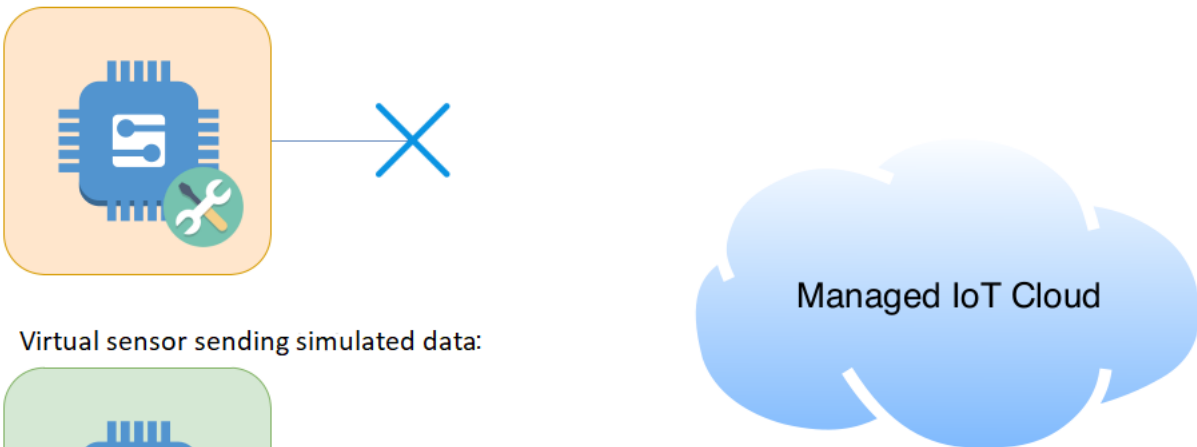


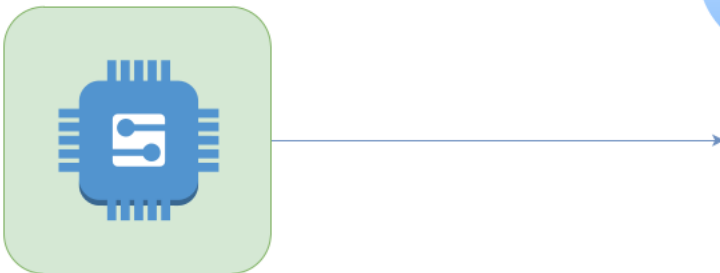
Simulated Data in Managed IoT Cloud

This document describes how to easily set up a Thing (sensor) in Telenor's cloud solution called Managed IoT Cloud (MIC), and how to send simulated data to your dashboard. By simulating data streams you can start designing your visual dashboard and explore all the possibilities MIC offers before you have made any physical IoT devices:

Real sensor that is not yet ready:



Virtual sensor sending simulated data:



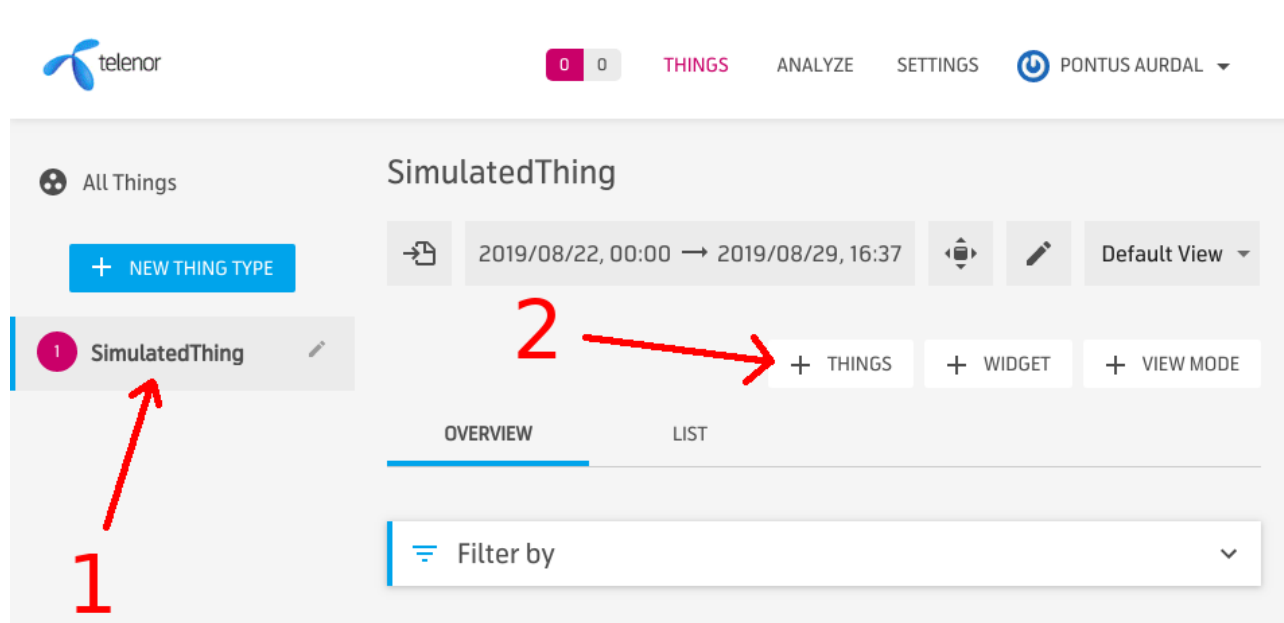
What you'll need:

- An account in MIC
- Node.js needs to be installed on your computer. This can be done by following the steps as described here (Windows): <https://www.guru99.com/download-install-node-js.html>
- Script to represent the simulated Thing (attached)

So, how can the included script take the role as a Thing and send data from something that doesn't yet exist? It's quite simple; we create a Thing in MIC and download the unique keys (certificates). We then configure the script to use these keys and run the script. It's that simple ☺

Create a new Thing and Download the Keys

Login to your account in MIC. Click on the desired Thing Type (1) that you'll find in the left-side menu and then click the "+ THINGS" button (2):



A popup-menu will appear. Choose 1 as the number of things to add (1) and pick a domain that has been assigned to your organization (2). Then, click "CREATE" to create the new Thing (3):

Add new things for: SimulatedThing

Create batch



Number of things to add

1

Domain

min.organisasjon

Things are created in batch and an Event notification is created when finished.

Cancel

CREATE

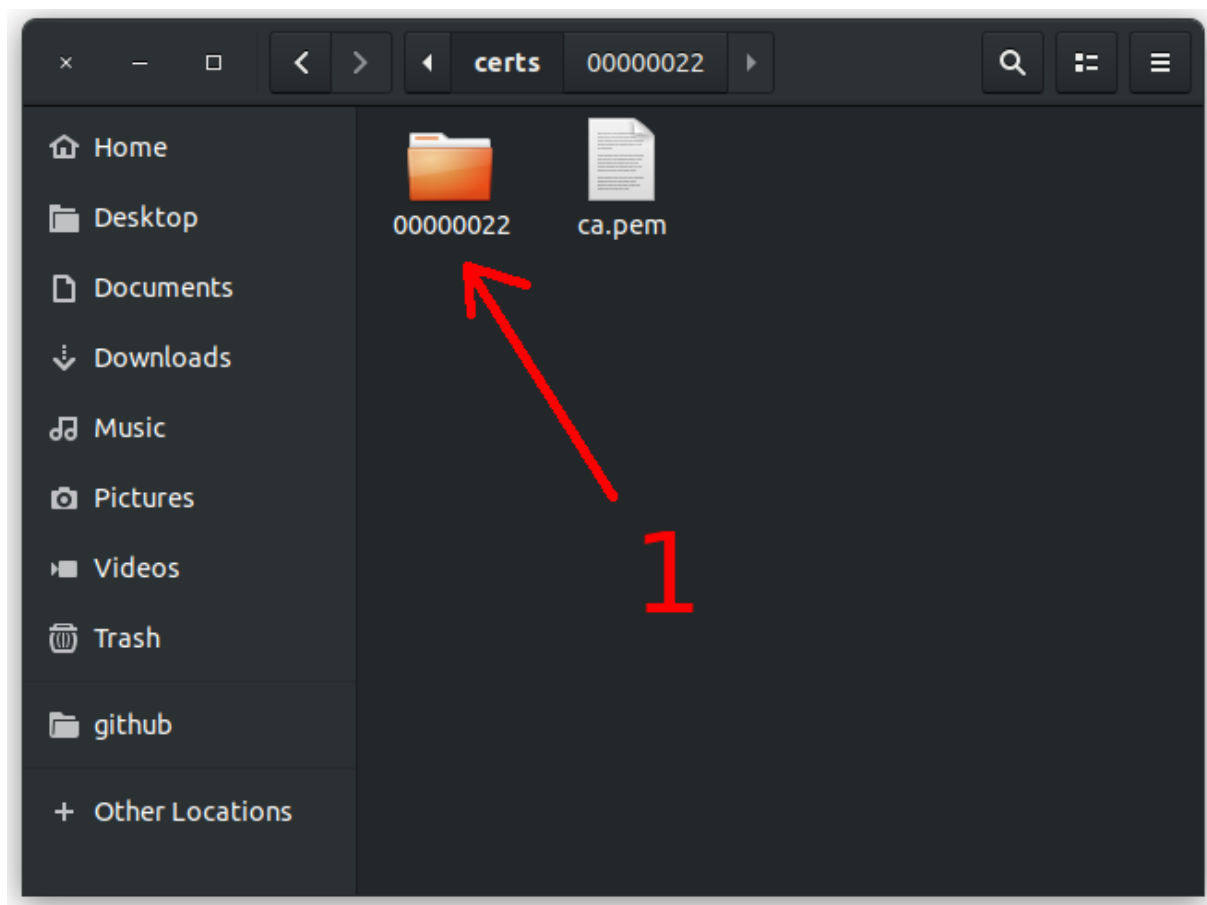
We are now ready to download the keys! If you click the “LIST” tab you’ll get a list of all the created Things for this Thing Type (1). If you hover your newly created Thing you’ll see a download button that you can use to download the keys (2):

The screenshot shows the Telenor Things web interface. At the top, the Telenor logo is on the left, and navigation links for 'THINGS', 'ANALYZE', 'SETTINGS', and 'PONTUS AURDAL' are on the right. A sidebar on the left contains 'All Things' and a 'NEW THING TYPE' button. The main content area is titled 'SimulatedThing' and has two tabs: 'OVERVIEW' and 'LIST'. A red arrow labeled '1' points to the 'LIST' tab. Below the tabs is a 'Filter by' dropdown menu. A table titled '1 thing' displays a single entry with columns: 'Thing', 'Domain', 'Last Heard', 'Created', and 'Status'. The entry has the value '00000022' for Thing, 'Gjøvik Kom...' for Domain, and '2 hours' for Last Heard. A red arrow labeled '2' points to a blue button with a download icon in the row's action column. At the bottom, there is a pagination bar showing 'Rows per page: 10', 'Page 1/1', and a 'Download certificate' button.

Thing ↑	Domain	Last Heard	Created	Status
00000022	Gjøvik Kom...	2 hours	[Download icon]	[Edit icon] [Delete icon]

Configure the Script to Use the New Keys

When we download the keys from MIC we get a ZIP-archive that needs to be uncompressed. Use a ZIP-compatible program to uncompress the archive (e.g. 7zip). If you open the archive you'll find another folder with a name similar to this: **000000XX**. This folder has the same name as the virtual Thing in MIC, and contains the keys that you'll use. Copy this folder into the "certs" folder that is located in the attached script folder (1):



We now need to tell the attached script to use these keys. Open "index.js" and modify the variable called "THING_NAME". Change it to the same value as the name of the folder that we copied in the previous step. The script will now know where to look for the correct keys to be used:

```
7  /**
8   * Simuler data fra tingen med dette navnet
9   * i MIC.
10  *
11  * NB: Hvis dette byttes må man laste ned
12  * sertifikat-filer (nøkler) for den nye tingen
13  * og plassere dem i /certs -mappen likt det
14  * allerede er blitt gjort.
15  */
16  var THING_NAME = '00000022';
```

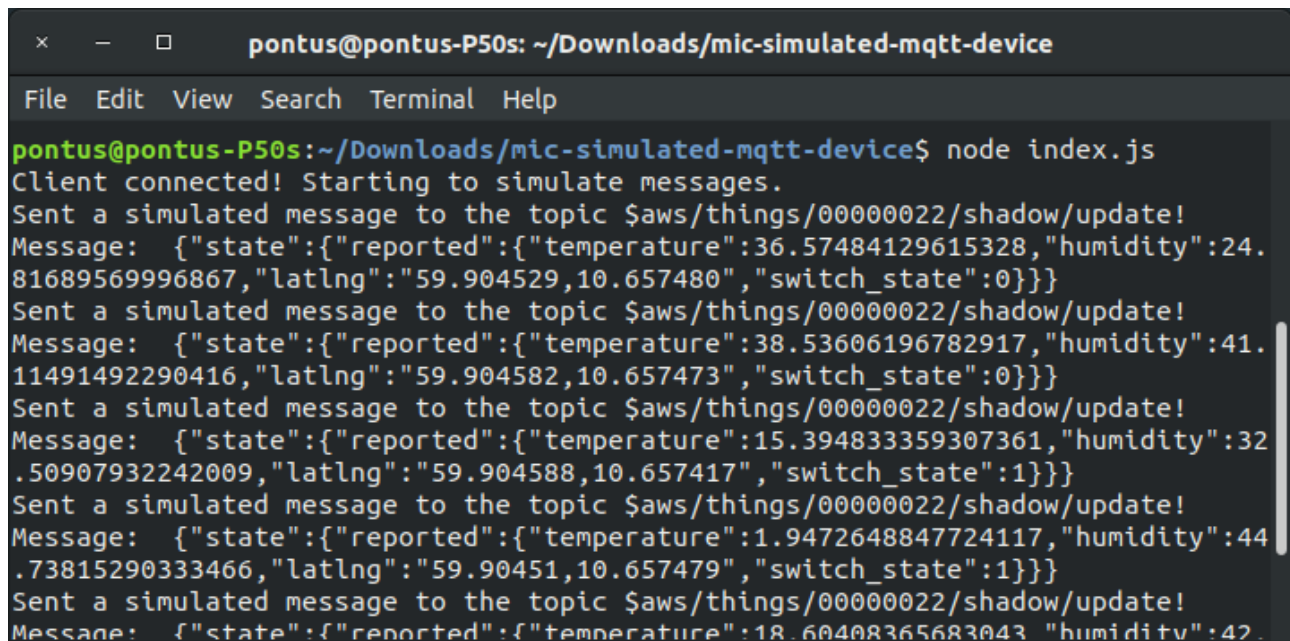
Run the Script

The script is written in Node.js and requires that Node.js is installed on the computer.

There are many websites that describes how to install it on your computer. E.g. this

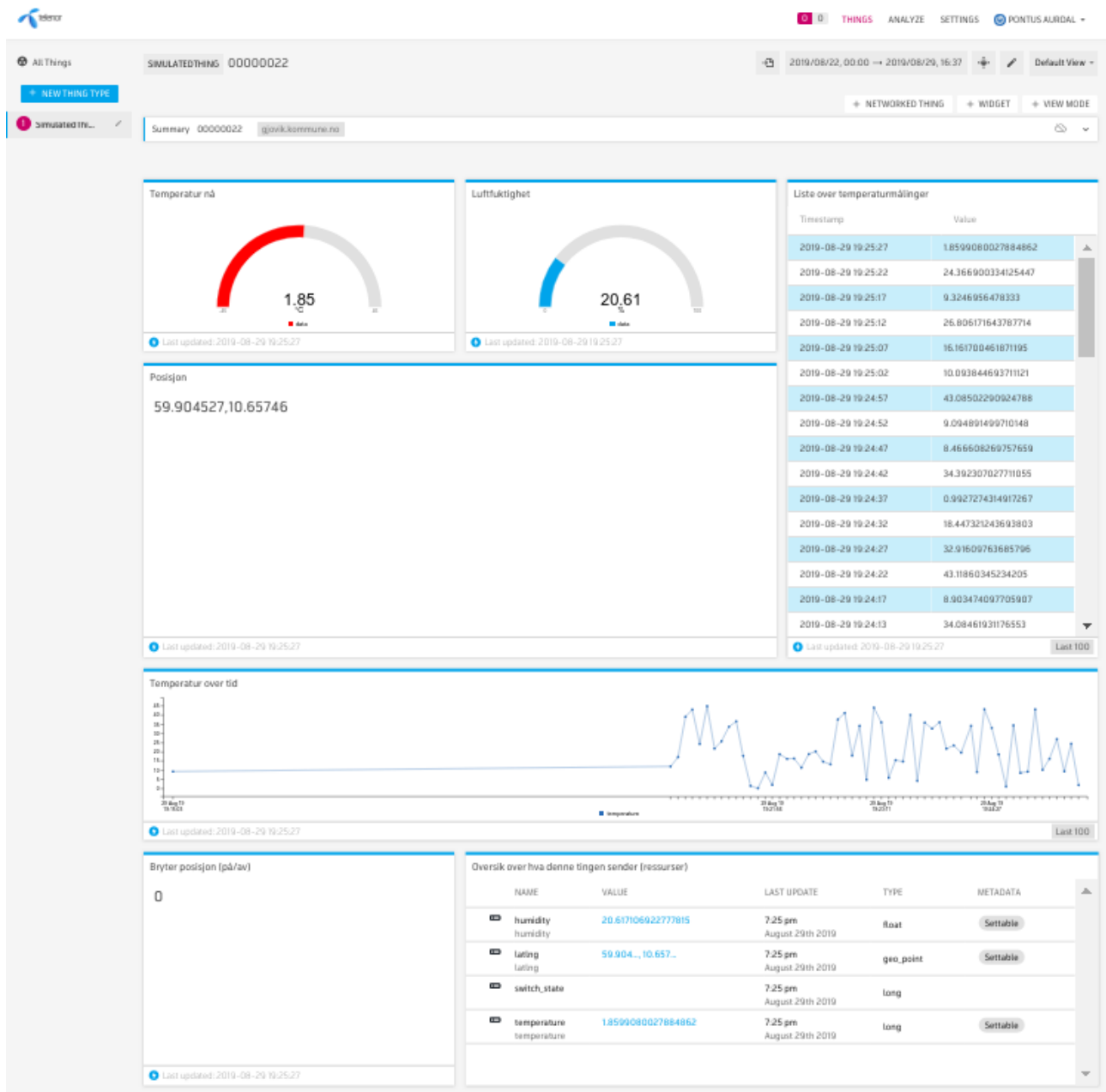
(Windows): <https://www.guru99.com/download-install-node-js.html>

When Node.js is installed, open a terminal (also called a “command prompt” on Windows), navigate to the script location and run «npm install» (to install required dependencies) and then «node index.js» (to start the simulation):



```
pontus@pontus-P50s: ~/Downloads/mic-simulated-mqtt-device
File Edit View Search Terminal Help
pontus@pontus-P50s:~/Downloads/mic-simulated-mqtt-device$ node index.js
Client connected! Starting to simulate messages.
Sent a simulated message to the topic $aws/things/00000022/shadow/update!
Message: {"state":{"reported":{"temperature":36.57484129615328,"humidity":24.81689569996867,"latlng":"59.904529,10.657480","switch_state":0}}}
Sent a simulated message to the topic $aws/things/00000022/shadow/update!
Message: {"state":{"reported":{"temperature":38.53606196782917,"humidity":41.11491492290416,"latlng":"59.904582,10.657473","switch_state":0}}}
Sent a simulated message to the topic $aws/things/00000022/shadow/update!
Message: {"state":{"reported":{"temperature":15.394833359307361,"humidity":32.50907932242009,"latlng":"59.904588,10.657417","switch_state":1}}}
Sent a simulated message to the topic $aws/things/00000022/shadow/update!
Message: {"state":{"reported":{"temperature":1.9472648847724117,"humidity":44.73815290333466,"latlng":"59.90451,10.657479","switch_state":1}}}
Sent a simulated message to the topic $aws/things/00000022/shadow/update!
Message: {"state":{"reported":{"temperature":18.60408365683043,"humidity":42.50907932242009,"latlng":"59.904588,10.657417","switch_state":1}}}
```

Good job! Data is now simulated and sent to MIC and you can start designing your dashboard to visualize the data:



Pontus Edvard Aurdal

Technical Expert IoT

Telenor Norway

+47 40 60 62 81

pontus.aurdal@telenor.com