

IBM Watson – T0 – MKR1000 – Getting started

Käyttöönottotehtävä 0

1. IoT –alusta ja mikro-ohjain

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

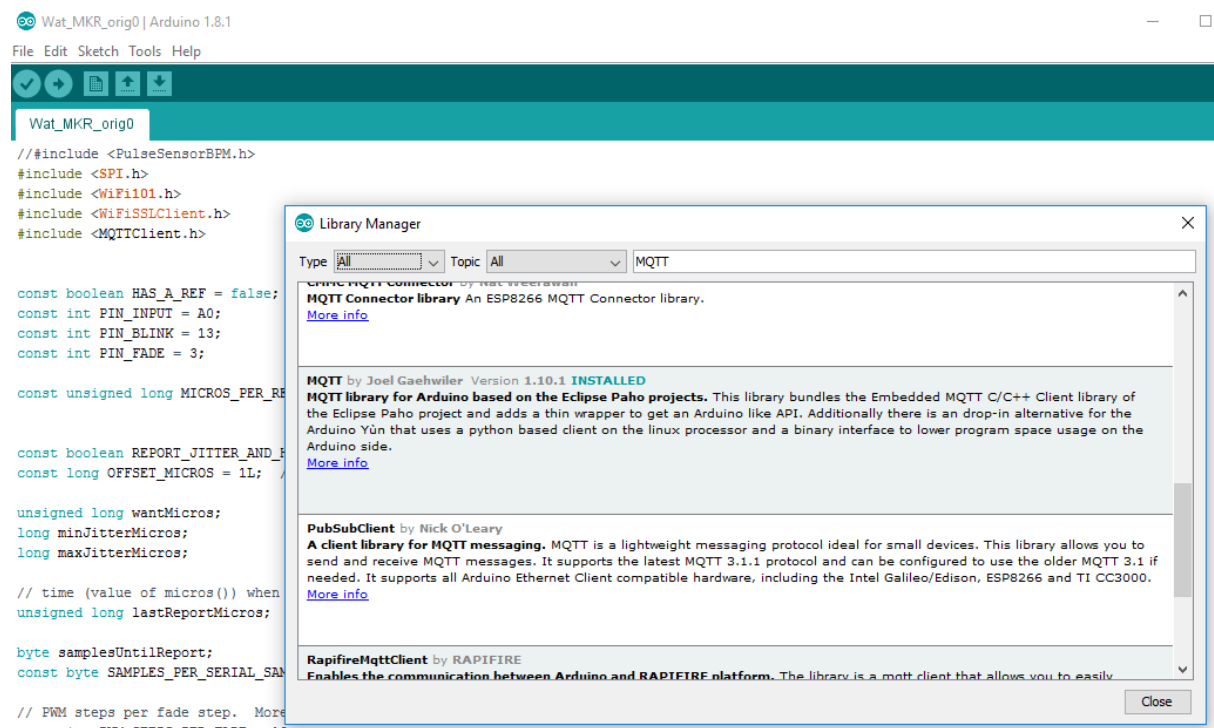
<https://www.ibm.com/internet-of-things/platform/iot-developer/>

Anturidatan keräämiseen käytämme Arduino MKR1000 –korttia.

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

2. MKR1000 -koodi

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



Kuva 2.1 MQTT-kirjasto Joel Gaehwiler.

Lataa kirjasto.

Käytämme koodimme lähtökohtana 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>  
  
// WLAN  
char ssid[] = "Moto_Z2_TK"; // your network SSID (name)  
char pass[] = "xxxxxxxxxxxxx"; // your network password (use for WPA)  
  
//char ssid[] = "HAMKWlan"; // your network SSID (name)  
//char pass[] = "xxxxxxxxxxxxxxxxxxxxx"; // 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:v8nnas:A_MKR1000:DF48";  
char *user_id = "use-token-auth"; // telling that authentication will be done with token  
char *authToken = "xxxxxxxxxxxxx"; // Your IBM Watson Authentication Token  
  
//char *ibm_hostname = "a@your-org-id.messaging.internetofthings.ibmcloud.com";  
char *ibm_hostname = "v8nnas.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:iqwckl: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 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 TK 14.11.2017

Ohjelmakoodin esimerkki toimii vain, jos organisaatio ja laite on rekisteröity IBM Bluemix, 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", "TLS Optional" .

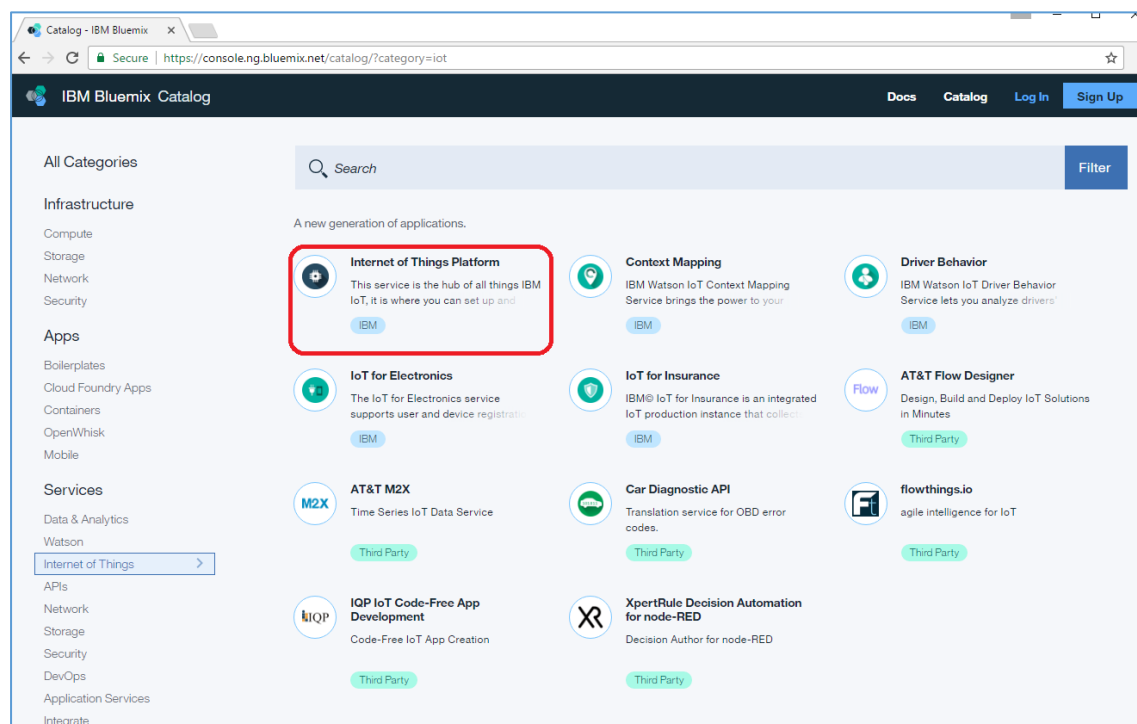
Riippumatta siitä, mitä IoT-alustaa tai mqtt-brokeria käytetään, on jokaiselle alustaan tai brokeriin yhteydessä olevalle laitteelle luotava oma tunnus client ID ja salasana authentication token. Jos samoilla tunnuksilla yritetään kirjautua samanaikaisesti useasta laitteesta, estää IoT-alusta tai mqtt-broker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

3. IBM Bluemix

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

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

Valikosta vasemmalla valitse "Internet of Things". Alla olevat kuvat ovat IBM Bluemix:stä siinä muodossa kuin sivut aukesivat 19.2.2017 .



kuva 3.1 IBM Bluemix, Internet of Things, Internet of Things Platform

Valitse maksuohjelmaksi ilmainen vaihtoehto

Pricing Plans

Monthly prices shown are for country or region: [United States](#)

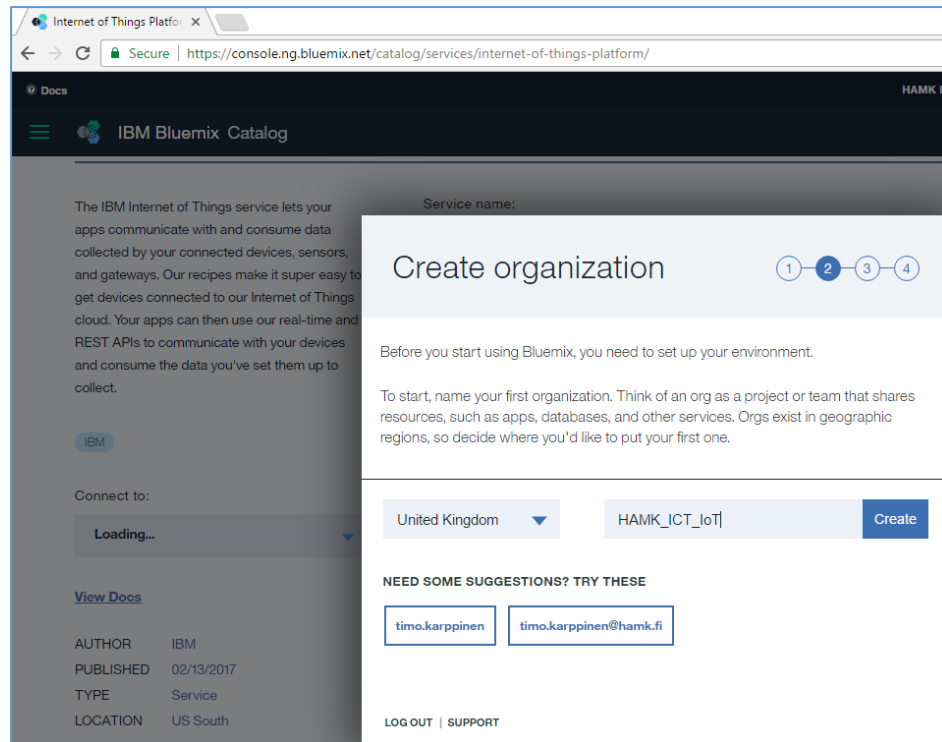
PLAN	FEATURES	PRICING
<div>✓</div> <div>Lite</div>	<div>Includes up to 20 registered devices, and a maximum of 100 MB of each data metric</div> <div>Maximum of 20 registered devices</div> <div>Maximum of 10 application bindings</div> <div>Maximum of 100 MB of each of data exchanged, data analyzed and edge data analyzed</div>	<div>Free</div>
<div>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.</div>		
<div>Standard</div>	<div>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.</div> <div>Charge per MB of data exchanged (tiered by usage in MB)</div> <div>Charge per MB of data analyzed</div> <div>Charge per MB of edge data analyzed</div> <div>Multi-Tiered</div>	<div>Expand each section to view details</div>

kuva 3.2. Ilmainen käyttö sopimus

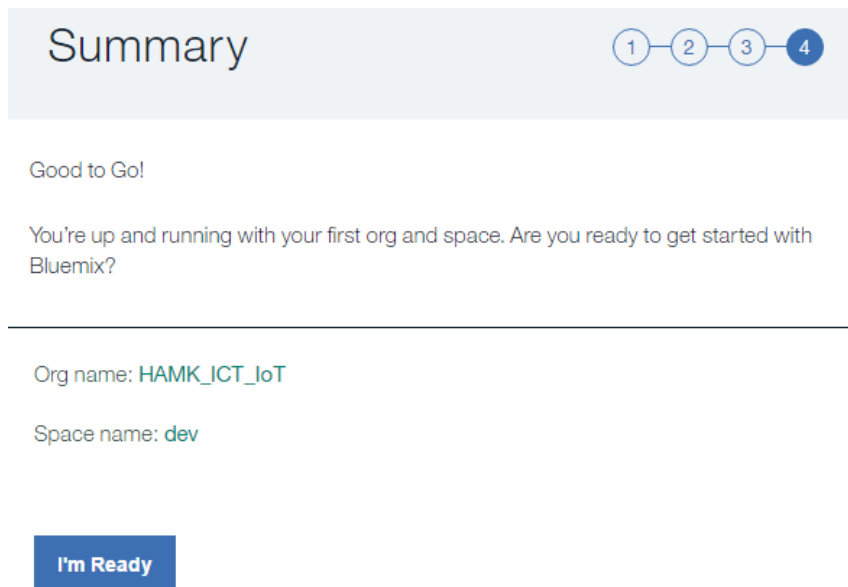
kuva 3.3. Ilmainen kokeilusopimus on voimassa 30 vrk.

Luo tili ja seuraa kirjautumisohjeita. 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.

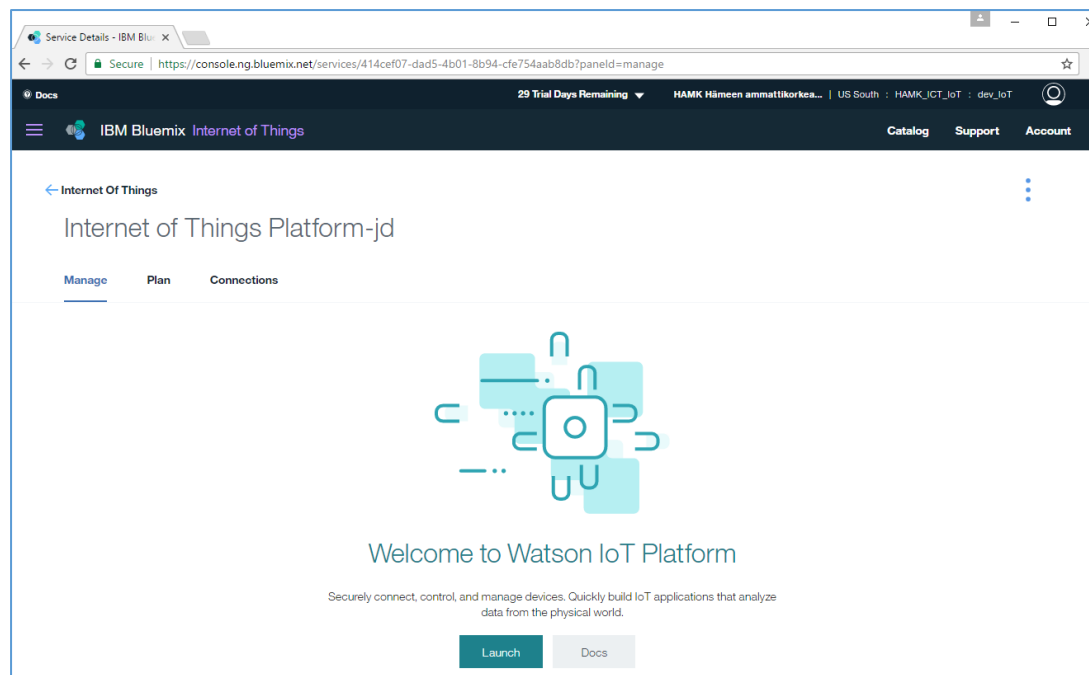


kuva 3.4 Tili sidotaan käyttämään palveluita esim. UK servereiltä.



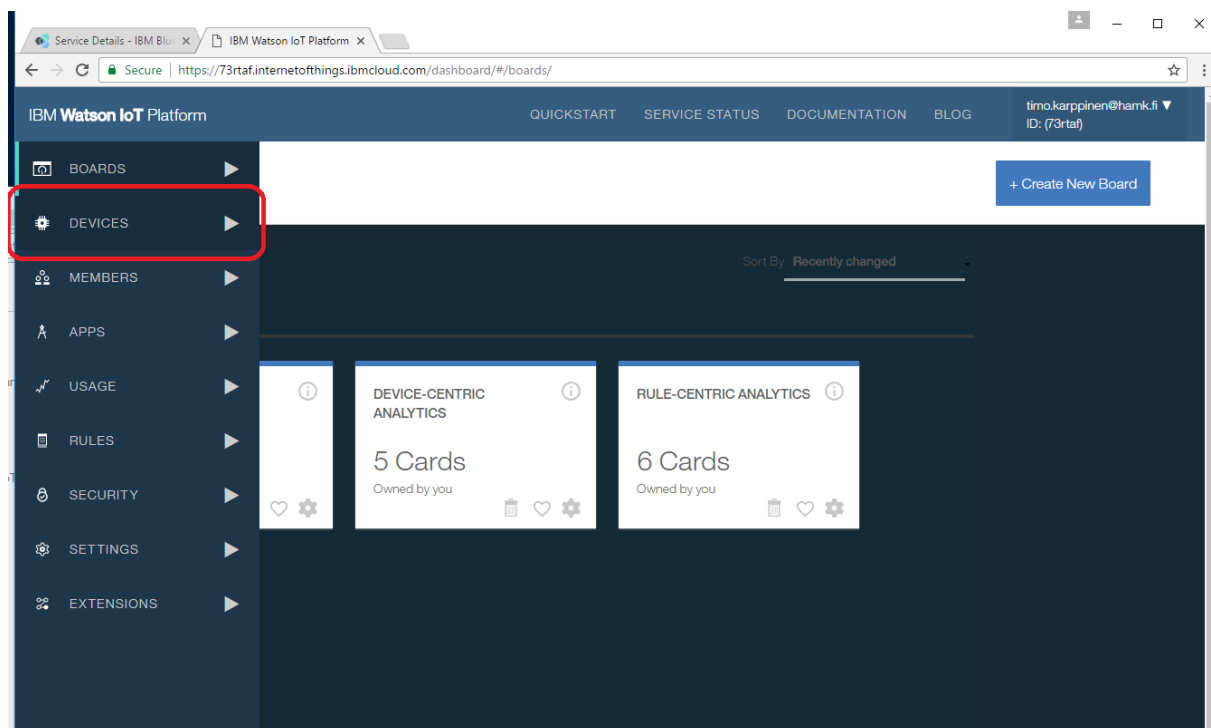
kuva 3.5 Tilin tiedot

Päätynyt takaisin kuvan 3.2 tilanteeseen. valitse alareunasta sininen painike "Create".
Nyt päädyt uudelle sivulle.



Kuva 3.6 IoT Platform

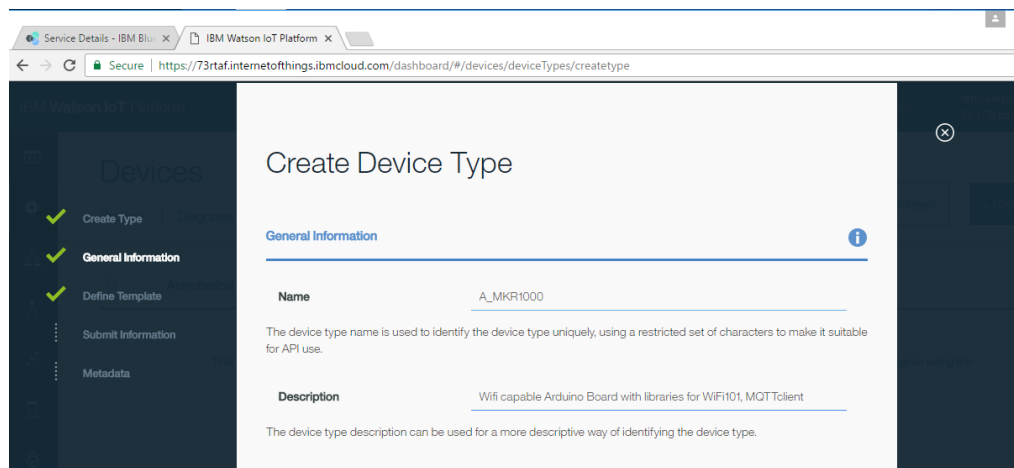
Valitse "Launch"



Kuva 3.7 IoT platform

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

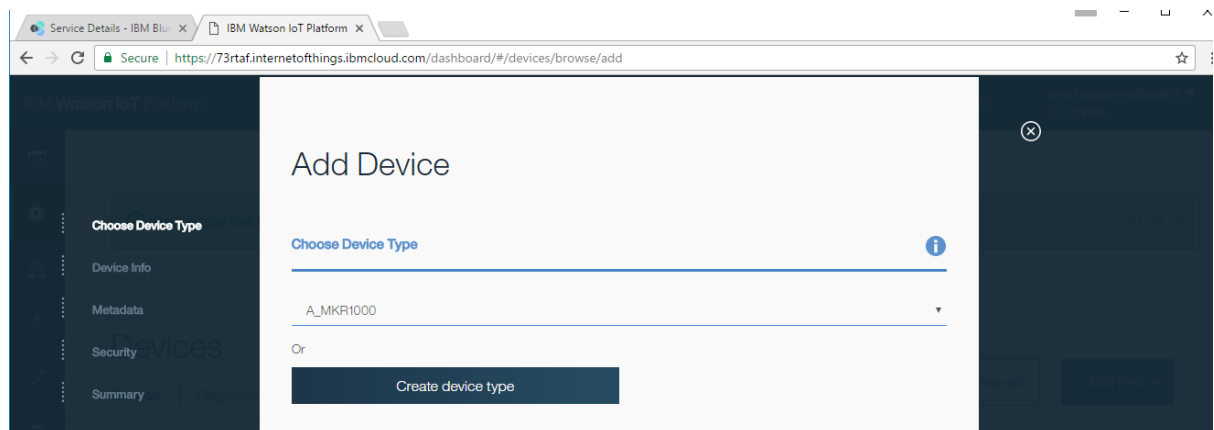
Täytä laitteelle tunnistetiedot. Etenet aina painikkeella "Next".



The screenshot shows the 'Create Device Type' page in the IBM Watson IoT Platform. The left sidebar contains a 'Devices' menu with options: 'Create Type', 'General Information', 'Define Template', 'Submit Information', and 'Metadata'. The main content area is titled 'Create Device Type' and has a 'General Information' section. It contains two fields: 'Name' with the value 'A_MKR1000' and 'Description' with the value 'Wifi capable Arduino Board with libraries for WiFi101, MQTTClient'. Below the description field, there is a note: 'The device type description can be used for a more descriptive way of identifying the device type.'

Kuva 3.8 laitteesi tiedot.

Kun olet täyttänyt tiedot, ilmestyy laite laiteluetteloosi.

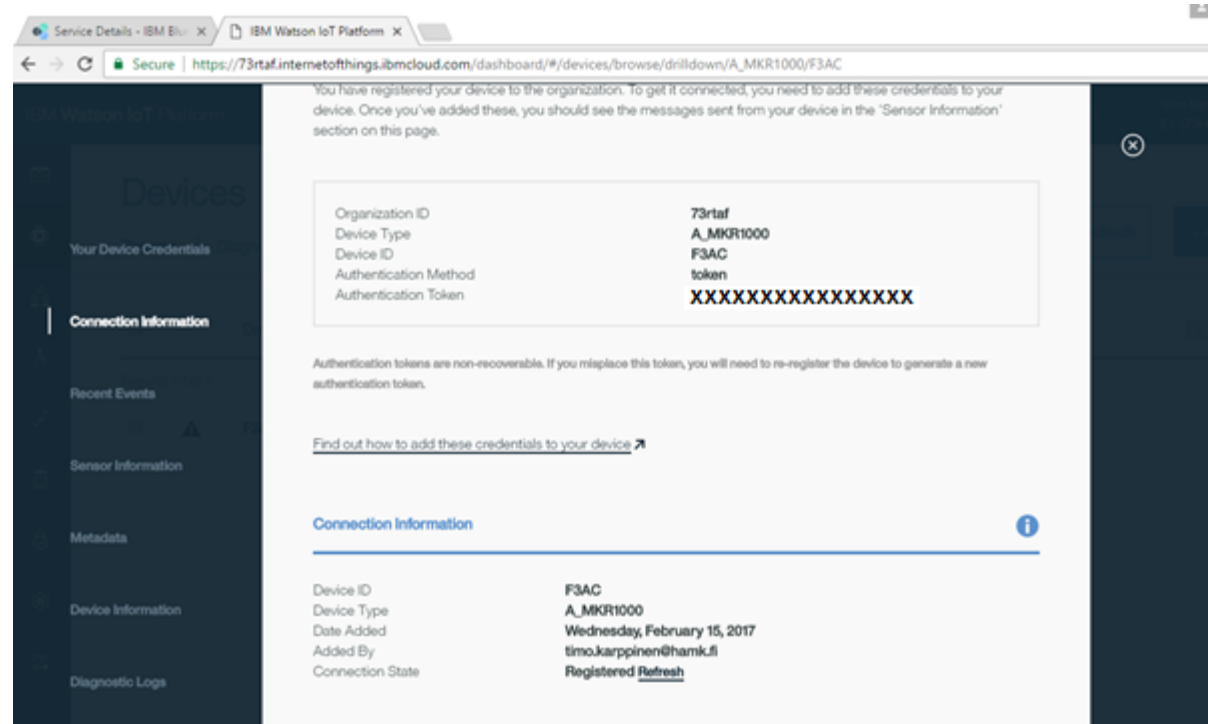


The screenshot shows the 'Add Device' page in the IBM Watson IoT Platform. The left sidebar contains a 'Choose Device Type' menu with options: 'Choose Device Type', 'Device Info', 'Metadata', 'Security', and 'Summary'. The main content area is titled 'Add Device' and has a 'Choose Device Type' section. It contains a dropdown menu with the value 'A_MKR1000' and a 'Create device type' button.

Kuva 3.9. Äskettäin luotu laite.

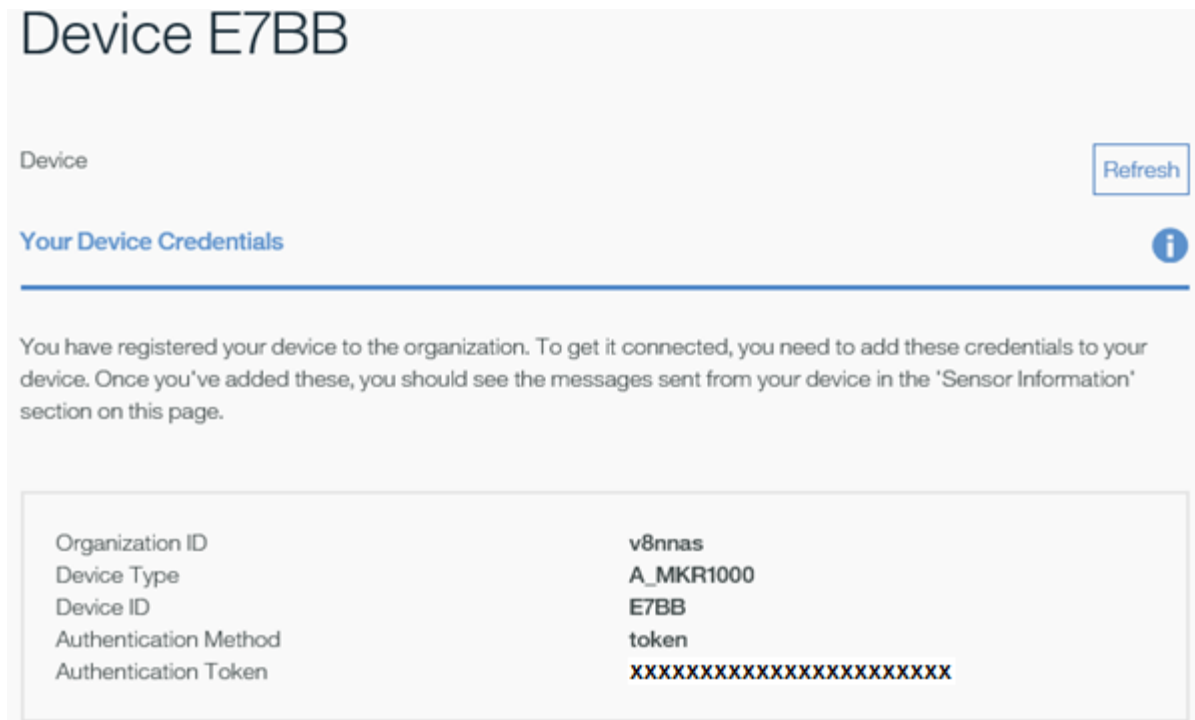
Valitse laite ja jatka eteenpäin.

Laite ja tunnistetiedot luotiin nyt.



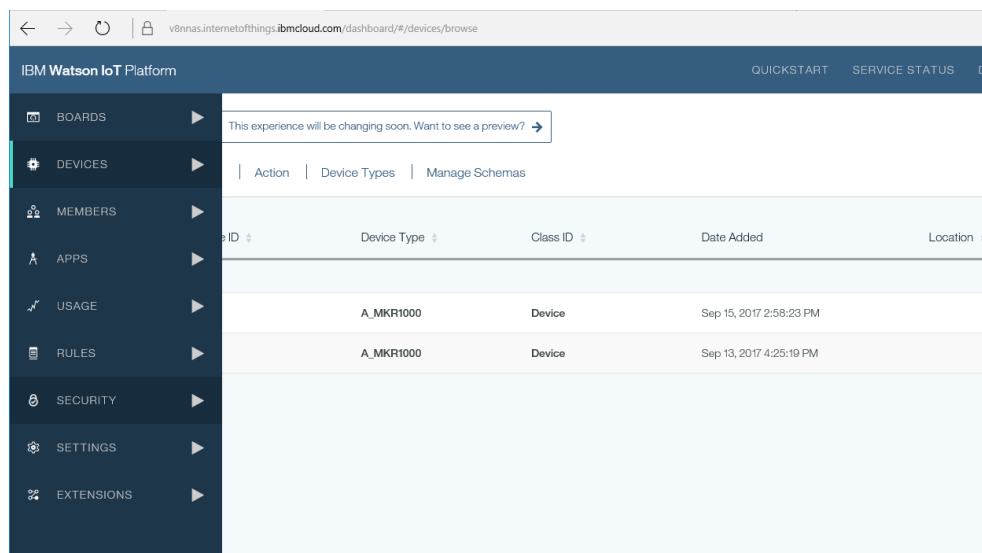
kuva 3.10 Tunnistetiedot.

Loin toisen organisaation ja laitteen 13.9.2017

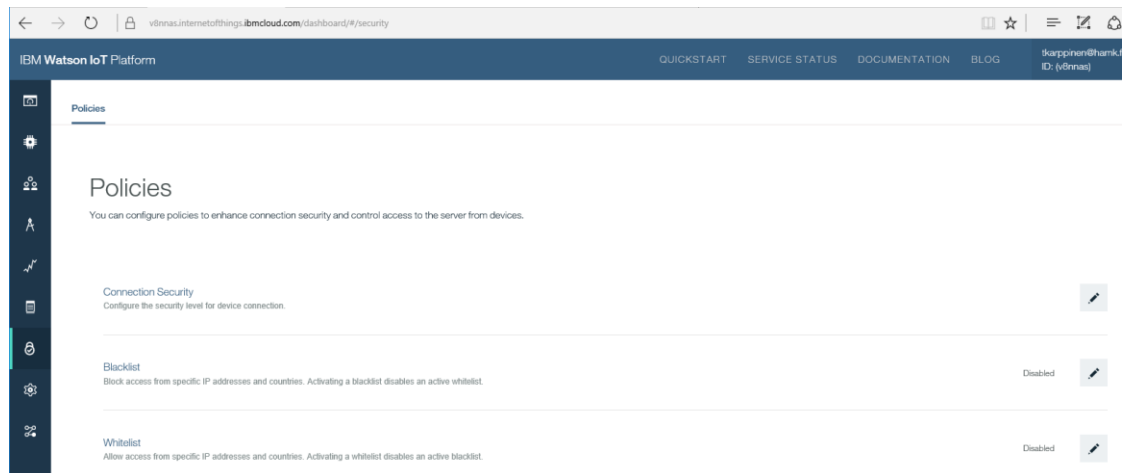


kuva 3.11. Toisen organisaation ja laitteen tunnistetiedot

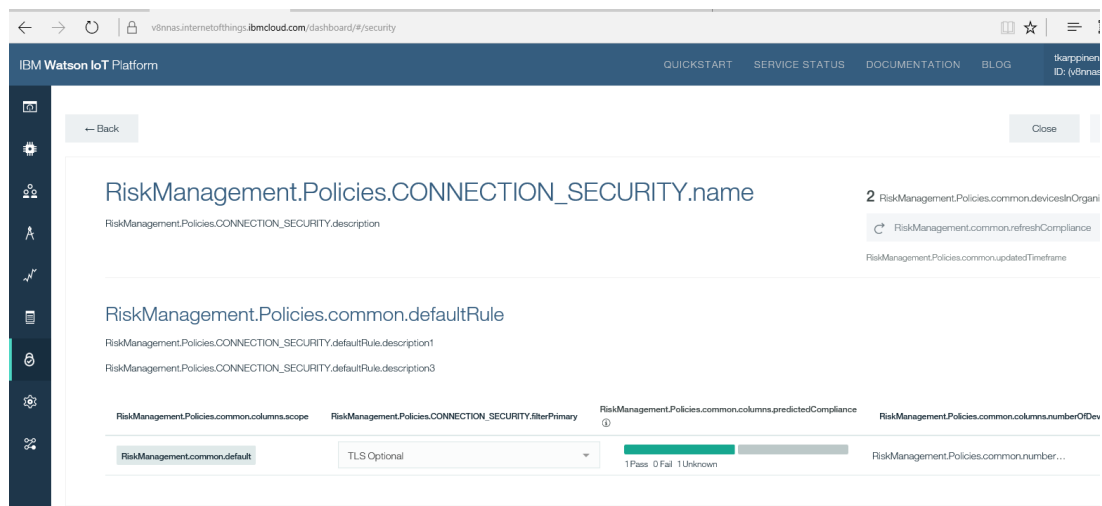
Watson IoT:n turvallisuusasetukset ovat muuttuneet. IBM Watson IoT Dashboard:ssa on muutettava Security –asetuksia.



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



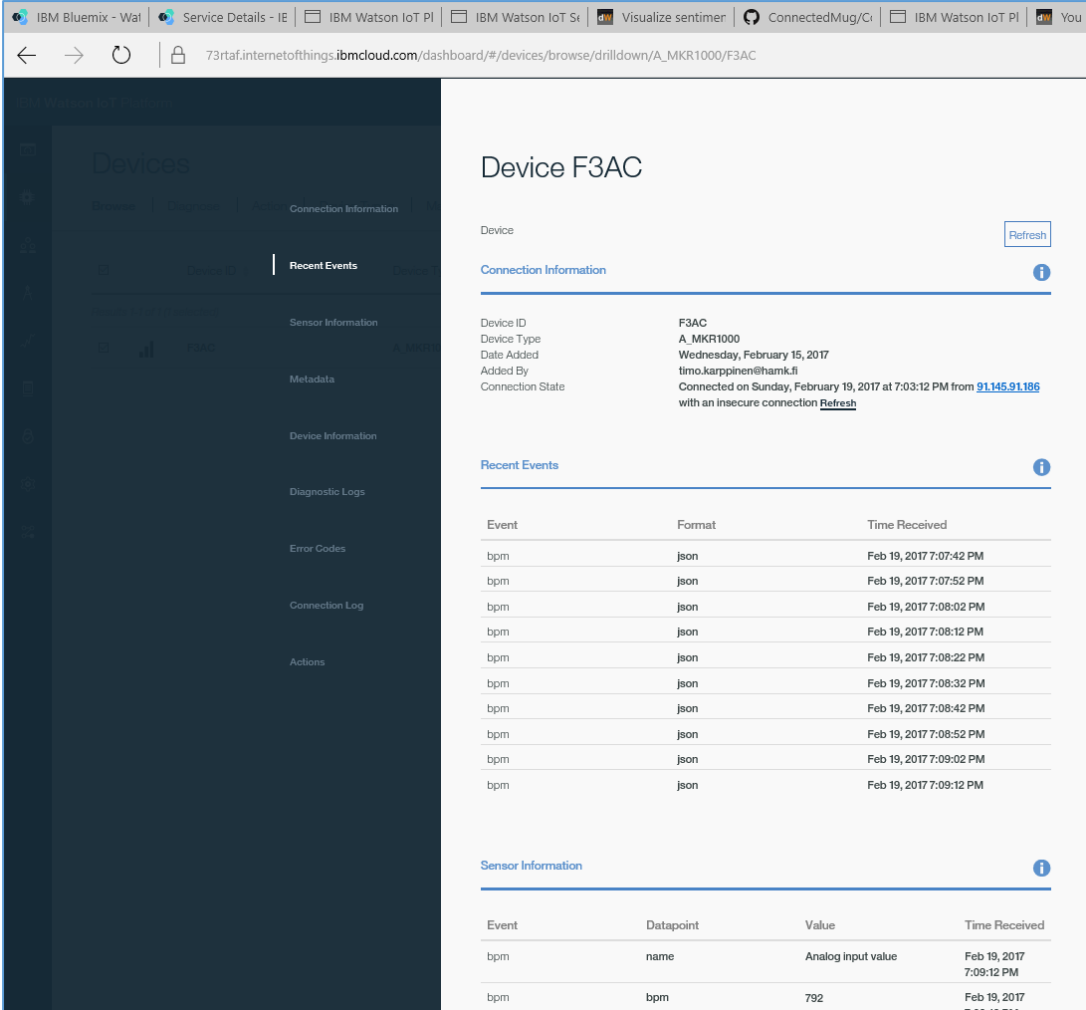
Valitse ”Connection Security” klikkaamalla kynän kuvaa.



Valitse ”...CONNECTION SECURITY...”, ”TLS Optional”

4. Anturitiedot IBM Watson:ssa

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



The screenshot shows the IBM Watson IoT Dashboard interface. The left sidebar contains a navigation menu with options like 'Devices', 'Recent Events', 'Sensor Information', 'Metadata', 'Device Information', 'Diagnostic Logs', 'Error Codes', 'Connection Log', and 'Actions'. The main content area is titled 'Device F3AC' and includes a 'Refresh' button. Below this, there are three sections: 'Connection Information', 'Recent Events', and 'Sensor Information'.

Connection Information

Device ID	F3AC
Device Type	A_MKR1000
Date Added	Wednesday, February 15, 2017
Added By	timo.karppinen@hamk.fi
Connection State	Connected on Sunday, February 19, 2017 at 7:03:12 PM from 91.145.91.186 with an insecure connection Refresh

Recent Events

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

Sensor Information

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

Kuva 4.1 Laitteen lähettämä tieto on näkyvissä IBM Watson IoT, Dashboard, Device -sivulla.

https://73rtaf.internetofthings.ibmcloud.com/dashboard/#/devices/browse/drilldown/A_MKR1000/F3AC

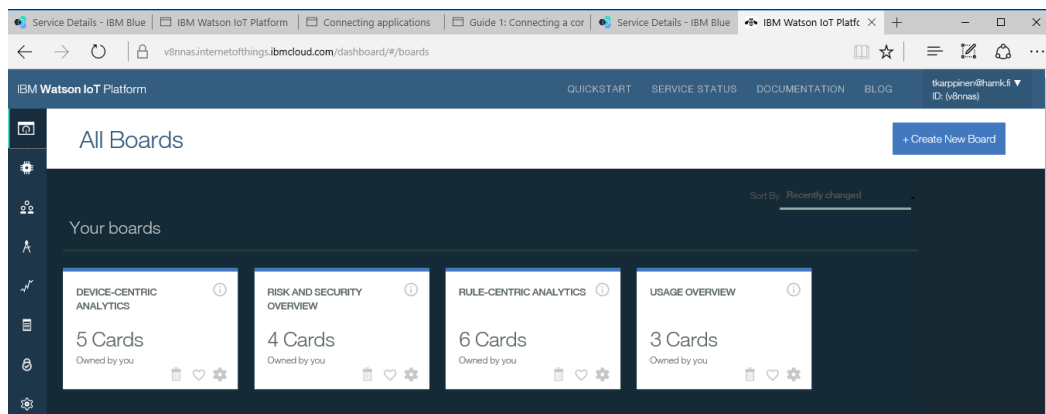
5. Datan näyttö Dash Board-näkymässä IBM Watsonissa

Jatkamme datan analysoinnilla. Alla olevat kuvat ovat IBM Bluemix:stä siinä muodossa kuin sivut aukesivat 27.9.2017 .

Datan analysoinnin aloittamisesta on hyvä ohje IBM Bluemix osassa IBM Bluemix Docs ja edelleen Internet of Things Platform, HOW TO ja valikosta edelleen Getting Started Guides, Guide 1: Connecting a conveyor belt device .

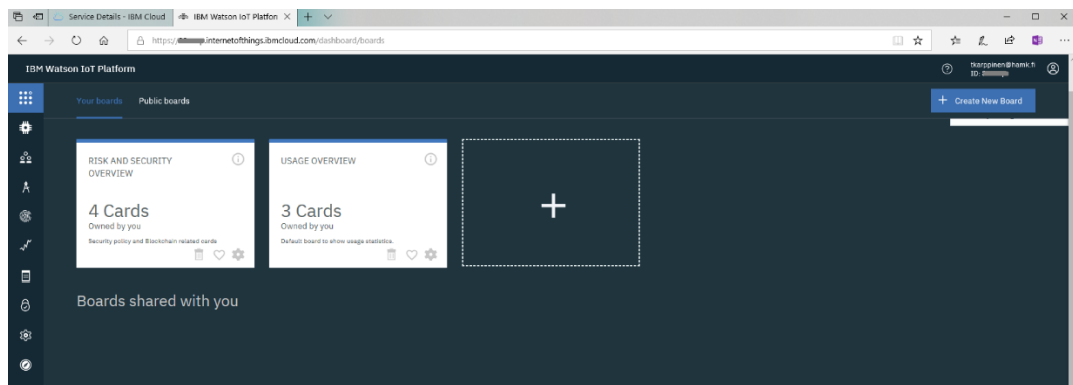
https://console.bluemix.net/docs/services/iot/getting_started/getting-started-iot-conveyor.html#guide-1-connecting-a-conveyor-belt-device

Oletamme, että olet laite-esimerkin ohjeen mukaisesti luonut tunnukset ja kirjautunut IBM Bluemix –palveluun. Valitse IoT Dashboard –näkyssä valikosta vasemmalla Boards.

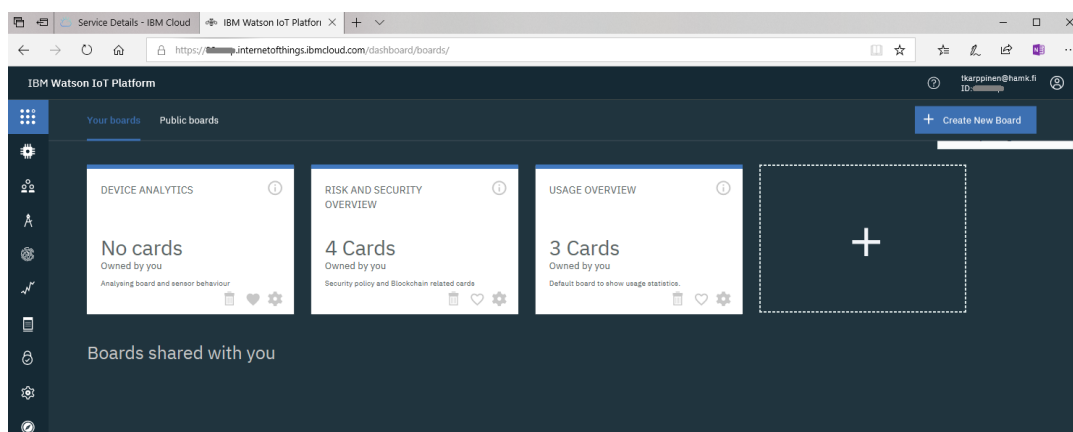


Tästä voit valita Device Centric Analytics.

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

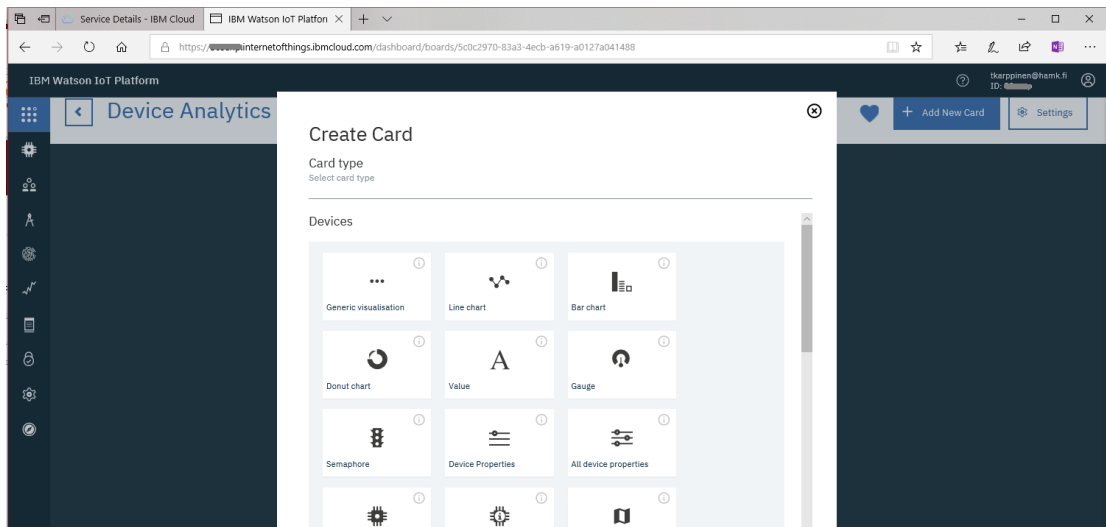


Lisää tähän näkymään uusi Board-kortti. Anna sille nimeksi vaikka Device Analytics.

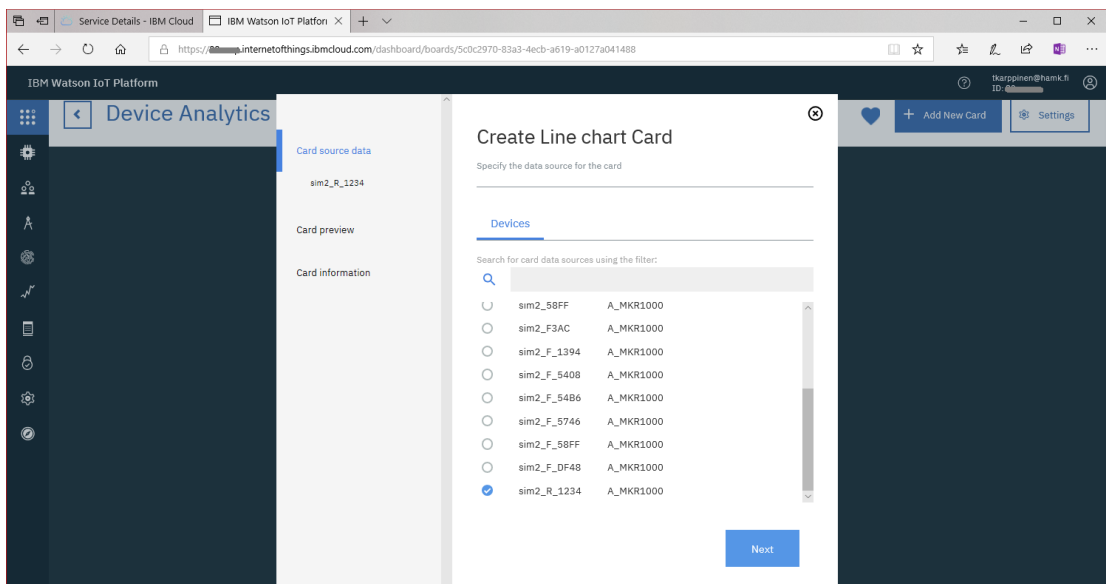


Uusi Board-näkymä on luotu.

Valitse tämä uusi Board ja edelleen Create Card.

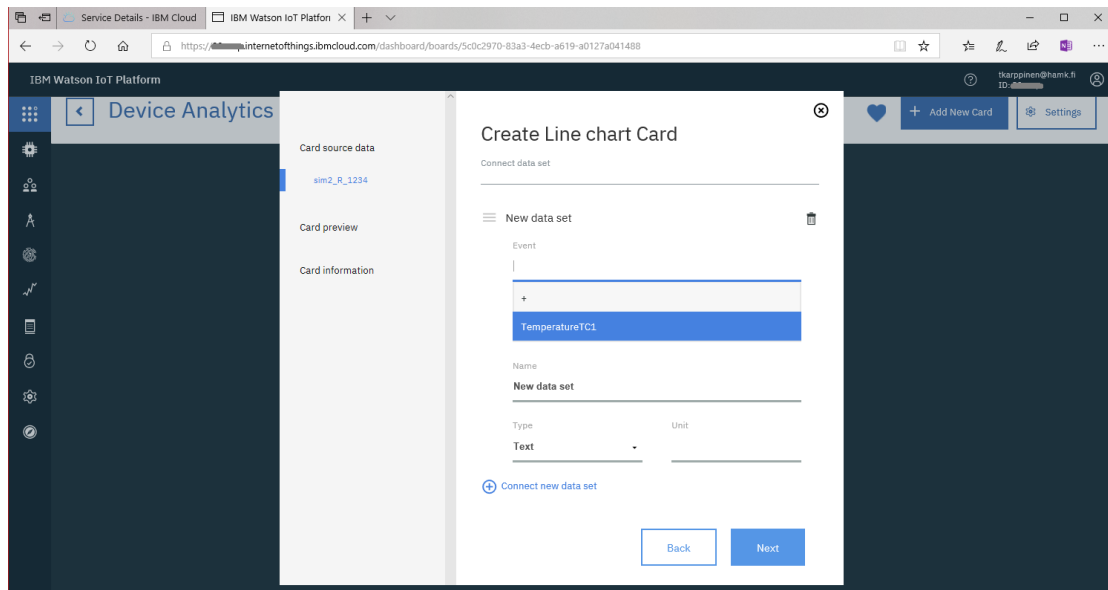


Avautuvasta valikosta valitset Line chart.

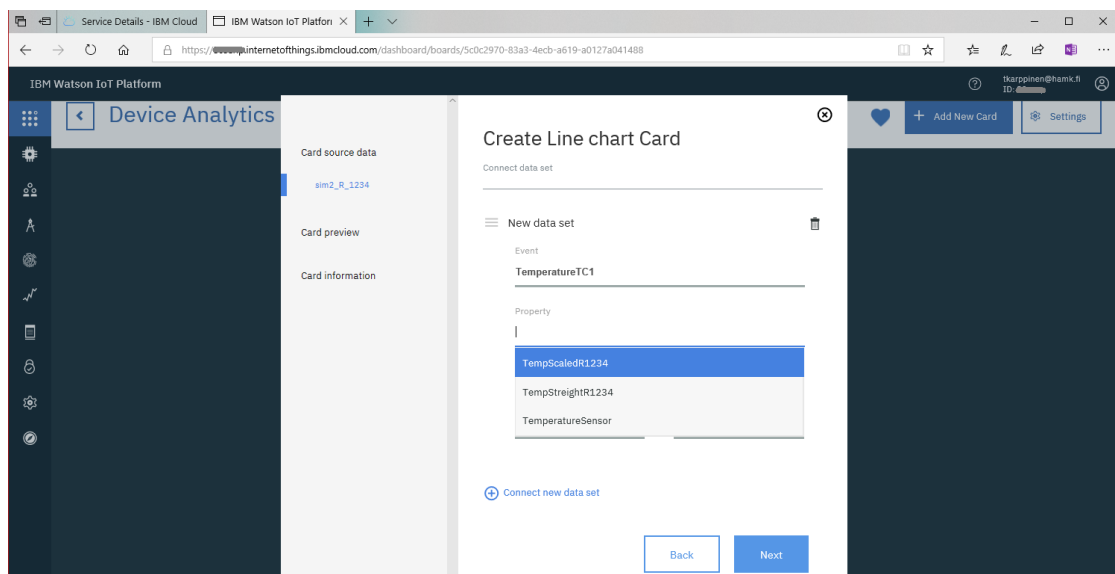


Valitse, minkä laitteesi mittaustietoja haluat tarkastella.

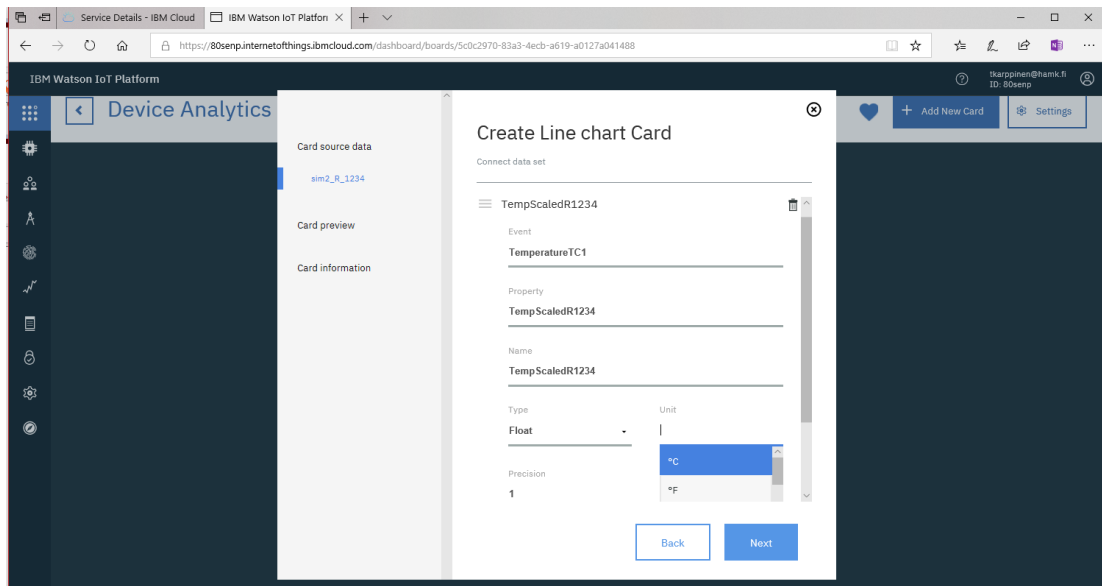
Valitse Connect new dataset. Jos laitteesi lähettää parhaillaan mqtt-protokollan mukaisia laitteesi autentikointitietojen mukaisia viestejä, osaa Watson purkaa sanoman ja ehdottaa sen mukaan oikeita määrittelyjä.



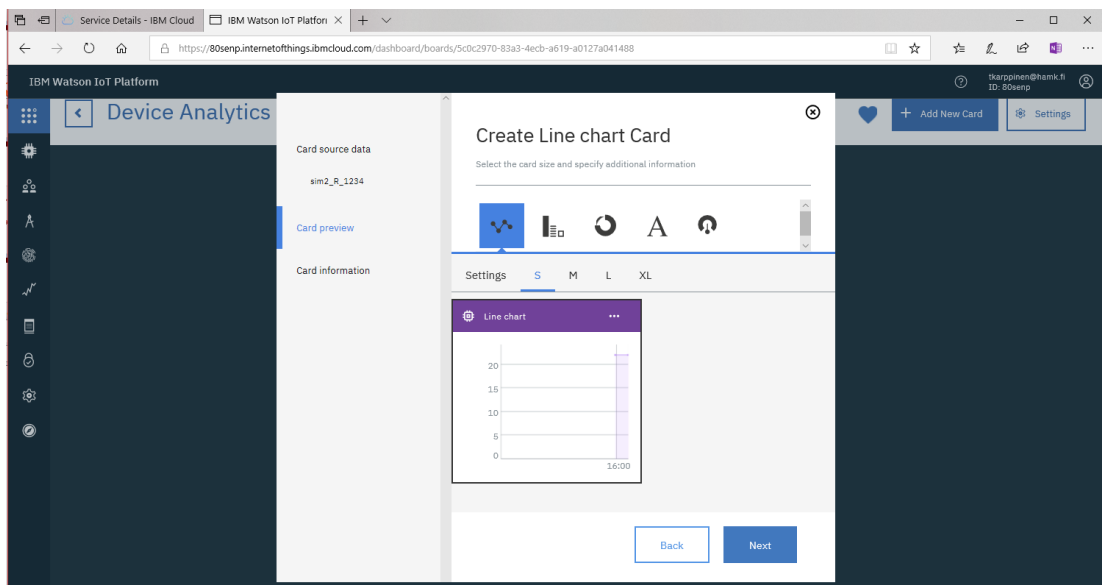
New Data Set event-tyyppiksi poimitaan itse sanomassa määrittämäsi event-nimi.



Valitse Line chart -kaavioon hyvin sopiva numeerinen tieto.

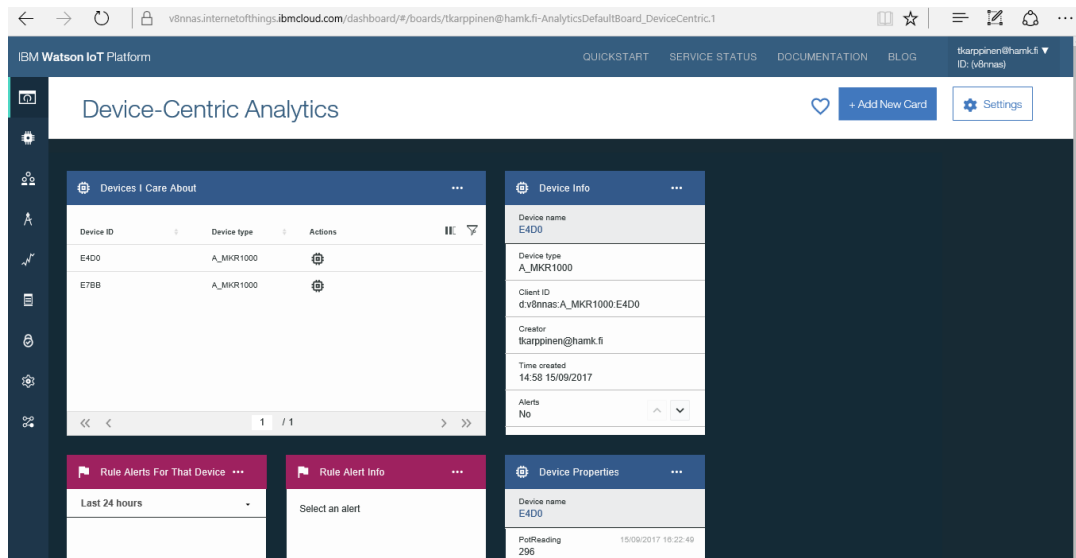


Määritä edeleen numeerille muuttujalle oikea tyyppi ja yksikkö.

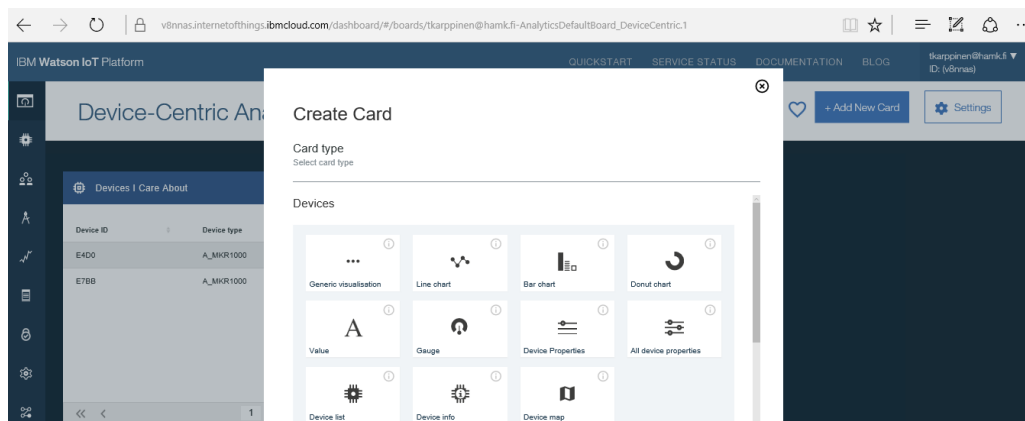


Nyt sinun pitäisi saada näkymä, johon piirtyy reaaliajassa mittausarvoja.

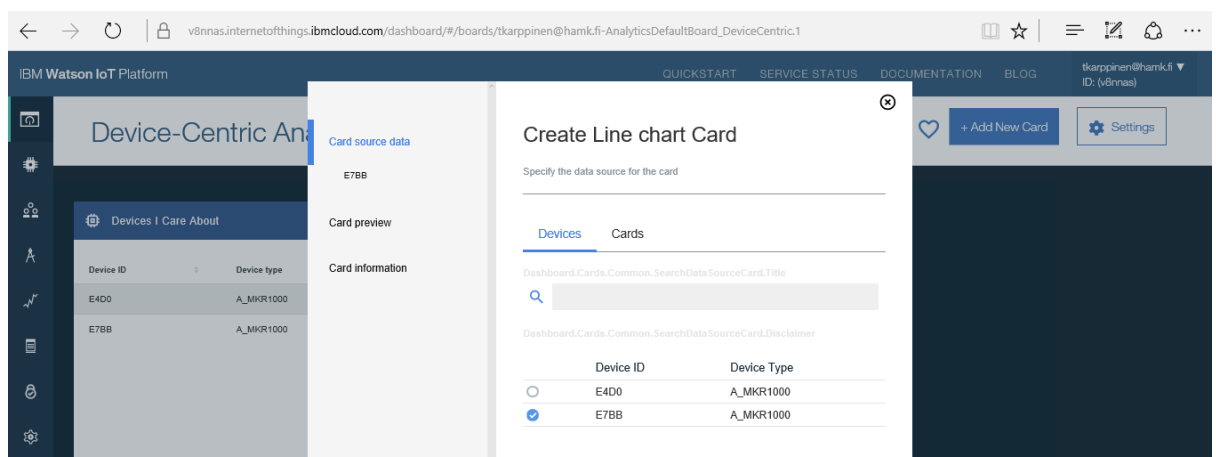
Jos sinulla oli valmiina Board -tyyppinä Device Centric Analytics, oli sinne valmiiksi valittu muutamia Card-tyyppejä. Mutta siihenkin pystyit itse luomaan aivan vastaavasti kuin yllä uuden Line chart -kortin.



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

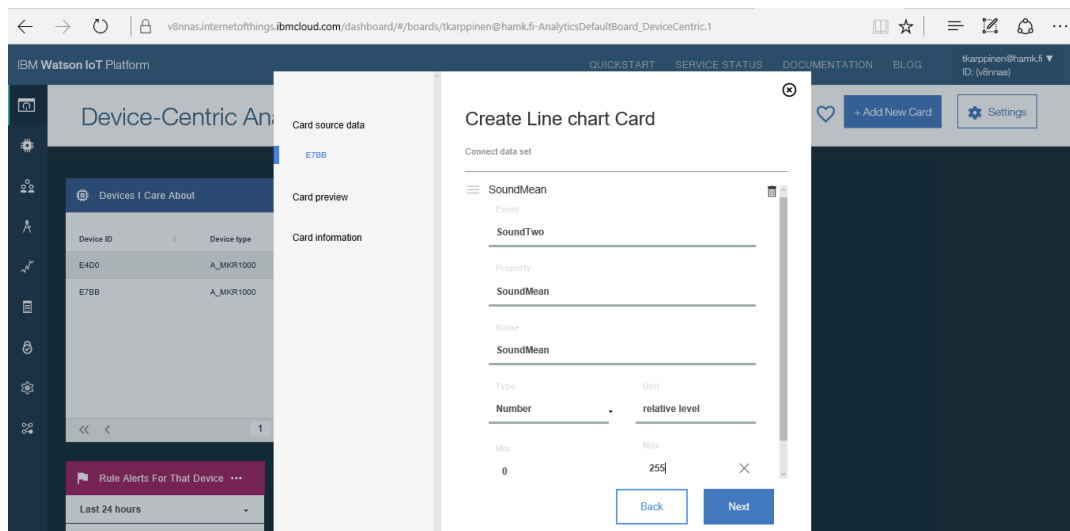


Valitse esitystavaksi Line chart.

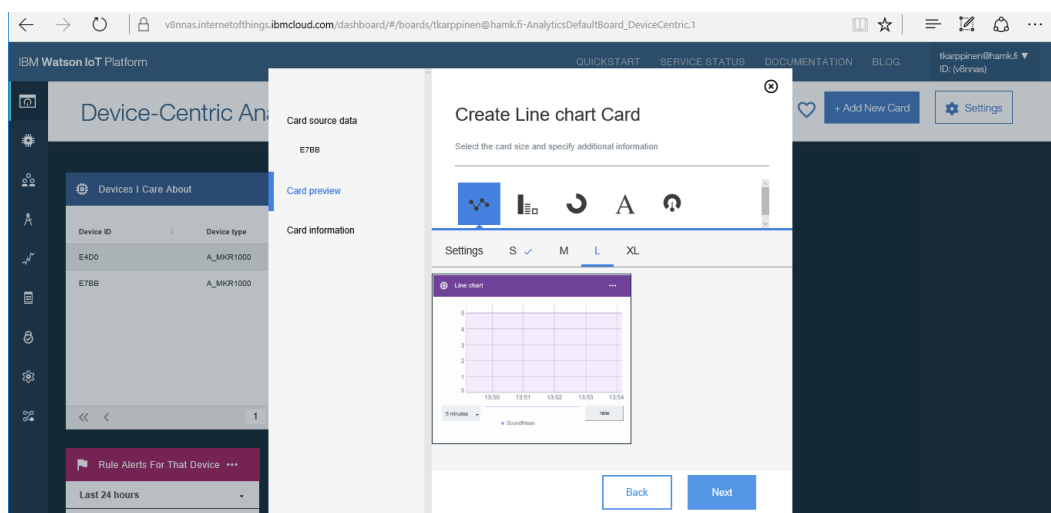


Valitse, miltä laitteeltasi haluat tiedot.

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

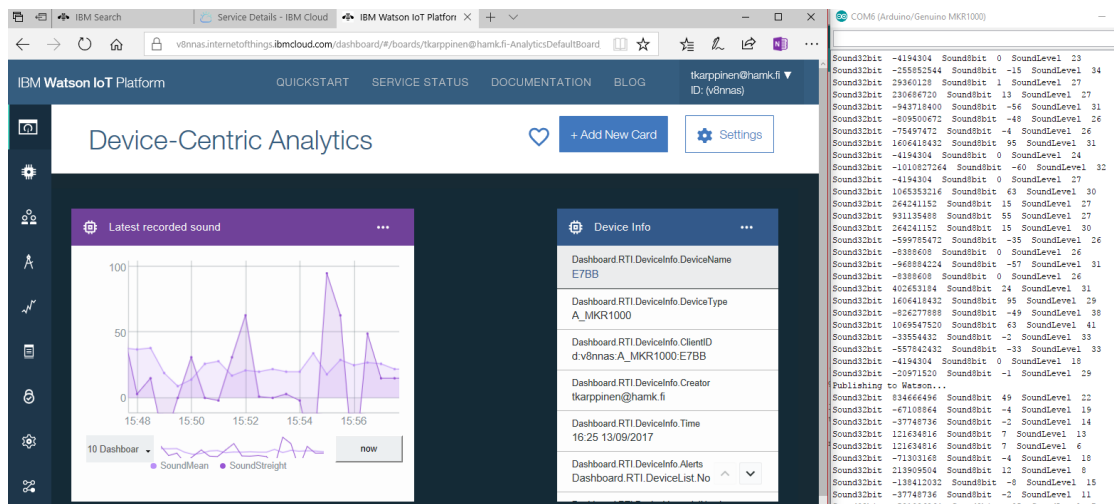


Voit edelleen vaikuttaa esityksen ulkoasuun.



valitse Next ja kirjoita kaaviollesi nimi. Submit-painikkeella avaat kaaviosi.

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

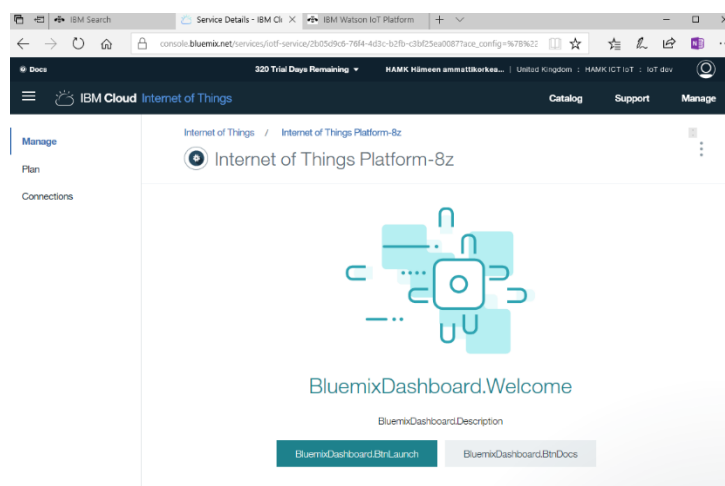


Dashboard – Card -näkömää osaa skaalata näkymän oikealle asteikolle. Se vain kestää jonkin aikaa!

6. Datat analysointi IBM Watsonissa

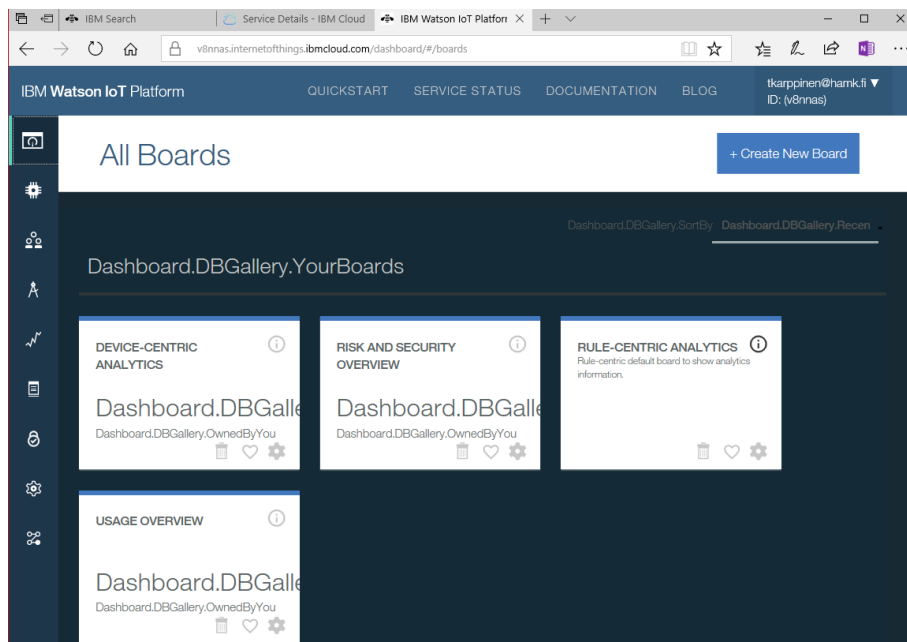
Kirjaudutaan IBM Bluemix -tilille omilla käyttäjätunnuksilla. Oletamme, että tälle käyttäjälle jaetaan Watson IoT:hen tallentuvaa tietoa.

Hakeudutaan sivulle console Internet of Things



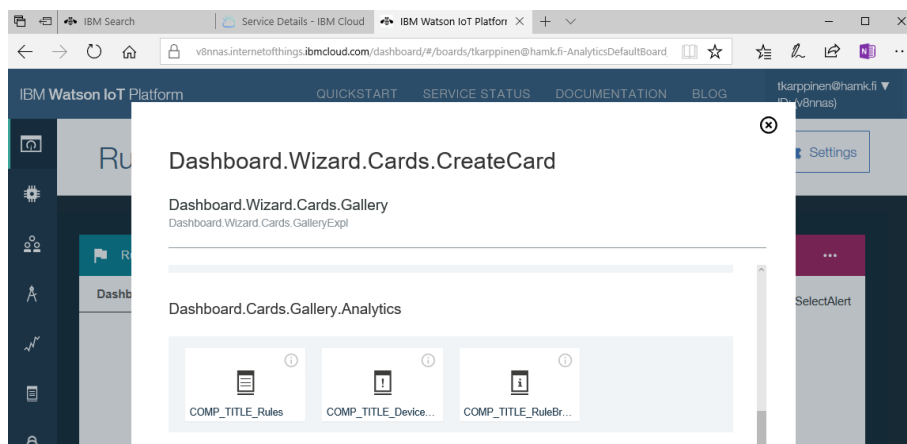
Klikataan painiketta BluemixDashboard.BtnLaunch.

Näkymä siirtyy IBM Watson IoT Platform -näkömäksi kuten kuvassa alla.



Valitaan RULE-CENTRIC ANALYTICS.

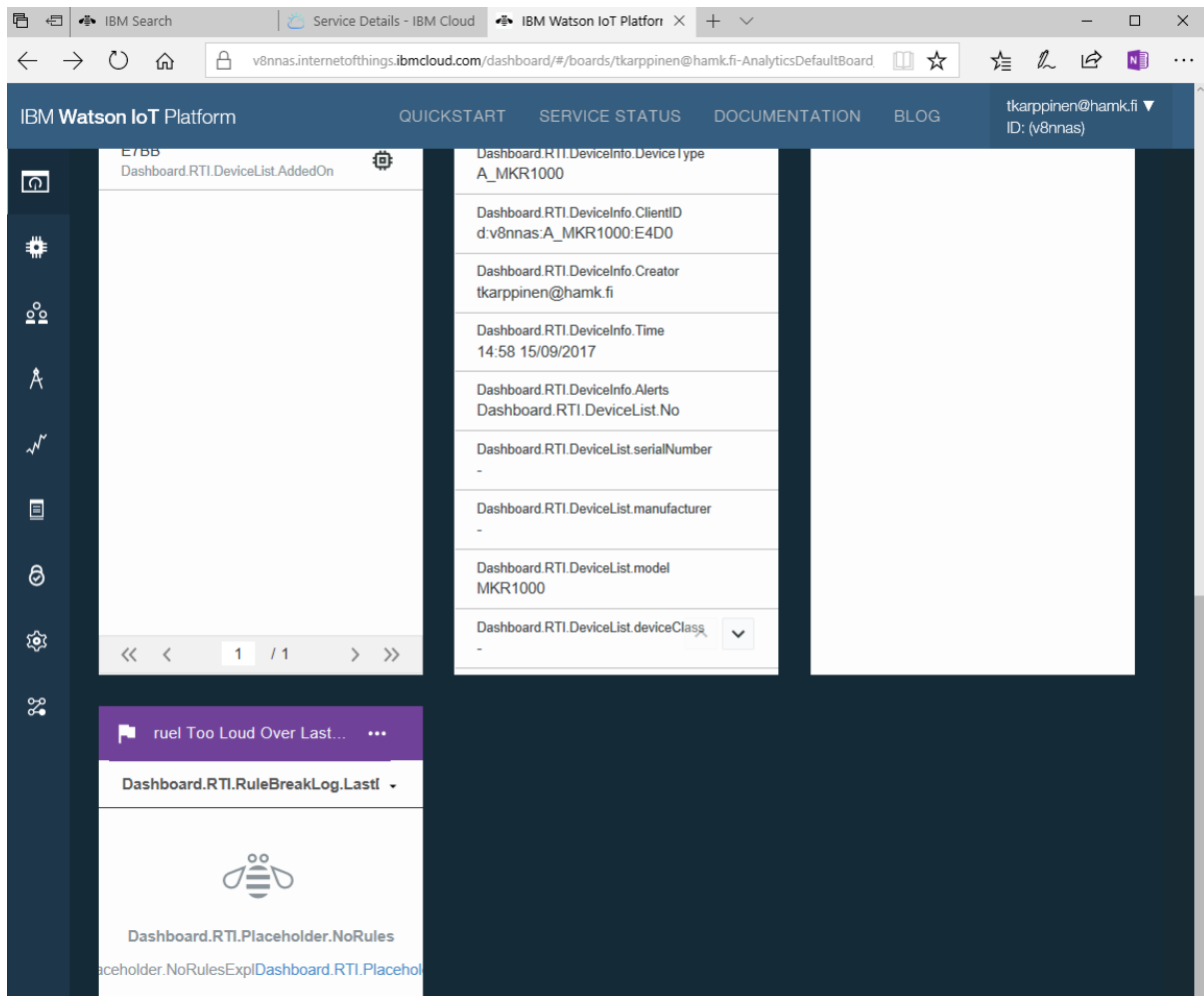
Avautuvassa näkymässä klikataan oikealla ylhäällä Add New Card.



Valitaan COMP_TITLE_RULES

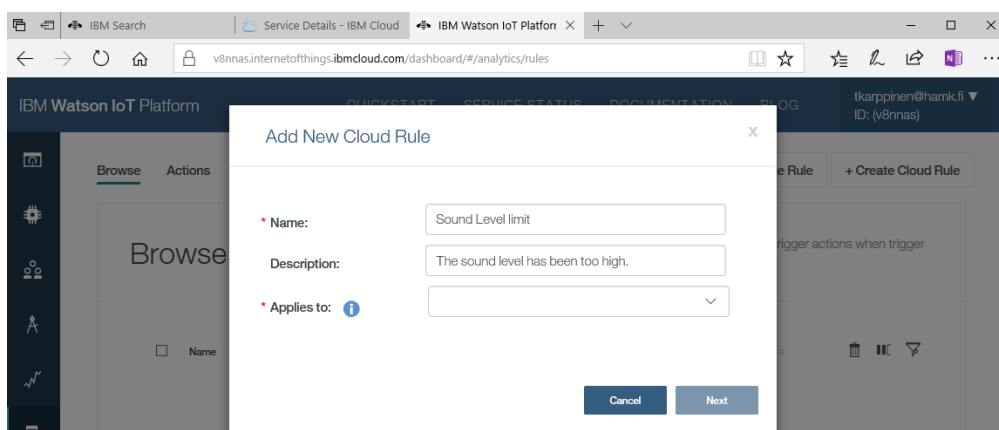
Täytetään kaavake nimeämällä kaavake säännöksi rule Too Loud Over Last Hour.

Sääntö ilmestyy uudeksi sääntökortiksi.



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

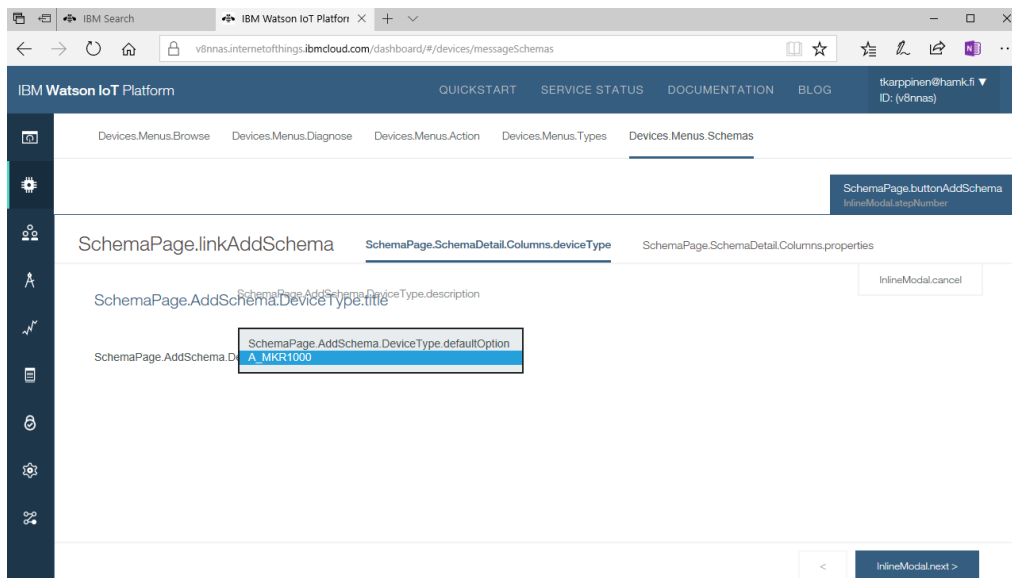
Valitaan avautuvassa näkymässä ylhäältä oikealta +Create Cloud Rule



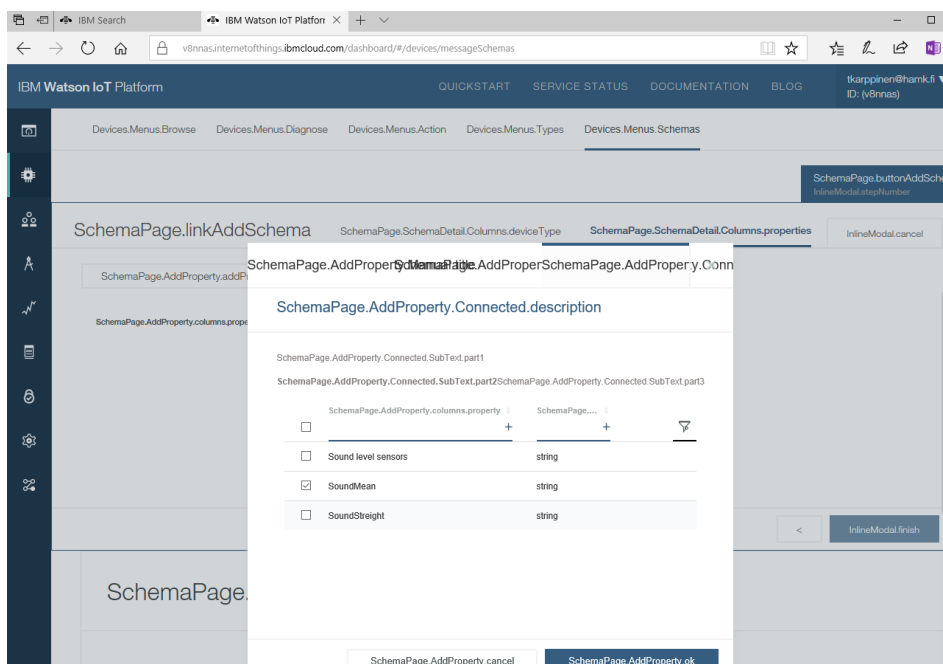
Nimetään sääntö. kuvakkeesta i aukeaa ohje ” Go to Devices > Manage Schemas and add a schema for the device type. “

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

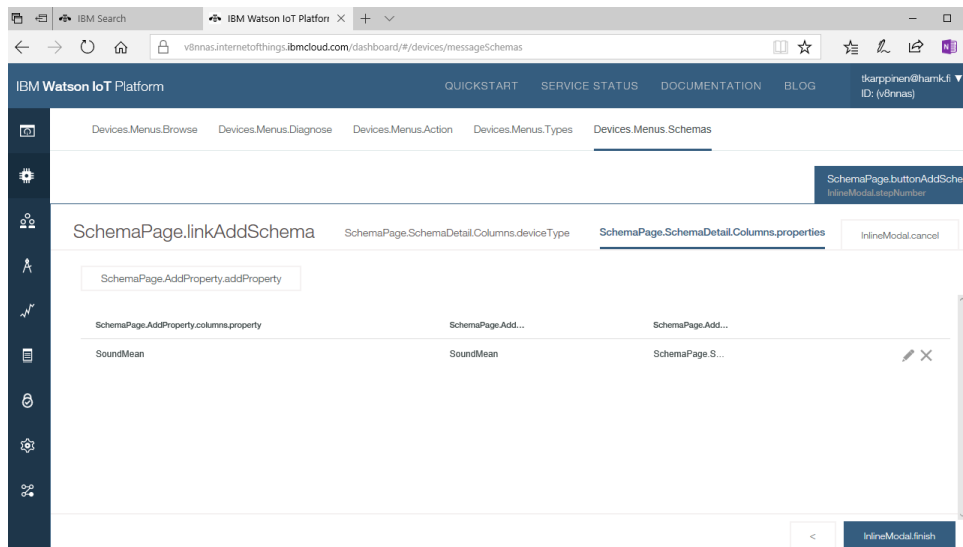
Luot Schema:n laitetyypillesi, esim. A_MKR1000.



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.



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



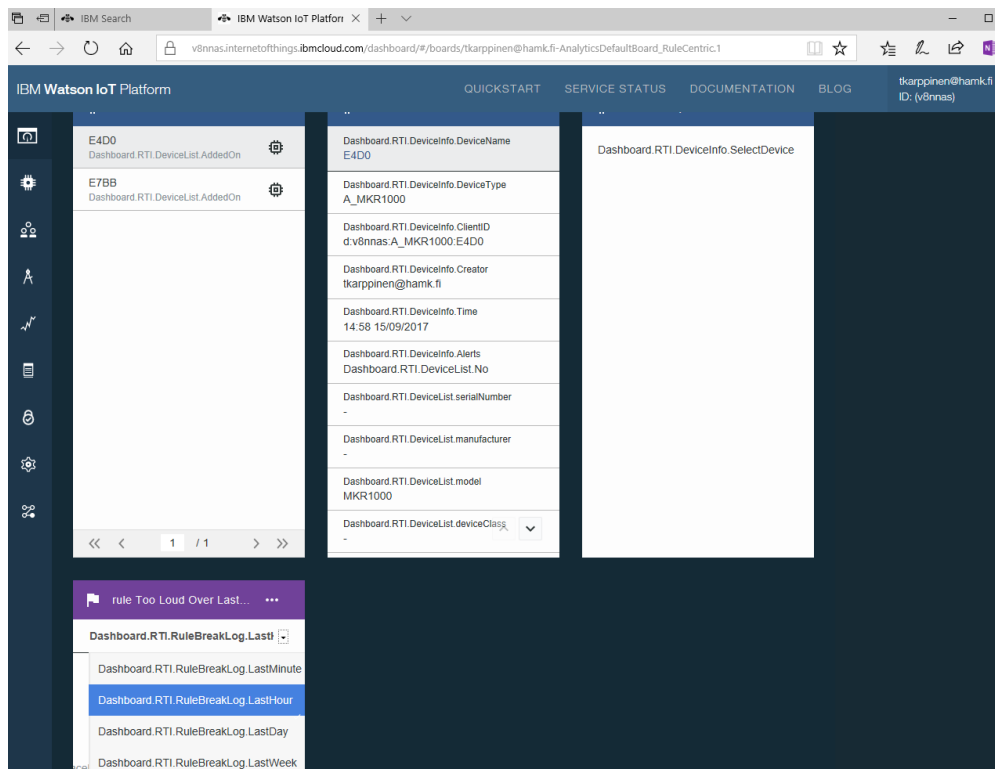
Olet luonut Scheman jolla on ominaisuudet

SchemaPage.SchemaDetailViewPropertis.columns.property = SoundMean

SchemaPage.SchemaDetailViewPropertis.columns.name = SoundMean

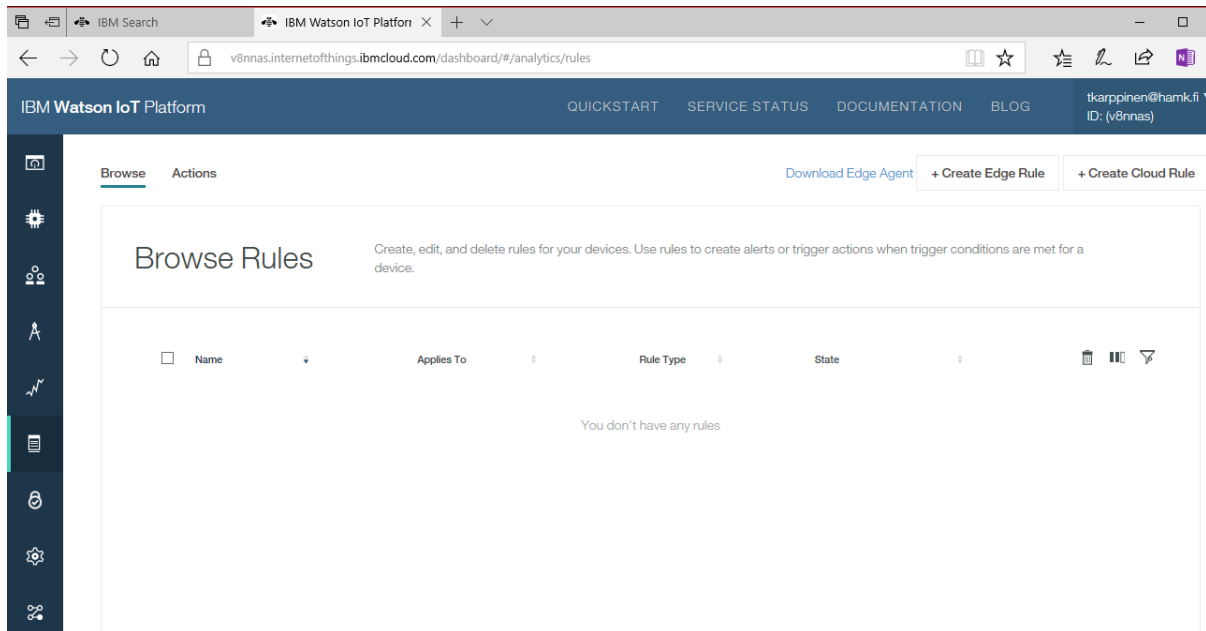
SchemaPage.SchemaDetailViewPropertis.columns.type = SchemaPage.Schemadetail.typestring

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 Dashboard.RTI.Placeholder.

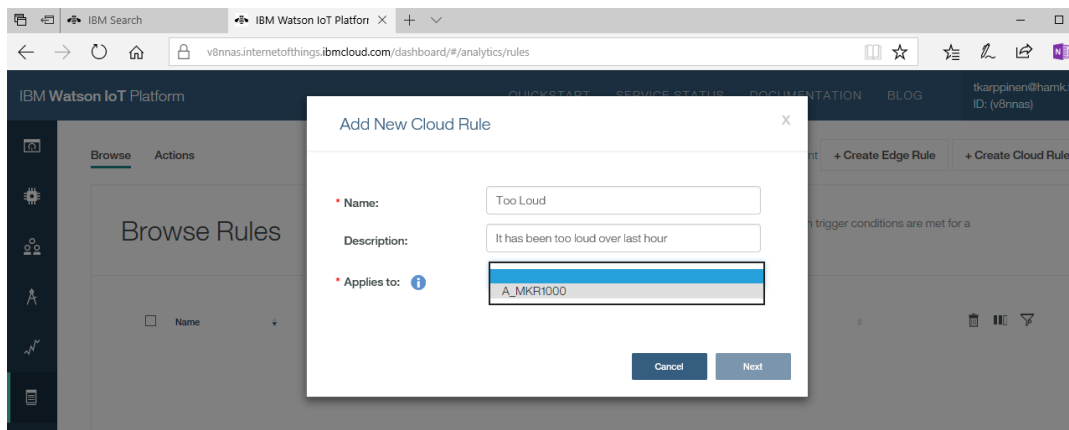


Tai samaan tilanteeseen pääset valitsemalla vasemmalla kehyksessä RULES.

Nyt voit uudelleen valita ylhäältä oikealta +Create CloudRule.

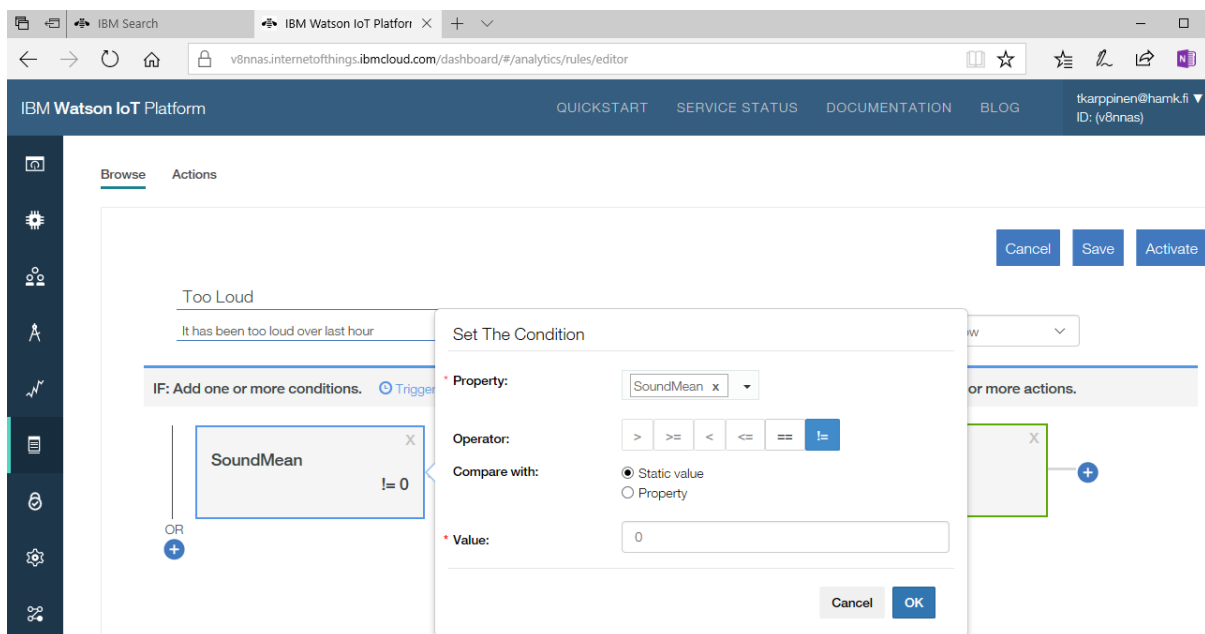
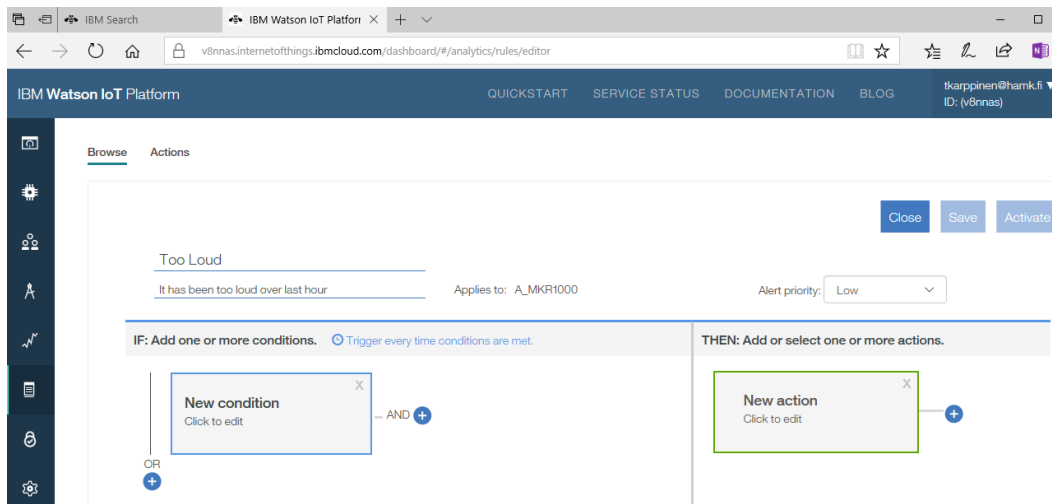


Nyt voit asettaa laitteellesi säännön.



Valitse laitteesi ja Next.

Avautuvassa näkymässä voit aloittaa sääntöjen luonnin.



Ja kappas, valittavan 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.....

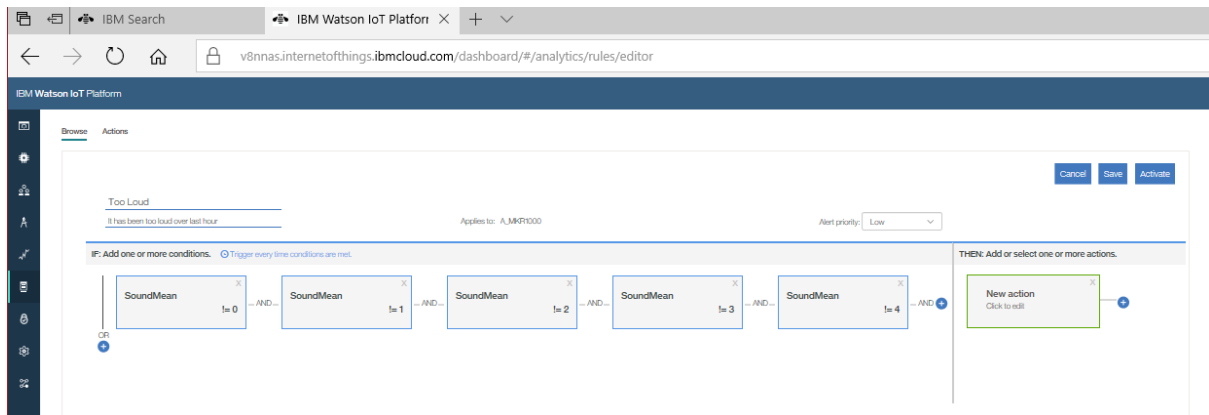
Näin tapahtui, koska client-laite eli tässä tapauksessa Arduino MKR1000 -korttimme oli vahingossa sanoman sisällön määräämällä ohjelmavivillä asetettu lähettämään muuttuja-arvoa muunnettuna teksti-string-muodossa.

Ohjelmavivi, 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);
```

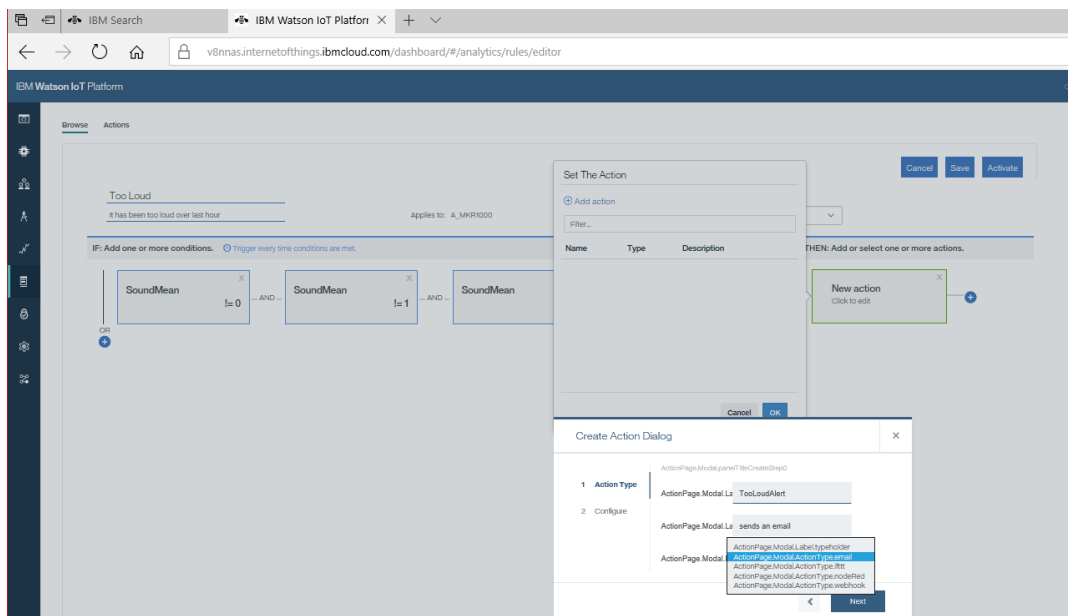
Tämä on siis se oikea muoto, jolla Watson osaa lukea muuttujan arvon desimaalilukuna.



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 .



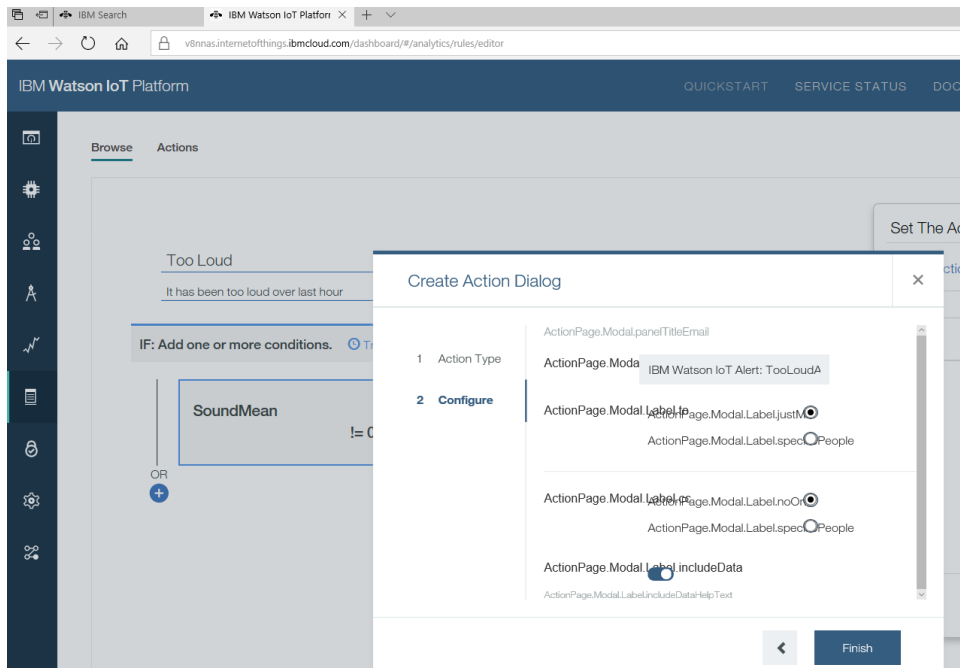
On ehkä turhaa lähettää sähköposti, kun äänenpaineen taso ylittyy. Käytämme tätä kuitenkin ensimmäisenä esimerkkinä. (Voi sopia joihinkin kiinteistönhoidon tehtäviin ;-))

Klikkaa Next ja täytä arvot

ActionPage.Modal.Label.tospecialPeople [sopiva sähköpostiosoite]

ActionPage.Modal.Label.cc noon

Aseta valituksi ...includeData.



Ja klikkaa Finish.

7. Metadatan liittäminen mitattavaan arvoon.

7.1. Anturoitavan kohteen jako perustyyppeihin

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 identifioidun aktiviteettitiedon aina, kun tuotetta käytetään. Rekisteröinnistä tuomme asiakkaan yksilöllisen 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.

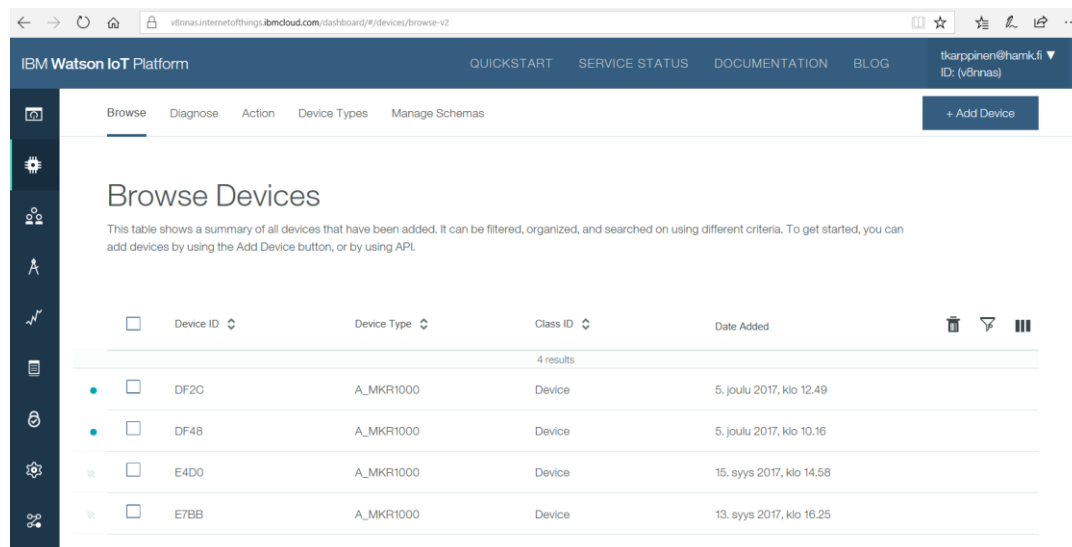
Suurin osa IoT-sovelluksista on ja tulee yhä enenemässä määrin olemaan tämän tyyppin B kaltaisia.

7.2. Metadatan liittäminen mittaustietoon IoT-alustalla

Käytämme tässä jälleen esimerkkinä IBM Bluemix Watson –alustaa.

Oletamme seurattavan kohteen olevan tyyppiä B. Emme kuitenkaan nyt oletta 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.

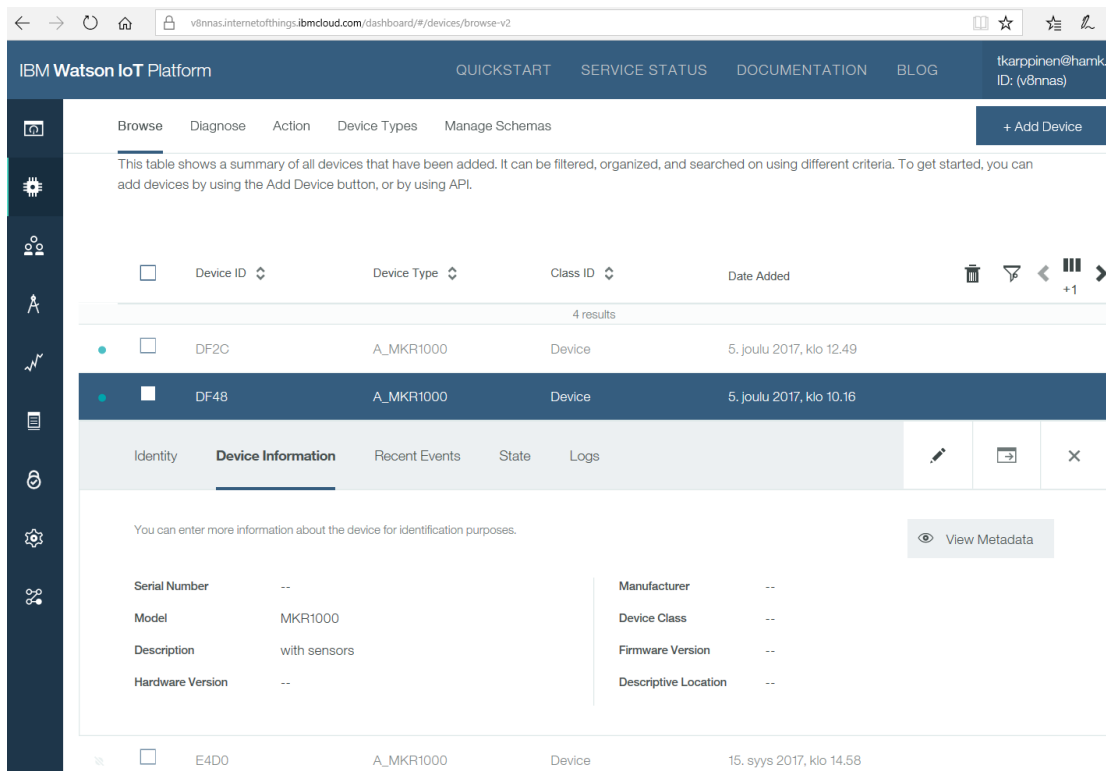


Device ID	Device Type	Class ID	Date Added
DF2G	A_MKR1000	Device	5. joulu 2017, klo 12.49
DF48	A_MKR1000	Device	5. joulu 2017, klo 10.16
E4D0	A_MKR1000	Device	15. syys 2017, klo 14.58
E7BB	A_MKR1000	Device	13. syys 2017, klo 16.25

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 näkyvässä **laitteen riviä** ja edelleen **Device Information**.



The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'QUICKSTART', 'SERVICE STATUS', 'DOCUMENTATION', and 'BLOG'. The user is logged in as 'tkarpinen@hamk.fi' with ID 'v8nnas'. The main content area displays a table of devices. The device DF48 is selected, and the 'Device Information' tab is active. The device details show a Serial Number, Model (MKR1000), Description (with sensors), and Hardware Version.

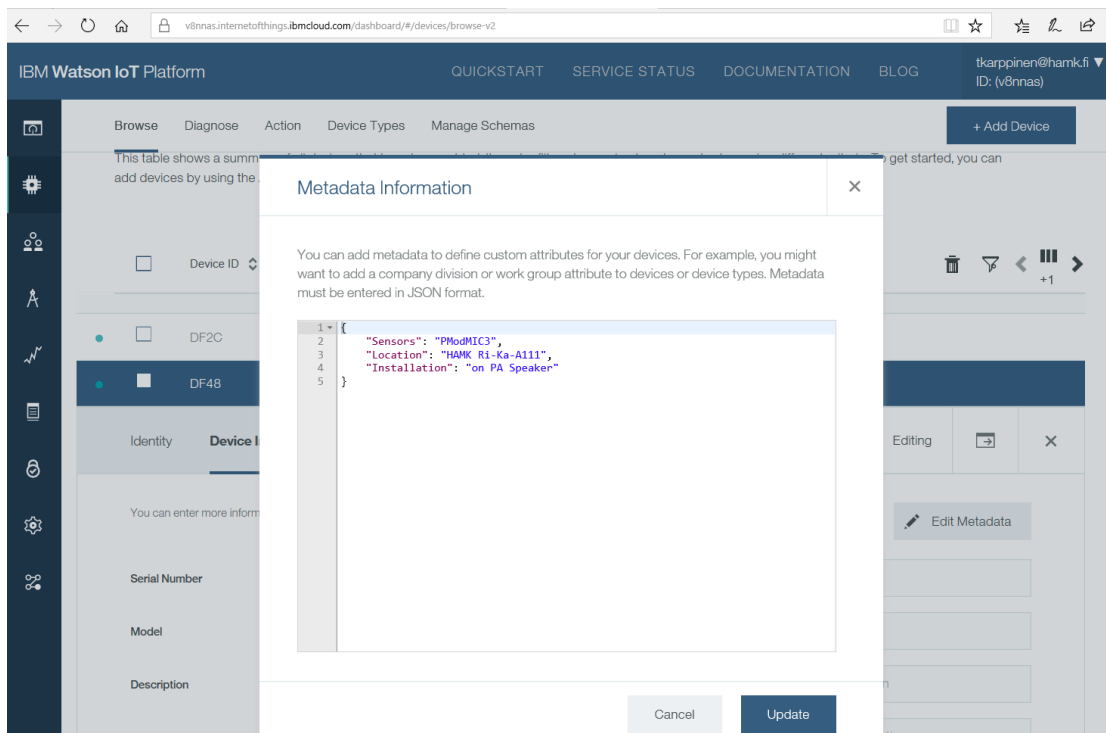
Device ID	Device Type	Class ID	Date Added
DF2C	A_MKR1000	Device	5. joulukuuta 2017, klo 12.49
DF48	A_MKR1000	Device	5. joulukuuta 2017, klo 10.16
E4D0	A_MKR1000	Device	15. syyskuuta 2017, klo 14.58

Device Information for DF48:

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

IoT-laitteelle perustamisvaiheessa kirjattuja tietoja.

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



The screenshot shows the 'Metadata Information' dialog box. The dialog box contains a JSON schema for device metadata, including fields for Sensors, Location, and Installation.

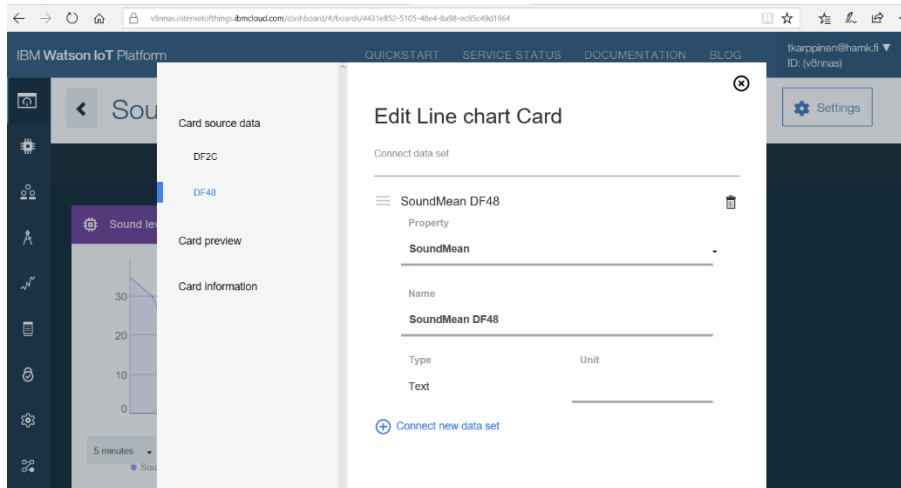
```
1 {
2   "Sensors": "PModMIC3",
3   "Location": "HAMK Ri-Ka-A111",
4   "Installation": "on PA Speaker"
5 }
```

The dialog box also includes a 'Cancel' button and an 'Update' button.

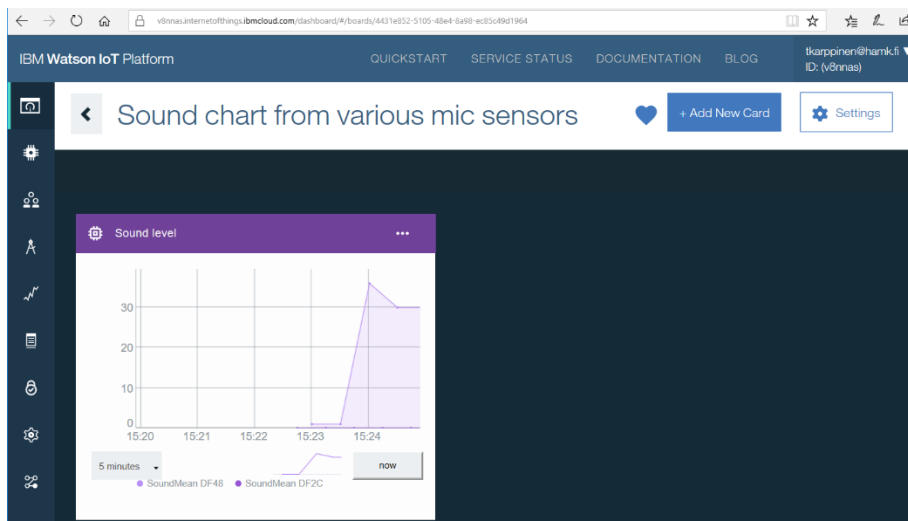
Yksittäisen laitteen Metadatan editointi

Kirjoita metadata Jason-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.



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



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**.