

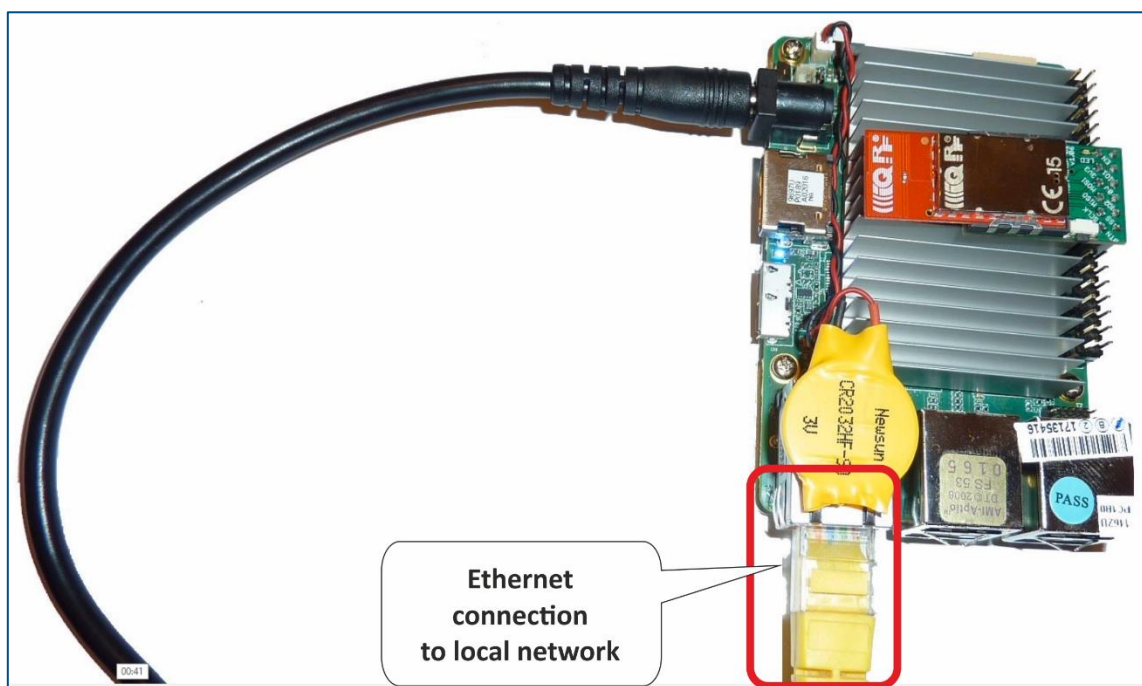
UP-IQRF IoT Starter Kit – Part 3: Connect to the cloud – AWS IoT

Note: If the PDF Guide is opened in a viewer mode, we strongly recommend downloading it and open on your computer locally to have hyperlinks functional and to be able to copy strings. The Download button you will find at the top of the page with a PDF preview.

IoT Starter Kit is designed in the way to be connectable to different clouds via bidirectional MQTT channel. So, you can collect, store, process and visualize data in a cloud or you can send your commands to the IQRF network remotely. In this part, we will configure the UP board to communicate with the Amazon Web Services (AWS) through the MQTT channel.

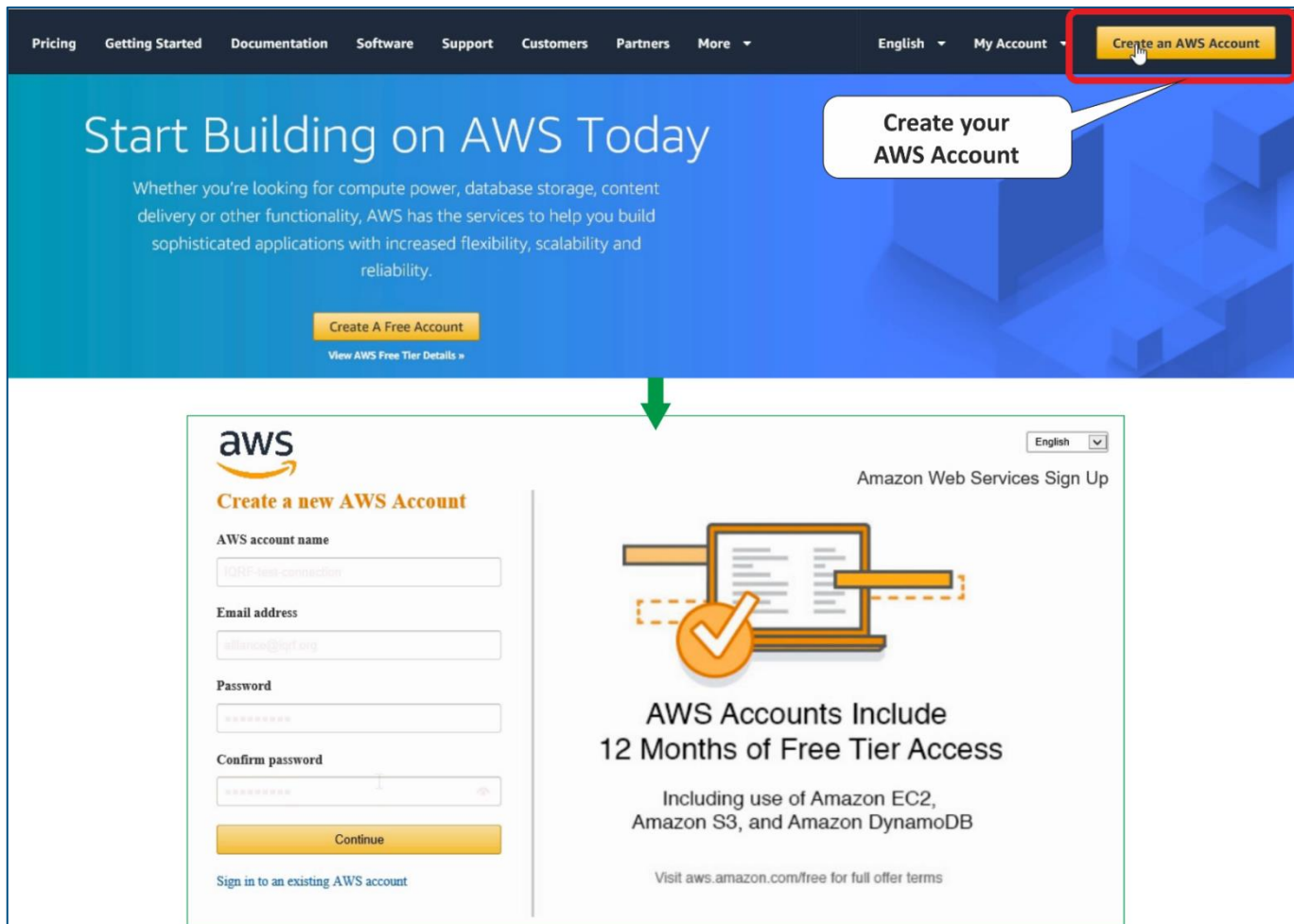
1 Local network

Connect your UP board to your local network so it can obtain an IP address using DHCP. In the following steps, you will enter this address into your web browser on your computer (which is in the same local network as the UP board) and configure your gateway through the IQRF Daemon Web application.



2 Amazon Web Services account

First, create an Amazon Web Services account (aws.amazon.com). You must fill in your personal or company data and add your credit card details. Your credit card will be used for payments in a case you exceed limits of the selected services.



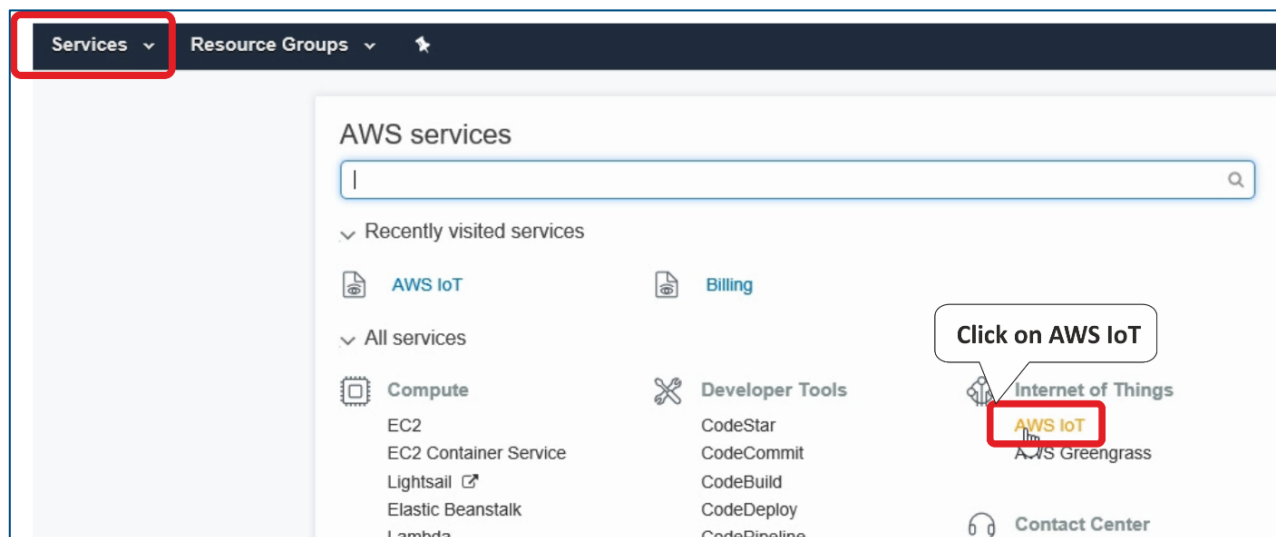
The screenshot shows the AWS website. At the top, there is a navigation bar with links: Pricing, Getting Started, Documentation, Software, Support, Customers, Partners, and More. On the right, there are links for English, My Account, and a highlighted 'Create an AWS Account' button. Below the navigation bar, the main heading is 'Start Building on AWS Today'. A callout box says 'Create your AWS Account'. Below this, there is a 'Create A Free Account' button and a link to 'View AWS Free Tier Details'. A green arrow points down to the 'Create a new AWS Account' sign-up form. The form has fields for 'AWS account name' (with the example 'IQRF-test-connection'), 'Email address' (with the example 'alliance@iqrf.org'), 'Password', and 'Confirm password'. There is a 'Continue' button and a link to 'Sign in to an existing AWS account'. To the right of the form, there is a graphic of a laptop with a checkmark and the text 'AWS Accounts Include 12 Months of Free Tier Access'. Below this, it says 'Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB' and a link to 'Visit aws.amazon.com/free for full offer terms'.

3 Set up the connection

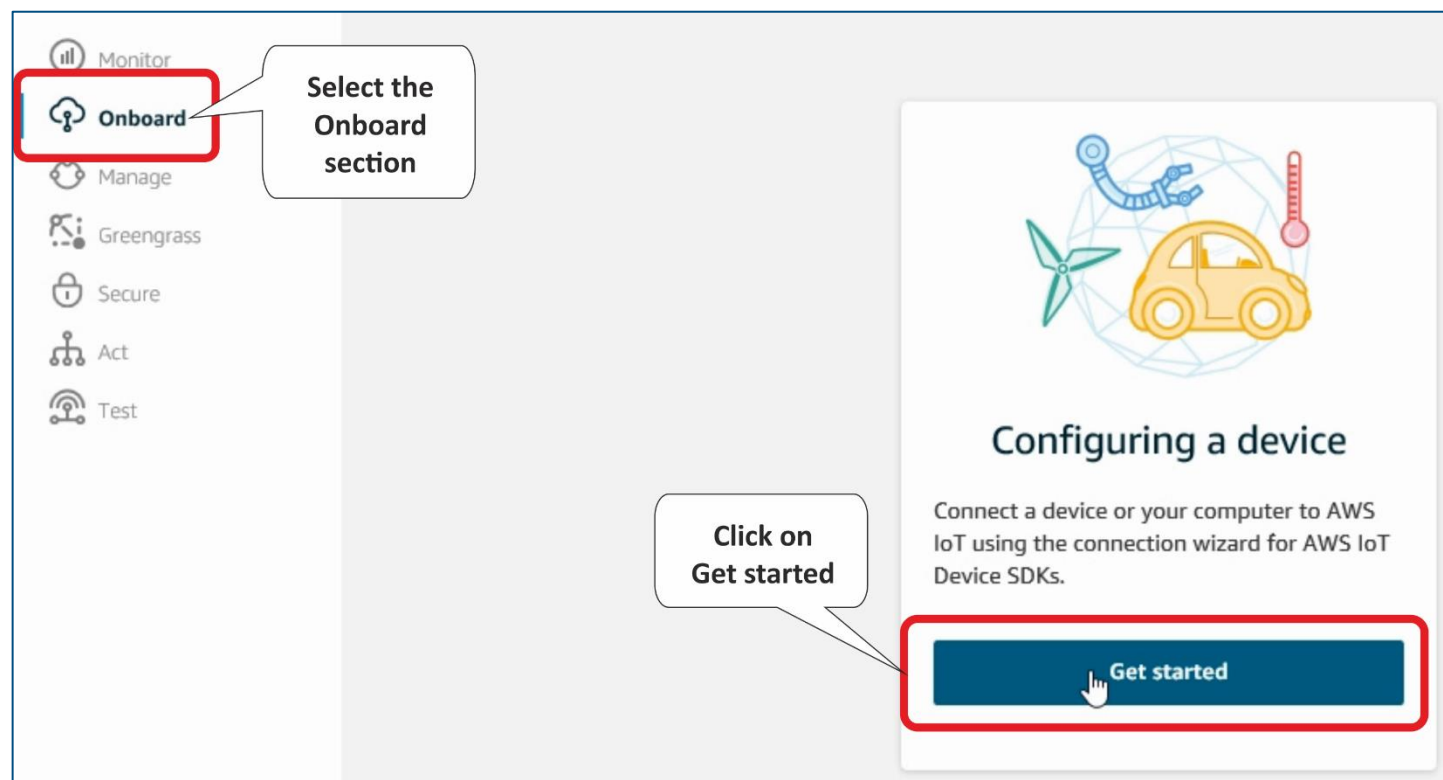
To set up the connection between AWS and your UP board, you need to do some configuration steps on both sides.

In **Services**, in the **Internet of Things** section of AWS, find **AWS IoT**.

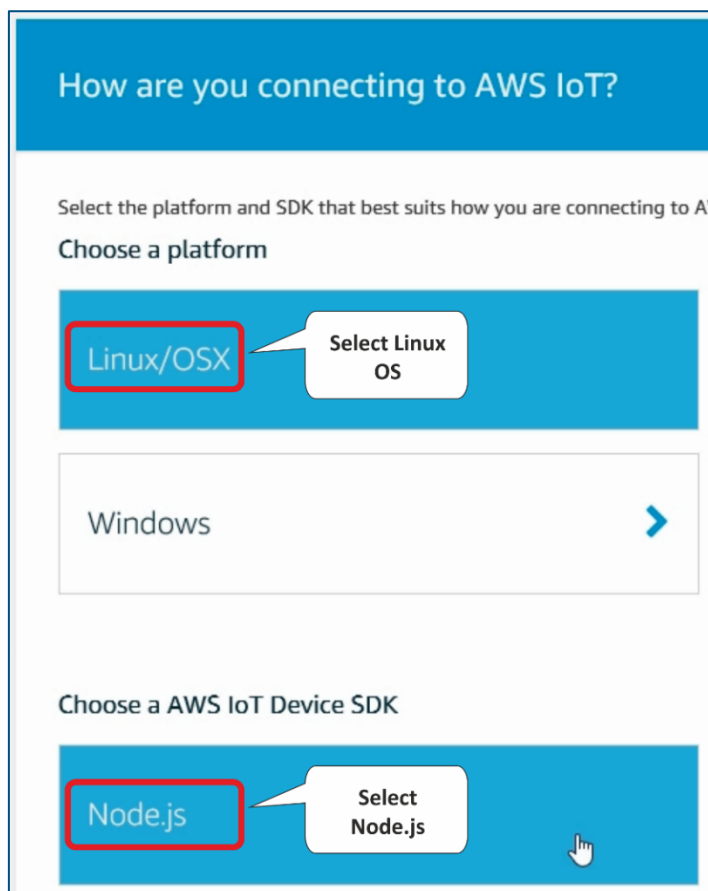
Note: the environment of AWS may look different because of often changes and its personalization. This guide shows the status of March 2018. You need to look for appropriate items to configure the MQTT connection.



Click on **Get started** in the **Onboard** section. You will register your device, download the connection kit, and configure and test the connection with your device.

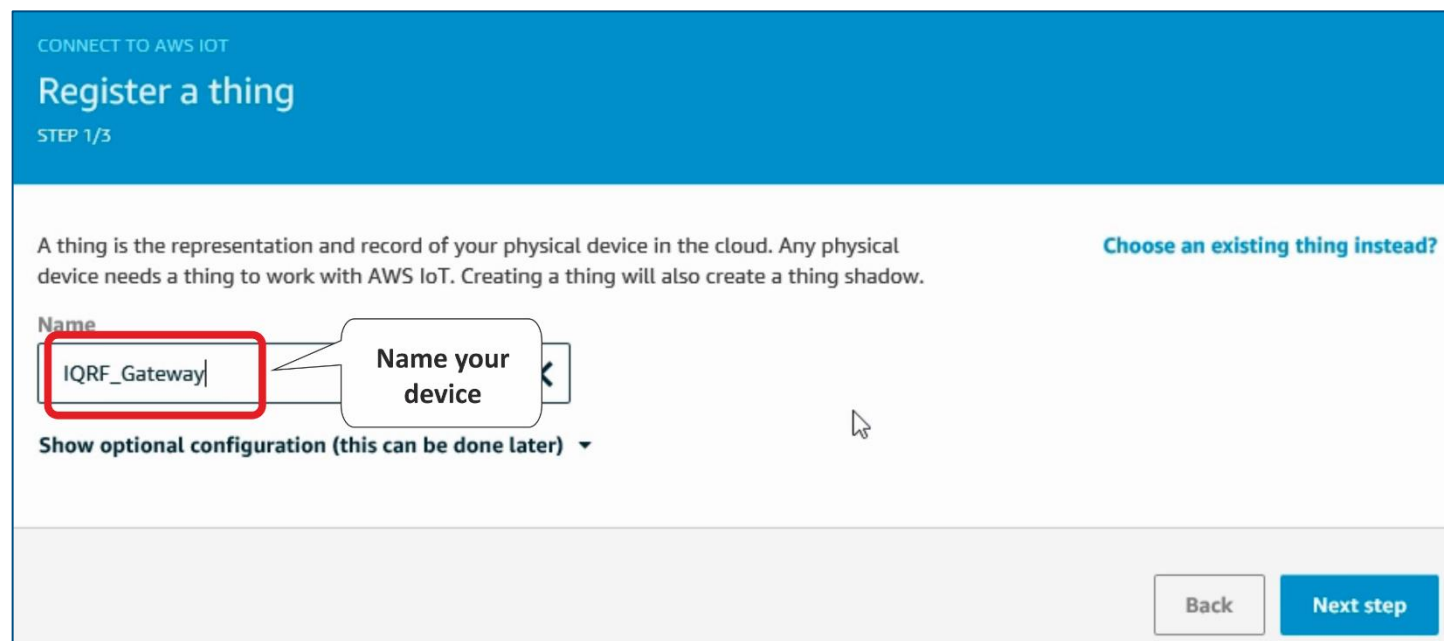


Set up how you will be connected to the AWS IoT. Select the **Linux** operating system and **Node.js** as the AWS IoT Device SDK.



Enter the name of your connected device.

Note: You can choose your own name. In that case in later steps, you need to use the given name.



Download the connection kit to get a certificate and keys for a secure MQTT connection.

CONNECT TO AWS IOT

Download a connection kit

STEP 2/3

The following AWS IoT resources will be created:

A thing in the AWS IoT registry	IQRF_Gateway
A policy to send and receive messages	IQRF_Gateway-Policy

[Preview policy](#)

The connection kit contains:

A certificate and private key	IQRF_Gateway.cert.pem, IQRF_Gateway.private.key
AWS IoT Device SDK	Node.js SDK
A script to send and receive messages	start.sh

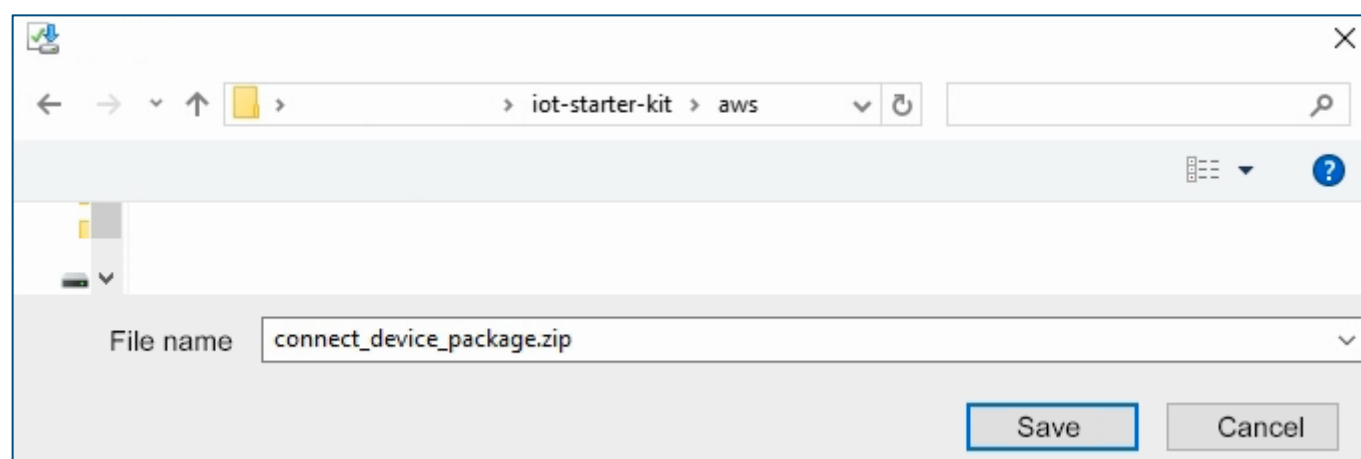
Before your device can connect and publish messages, you will need to download the connection kit.

Download connection kit for

Linux/OSX

Download connection kit

Save and unzip this file to your computer. Store the certificate and the keys for further use.



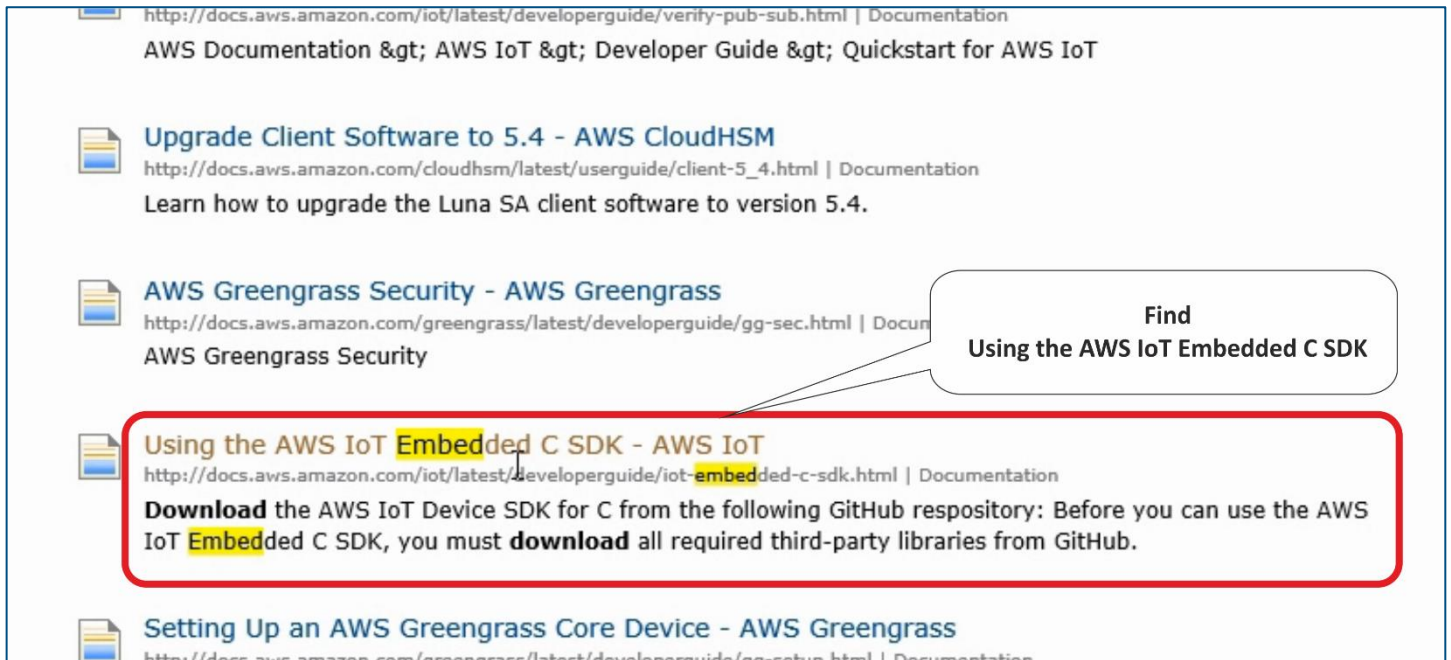
After saving process, go to the documentation.

The screenshot shows the AWS IoT console interface. At the top, there's a status bar with a green checkmark icon and the text "Received messages from the device", with a "Learn more" link. Below this is a section asking "Want to learn more about the components of AWS IoT?" with a "Try the interactive overview" link and a "Done" button. The main section is titled "Explore other parts of AWS IoT" and contains three cards: "See all of your things", "Learn about policies", and "Dive into the documentation". The "Dive into the documentation" card is highlighted with a red rectangular box. A callout bubble points to this card with the text "Go to the documentation".





Here, look up the **Download root CA** string. Search in the **Entire site** to be sure to find it.

The screenshot shows the AWS IoT Developer Guide search interface. At the top, there's a "Menu" button and the AWS logo. Below this is the "AWS IoT Developer Guide" header. A search bar is present with a magnifying glass icon. The search bar has a dropdown menu with "Entire Site" selected, and the text "download root CA" is entered in the search field. A callout bubble points to the search bar with the text "In the entire site look up the Download root CA". Below the search bar, there are links for "What Is AWS IoT?", "How AWS IoT Works", and "Getting Started with AWS IoT". On the right side, there's a section titled "AWS IoT Components" with a sub-header "AWS IoT provides secure, bi-directional communication between IoT appliances and the AWS cloud, enabling your users to control the devices".

In the search results, find the article **Using the AWS IoT Embedded C SDK**. The number of records in a search result can exceed the page limit so you need to go through more pages.

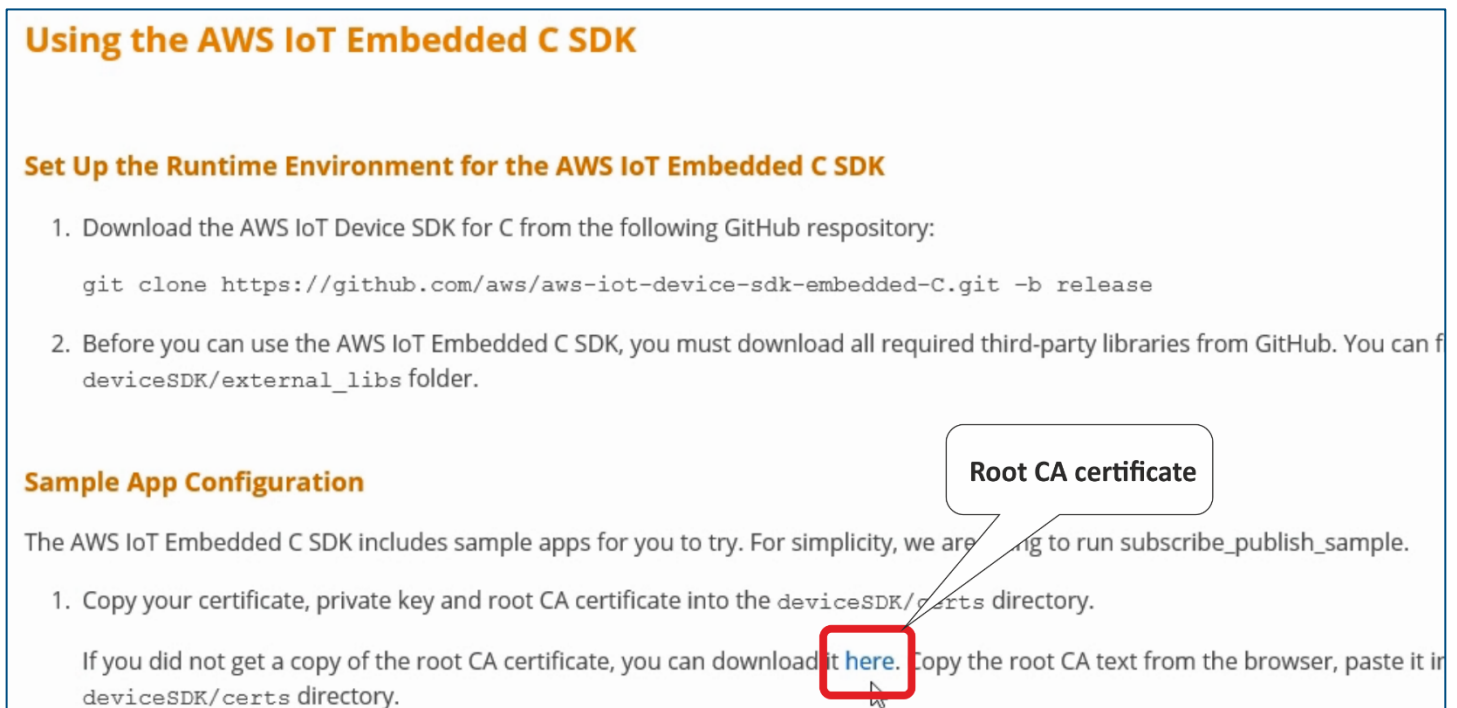


The screenshot shows a list of search results from AWS documentation. The results include:

- <http://docs.aws.amazon.com/iot/latest/developerguide/verify-pub-sub.html> | Documentation
AWS Documentation > AWS IoT > Developer Guide > Quickstart for AWS IoT
-  **Upgrade Client Software to 5.4 - AWS CloudHSM**
http://docs.aws.amazon.com/cloudhsm/latest/userguide/client-5_4.html | Documentation
Learn how to upgrade the Luna SA client software to version 5.4.
-  **AWS Greengrass Security - AWS Greengrass**
<http://docs.aws.amazon.com/greengrass/latest/developerguide/gg-sec.html> | Documentation
AWS Greengrass Security
-  **Using the AWS IoT Embedded C SDK - AWS IoT**
<http://docs.aws.amazon.com/iot/latest/developerguide/iot-embedded-c-sdk.html> | Documentation
Download the AWS IoT Device SDK for C from the following GitHub repository: Before you can use the AWS IoT Embedded C SDK, you must **download** all required third-party libraries from GitHub.
-  **Setting Up an AWS Greengrass Core Device - AWS Greengrass**
<http://docs.aws.amazon.com/greengrass/latest/developerguide/gg-setup.html> | Documentation

A red box highlights the article "Using the AWS IoT Embedded C SDK - AWS IoT". A callout bubble points to this article with the text "Find Using the AWS IoT Embedded C SDK".

Here you can find the root certificate.



The screenshot shows the content of the article "Using the AWS IoT Embedded C SDK". The main heading is "Using the AWS IoT Embedded C SDK". Below it is the subheading "Set Up the Runtime Environment for the AWS IoT Embedded C SDK". The article contains two numbered steps:

1. Download the AWS IoT Device SDK for C from the following GitHub repository:

```
git clone https://github.com/aws/aws-iot-device-sdk-embedded-C.git -b release
```
2. Before you can use the AWS IoT Embedded C SDK, you must download all required third-party libraries from GitHub. You can find them in the `deviceSDK/external_libs` folder.

Below the steps is the section "Sample App Configuration". The text states: "The AWS IoT Embedded C SDK includes sample apps for you to try. For simplicity, we are going to run `subscribe_publish_sample`."

1. Copy your certificate, private key and root CA certificate into the `deviceSDK/certs` directory.

If you did not get a copy of the root CA certificate, you can download it [here](#). Copy the root CA text from the browser, paste it in the `deviceSDK/certs` directory.

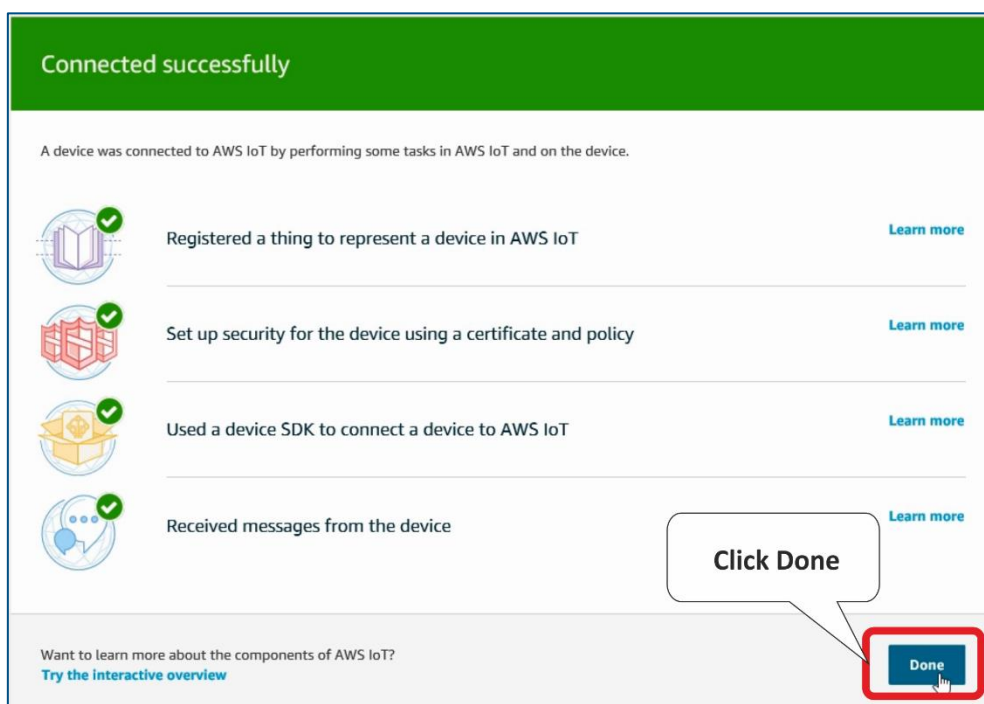
A callout bubble points to the link "here" with the text "Root CA certificate".

Copy the string to a text file and save it as **rootCA.pem** to the directory with other certificates on your computer.

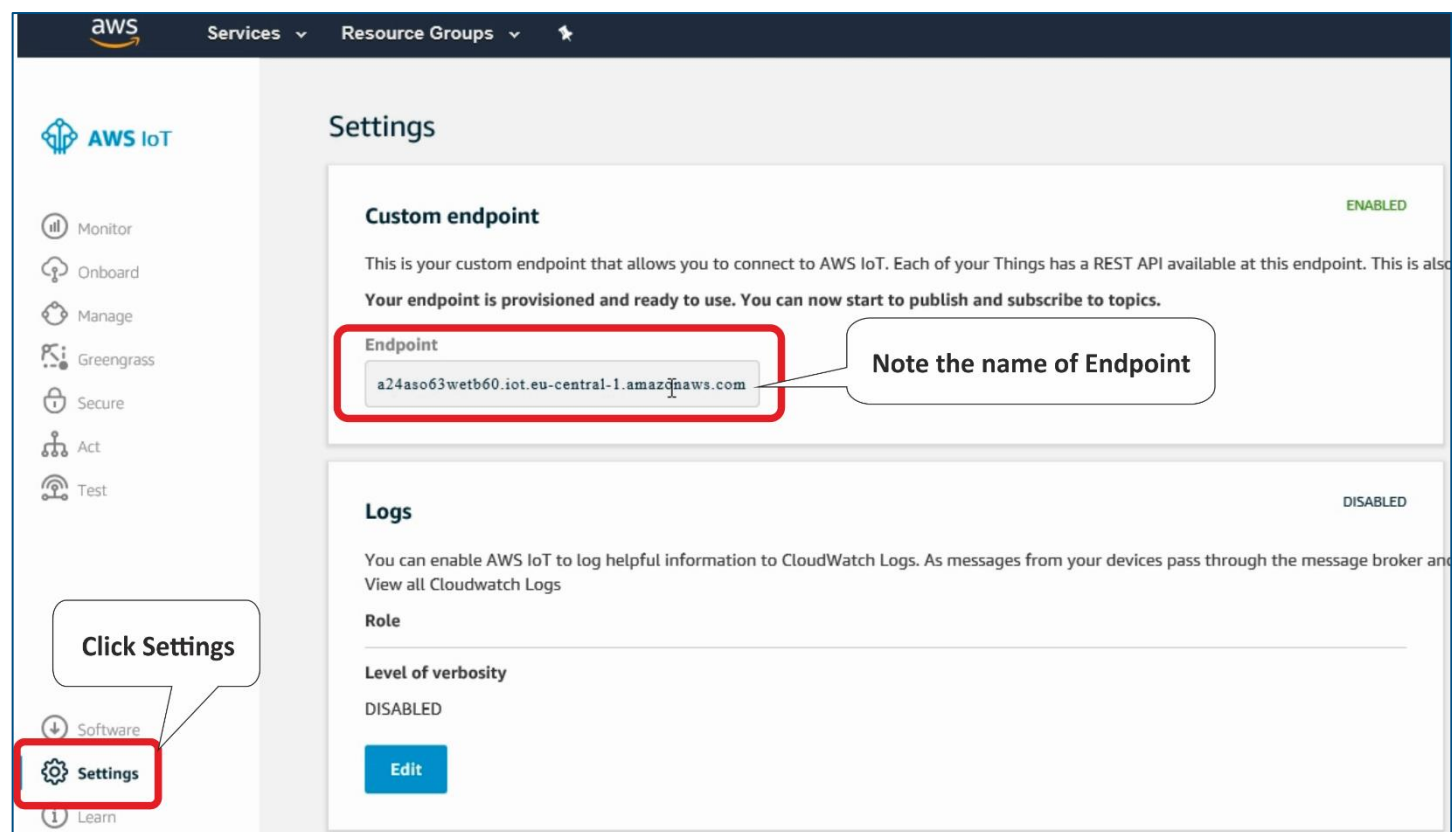
Note: You can choose your own name. In that case in later steps, you need to use the given name.



The message Connected successfully is shown automatically after finishing the process of configuring a device. Next, click Done.



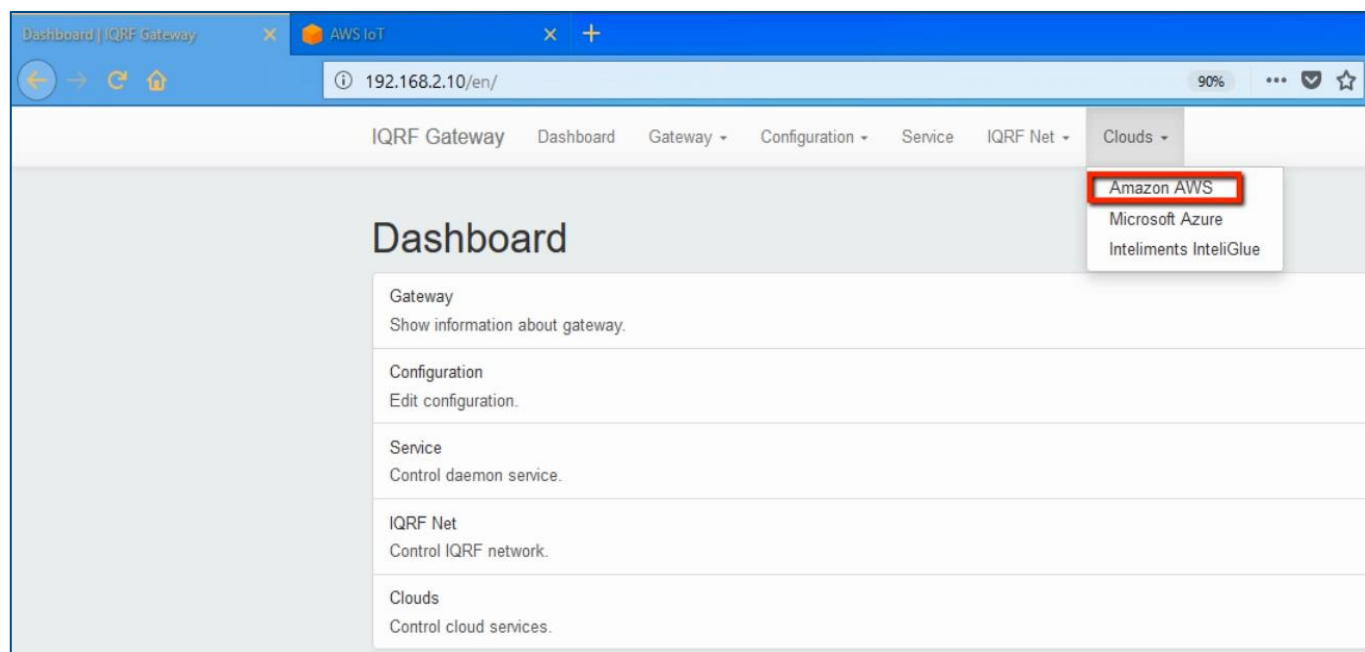
In the **Settings**, note the name of your **endpoint** because you will need it for the UP board configuration.



Files **rootCA.pem** (root certificate), **IQRF_Gateway.private.key** (private key file), and **IQRF_Gateway.cert.pem** (certificate file) should be already unzipped. We will transfer them to the UP board through the IQRF Gateway Daemon web application.

In the web browser on your computer, insert the IP address of your UP board, and login to it as *admin* with password *iqrf*. Ask your network administrator how to find out your IP address or you can use common network tools.

In the **IQRF Gateway Daemon web application**, click on the **Amazon AWS** item in the **Clouds** menu.



Enter the name of the **Endpoint** (find it in Settings of your AWS IoT). Select **rootCA.pem** as a Root CA certificate, **IQRF_Gateway.cert.pem** as a Certificate and **IQRF_Gateway.private.key** as a Private key file. Save the configuration.

Note: If you named your virtual device in AWS with a different name, names of files contain this name instead of IQRF_Gateway.

Add new MQTT interface

Endpoint
a24aso63wetb60.iot.eu-central-1.amazonaws.com

Root CA certificate
ice_package\rootCA.pem Procházet...
Select rootCA.pem file

Certificate
IQRF_Gateway.cert.pem Procházet...
Select IQRF_Gateway.cert.pem - certificate file

Private key
IQRF_Gateway.private.key Procházet...
Select IQRF_Gateway.private.key - private key file

Save

Inspect the new MQTT interface for AWS.

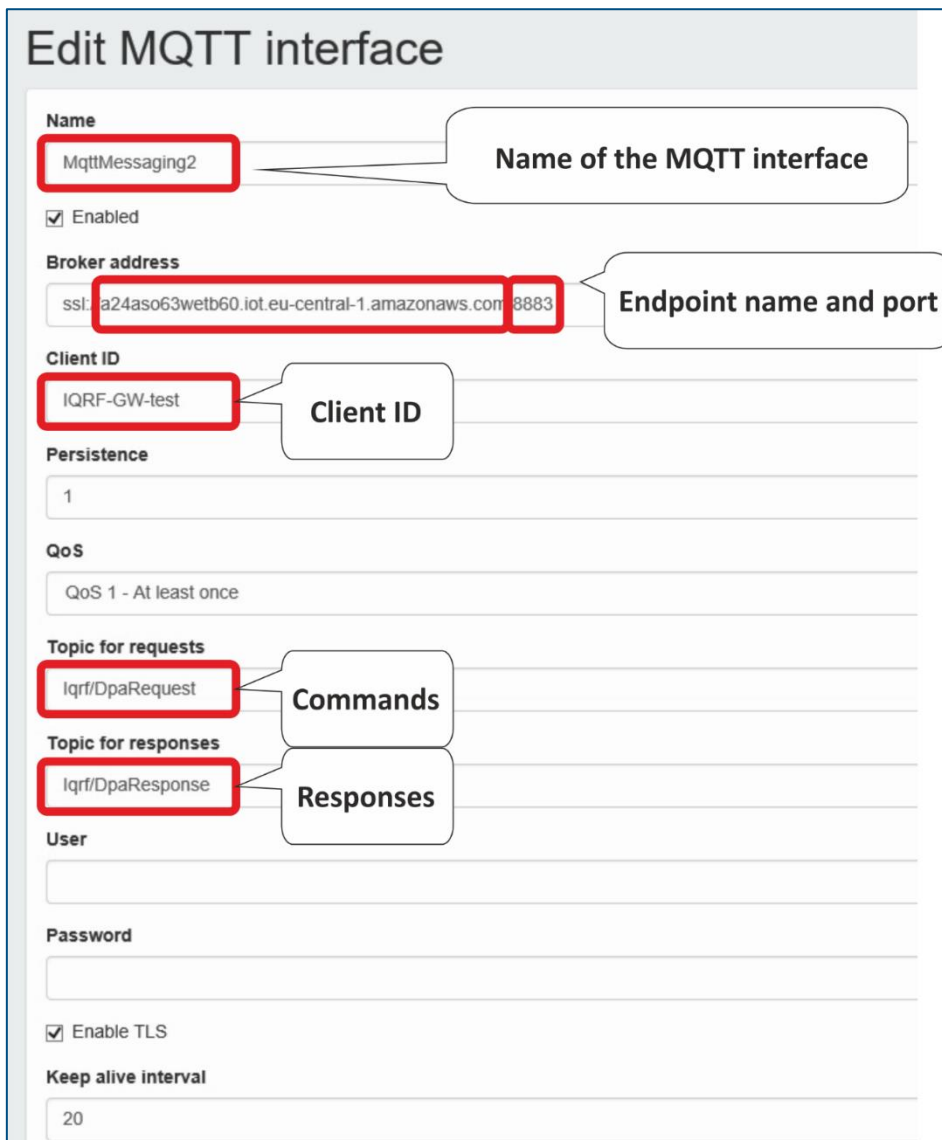
MQTT interface

Name	Broker	Client ID	TLS	Enabled	Edit	Remove
MqttMessaging1	tcp://127.0.0.1:1883	Local-app	✓	✓		
MqttMessaging2	ssl://a24aso63wetb60.iot.eu-central-1.amazonaws.com:8883	IQRF-GW-test	✓	✓		

Add

Edit the MQTT interface for AWS

Address of the **endpoint** goes after the **SSL** protocol and at the end of Broker address is the port number **8883**.
lqrf/DpaRequest is set as the topic for commands, and **lqrf/DpaResponse** is set as the topic for responses.



Edit MQTT interface

Name
MqttMessaging2
Name of the MQTT interface

☒ Enabled

Broker address
ssl://a24aso63wetb60.iot.eu-central-1.amazonaws.com:8883
Endpoint name and port

Client ID
IQRF-GW-test
Client ID

Persistence
1

QoS
QoS 1 - At least once

Topic for requests
lqrf/DpaRequest
Commands

Topic for responses
lqrf/DpaResponse
Responses

User

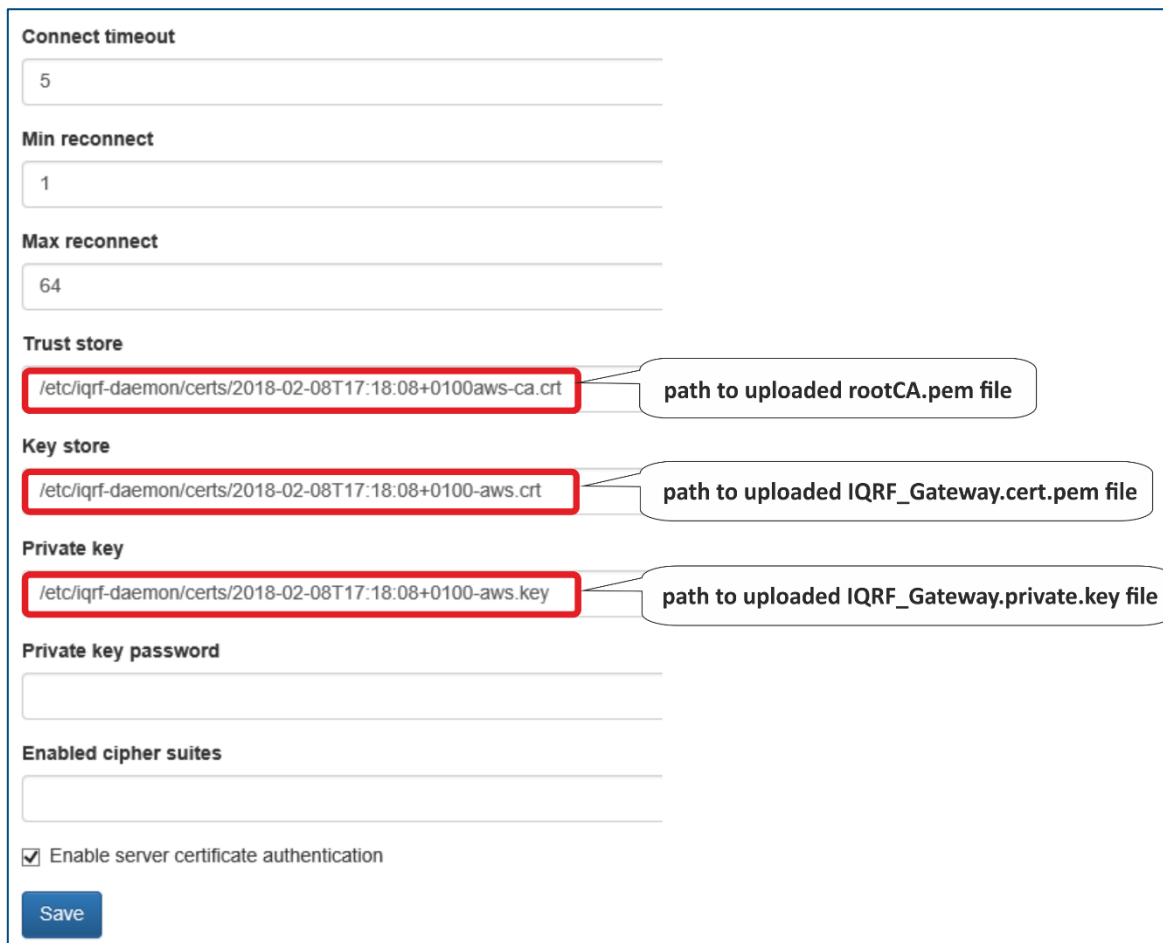
Password

☒ Enable TLS

Keep alive interval
20

Note: your files and name of the endpoint may differ from the names shown in the pictures.

There are the **timeout**, the **minimum**, and **maximum** connections set, and the path to the uploaded files that set up a secure connection between the gateway and the cloud. Check the **Enable server certificate authentication** item.



Connect timeout
5

Min reconnect
1

Max reconnect
64

Trust store
/etc/iqrf-daemon/certs/2018-02-08T17:18:08+0100aws-ca.crt
path to uploaded rootCA.pem file

Key store
/etc/iqrf-daemon/certs/2018-02-08T17:18:08+0100-aws.crt
path to uploaded IQRF_Gateway.cert.pem file

Private key
/etc/iqrf-daemon/certs/2018-02-08T17:18:08+0100-aws.key
path to uploaded IQRF_Gateway.private.key file

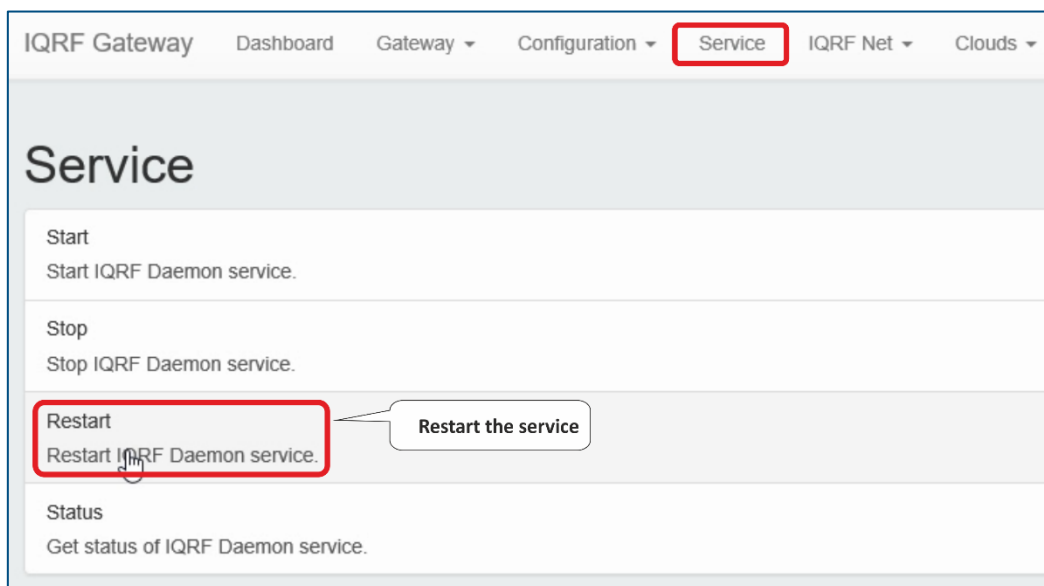
Private key password

Enabled cipher suites

☒ Enable server certificate authentication

Save

Restart IQRF Gateway Daemon. After restarting, check the status of the UP board if the selected services are running.



Service

Start
Start IQRF Daemon service.

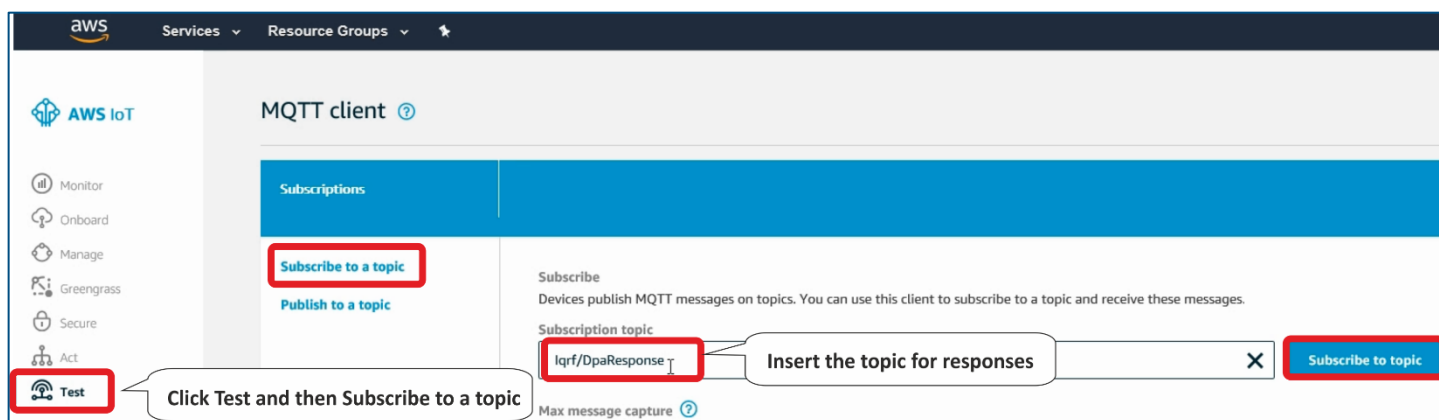
Stop
Stop IQRF Daemon service.

Restart
Restart IQRF Daemon service.
Restart the service

Status
Get status of IQRF Daemon service.

4 Test the connection

In the web browser on your computer, in AWS IoT, click **Test**. Enter the **lqrf/DpaResponse** to the Response topic to retrieve the gateway responses and click on **Subscribe to topic**.



To send commands from the cloud to the gateway, set the **lqrf/DpaRequest** as the topic for requests. Gateway will expect commands in this topic.



Insert a DPA packet in the JSON format into the text box and click on **Publish to topic**. In our example, we sent a command to turn on the red LED on the coordinator.

```
{
  "ctype": "dpa",
  "type": "raw",
  "msgid": "1510754980",
  "request": "00.00.06.01.FF.FF",
  "request_ts": "",
  "confirmation": "",
  "confirmation_ts": "",
  "response": "",
  "response_ts": ""
}
```

In the *“request”* item, you can insert other DPA commands for the network control and monitoring. You can find these commands in macros for IoT Starter Kit or you can set up them in the Terminal window in the IQRF IDE.

Examples:

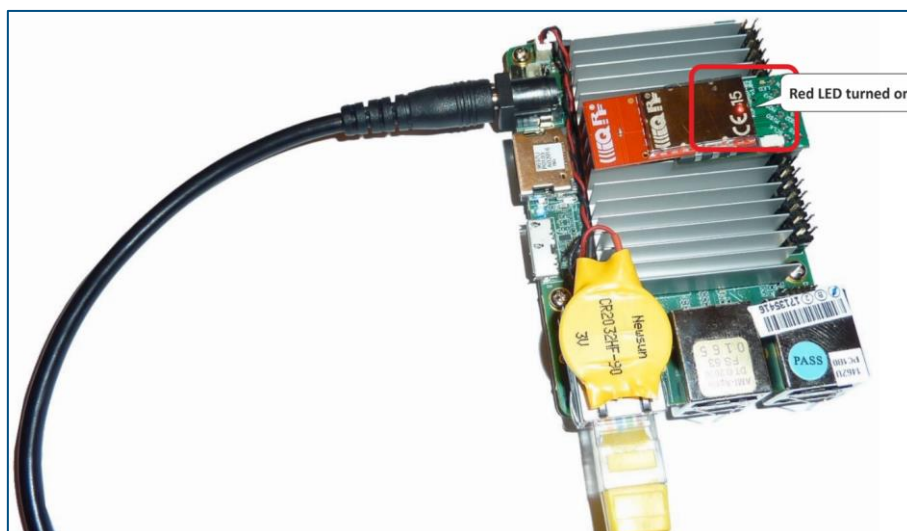
- Collecting all sensoric data from the Node #1 with the connected DDS-SE kit:
{"ctype": "dpa", "type": "raw", "msgid": "1510754980", "request": "01.00.5E.01.FF.FF.FF.FF.FF.FF", "request_ts": "", "confirmation": "", "confirmation_ts": "", "response": "", "response_ts": ""}
- Turning on both relays on the Node #2 with connected DDC-RE kit:
{"ctype": "dpa", "type": "raw", "msgid": "1510754980", "request": "02.00.4B.00.FF.FF.0C.00.00.00.01.01", "request_ts": "", "confirmation": "", "confirmation_ts": "", "response": "", "response_ts": ""}
- Getting temperature from the Node #3:
{"ctype": "dpa", "type": "raw", "msgid": "1510754980", "request": "03.00.0A.00.FF.FF", "request_ts": "", "confirmation": "", "confirmation_ts": "", "response": "", "response_ts": ""}

For more information about macros and the IQRF network read the [IoT Starter Kit – Part 1: Build your IQRF network](#).

We can see that the gateway picked up and executed the command, and sent a confirmation with "No Error" into the **Iqrf/DpaResponse** topic.

The screenshot shows the AWS IoT console interface. At the top, the 'Publish' section is active, showing a message published to the 'Iqrf/DpaRequest' topic. The message content is a JSON object: {"ctype": "dpa", "type": "raw", "msgid": "1510754980", "request": "00.00.06.01.FF.FF", "request_ts": "", "confirmation": "", "confirmation_ts": "", "response": "", "response_ts": ""}. A red box highlights this JSON, with a callout bubble stating 'DPA command in JSON format'. To the right of the message input field, a 'Publish to topic' button is highlighted with a red box. Below this, the 'Iqrf/DpaResponse' topic is shown with a message received at 'Nov 21, 2017 8:30:52 AM +0100'. The response is a JSON object: {"ctype": "dpa", "type": "raw", "msgid": "1510754980", "request": "00.00.06.01.FF.FF", "request_ts": "2017-11-21 07:30:52.71517", "confirmation": "", "confirmation_ts": "", "response": "00.00.06.81.00.00.00.00", "response_ts": "2017-11-21 07:30:52.119054", "status": "STATUS_NO_ERROR"}. A red box highlights this response JSON, with a callout bubble stating 'Response'.

We can visually double check the result of this command. The red LED turned on.



5 Summary

The bidirectional communication between IQRF network and the Amazon Web Services is up and running. Now it's just up to you to use it for your own IoT solution. In next parts, we will show you how to add other sensors and actuators of our industrial partners (CO₂ sensor, wirelessly controlled power socket etc.).

IQRF transceivers have from a factory these default settings: TX power: 7, RX filter: 0, RF channel A: 52. Because of those settings (TX power, RX filter), you can cover with the wireless IQRF signal an area of 500 m radius in open space.