

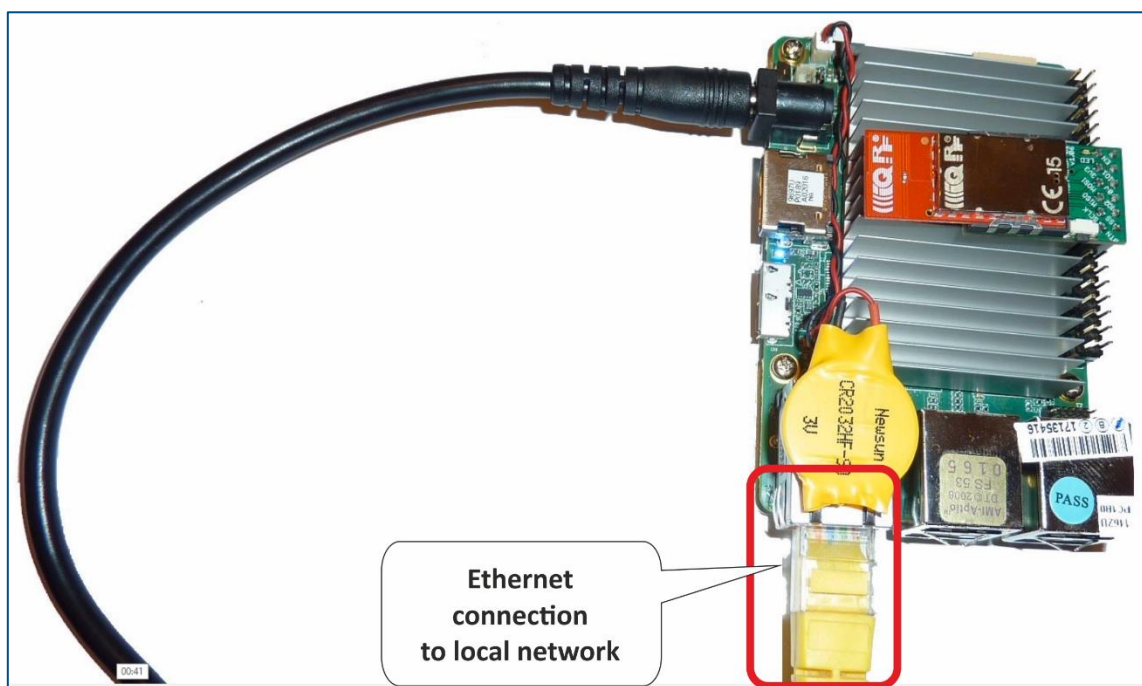
# UP-IQRF IoT Starter Kit – Part 3: Connect to the cloud – AWS IoT

**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.

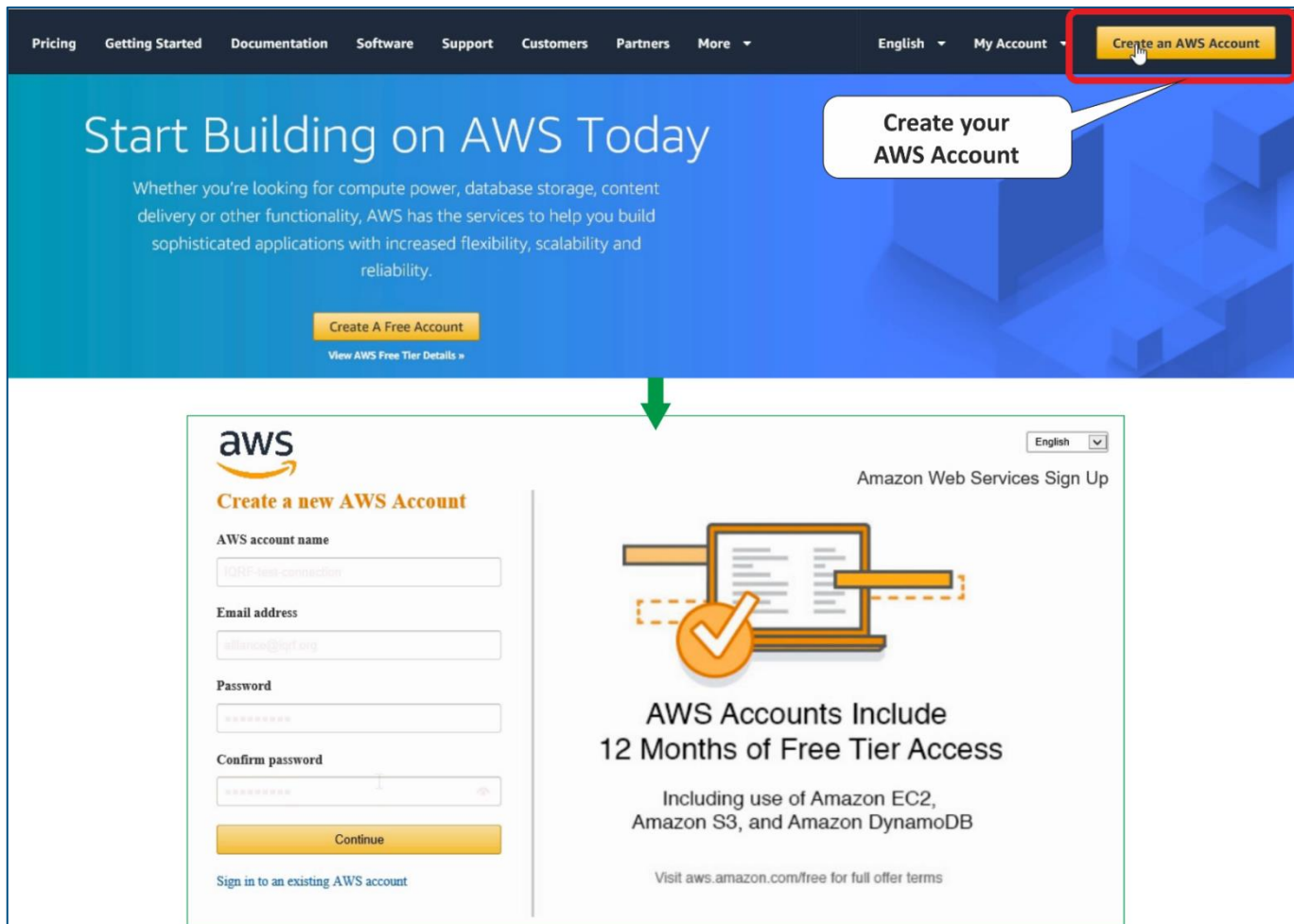
## 1 Local network

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](https://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.



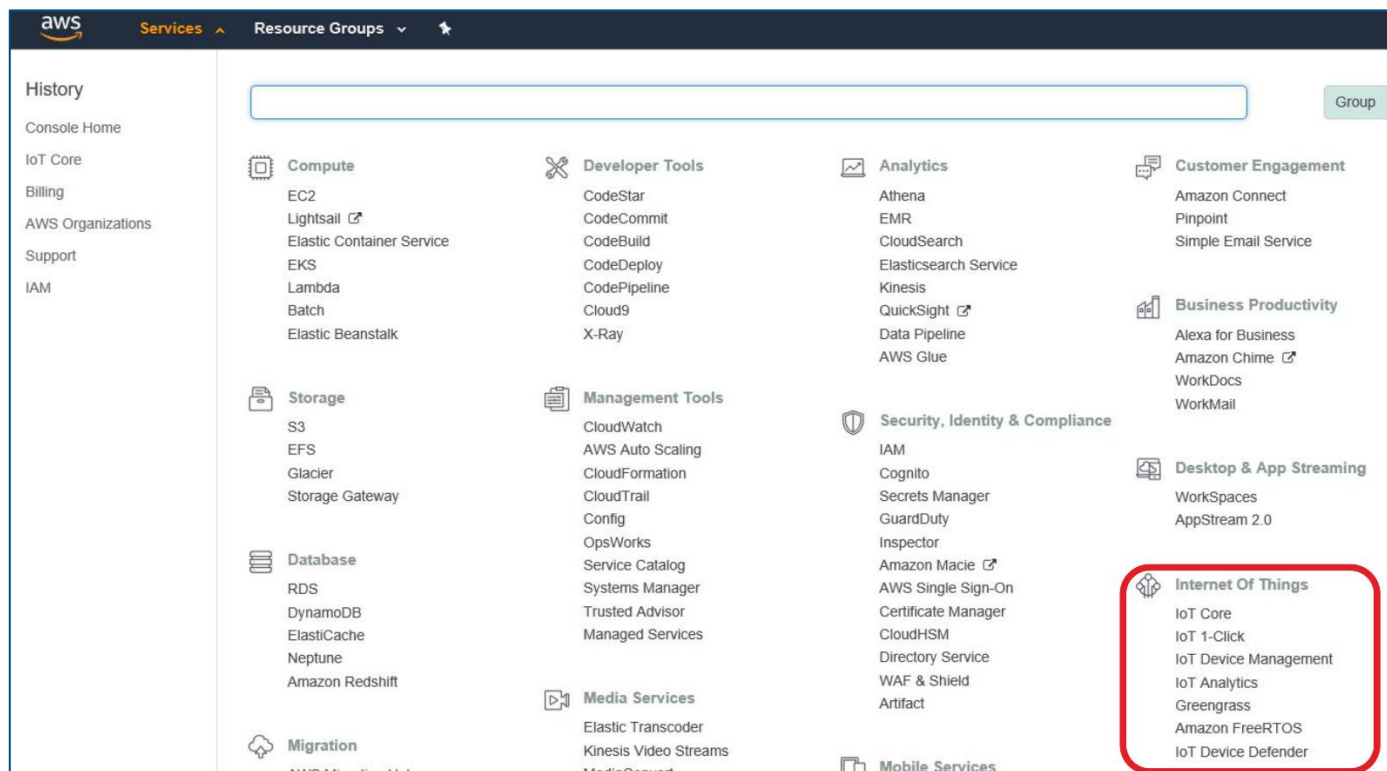
The screenshot shows the AWS website. At the top, there is a navigation bar with links: Pricing, Getting Started, Documentation, Software, Support, Customers, Partners, and More. On the right, there are links for English, My Account, and a highlighted 'Create an AWS Account' button. Below the navigation 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.

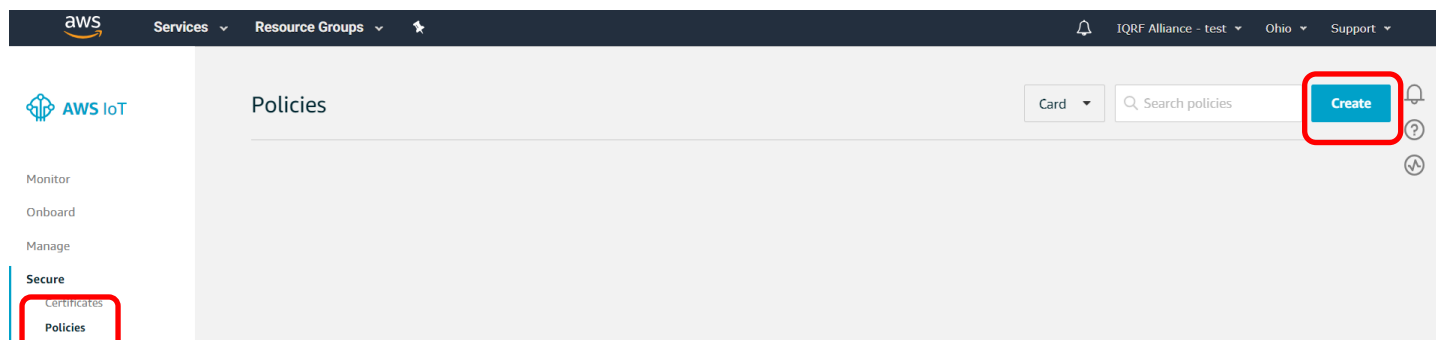
In **Services**, in the **Internet of Things** section of AWS, find **IoT Core**.

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



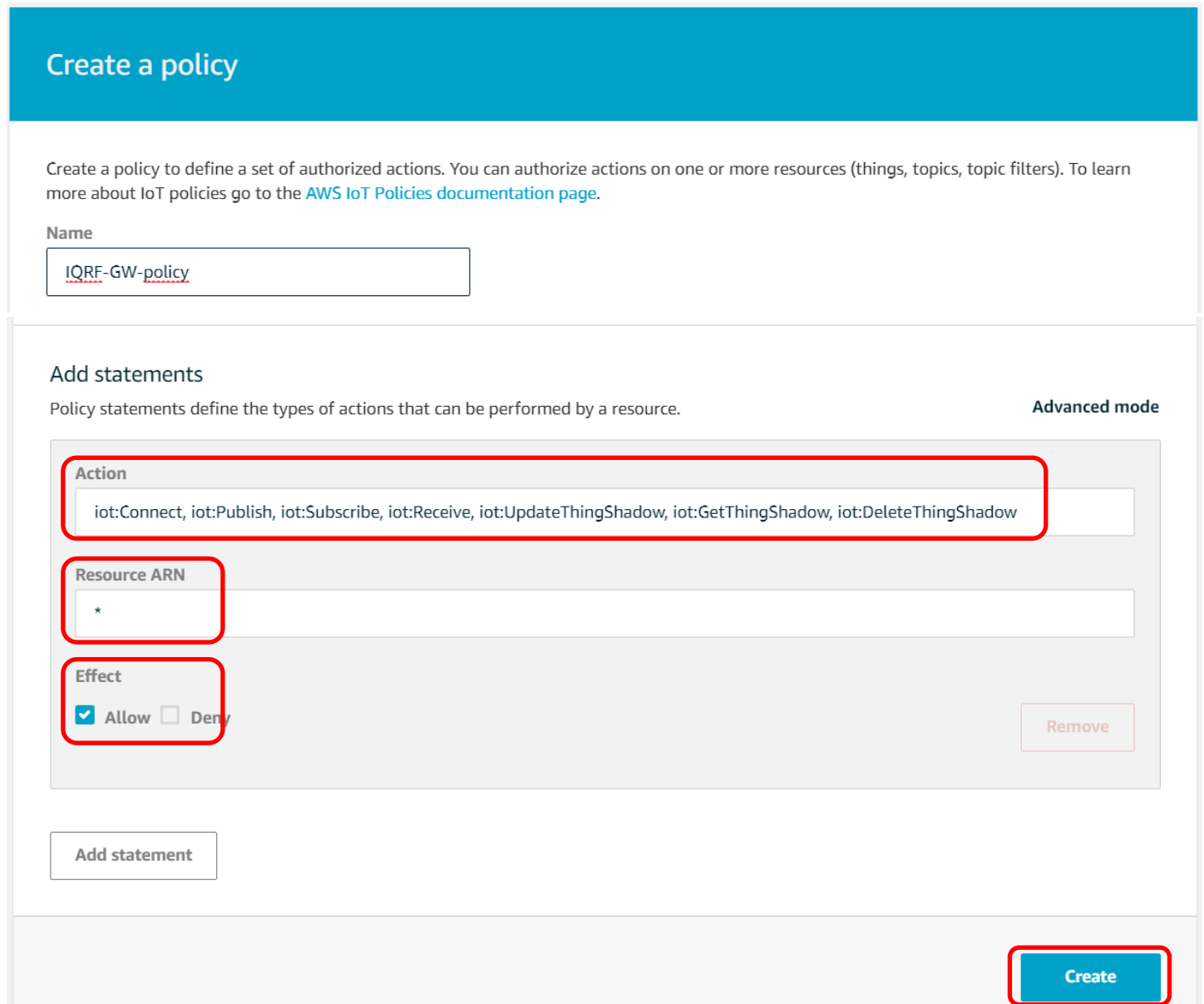
Create a default policy for the device. This step involves creating a default policy for the new device, skip if an existing policy is already available.

Access the main screen of the console and select **Secure -> Policies** from the left side menu and then press the **Create** button, in the top right area of the screen.



Fill the form as follows and then press the **Create** button:

- Action -> iot:Connect, iot:Publish, iot:Subscribe, iot:Receive, iot:UpdateThingShadow, iot:GetThingShadow, iot:DeleteThingShadow
- Resource ARN -> \*
- Effect -> Allow



### Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

**Name**

IQRF-GW-policy

**Add statements**

Policy statements define the types of actions that can be performed by a resource. Advanced mode

**Action**

iot:Connect, iot:Publish, iot:Subscribe, iot:Receive, iot:UpdateThingShadow, iot:GetThingShadow, iot:DeleteThingShadow

**Resource ARN**

\*

**Effect**

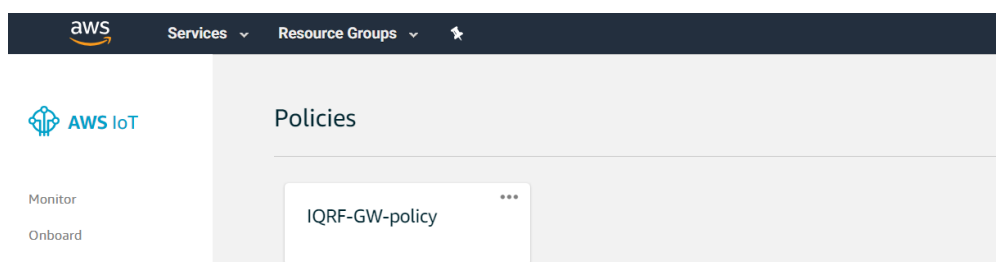
☒ Allow ☐ Deny

Remove

Add statement

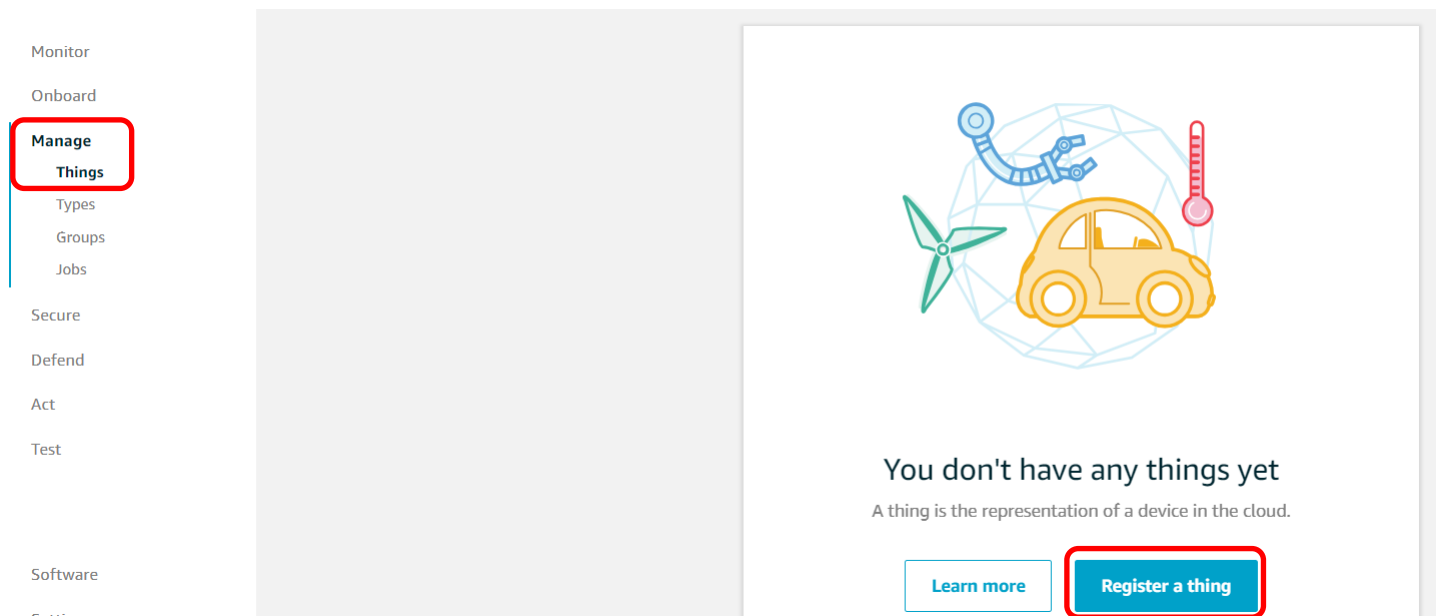
Create

This will create a policy that allows a device to connect to the platform, publish/subscribe on any topic and manage its *thing shadow*.

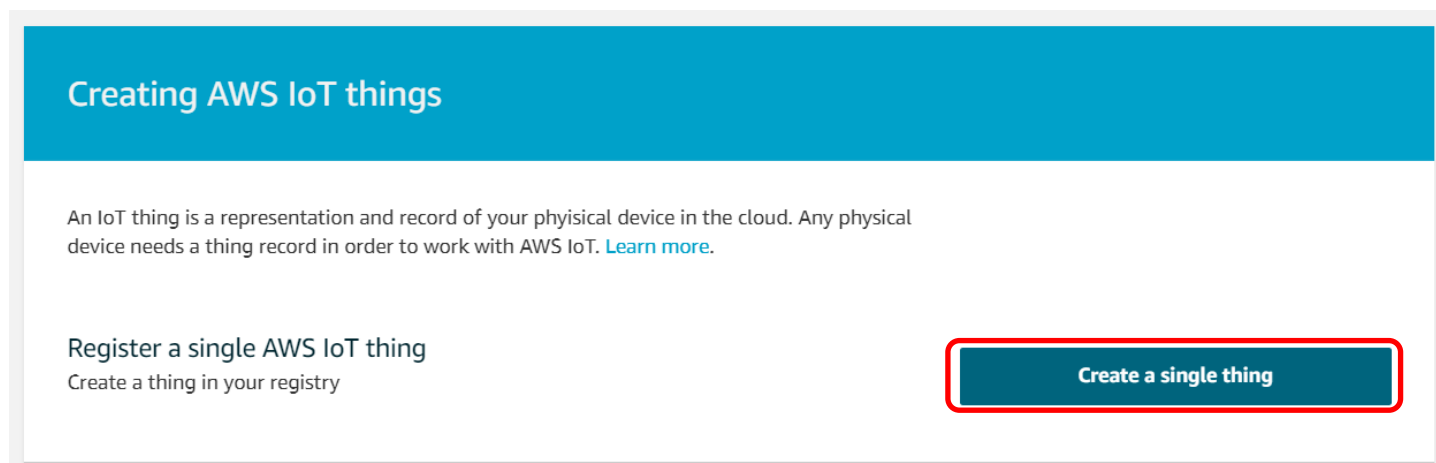


Register a new device.

Devices on the AWS IoT platform are called things. In order to register a new thing, select **Manage** -> **Things** from the left side menu and then press the **Create** button, in the top right section of the screen (or press the **Register a thing** button).



Select **Create a single thing**.



Enter a name for the new device and then press the **Next** button.

CREATE A THING

STEP 1/3

Add your device to the thing registry

This step creates an entry in the thing registry and a thing shadow for your device.

Name

IQRF-GW

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected

Create a type

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups /

Create group Change

Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key

Provide an attribute key, e.g. Manufacturer

Value

Provide an attribute value, e.g. Acme-Corporation

Clear

Add another

Show thing shadow ▼

Cancel

Back

Next

Create a new certificate for the device.

The AWS IoT platform uses SSL mutual authentication, for this reason it is necessary to download a public/private key pair for the device and a server certificate. Click on **Create certificate** to quickly generate a new certificate for the new device. Certificates can be managed later on by clicking on **Secure** -> **Certificates**, in the left part of the console.

CREATE A THING

STEP 2/3

### Add a certificate for your thing

A certificate is used to authenticate your device's connection to AWS IoT.

**One-click certificate creation (recommended)**

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

**Create certificate**

**Create with CSR**

Upload your own certificate signing request (CSR) based on a private key you own.

**Create with CSR**

**Use my certificate**

Register your CA certificate and use your own certificates for one or many devices.

**Get started**

**Skip certificate and create thing**

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

**Create thing without certificate**

Download the device SSL keys.

### Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	52e260e9d3.cert.pem	<a href="#">Download</a>
A public key	52e260e9d3.public.key	<a href="#">Download</a>
A private key	52e260e9d3.private.key	<a href="#">Download</a>

Successfully created thing.

Successfully generated certificate. Please download certificate files.

Download the 3 files listed in the table and store them in a safe place, they will be needed later.

Press the **Activate** button, and then on **Attach a policy**.

You also need to download a root CA for AWS IoT:  
A root CA for AWS IoT [Download](#)

**Activate**

[Cancel](#) [Done](#) **Attach a policy**

Assign the default policy to the device.

Select the desired policy and then click on **Register thing**.

CREATE A THING

### Add a policy for your thing

STEP 3/3

Select a policy to attach to this certificate:

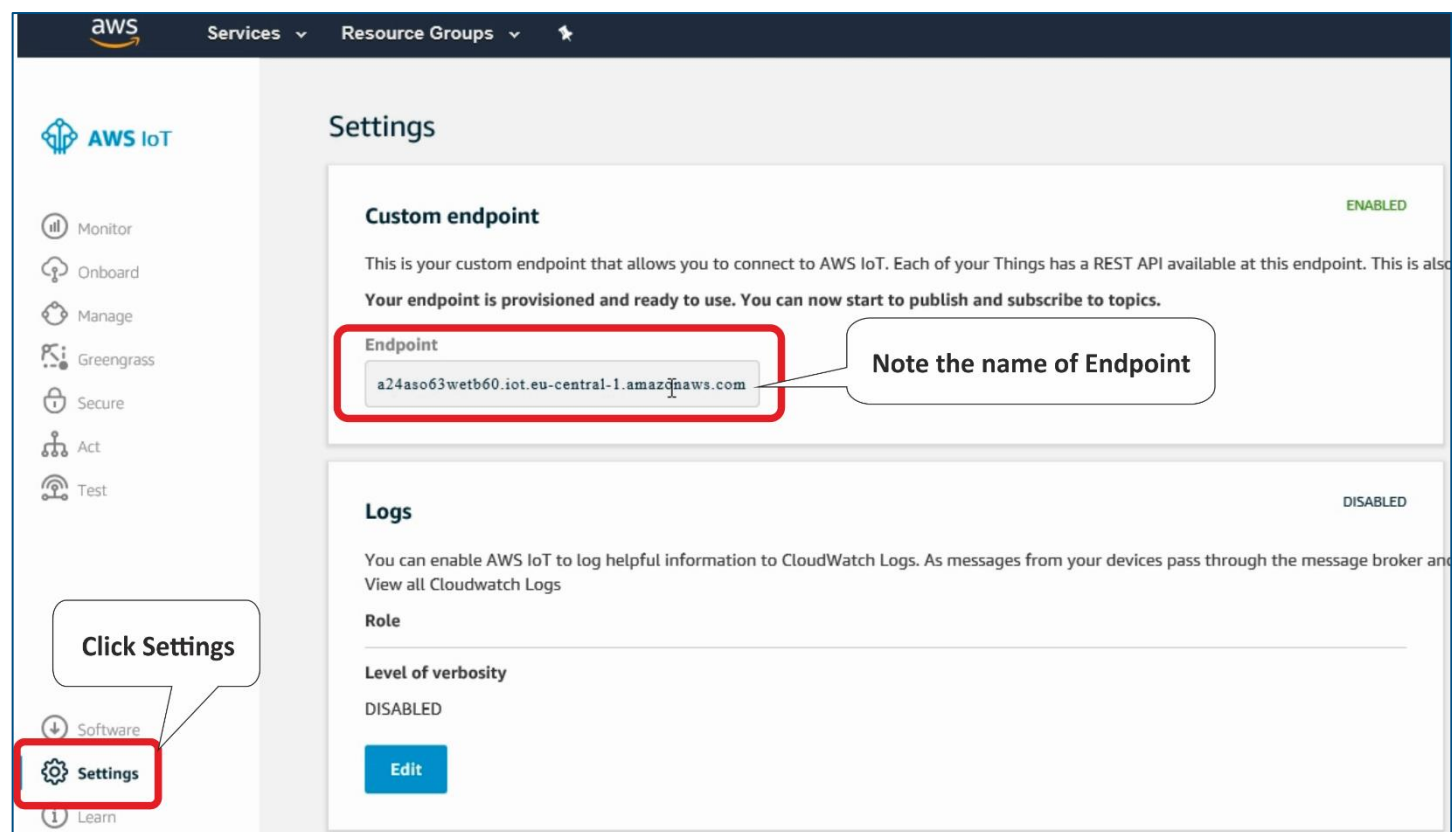
☒ IQRF-GW-policy [View](#)

1 policy selected

**Register Thing**



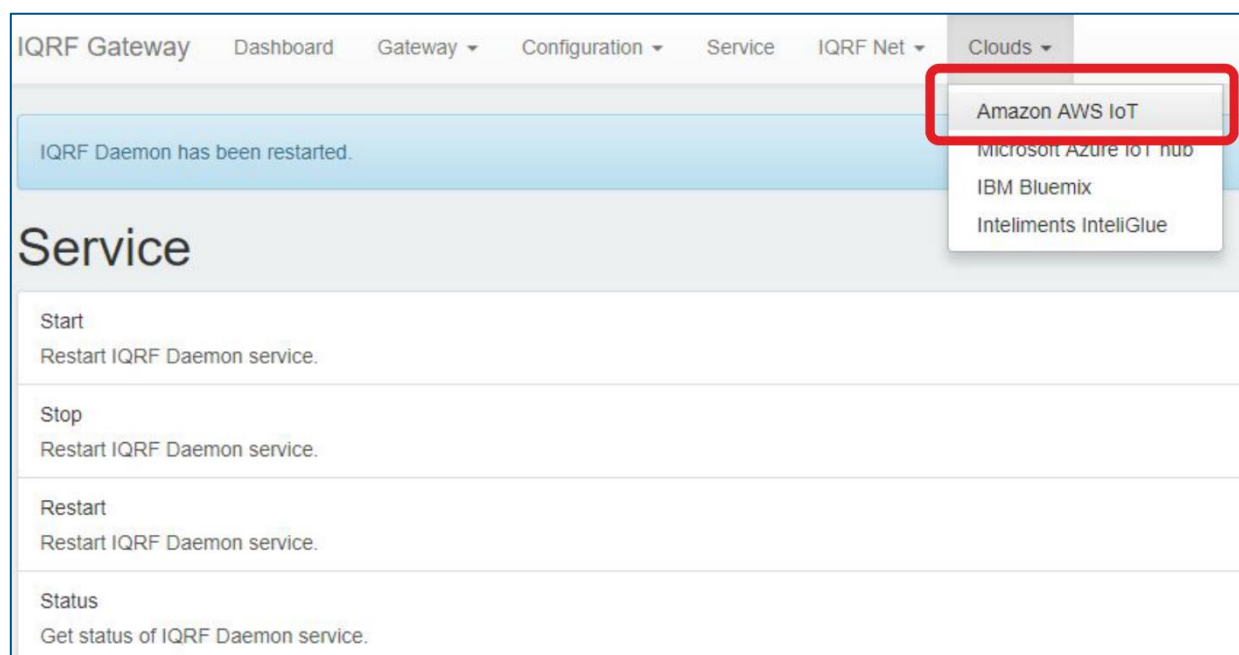
In the **Settings**, copy the name of your **endpoint**, you will need it for the UP board configuration in next steps.



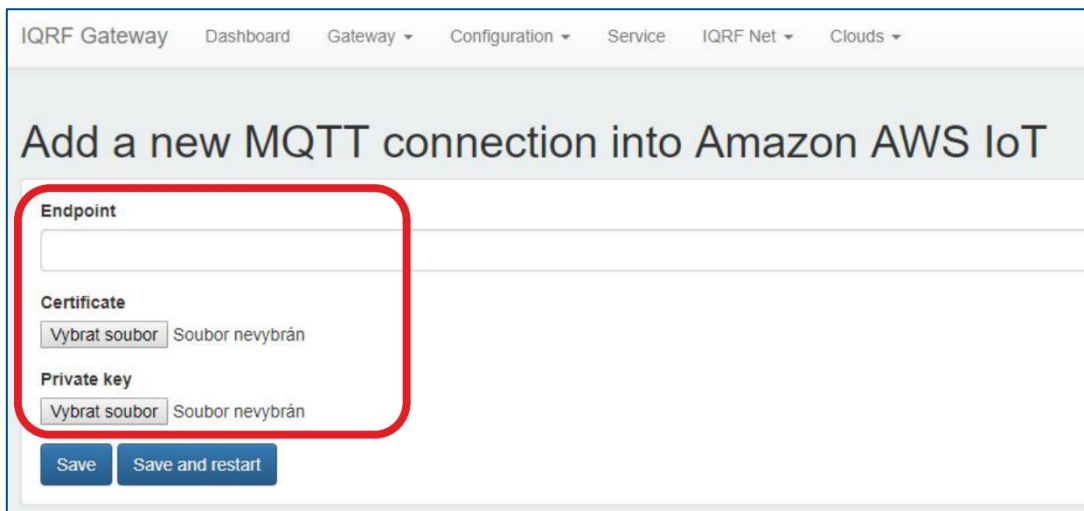
Files private key and certificate file (SSL keys) should be already downloaded. You will transfer them to the UP board through the IQRF Gateway Daemon web application.

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 IoT** item in the **Clouds** menu.



Paste the name of the **Endpoint** (you have copied it from Settings of your AWS IoT). Select certificate and a private key file. Save the configuration.



IQRF Gateway Dashboard Gateway Configuration Service IQRF Net Clouds

## Add a new MQTT connection into Amazon AWS IoT

**Endpoint**







**Certificate**  
Vybrat soubor Soubor nevybrán

**Private key**  
Vybrat soubor Soubor nevybrán

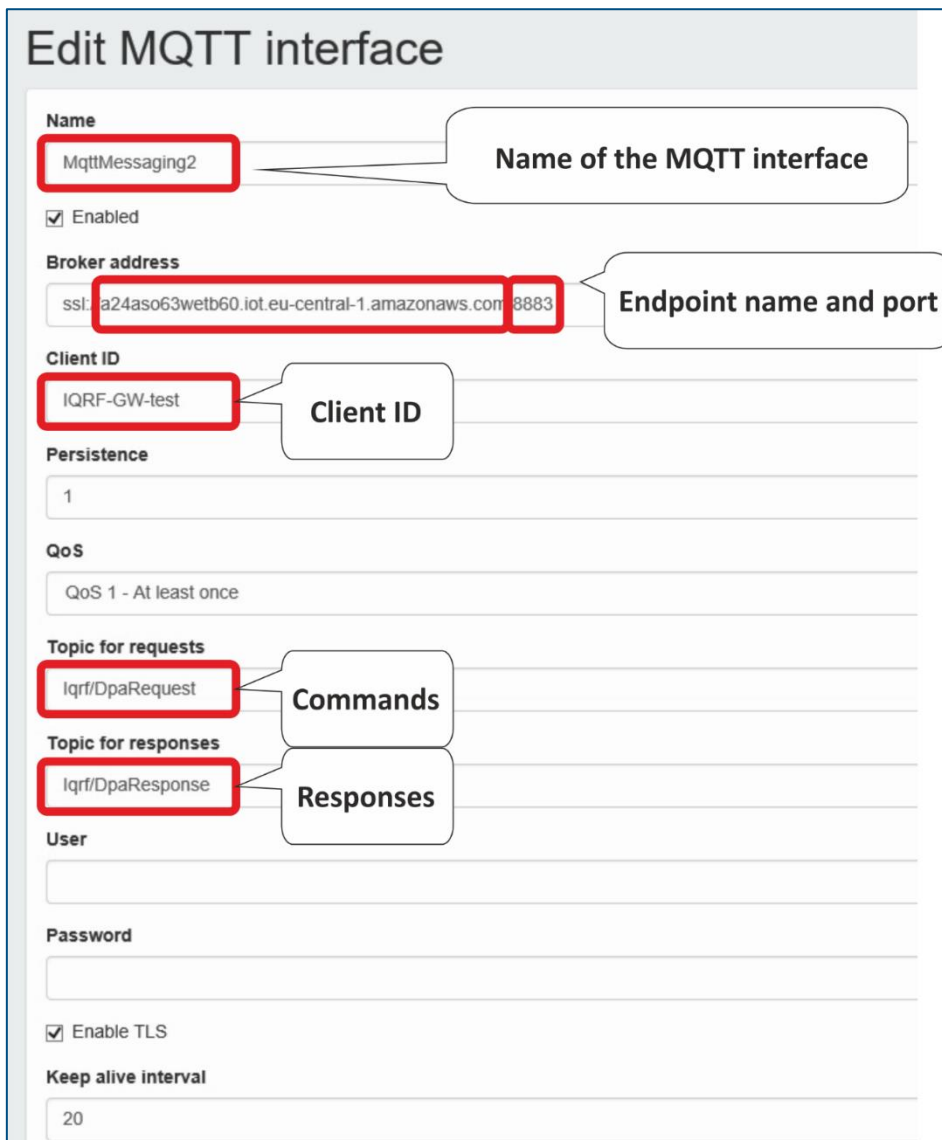
Save Save and restart

Inspect the new MQTT interface for AWS.

## MQTT interface

Name	Broker address	Client ID	Enabled TLS	Enabled	Edit	Remove
MqttMessaging1	tcp://127.0.0.1:1883	IqrfDpaMessaging1	✗	✓		
MqttMessaging2	tcp://iot.eclipse.org:1883	IqrfDpaMessaging2	✗			
MqttMessagingAws	ssl://arsz5u1hab560-ats.iot.us-east-2.amazonaws.com:8883	IqrfDpaMessaging1	✓	✓		

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.



The screenshot shows the 'Edit MQTT interface' configuration form. The following fields are highlighted with red boxes and callouts:

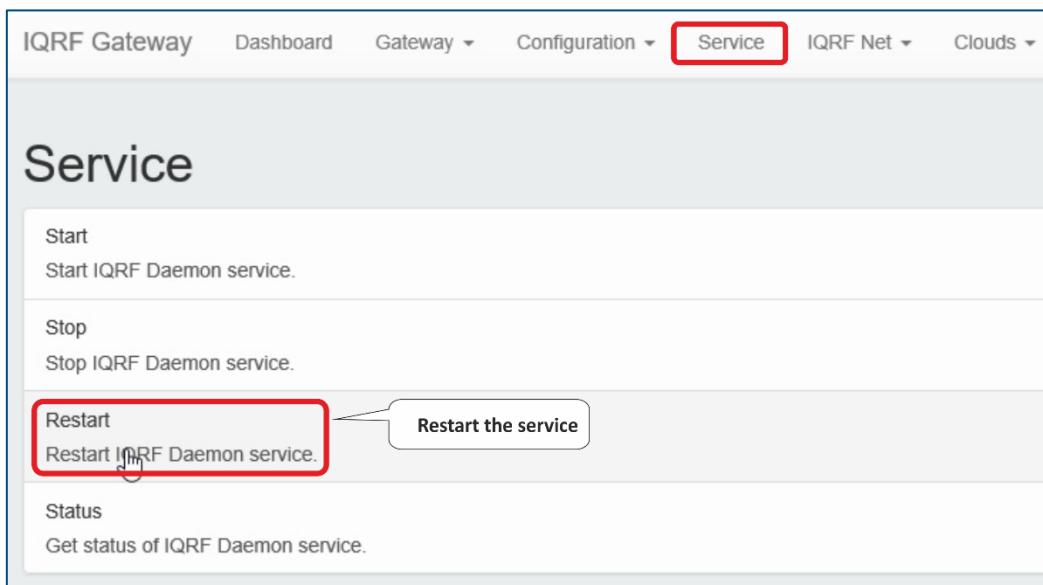
- Name:** 'MqttMessaging2' (Callout: Name of the MQTT interface)
- Enabled:** ☒
- Broker address:** 'ssl://a24aso63wetb60.iot.eu-central-1.amazonaws.com:8883' (Callout: Endpoint name and port)
- Client ID:** 'IQRF-GW-test' (Callout: Client ID)
- Persistence:** '1'
- QoS:** 'QoS 1 - At least once'
- Topic for requests:** 'lqrf/DpaRequest' (Callout: Commands)
- Topic for responses:** 'lqrf/DpaResponse' (Callout: Responses)
- User:** (empty field)
- Password:** (empty field)
- 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.

Connection timeout
<input type="text" value="5"/>
Minimal count of reconnects
<input type="text" value="1"/>
Maximal count of reconnects
<input type="text" value="64"/>
CA certificate
<input type="text" value="/etc/iqrf-daemon/certs/aws-ca.crt"/>
Certificate
<input type="text" value="/etc/iqrf-daemon/certs/2018-10-01T15:00:45+0200-aws.crt"/>
Private key
<input type="text" value="/etc/iqrf-daemon/certs/2018-10-01T15:00:45+0200-aws.key"/>

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.

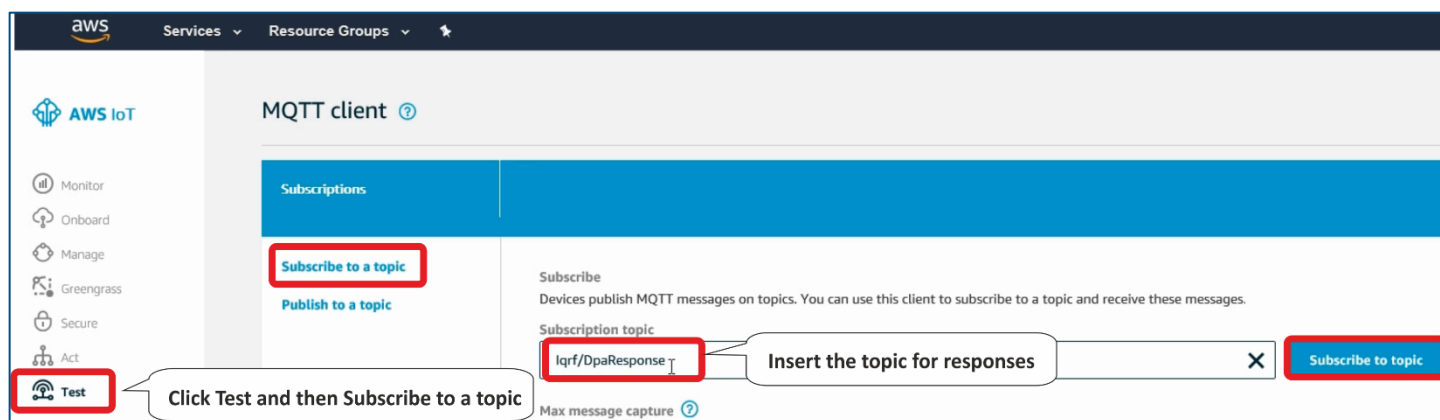
Stop  
Stop IQRF Daemon service.

**Restart**  
Restart IQRF Daemon service.

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": ""
}
```

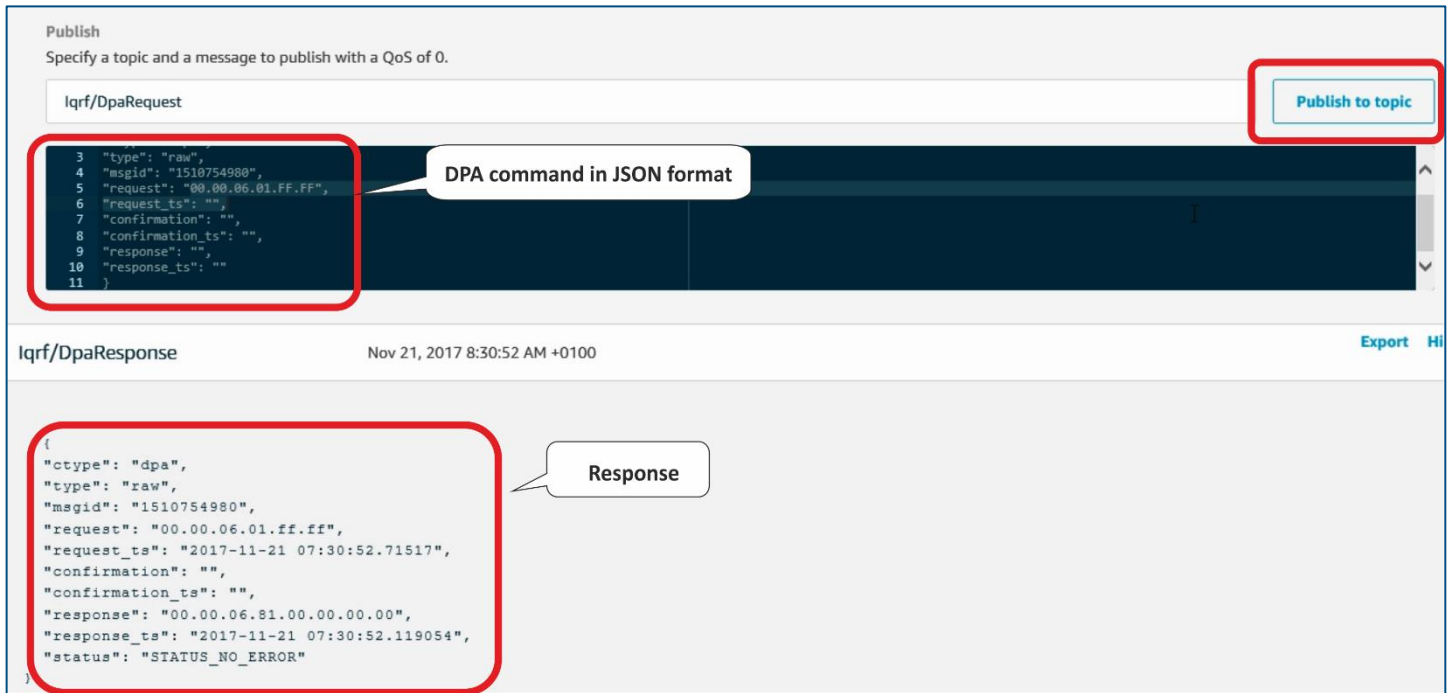
In the *“request”* item, you can insert other DPA commands for the network control and monitoring. You can find these commands in macros for IoT Starter Kit or you can set up them in the Terminal window in the IQRF IDE.

Examples:

- Collecting all sensoric data from the Node #1 with the connected DDS-SE kit: 01.00.5E.01.FF.FF.FF.FF.FF.
- Turning on both relays on the Node #2 with connected DDC-RE kit: 02.00.4B.00.FF.FF.0C.00.00.00.01.01.
- Getting temperature from the Node #3: 03.00.0A.00.FF.FF.

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 **lqrf/DpaResponse** topic.



The screenshot shows the AWS IoT console interface. At the top, the 'Publish' section is active, with a text input field containing 'lqrf/DpaRequest'. A red box highlights the 'Publish to topic' button. Below the input field, a code editor shows a JSON message: 

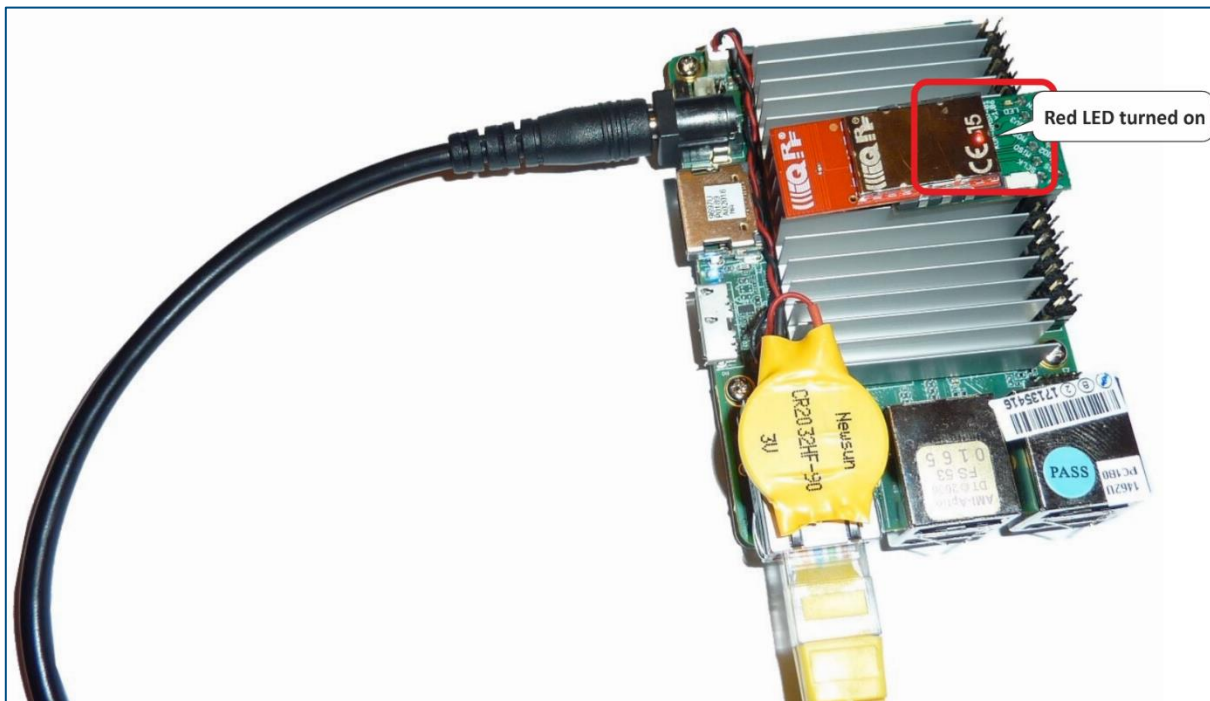
```
3 {
4   "type": "raw",
5   "msgid": "1510754980",
6   "request": "00.00.06.01.FF.FF",
7   "request_ts": "",
8   "confirmation": "",
9   "confirmation_ts": "",
10  "response": "",
11  "response_ts": ""
12 }
```

 A red box highlights this JSON, with a callout bubble stating 'DPA command in JSON format'. Below the code editor, the 'lqrf/DpaResponse' topic is selected, showing a message received on Nov 21, 2017 8:30:52 AM +0100. A red box highlights the response JSON: 

```
{
  "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 callout bubble points to this JSON with the text '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 (CO<sub>2</sub> 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.