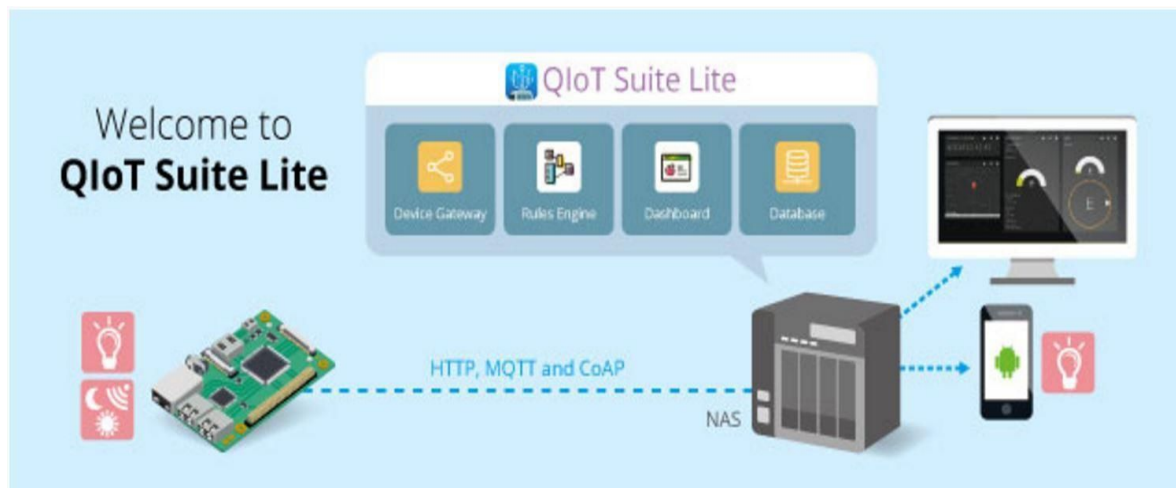


Get started with Raspberry Pi (Nodejs)

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

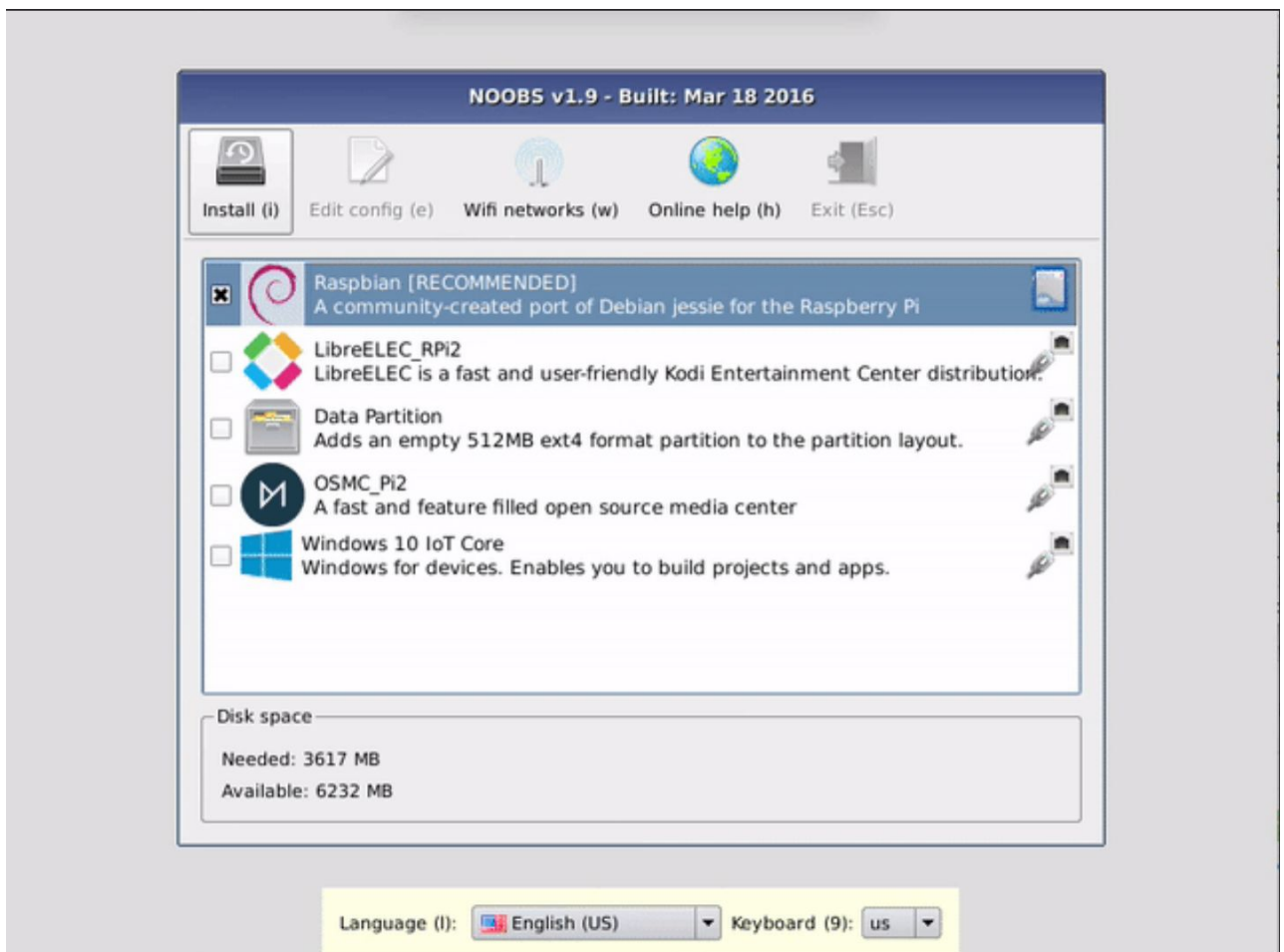
In this lesson, you configure your Raspberry Pi device with an operating system, set up your development environment, and deploy an application to Raspberry Pi.

1.1 What will you need

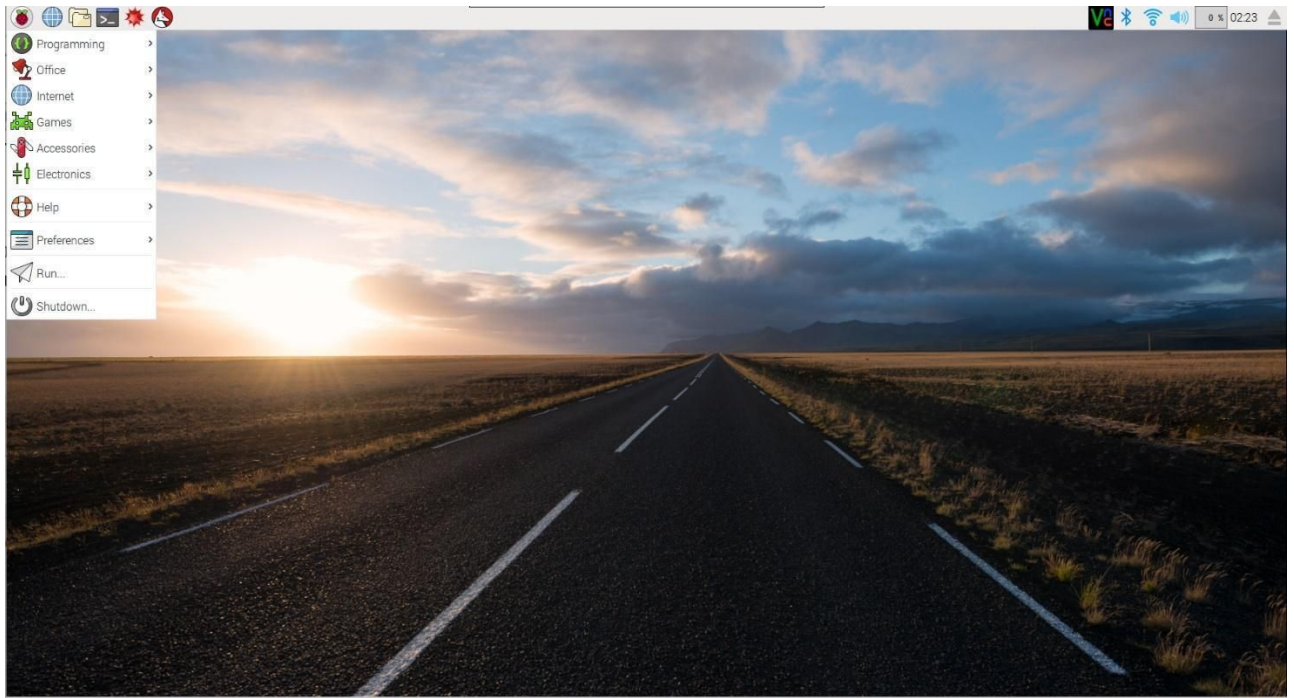
- A Raspberry Pi
- SD card : an 8GB class 4 SD card is recommended, ideally preinstalled with NOOBS.
- A HDMI television or monitor : You will need to connect your Raspberry Pi to a display, which means you'll need an HDMI-enabled screen of some kind.
- Keyboard and mouse : Any standard USB and mouse 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

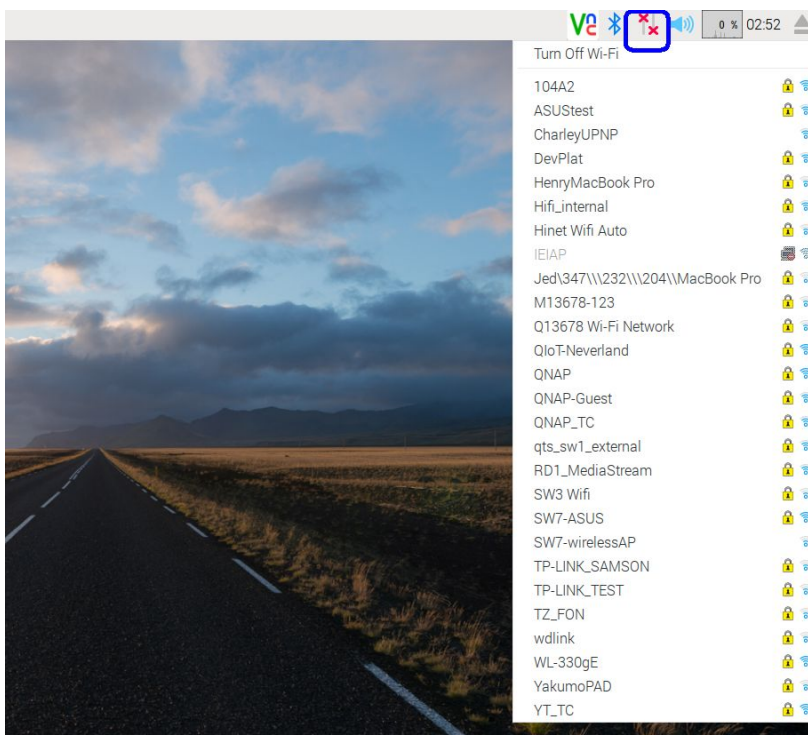
- Put the SD card containing extracted NOOBS files into Raspberry Pi
- Connect keyboard, mouse, and HDMI monitor to the Raspberry Pi, then plug in your power adapter. The Raspberry Pi does not have a power switch, so once you connect the power adapter, it'll turn on all by itself.
- A simple menu will 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.



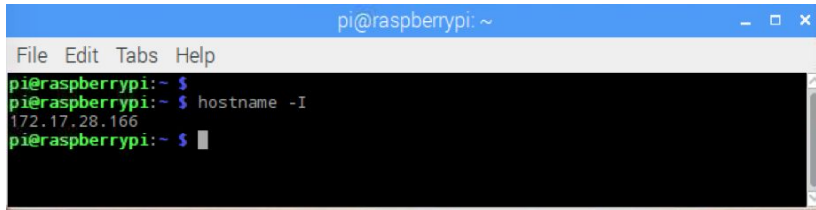
- Now, let NOOBS run the installation process, which may take 10 or 20 minutes. When it's complete, it will restart and send you straight into the Raspbian desktop, where you can configure everything else.



- 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/>

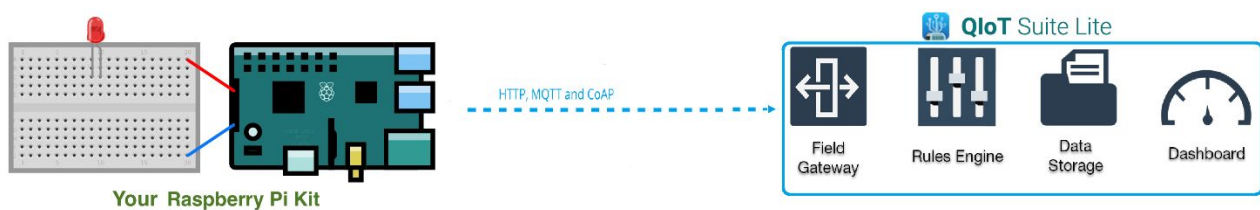
1.3 Install Nodejs and required libraries

- Press Ctrl + Alt + t to open Terminal. Install node.js

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

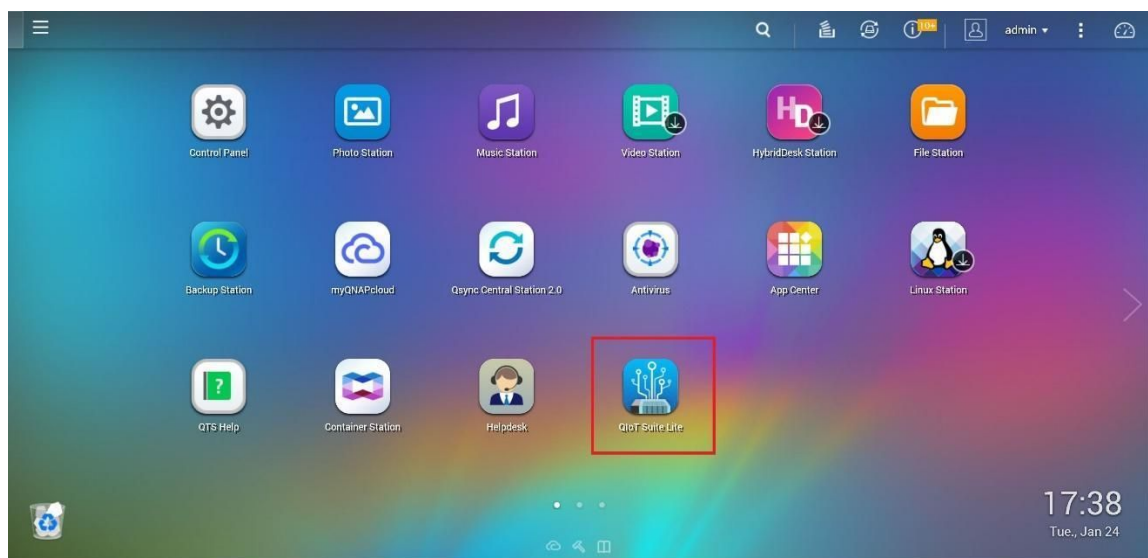
Lesson 2: Create your device in QIoT Suite Lite

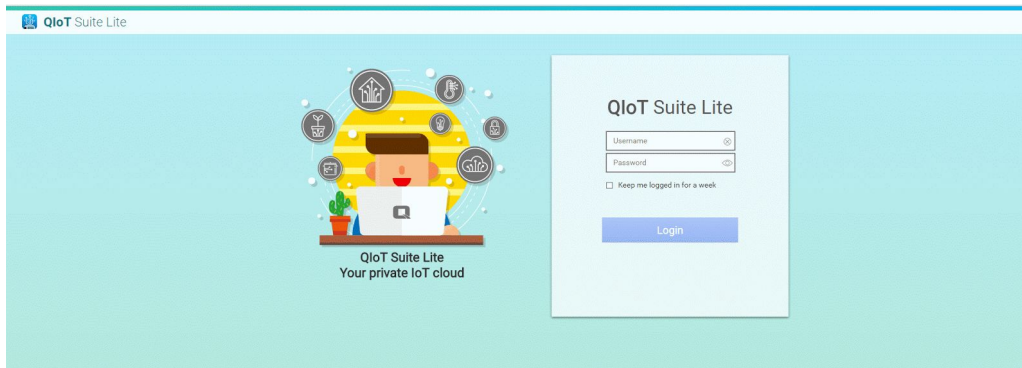
In this lesson, you provision your QNAP QIoT Suite Lite software, and create your first device in QIoT Suite Lite.



2.1 Install QIoT Suite Lite

- Go to QNAP App Center and download QIoT Suite Lite application.
- Launch and log in QIoT Suite Lite software. Use Nas admin and password to login.





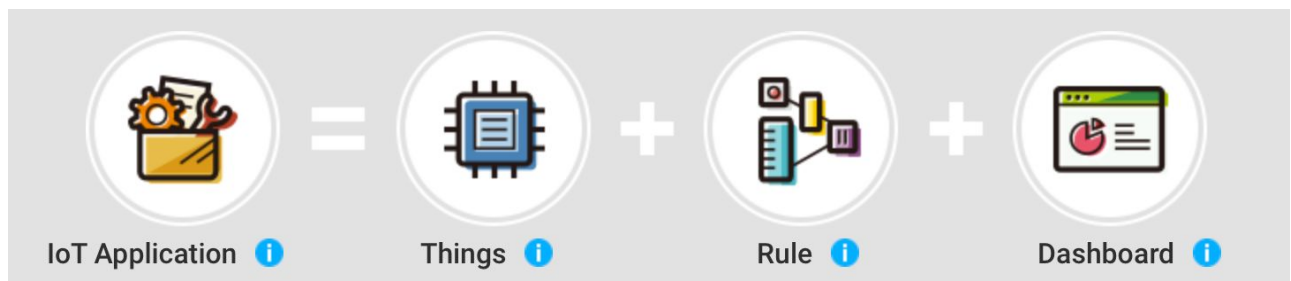
2.2 Create a new IoT application

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.

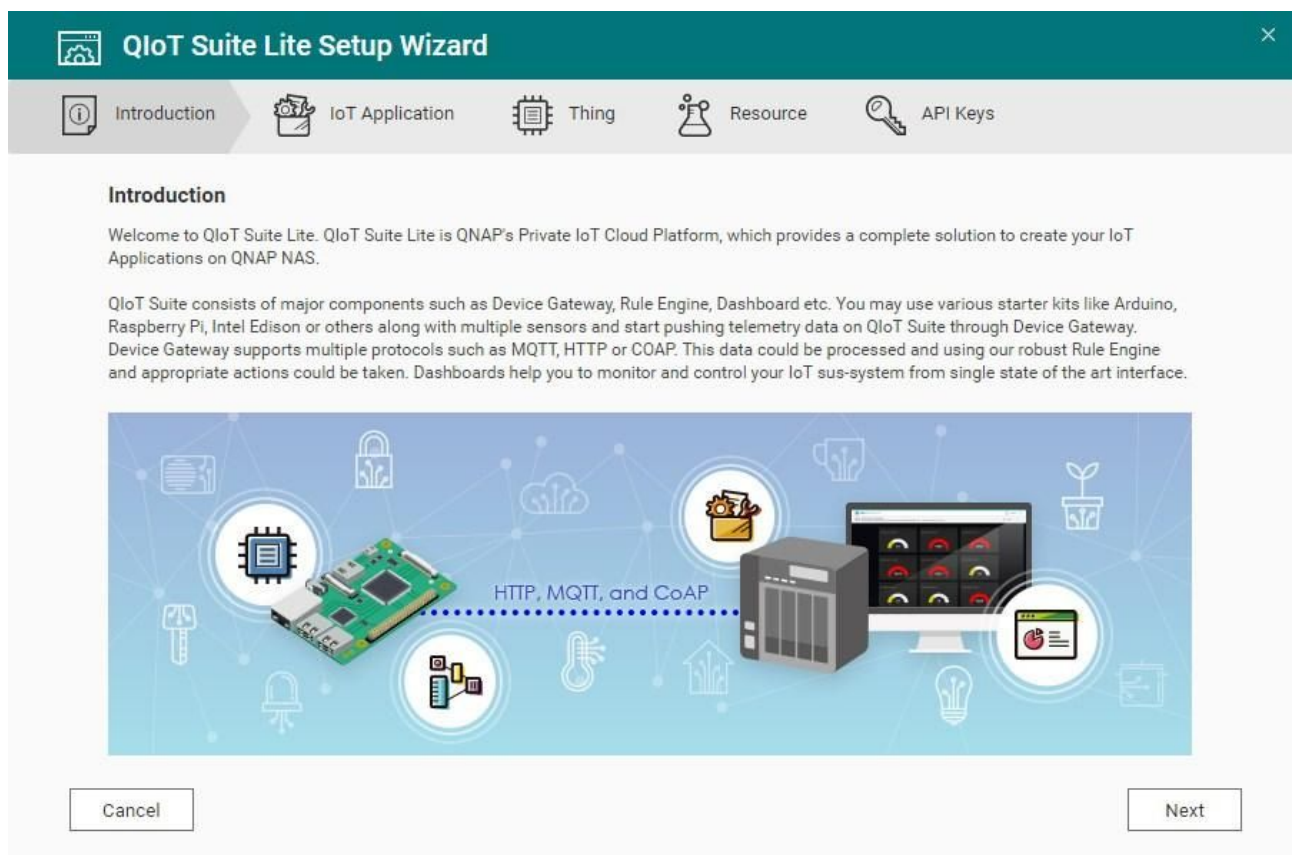
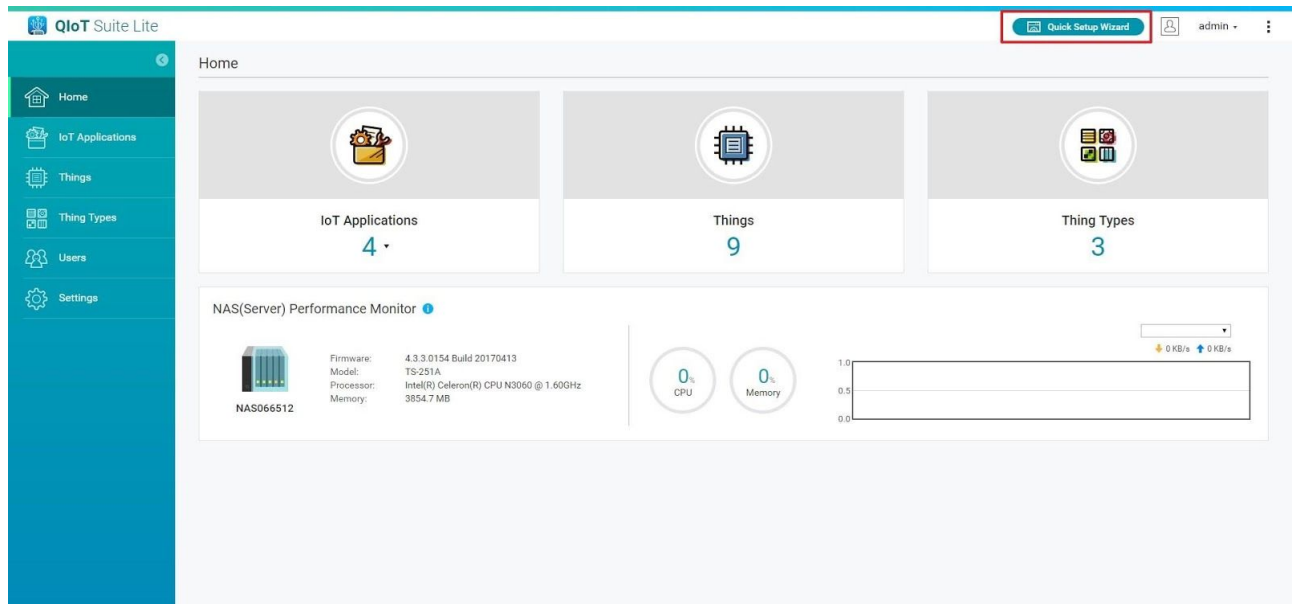
The Rule makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices based on business rules you define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel.

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 “Quick Setup Wizard” to start quick setup wizard . After you read the QIoT Suite Lite introduction in pop window, click Next button.







- Create a New IoT application
 - Enter IoT Application name, e.g., app_1.
 - Rule name and Dashboard name will be generated automatically based on the name of IoT Application you fill in.
 - Click “Next” to complete create a new IoT application

QIoT Suite Lite Setup Wizard

Introduction 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 Name *: ✓

Description:

Rule Name:

Description:

Note: Inputs with * are required field

Cancel Back Next



- Click “+Add” to add this application’s device.

QIoT Suite Lite Setup Wizard

Introduction 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.

Cancel Next

- After click “+Add ”,In “Add Thing” pop window:
 - Set device’s name (e.g., raspberrypi).
 - Qlot Suite Lite now support Arduino Yun,Raspberry Pi,and Edison ,so you can select “QIoT Supported” thing category and select “Thing Type” is “Raspberry Pi”
 - (optional) add attribute to device’s detail information (e.g., its serial number, manufacturer, and more).
- Click “Add” to add the device to complete create a device.

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: ☒ QIoT Supported ☐ Custom

Thing Type

Thing Type:

Attributes for raspberrypi (Optional):

Note: Inputs with * are required field

- Please provide IP address, user name, and password of your device, then click “Connect” .After waiting test connection between your device and Qlot Suite Lite success, you could click “Next” to next step.

QIoT Suite Lite Setup Wizard
×

Introduction
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.

+ Add
Delete

	Thing Name	IP	User Name	Password	Action
<input type="checkbox"/>	raspberrypi	172.17.30.148	pi	*****	Connect ✓ ✕

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

Cancel
Next

- In “Resources” step,Click “Add Resource”.

QIoT Suite Lite Setup Wizard
×

Introduction
IoT Application
Thing
Resource
API 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:

raspberrypi(0) ✕

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

⏪
⏩
Page 1 / 1
⏪
⏩

0 - 0 of 0

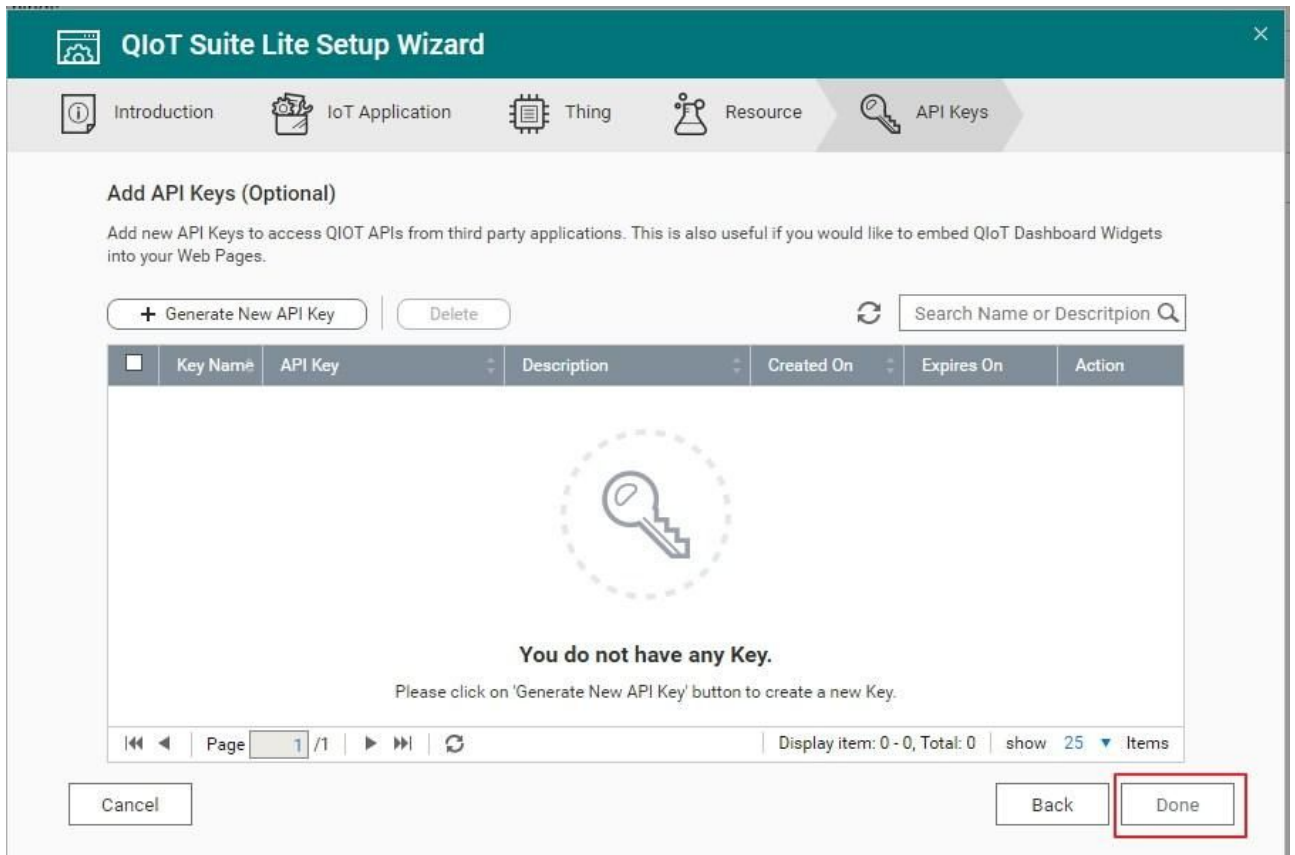
Cancel

Back
Next

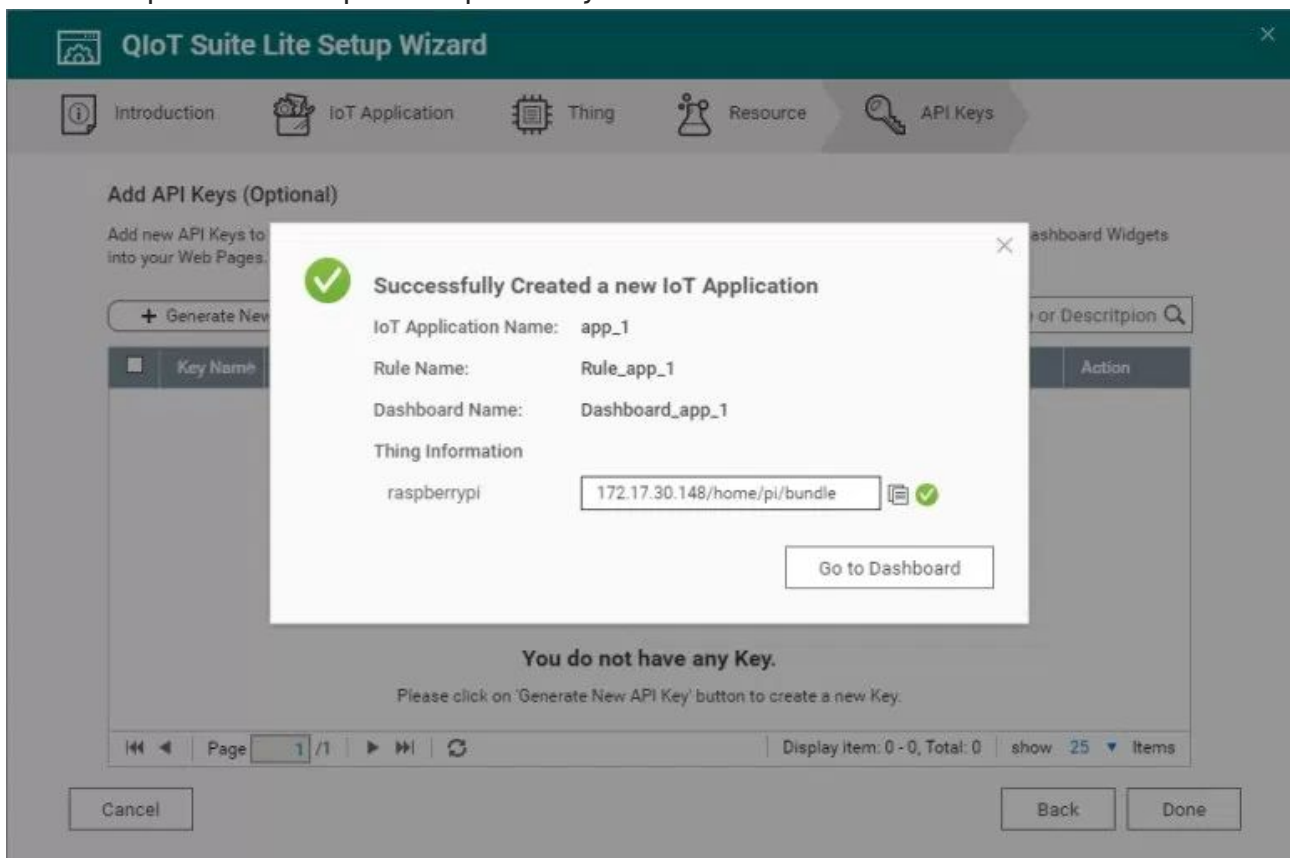
- After click “Add Resource”,In “Add Thing Resource” pop window:
 - Set resource’s name (e.g., temp).
 - Set resource’s id.Resource id will be used to create a topic in the QIoT broker.This ID should be unique for the device and no duplicates should be allowed for the same device
 - And set another optional attribute.

The screenshot shows the "Add Thing Resource" dialog box. The "Resource Name" field contains "temp" and the "Resource ID" field contains "temp". Both fields are highlighted with red rectangles and have green checkmarks to their right. The "Resource Type" is set to "Temperature", "Data Type" is set to "Float", and "Unit" is set to "°C". There are also checkboxes for "Set Range" and "Set Default Value", each with sub-fields for minimum, maximum, and default values. A note at the bottom states "Note: Inputs with * are required field". At the bottom right are "Add" and "Cancel" buttons.

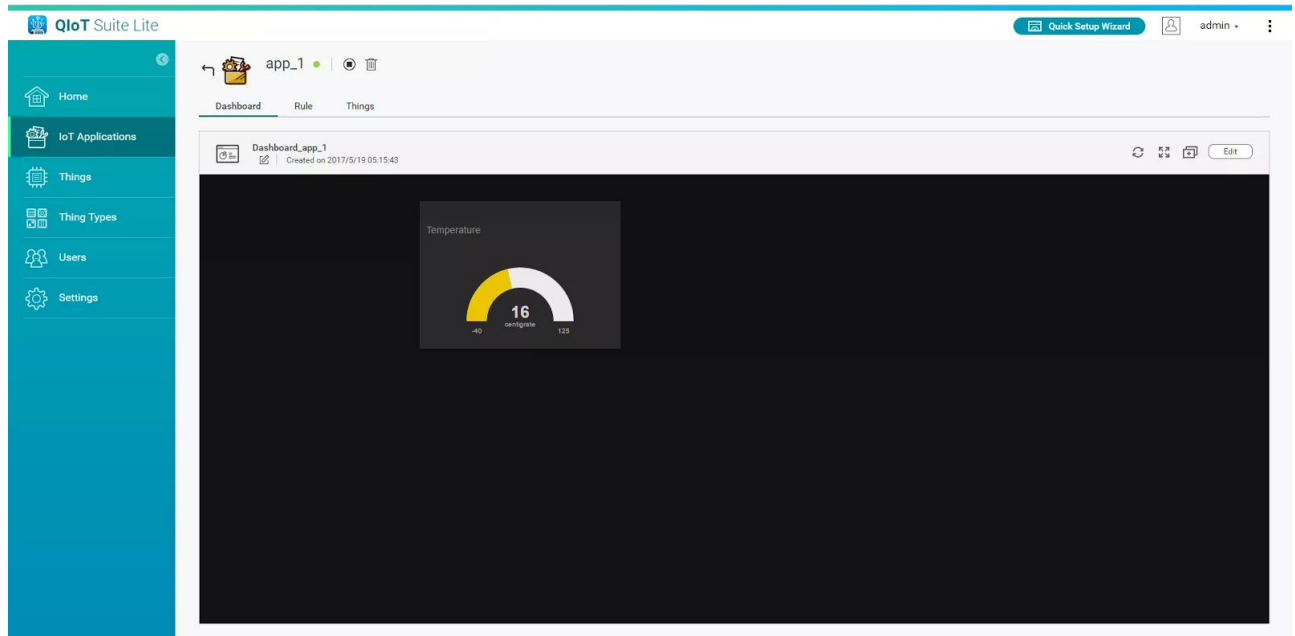
- Click Next after you add all the resources (sensors) on your device.
- In “API Keys” step :
 - If you would like to embed QIoT dashboard widgets into your web pages or access QIoT APIs from third party applications.You could click “generate new API Key” to create API key or click “done” start deploy sample code to your device.



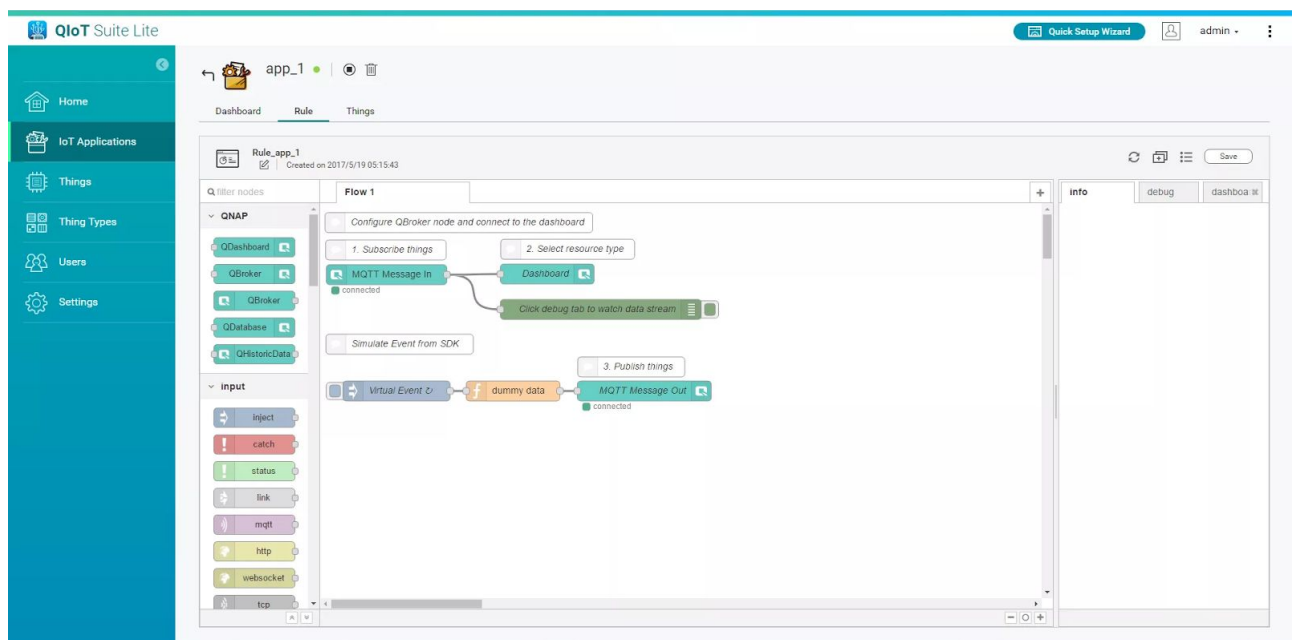
- For a while, sample codes and related files (certificate, resource information) have been upload to the specified path on your device.



- Your IoT application already created successfully .You could click”Go to Dashboard” to your application page.
- Select “Dashboard” tab , you could see a sample dashboard is created.



- Select “Rule” tab, you could 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 RaspberryPi to QIoT Suite Lite.



3.1 Run Sample Code

- Open Terminal application (e.g., [PuTTY](#)) on your PC. Connect to your device by SSH and enter the folder where put sample code (e.g., /home/root/bundle).

```
172.17.30.148 - PuTTY
login as: pi
pi@172.17.30.148's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 18 22:30:03 2017 from 172.17.30.144
pi@berry:~$
```

- Install sample code dependency, enter command as following
`pi@berry:~$ cd /home/{{user}}/bundle`
`pi@berry:~/bundle$ npm install`

- Run sample code in device will publish message to topic “temp” by MQTTS as following picture. Topic is define from resource id that you setted.

```
/**
 * Send data to QIoT Suite Lite.
 * content of ./res/resourceinfo.json
 * {
 *   ...
 *   "resources": [
 *     {
 *       ...
 *       "resourceid": "temp",
 *       "topic": "qiot/things/admin/abccccc/temp",
 *       ...
 *     }
 *   ]
 * }
 */
setInterval(function() {
  // TODO0: you could replace "temp" by any resource id set form QIoT Suite Lite
  connection.publishById("temp", getRandomInt(0, 50));

  // or publish by resource topic
  // TODO0: 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@berry:~\$ node mqtt.js

- device will send message to topic “temp” or that you defined ,as below image.

```
pi@berry:~/bundle$ node mqtt.js
privatekey full path = 2376772017-05-16_10-31-44_privatekey.pem
ready to conneciton:
send message to [mqtt(s)://172.17.30.174:8883], topic_Pub = qiot/things/admin/raspberrypi/temp, value = {"value":32}
send message to [mqtt(s)://172.17.30.174:8883], topic_Pub = qiot/things/admin/raspberrypi/temp, value = {"value":20}
send message to [mqtt(s)://172.17.30.174:8883], topic_Pub = qiot/things/admin/raspberrypi/temp, value = {"value":3}
send message to [mqtt(s)://172.17.30.174:8883], topic_Pub = qiot/things/admin/raspberrypi/temp, value = {"value":10}
send message to [mqtt(s)://172.17.30.174:8883], topic_Pub = qiot/things/admin/raspberrypi/temp, value = {"value":28}
```

3.2 Another protocol

- Click “Connection a device” button
- You can choose another protocol you would like to use

The screenshot shows a 'Thing Information' dialog box with a close button (X) in the top right corner. It has three tabs: 'Details', 'Connect a Device' (which is selected), and 'Policy'. Below the tabs, a message states: 'Following Process will guide you to make your device start communicating to QIoT Suite. Choose the protocol you would like to use:'. There are five radio button options: MQTT, MQTTS, HTTP, HTTPS, and COAP. The 'MQTTS' option is selected and highlighted with a red rectangular box. Below this, a section titled 'Generate Device Certificate:' contains two paragraphs of text explaining the need for security credentials and how to generate them. A large teal button labeled 'Generate a Device Certificate' is centered below the text. At the bottom left of the dialog is a 'Learn More' link, and at the bottom right is a 'Close' button.

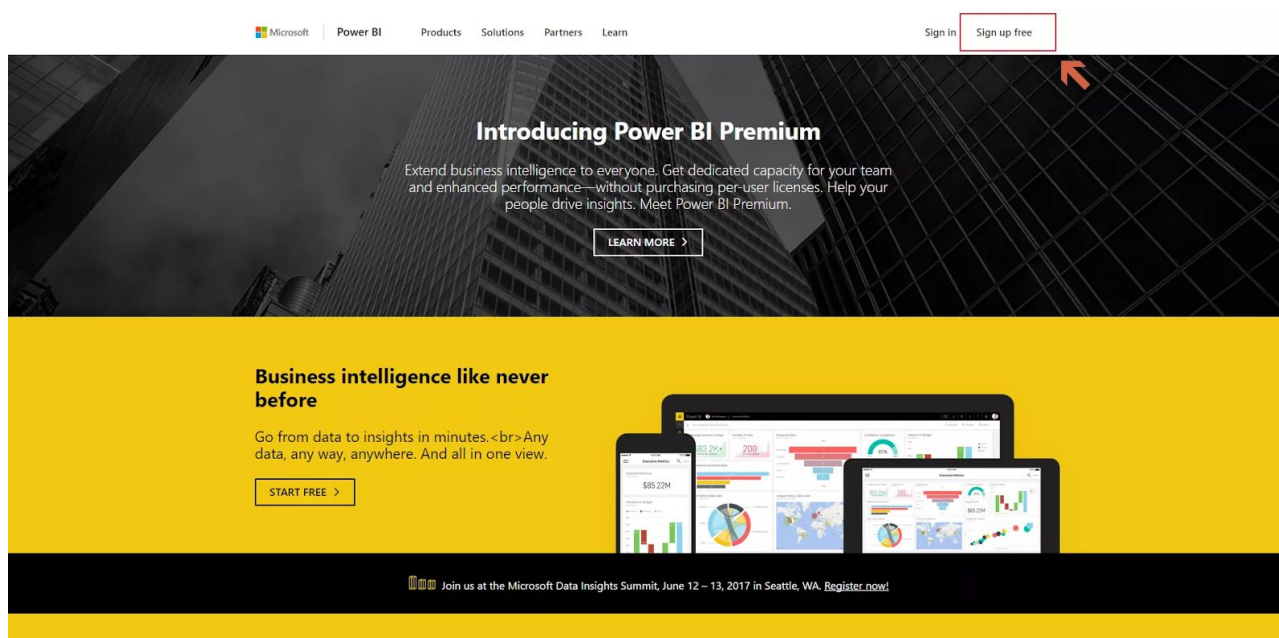
- SSH to your device , and input command as following.

```
pi@berry:~$# cd /home/{{user}}/bundle
// mqtt(dont' need certificate,just put JSON file to "res" folder):
pi@berry:~/bundle$ node mqtt.js
// http
pi@berry:~/bundle$ node http.js
// https
pi@berry:~/bundle$ node https.js
// coap
pi@berry:~/bundle$ node coap.js
```

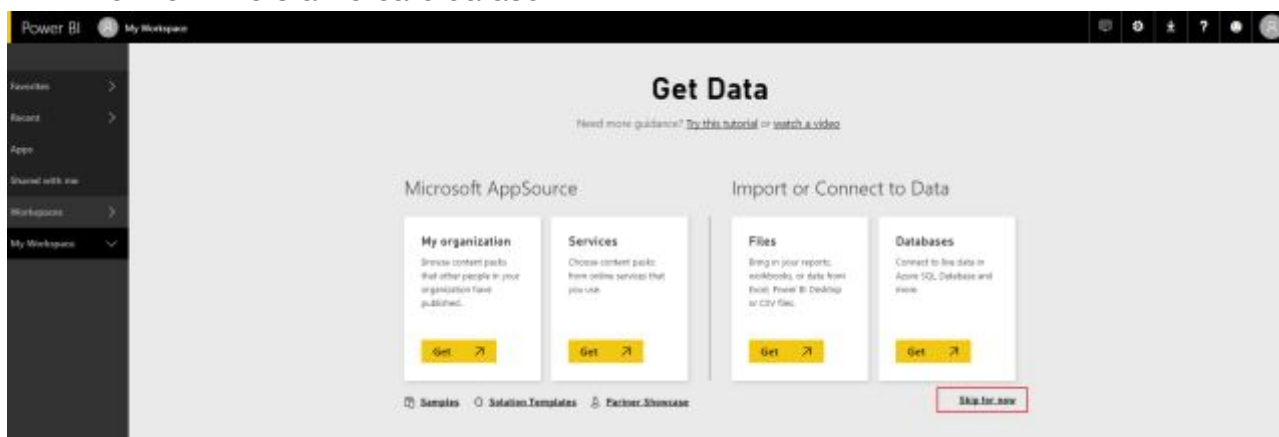
Lesson 4: Integrate Power BI

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.



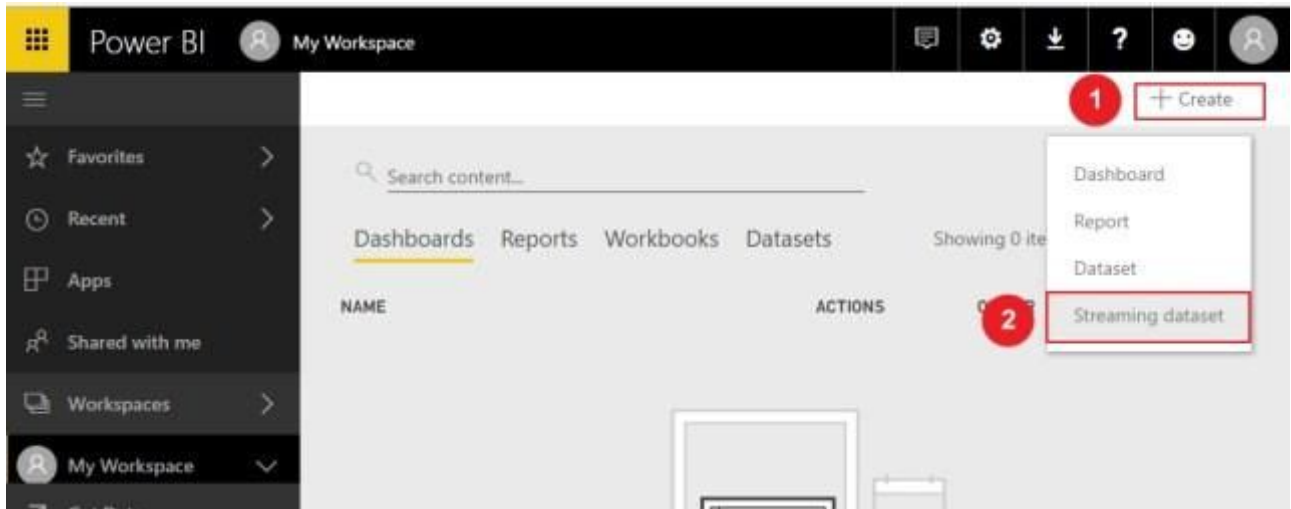
- After a sequence of registration, the page will lead you to below page, you can press “My workspace”, and “skip for now” button appear. You could click “skip for now” to start create dataset.



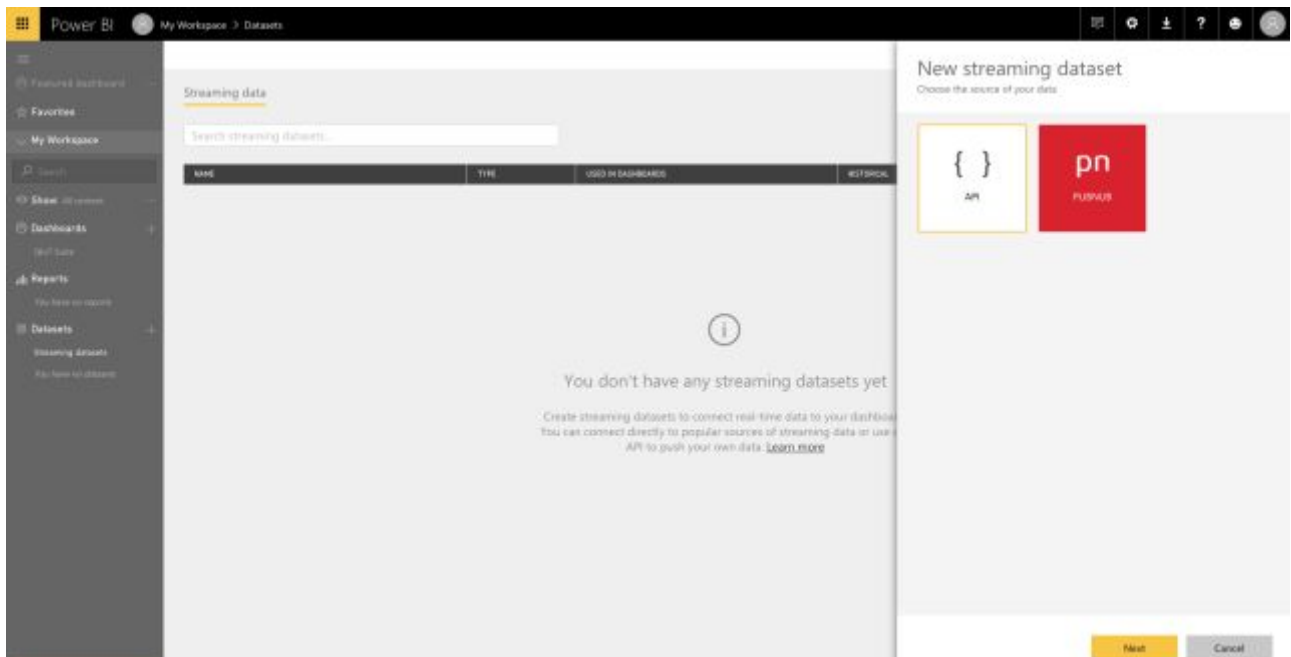
4.2 Setup your streaming dataset API

- Create “Datasets”
 - Click “Create” in screen upper right corner

- And then click “Streaming dataset”



- Select “API”, and click “Next”.



- Define your values from stream(e.g., temp,max,min), and you will get a result of JSON in textbox. Qlot Suite’s application will post this data format to Power BI. Click the “Create” button to finish create streaming dataset.

Edit streaming dataset

Create a streaming dataset and integrate our API into your device or application to send data. [Learn more about the API.](#)

Dataset name *

Values from stream *

Number ▼



Number ▼



Number ▼



Text ▼

```
[
  {
    "temp" : 98.6,
    "max" : 98.6,
    "mix" : 98.6
  }
]
```

Historic data analysis



On

Back

Done

Cancel

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

✓ Streaming dataset created

The schema for temp2 is created.

Push URL

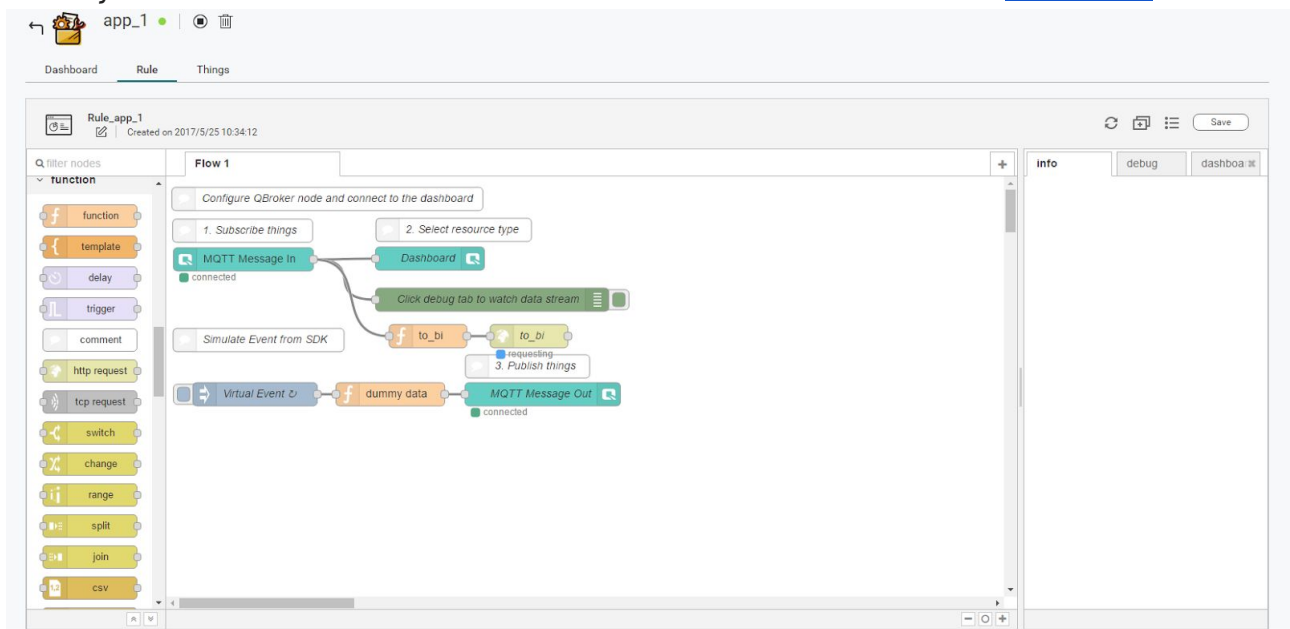
```
https://api.powerbi.com/beta/bb3391c7-d712-450b-949c-14d42c1dff4e/data
```

Raw cURL PowerShell

```
[
  {
    "temp" : "AAAAA555555",
    "max" : "AAAAA555555",
    "min" : "AAAAA555555"
  }
]
```

4.3 Configure Node-RED's nodes in IoT application

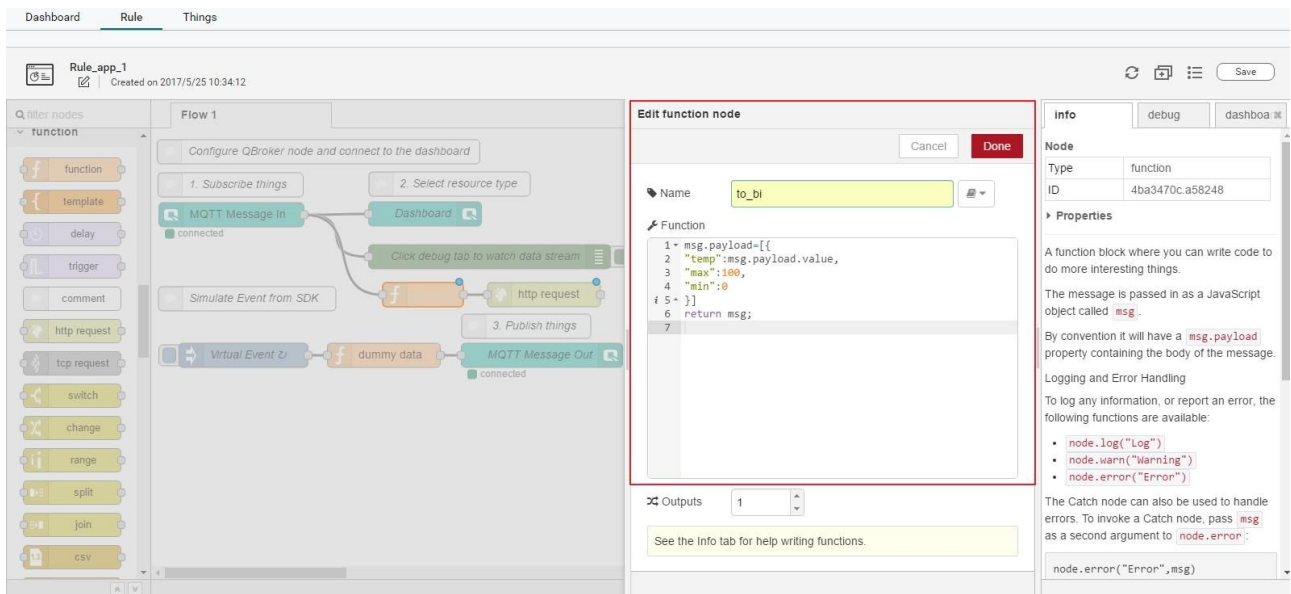
- Create IoT application in QIoT suite.
- The following one is your first node-red flow, and then you can start to create your own IoT flow. more node-red information can be found in "[Node-Red](#)".



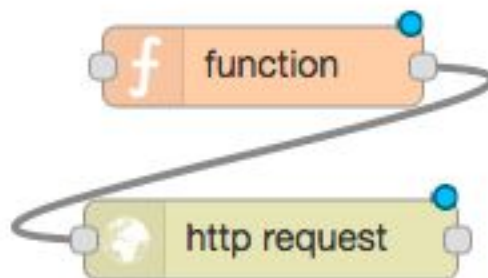
- Before you start to push live data to Power BI.
- We need a "function" node to convert IoT data to streaming data dataset. Here you can replace `msg.payload` to your JSON dataset.

- Double click function node, and type Function code as following:

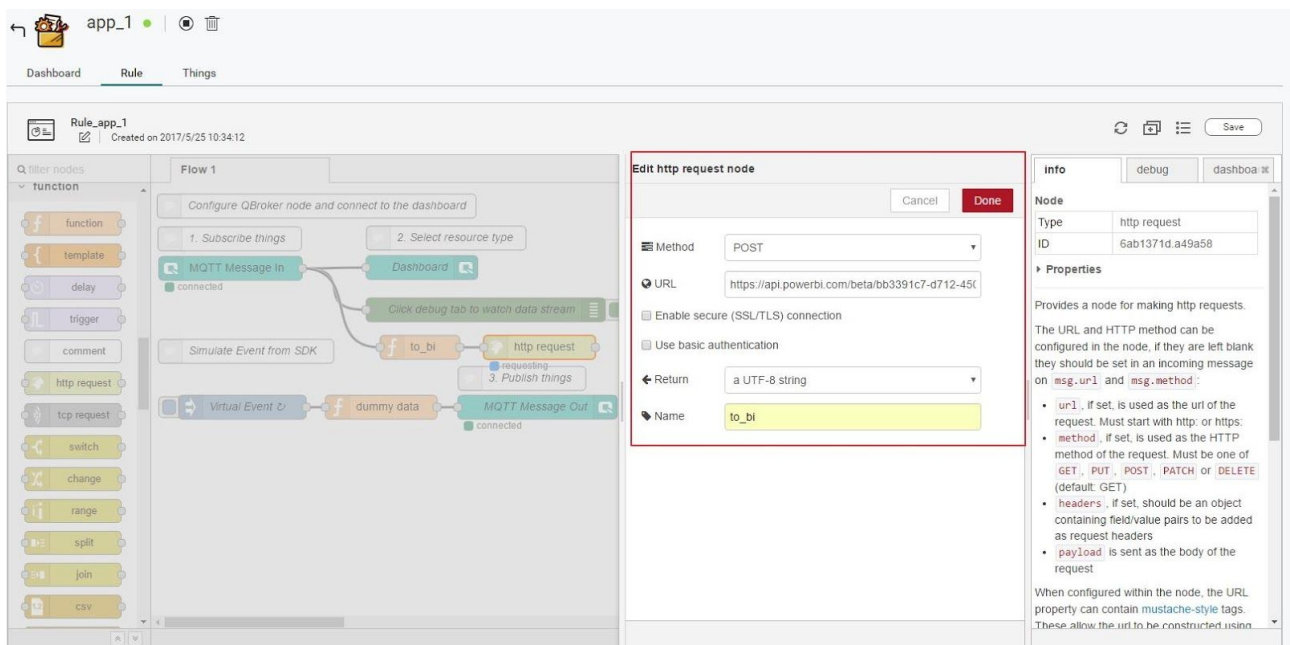
```
msg.payload=[{
  "temp":msg.payload.value,
  "max":100,
  "min":0
}]
return msg;
```



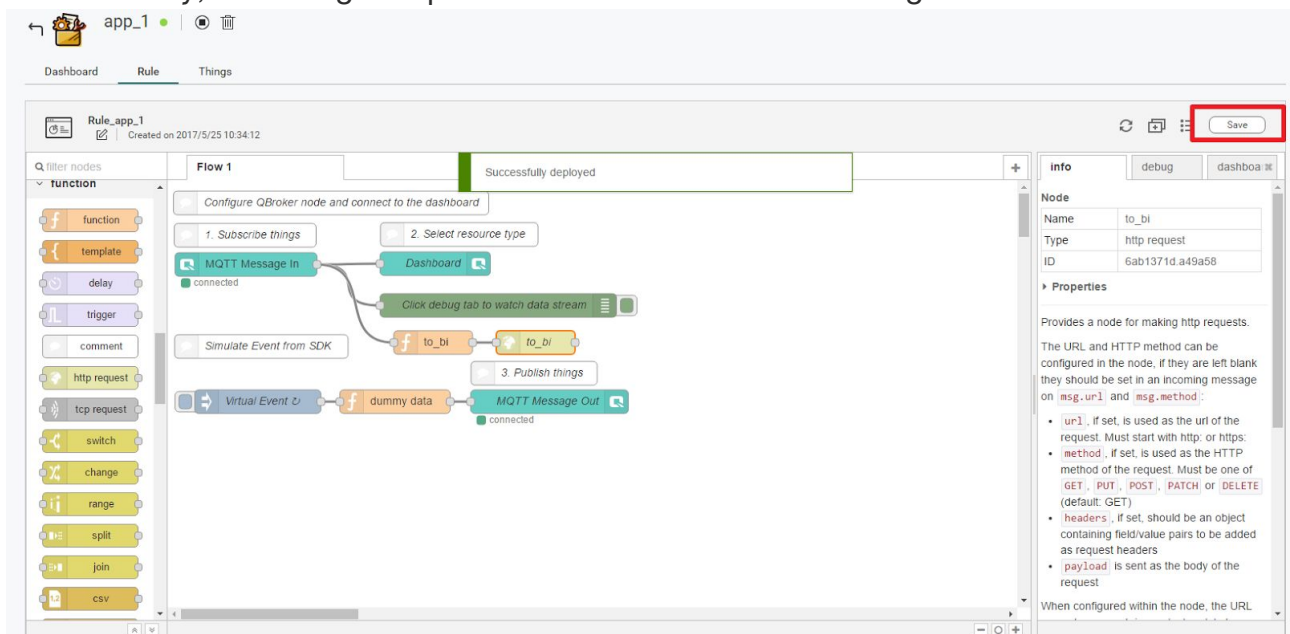
- We need a “http request” node to help us to push live data to Power BI. Just drop and drag “http request” node and connect to tail of “function” node.



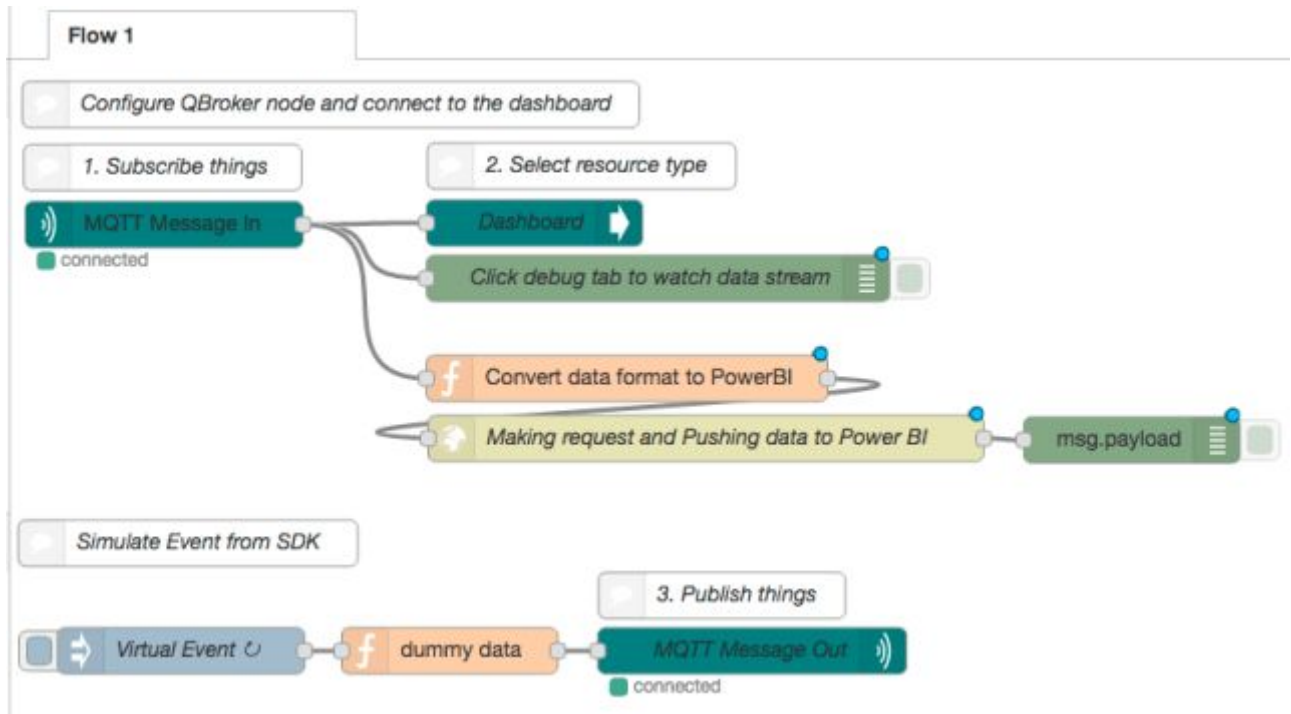
- Double click http request node, copy and paste REST API URL that you got from Power BI console, and set http method to POST.



- Finally, don't forget to press "Save" button to save changes.

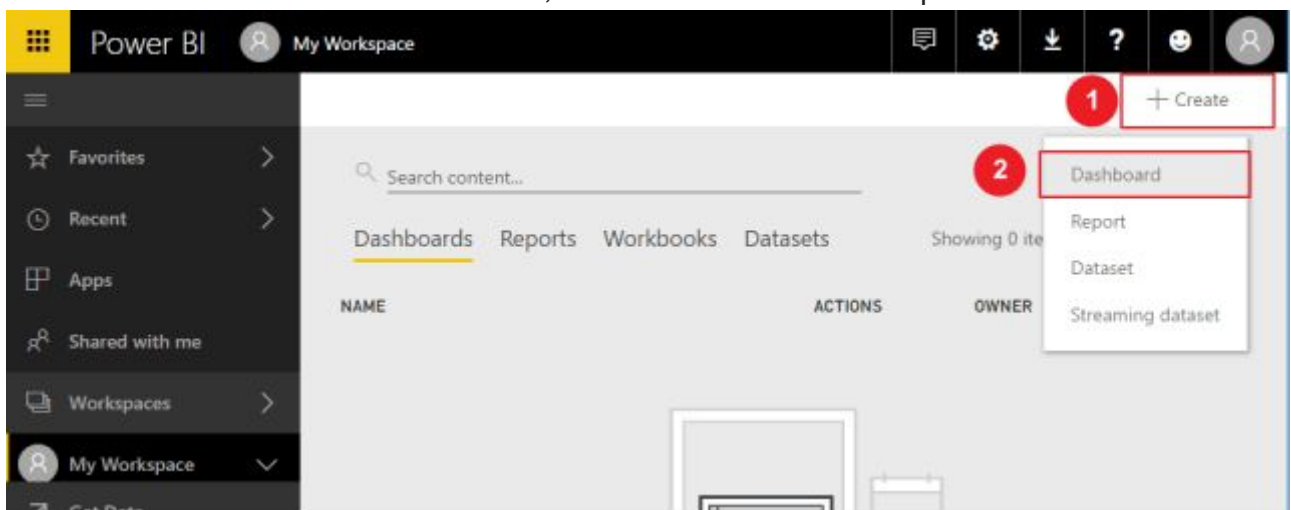


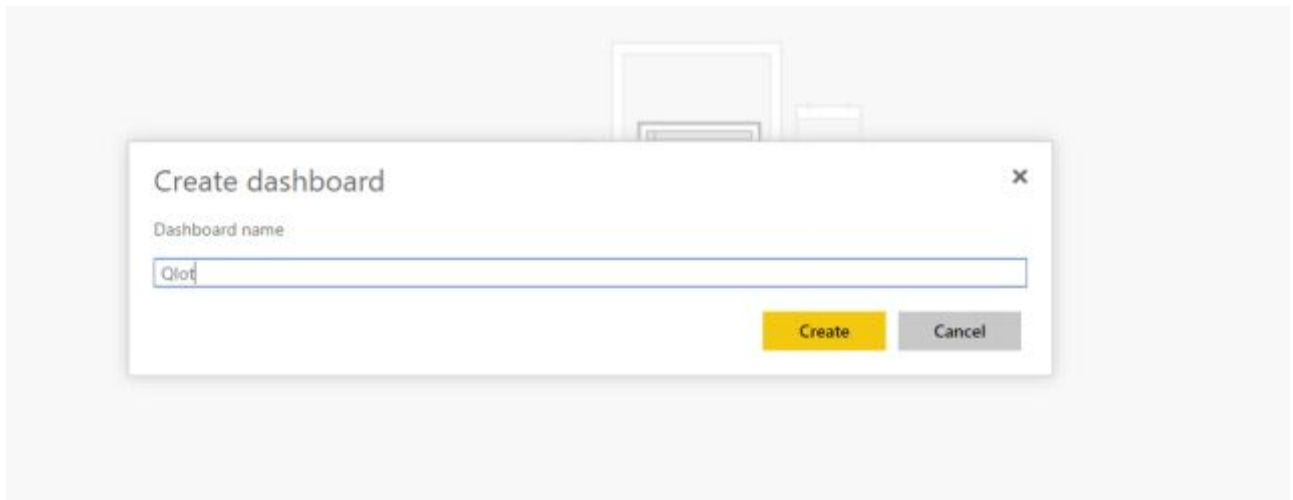
- Finally, your node-red flow will look like below one.



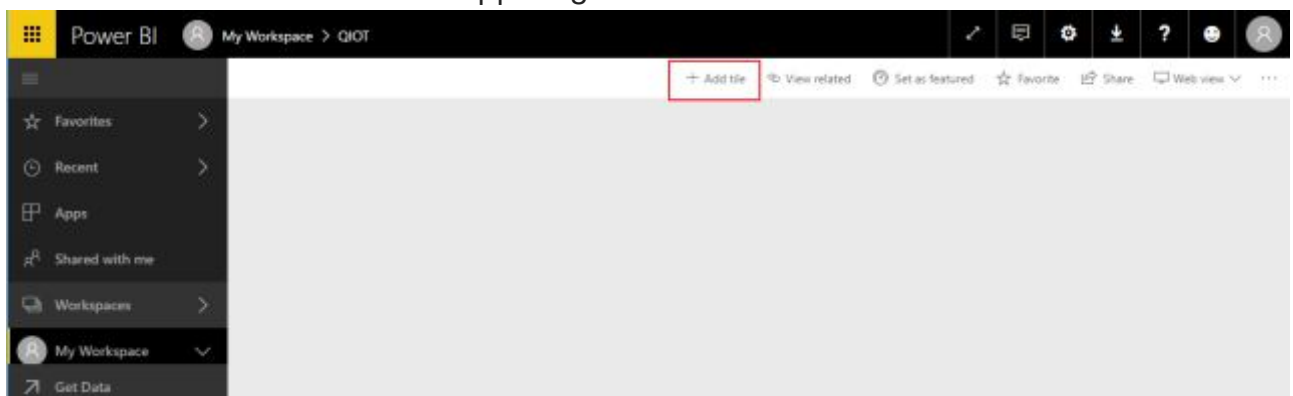
4.4 Add tile to display real-time data

- Create “Dashboard”
 - Click “Create” in screen upper right corner
 - And then click “Dashboard”
 - Enter dashboard’s name, and click “Create” to complete create dashboard.

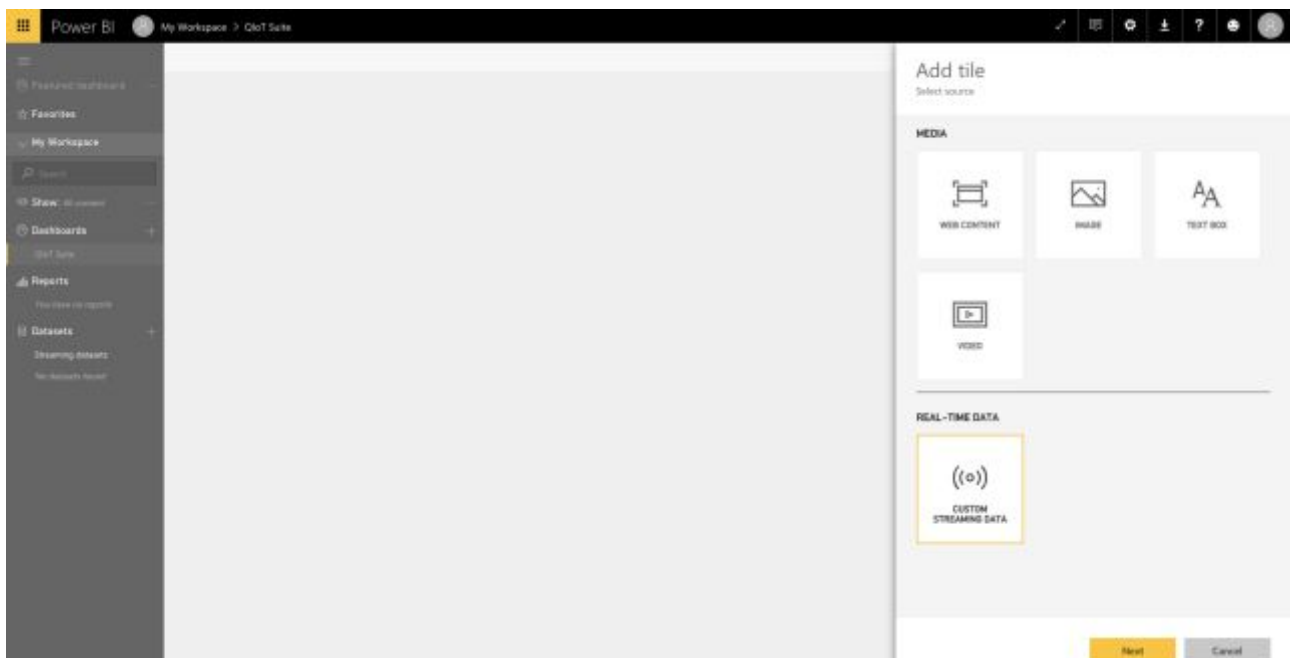




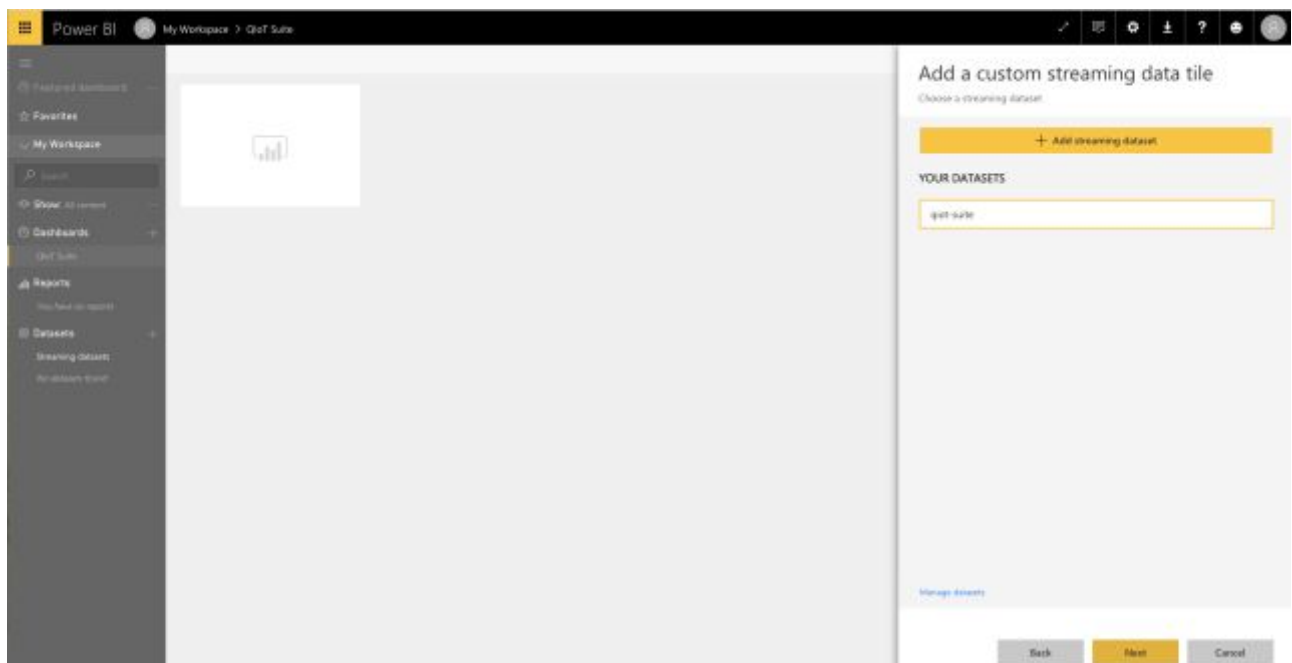
- Click “Add tile” in screen upper right corner



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



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



- Select visualization type (e.g., gauge), and set value, min, and max value.

Add a custom streaming data tile

Choose a streaming dataset > Visualization design

Visualization Type

Gauge

Value

temp

+ Add value

Minimum value

min

+ Add value

Maximum value

max

+ Add value

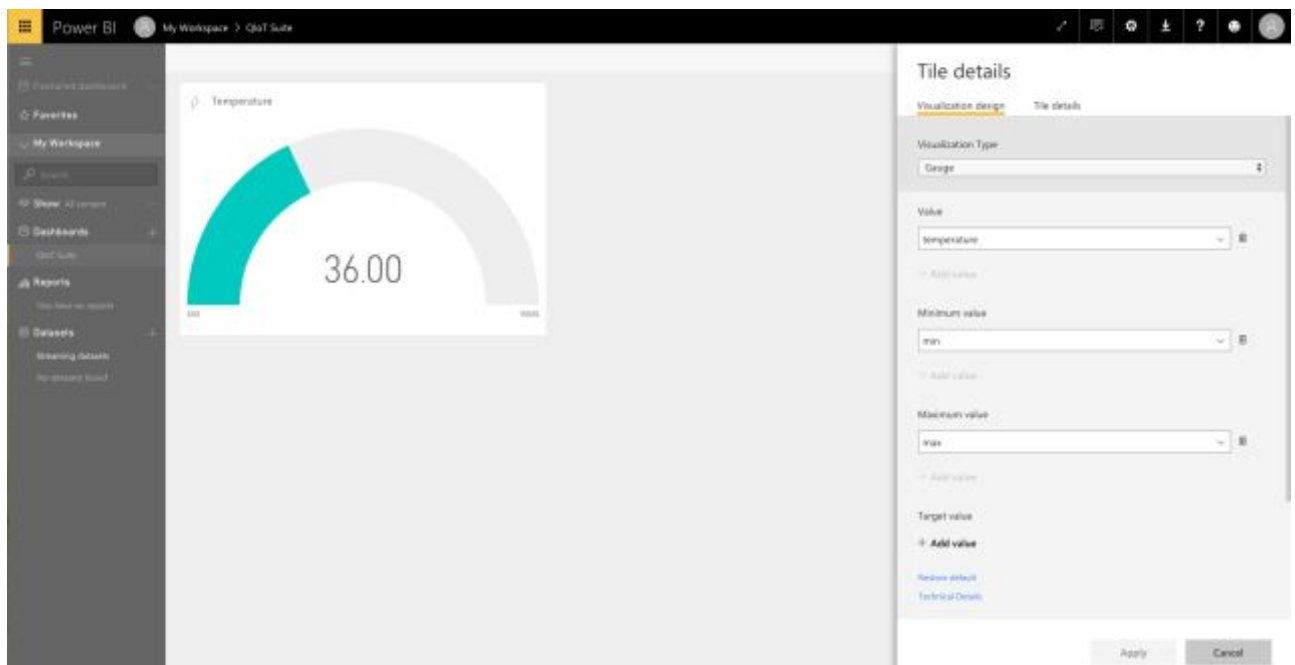
Target value

+ Add value

[Manage datasets](#)

Back Next Cancel

- 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 "port": 8883, # mqtts port }</pre>

	<pre> "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 "resourcetype": "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>