

|  |
| --- |
| Nätverksprogrammering  2021 |
|  |
| 15 januari  STI - IOT19  Författare: Evy Svensson |

Förord

I det här projektet ville jag utmana mig själv och fortsätta utforska protokoll som kan användas i IoT-sammanhang. Bakgrunden till detta är det arbete jag fick utföra på min LIA1 där det blev tydligt att konnektiviteten ibland kan vara en av de största utmaningarna. Därför har jag nu påbörjat en resa med att göra en protokollhub där jag kan testa enheter och protokoll även efter denna kurs som en del i mitt eget utforskande i teknikens fantastiska värld.

Innehåll:

Komponenter…………………………………………………………………………………………………………s.3

Utförande………………………………………………………………………………………………………………s.4

Reflektioner…………………………………………………………………………………………………………..s.7

Komponenter

De komponenter som är listade här vad gäller hårdvara och mjukvara är de jag använt för detta projekt. Allt är inte nödvändigt och en del har jag beslutat att använda p.g.a. preferensskäl utan hänsyn till prestanda.

Hårdvara

RaspberryPi 4, 4GB

Micro SD-kort

Mini-HDMI (till raspberryn)

WemosD1 mini, R1/Esp8266

HP-laptop

Micro USB kabel

Mjukvara

RaspberryPi:

RaspberryPi OS

NodeRed

Node.JS

HP:

Windows10 HomeEdition

VisualStudio2019

VisualMicro (Arduino-PlugIn för att möjliggöra Arduinokod i Visual Studio)

ArduinoIDE (Måste vara installerad för att kunna användas av Visual Studio)

My SQL Workbench

MySQL Connector 8.0

Moln:

Användarkonto på AmazonWebServices

Amazon RDS SQL-databas

Teknologier:

C++

APL

JavaScript

Json

Linux

SQL

AWS

GitHub

SQLConnector 8.0

OTA-Updates

WiFi

Websocket/TCP

HTTP

mDNS

Udp

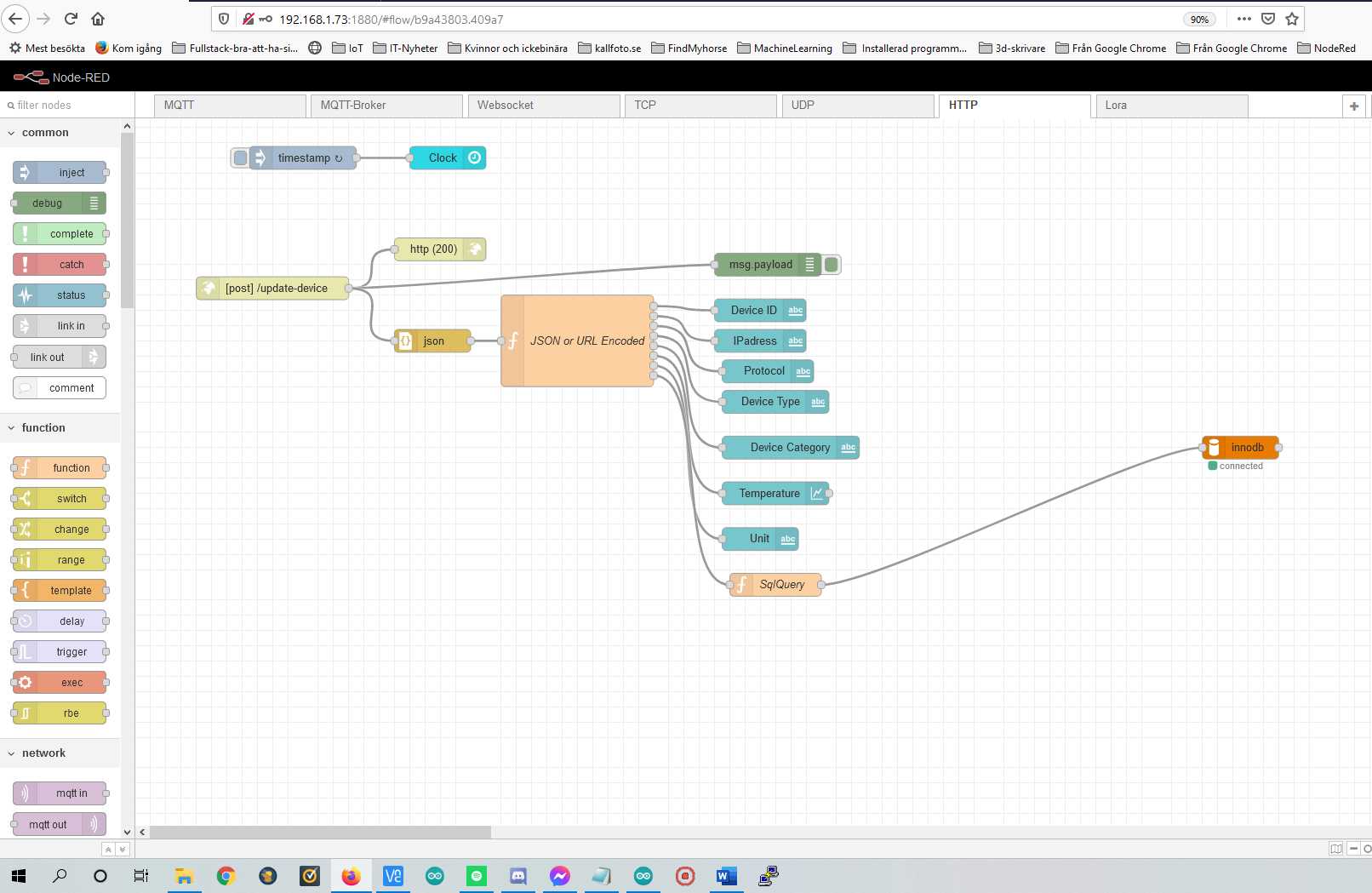
Utförande

RaspberryPi:

Min tanke med NodeRed är att snabbt och enkelt kunna skapa en samlingspunkt för data som skall transporteras enligt olika typer av protokoll och samtidigt ha ett lokalt gränssnitt utan att bygga en hel frontend-applikation.

Jag började med att installera och konfigurera operativsystemet på RaspberryPi och installerade sedan Node.JS och NodeRed samt möjliggjorde för enheten att bli tillgänglig för fjärruppkoppling via SSH och VNC-Viewer så att jag kunde styra allt via min laptop istället för att byta HDMI-anslutning då det sliter mycket på den fysiska hårdvaran. I Raspberryns Terminalfönster angav jag’ hostname -I’ för att ta reda på vilken IP-adress enheten hade som jag skulle ansluta till. Därefter skrev jag in ’Node-Red’ i terminalen för att starta NodeRed och komma åt den instansen från en annan enhet i mitt nätverk genom att gå in på en webbläsare och skriva i ’192.168.1.73:1880’.

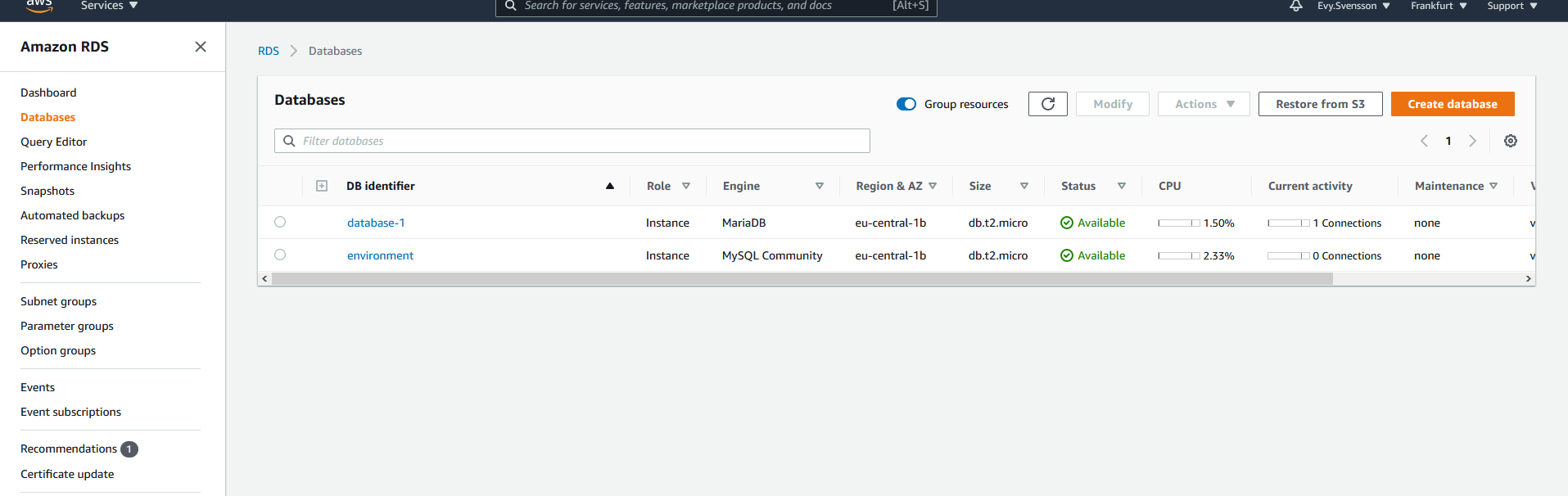
*NodeRed:*

**

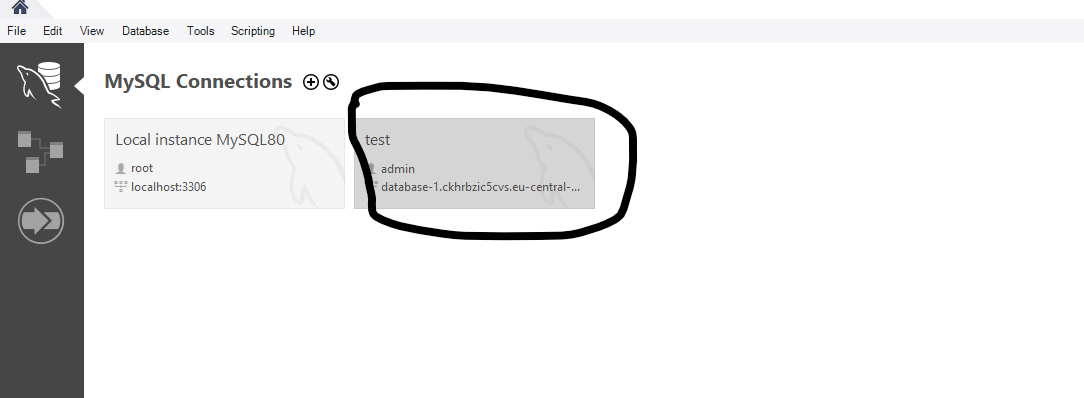
AWS/SQL:

Jag loggade in på mitt gratiskonto och skapade en Maria DB-databas enligt deras installationsguide. Jag laddade ner My SQL Workbench samt MySQLConnector 8.0 och skapade en anslutning till den endpoint som hörde till databasen. I min databas lade jag till en tabell med kolumner som kunde representera relevant data i en verklig situation.

*Amazon RDS:*



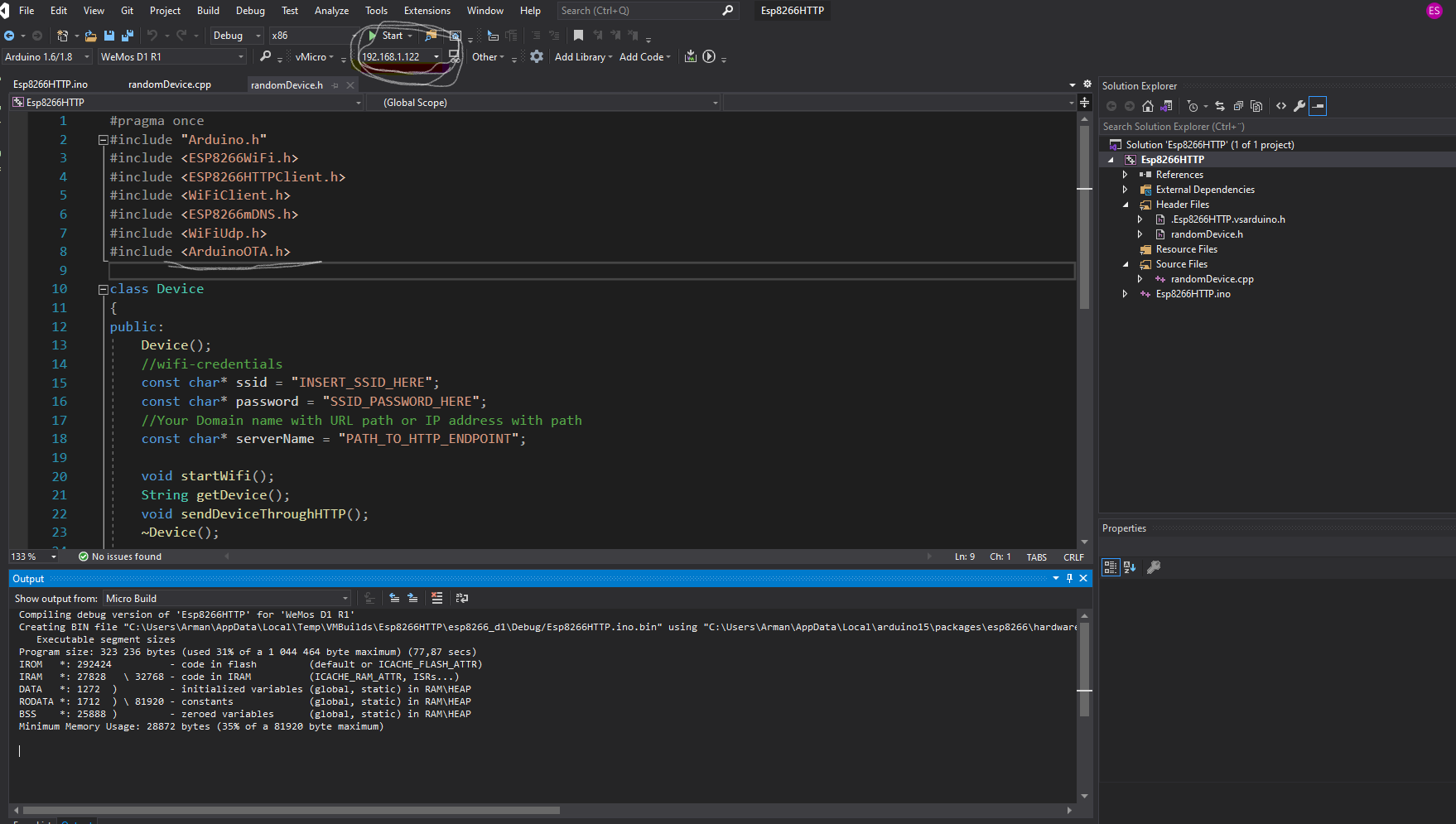
*Anslutning till Aws RDS databas:*



Arduino:

Jag bestämde mig för att jag ville ha möjligheten att uppdatera mikrokontrollern med ny kod utan att ha den ansluten via en USB-kabel till min laptop då detta som tidigare nämnts, sliter onödigt mycket på den fysiska hårdvaran. Genom att ladda upp en sketch från biblioteket ”ArduinoOTA” förbereddes enheten för OTA över WiFi och jag kunde ladda upp nya sketcher via en virtuell nätverksport kopplat till dess IP-adress. ArduinosIDE hittar denna tack vare biblioteken ”ESP8266mDNS.h” (Multicast DNS-protokoll) och ”WiFiUdp.h” (User Datagram Protocol). Jag bestämde mig också för att den här enheten skulle vara dedikerat till modern API-teknik genom att använda http-protokoll för att göra en POST med en sträng formaterad som ett Json-objekt där en httpIN-node i NodeRed får agera endpointadress, en funktion i JavaScript för att dela upp den inkommande payloaden till egna värden som kan användas för att bygga ett lokalt UI men här också används för att skicka vidare meddelandet till en SQL-node som skickar ’INSERT INTO…’ tillsammans med värdena till en Maria DB-databas som jag skapat i AWS.

*Nätverksport, Arduino OTA:*

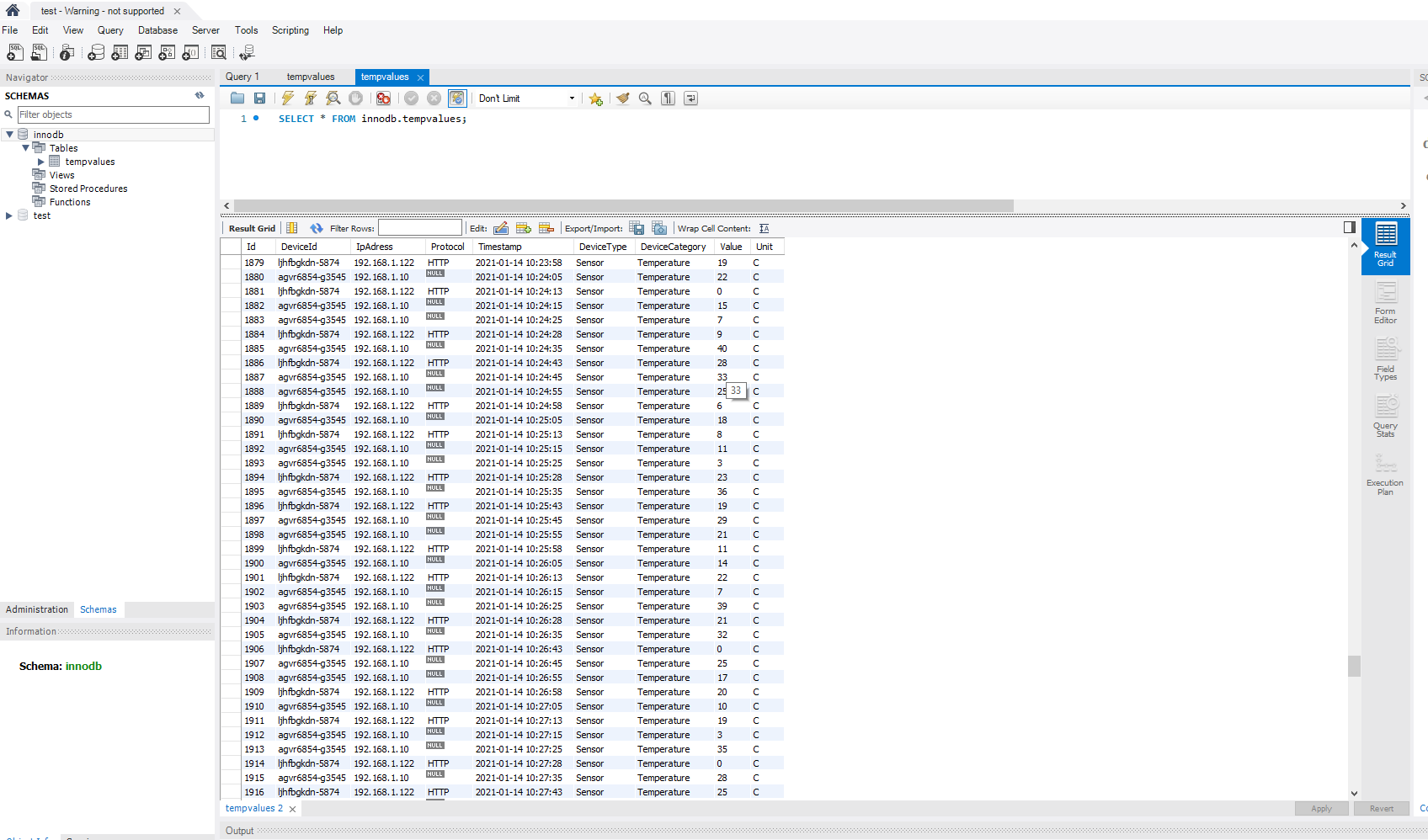


Laptop:

I Visual Studio öppnade jag ett nytt projekt och la till filsökvägarna för My SQL Connector 8.0 till ”Additional Library Dependencies” med flera i projektinställningarna som står beskrivet i dokumentationen på GitHub.

Tidigare i kursen hade vi i uppgift att programmera en websocketserver och klient i C++ där klienten skickar information i en sträng till servern för att omvandla om den inkommande strängen till SQL-format och skicka det till en MySQL-databas som kördes lokalt på datorn. För att simulera en extra extern enhet så använde jag mig av samma program för att skicka data direkt från servern till min databas i AWS för att skapa en variation i inkommande data och simulera en verklig situation där flera enheter skickar in olika enhetsnamn och värden vilket skapar en bättre grund för aggregerade data, fler urvalsmöjligheter och en bättre grund för att kunna exportera delar av databasen till en instans med machinelearning.

*Mer varierade data i databasen:*

**

Reflektioner

Förutom att jag har lärt mig integrera många teknologier i ett och samma projekt så har jag också hittat ett arbetssätt som möjliggör detta under kontrollerade former genom att behandla ett protokoll i taget.

Vägen framåt för projektet blir att lägga till fler noder under protokollflikarna i NodeRed, näst på tur är LoRa som jag redan börjat forska i där jag har kopplat en CubeCell HTCC-AB01 mikrokontroller som skickar data via en LoRaWan gateway till TheThingsNetwork och en nod i NodeRed hämtar den uppskickade datan i TTN.

Detta projekt kommer att hänga med länge framöver.

Bonus: Har tack vare det här projektet gjort mitt livs allra första Youtube-video för min egen skull som beskriver delarna i det här projektet då jag ibland har lite svårt att komma ihåg vad och varför jag gjort saker och utforskar snabba, effektiva medel för dokumentation.

<https://youtu.be/V_4ikL6q_5o>