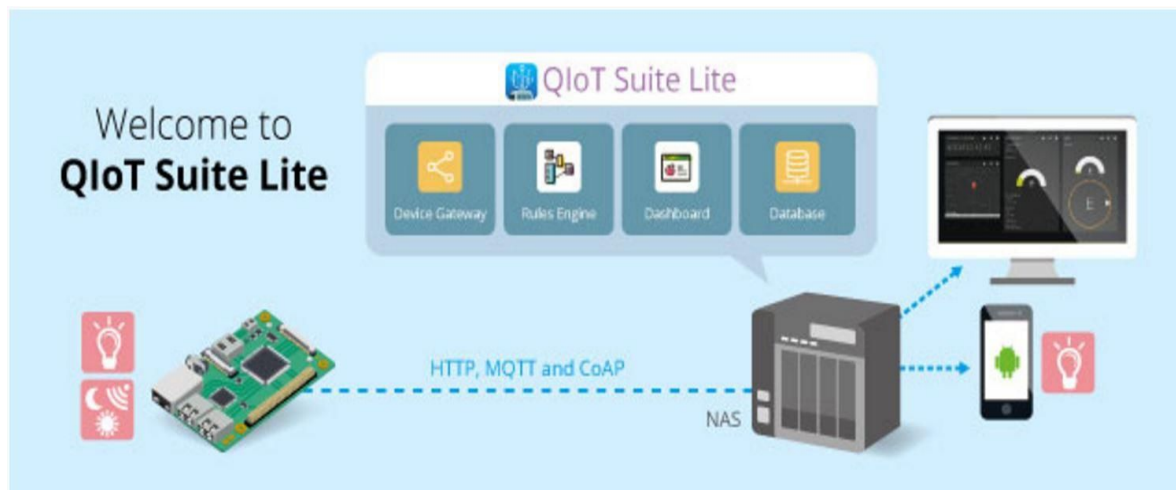


Get started with Raspberry Pi (Nodejs)

In this tutorial, you begin by learning the basics of working with Raspberry Pi that's running Raspbian which is a free operating system based on Debian optimized for the Raspberry Pi. You then learn how to seamlessly connect your devices to QNAP NAS by using QIoT Suite Lite. Please ensure your Raspberry Pi and NAS is under the same LAN.



Lesson 1: Configure your device

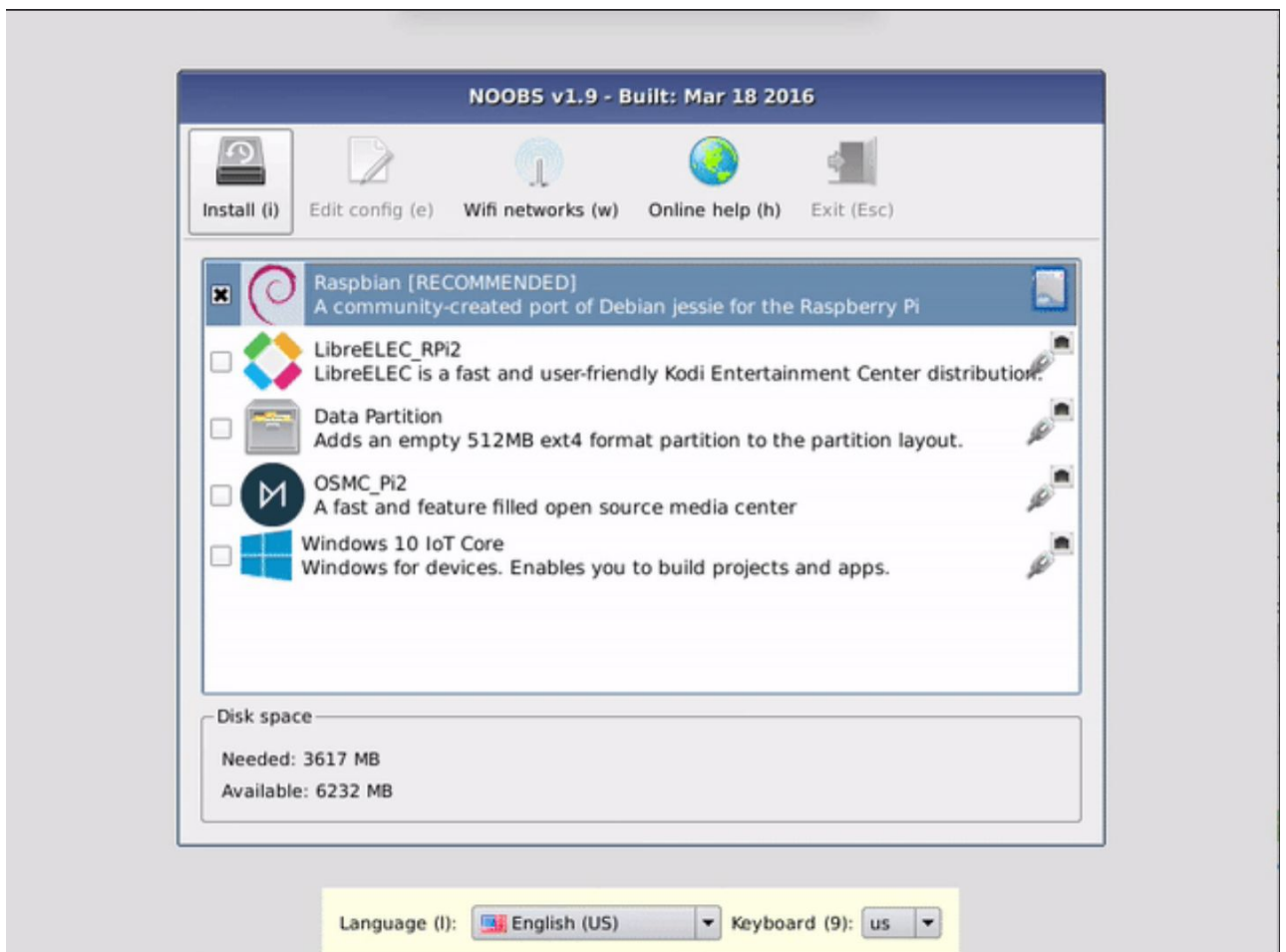
In this lesson, you configure your Raspberry Pi device with an operating system, set up your development environment, and deploy an application to Raspberry Pi.

1.1 What will you need

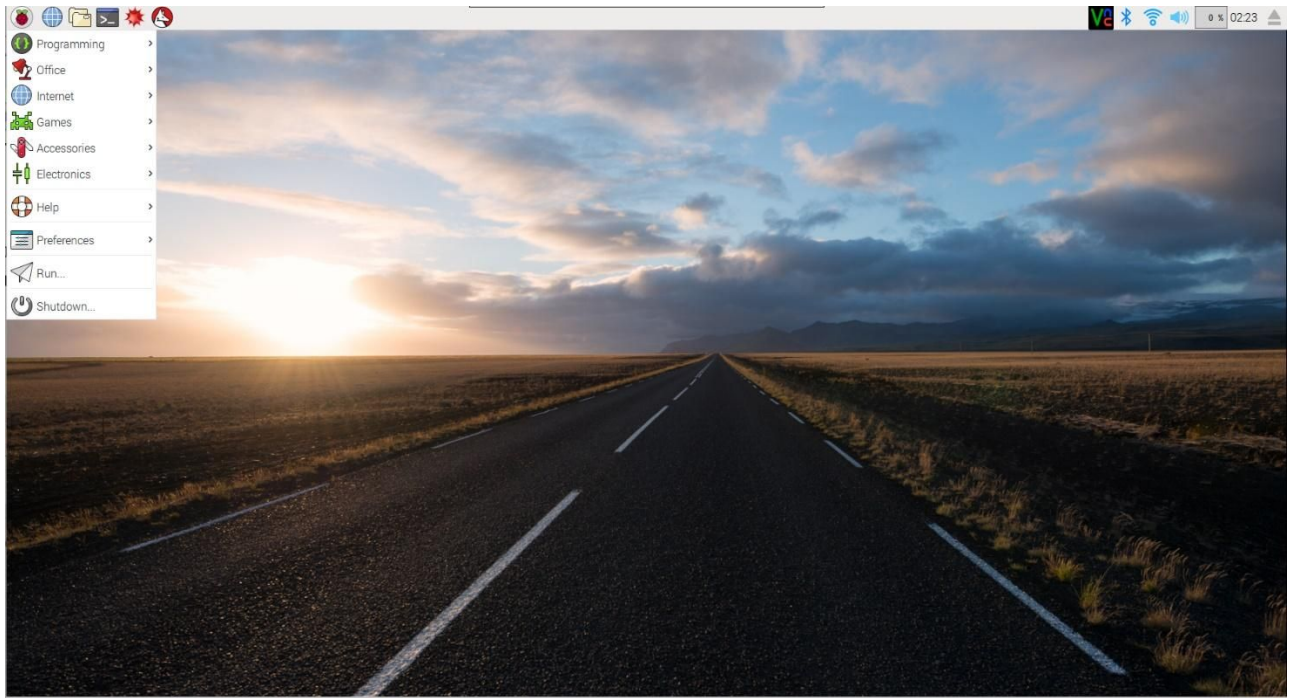
- A Raspberry Pi
- SD card : an 8GB class 4 SD card is recommended, ideally preinstalled with [NOOBS](#).
- A HDMI television or monitor : You will need to connect your Raspberry Pi to a display, which means you'll need an HDMI-enabled screen of some kind.
- Keyboard and mouse : Any standard USB keyboard and mouse will work with your Raspberry Pi.
- Power Supply : The Pi is powered by a USB Micro power supply (like most standard mobile phone charger). You'll need a good-quality power supply that can supply at least 2A at 5V for the Model 3B, or 700mA at 5V for the earlier, lower powered models.

1.2 Set up Raspbian

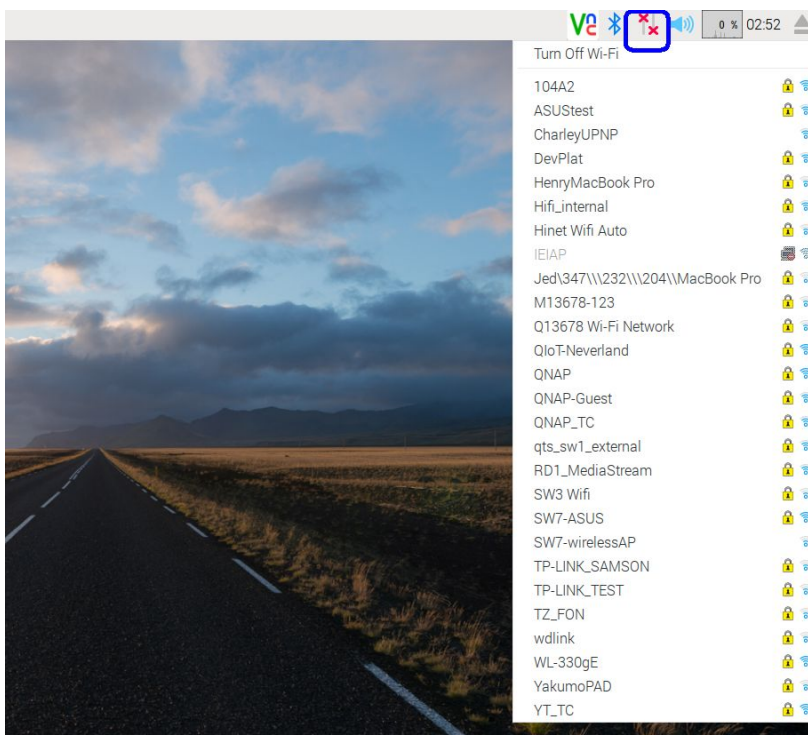
- Put the SD card containing extracted NOOBS files into Raspberry Pi
- Connect keyboard, mouse, and HDMI monitor to the Raspberry Pi, then plug in your power adapter. The Raspberry Pi does not have a power switch, so once you connect the power adapter, it'll turn on all by itself.
- A simple menu will ask which OS you want to install. At the bottom of the screen, choose your language and keyboard layout for your region. Click the checkbox next to Raspbian, then click Install.



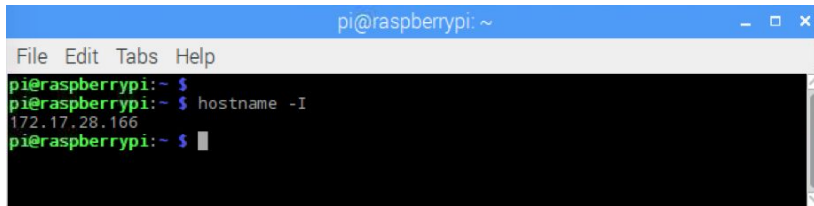
- Now, let NOOBS run the installation process, which may take 10 or 20 minutes. When it's complete, it will restart and send you straight into the Raspbian desktop, where you can configure everything else.



- **Configure WiFi** – In the Desktop environment locate the Wireless Network icon in the top right hand side and click on the icon to see the list of available WiFi networks to connect to. Select your WiFi SSID and enter correct password, then you should be able to connect to your WiFi.



- In order to connect to your Raspberry Pi from another machine, you need to know the Pi's IP address. Using the terminal (Ctrl + Alt + t to open Terminal), simply type `hostname -I` which will reveal your Pi's IP address.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~ $  
pi@raspberrypi:~ $ hostname -I  
172.17.28.166  
pi@raspberrypi:~ $
```

- More Raspberry Pi setup guide, please refer to <https://www.raspberrypi.org/learning/software-guide/>

1.3 Install Nodejs and required libraries

- Press Ctrl + Alt + t to open Terminal.
- Install the latest nodejs

```
pi@raspberrypi:~ $ sudo apt update  
pi@raspberrypi:~ $ sudo apt install nodejs
```

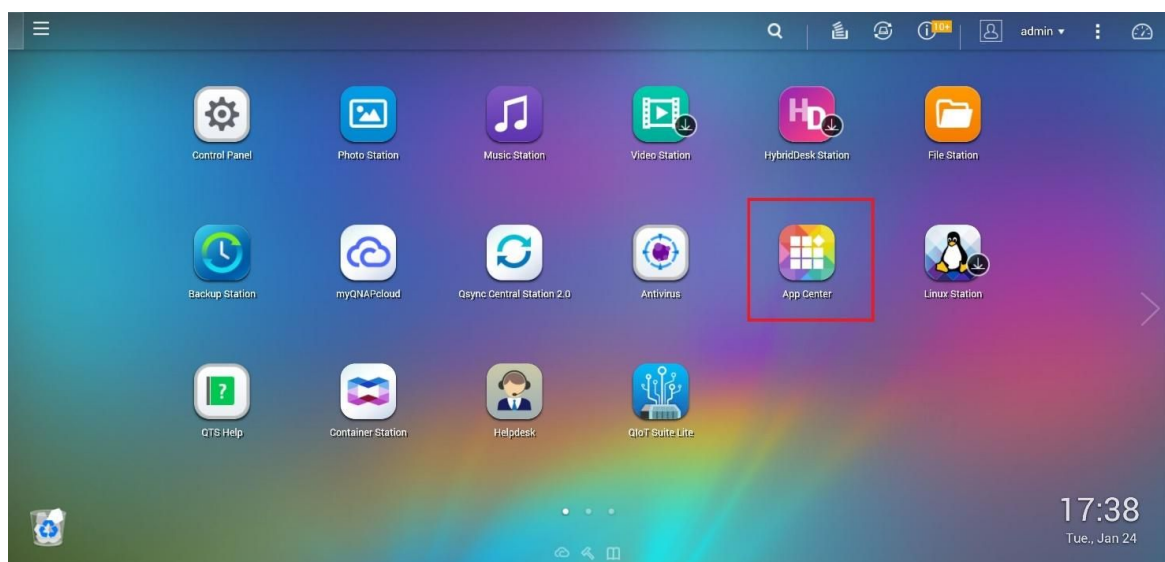
Lesson 2: Create your device in QIoT Suite Lite

In this lesson, you provision your QNAP QIoT Suite Lite software, and create your first device in QIoT Suite Lite.

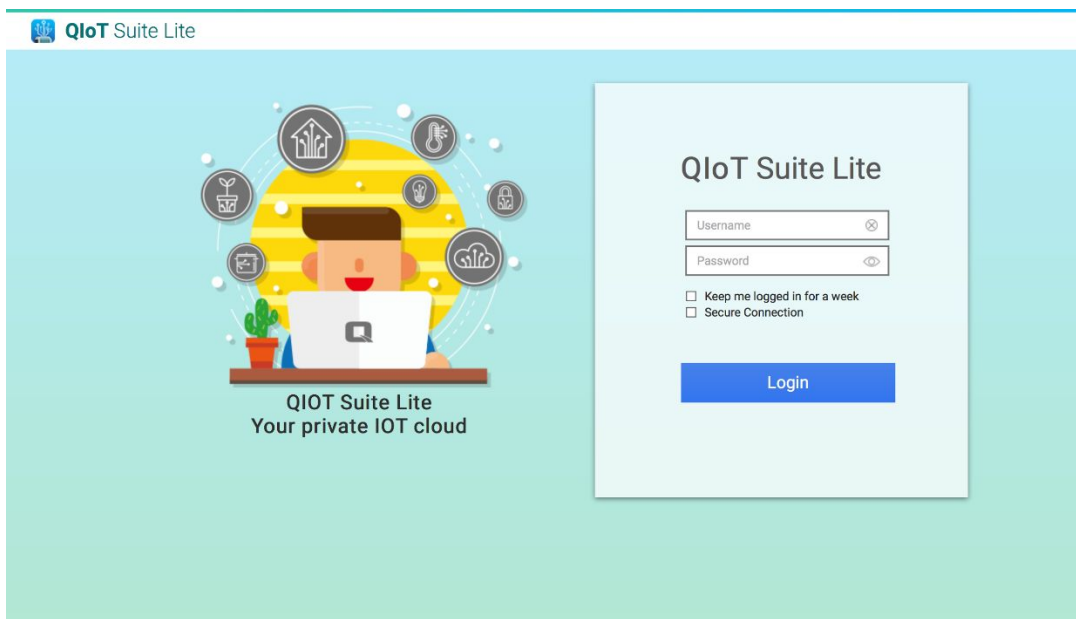
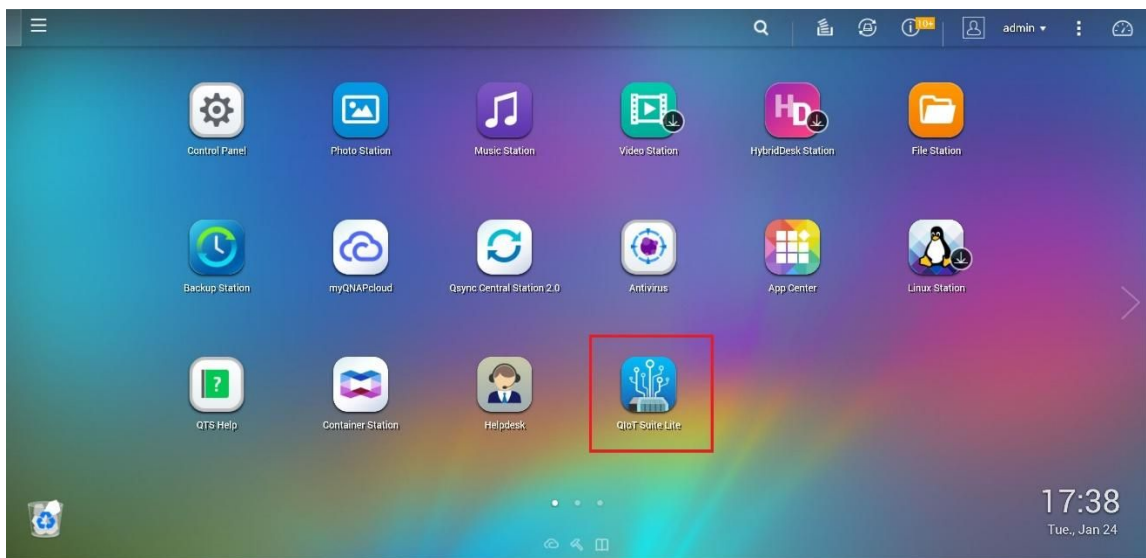


2.1 Install QIoT Suite Lite

- Go to QNAP App Center and download QIoT Suite Lite software.



- Launch and log in QIoT Suite Lite software. Use Nas admin and password to login.



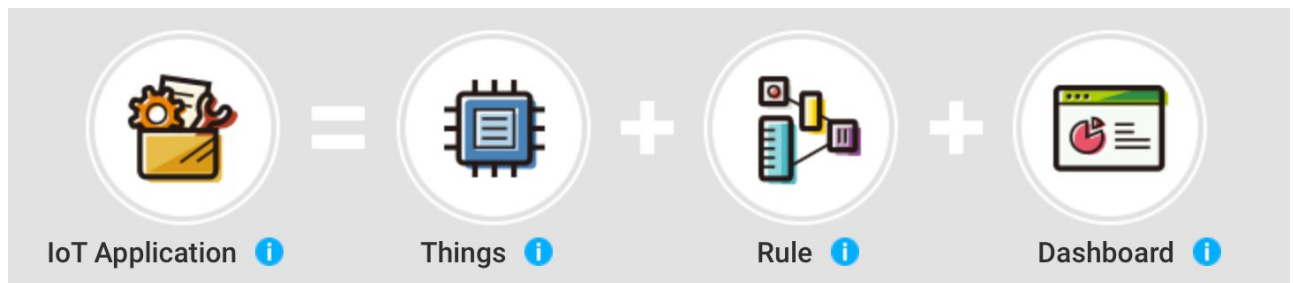
2.2 Create a new IoT application

IoT Application is a combination of multiple Things, Rule, and Dashboard. We recommend that you first create a “Things” in IoT Application. This IoT Application allows you to keep a record of all of the devices that are connected to your NAS.

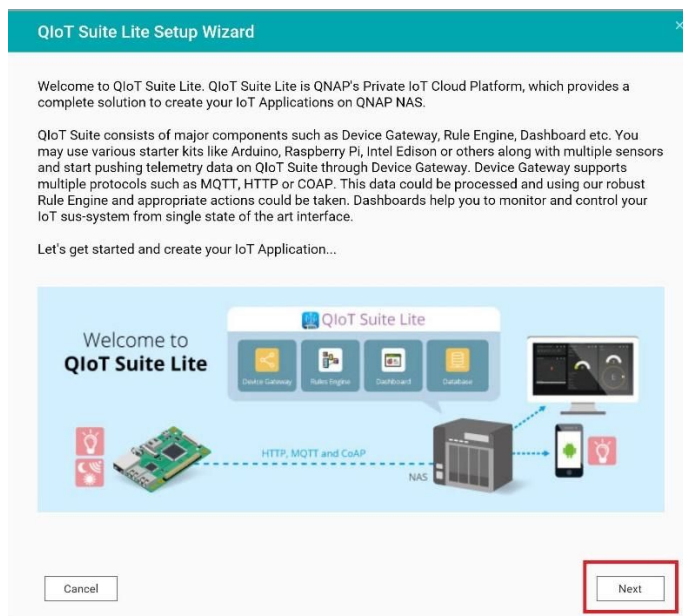
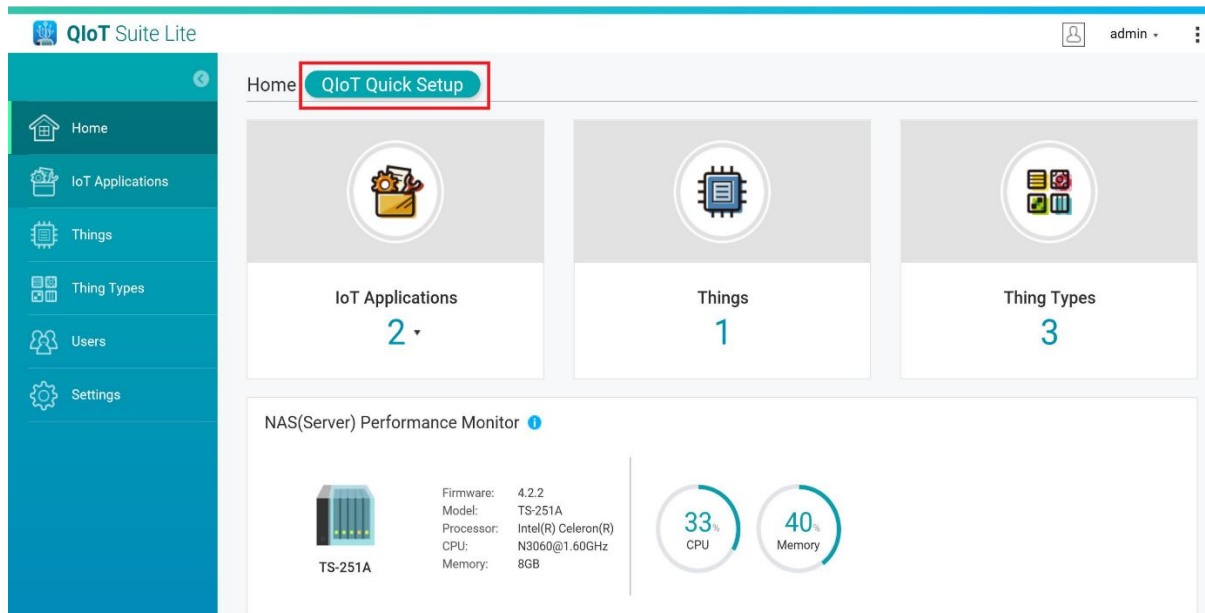
The Rule makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices based on business rules you define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel.

With Dashboard, you can turn your data processing efforts into analytics and reports that provide real-time insights into your business.

All these elements provide user a complete IoT Application environment.



- If this is your first time to use QIoT Suite Lite, QIoT Suite Lite provides a wizard to help you quick setup a IoT application. Click on **QIoT Quick Setup**. After you read the QIoT Suite Lite introduction, click **Next** button.







- In **IoT Application** tab, type a name for your IoT Application, e.g. myIoT_App-1. Rule name and Dashboard name will be generated automatically based on the name of IoT Application you fill in. Then click **Next** button.

QIoT Suite Lite Setup Wizard




IoT Application Thing Resource API Keys

Create a New IoT Application

IoT Application is a combination of multiple Things, Rule and Dashboard. All these elements as an IoT Application provides user complete IoT Application environment. Please use enter following information to create an IoT Application.

IoT Application Things Rule Dashboard

IoT Application Name *:

Description:

Rule Name:

Description:

Dashboard Name:

Description:

Note: Inputs with * are required field


- In **Thing** tab, click on **Add**. Please provide a name for your device (ex. myThing-1). You can also choose **Add attribute** to provide information about your device (for example, its serial number, manufacturer, and more). If your device is already supported by QIoT software, you can choose **Thing Type** predefined by QNAP. Click on **Add** to add the device.

QIoT Suite Lite Setup Wizard

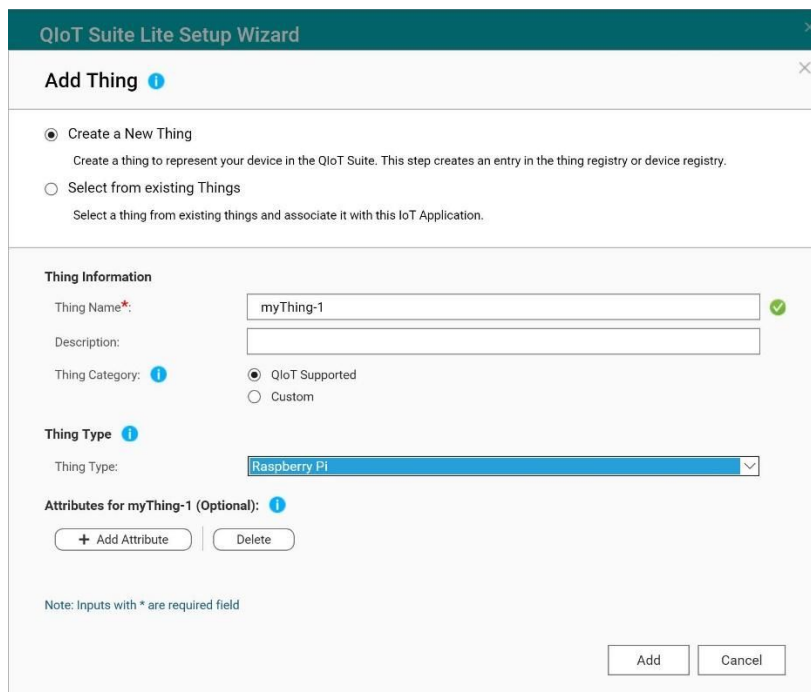
IoT Application Thing Resource API Keys

Add Things

You may add a new thing or select an existing thing from thing list below. In next step, you may then integrate your things with QIoT and receive related readings from things.

<input type="checkbox"/>	Thing Name	IP	User Name	Password	Action
 <p>You do not have any Thing.</p> <p>Please click on "Add Thing" button to add new Thing.</p>					

Note: You must connect every thing in this QIoT application first.



QIoT Suite Lite Setup Wizard

Add Thing

☒ **Create a New Thing**
 Create a thing to represent your device in the QIoT Suite. This step creates an entry in the thing registry or device registry.

☐ **Select from existing Things**
 Select a thing from existing things and associate it with this IoT Application.

Thing Information

Thing Name*: ✔
 Description:
 Thing Category: i
☒ QIoT Supported
☐ Custom

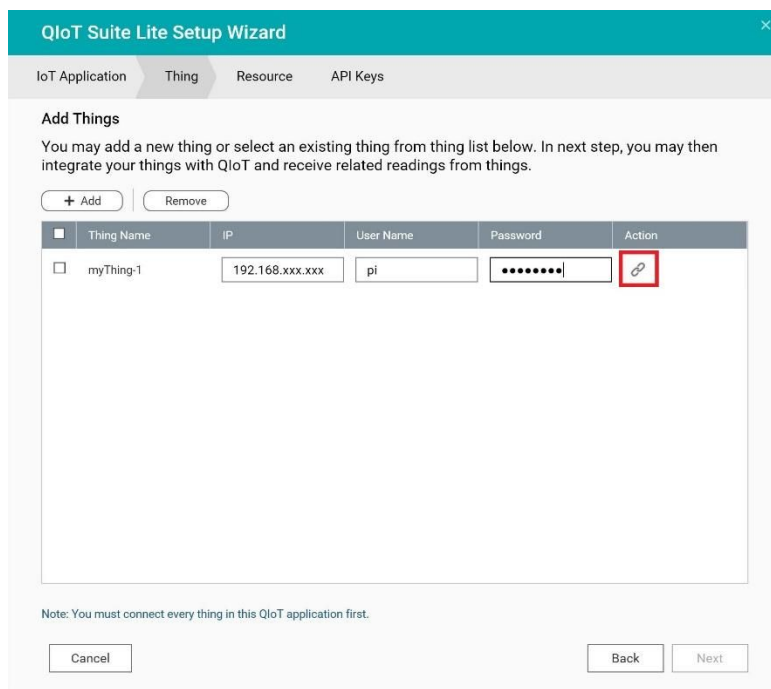
Thing Type i

Thing Type: Raspberry Pi

Attributes for myThing-1 (Optional): i

Note: Inputs with * are required field

Please provide IP address, User Name, and Password of your device, then click Connect icon. QIoT Suite Lite will check the connection with the device. After the connection is verified, please click **Next**.



QIoT Suite Lite Setup Wizard

IoT Application **Thing** Resource API Keys

Add Things

You may add a new thing or select an existing thing from thing list below. In next step, you may then integrate your things with QIoT and receive related readings from things.

<input type="checkbox"/>	Thing Name	IP	User Name	Password	Action
<input type="checkbox"/>	myThing-1	192.168.xxx.xxx	pi	🔗

Note: You must connect every thing in this QIoT application first.

- In **Resource** tab, click on **Add Resource**. Give name and ID for the sensor you use into **Resource Name** and **Resource ID**. Please be noted that Resource ID will

be used to create a Topic in the QIOT Broker. This ID should be unique for the device and no duplicates should be allowed for the same device.

QIoT Suite Lite Setup Wizard

IoT ApplicationThingResourceAPI Keys

Add Resource

Thing Resources represent Sensors, Thing Peripherals, Switches or any dynamic properties of the Thing like CPU usage etc. Please add these resources to the device with desired configurations.

+ Add Resource

Delete

Thing Name:

myThing-1(0)

Resource Name	Resource ID	Resource Type	Data Type	Unit	Description	Actions
<div><div></div><div>You do not have any Thing Resource.</div><div>Please click on "Add Resource" button to add new resource.</div></div>						

Page 1 / 0

1 - 0 of 0

Cancel

Back

Next

Add Thing Resource

Resource Name*:myTemperature-1

Resource Description:Please enter the description

Resource ID*:Id_myTemperature-1

Resource Type:Temperature

Data Type:Float

Unit:*C

Set Range

Minimum Value:Please enter minimum value

Maximum Value:Please enter maximum value

Set Default Value

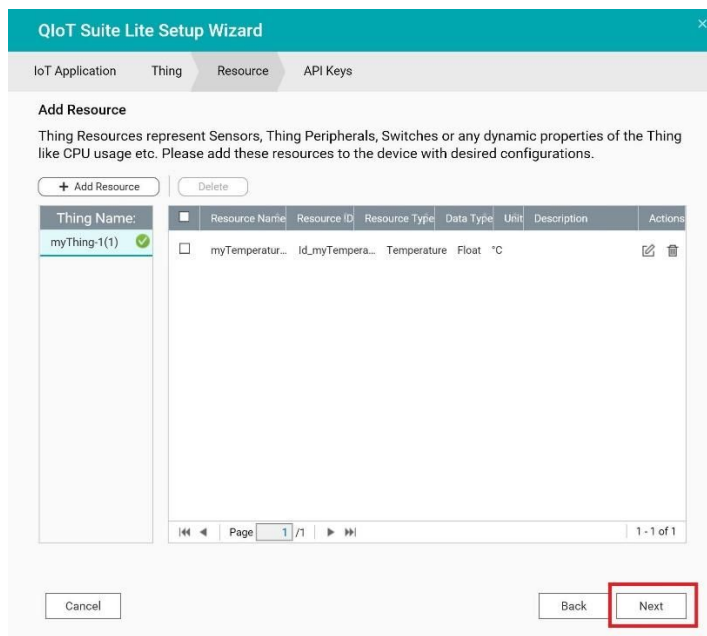
Default Value:Please enter default value

Note: Inputs with * are required field

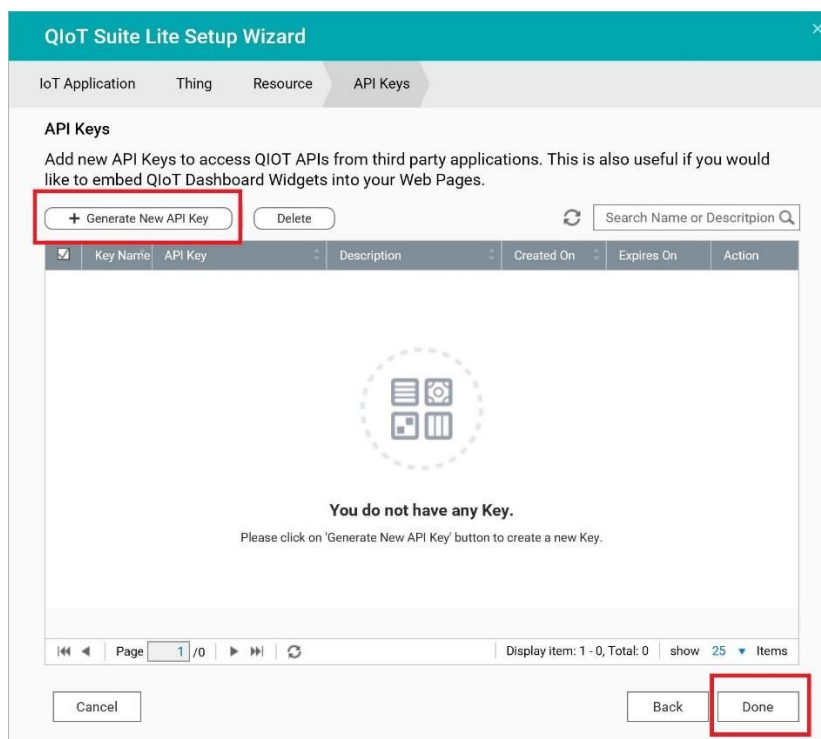
Add

Cancel

Click **Next** after you add all the resources (sensors) on your device.

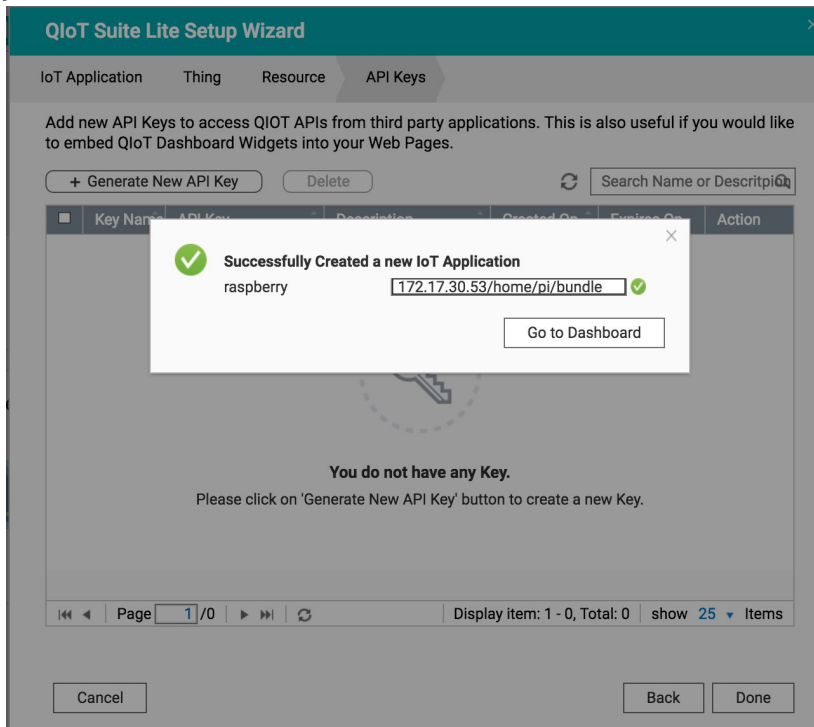


- In API Keys tab, click on **Generate New API Key** if you would like to embed QIoT dashboard widgets into your web pages or access QIoT APIs from third party applications. Click on **Done**.

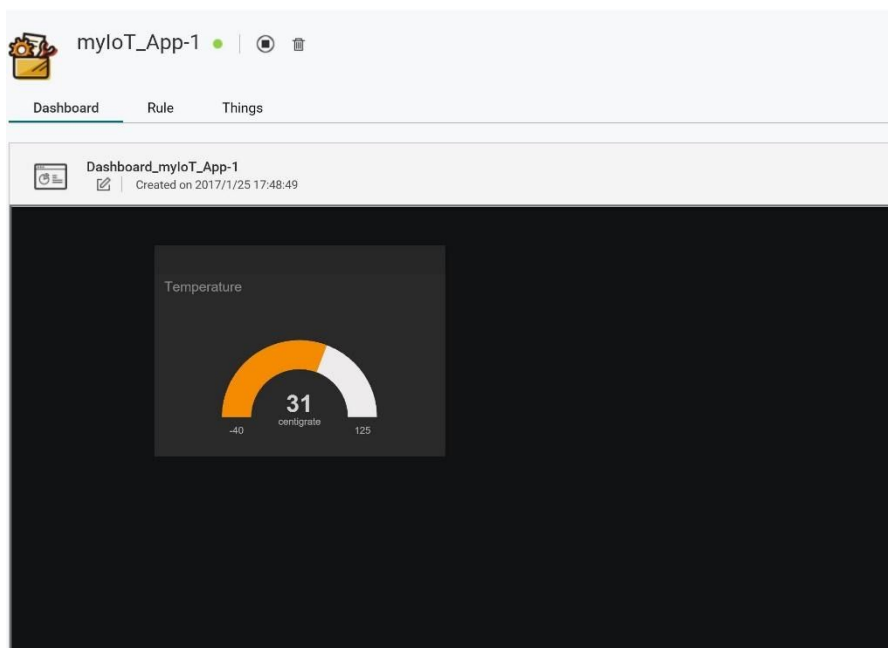


Your IoT application is created successfully. Node.js sample codes and related

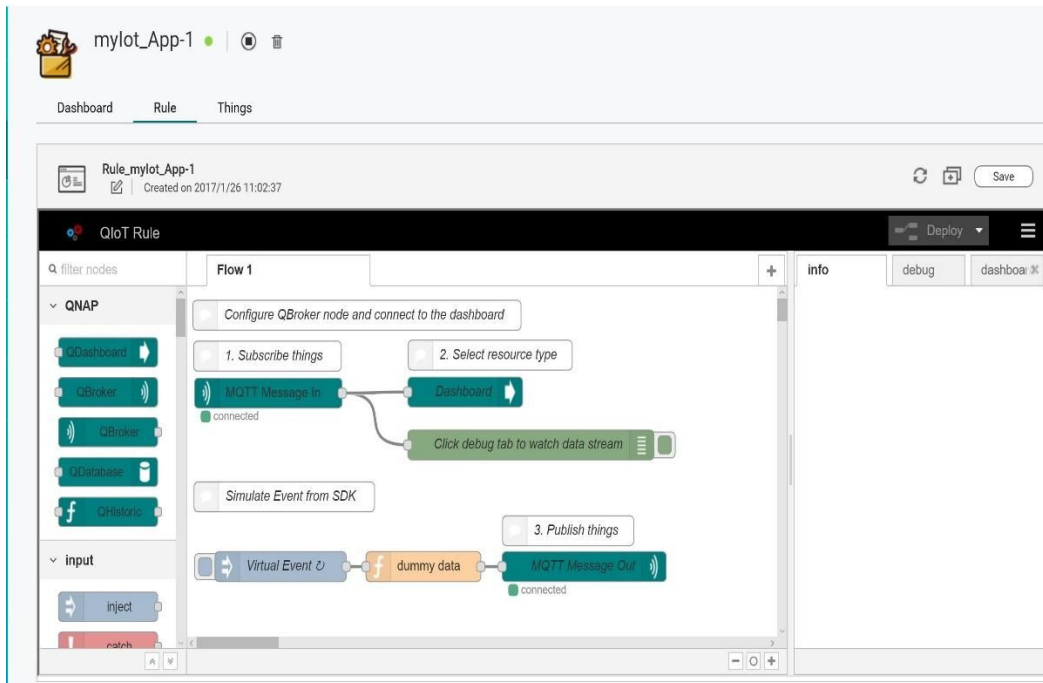
files (certificate, resource information) have been upload to the specified path on your device.



- A sample dashboard is created.

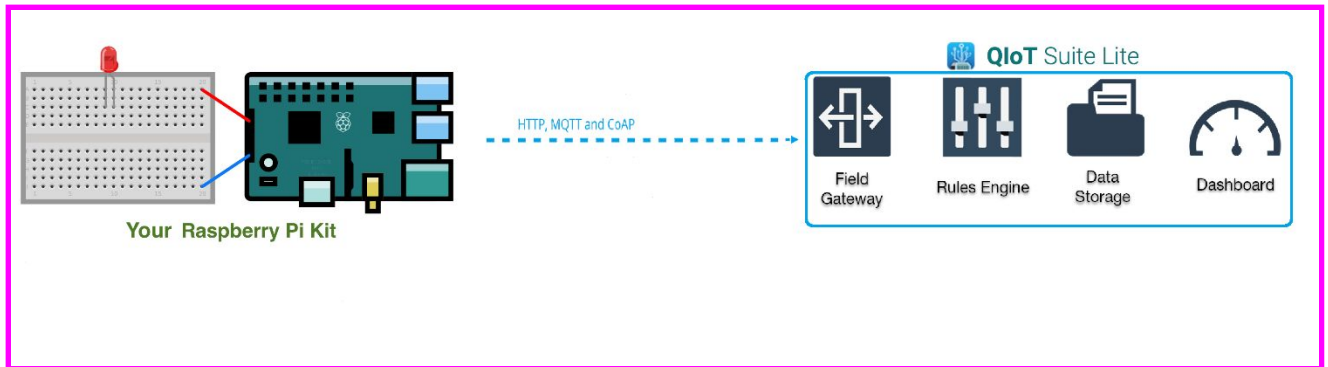


You may go to **Rule** tab to define the flow or rule about how to process the data sent from the device, and how to present in dashboard.



Lesson 3: Connect your device to QIoT Suite Lite using MQTTS

In this lesson, you generate certificate from QIoT Suite Lite, download SDK, and connect Raspberry Pi to QIoT Suite Lite.



3.1 Connect to QIoT Suite.

- On your Raspberry Pi, open Terminal application.
- Go to the directory where QIoT Suite Lite uploads sample codes to.

```
pi@raspberrypi:~ $ cd /home/{{user}}/bundle
pi@raspberrypi:~ $ npm install
```

```
pi@raspberrypi:~/bundle $ pwd
/home/pi/bundle
pi@raspberrypi:~/bundle $ ls
coap.js  coap-observe.js  http-get.js  http.js  https-get.js  https.js  lib  mqtt.js  mqtt-subscribe.js  package.json  res  ssl
pi@raspberrypi:~/bundle $ npm install
got@1.0.0 /home/pi/bundle
├── coap@0.20.0
│   ├── ble@1.2.0
│   ├── capitalize@1.0.0
│   ├── coap-packet@0.1.14
│   ├── debug@2.6.3
│   ├── ms@0.7.2
│   ├── fastseries@1.7.2
│   └── reusify@1.0.1
```

- Edit the sample code for resourceid (topic name).

```

/**
 * Send data to QIoT Suite Lite.
 * content of ./res/resourceinfo.json
 * {
 *   ...
 *   "resources": [
 *     {
 *       ...
 *       "resourceid": "temp",
 *       "topic": "qiot/things/admin/abccccc/temp",
 *       ...
 *     }
 *   ]
 * }
 */
setInterval(function() {
  // TODO: you could replace "temp" by any resource id set form QIoT Suite Lite
  connection.publishById("temp", getRandomInt(0, 50));

  // or publish by resource topic
  // TODO: you could replace "qiot/things/admin/edison/temp" by any Topic form QIoT Suite Lite like following
  // connection.publishByTopic("qiot/things/admin/edison/temp", getRandomInt(0, 50));
}, 1000);

```

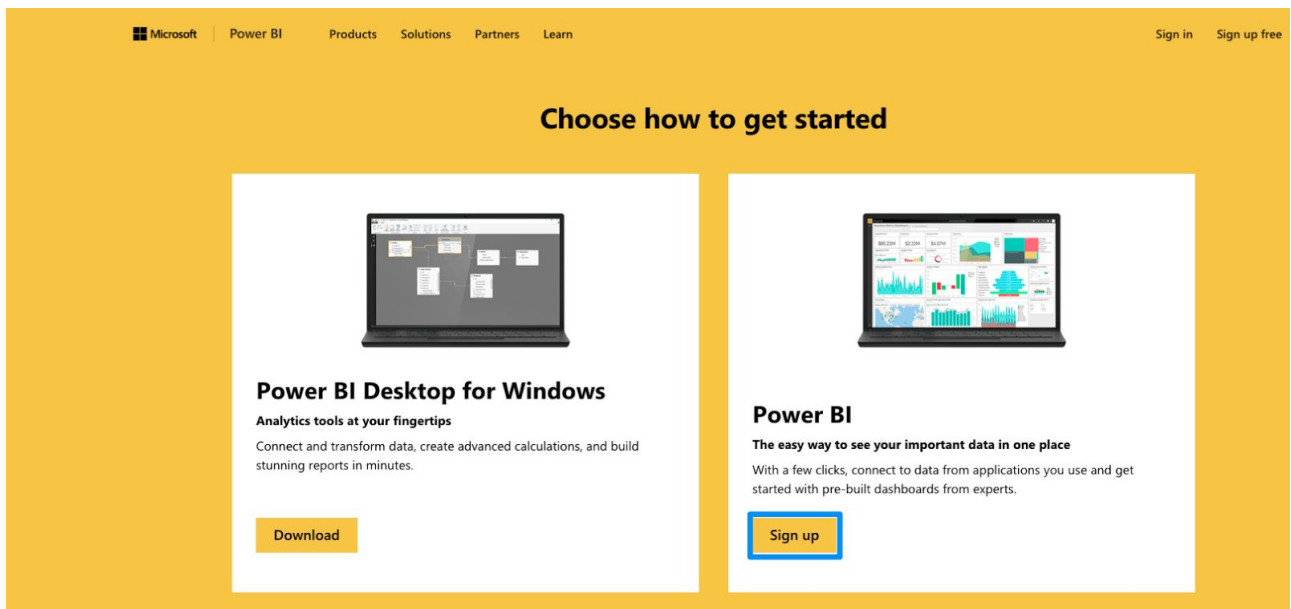
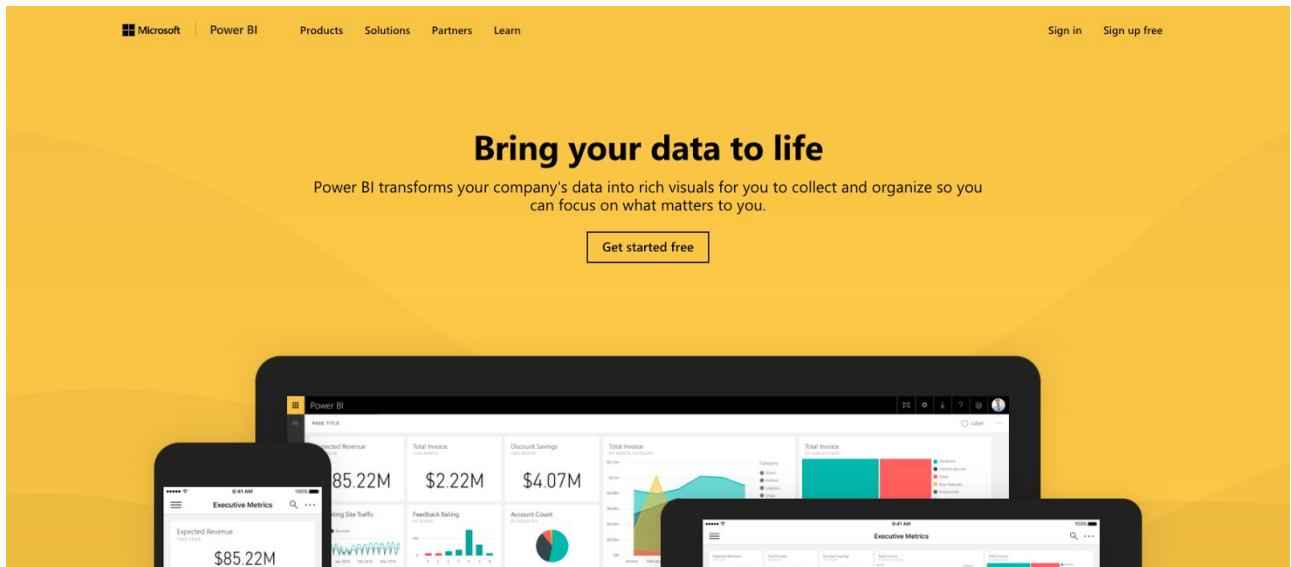
- Run the sample application.

pi@raspberrypi:~ \$ node mqtt.js

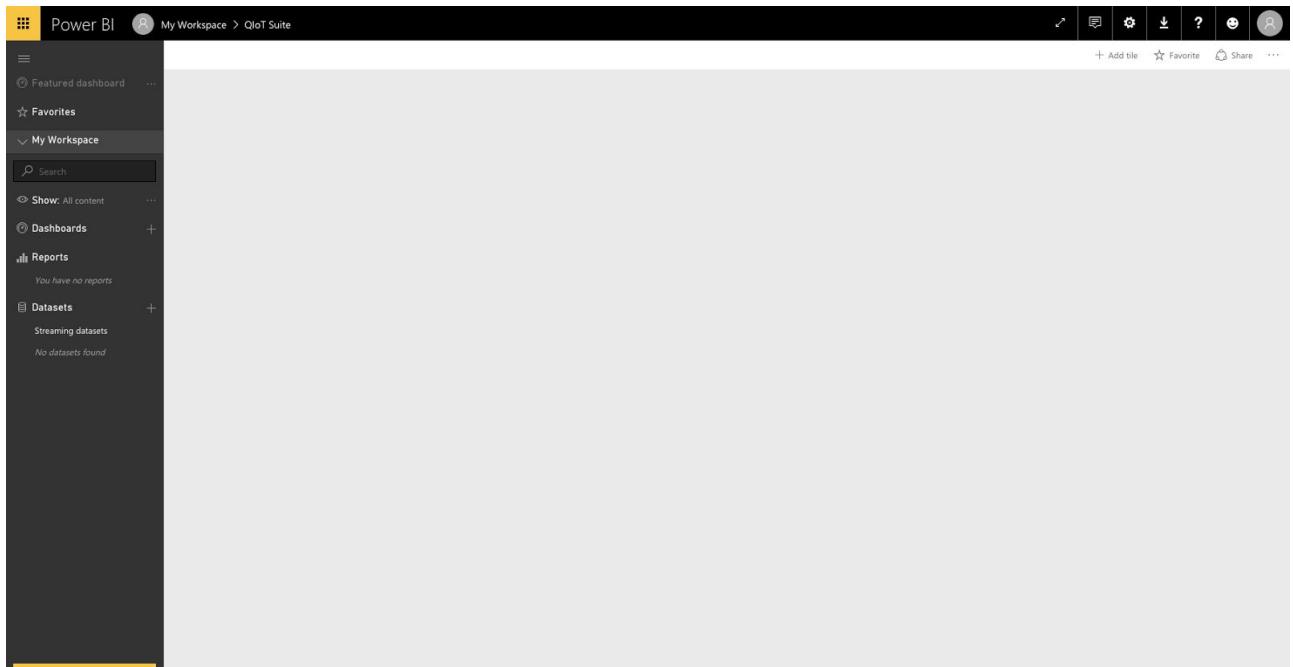
Lesson 4: Integrate Power BI

4.1 Get your first Power BI account

- Go to the official website “<https://powerbi.microsoft.com/en-us/>” to sign up your free account.

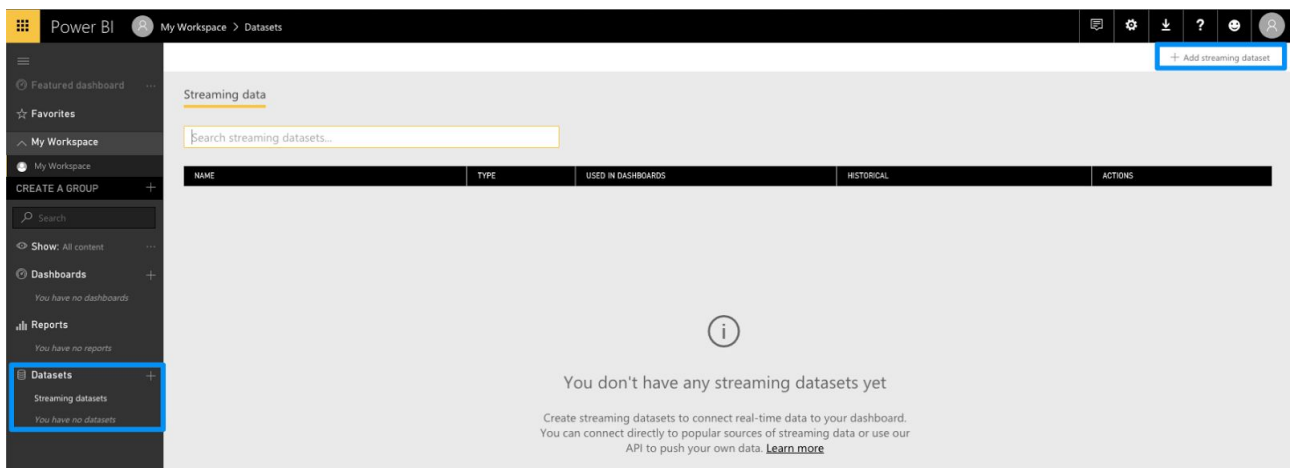


- After a sequence of registration, the page will lead you to below page

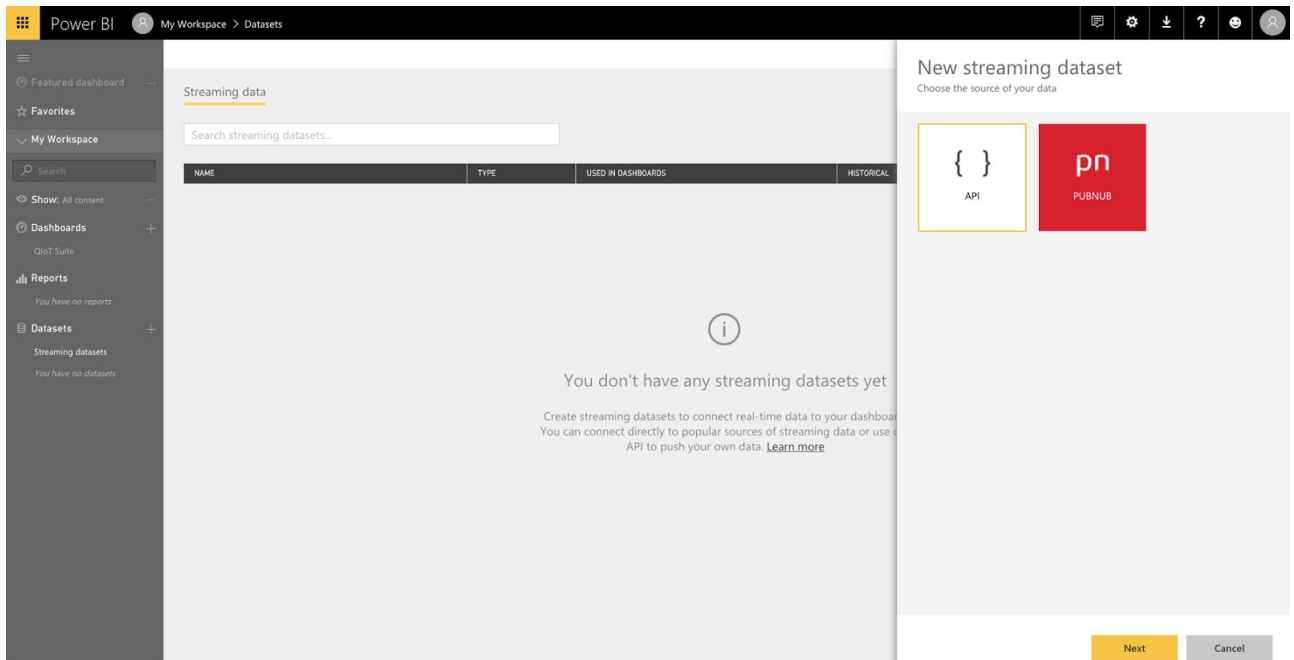


4.2 Setup your streaming dataset API

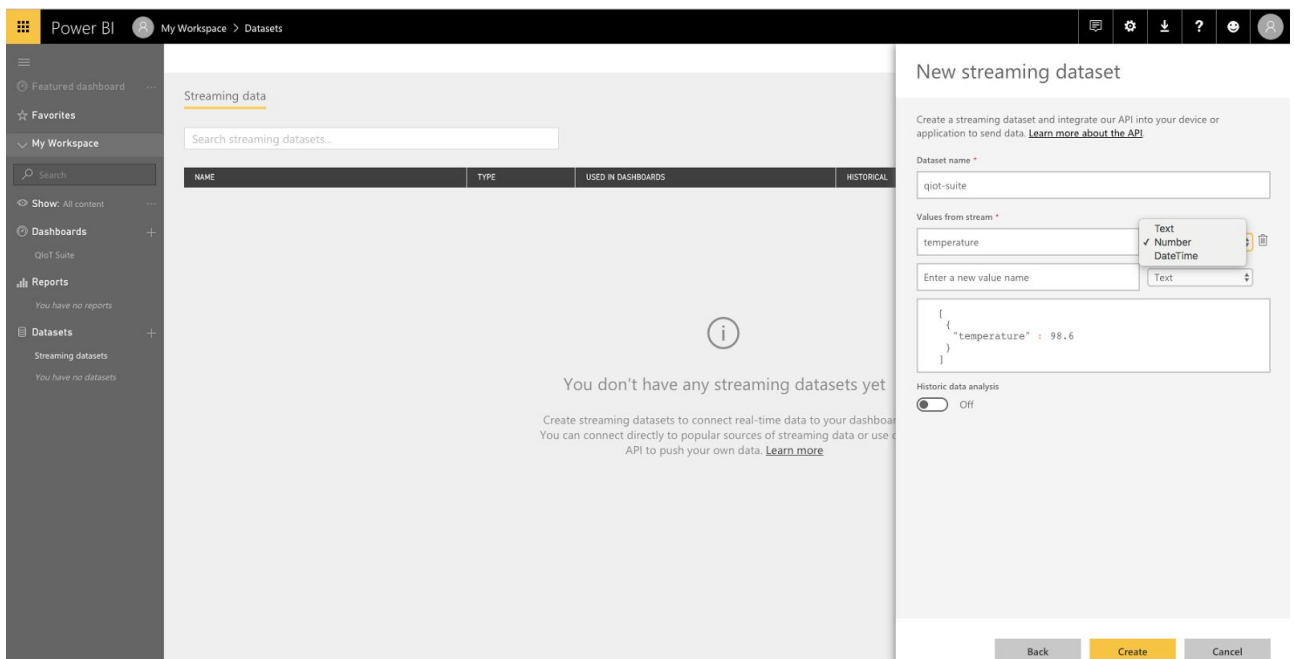
- Go to “**Datasets -> Streaming datasets**” section at left menu, and click the “+ **Add streaming dataset**” button at right-top corner.



- Select “**API**” as your source of iot data, and click the “**Next**” button.



- Define your values from stream, and you will get a result of JSON in textbox. We will use this JSON template later to push data to IoT application. Click the “**Create**” button to finish flow of push API creation.



- Once you successfully create your data stream, you get REST API URL which IoT application can call using POST request to push your live data to streaming data dataset you created.

Power BI My Workspace > Datasets

Streaming data

Search streaming datasets...

NAME	TYPE	USED IN DASHBOARDS	HISTORICAL
qiot-suite	API	QIoT Suite	Enabled

API info on qiot-suite

Use the API endpoint URL and one of the examples shown below to send data to your streaming dataset. For more information, [read our API documentation and integration guide](#).

Push URL

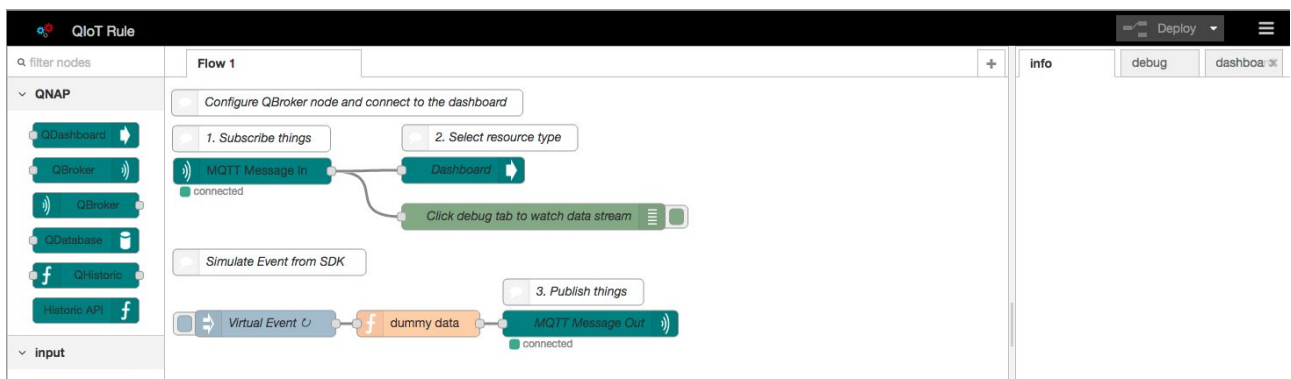
`https://api.powerbi.com/beta/bb3391c7-d712-450b-949c-14d42c1dffa4e/data`

Raw cURL PowerShell

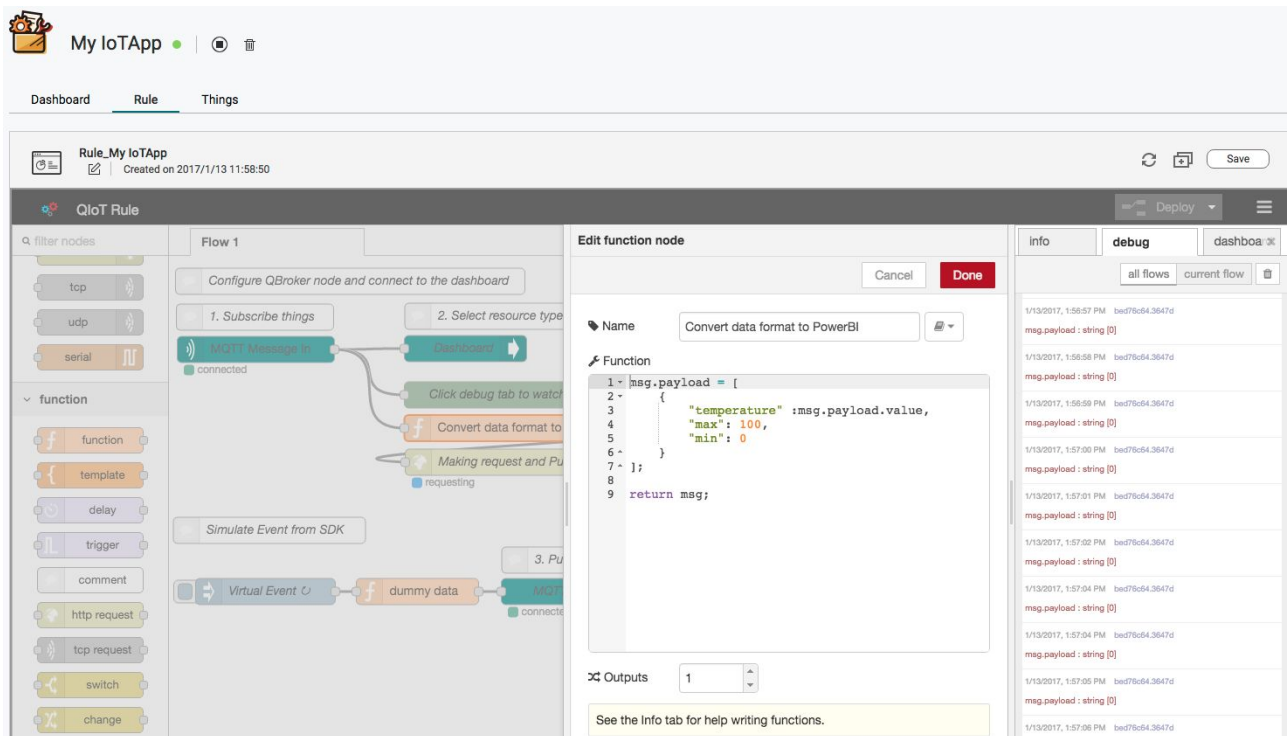
```
{
  "temperature": 98.6,
  "min": 98.6,
  "max": 98.6
}
```

4.3 Configure Node-RED's nodes in IoT application

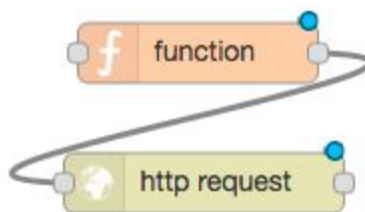
- Create IoT application in QIoT suite. The following one is your first node-red flow, and then you can start to create your own IoT flow. more node-red information can be found in “<https://nodered.org/>”.



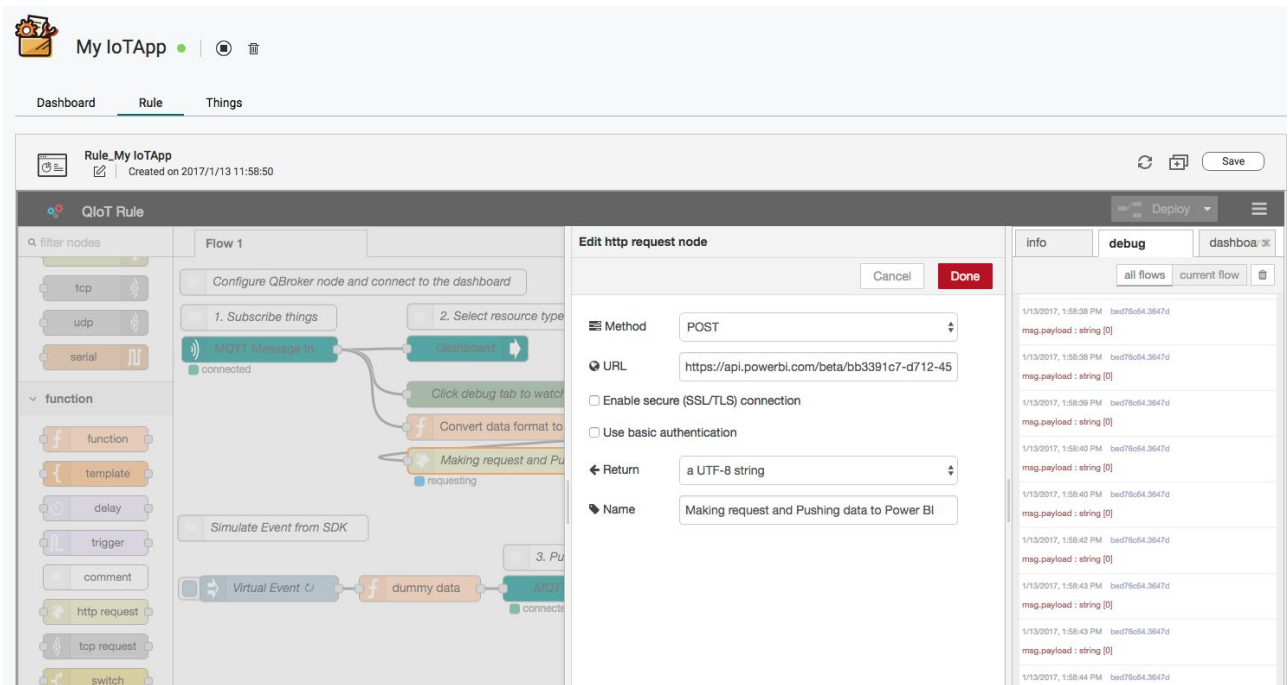
- Before you start to push live data to Power BI. We need a “function” node to convert IoT data to streaming data dataset. Here you can replace msg.payload to your JSON dataset.



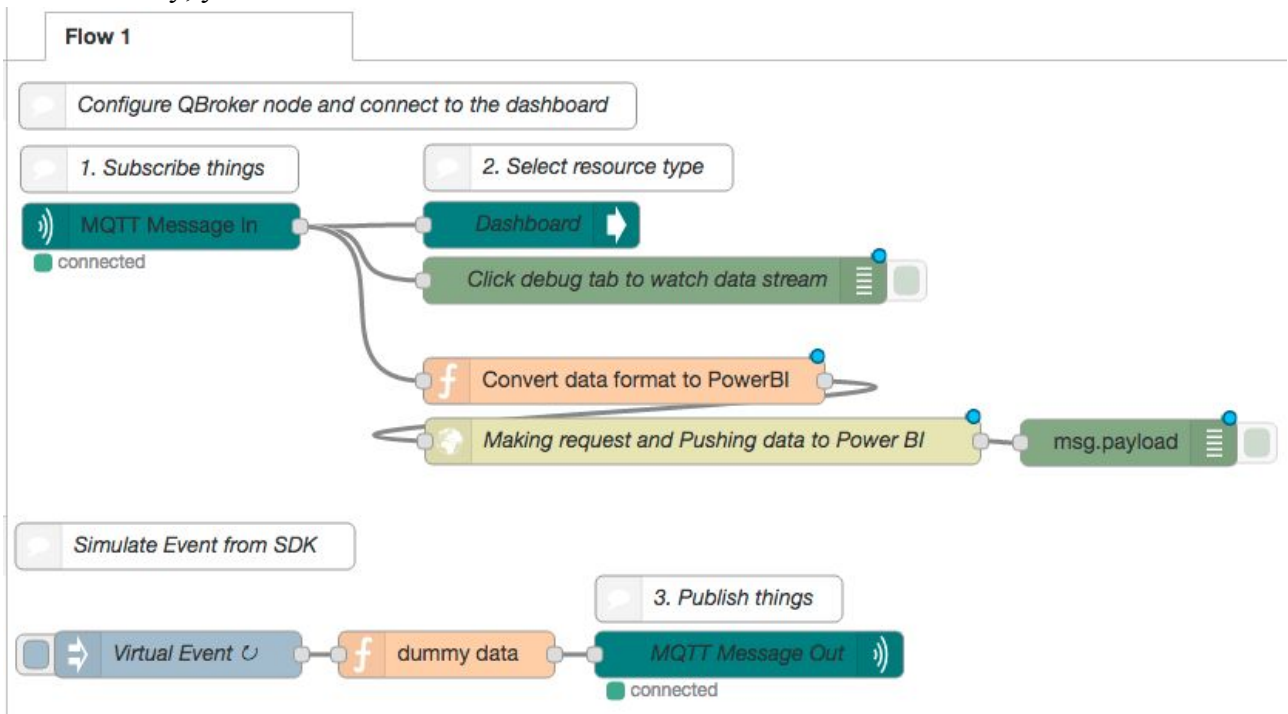
- We need a “http request” node to help us to push live data to Power BI. Just drop and drag “http request” node and connect to tail of “function“ node.



- Copy and paste REST API URL that you got from Power BI console, and set http method to POST. Finally, don't forget to press “**Deploy**” button to save changes.

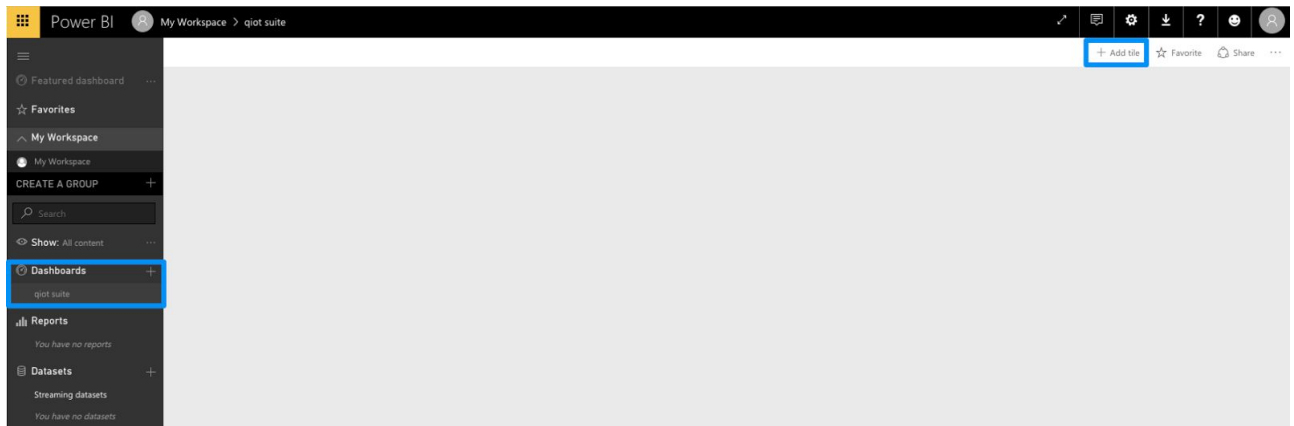


- Finally, your node-red flow will look like below one.

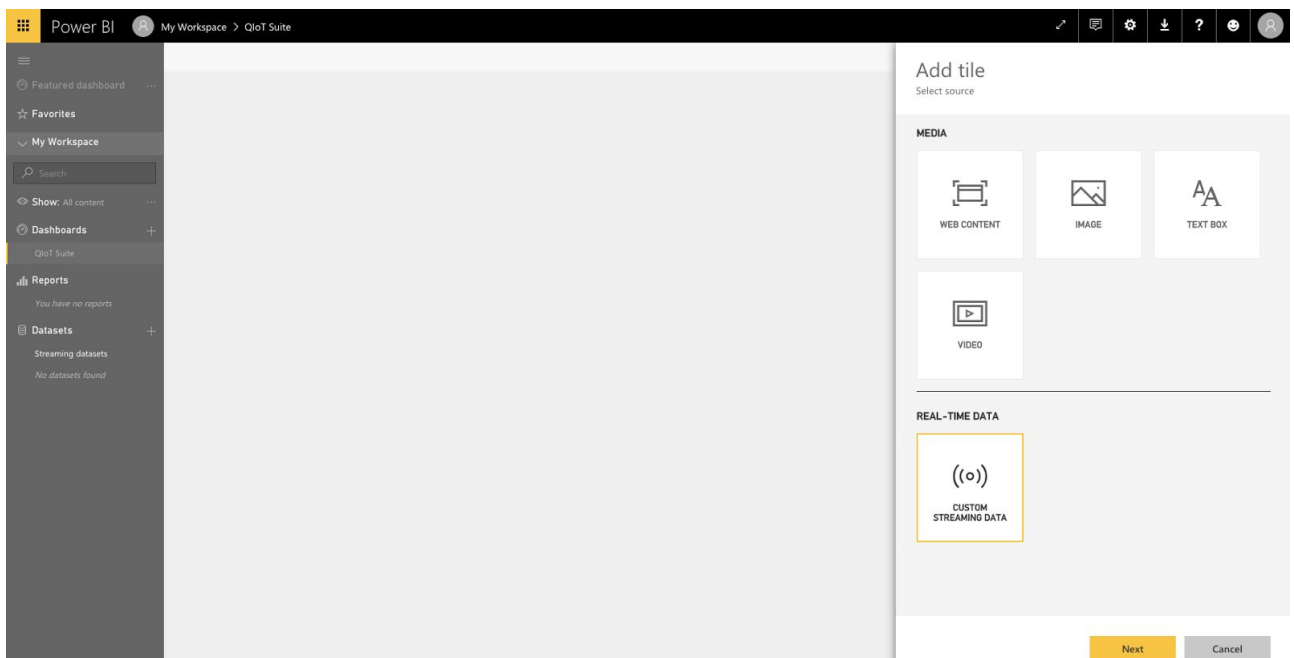


4.4 Add tile to display real-time data

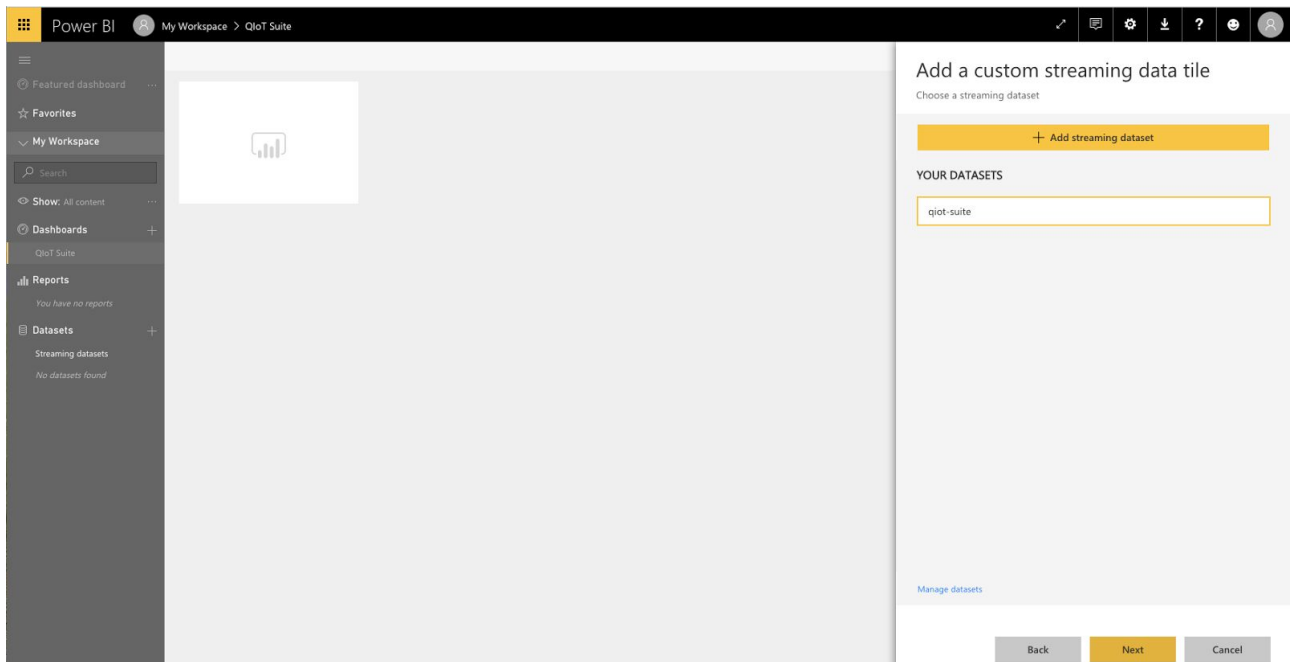
- Go to “**Dashboards +**” to create your first dashboard, then click the “+ **Add tile**” button to configure a widget.



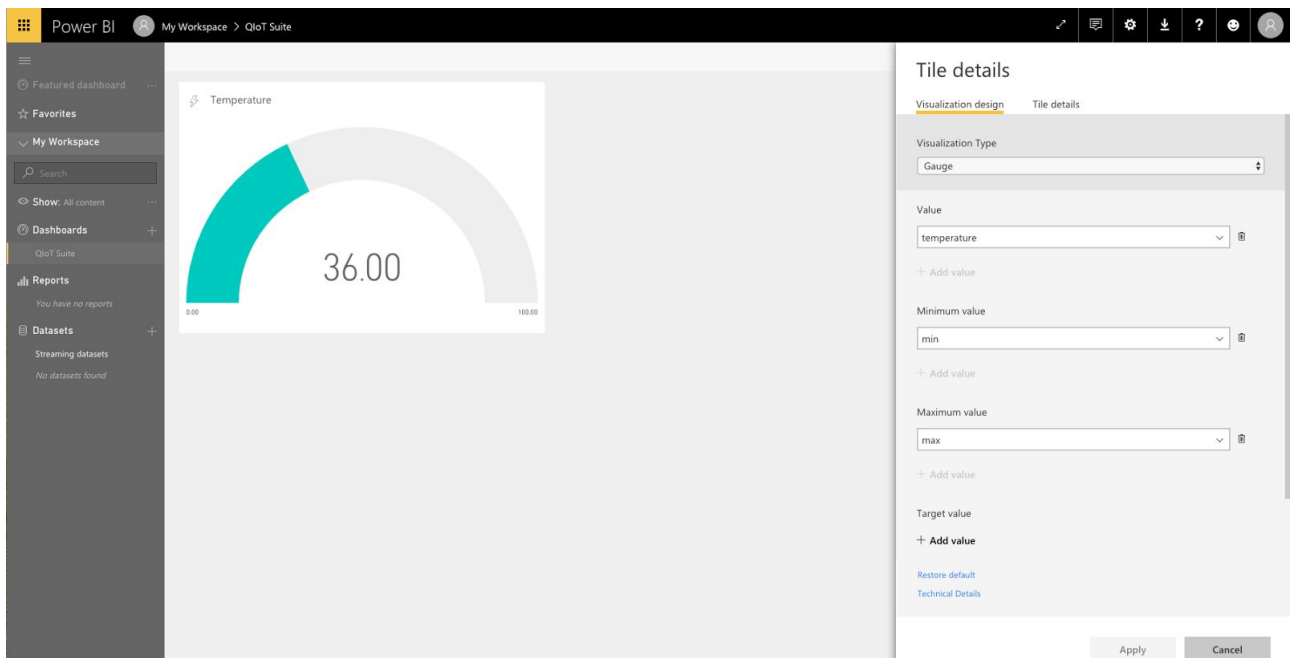
- Select “CUSTOM STREAMING DATA” and then select the “Next” button.



- Select datasets and then click the “Next” button.



- You have a streaming dataset to work with, you can get a real time gauge that looks like as following.



Appendix

QNAP QIoT Startkit Sample Code Introduction

- GitHub: [Sample Code](#)
- Sample Code Structure

```
qnap-qiotsdks/  
  nodejs/                                     # nodejs program language  
    device/  
      intel-edison/                           # intel-edison/raspberrypi...  
        examples/  
          lib/                                # QIoT command Lib  
          res/                                # QIoT resourceinfo.json folder  
          ssl/                                # QIoT certificate files folder.  
          mqtt.js                             # sample code - mqtt/mqtts publish  
          http.js                             # sample code - http post  
          https.js                            # sample code - https post  
          coap.js                             # sample code - coap postt  
          mqtt-subscribe.js                   # sample code - mqtt/mqtts subscribe  
          http-get.js                         # sample code - http get  
          https-get.js                       # sample code - https get  
          coap-observe.js                    # sample code - coap get  
          package.json                       # npm packages document  
python/  
  device/  
    intel-edison/  
      examples/  
        lib/  
        res/  
        ssl/  
        mqtt.js  
        http.js  
        ...
```

- content of resourceinfo.json

protocol	resourceinfo.json content
mqtts	<pre>{ "host": ["172.17.28.28" # nas ip], "myqnapcloudHost": "Not Available", # myqnapcloudHost</pre>

	<pre> "port": 8883, # mqttts port "clientId": "adfa_1491561635", # thing Id "username": "00477f86-425b-49de-8590-xx", # username "password": "r:2825dedfb012969e1dfb6adb8", # password "resources": [# resource des { "resourcename": "adf", # resource name "resourceid": "dfadf", # resource id "resourcetyponame": "Temperature", # resource type "datatype": "Float", # data type "unit": "°C", # data unit "description": "adfa", # resource des "topic": "qiot/things/admin/adfa/dfadf" # topic name }], "caCert": "/v1/media/ca-crt.pem", # certificate file "clientCert": "/v1/media/xx-04-07_10-40-35/xx_certificate.pem", "privateCert": "/v1/media/xx-04-07_10-40-35/xx_privatekey.pem" } </pre>
https	<pre> { "accesstoken": "r:2825dedfb012969e1dfb6adb8", # password "myqnapcloudHost": "Not Available", "clientId": "adfa_1491562164", "host": ["172.17.28.28"], "requesterid": "00477f86-425b-49de-8590-xx", # username "port": 3443, # https port ... } </pre>
CoAP	<pre> { "myqnapcloudHost": "Not Available", "clientId": "adfa_1491562176", "host": ["172.17.28.28"], "r": "00477f86-425b-49de-8590-1282c65b4348", # username "t": "r:2825dedfb012969e1dfb6adb80a419df", # password "port": 5683, # coap port ... } </pre>