

IoT Starter Kit – Part 3: Connect to the cloud – IBM Cloud

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.

In this part we will connect an IQRF gateway to IBM Cloud. It is one of the possible clouds that you can get connected to from your IQRF Gateway Daemon using the MQTT channel.

1 IBM Cloud and Watson IoT platform

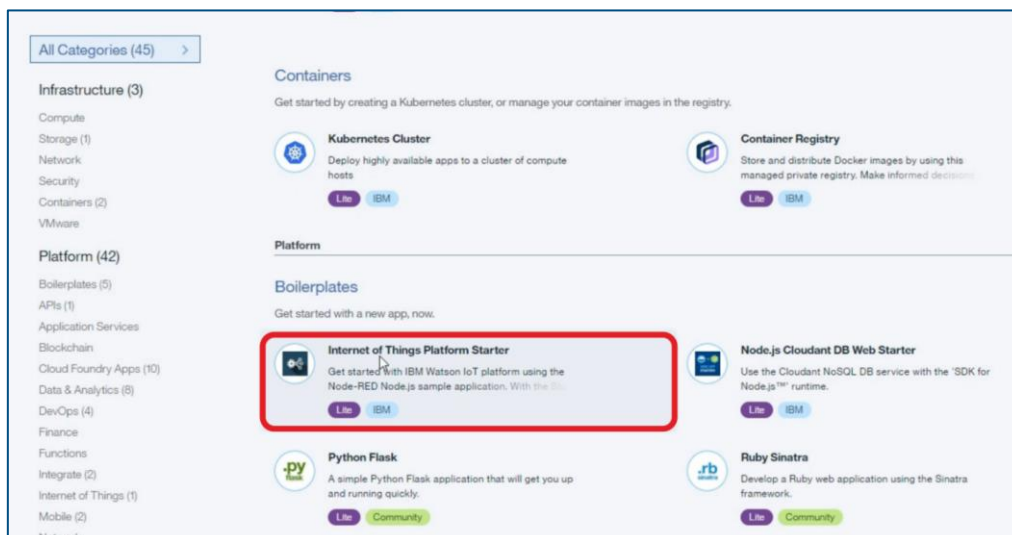
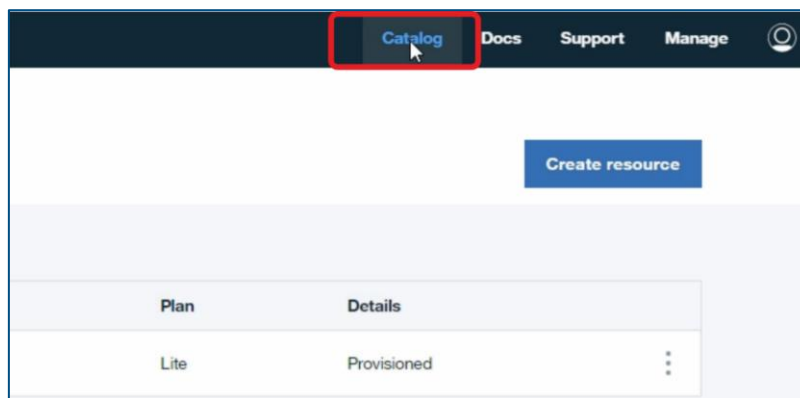
Note: the environment of IBM Cloud may look different because of possible changes. This guide shows the status of March 2018. You need to look for appropriate items to configure the MQTT connection.

IBM provides developers some free services for a limited time and you don't have to enter any credit card details on the beginning. Create your IBM Cloud account and log into it on console.bluemix.net.

To connect remotely to your IQRF network from IBM cloud, you need to set up some services first.

1.1 Internet of Things Platform Starter

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**. Your application will be available at the given address.

App name:

IQRTest-demo

Host name:

IQRTest-demo

Domain:

eu-gb.mybluemix.net

Choose a region/location to deploy in:

United Kingdom

Choose an organization:

ivona.spurna@iqrf.org

Choose a space:

dev

Selected Plan:

SDK for Node.js™

Lite

Cloudant NoSQL DB

Lite

Internet of Things Platform

Lite

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](#)

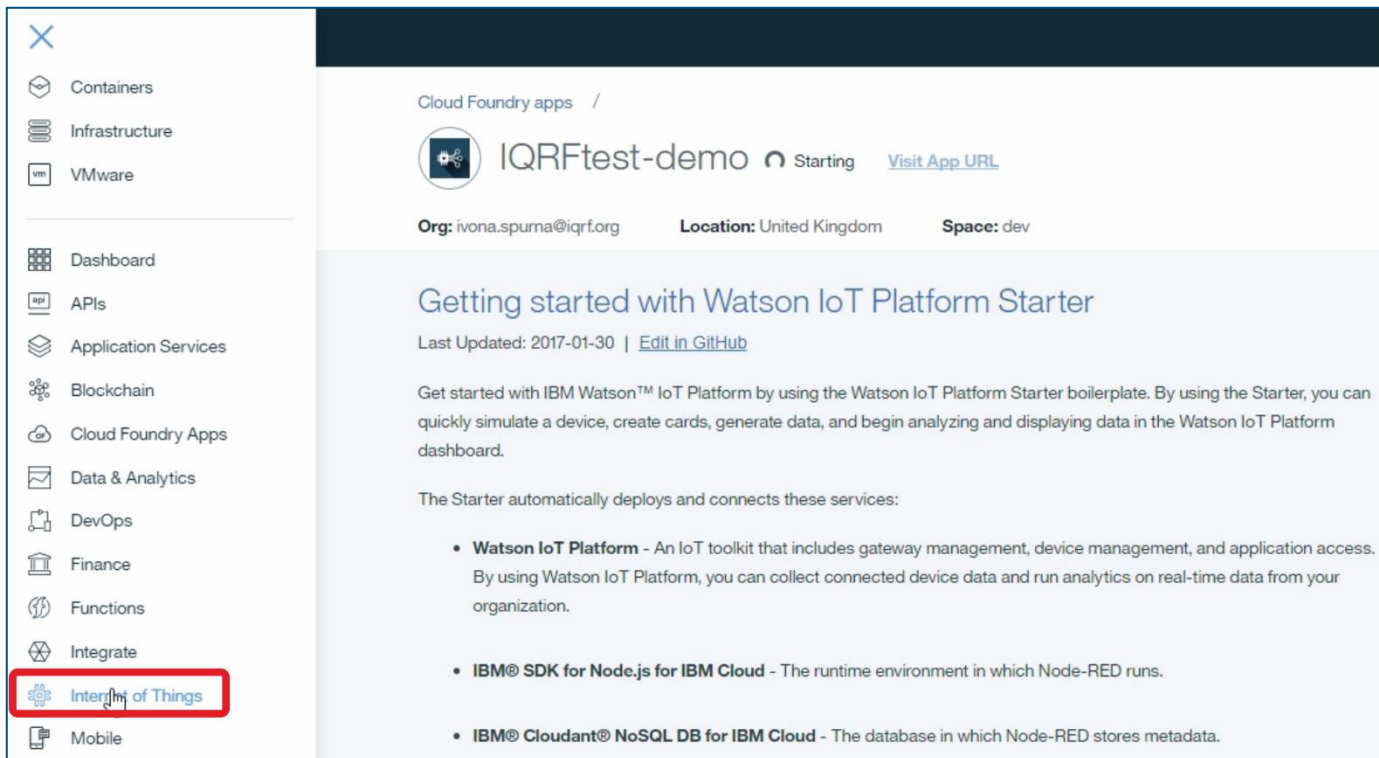
PLAN	FEATURES	PRICING
✓ Lite	<p>Lite apps are free</p> <p>You get up to 256 MB of memory while you work on your apps.</p> <hr/> <p>Lite apps sleep after 10 days of development inactivity.</p>	Free
Standard 256 MB+		€0.0526 EUR/GB-Hour

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

Create

1.2 Watson IoT Platform

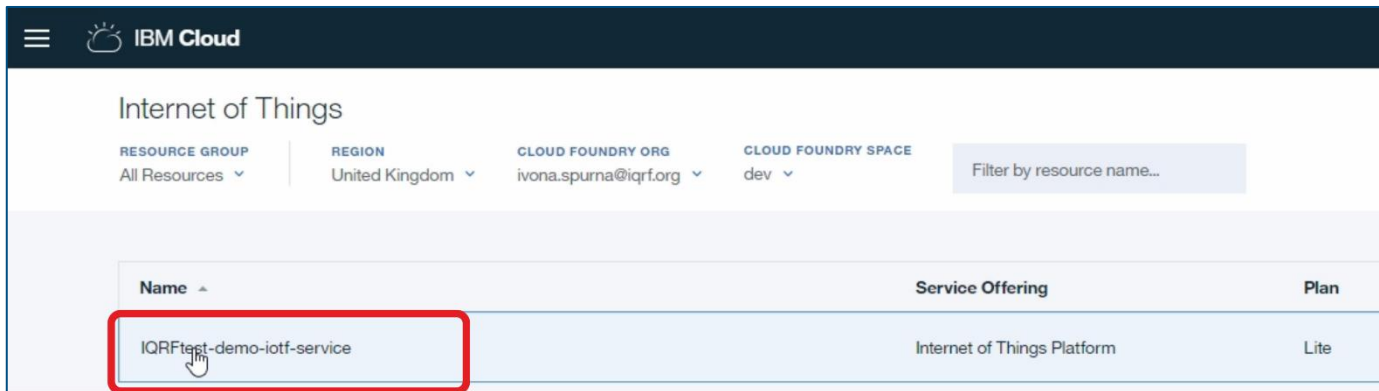
Click on the **Internet of Things** item in the left menu.



The screenshot shows the IBM Cloud Foundry dashboard. On the left, a sidebar menu lists various services: Containers, Infrastructure, VMware, Dashboard, APIs, Application Services, Blockchain, Cloud Foundry Apps, Data & Analytics, DevOps, Finance, Functions, Integrate, **Internet of Things** (highlighted with a red box), and Mobile. The main content area displays the 'IQRFTest-demo' application, which is in a 'Starting' state. It provides details such as 'Org: ivona.spurna@iqrf.org', 'Location: United Kingdom', and 'Space: dev'. Below this, there is a section titled 'Getting started with Watson IoT Platform Starter' with a 'Last Updated: 2017-01-30' and a link to 'Edit in GitHub'. The text explains that the starter boilerplate allows users to quickly simulate a device, create cards, generate data, and begin analyzing and displaying data in the Watson IoT Platform dashboard. It also lists the services automatically deployed and connected by the starter:

- **Watson IoT Platform** - An IoT toolkit that includes gateway management, device management, and application access. By using Watson IoT Platform, you can collect connected device data and run analytics on real-time data from your organization.
- **IBM® SDK for Node.js for IBM Cloud** - The runtime environment in which Node-RED runs.
- **IBM® Cloudant® NoSQL DB for IBM Cloud** - The database in which Node-RED stores metadata.

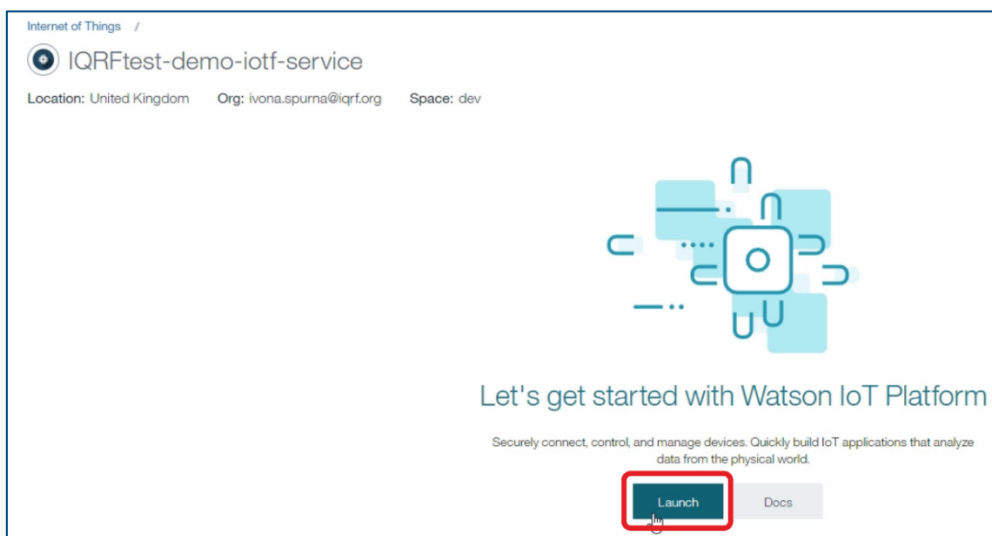
Click on the service which was created when you set up your cloud application in the previous step.



The screenshot shows the IBM Cloud 'Internet of Things' service catalog. At the top, there are filters for 'RESOURCE GROUP' (All Resources), 'REGION' (United Kingdom), 'CLOUD FOUNDRY ORG' (ivona.spurna@iqrf.org), and 'CLOUD FOUNDRY SPACE' (dev). A search bar labeled 'Filter by resource name...' is also present. Below the filters is a table with the following columns: Name, Service Offering, and Plan. The table contains one entry, 'IQRFTest-demo-iotf-service', which is highlighted with a red box. The 'Service Offering' for this entry is 'Internet of Things Platform' and the 'Plan' is 'Lite'.

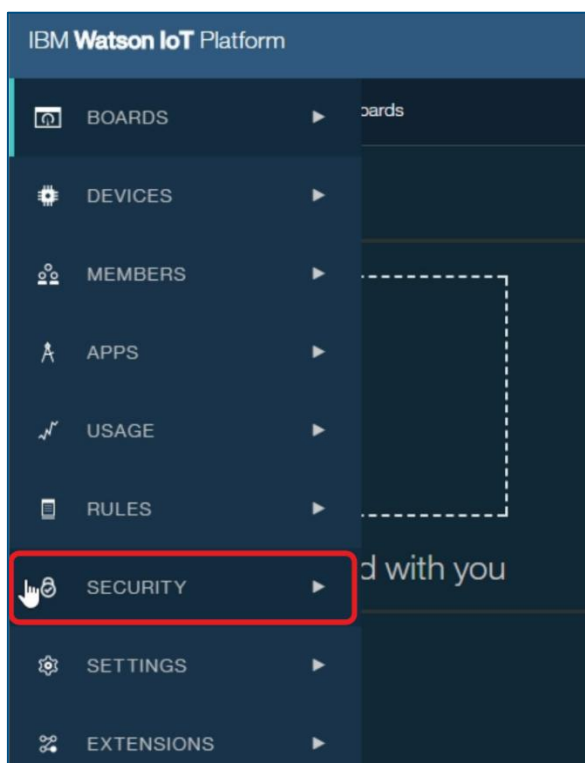
Name	Service Offering	Plan
IQRFTest-demo-iotf-service	Internet of Things Platform	Lite

Launch the **Watson IoT platform**.

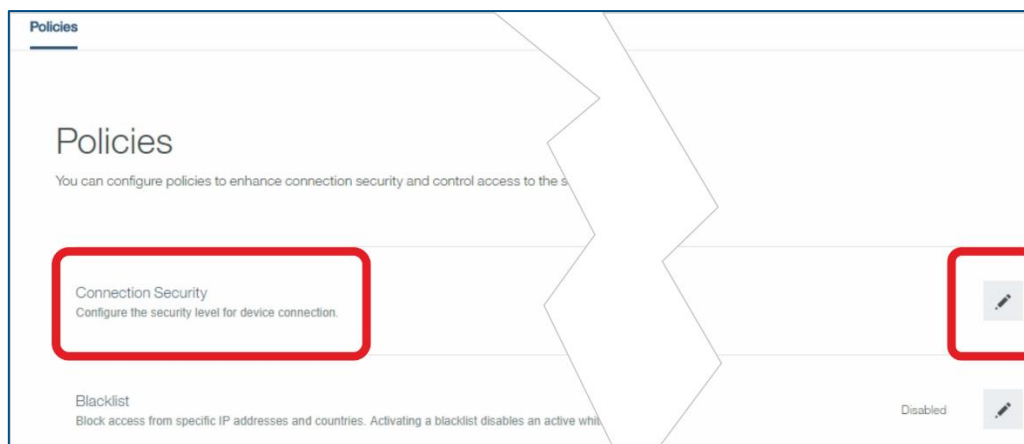


1.2.1 Security

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

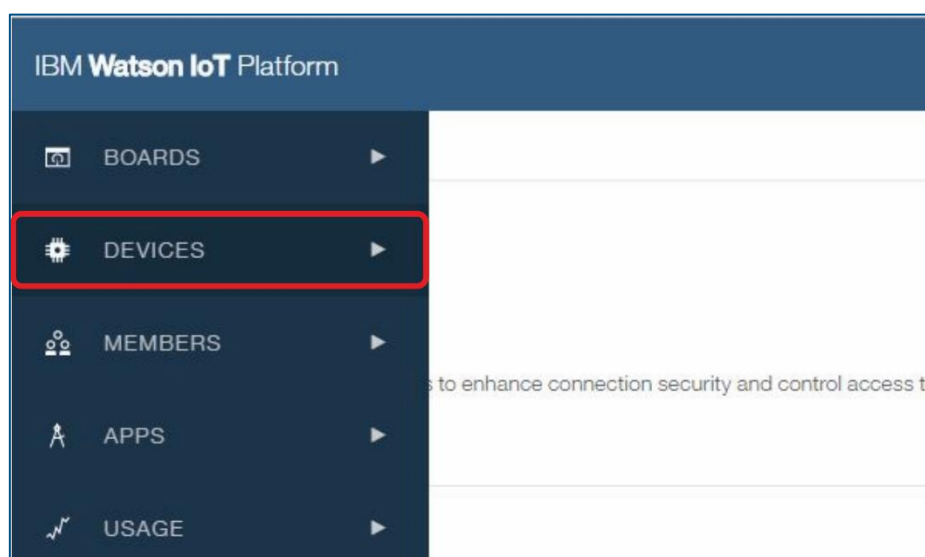


Set up the **Security** level. We have chosen the **TLS Optional**. Save the configuration.

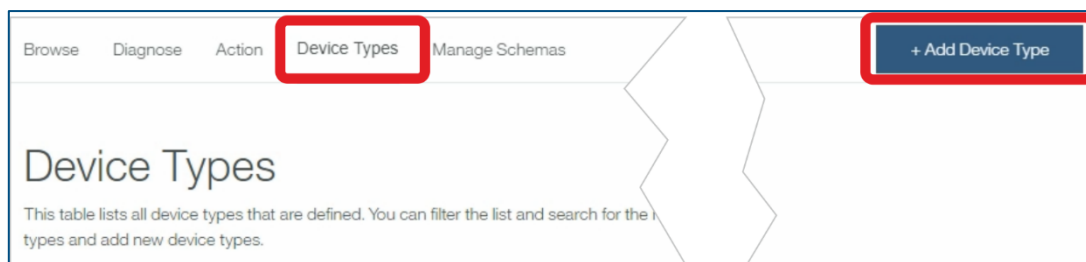


1.2.2 Create the device type

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



First, add 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

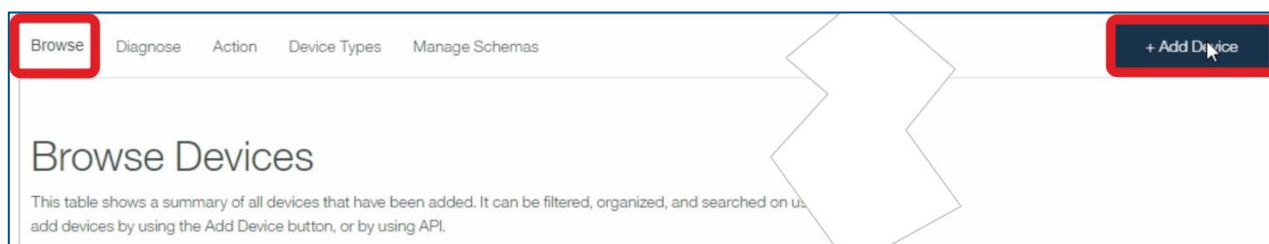
Name

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.

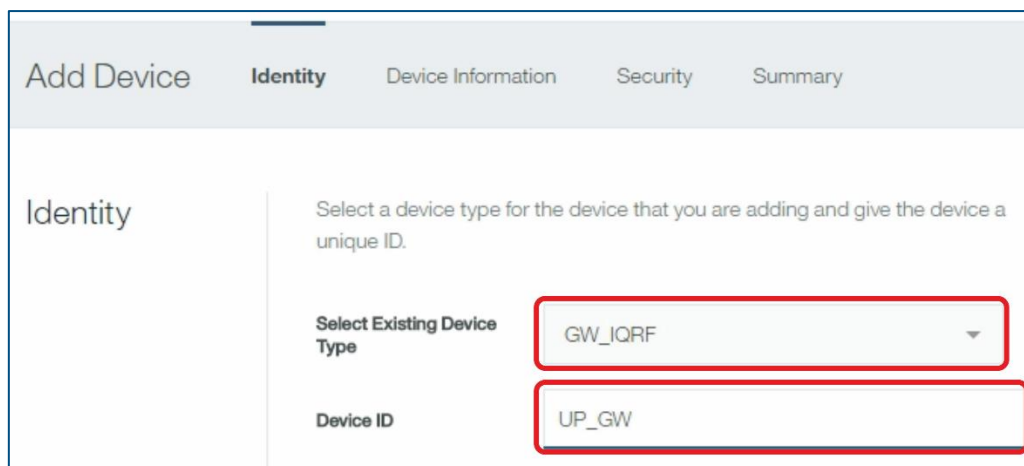
Description

1.2.3 Create the device

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



Select the **Device Type** which you created in the previous step, enter the **Device ID** and click on **Next**.



Add Device Identity Device Information Security Summary

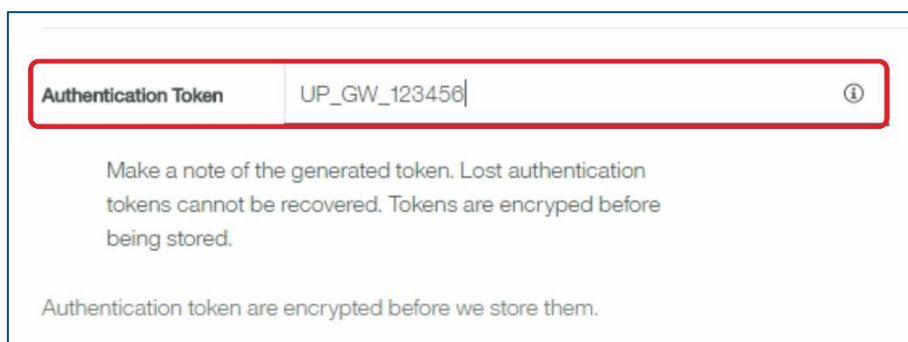
Identity

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 your **Authentication Token** and click on Next.

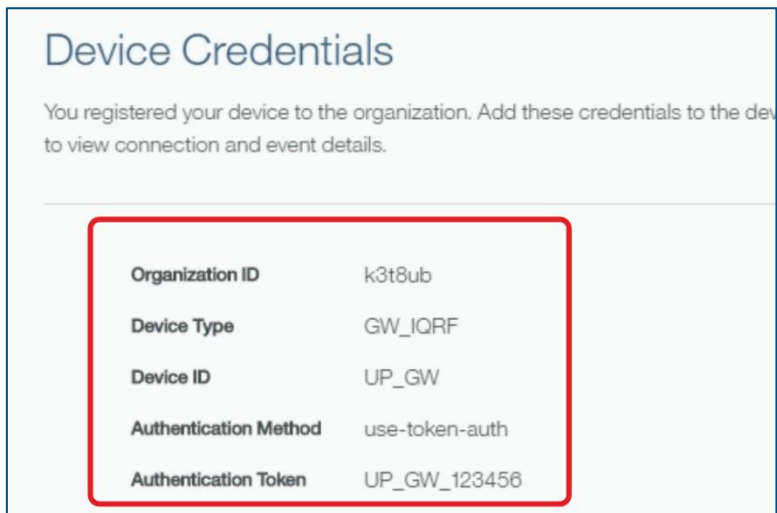


Authentication Token: UP_GW_123456

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.



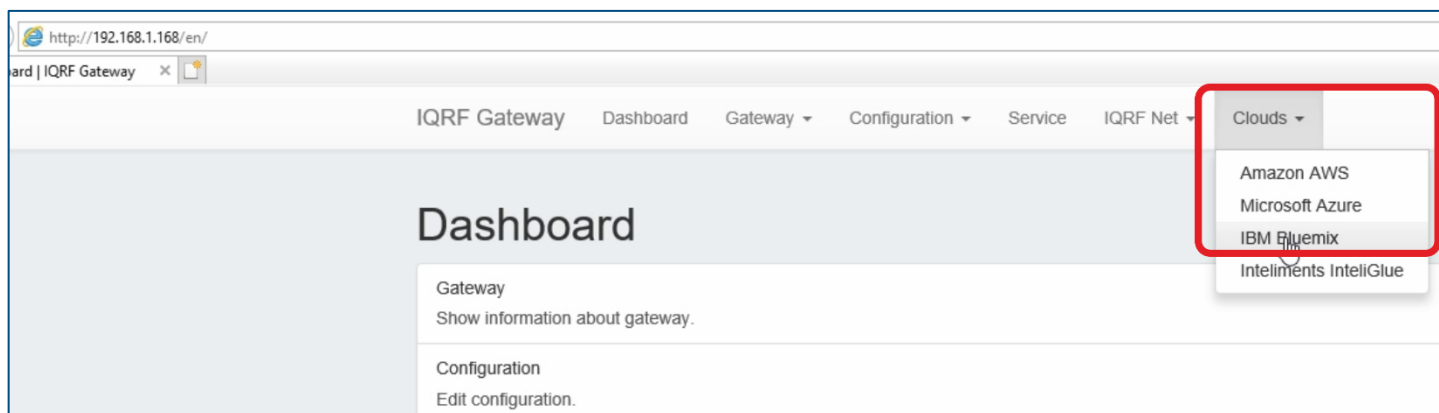
Device Credentials

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

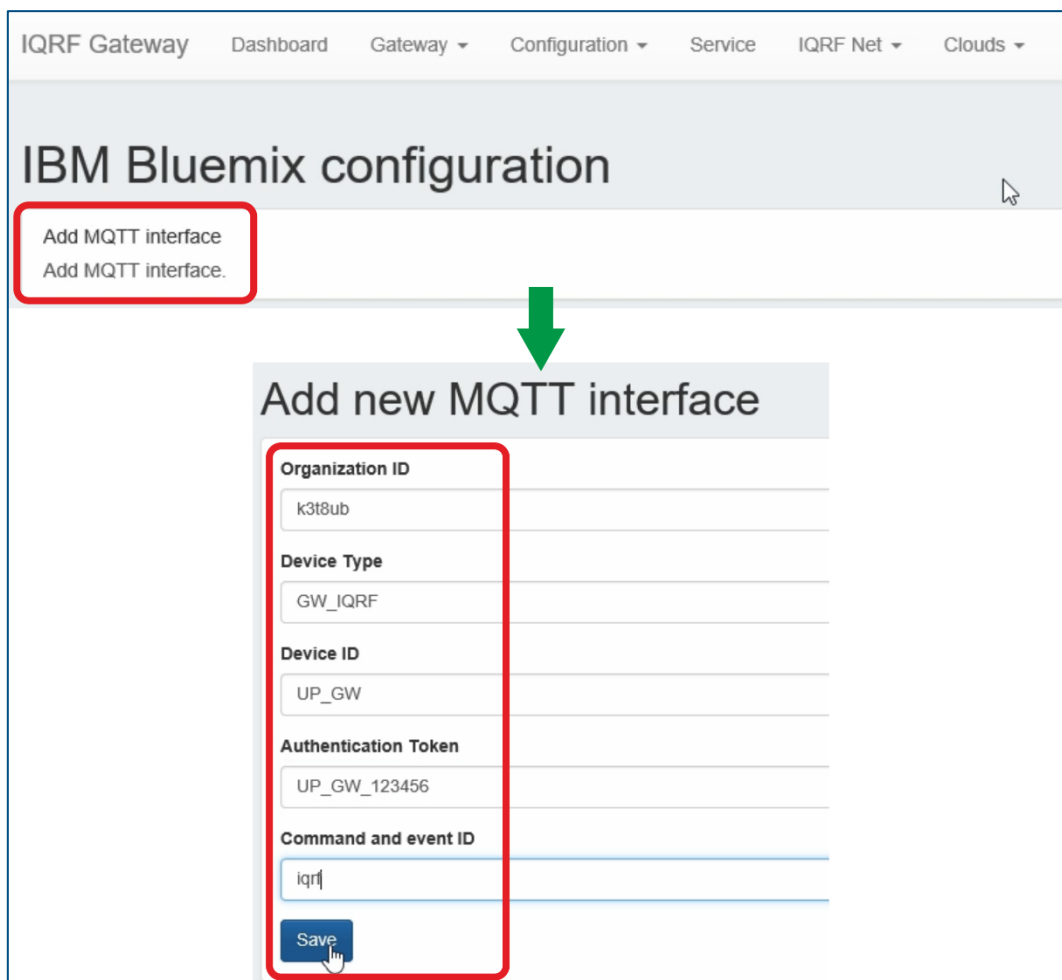
Organization ID	k3t8ub
Device Type	GW_IQRF
Device ID	UP_GW
Authentication Method	use-token-auth
Authentication Token	UP_GW_123456

2 Set up the MQTT interface on the IQRF Gateway

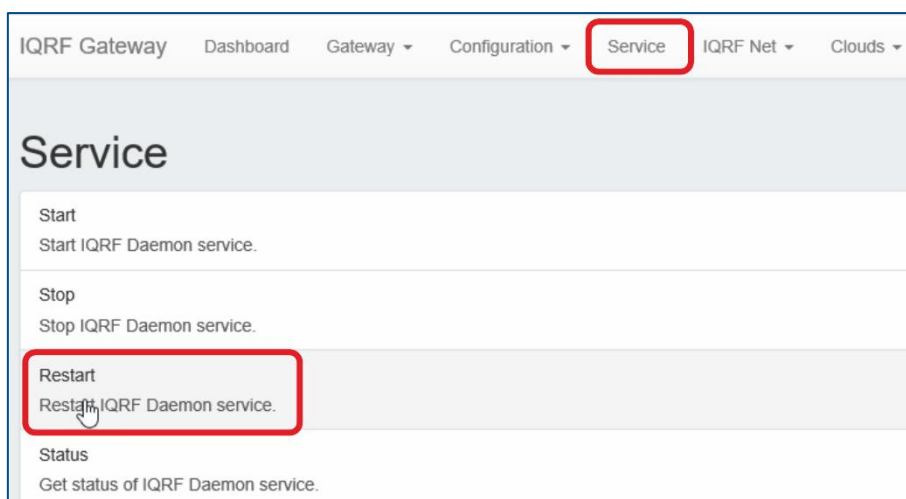
Configure the MQTT interface to IBM Cloud. In the web browser on your computer, insert the IP address of your UP board, and login to it as *admin* with password *iqrf*. In the IQRF Gateway Daemon Web application click on the **IBM Bluemix** item in the **Clouds** menu.



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



Restart the IQRF Gateway Daemon service.

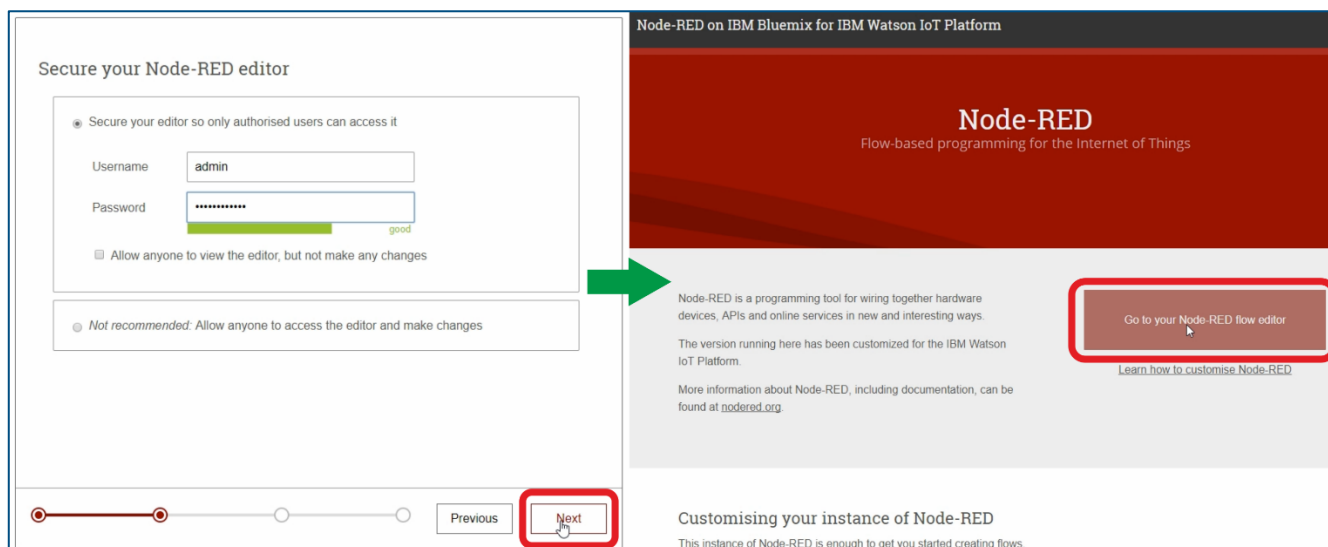


3 Node-RED

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

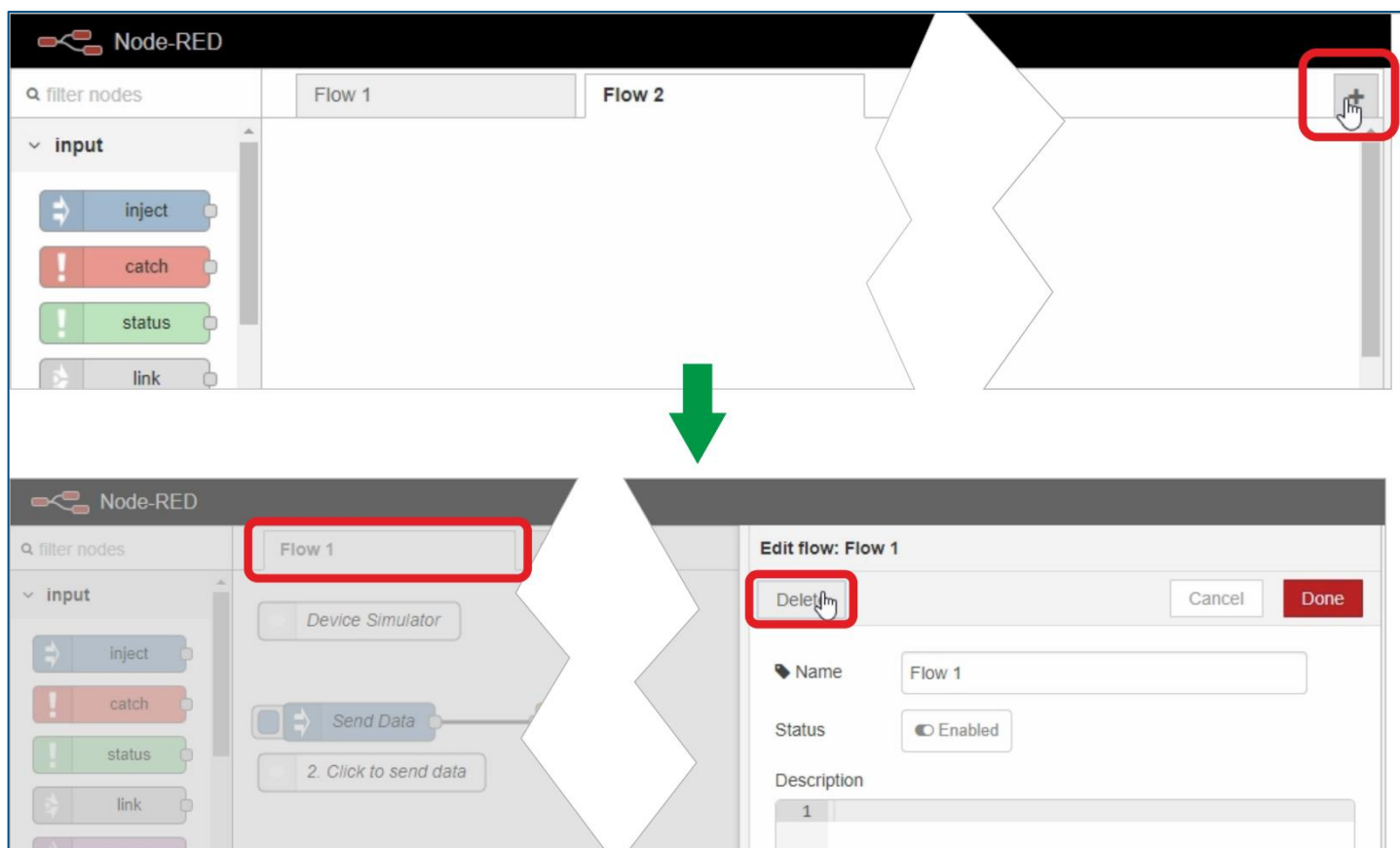


Click on the link of your web application. Go through the wizard and set up the **Node-RED** application. Go to your Node-RED flow editor and log in to it.

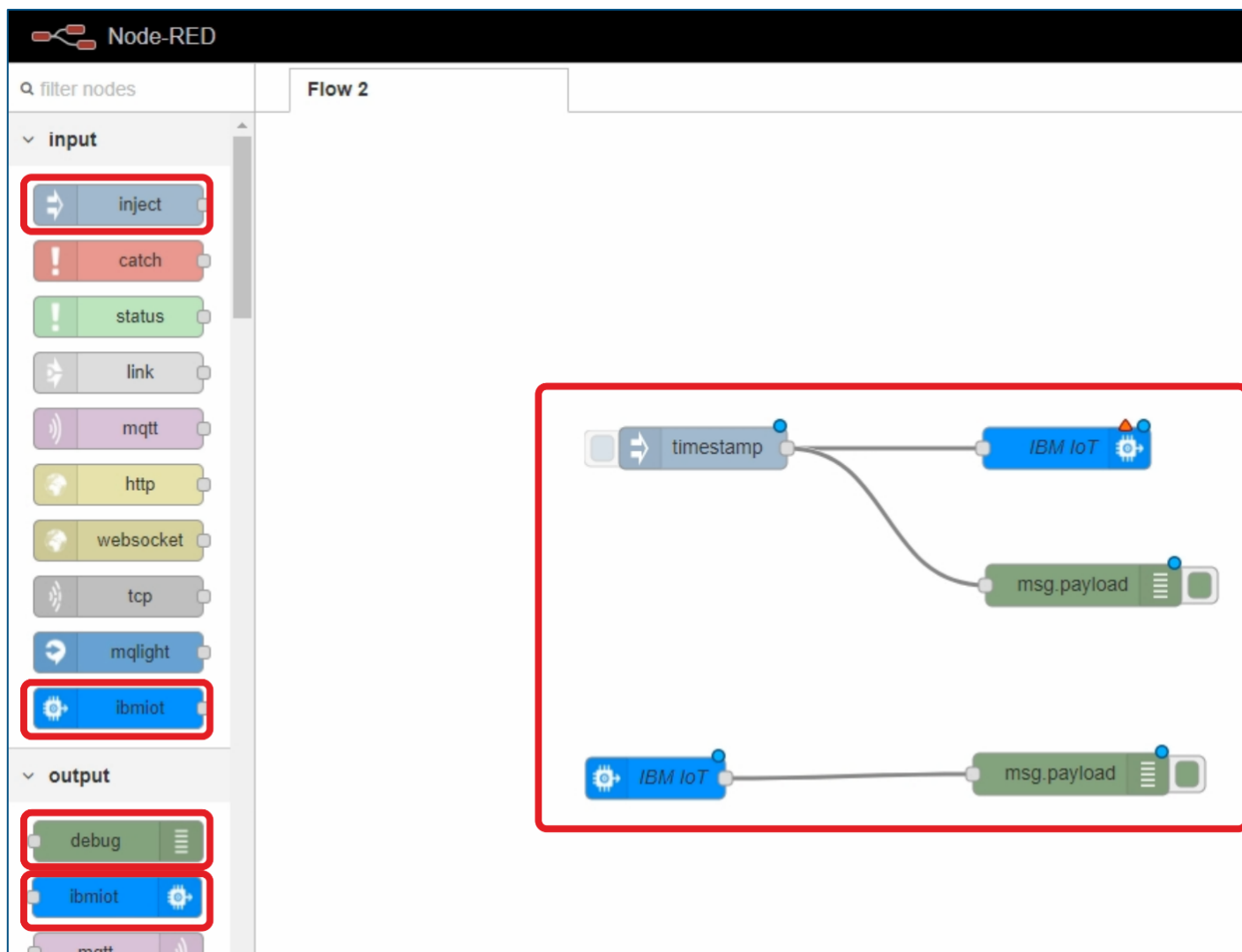


3.1 Node-RED flow

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



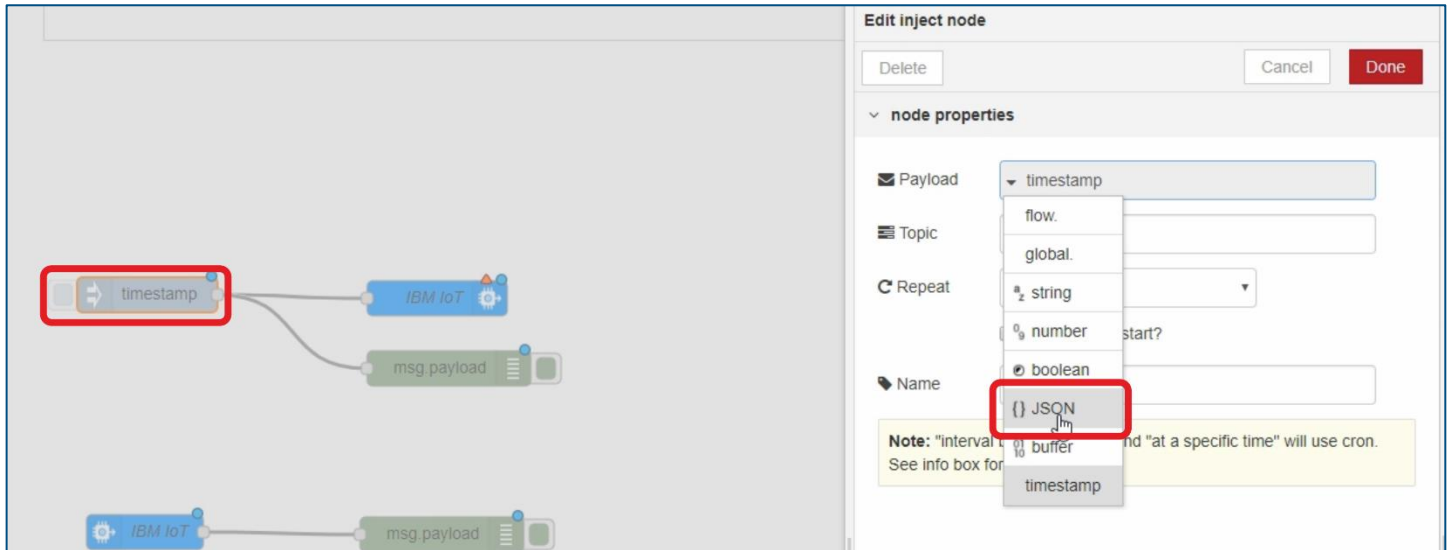
Insert **ibmiot input**, **ibmiot output**, two **debug outputs** and **inject input**. Connect the objects like this.



With the inject input we will send **DPA commands** to the MQTT broker on the IBM Cloud and our UP board will collect them from there. 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.

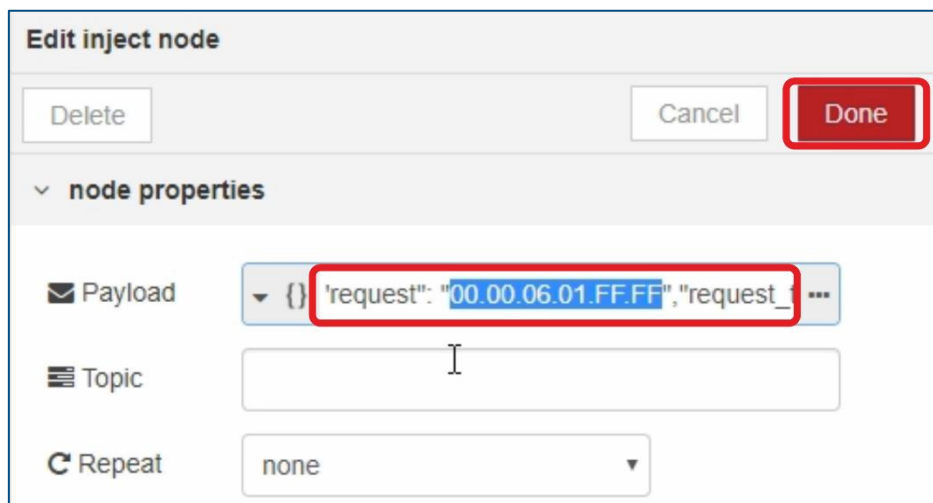
3.1.1 Modify the inject input

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



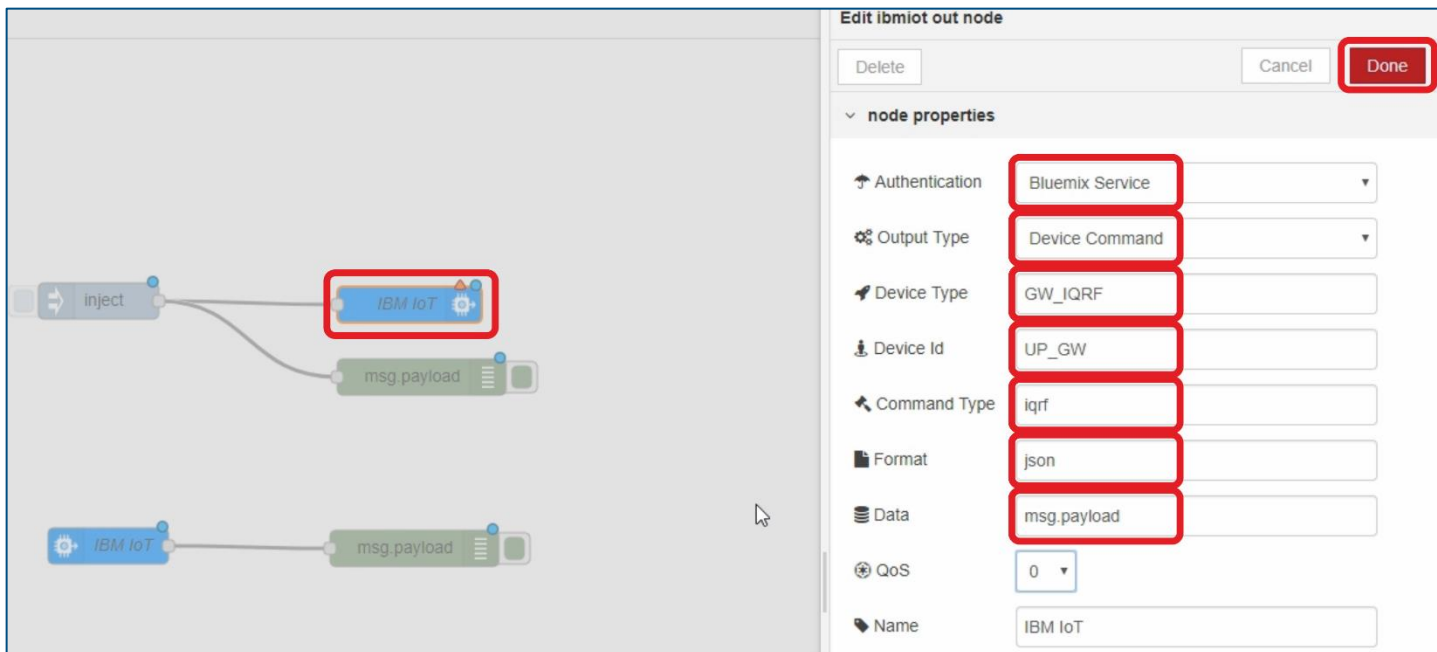
We used the command which turns on the red LED on the IQRF coordinator. You need to modify the command to one-line form. Click on **Done**.

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



3.1.2 Modify the ibmiot output

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**.



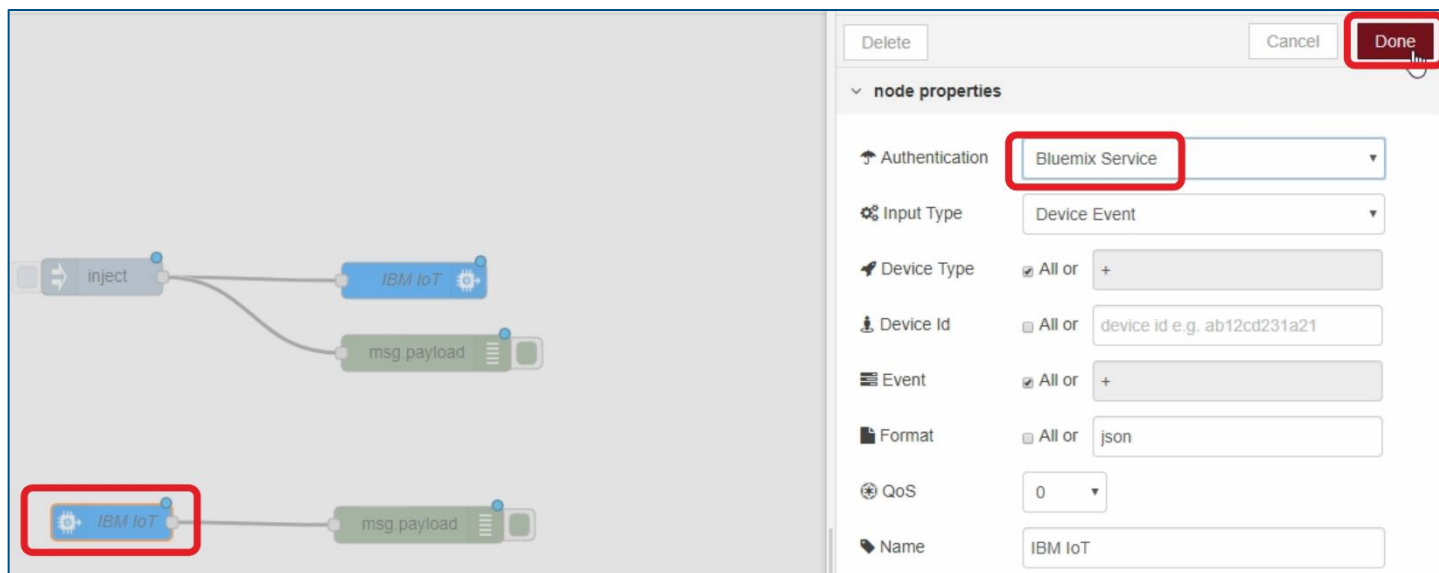
The screenshot shows the 'Edit ibmiot out node' dialog box. The 'node properties' section is expanded, showing the following configuration:

- Authentication: Bluemix Service
- Output Type: Device Command
- Device Type: GW_IQRF
- Device Id: UP_GW
- Command Type: iqrf
- Format: json
- Data: msg.payload
- QoS: 0
- Name: IBM IoT

The 'Done' button is highlighted in red.

3.1.3 Modify the ibmiot input

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



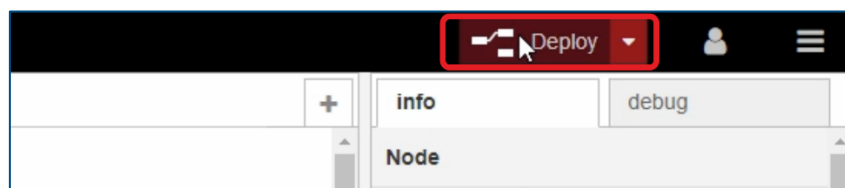
The screenshot shows the 'Edit ibmiot in node' dialog box. The 'node properties' section is expanded, showing the following configuration:

- Authentication: Bluemix Service
- Input Type: Device Event
- Device Type: ☒ All or +
- Device Id: ☐ All or device id e.g. ab12cd231a21
- Event: ☒ All or +
- Format: ☐ All or json
- QoS: 0
- Name: IBM IoT

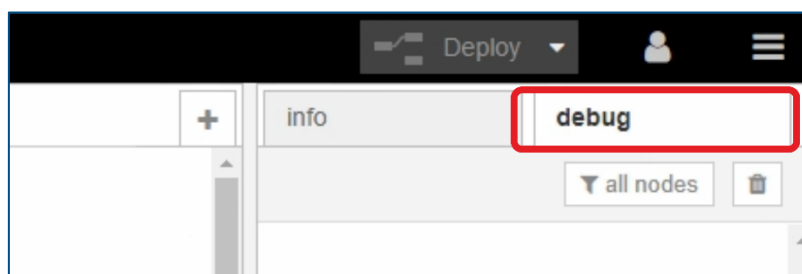
The 'Done' button is highlighted in red.

4 Test the connection

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.

5 Summary

The bidirectional communication between IQRF network and the IBM Cloud is up and running. Now it's just up to you to use it for your own IoT solution.