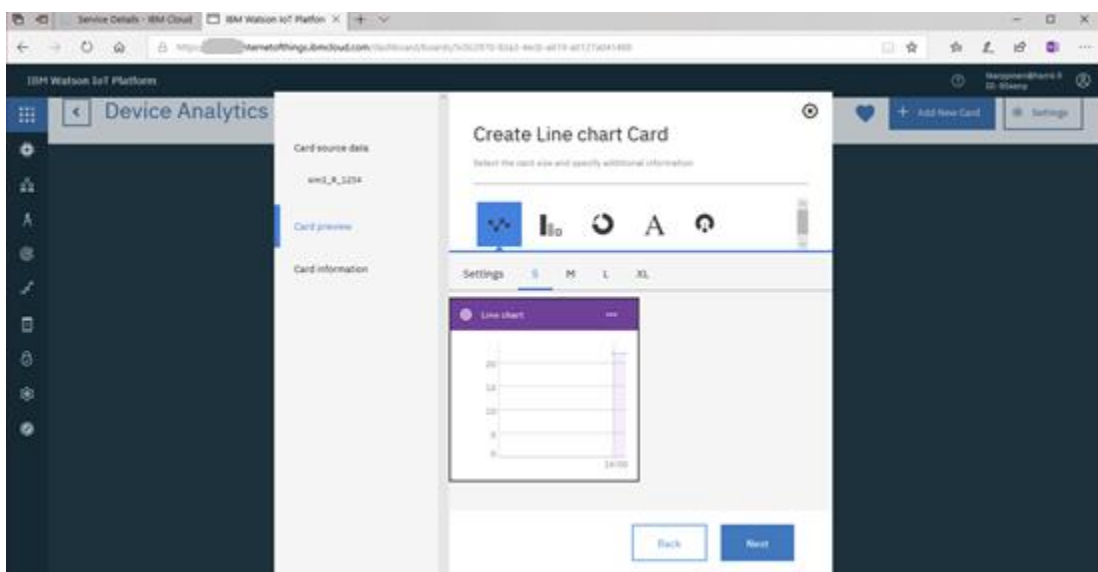
Kuva 5.7 Property-tyyppit saadaan laitteen viimeksi lähettämistä mqtt-viesteistä. / 2019

Valitse Line chart -kaavioon hyvin sopiva numeerinen tieto.



Kuva 5.8 Suureen tyyppi ja yksikkö / 2019

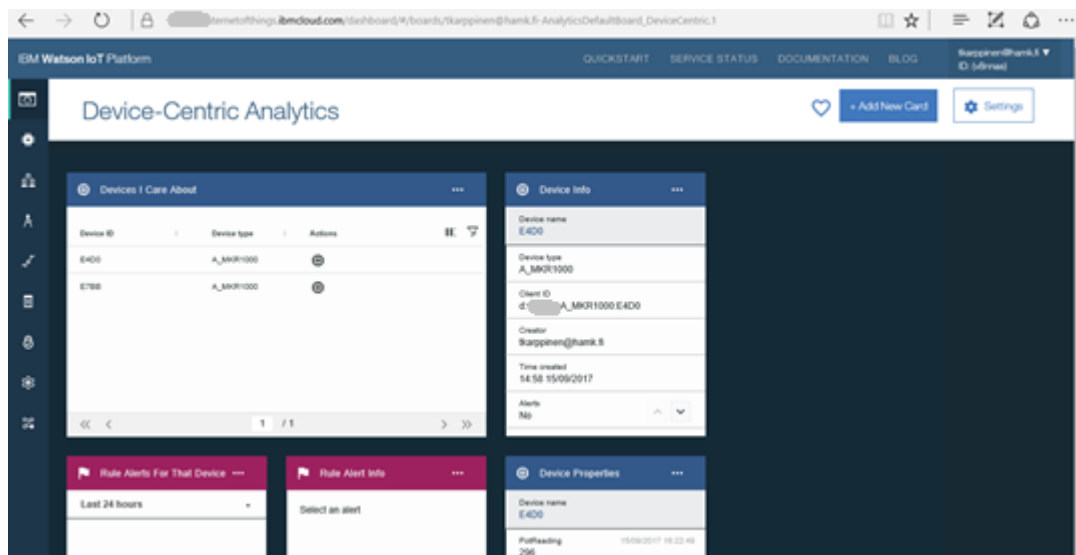
Määritä edelleen numeeriselle muuttujalle oikea tyyppi ja yksikkö.



Kuva 5.9 Mittausarvot piirtyvät näkyviin. / 2019

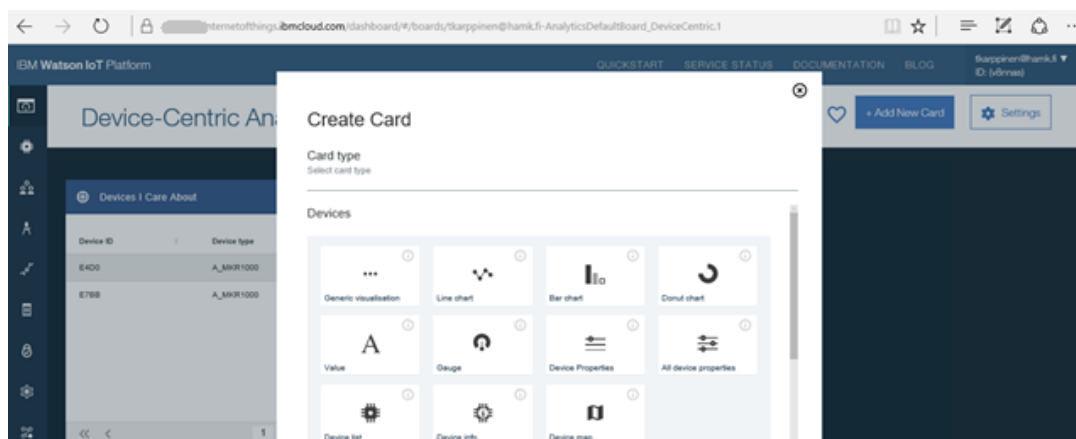
Nyt sinun pitäisi saada näkymä, johon piirtyy reaaliajassa mittausarvoja.

Jos sinulla oli valmiina Board -tyyppinä Device Centric Analytics, oli sinne valmiiksi valittu muutamia Card-tyyppejä. Mutta siihenkin pystyit itse luomaan aivan vastaavasti kuin yllä uuden Line chart -kortin.



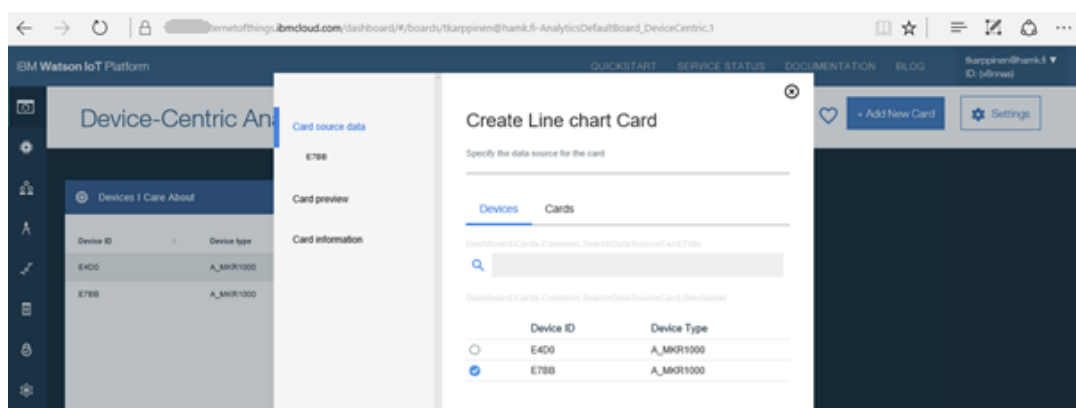
kuva 5.10 Device Centric Analytics, valmis esimerkkikortti / 2017

Valitse tästä näkymästä laitteesi. Valitse ylhäältä oikealta Add New Card.



Kuva 5.11 Card-tyyppejä / 2017

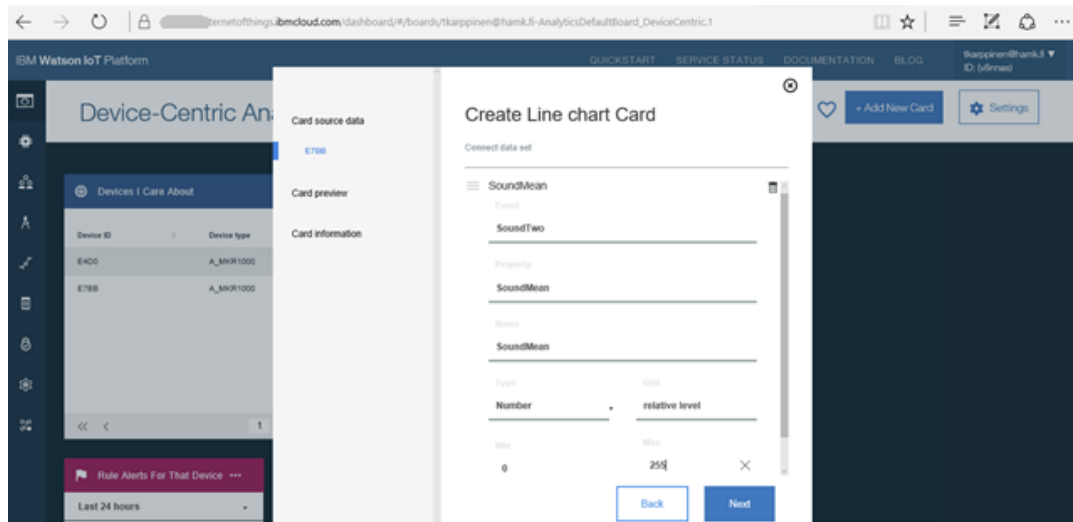
Valitse esitystavaksi Line chart.



Kuva 5.12 Seurattavan laitteen valinta / 2017

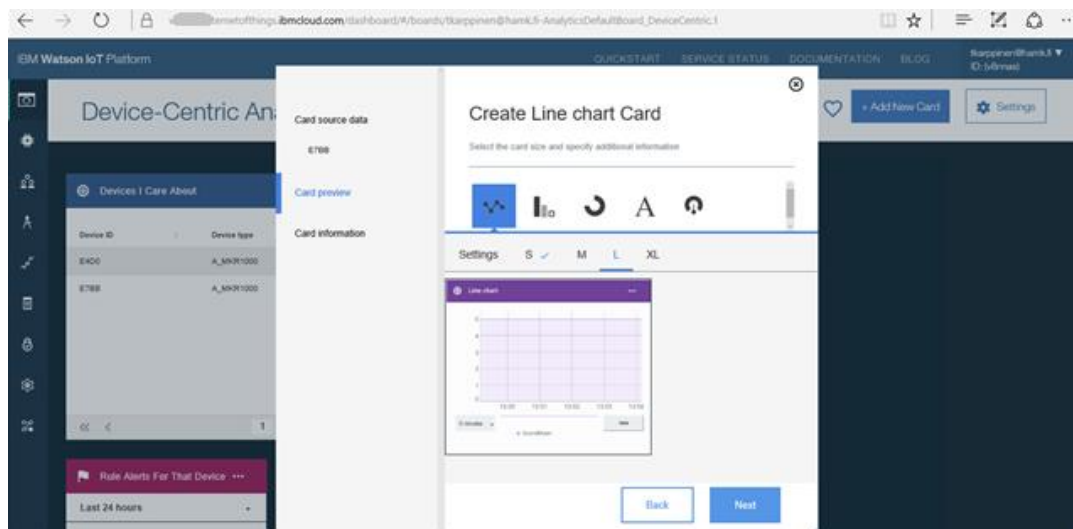
Valitse, miltä laitteeltasi haluat tiedot.

Seuraavaksi avautuu lehti, jolla määrittelet, mitä tietoja ja millä asteikolla haluat näyttää. Järjestelmä muistaa, mitä Event, Property, Name...tietoja olet aiemmin käyttänyt julkaisiessasi Watson –palvelussa. Saat riviä klikkaamalla nämä näkyviin alasvetovalikkona.



Kuva 5.13 Seurattavan Event-tapahtuman eli mittausarvon luennan määrittäminen

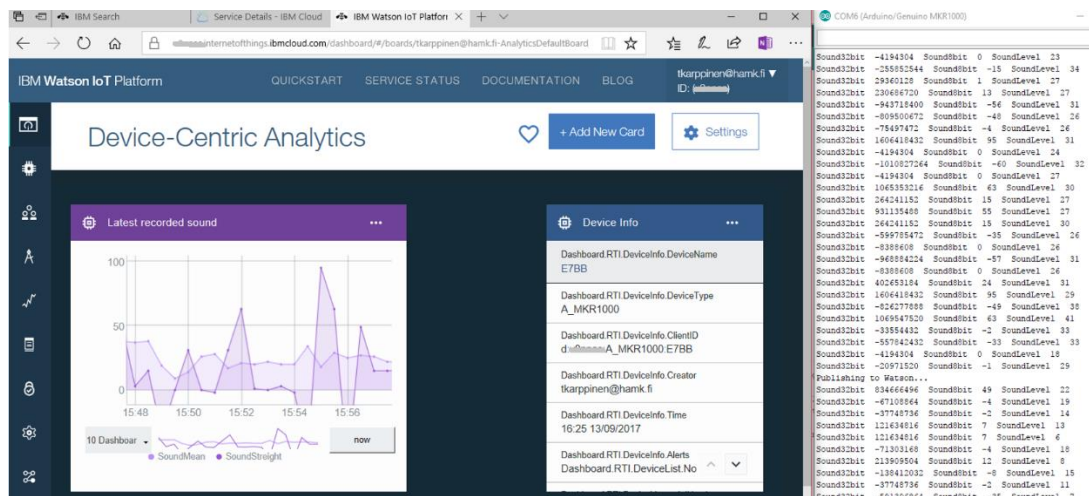
Voit edelleen vaikuttaa esityksen ulkoasuun.



Kuva 5.14 Esitystavan valinta

Valitse Next ja kirjoita kaavioillesi nimi. Submit-painikkeella avaat kaaviosi.

Kaavion asteikkojen laajuudesta ei tarvitse välittää. Watson vastaanottaa hetken aikaa mittausarvoja ja määrittää sopivat pysty- ja vaakasteikon laajuudet.



kuva 5.15 Mittausarvot näkyvät kortilla. Kuvassa oikealla näkymä Arduino IDE Serial Monitor-ikkunasta.

Dashboard – Card -näkö näkymä osaa skaalata näkymän oikealle asteikolle. Se vain kestää jonkin aikaa!

KÄYTTÖÖNOTTOTEHTÄVÄN 0 VALINNAINEN OSUUS.

Kappaleen 6 ja 7 toimintoja ei välttämättä tarvitse tehdä omaan järjestelmään. Ilman niitäkin ovat käyttöönottotehtävät 1, 2, 3, jne suoritettavissa.

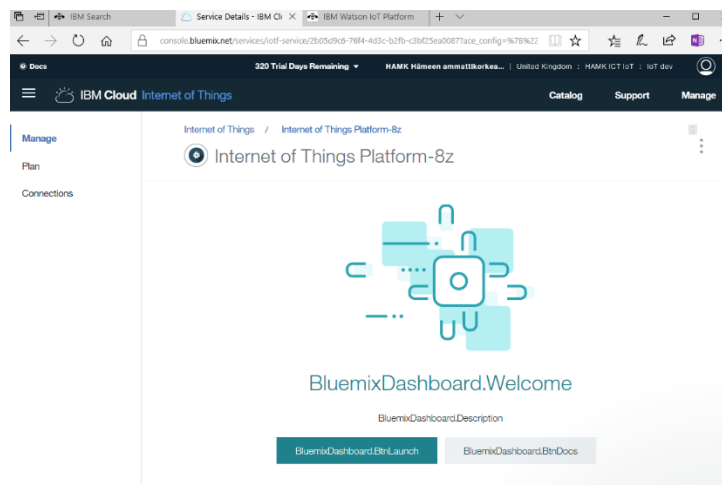
Kappaleet 6 ja 7 näyttävät, miten yksinkertaista on tehdä mittaustiedoille muutama analysointisääntö.

6. Datan analysointi IBM Watsonissa Dashboard:ssa

Watson IoT Dashboard:iin voidaan luoda joitakin yksinkertaisia datan analysointeja. Nämä on tarkoitettu laitteiston ylläpidon käyttöön – ei siis loppuasiakkaalle tuotettavaksi analyysiksi.

Kirjaudutaan IBM Cloud -tilille omilla käyttäjätunnuksilla. Oletamme, että tälle käyttäjälle jaetaan Watson IoT:hen tallentuvaa tietoa.

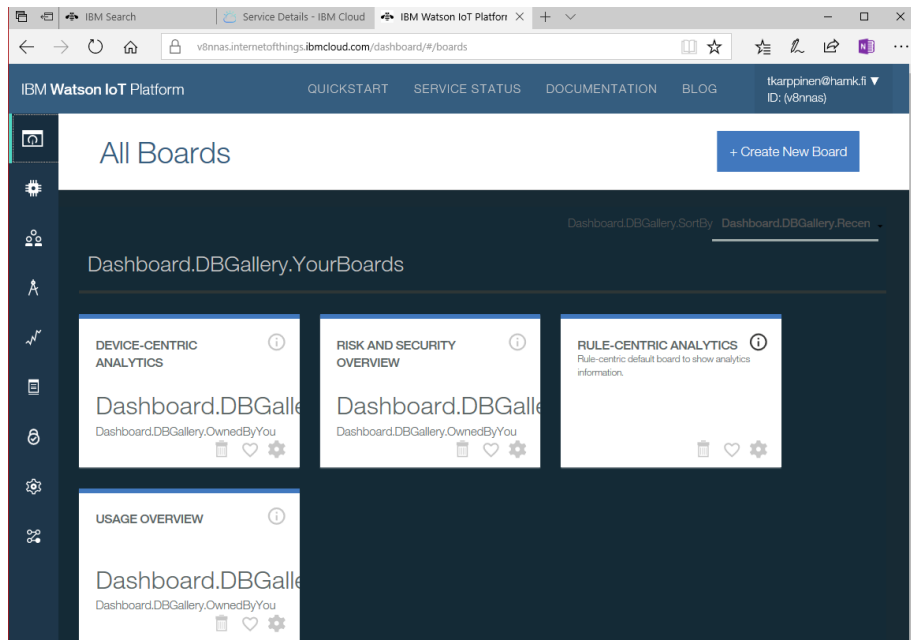
Hakeudutaan sivulle Console Internet of Things



Kuva 6.1 Käynnistetään aiemmin luotu IoT-alusta.

Klikataan painiketta BluemixDashboard.BtnLaunch. Painikkeen nimi voi olla myös yksinkertaisesti "Launch".

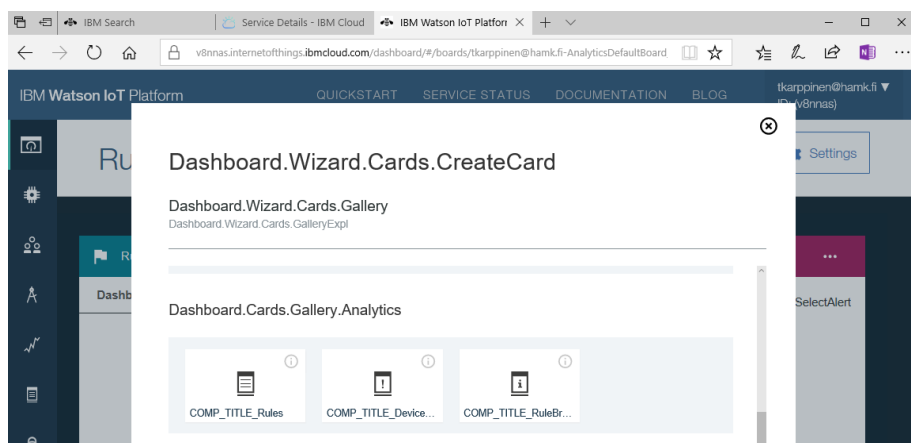
Näkymä siirtyy IBM Watson IoT Platform -näkömäksi. Valitaan valikosta vasemmalta Boards-näkymä kuten kuvassa alla.



Kuva 6.2 Boards-näkymä valittuna valikosta vasemmalta.

Valitaan RULE-CENTRIC ANALYTICS.

Avautuvassa näkymässä klikataan oikealla ylhäällä Add New Card.

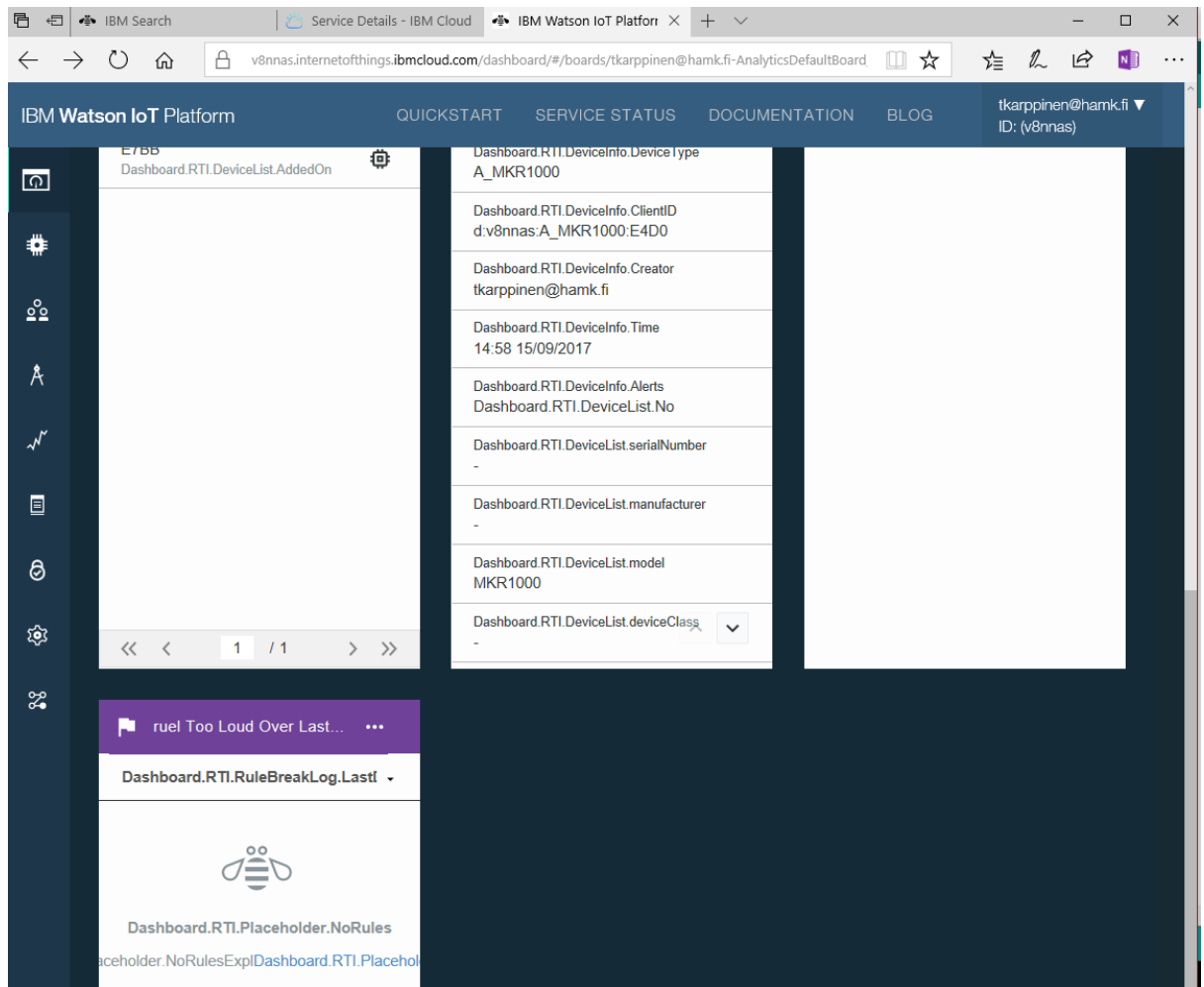


Kuva 6.3 Uusi kortti

Valitaan COMP_TITLE_RULES

Täytetään kaavake nimeämällä kaavake säännöksi rule Too Loud Over Last Hour.

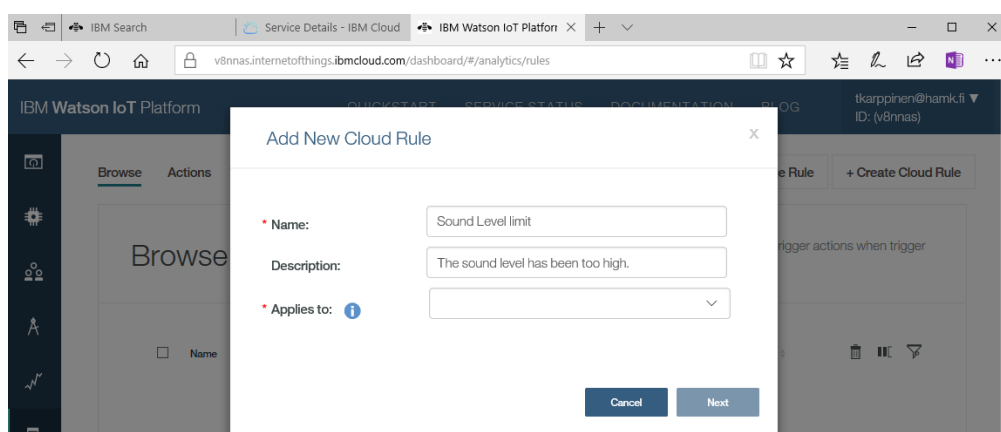
Sääntö ilmestyy uudeksi sääntökortiksi.



6.4 Uusi kortti kuvassa vasemmalla alhaalla

Uudessa sääntökortissa – kuvassa alimmainen – klikataan Dashboard.RTI.Placeholder .

Valitaan avautuvassa näkymässä ylhäältä oikealta +Create Cloud Rule

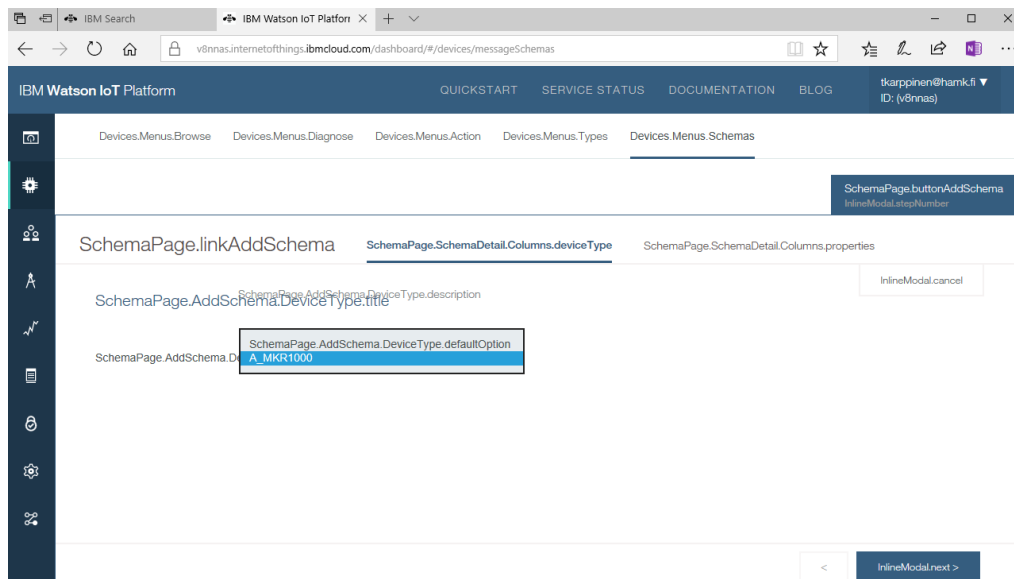


Kuva 6.5 Nimetään sääntö

Nimetään sääntö. kuvakkeesta i aukeaa ohje ” Go to Devices > Manage Schemas and add a schema for the device type. “

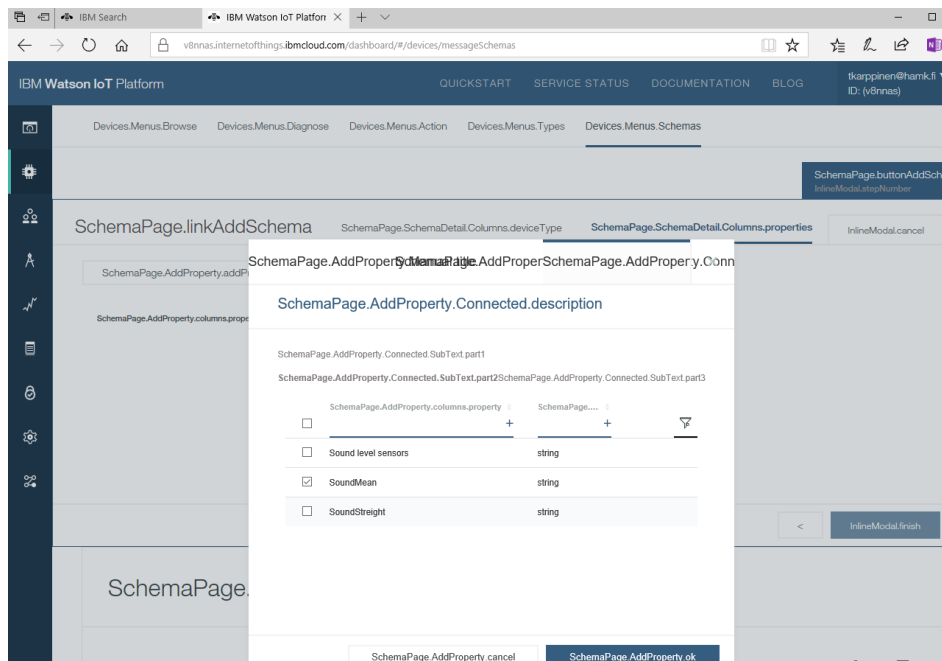
Sivun vasemmassa kehyksessä valitse Devices ja avautuvan sivun ylälaidassa Devices.Menus:Schemas. Ja edelleen SchemaPage.button.AddSchema.

Luot Schema:n laitetyypillesi, esim. A_MKR1000.



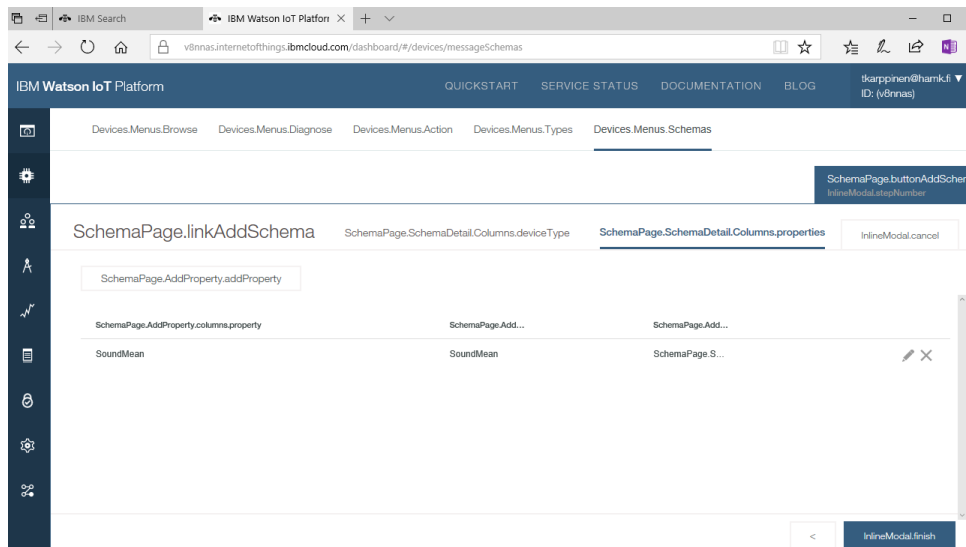
Kuva 6.6 Luodaan uusi "Schema".

Painikkeesta oikealta alhaalta eteenpäin. Kestää yllättäen jonkin aikaa, ennen kuin aukeavaan kaavakkeeseen ilmestyvät sinun IoT Platformilla käytössä olevat ominaisuudet eli "property" -tyypit. Valitse esim. SoundMean.



Kuva 6.7 Valitaan analysoitavat suureet

Ja eteenpäin painikkeesta SchemaPage.AddPropert.ok . ja edelleen InlineModal.finish .



Kuva 6.8 Valmis "schema"

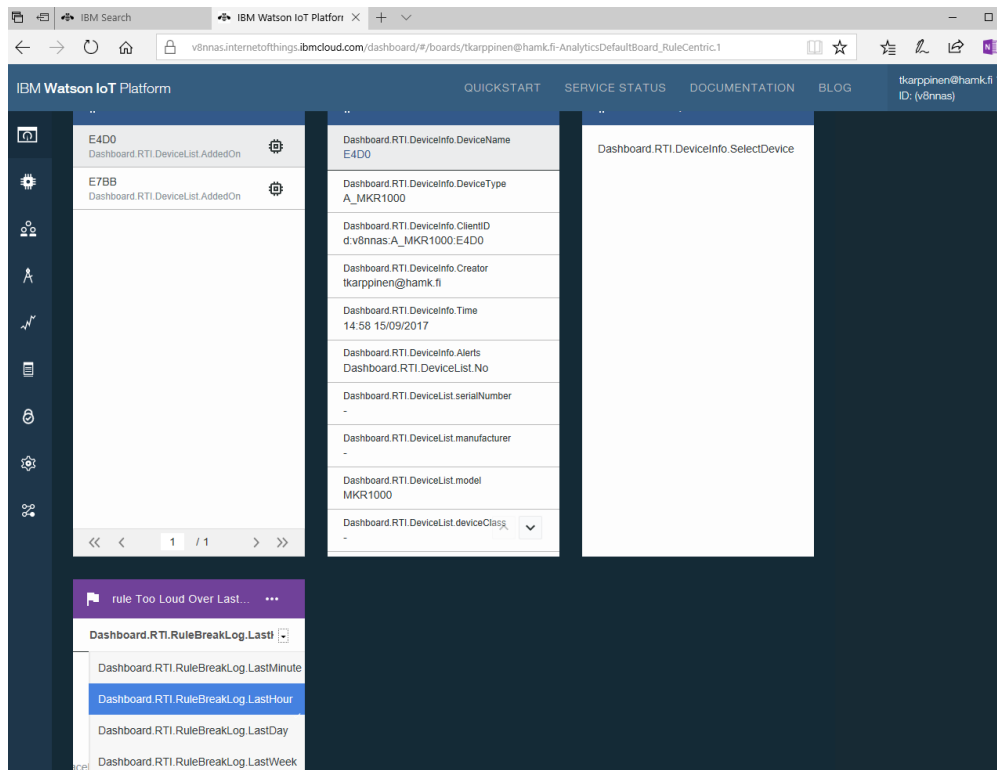
Olet luonut Scheman, jolla on ominaisuudet:

`SchemaPage.SchemaDetailViewPropertis.columns.property = SoundMean`

`SchemaPage.SchemaDetailViewPropertis.columns.name = SoundMean`

`SchemaPage.SchemaDetailViewPropertis.columns.type =`
`SchemaPage.Schemadetail.typestring`

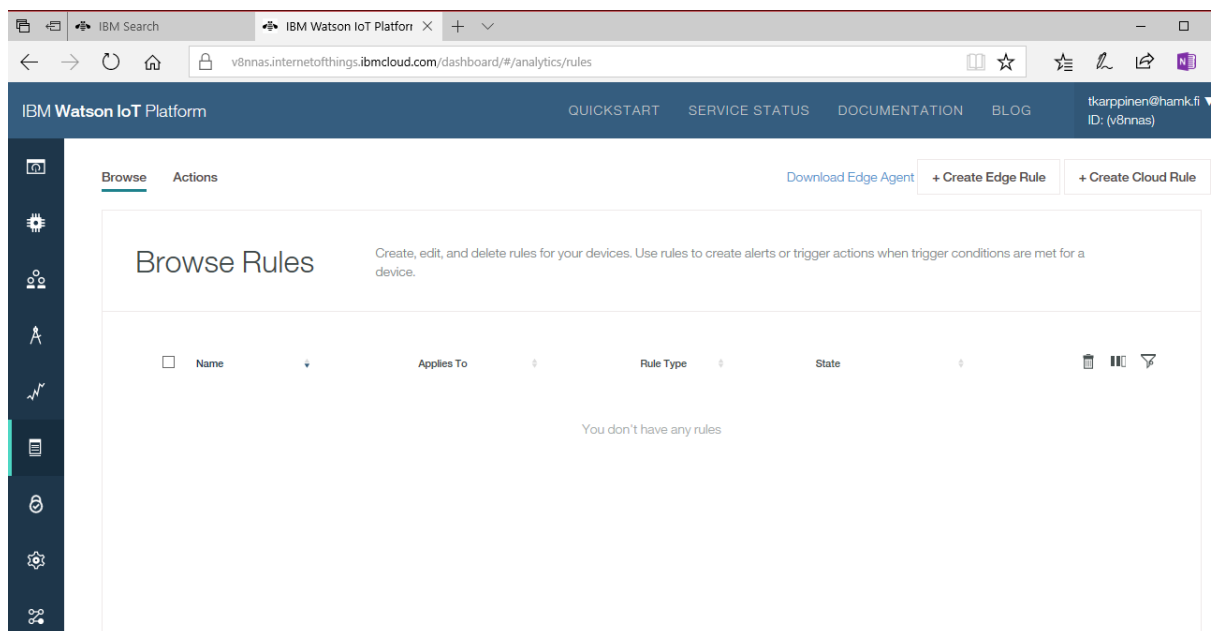
Palaamme nyt takaisin luomaan sääntöjä. Valitse vasemmalla kehyksessä Boards ja edelleen RULE-CENTRIC ANALYTICS. Valitse edellä luomasi uusi sääntökortti, esim. "rule Too Loud Over Last Hour ja siinä Dashboard.RTI:RulesBreakLog.LastHour ja edelleen kortin keskeltä Dasboard.RTI.Placeholder.



6.9 Sääntöön otetaan mukaan mittauksia tunnin ajalta.

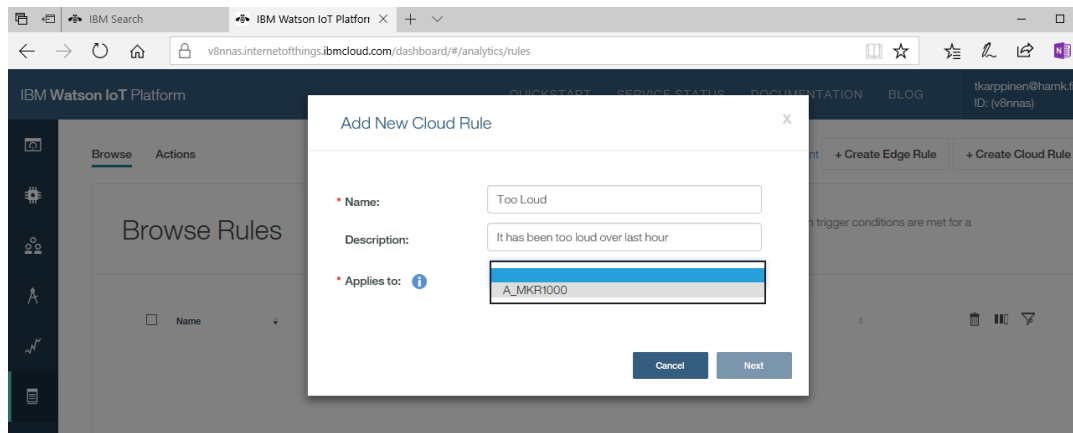
Tai samaan tilanteeseen pääset valitsemalla vasemmalla kehiksessä RULES.

Nyt voit uudelleen valita ylhäältä oikealta +Create CloudRule.



Kuva 6.10 Näkymä, johon säännöt luodaan.

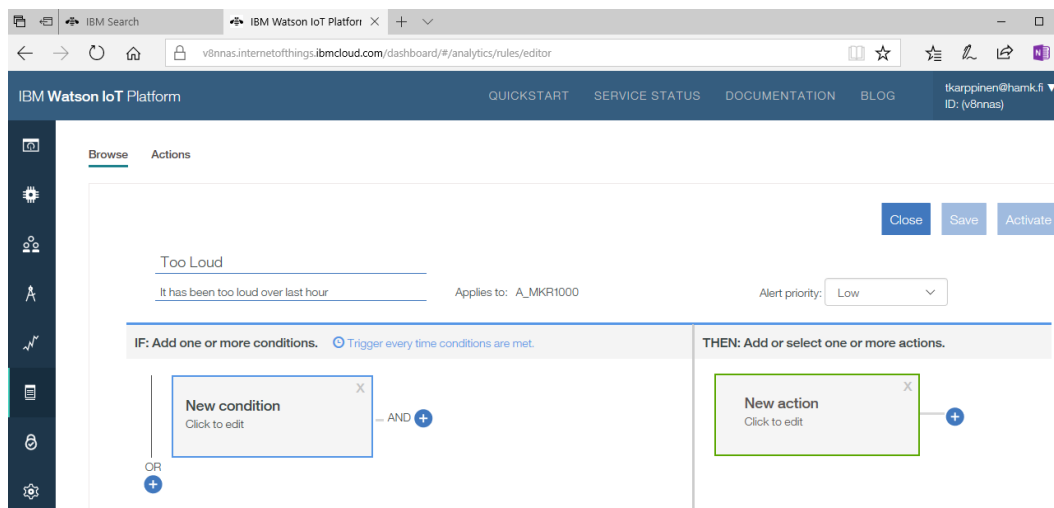
Nyt voit asettaa laitteellesi säännön.



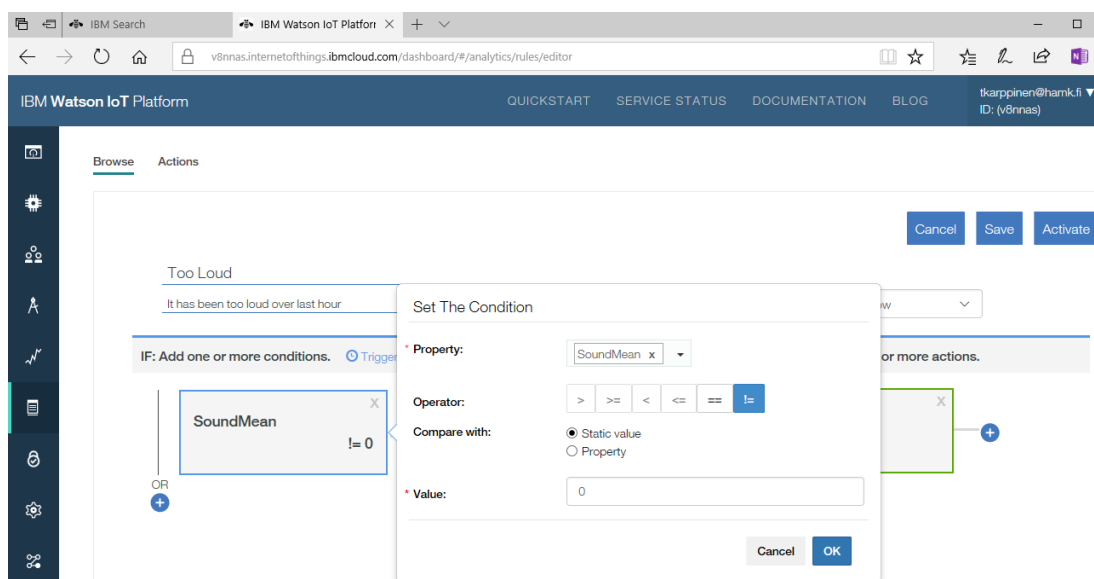
Kuva 6.11 Määritetään laite, jolle sääntö luodaan

Valitse laitteesi ja Next.

Avautuvassa näkymässä voit aloittaa sääntöjen luonnin.



Kuva 6.12 Sääntöjen logiset riippuvuudet



6.13 Yhden säännön määrittely.

Ja kappas, valittavana ovat vain operaattorit == sekä != . Siirsimme ääninäytteiden keskiarvon muodossa string. String-tyyppistä arvoa voi vain testata yhtäsuuruudella tai erisuuruudella toiseen string-tyyppiseen arvoon. Kirjoitamme siis vertailuarvoksi 0. Toki voimme kirjoittaa säännön arvo ei ole 0 ja arvo ei ole 1 ja arvo ei ole 2 ja arvo ei ole 3 ja arvo ei ole 4.....

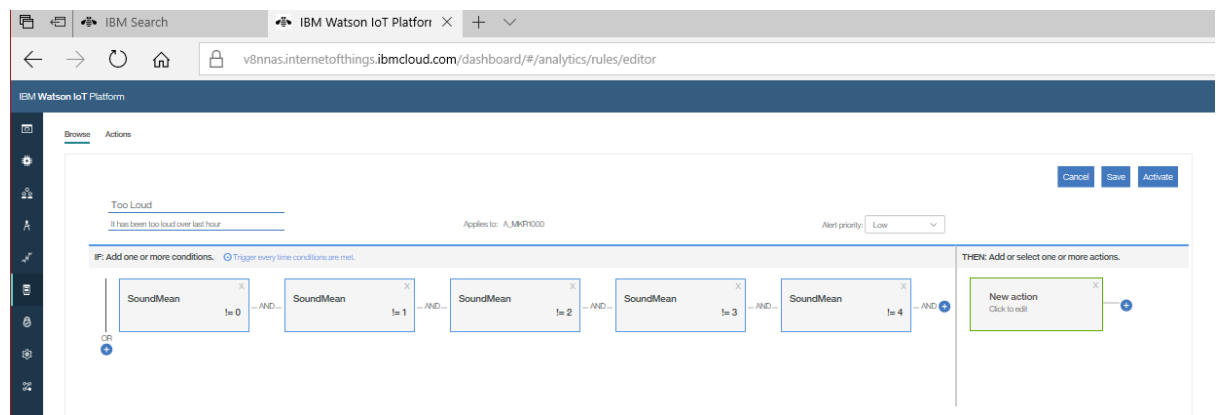
Näin tapahtui, koska client-laite eli tässä tapauksessa Arduino MKR1000 -korttimme oli vahingossa sanoman sisällön määräämällä ohjelmarivillä asetettu lähettämään muuttuja-arvoa muunnettuna teksti-string-muodossa.

Ohjelmarivi, joka lähettää muuttujan float-muodossa mutta JSON-rakenteen sisällä ascii-merkkijonona:

```
String wpayload = "{\"d\":{\"TemperatureSensor\":\"TC1 \",\"TempScaledR1234\":\"" +  
String(tempScaledF)+ ", \"TempStreightR1234\":\"" + String(temp14bit)+"\"}"};
```

```
MQTTc.publish("iot-2/evt/TemperatureTC1/fmt/json", wpayload);
```

Tämä on siis se oikea muoto, jolla Watson osaa lukea muuttujan arvon desimaalilukuna.

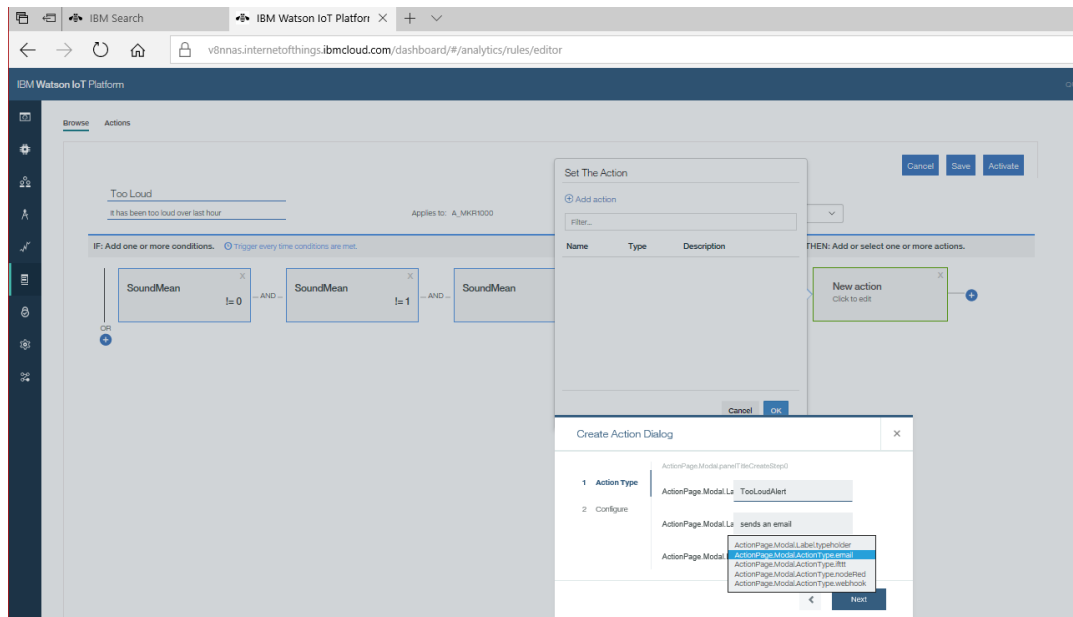


6.14 Useita sääntöjä yhdistettynä logiikan operaatioin

Seuraavaksi määritämme, mitä säännön toteutuessa tapahtuu. Klikataan saman näkymän New Action

ja avautuvassa ruudussa Add action.

Lisää nimi, kuvaus ja tyyppi: TooLoudAlert, sends an email, ActionPage.Modal.ActionType:email .



Kuva 6.15 Näkymä, jossa voidaan lisätä säännön toteutumista seuraava toimenpide.

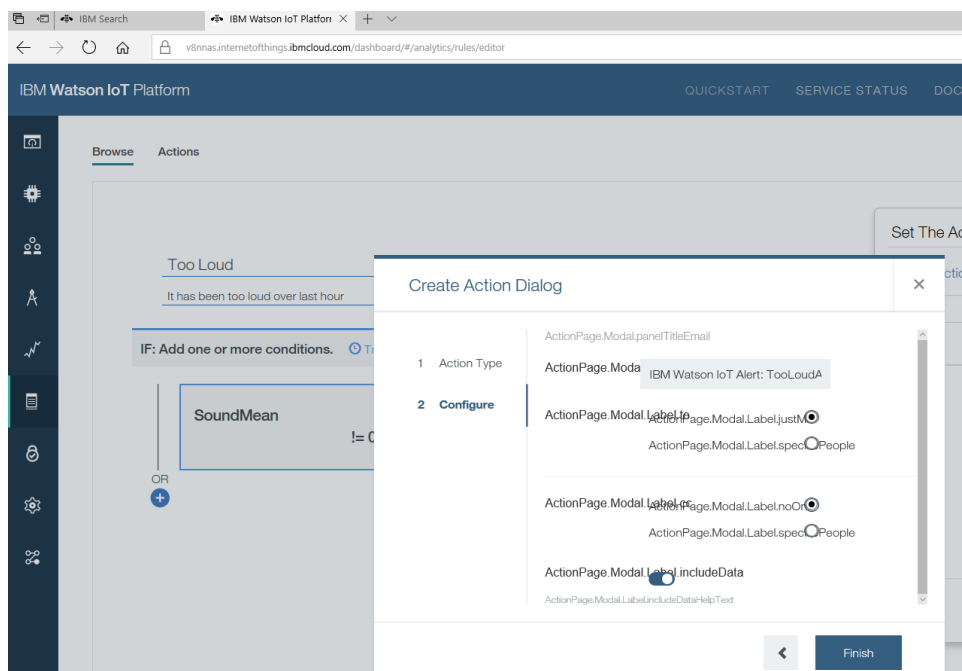
On ehkä turhaa lähettää sähköposti, kun äänenpaineen taso ylittyy. Käytämme tätä kuitenkin ensimmäisenä esimerkkinä. Voi tämä sopia esimerkiksi joihinkin kiinteistöhoidon tehtäviin.

Klikkaa Next ja täytä arvot

ActionPage.Modal.Label.tospecialPeople [sopiva sähköpostiosoite]

ActionPage.Modal.Label.cc noon

Aseta valituksi ...includeData.



6.16 Toimenpide

Ja klikkaa Finish.

7. Metadatan liittäminen mitattavaan arvoon.

7.1. Anturoitavan kohteen jako perustyypeihin

Voidaan ajatella olevan kahta eri päätyyppiä kohteita, jotka lähettävät tietoa antureilta IoT-tietokantaan.

A. Ainutkertainen kohde

Kohde voi olla ainutkertainen. Tällainen on esimerkiksi tehtaan tuotantolinja.

Tuotantolinjassa voi olla satoja mittauspisteitä, joiden tietoja halutaan siirtää IoT-alustaan.

Mittauspisteet voivat olla päivittäistä käyttöä ohjaavasta automaatiojärjestelmästä erillään tai ne voivat olla automaatiojärjestelmän kanssa yhteisiä. Tyypillisesti mittauspisteet kuitenkin pysyvät täsmälleen samassa käytössä vuosikausia.

Tällöin kunkin mittauspisteen dataan voidaan liittää metadata eli määrittelytietoja samassa ohjelmistossa, joka lukee anturiarvoja ja valmistelee tiedon lähetettäväksi IoT-alustalle.

B. Suurina määrinä valmistettu kohde

Kohteen voidaan ajatella olevan osa jotain kuluttajamarkkinoille suunnattua tuoteperhettä.

Urheiluvälinevalmistaja voi esimerkiksi sijoittaa IoT-laitteen jokaiseen valmistamaansa juoksutossuun. Käyttäjä voidaan houkutella rekisteröimään tossunsa esimerkiksi tarjoamalla juoksusuoritusten analysointipalveluja. Samalla luomme itsellemme suoran yhteyden asiakkaaseemme markkinointitoimenpiteitä ajatellen.

Tällöin emme vaivaa asiakasta sillä, että hänen pitäisi kirjata juoksutossunsa IoT-laitteeseen joitain omistajatietoja. Käyttäjä kirjaa vain tuotetta rekisteröidessään tuotteensa yksilöllisen sarjanumeron ja tietenkin omat yhteystietonsa urheiluvälinevalmistajan verkkosivulla.

IoT alustaan tuotteemme IoT-laite tuo sarjanumerolla identifioitavan aktiviteettitiedon aina, kun tuotetta käytetään. Rekisteröinnistä tuomme asiakkaan yksilöivän tiedon IoT-alustaan.

Helposti voimme IoT alustaan luoda päättelysäännöt, joiden perusteella lähtee oikealle asiakkaalle oikeaan aikaan markkinointiviesti esim. mainosbannerin muodossa. Oikea aika markkinointiviestille voi olla, kun tuotetta käytetään väärissä sille sopimattomissa sääolosuhteissa.

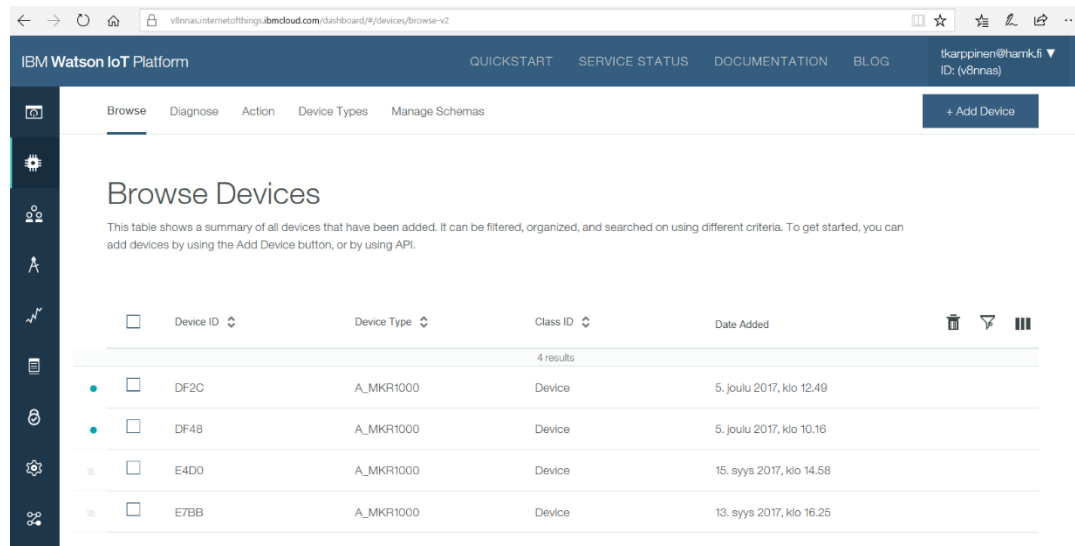
Suurin osa IoT-sovelluksista on ja tulee yhä enenemässä määrin olemaan tämän tyyppin B kaltaisia.

7.2. Metadatan liittäminen mittaustietoon IoT-alustalla

Käytämme tässä jälleen esimerkkinä IBM Bluemix Watson –alustaa.

Oletamme seurattavan kohteen olevan tyyppiä B. Emme kuitenkaan nyt olettaa asiakkaita olevan niin paljon, että metadatan liittäminen yksilöityyn IoT-tuotteeseen olisi automatisoitu.

Tässä esimerkissä yksilöityyn IoT-laitteeseen liittyvä metadata siirretään Jason-muotoisena määrittelynä Watson IoT:ssä sille varattuun kenttään.



Device ID	Device Type	Class ID	Date Added
DF2C	A_MKR1000	Device	5. joulukuuta 2017, klo 12:49
DF48	A_MKR1000	Device	5. joulukuuta 2017, klo 10:16
E4D0	A_MKR1000	Device	15. syyskuuta 2017, klo 14:58
E7BB	A_MKR1000	Device	13. syyskuuta 2017, klo 16:25

Kuva 7.1 Valitaan laite, johon metadata liitetään.

Valitaan laitekannasta laite, johon metadata halutaan liittää.

Tässä esimerkissä laitteet ovat rakenteeltaan ja ohjelmoinniltaan keskenään identtisiä. Uutta laitetta Watson IoT –alustaan perustettaessa on kirjattu Device ID –tieto. Tämä voisi olla tuotteen sarjanumero. Esimerkissämme tähän on kirjattu MAC –numeron neljä viimeistä heksadesimaalimerkkiä. Valitaan laite DF48. **Klikkaa** yllä olevan kuvan näkymässä **laitteen riviä** ja edelleen **Device Information**.

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes links for QUICKSTART, SERVICE STATUS, DOCUMENTATION, and BLOG. The user is logged in as tkarpainen@hamk.fi with ID: (v8nnas). The main section is titled 'Browse' and contains a table of devices. The table has columns for Device ID, Device Type, Class ID, and Date Added. There are 4 results shown. The first device is DF2C, and the second is DF48. The third device is E4D0. Below the table, there is a section for 'Device Information' for the selected device DF48. This section includes fields for Serial Number, Model, Description, Hardware Version, Manufacturer, Device Class, Firmware Version, and Descriptive Location. The 'View Metadata' button is visible in the top right corner of the device information section.

Device ID	Device Type	Class ID	Date Added
DF2C	A_MKR1000	Device	5. joulu 2017, klo 12.49
DF48	A_MKR1000	Device	5. joulu 2017, klo 10.16
E4D0	A_MKR1000	Device	15. syys 2017, klo 14.58

Device Information for DF48:

Field	Value
Serial Number	--
Model	MKR1000
Description	with sensors
Hardware Version	--
Manufacturer	--
Device Class	--
Firmware Version	--
Descriptive Location	--

Kuva 7.2 IoT-laitteelle perustamisvaiheessa kirjattuja tietoja.

Avautuvassa näkymässä **klikkaa kynän kuvaa** päästäksesi editoimaan ja **edit Metadata**.

The screenshot shows the IBM Watson IoT Platform dashboard with the 'Metadata Information' dialog box open. The dialog box contains a text area for entering metadata in JSON format. The text area is pre-filled with the following JSON:

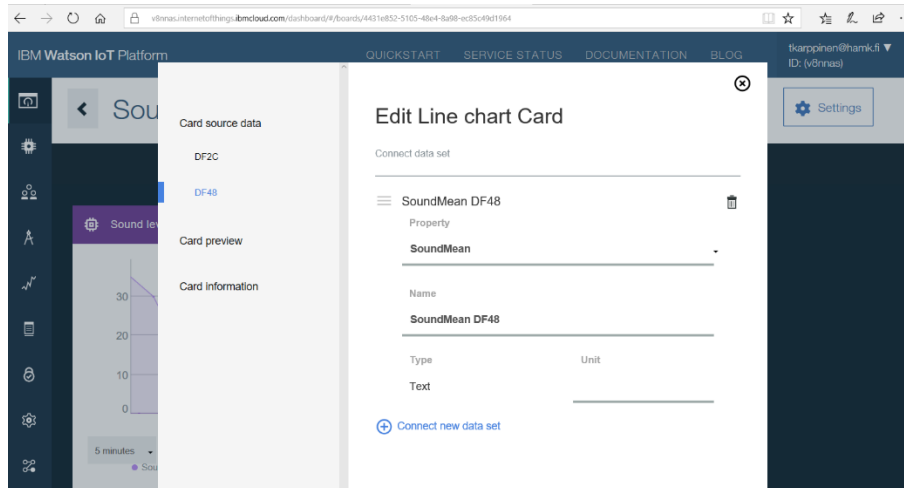
```
{
  "Sensors": "PModMIC3",
  "Location": "HAMK Ri-Ka-A111",
  "Installation": "on PA Speaker"
}
```

The dialog box also includes a 'Cancel' button and an 'Update' button. The background shows the same device list as in the previous screenshot, with device DF48 selected.

Kuva 7.3 Yksittäisen laitteen Metadatan editointi

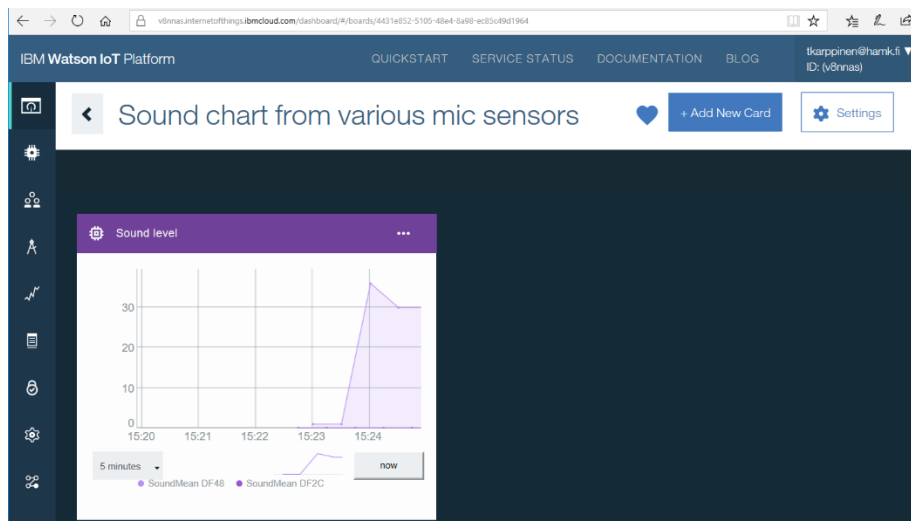
Kirjoita metadata JSON-muodossa ja tallenna klikkaamalla ensin editointi-ikkunan **Update** ja heti perään alhaalla **Save**.

Luo edellisen kappaleen ohjeen mukaisesti uusi käyttöliittymäkortti eli ”Board”. Tuo siihen vähintään kahden eri laitteen anturitiedot. Luodessasi korttia valitset laitteen ja laitteen lähettämän anturitiedon. Kirjaa anturitiedon **kenttään Name** valitsemasi anturitiedon nimeksi mittauksen yksilöivä nimi. Tässä esimerkissä yksilöintitietona on käytetty IoT-laitteen WLAN-liittymän MAC-numeron neljää viimeistä merkkiä.



Kuva 7.4 Laitteen lähettämän anturidatan mukaisesti valitaan oikea anturitieto ja sille nimi.

Näin saat kortille usean IoT-laitteen tiedot ja voit erottaa ne toisistaan.



Kuva 7.5 Board –kortilla näkyy kahden eri IoT-laitteen anturiarvoja.

Kun haluat lukea kyseiseen anturiarvoon liittyvän metadatan, on palattava takaisin laitenäkymään. Valitse vasemmalta toiseksi ylin kuvake eli **Devices**.

Valitse taas näkymässä **haluamasi laite** ja edelleen **Device Information, View Metadata**.