Project Objectives

Watson IoT :

Watson IoT Platform **connects devices, ingests device data, and transforms that data into meaningful insights using capabilities such as AI, analytics and blockchain.**

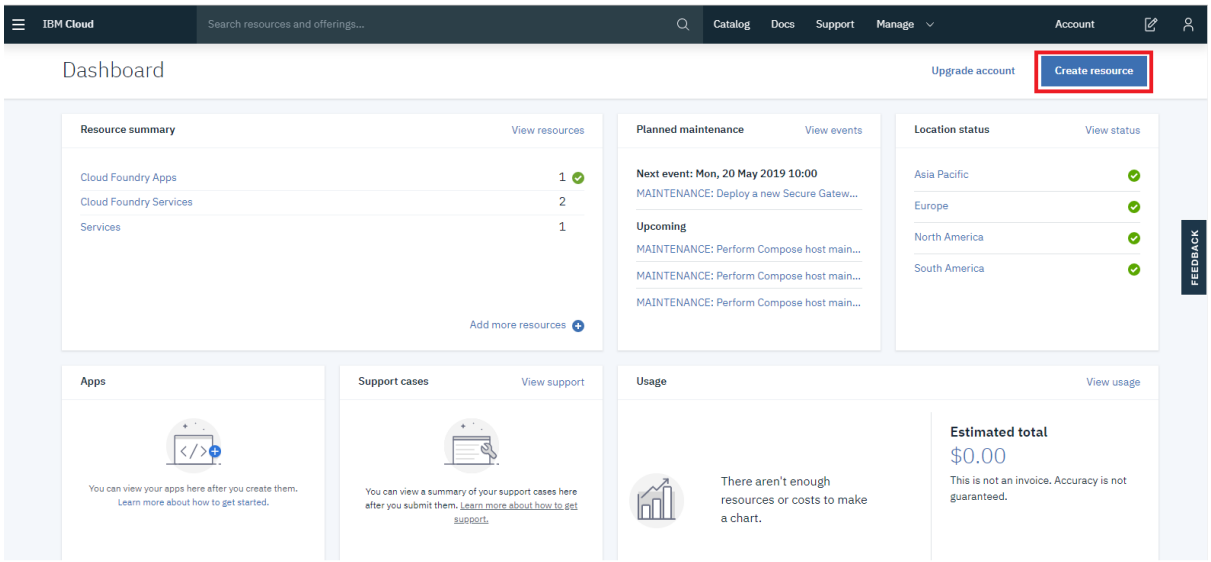
Watson **helps organizations predict future outcomes, automate complex processes, and optimize employees' time**. Watson IoT can connect IoT devices, networks and gateways through a growing ecosystem that uses open standards-based communications (MQTT, HTTPS). Globally scalable, starting with a single device.

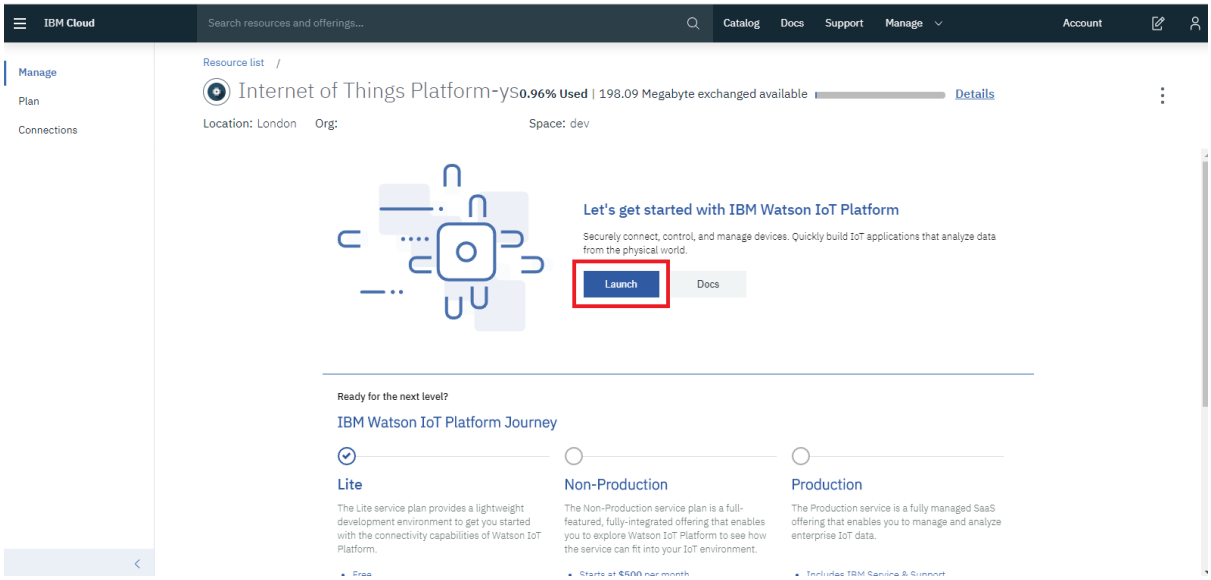
IoT platforms provide many integrated services and infrastructures like data storage, connectivity, display, control etc. Hence, they reduce the amount of investment required to deploy IoT solutions and this is one of the main reasons behind some of the most successful IoT solutions around. IBM Watson IoT platform is an industry grade IoT platform frequently used by big industries to store and analyze the data gathered from IoT devices.
TapNLink could be used as a facilitator for connecting your device to IBM Watson IoT Platform. In this tutorial we will setup an IBM Watson account, connect a TapNLink and connect IoTize Studio to it.

Step1: Setup your IBM Watson IoT Platform account

The first step is to create an IBM Cloud account: https://cloud.ibm.com/.

Then create an IBM Watson IoT Platform, by going to "Create resource" on the IBM Cloud Dashboard and selecting the Internet of Things platform.



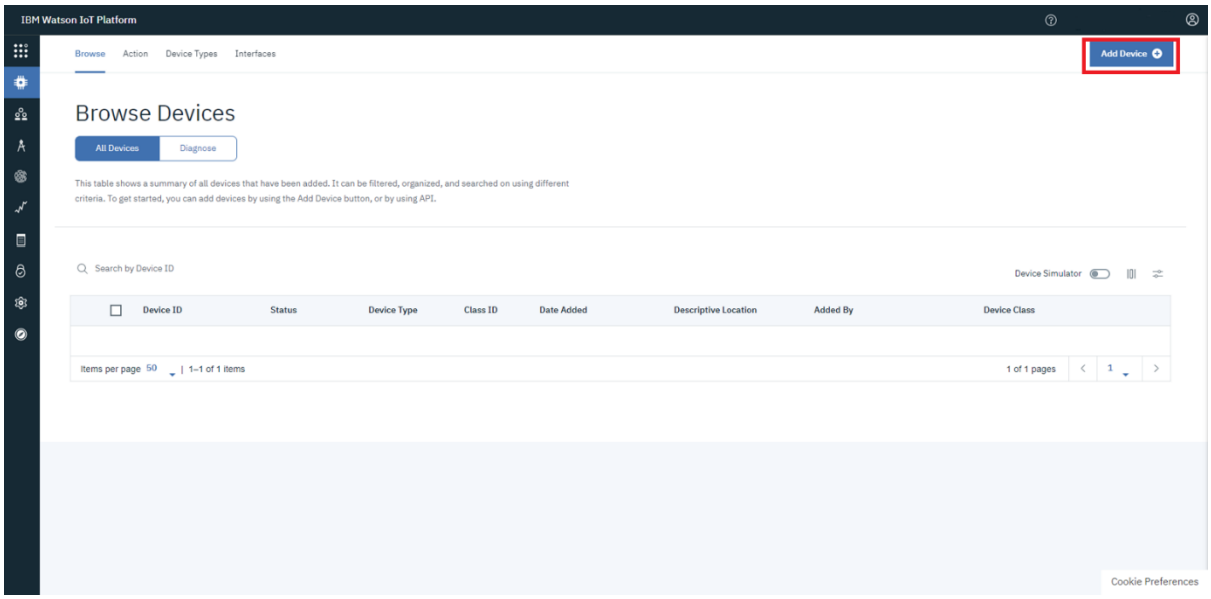


Step2: Register your device on your IoT Platform

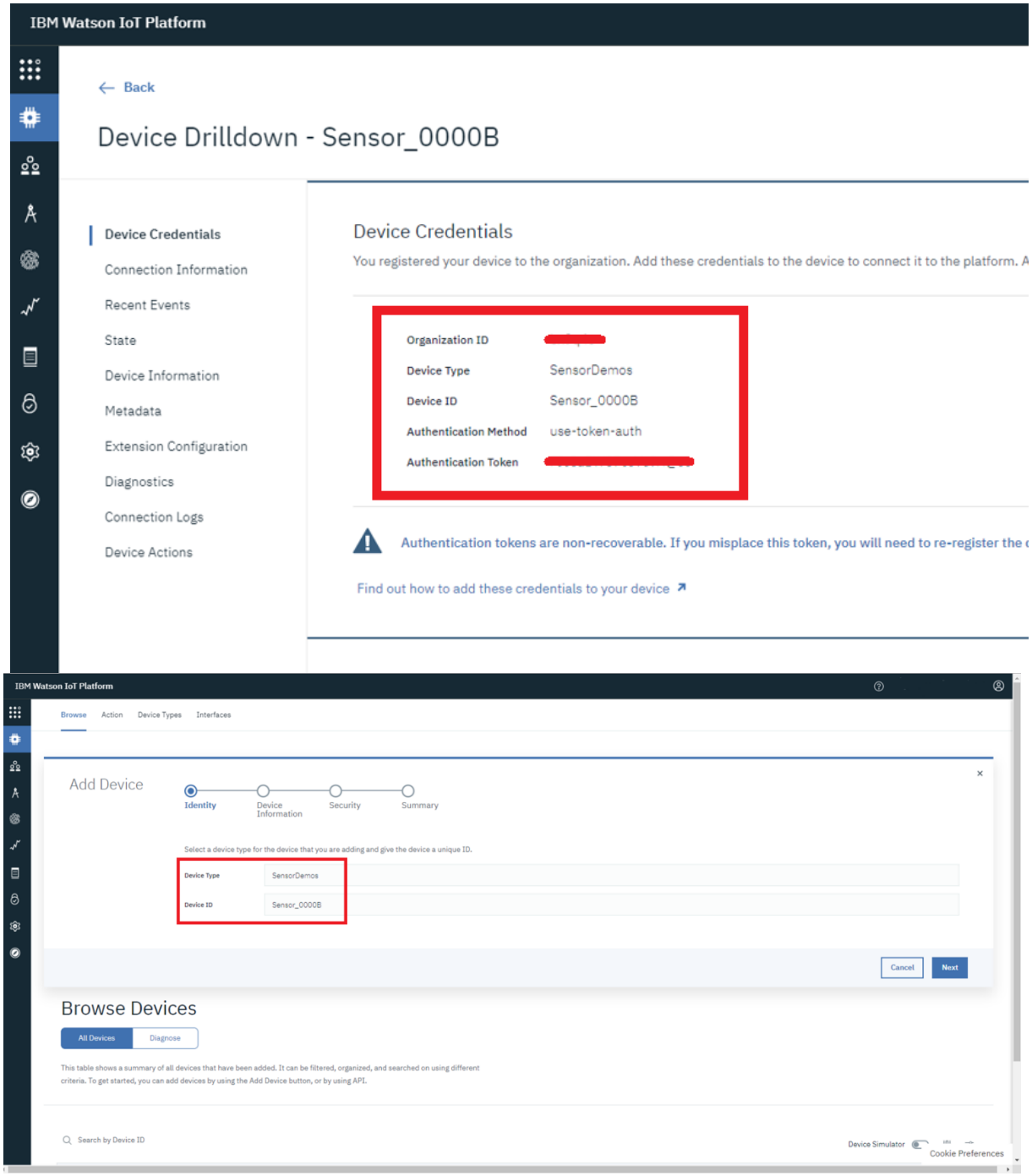
In this step we will create a device twin in Watson IoT Platform service that represents your physical device.

Watson IoT Platform service includes a device twin feature. This cloud-based digital representation of your device is connected to Watson IoT Platform service. Once it is defined and instantiated, the device twin provides a consistent means of interacting with your device from the IoT hub.

- **Within your IBM Watson IoT Platform, select Devices tab and click on Add Device button**



- **Fill in the device details, the DeviceID will be a unique identifier for your device. You can use the DeviceType to group your devices.**



Step3: Configure TapNLink to handle IBM Watson Connectivity

The STM32 Blue Pill board that is included in the TapNLink Primer Evaluation Kit is pre-programmed with the STM32_Sensor application which demonstrates a few simple features like blinking the LED, measuring the internal temperature and voltage etc. In this guide, we will configure TapNLink to send some information to Watson IoT Platform.

Launch IoTize Studio and open sensor_demo.iotz config file installed in the Sensors_STM32_Demo subdirectory of the installed examples.

Select IoT Platform (MQTT) and setup the configuration:

- **Set Enable Relay to Yes.*This allows the tap to use MQTT to receive LWM2M commands***
- **Select IBM Watson in IoT Platform**
- **Cloud Profile: A specific profile to control access privileges of the connected 'IoT Platform'.**
- **IBM Watson information: Provide the previously created Device Twin information to enable TapNLink to connect directly to your Watson IoT Platform.**
    - **Organization ID**
    - **Device Type**
    - **Device ID**
    - **Authentication Token**
    - **IBM Watson messaging root certificate**: If you set up a Root CA to aunthenticate your devices, set it here. Leave it empty otherwise
    - **Use SSL protocol**
    - **IBM Broker login summary (MQTT)**: This shows the actual MQTT connection information that will be used by the Tap. These are created from the **IBM Watson information**. It also shows the topics used to receive commands and send answers.

IoT Platform (MQTT)	
Enable Relay	Yes
Enable Cloud	No
Platform	
IoT Platform	IBM Watson
Cloud Profile	anonymous
IBM Watson information	
Organization ID	8g4z8j
Device Type	SensorDemo
Device ID	1106
Authentication Token	5&GcJ8UcfzrKk(KDb?
IBM Watson messaging root certificate	
Use SSL protocol	No
IBM Broker login summary (MQTT)	
Host name/IP address	8g4z8j.messaging.internetofthings.ibmcloud.com
Service name/Port	1883
Username	use-token-auth
Password (token)	5&GcJ8UcfzrKk(KDb?
Client ID	d:8g4z8j:SensorDemo:1106
Topic for command	iot-2/cmd/COMMAND/fmt/txt
Topic for answer	iot-2/evt/ANSWER/fmt/txt

Step4: Setup the WiFi Settings and Configure TapNLink

Set Incoming communication (Wireless) | WiFi:

- **Network mode to 'Network(Station)'**
- **SSID to your WiFi network**
- **WEP key to your WiFi network's security key,**

Click on the "Configure" button to re-configure TapNLink, then use Test|Reboot Tap to restart TapNLink. Now TapNLink will dynamically connect to your IBM Watson IoT Platform and wait for any incoming LWM2M request.

Step5: Connect IoTize Studio to IBM Watson

In order to connect IoTize Studio to IBM Watson, we need to create an application API Key. We only need to create it once: the API Key will be able to communicate with every device you registered in your IBM Watson cloud service.

- **In the Platform Service dashboard, go to Apps > API Keys.**
- **Click on Generate API Key. You will then see the API Key and its token. It is very important that you keep a note of your token when it appears on the summary screen, as when you proceed past the summary stage, the token will not appear again.**
- **Click on Finish.**

Now, go to **Studio | Connection to Tap** in IoTize Studio:

- Set **Protocol** to **MQTT Relay**
- Set **Adapt broker information from Tap MQTT settings** to **Yes**.
- Set **Application API Key** and **API Key Token** to the one you created on your IBM Watson dashboard.

Connection to Tap	
Tap Id (serial number)	IoTize006100000000
Encryption	Low
Protocol	MQTT Relay
Adapt broker information from Tap MQTT settings	Yes
IoT Platform	IBM Watson
MQTT Host name	8g4z8j.messaging.internetofthings.ibmcloud.com
MQTT Service name/Port	8883
Client ID	a:8g4z8j:IoTizeStudio
Application API Key	a-8g4z8j-ylgfujnrcb
API Key Token	-dNtOkW+XltAw)RQik
Broker Certificate	
Topic for command	iot-2/type/SensorDemo/id/1106/cmd/COMMAND/fmt/txt
Topic for answer	iot-2/type/SensorDemo/id/1106/cmd/COMMAND/fmt/txt
Netkey	testnetkey

Click on **Monitor**. IoTize Studio will connect to the MQTT Broker, and communicate with the Tap. You are now able to communicate with your Tap through your IBM Watson MQTT broker

PYTHON SCRIPT:

```
import random

from time import sleep

def generate_values():

  temperature = random.randint(10, 50)
```

```
  humidity = random.randint(10, temperature)
  return humidity, temperature
humidity = temperature = 0
while temperature < 45:
  humidity, temperature = generate_values()
  print('Humidity:', humidity, 'Temperature:', temperature)
  sleep(0.50)
print('High Temperature Detected')
```

DEVELOP A NODE RED SERVICES:

Create a Node-RED starter application

Follow the steps in this tutorial, "Create a Node-RED starter application."

After the status of your application changes to *Running*, click Visit App URL. Click Go to your Node-RED flow editor. A new flow called Flow 1 is opened in the Node-RED flow editor. If you secured your Node-RED flow editor, you will first be asked to enter the username and password that you just set up.

Develop the Application

Our Earthquake Monitoring System application has two main components:

- **The *web service* component**
- **The *dashboard* component**

The steps to create these two components in our Node-RED app follow. I encourage you to follow the steps to learn just how easy it is to create apps using Node-RED.

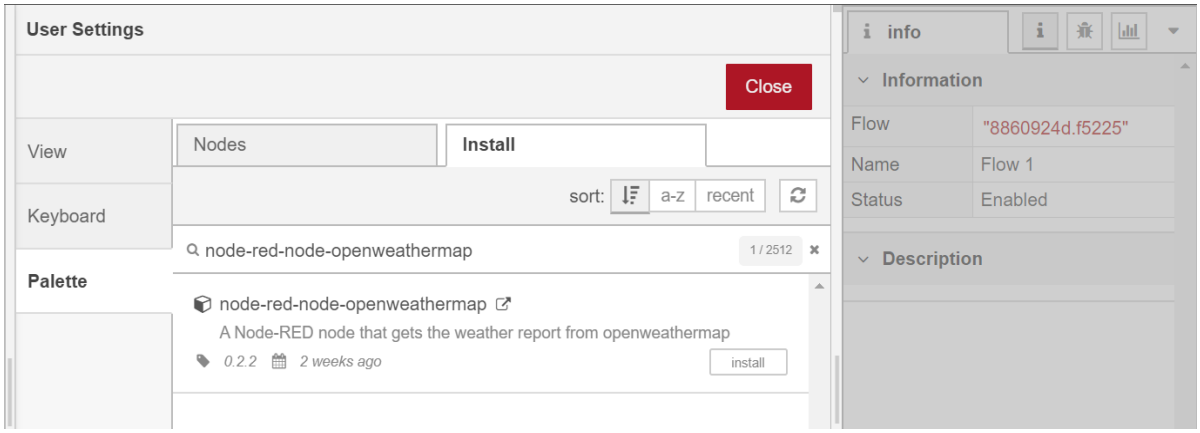
You can also import both of the flows explained in this tutorial. First, download or copy the contents of the flows.json file to the

clipboard. Then, go to the hamburger menu in your Node-RED editor, and select Import > Clipboard. Then, paste the contents into the dialog, and click Import. In case you selected to download the file, make sure to select select a file to import, then click Import. You'll still need to follow the steps in this tutorial to configure all the nodes and to make a package available to a function node.

Create the Web Service

1. **Double-click the tab with the flow name, and call it Earthquake Details.**
2. **Click the hamburger menu, and then click Manage palette. Look for node-red-node-openweathermap to install these additional nodes in your palette.**

3. **Add an HTTP input node to your flow.**
4. **Double-click the node to edit it. Set the method to GET and set the URL to /earthquakeinfo-hr.**
5. **Add an HTTP response node, and connect it to the previously added HTTP input node. All other nodes introduced in this sub-section is to be added between the HTTP input node and the HTTP response node.**
6. **Add an HTTP request node and set the *URL* to https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.geojson, the *Method* to GET and the *Return* to a parsed JSON object. This will allow extracting all earthquakes that occurred within the last hour. Name this node Get Earthquake Info from USGS.**


User Settings
Close
View
Keyboard
Palette
Nodes
Install
sort:
a-z
recent
node-red-node-openweathermap
1 / 2512
node-red-node-openweathermap
A Node-RED node that gets the weather report from openweathermap
0.2.2
2 weeks ago
install
info
Information
Flow
"8860924d.f5225"
Name
Flow 1
Status
Enabled
Description