

# IoT Starter Kit – Part 3c:

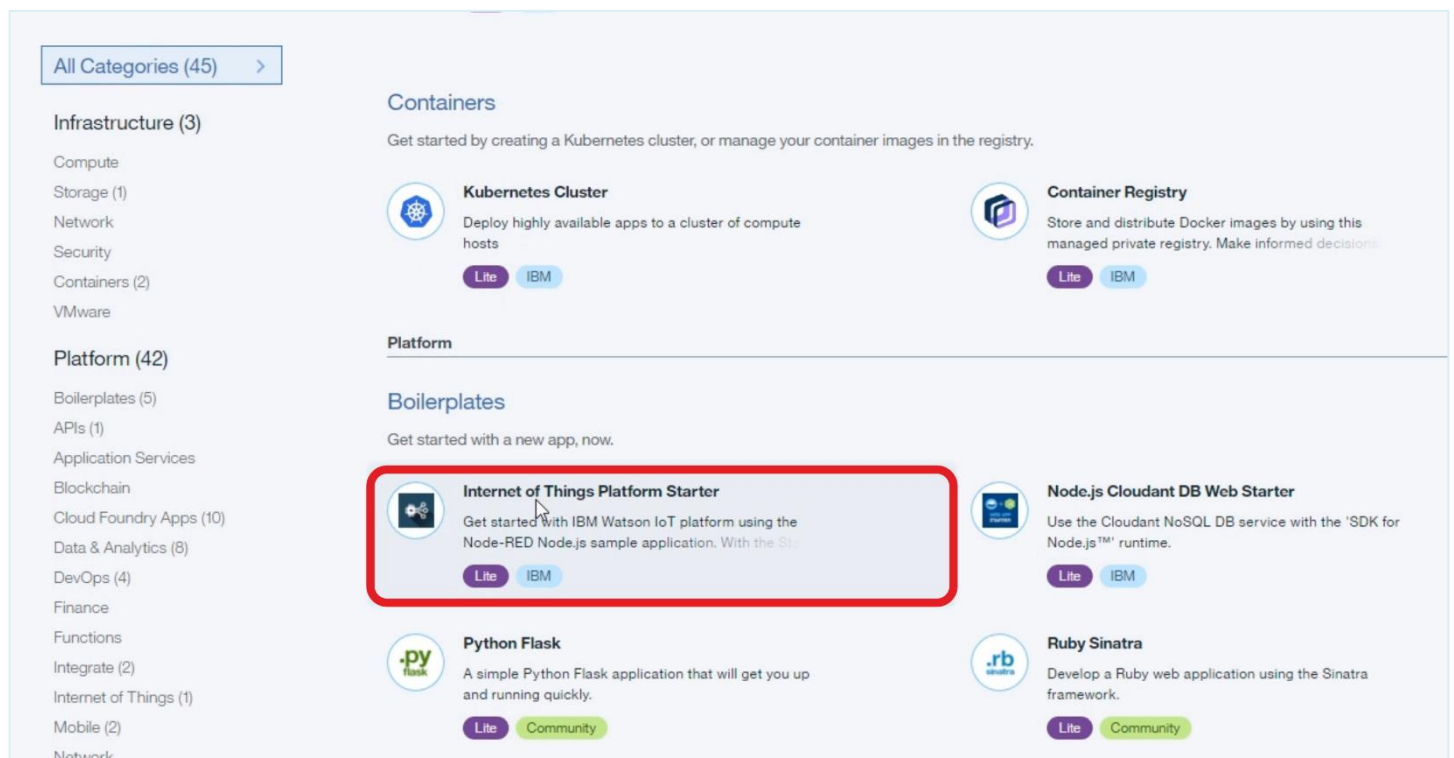
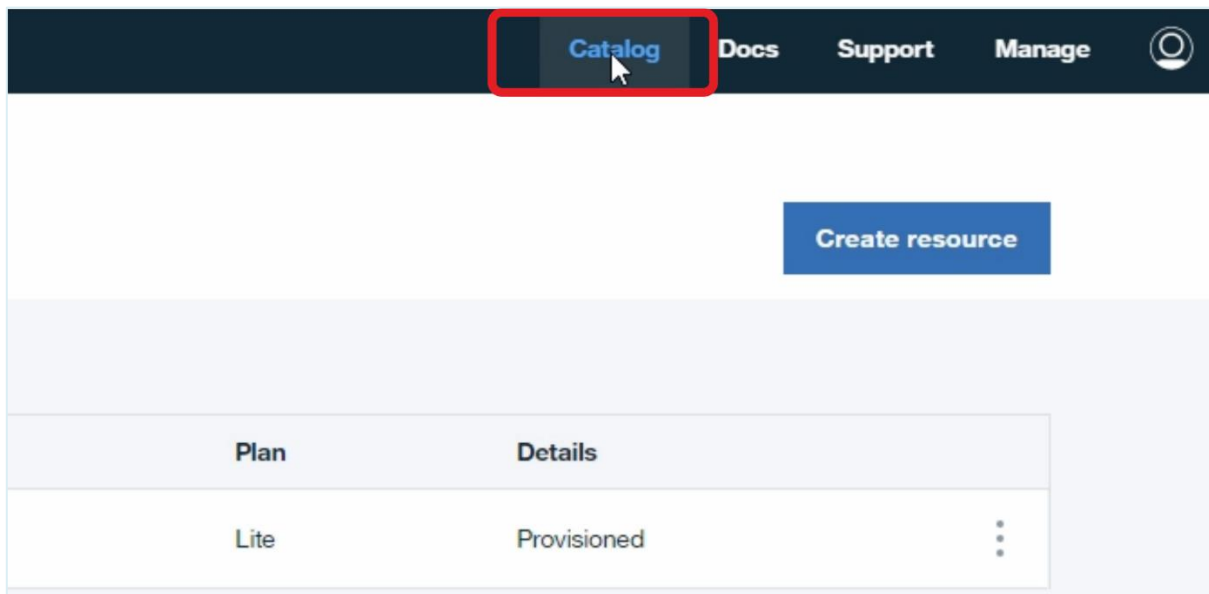
## How to connect to IBM Cloud

In this part we will connect an IQRF gateway to IBM Cloud. IBM provides developers some free services for a limited time and you don't have to enter any credit card details on the beginning. It is one of the possible clouds that you can get connected to from your IQRF Gateway Daemon using the MQTT channel.

### IBM Cloud and Watson IoT platform

Log in to your IBM Cloud account on [console.bluemix.net](https://console.bluemix.net).

Click on the **Catalog** button in the upright corner and find the **Internet of Things Platform Starter**.



Fill in this form to set up your cloud application. Type in a unique **app name**, select your **deployment location** and your **pricing plan** and click on **Create**.

App name:

Host name:

Domain:

Choose a region/location to deploy in:

Choose an organization:

Choose a space:

Selected Plan:

SDK for Node.js™

Cloudant NoSQL DB

Internet of Things Platform

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

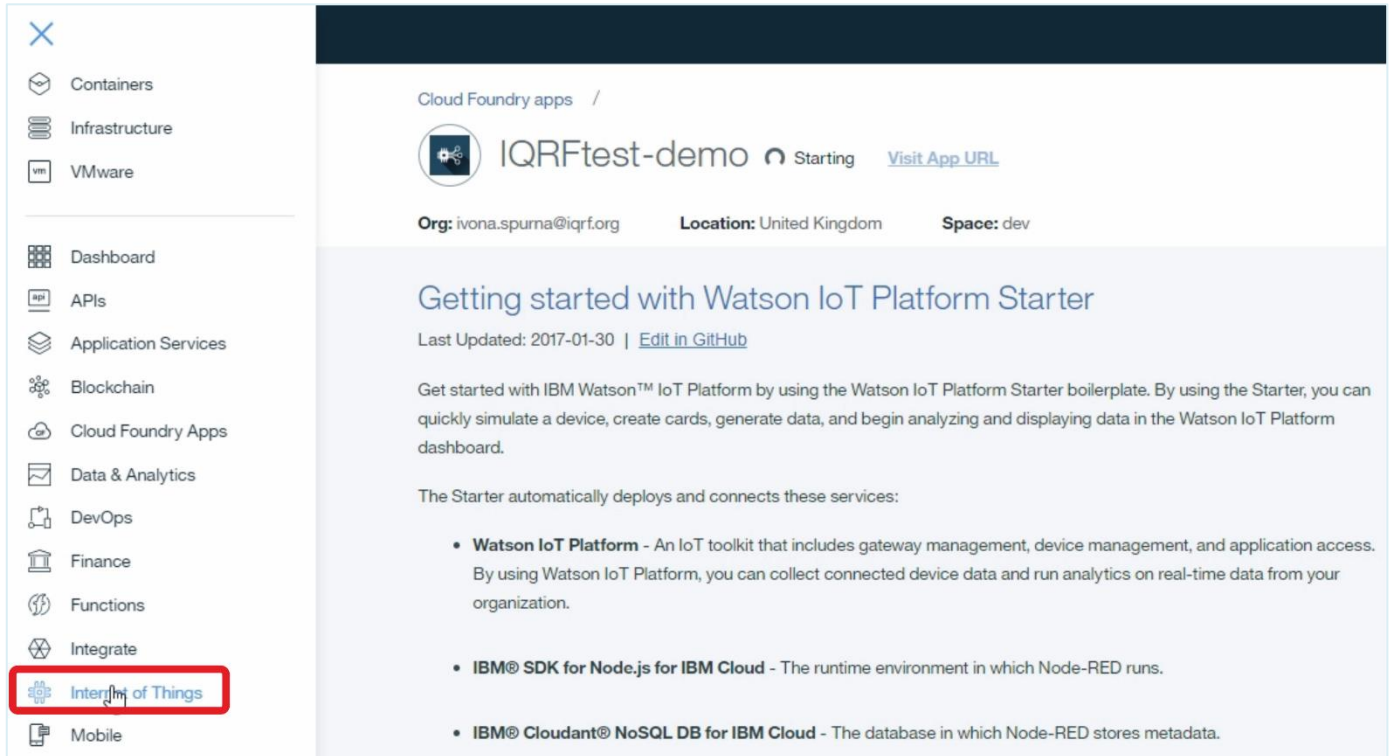
### Pricing Plans

Monthly prices shown are for country or region: [Czech Republic](#)

PLAN	FEATURES	PRICING
✓ Lite	<b>Lite apps are free</b> You get up to 256 MB of memory while you work on your apps.  Lite apps sleep after 10 days of development inactivity.	Free
Standard 256 MB+		€0.0526 EUR/GB-Hour

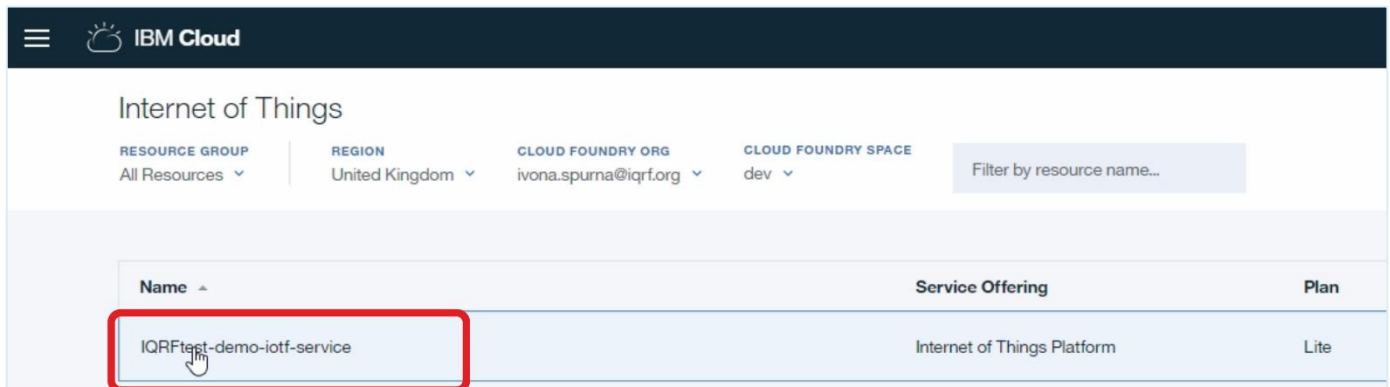
[View Docs](#) [Terms](#)

Click on the **Internet of Things** item in the left menu.



The screenshot shows the Cloud Foundry dashboard for the 'IQRfTest-demo' application. The left sidebar contains a menu with various categories. The 'Internet of Things' item, represented by a gear icon, is highlighted with a red rectangle. The main content area displays the application details, including the organization 'ivona.spurna@iqrf.org', location 'United Kingdom', and space 'dev'. The title 'Getting started with Watson IoT Platform Starter' is prominently displayed, followed by a 'Last Updated' timestamp and a link to the GitHub repository. The text describes the starter boilerplate and lists the services it automatically deploys and connects: Watson IoT Platform, IBM SDK for Node.js for IBM Cloud, and IBM Cloudant NoSQL DB for IBM Cloud.

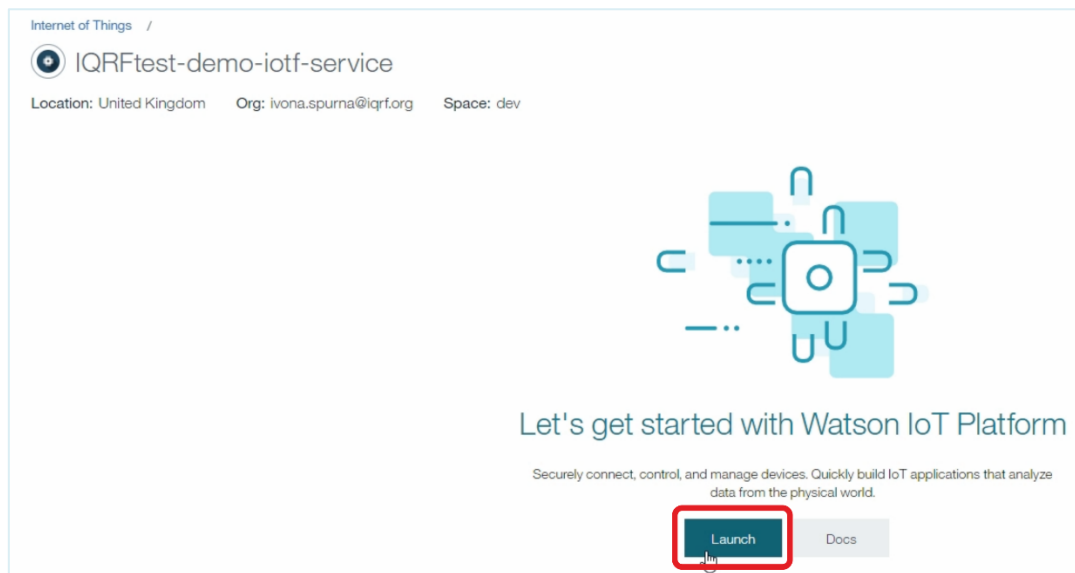
Click on the service which was created when you set up your cloud application.



The screenshot shows the IBM Cloud Internet of Things service catalog. The top navigation bar includes the IBM Cloud logo and filters for Resource Group, Region, Cloud Foundry Org, and Cloud Foundry Space. The main content area displays a table of services. The service 'IQRfTest-demo-iotf-service' is highlighted with a red rectangle. The table has columns for Name, Service Offering, and Plan.

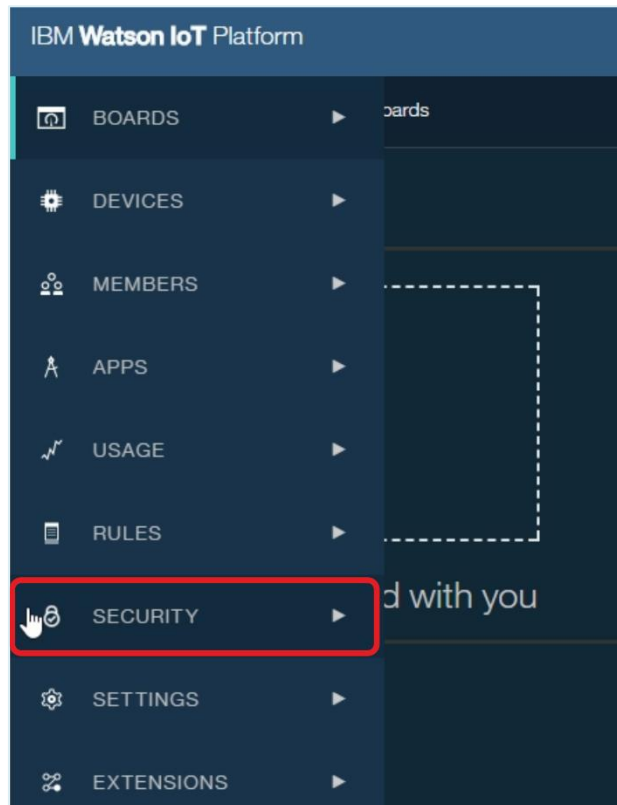
Name	Service Offering	Plan
IQRfTest-demo-iotf-service	Internet of Things Platform	Lite

Launch the **Watson IoT platform**.

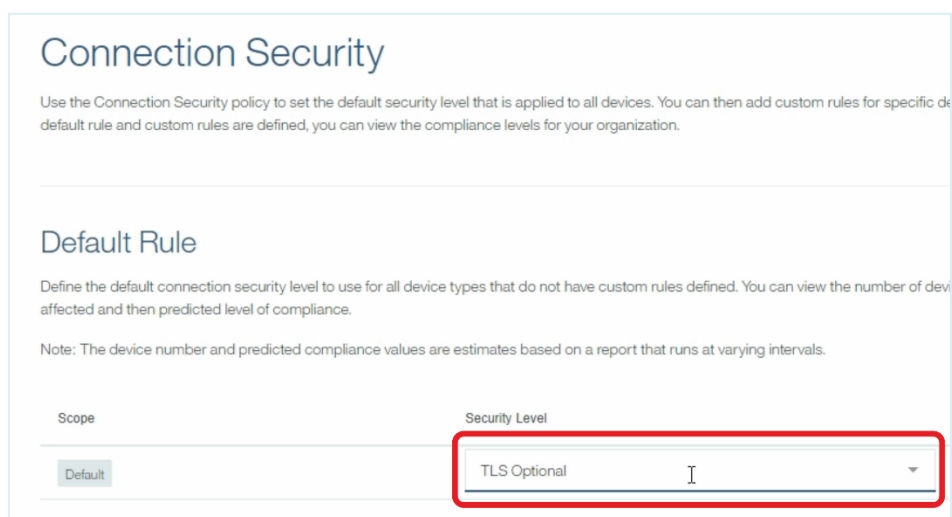
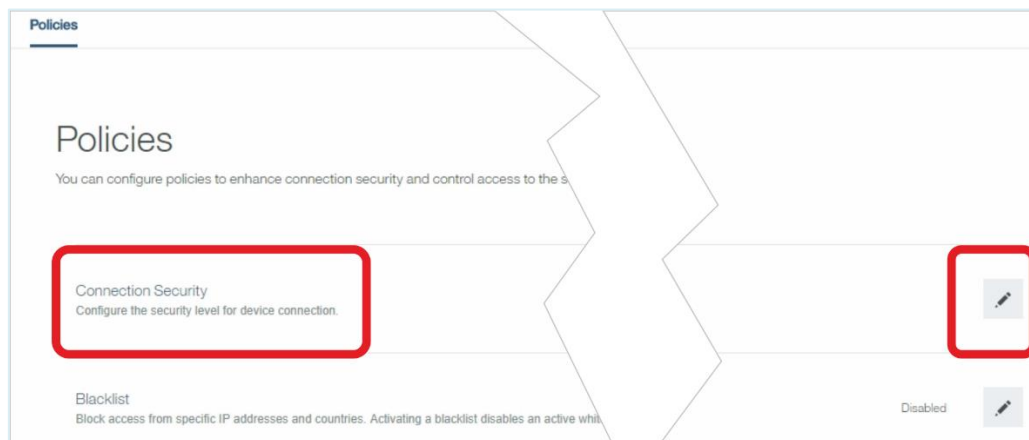


The screenshot shows the Watson IoT Platform launch page for the 'IQRfTest-demo-iotf-service'. The page displays the service name, location 'United Kingdom', organization 'ivona.spurna@iqrf.org', and space 'dev'. A large graphic of a central node connected to various devices is shown. Below the graphic, the text 'Let's get started with Watson IoT Platform' is displayed, followed by a description: 'Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.' A 'Launch' button is highlighted with a red rectangle, and a 'Docs' button is also visible.

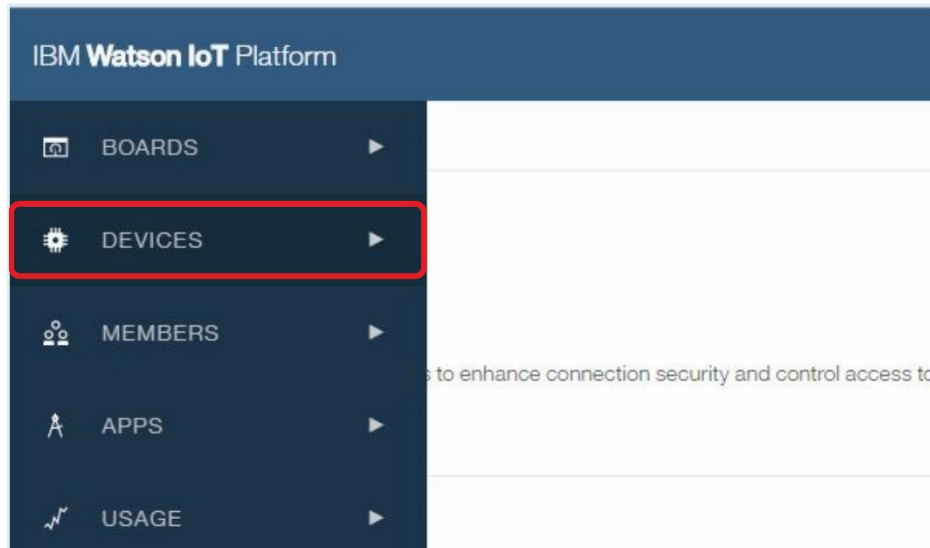
Click on the **Security** item in the left menu and check the connection security.



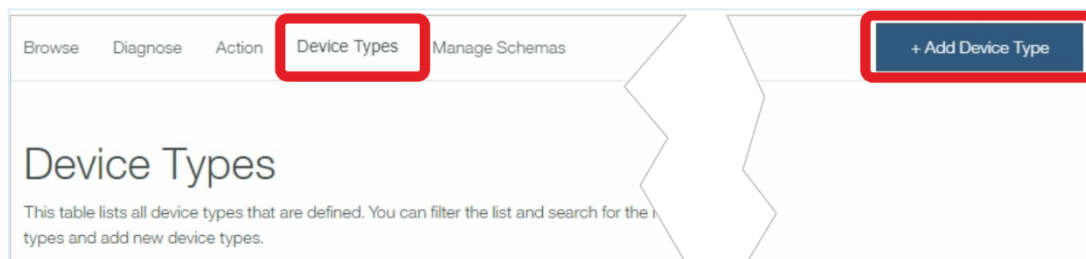
Set up the **Security** level. We have chosen the **TLS Optional**. Save the configuration.



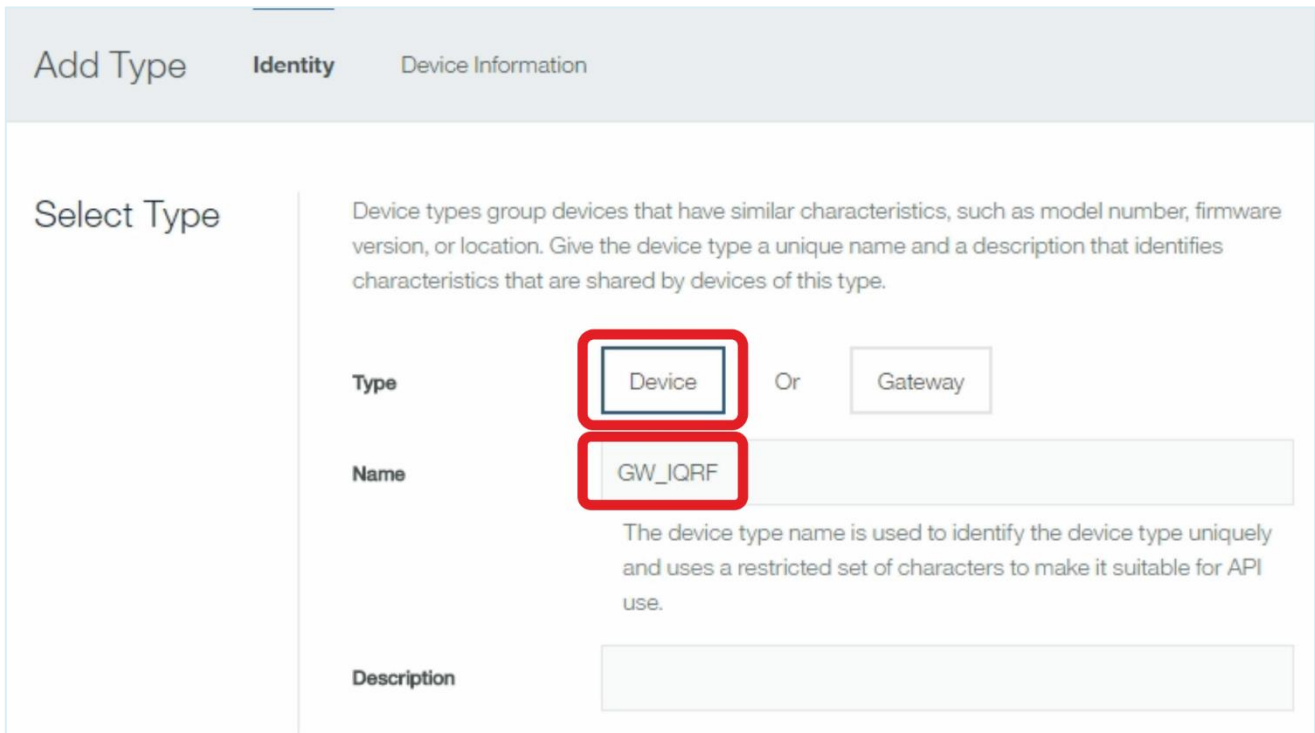
Click on the **Devices** item in the left menu.



First, create the device type.



It's important to select the type "**device**". Then, fill in your name of the device type.



**Add Type**   Identity   Device Information

**Select Type**

Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.

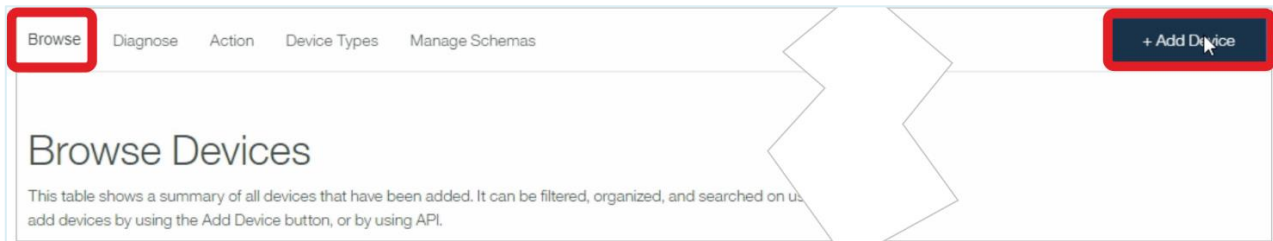
**Type**   ☒ Device   Or   ☐ Gateway

**Name**  


The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use.

**Description**

Click on the **Browse** menu. Create a new virtual device by clicking on **Add device**.



Select the **Device Type**, enter the **Device ID** and click on **Next**.



**Add Device** Identity Device Information Security Summary

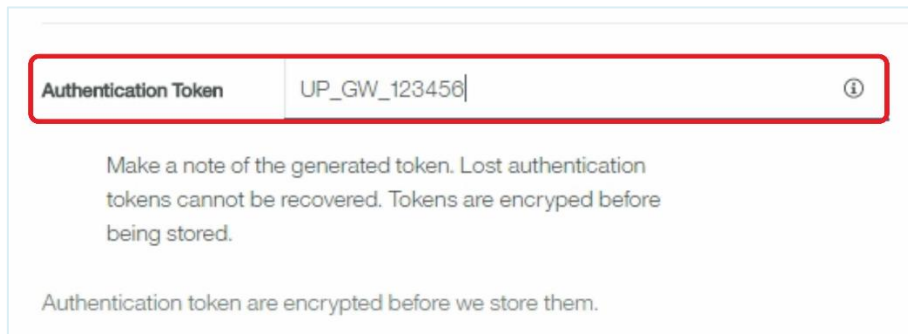
**Identity**

Select a device type for the device that you are adding and give the device a unique ID.

Select Existing Device Type: GW\_IQRF

Device ID: UP\_GW

Fill in the **Authentication Token** and click on Next.

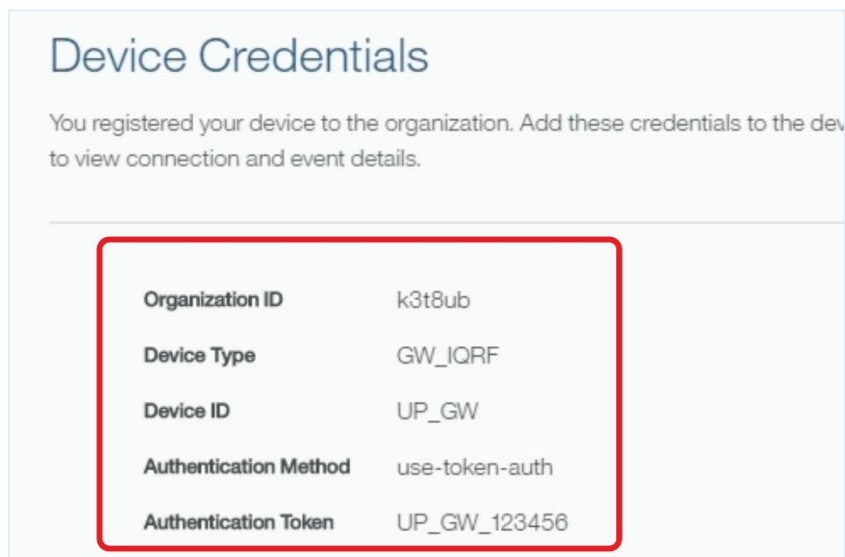


Authentication Token: UP\_GW\_123456

Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.

Authentication token are encrypted before we store them.

Copy the device credentials. You will use them in the next steps.



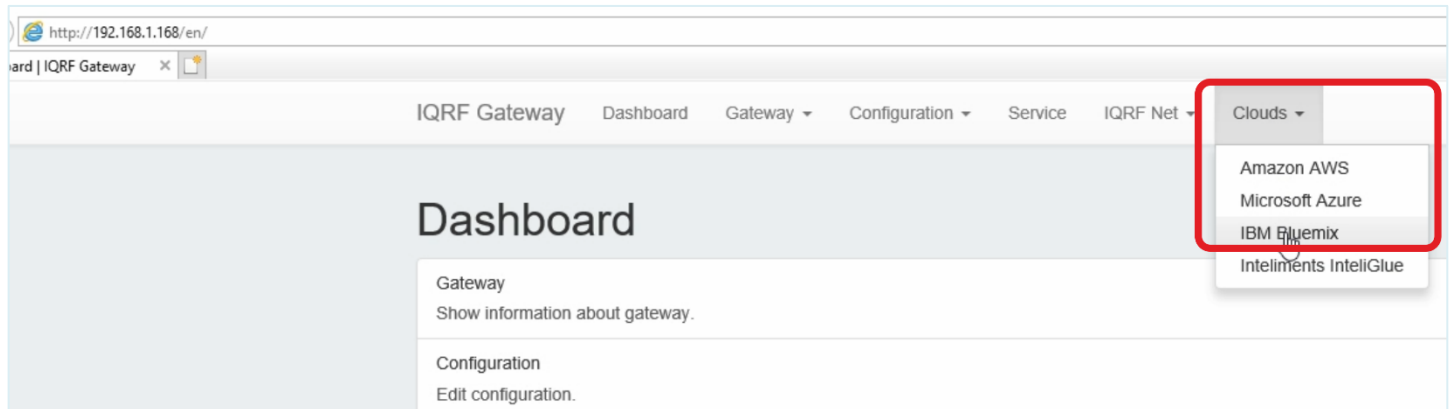
### Device Credentials

You registered your device to the organization. Add these credentials to the dev... to view connection and event details.

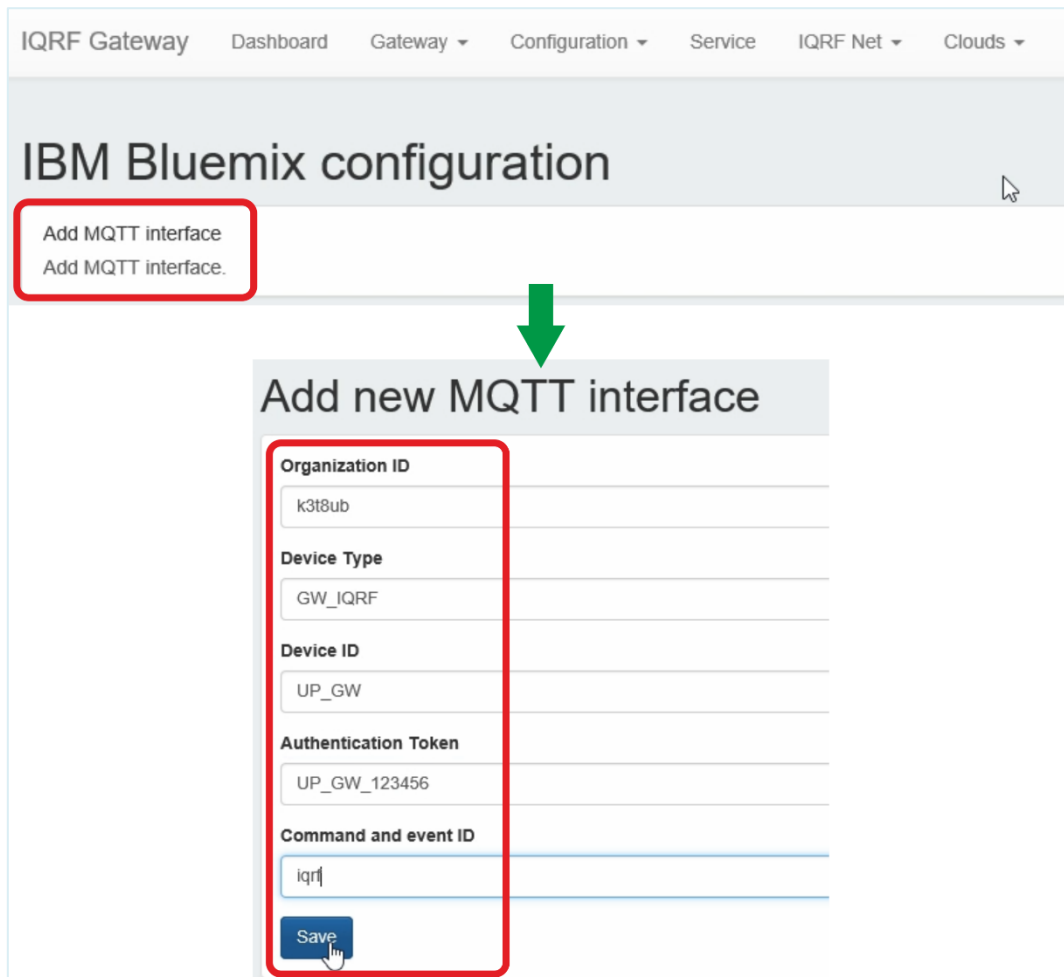
Organization ID	k3t8ub
Device Type	GW_IQRF
Device ID	UP_GW
Authentication Method	use-token-auth
Authentication Token	UP_GW_123456

## Configure MQTT channel on IQRF Gateway

Configure the MQTT channel to IBM Cloud. Go to the IQRF Gateway Daemon web application and click on the **IBM Bluemix** in the **Clouds** menu.



Click on Add MQTT Interface, fill in the copied information about the virtual device in IBM Cloud and save the configuration.

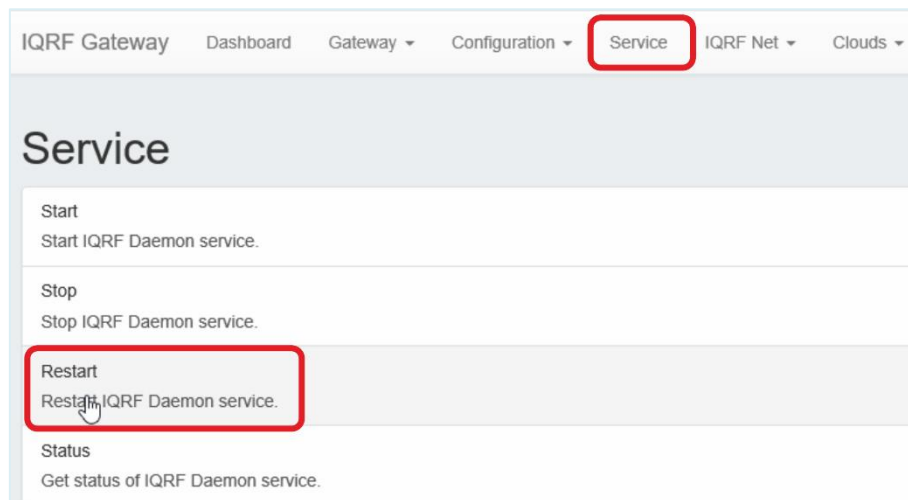


The screenshot shows the 'IBM Bluemix configuration' page. The 'Add MQTT interface' button is highlighted with a red box. A green arrow points to the 'Add new MQTT interface' form, which contains the following fields:

- Organization ID: k3t8ub
- Device Type: GW\_IQRF
- Device ID: UP\_GW
- Authentication Token: UP\_GW\_123456
- Command and event ID: iqr|

The 'Save' button is also highlighted with a red box.

**Restart** the IQRF Gateway Daemon service.

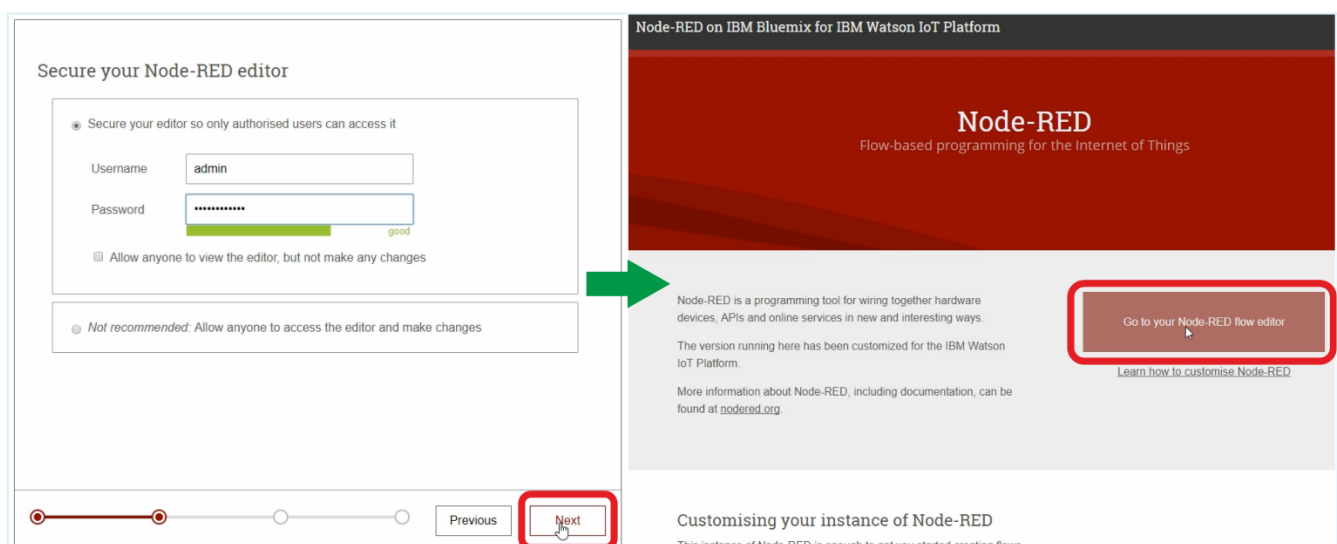


## Node-RED

Find **Cloud Foundry Apps** in the **IBM Cloud** and check the status of your application. It should be running.

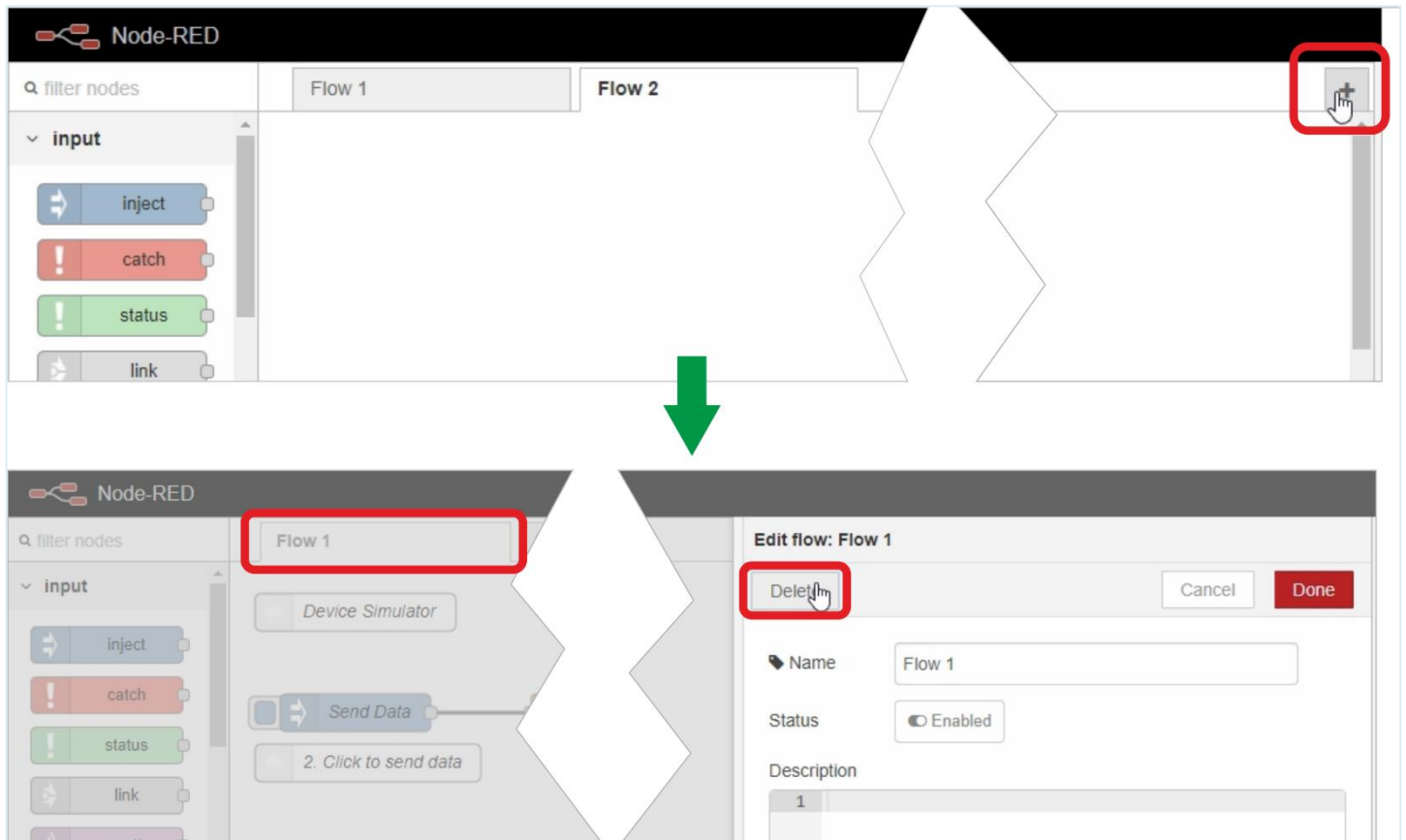


Click on the link of your web application. Go through the wizard and set up the **Node-RED** application. Go to your Node-RED flow editor and log in to it.

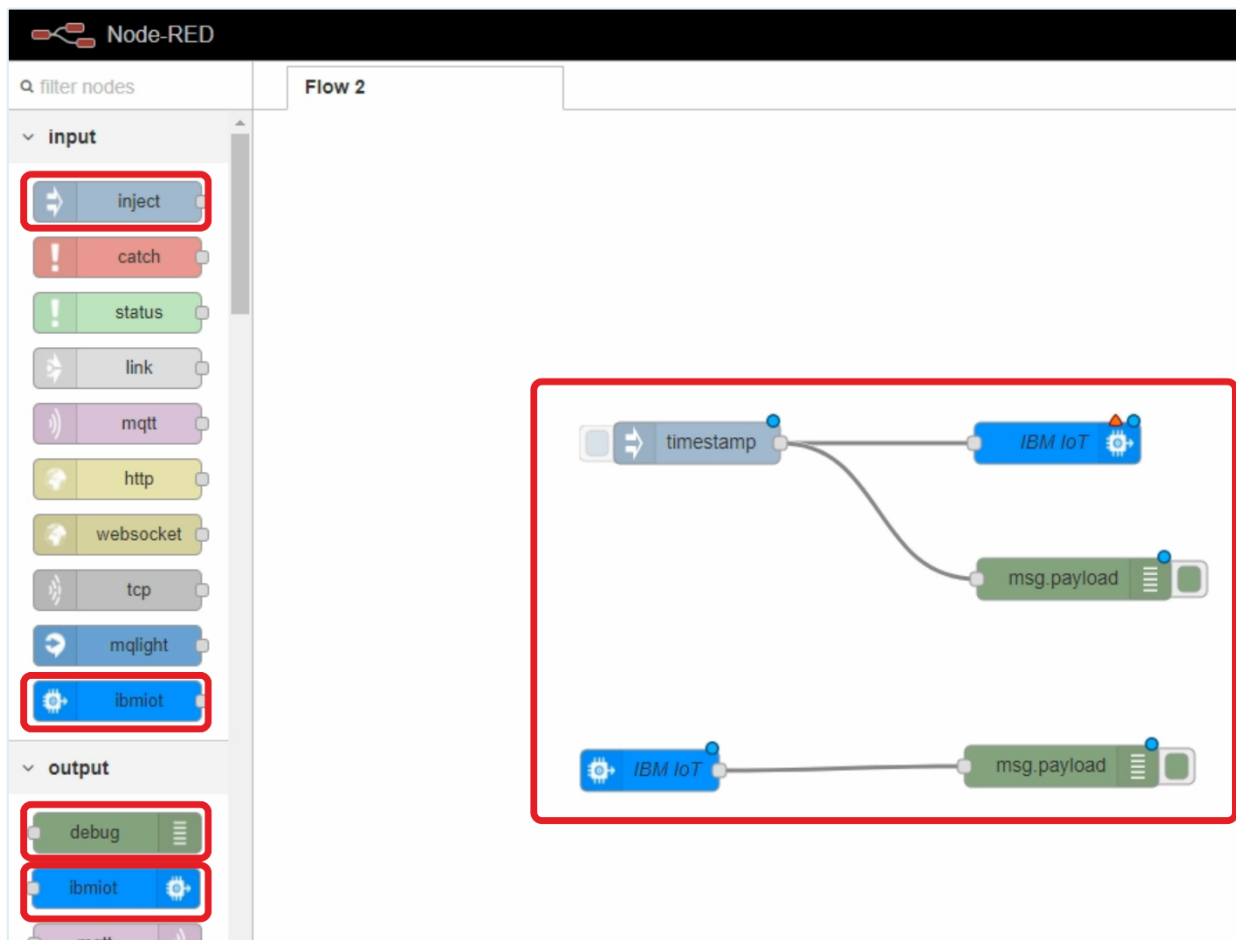




Create a **new flow** and **delete** the example. You will do it by double-clicking on the Flow 1 tab. Then press delete.

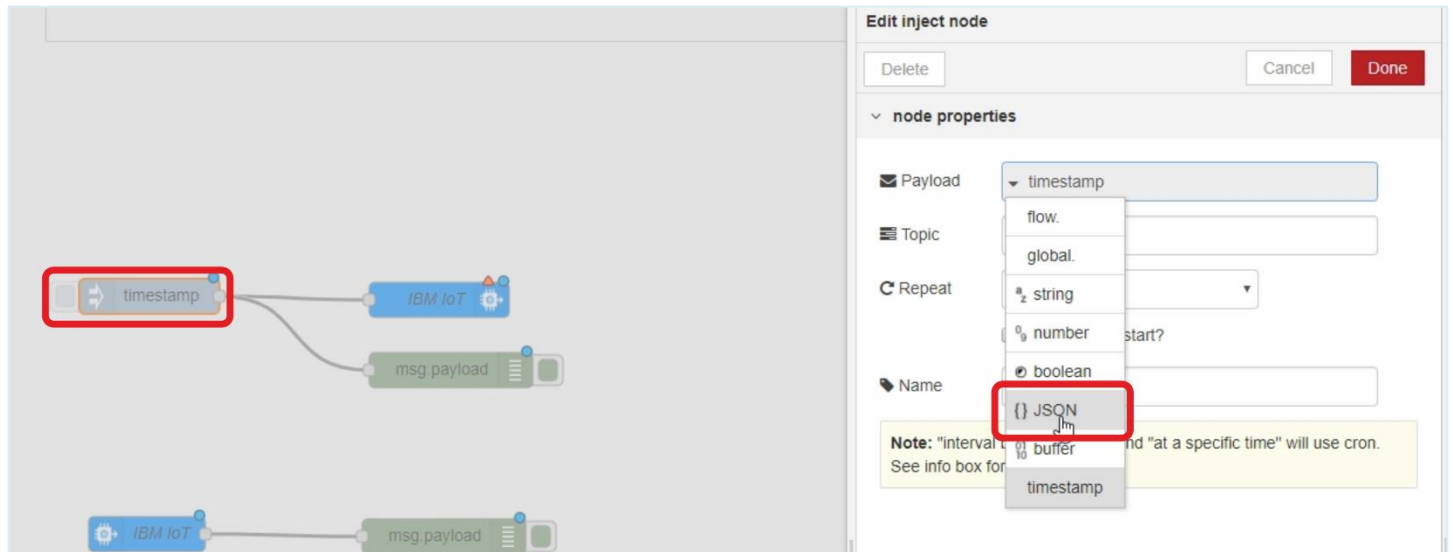


Insert **ibmiot input**, **ibmiot output**, two **debug outputs** and **inject input**. Connect the objects like this.



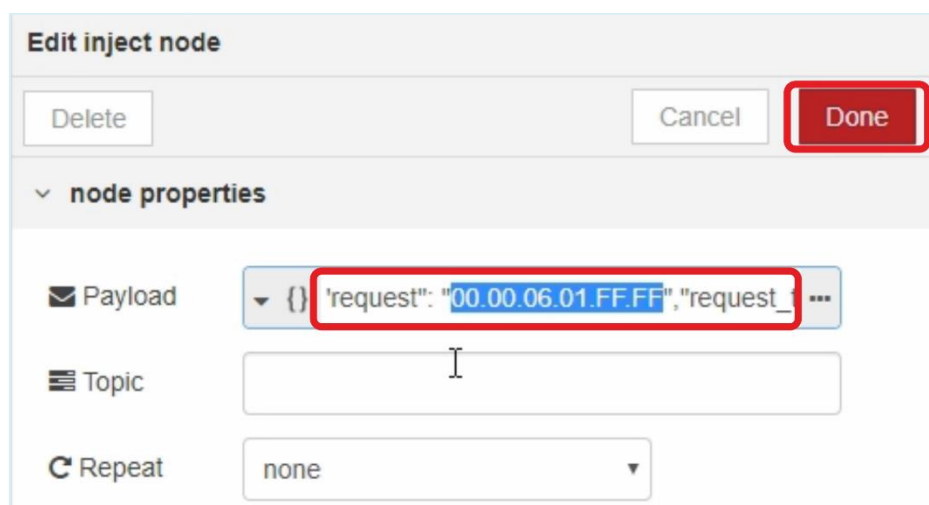
With the inject input we will send **DPA commands** to the MQTT broker on the IBM Cloud and from there our UP board will collect them. We will send the commands to the debug window as an output, as well. We will receive all messages from the MQTT broker and they will be displayed in the debug window.

Double-click on the **inject input**, change the payload type to JSON and insert here the DPA command in JSON format.

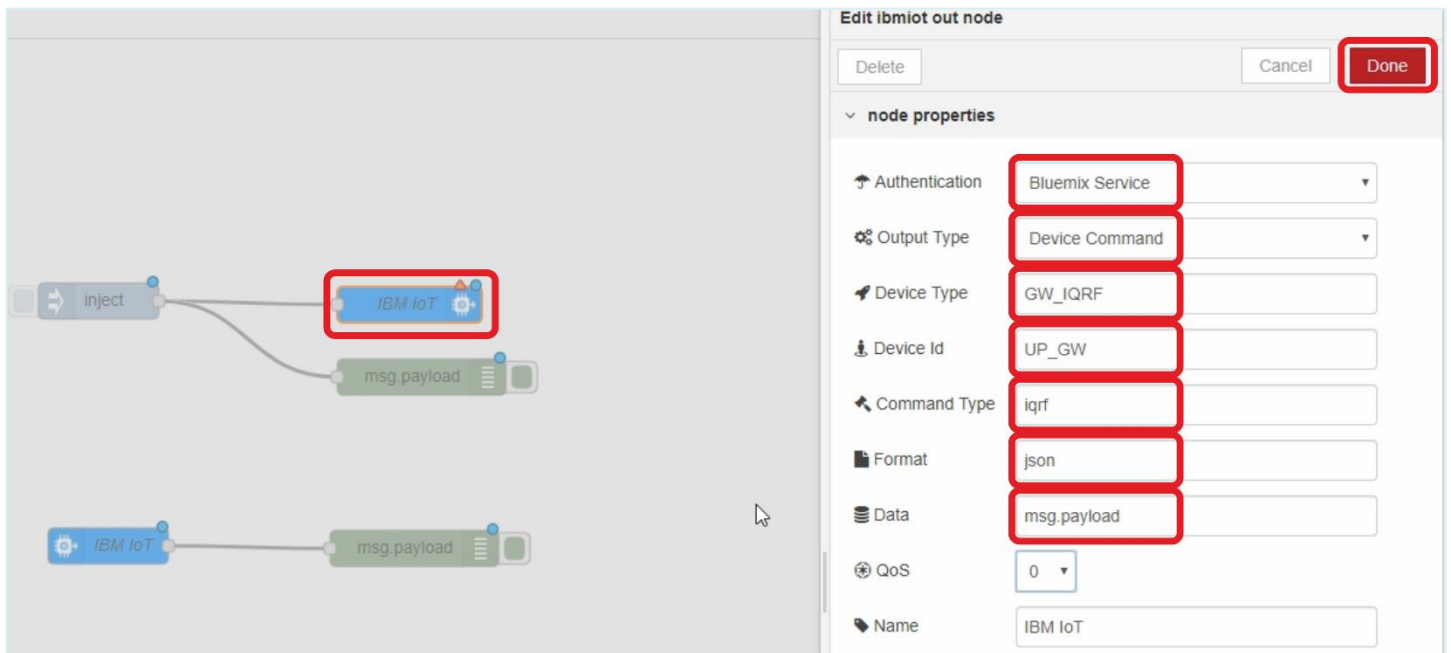


We use the command which will turn on the red LED on the IQRF coordinator. Click on **Done**.

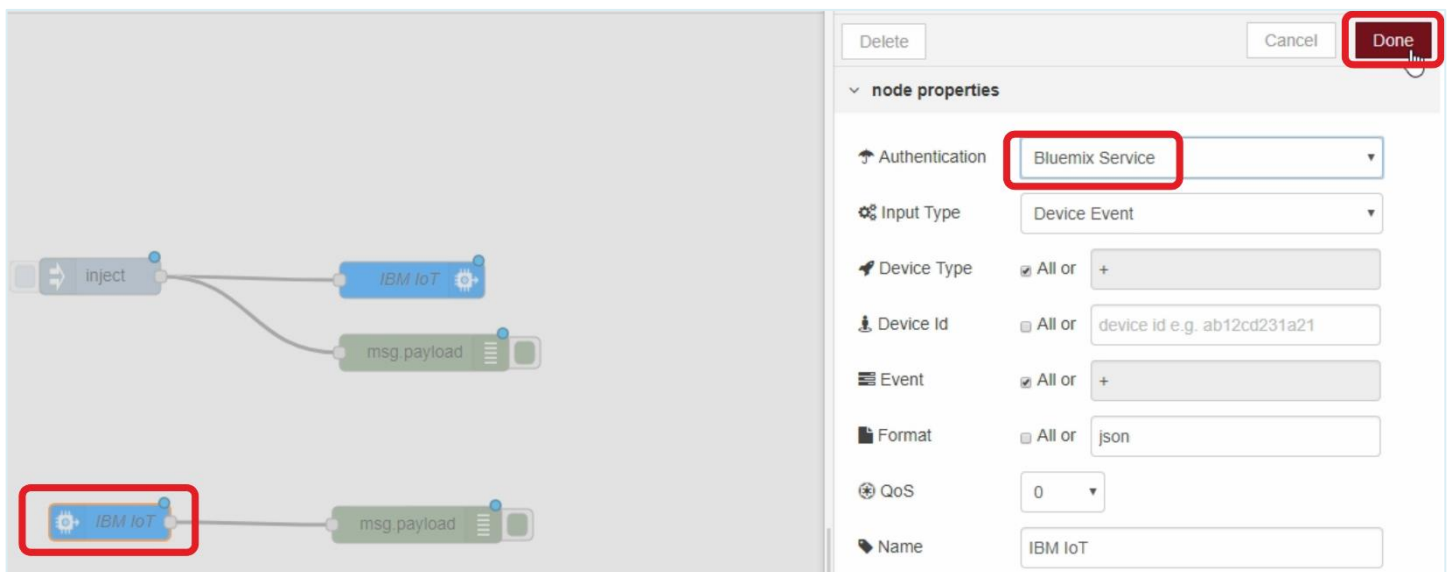
```
{
  "ctype": "dpa",
  "type": "raw",
  "msgid": "1510754980",
  "request": "00.00.06.01.FF.FF",
  "request_ts": "",
  "confirmation": "",
  "confirmation_ts": "",
  "response": "",
  "response_ts": ""
}
```



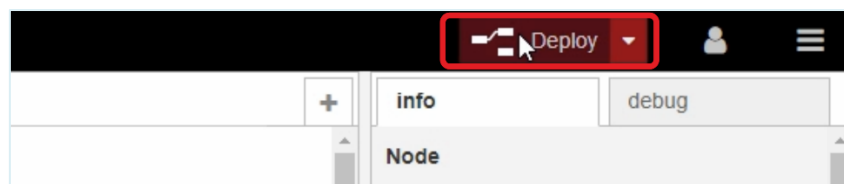
Click on the **ibmiot output**. Change the authentication to **Bluemix service**, set the output type to **Device** command, and fill in the information of your virtual device you have created earlier. Enter “**iqrf**” as the command type. Enter “**msg.payload**” as the Data and click on **Done**.



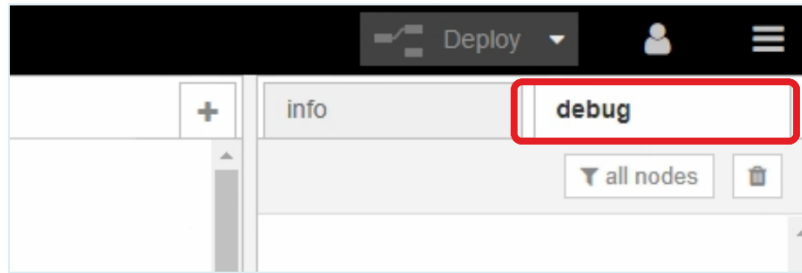
Click on the **ibmiot input** and select **Bluemix service** as the authentication type. **Save** the configuration.



Click on the **Deploy** button.



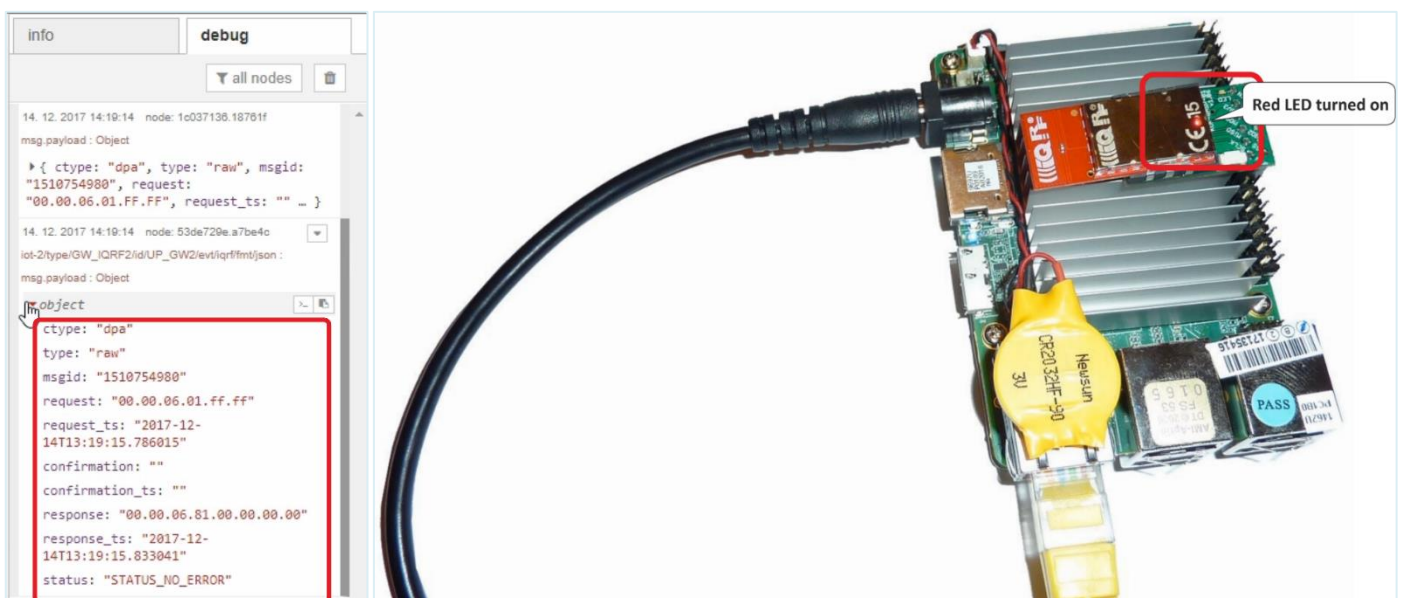
Show the **Debug** tab.



Click on the left corner of the **Inject** button. You will send the prepared command to the MQTT broker and to the debug output as well.



In the **Debug** tab, you can see the ongoing communication between IBM Cloud and the UP board. You can easily double check that the command has been executed.



In the same way, you can turn the red LED off as well as send any other DPA command to your network.