Tel. +370 5 212 7472
Fax. +370 5 276 1380
www.teltonika-gps.com
info@teltonika.lt

AWS IOT CUSTOM (MQTT) AND DEVICE CONFIGURATION GUIDE

Contents

UAB TELTONIKA TELEMATICS
Saltoniskiu st. 9B-1, LT-08105
Vilnius, Lithuania

Registration code 305578349
VAT number LT100013240611

Swedbank AB
LT71 7300 0101 6274 0043
S.W.I.F.T. HABALT22

1. General information

Firmware version:	03.27.10.Rev.520
Hardware revision:	Any
Configurator:	1.7.30_E.AWS_R.5
Notes:	This guide does not contain any information how to store received MQTT packets in the AWS. Refer to this link.

2. AWS IoT core configuration

2.1. Setting up AWS IoT core

When logged in the AWS console, click on *Services* on the top left hand side screen, to access IoT core.

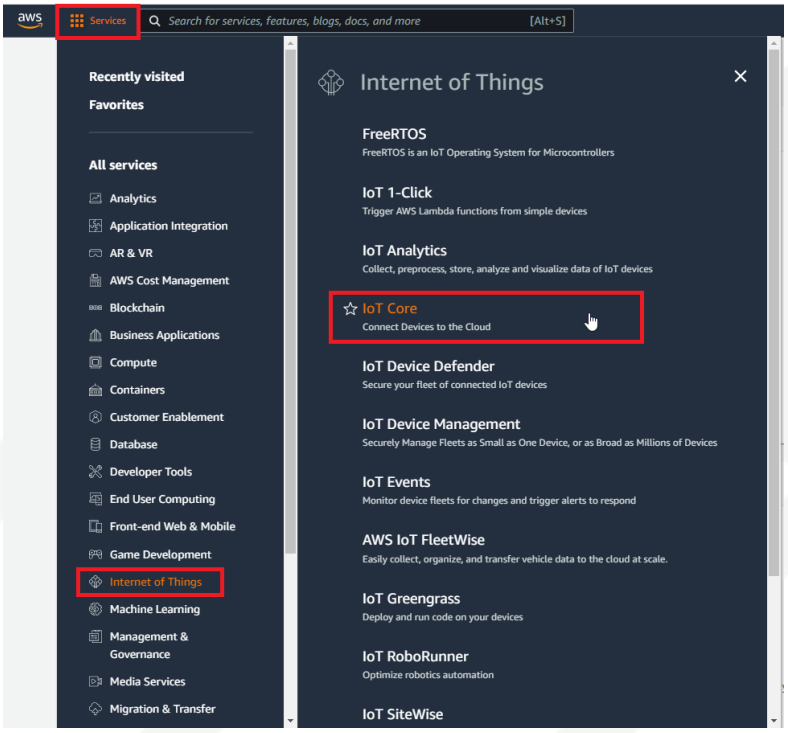


Figure 1. Accessing AWS IoT core from AWS console

After accessing AWS IoT core, select *Manage* on the sidebar on the left side, then select *Things (Manage->Things).* And click on *Create things.*

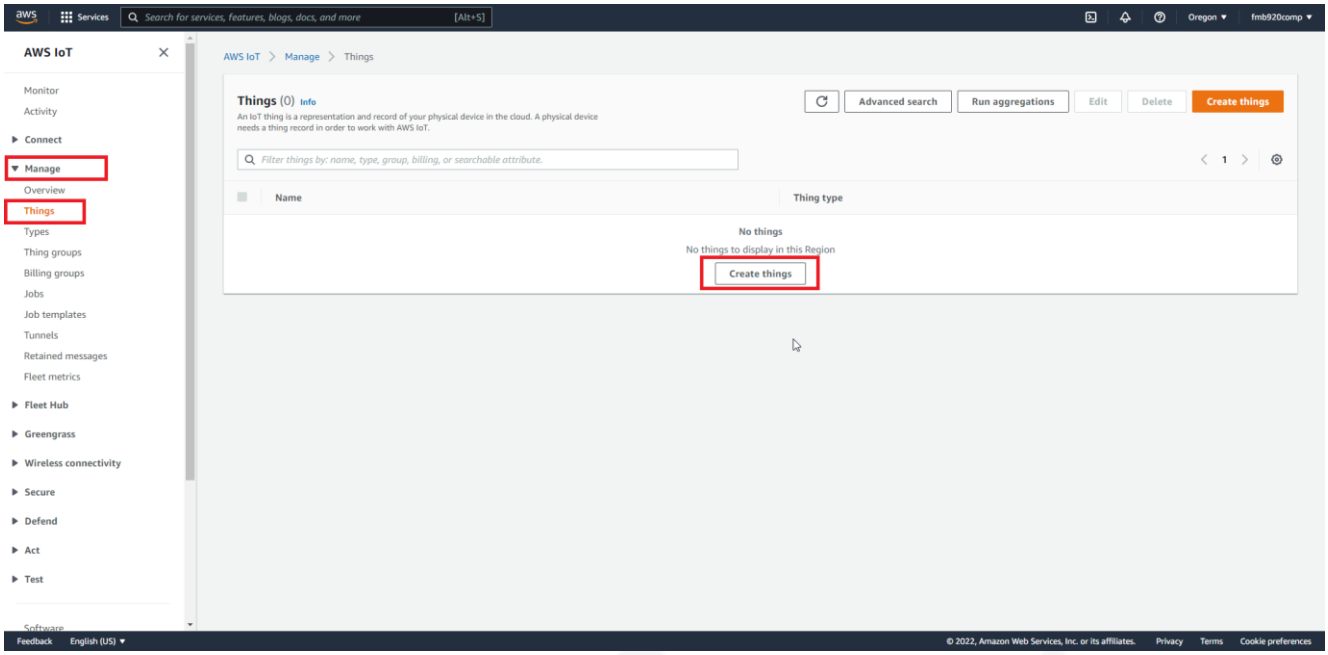


Figure 2. Accessing Things

Afterwards for select *Create single thing* and click *Next.*

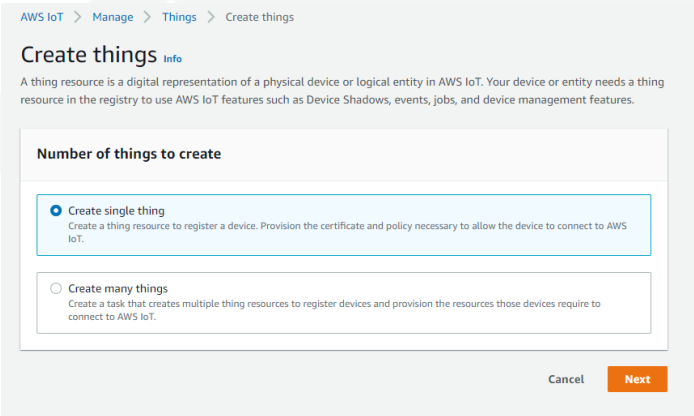


Figure 3. Creating single thing

After creating single thing, enter *Thing's name* and in the *Device Shadow tab* select *Unnamed shadow (classic).* Then click *Next.*

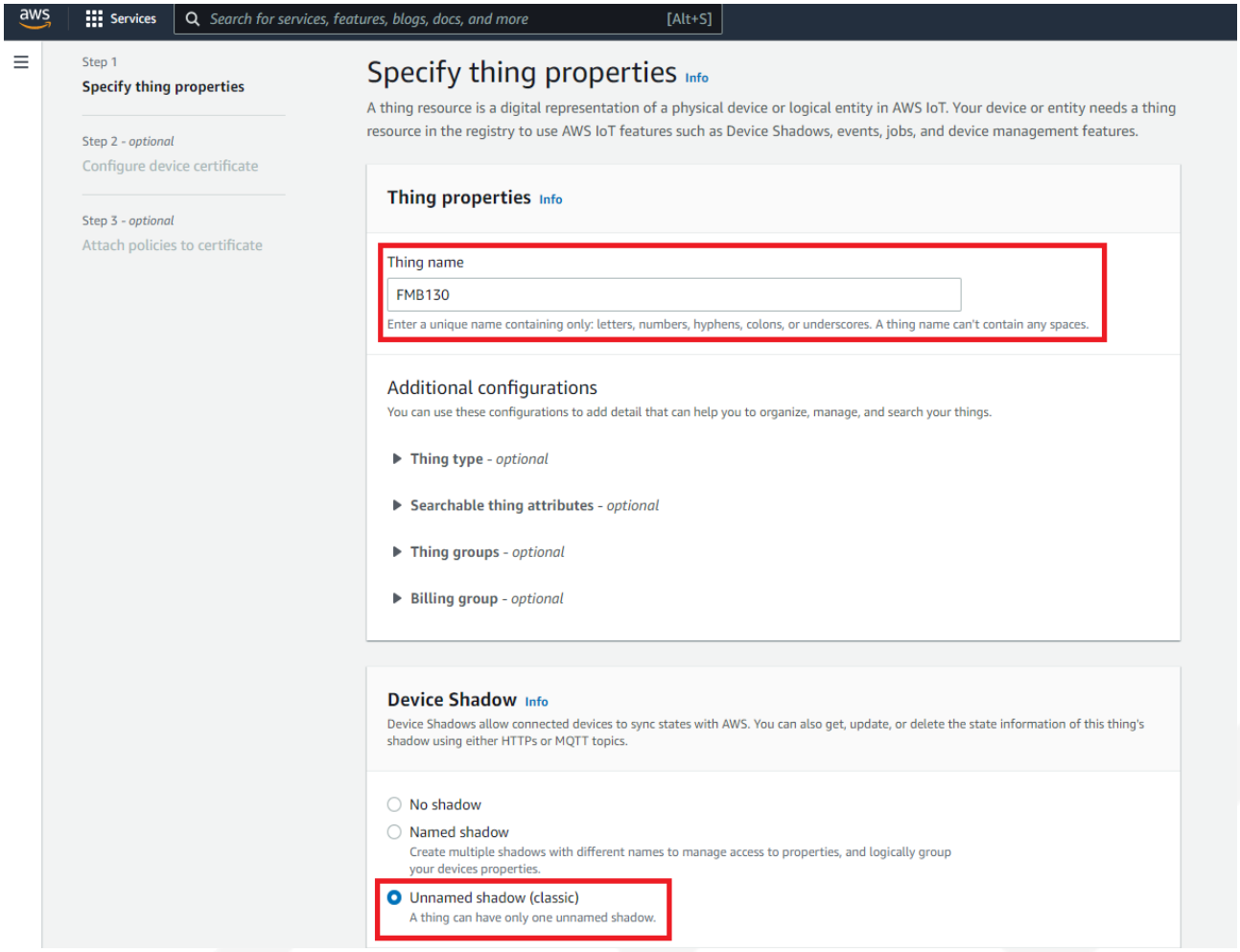


Figure 4. Specifying thing properties

Then when selecting *Device certificate,* select *Auto-generate a new certificate* and click *Next.*

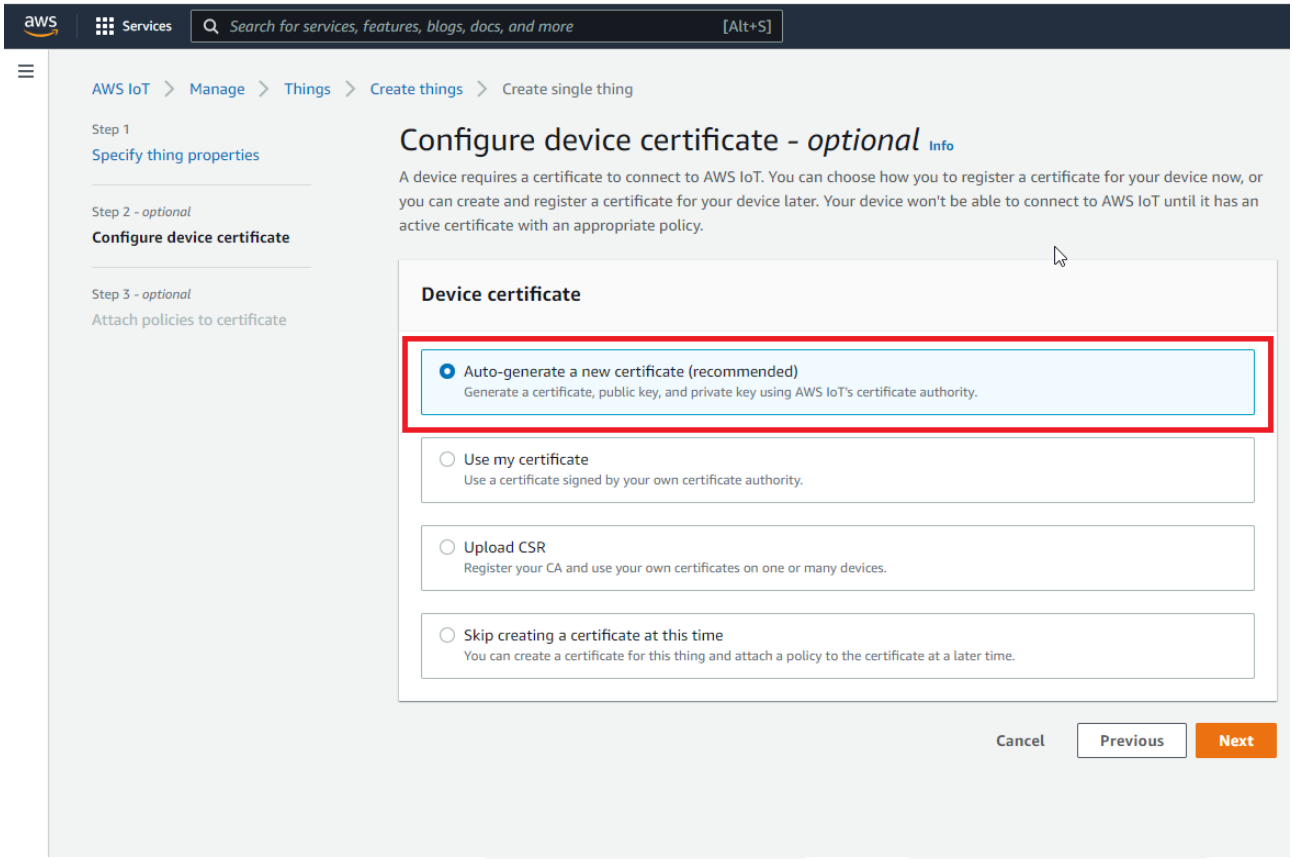


Figure 5. Selecting Certificate

After this select *Create policy* to create it and attach it to *Certificate.* In the Create Policy window, enter *Policy name.* In the ***Policy document (1)*** tab for ***Policy Action (2)*** select * and for Policy resource enter * .

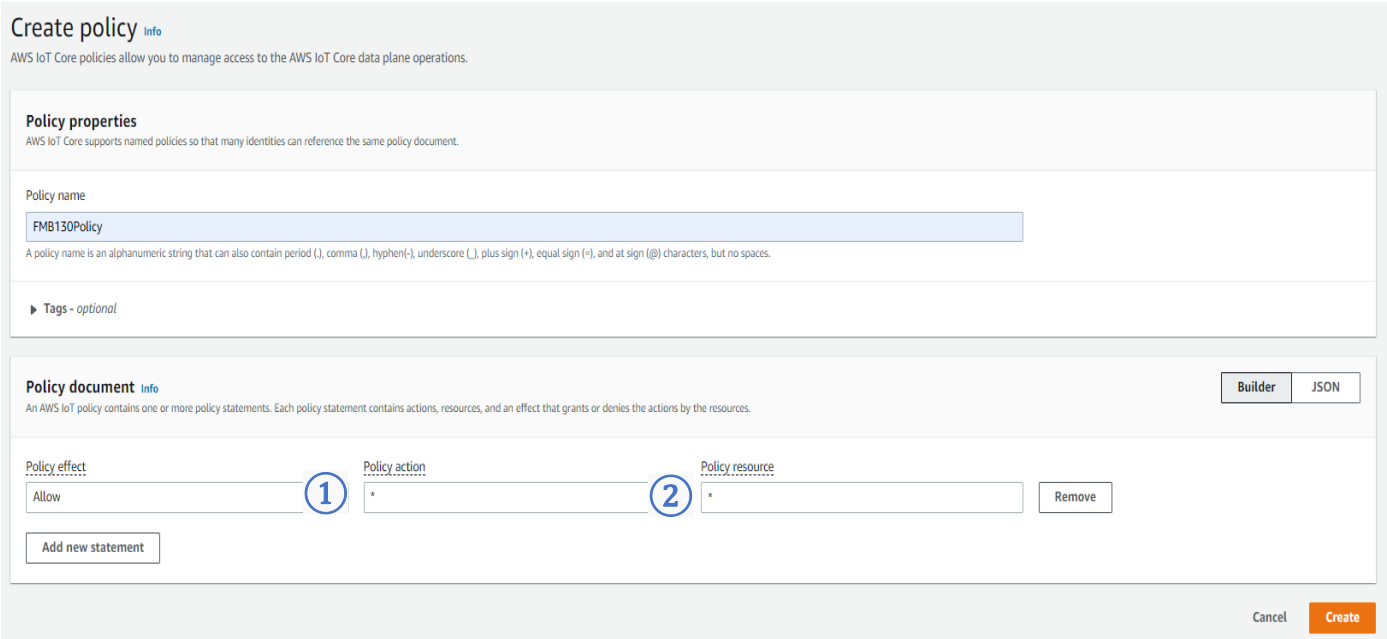


Figure 6. Creating policy for certificate

After creating policy, return to *Certificate* tab (Seperate tab after pressing *Create policy* should've popped out). Then select the created policy to attach it to the certificate and thing. After that click *Create thing.*

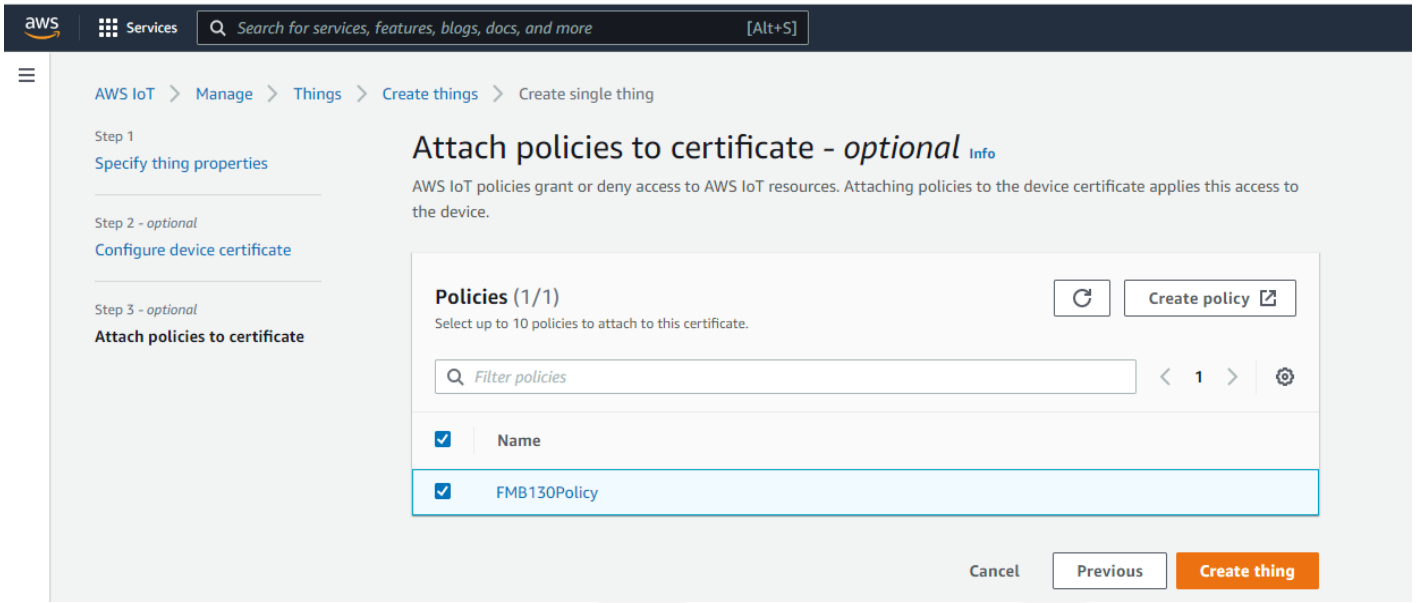


Figure 7. Attaching created certificate and creating thing

Then window with Certificate files and key files download options should pop out. It's recommended to download all files, because later some of them will not be available for download. The files that are required for usage with FMX devices are: ***Device certificate (1), private key(2), and Amazon Root CA 1 file(3)***, but it's recommended to download them all and store them in secured place.

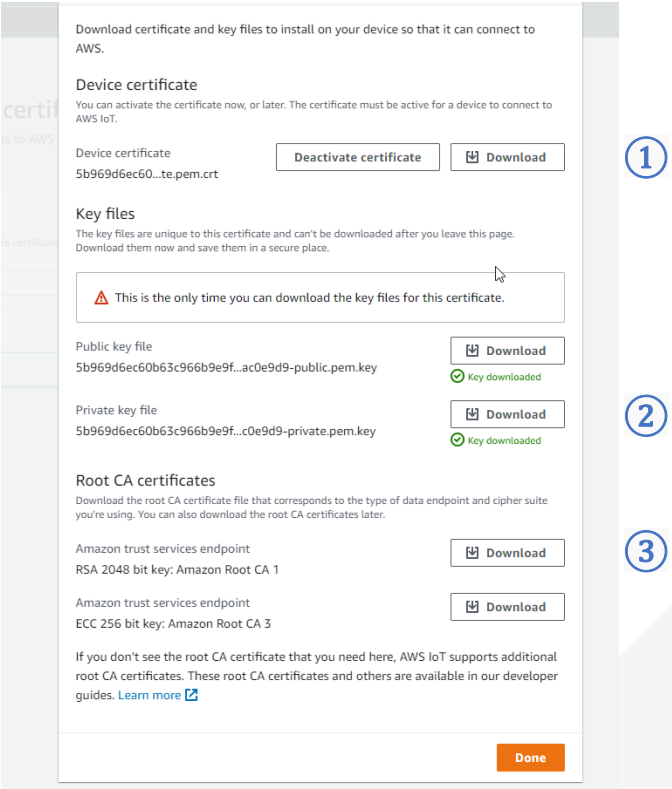


Figure 8. Certificate files download window

2.2. Finding decive data endpoint (server domain)

To receive server domain (in AWS endpoint) click on the side bar on the left *Settings.*

Or click on the side bar on left side *Things*, select the created thing, after it click *Interact->View Settings.* Whole path - *(Things->*YourThingName*->Interact->ViewSettings).* Page containing endpoint will open. **Copy the whole endpoint address**.

Port for accessing this endpoint is ***8883***.

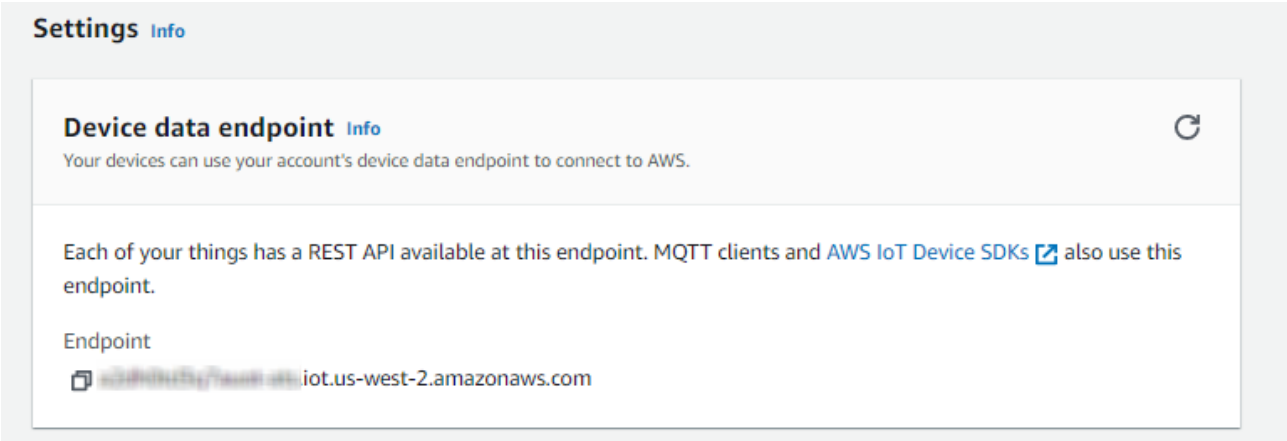
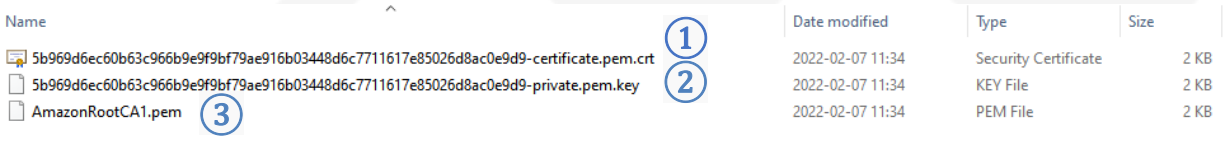


Figure 9. Device data endpoint

3. Configuring the device

3.1. Security and certificates

Find *Certificate file* ending with extension pem.crt *Private key file* and *AmazoonRootCA1 file* (no need to change filenames). These file should have been downloaded when creating *Thing* in AWS IoT Core.



Upload the mentioned files in the *Security tab* in the Teltonika Configurator.

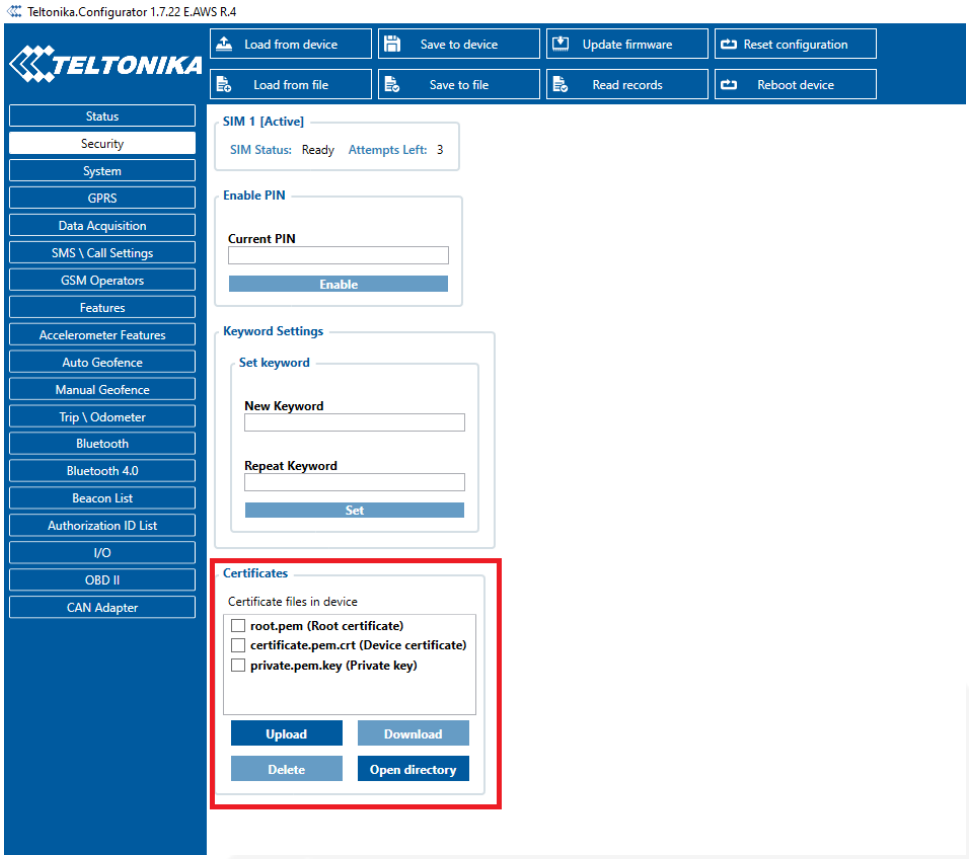


Figure 10. Uploading certificates

After uploading certificates, go to *System tab* and in Data protocol section select - *Codec JSON.*

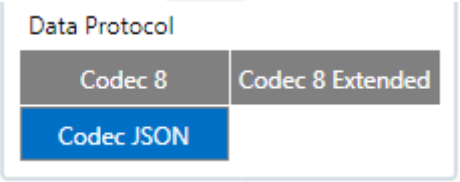


Figure 11. Selecting Data protocol

3.2. Device GPRS configuration for AWS IoT Custom MQTT settings

In the ***GPRS tab,*** under ***Server Settings*** select:

1. Domain – ***Endpoint*** from the AWS, Port: ***8883***
2. Protocol – ***MQTT***
3. TLS Encryption – ***TLS/DTLS***

In the ***MQTT Settings*** section select:

1. MQTT Client Type – ***AWS IoT Custom***
2. Device ID – enter device ***IMEI*** *(optional)*
3. Leave Data and Command Topics unchanged.

Save the configuration to the device.

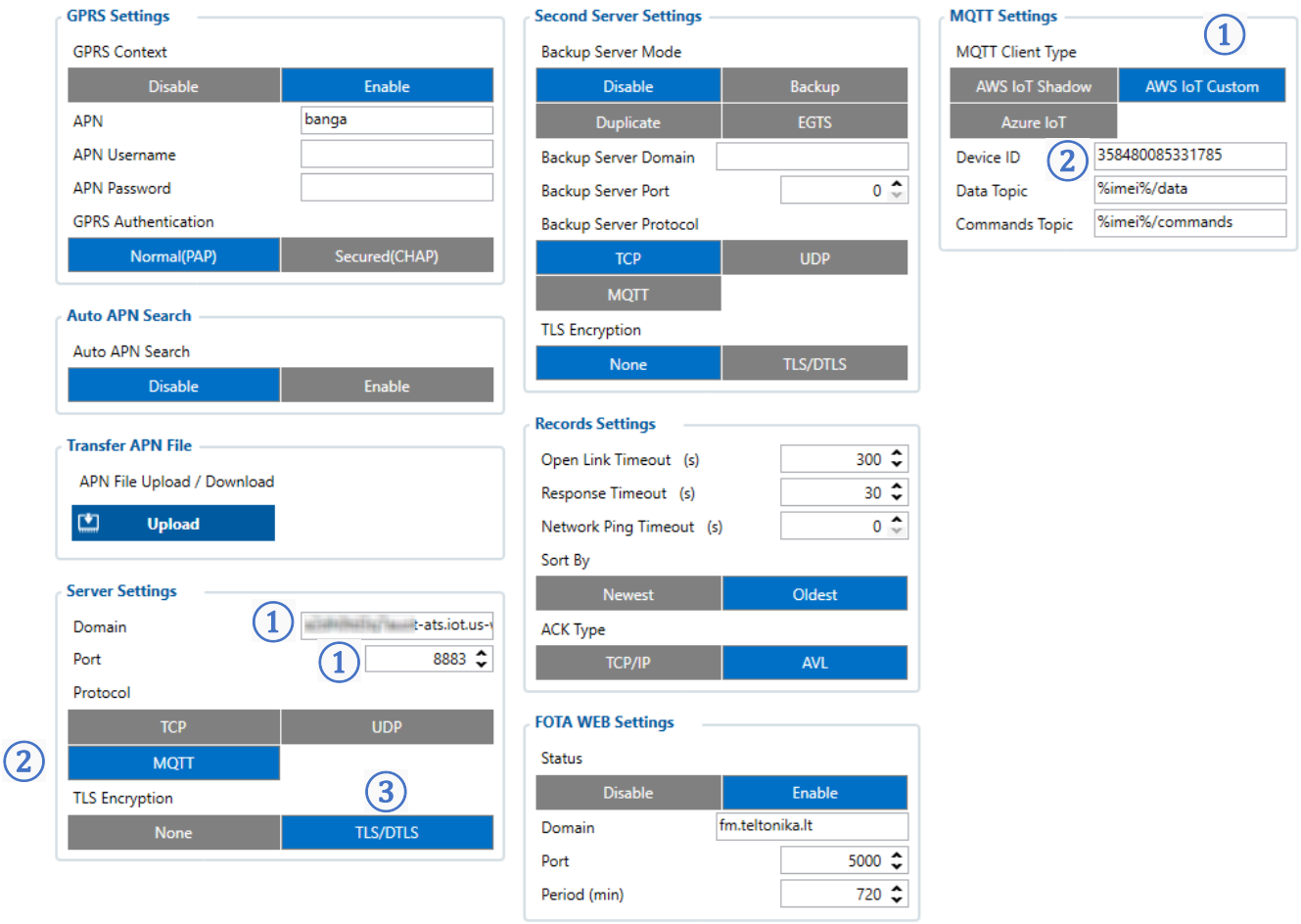


Figure 12. GPRS settings for MQTT AWS IoT Custom

4. Checking received data and sending commands in the AWS IoT core

The data received from the device can be found in the ***MQTT test client,*** which can be found in the bottom of sidebar on the left.

To see incoming data, subscribe to topic - ****DeviceImei*/data*** . Or subscribe to # to see all incoming outgoing data in the Topics.

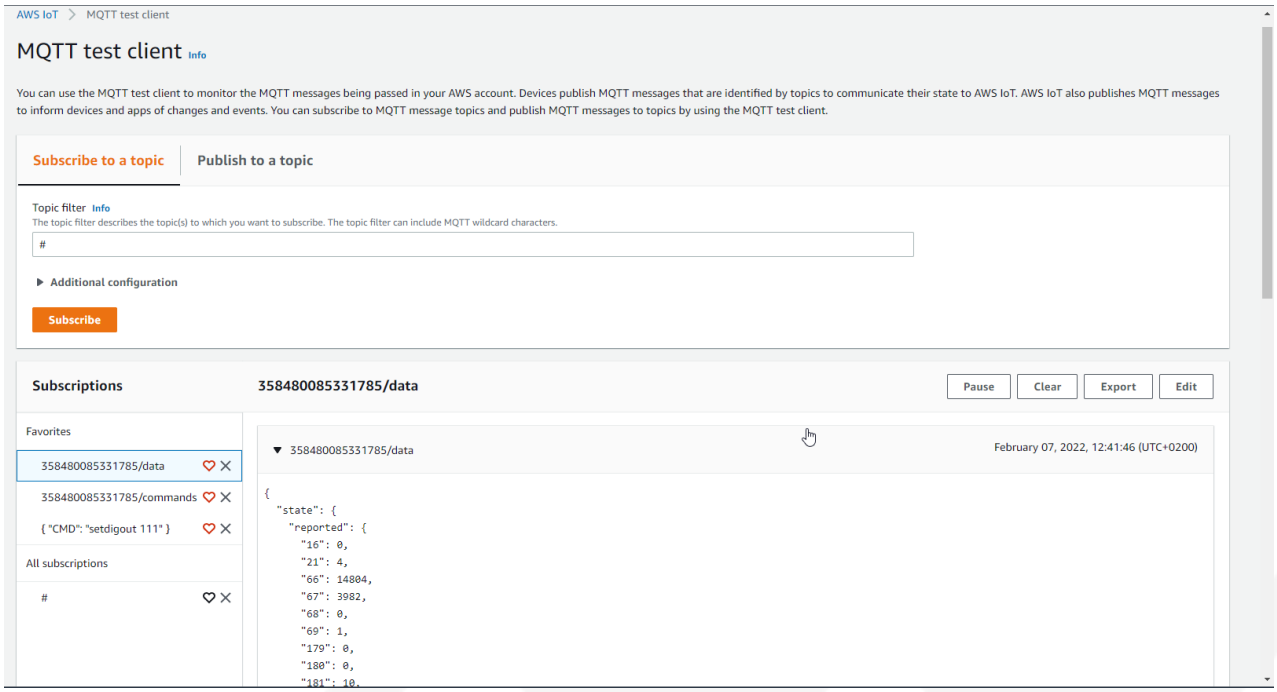


Figure 13. Subscribing to data topic

Incoming data is received in JSON format, for e.g.:


```
{
    "format": "json",
    "topic": "358480085331785/data",
    "timestamp": 1644227812734,
    "payload": {
      "state": {
        "reported": {
          "16": 0,
          "21": 4,
          "66": 14547,
          "67": 4019,
          "68": 144,
          "69": 1,
          "179": 0,
          "180": 0,
          "181": 9,
          "182": 6,
          "200": 0,
          "239": 1,
          "240": 0,
          "241": 24602,
          "380": 0,
          "ts": 1644227810000,
          "pr": 1,
          "latlng": "54.700450,25.259818",
          "alt": 128,
          "ang": 0,
          "sat": 15,
          "sp": 0,
          "evt": 0
        }
      }
    }
  }
```


Figure 14. Received data format

To send SMS/GPRS commands to the device, in the same MQTT test client window select *Publish to a topic.* Enter topic name - ***DeviceIMEI*/commands** . In the Message payload enter wanted GPRS/SMS command in following format and press *Publish*:

{"CMD": "<Command>"}

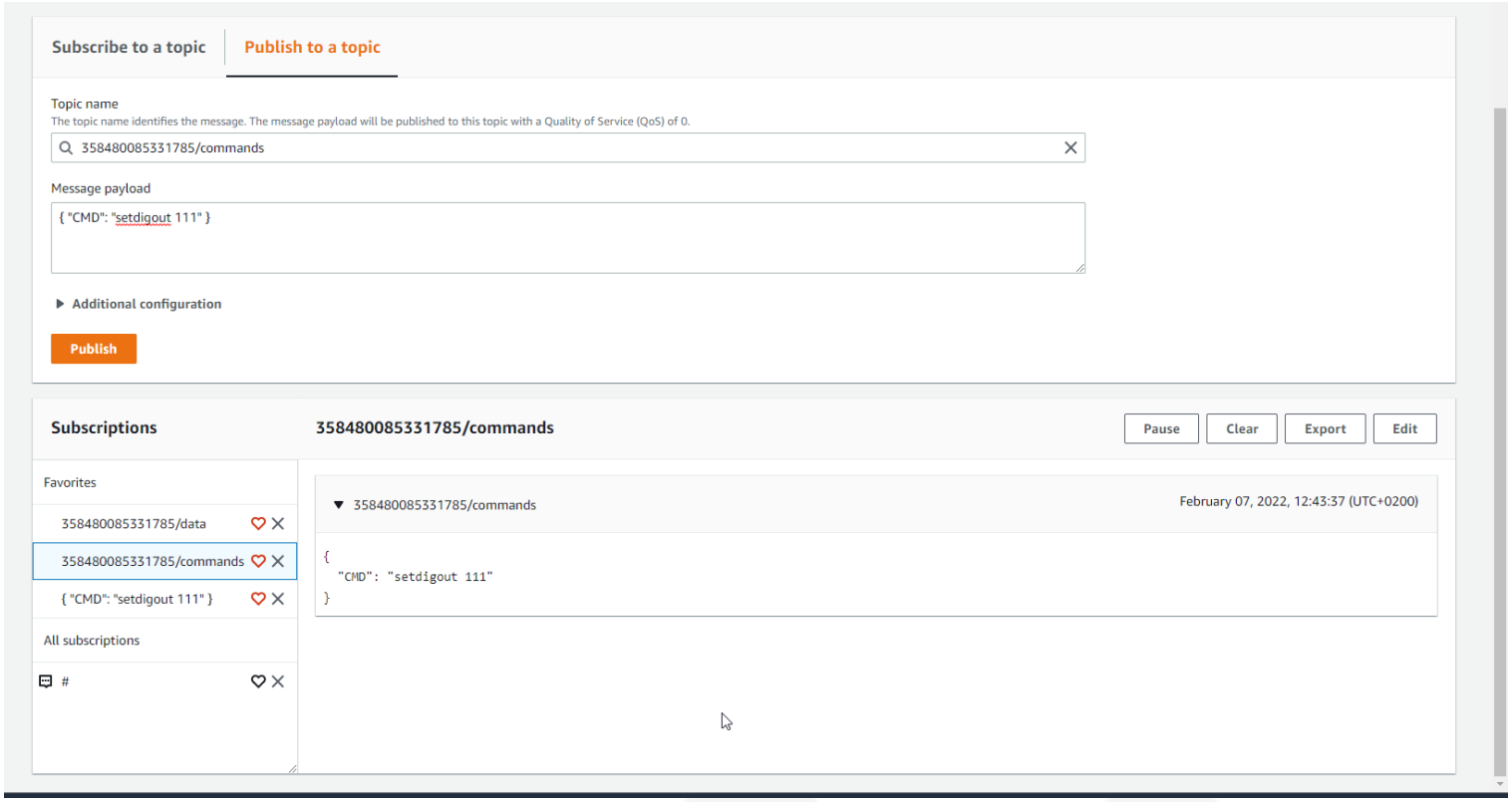


Figure 15. Sending Comand in AWS IoT Core

The response to the command will be shown in the **Data topic**:

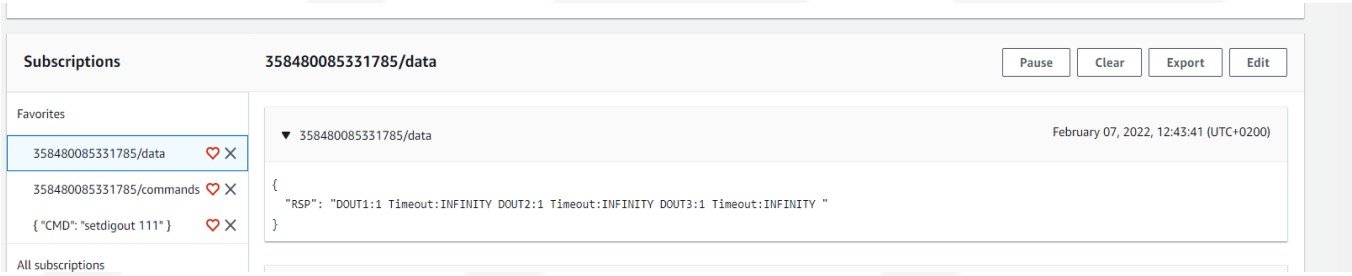


Figure 16. Response to a command in the data topic, the command was published in command topic