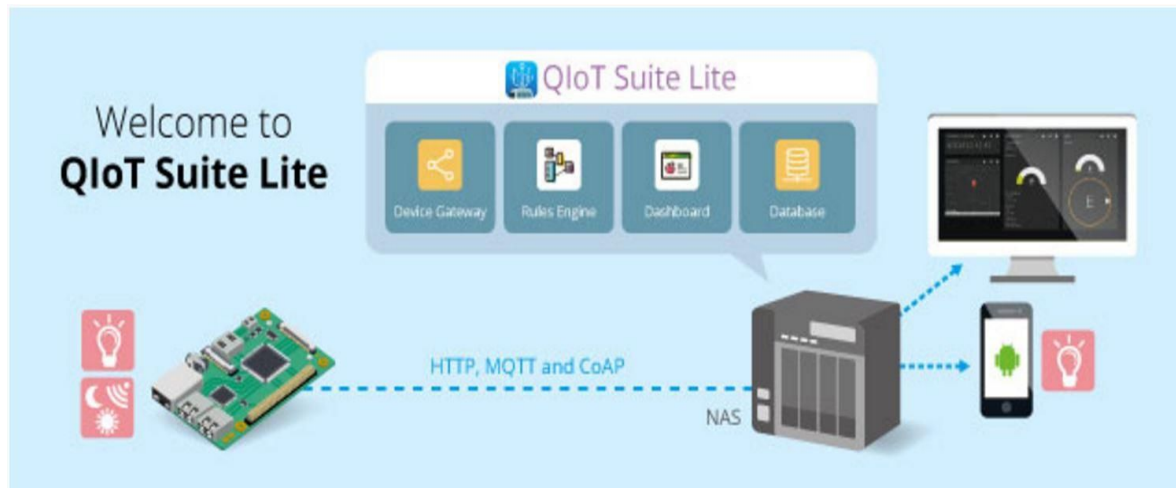


Get started with Arduino Yun (Python)

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



Lesson 1: Configure your device

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

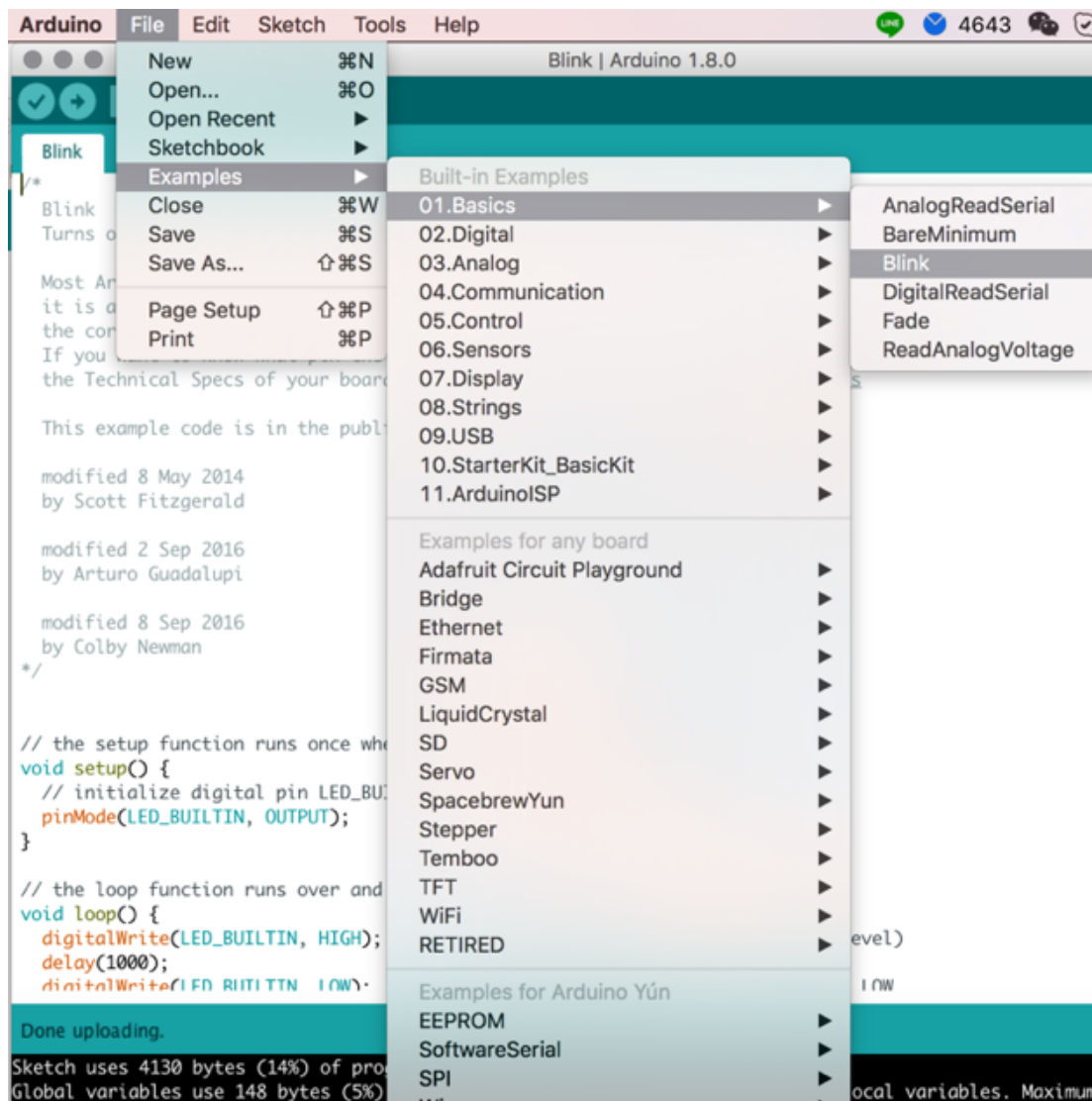
1.1 Download Arduino IDE

- Download and install Arduino IDE based on your host PC, available here: <https://www.arduino.cc/en/Main/Software>
- More Arduino tutorial: <https://www.arduino.cc/en/Tutorial/HomePage>

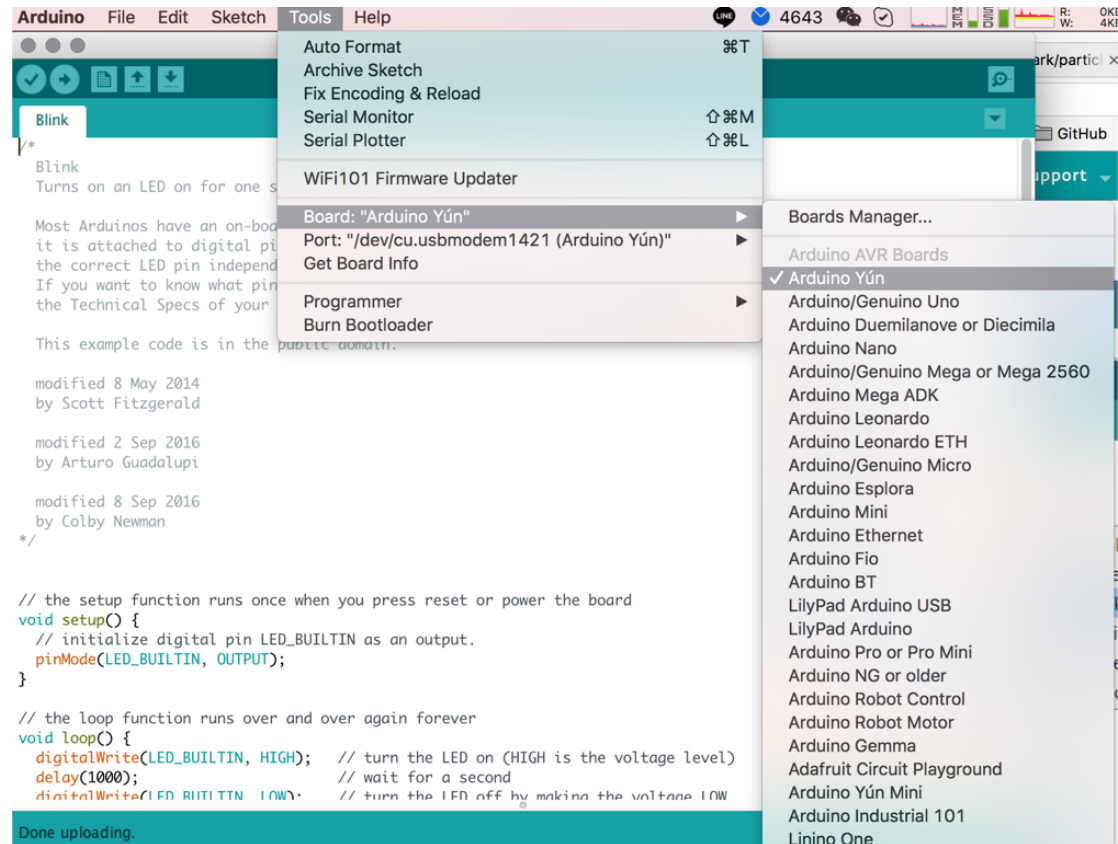
1.2 Configure and test your device

If this is the first time you use your Arduino Yun, you will have to follow some steps to assemble it.

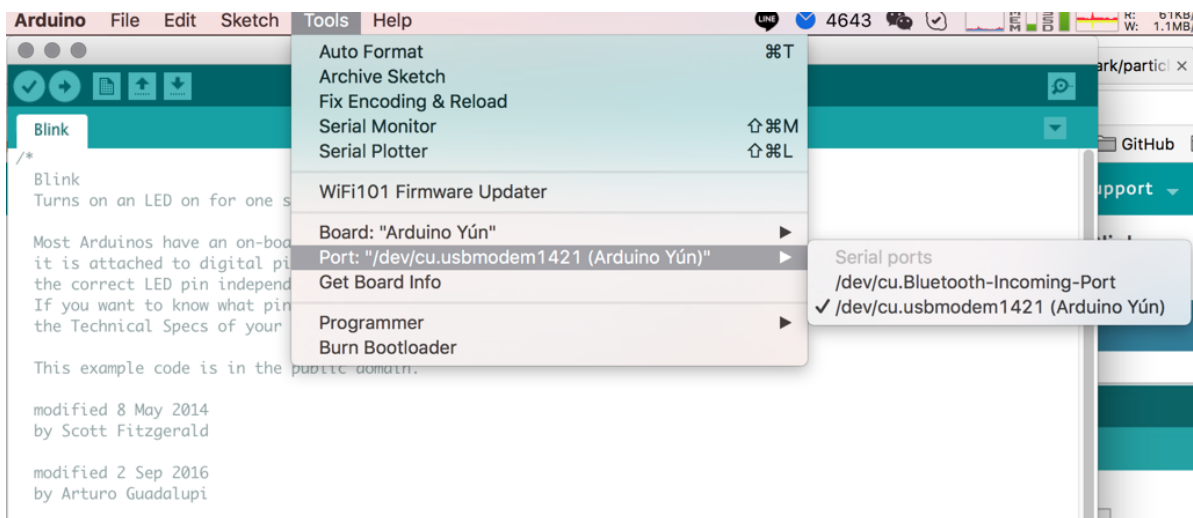
- Connect Arduino Yun to your PC with USB.
- Open your Arduino IDE and find Blink example.



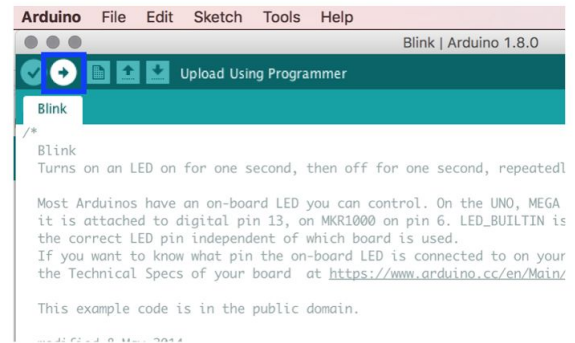
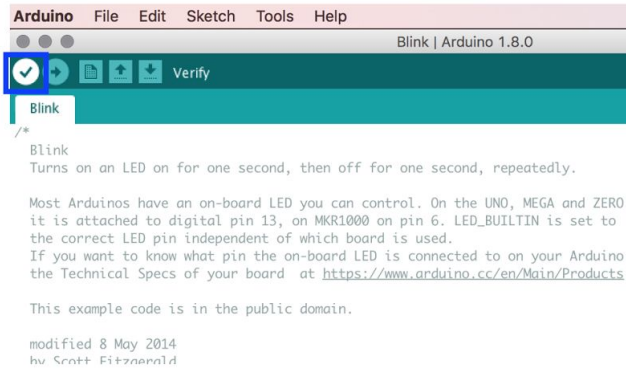
- Configure your board:



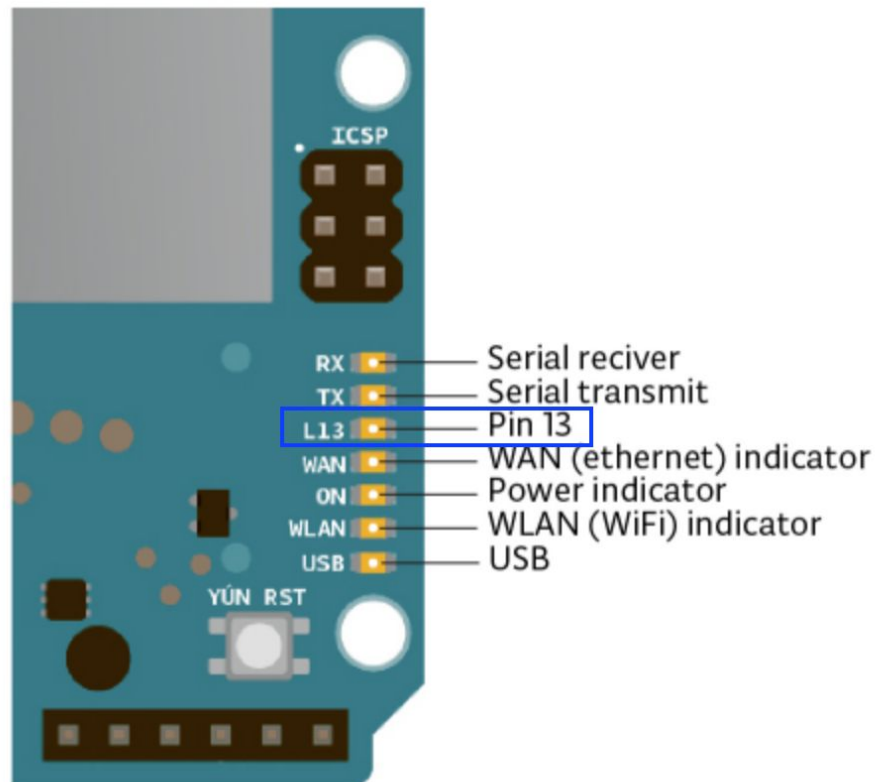
- Configure your port:



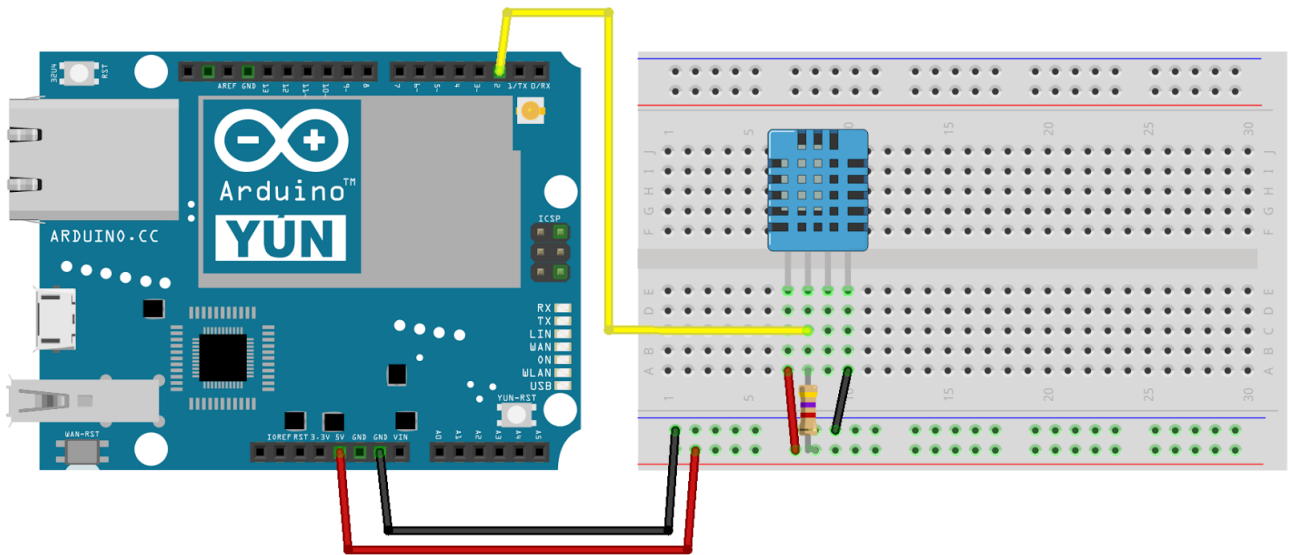
- Verify and upload your Blink example to Arduino Yun



- If you success upload your code to Arduino Yun, please check your Arduino Yun and you could see L13 is blink now:



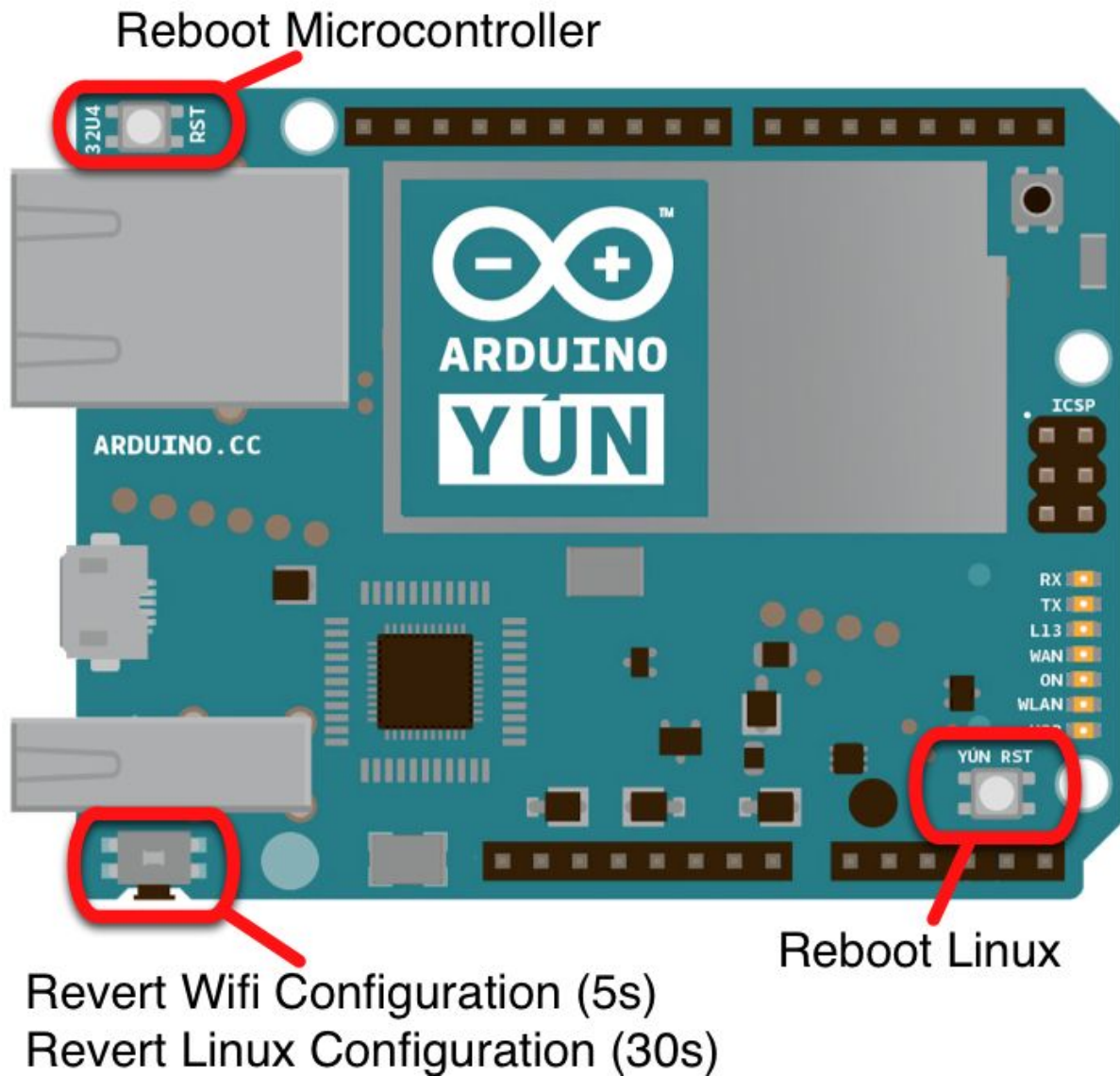
- Connect your DHT11 sensor to “Ground”, “5V”, “Pin 2(Digital)” on Arduino Yun respectively.



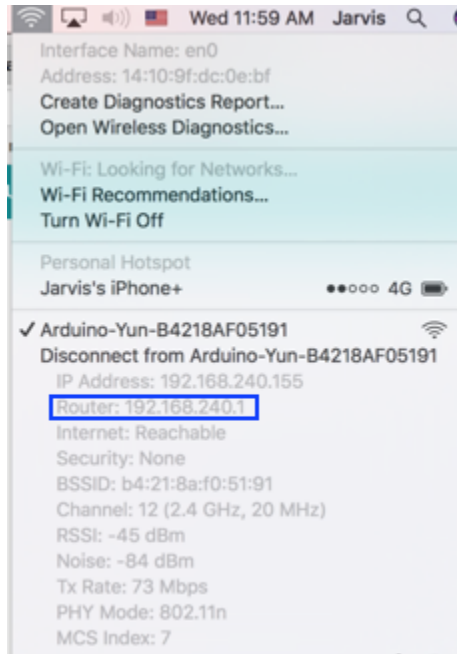
fritzing

1.3 Get IP address of your Arduino Yun

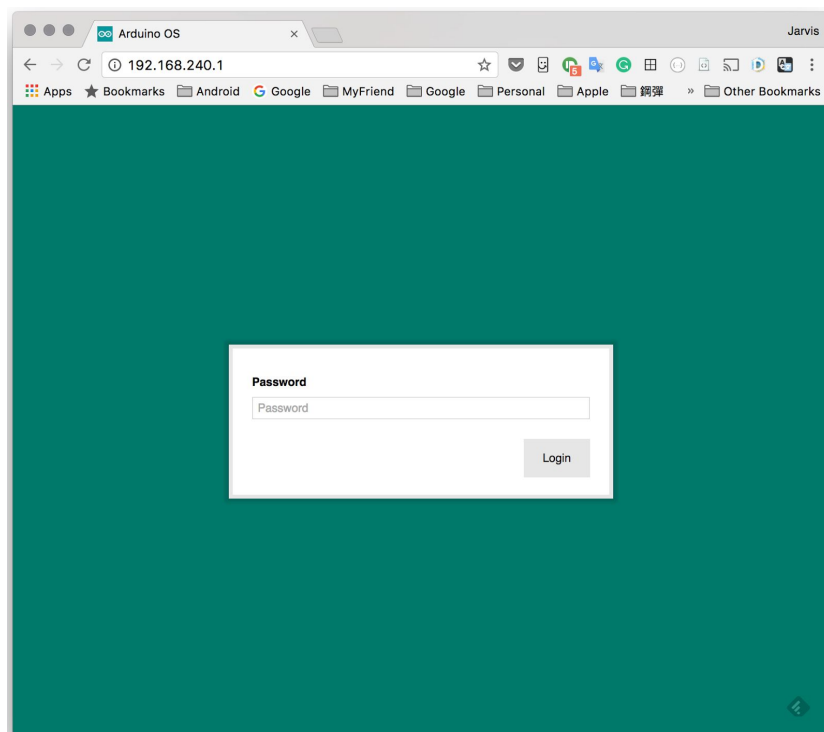
- Press Wifi reset button (need to press 5 seconds)



- Go to your Wifi settings and choose “Arduino-Yun-XXXXXXX”. The “XXXXXXX” is MAC address of your Arduino Yun.



- Open your browser and type in “<http://192.168.240.1>” or “<http://arduino.local>”. The default password is “arduino”.



- Complete your Board setting and link your Arduino Yun to the router you are using. Skip “API setting” and click on save to complete.



The screenshot shows the 'Board Settings' screen of the Arduino Configuration Wizard. The title bar reads 'Arduino Configuration Wizard'. On the left, there is a decorative graphic of colorful, curved lines. The main content area has the following fields: 'Board name' with the value 'arduino', 'Timezone' with the value 'Asia/Taipei', and 'Password' with two sub-fields: 'System Password' and 'System password (Repeat)'. At the bottom right, there are 'Back' and 'Next' buttons.

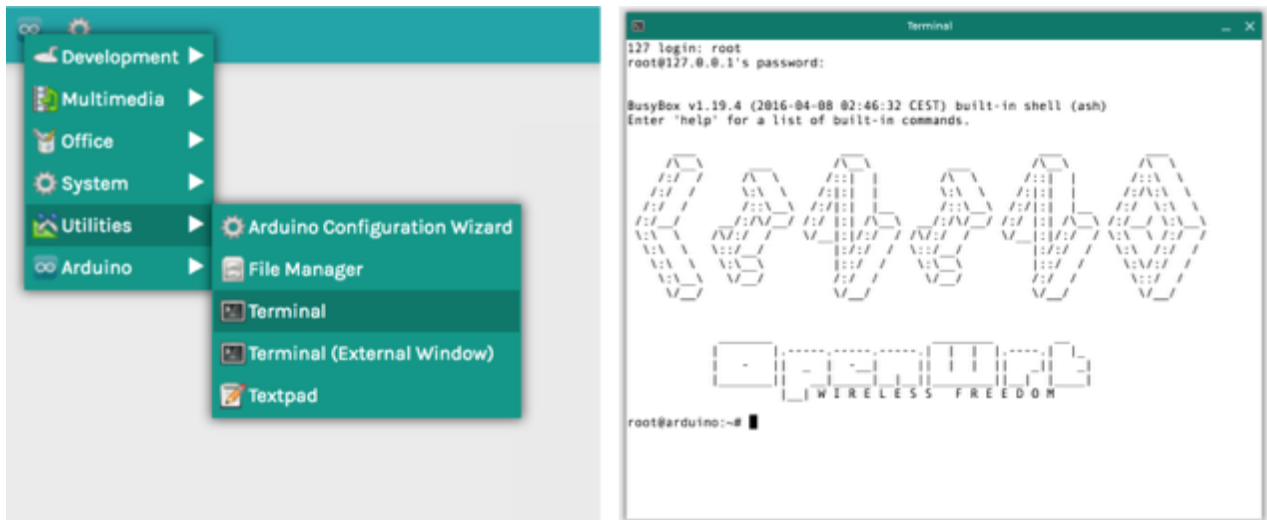


The screenshot shows the 'Wireless Settings' screen of the Arduino Configuration Wizard. The title bar reads 'Arduino Configuration Wizard'. On the left, there is a decorative graphic of colorful, curved lines. The main content area has the following fields: 'Wireless Network Name (SSID)' with the value 'Jen's Chung (default, 100% signal)' and a 'Scan' button, 'Security' with the value 'WPA2', and 'Password' with the value '12345678'. At the bottom right, there are 'Back' and 'Next' buttons.

- On your PC, change your Wifi network to your router and now your NAS and Arduino is under the same network.

1.4 Install development tools for your Arduino Yun

- Open Terminal from <http://arduino.local>
- Type your login account “root” and your password. The default password is “arduino”.



- Install PIP and relatives library by tying in the following commands.
root@arduino:~# opkg update
root@arduino:~# opkg install distribute
root@arduino:~# opkg install python-openssl
root@arduino:~# easy_install pip

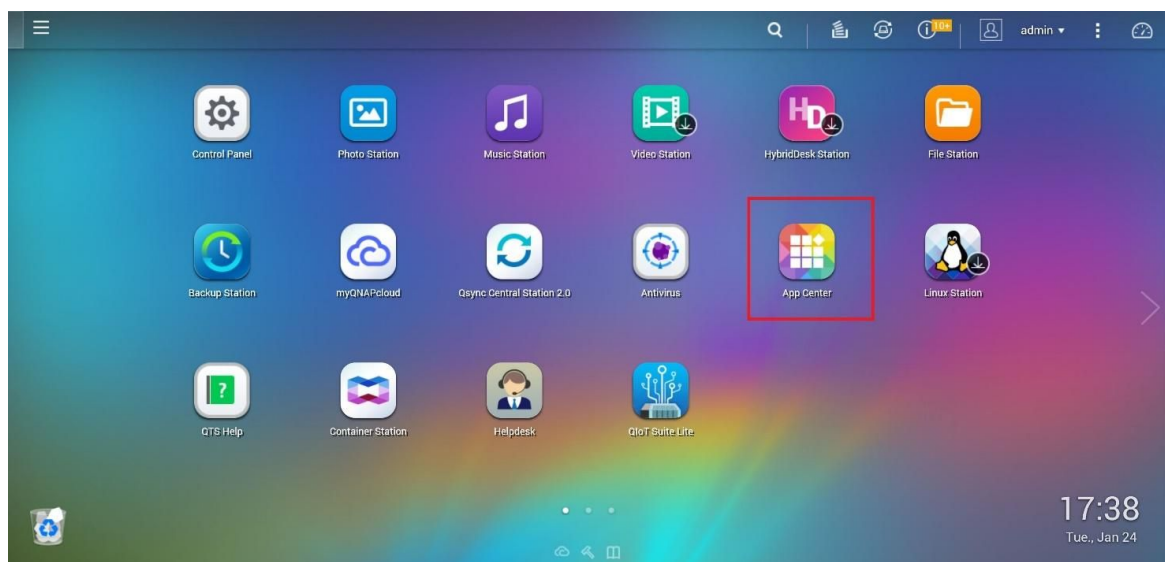
Lesson 2: Create your device in QIoT Suite Lite

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.



QIoT Suite Lite
Your private IOT cloud

QIoT Suite Lite

- ☐ Keep me logged in for a week
- ☐ Secure Connection

Login

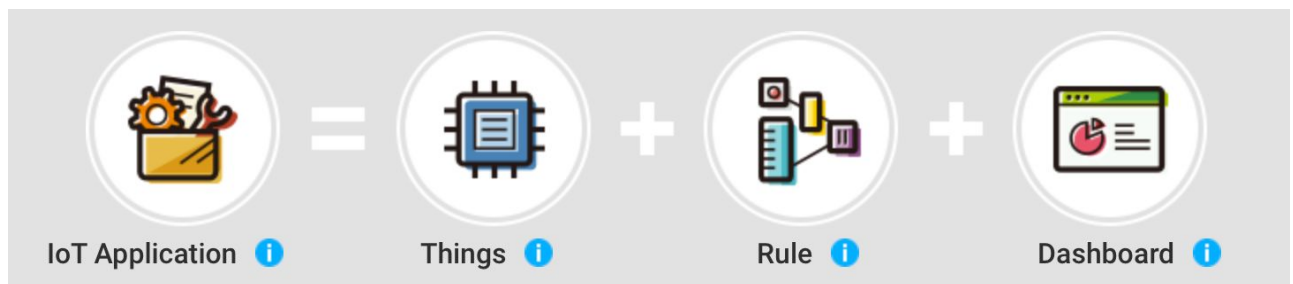
Create a new IoT application

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.

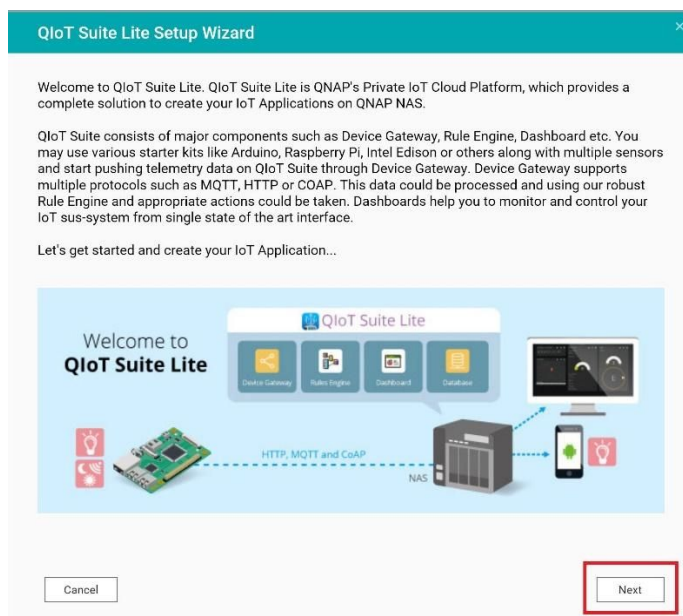
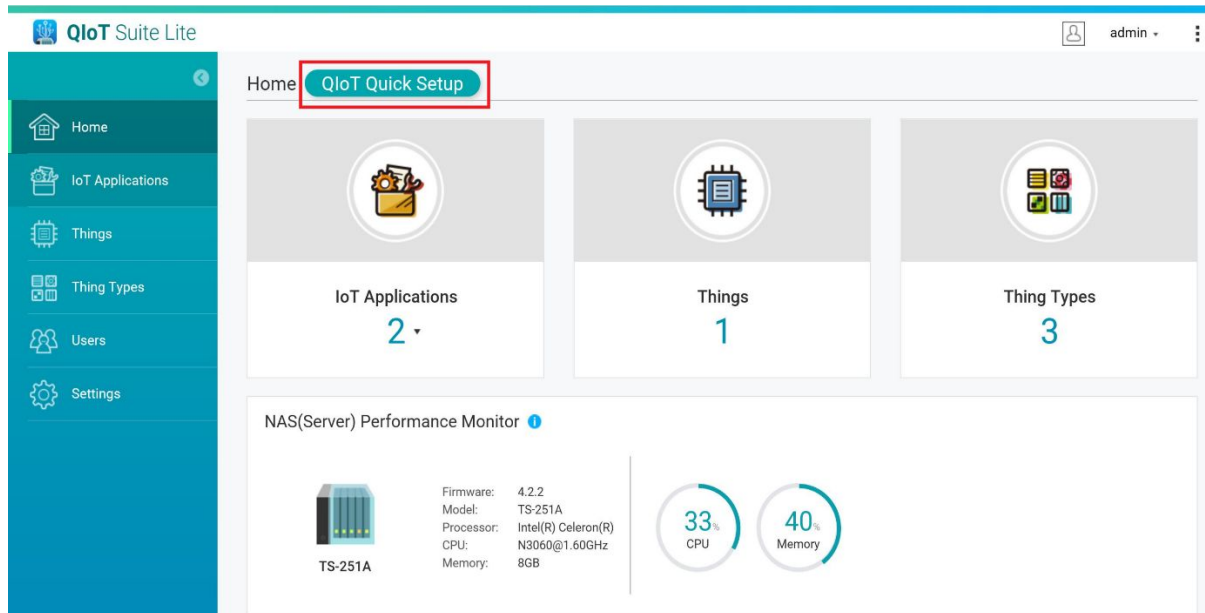
The Rule makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices based on business rules you define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel.

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 *

myIoT_App-1

Description:

Please enter the Description

Rule Name:

Rule_myIoT_App-1

Description:

Please enter the Description

Dashboard Name:

Dashboard_myIoT_App-1

Description:

Please enter the Description

Note: Inputs with * are required field

Cancel

Back

Next

- 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.

+ Add

Remove

	Thing Name	IP	User Name	Password	Action
<div> </div> <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

Back

Next

Page 14 of 32

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*

myThing

Description

Please enter the Description

Thing Category

QIoT Supported

Custom

Thing Type

Thing Type

Arduino Yun

Attributes for myThing (Optional)

+ Add Attribute

Delete

Note: Inputs with * are required field

Add

Cancel

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.

+ Add

Remove

	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.

Cancel

Back

Next

- 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 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:

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

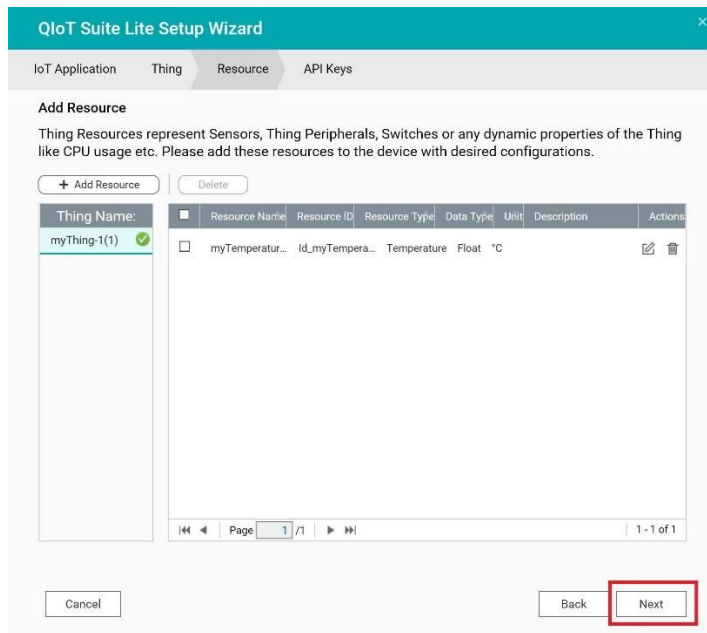
Note: Inputs with * are required field

Add

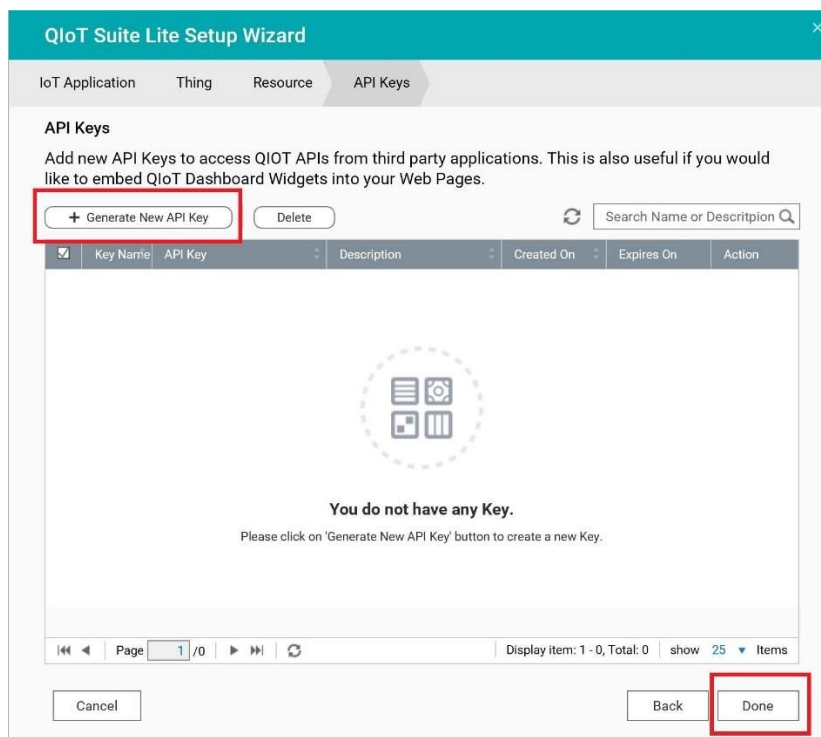
Cancel

Page 16 of 32

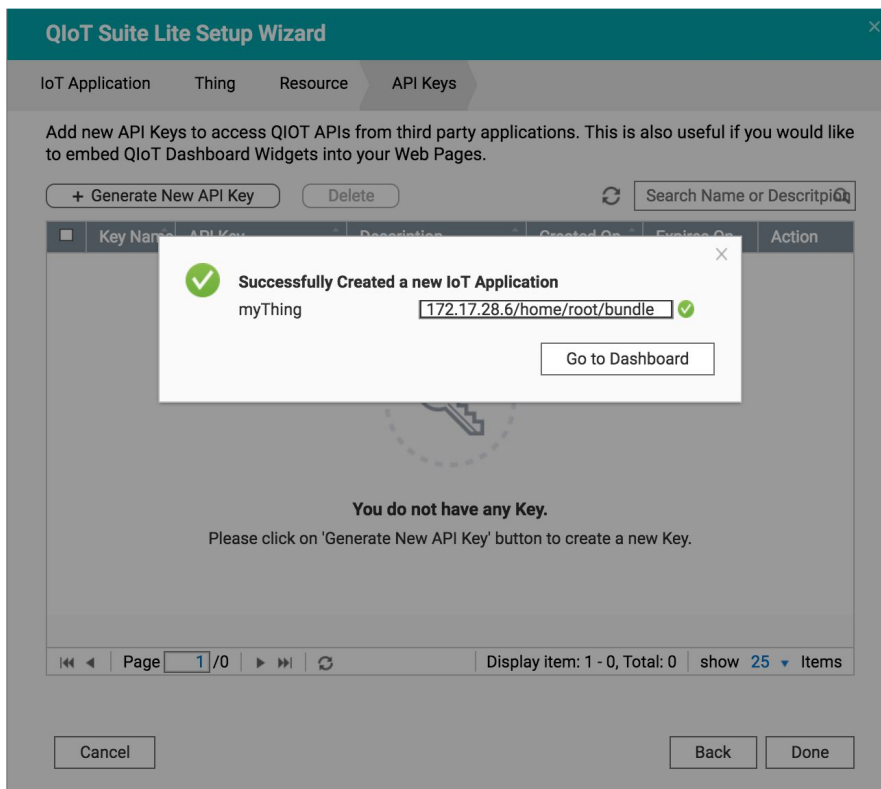
Click **Next** after you add all the resources (sensors) on your device.



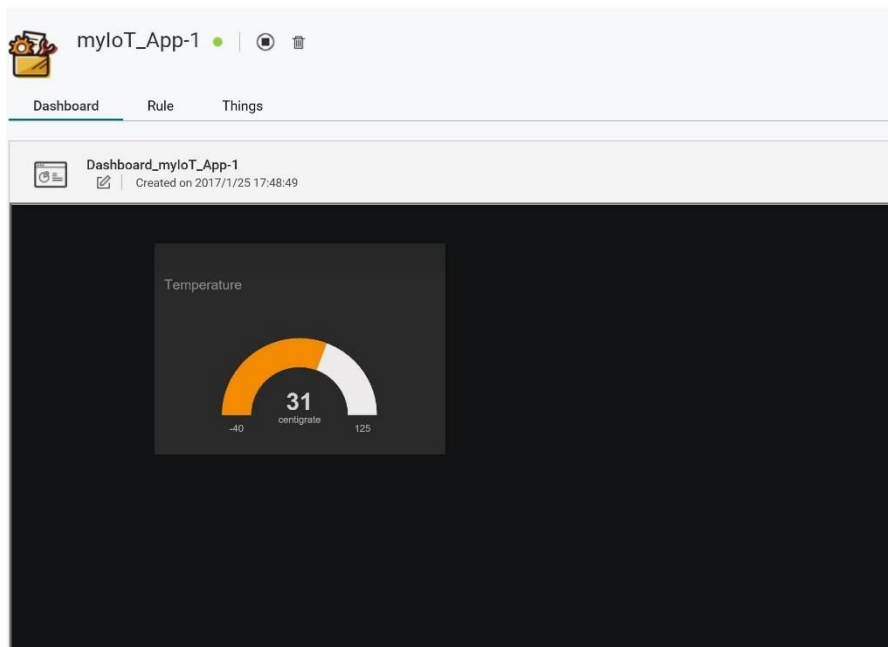
- In API Keys tab, click on **Generate New API Key** if you would like to embed QIoT dashboard widgets into your web pages or access QIoT APIs from third party applications. Click on **Done**.



Your IoT application is created successfully. Python 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.

mylot_App-1

Dashboard Rule Things

Rule_mylot_App-1
Created on 2017/1/26 11:02:37

QIoT Rule

Flow 1

Configure QBroker node and connect to the dashboard

1. Subscribe things

2. Select resource type

MQTT Message In

Dashboard

Click debug tab to watch data stream

Simulate Event from SDK

3. Publish things

Virtual Event

dummy data

MQTT Message Out

QDashboard

QBroker

QDatabase

QHistoric

input

Inject

Deploy

Save

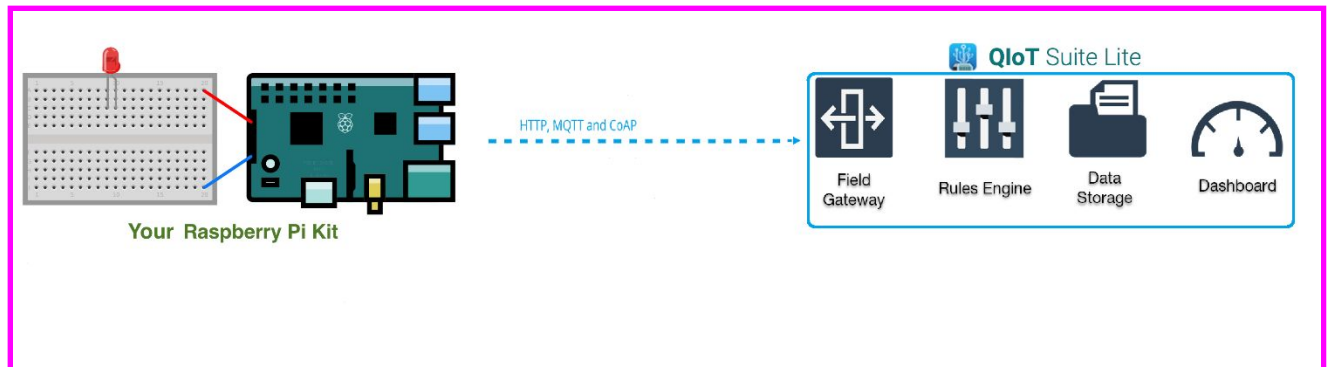
info

debug

dashboa

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 Arduino Yun to QIoT Suite Lite.



3.1 Connect to QIoT Suite.

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

```
root@arduino:~# cd /home/{{user}}/bundle
// mqtt
root@arduino:~# pip install paho-mqtt
// http
root@arduino:~# pip install requests
// coap
root@arduino:~# pip install coaphon
```

```
root@arduinoyun:/home/root/bundle# pip install paho-mqtt
Requirement already satisfied: paho-mqtt in /usr/lib/python2.7/site-packages/paho_mqtt-1.2-py2.7.egg
root@arduinoyun:/home/root/bundle# █
```

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

```

while 1:
    """
    about ./res/resourceinfo.json
    {
    .. ~
    .. "resources": [
    .. .. {
    .. .. .. "resourceid": "temp",
    .. .. .. "topic": "qiot/things/admin/abccccc/temp",
    .. .. .. }
    .. .. }
    .. }
    }
    TODO: you could replace "temp" by any resource id set form QIoT Suite Lite
    """
    connection.publish_by_id("temp", str(random.randint(0, 41)))
    """
    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));
    """
    time.sleep(1)
~

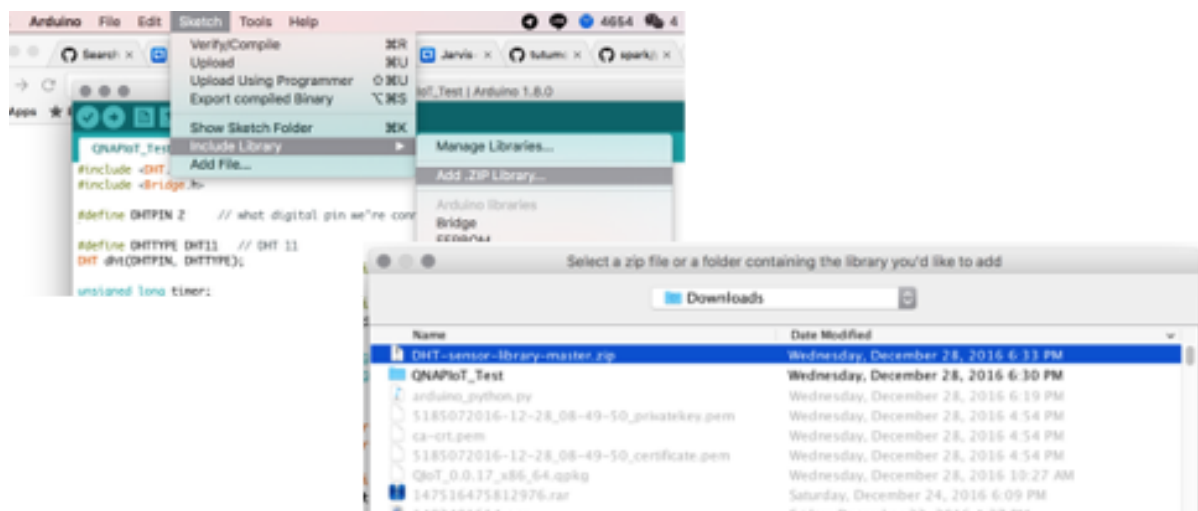
```

- Run the sample application.

root@arduino:~# python mqtt.py

3.2 Deploy Arduino Yun ino code and dht11 library

- Download the sample code (dht11.ino and DHT-sensor-library-master.zip) at GitHub: [Sample Code](#)
- DHT-sensor-library-master.zip comes from <https://github.com/adafruit/DHT-sensor-library>
- Open Arduino IDE. Find “DHT-sensor-library-master.zip” file and include it.



- Open the “dht11.ino” by Arduino IDE and upload your Arduino sample code.



```
QNAPIoT_Test | Arduino 1.6.11
QNAPIoT_Test
dht.begin();
}

void loop() {
  // float h = dht.readHumidity();
  // float t = dht.readTemperature();
  // float f = dht.readTemperature(true);
  // // Check if any reads failed and exit early (to try again).
  // if (isnan(h) || isnan(t) || isnan(f)) {
  //   Serial.println("Failed to read from DHT sensor!");
  //   return;
  // }
  // // float hif = dht.computeHeatIndex(f, h);
  // // float hic = dht.computeHeatIndex(t, h, false);

  if (millis() - timer > 200) {
    timer = millis();
    Bridge.put("temperature", String(t));
    // Bridge.put("humidity", String(h));
  }
}

Done Saving.
/Users/claudiohong/Downloads/DHT11/ino/DHT11-Bridge-Demo.ino
/Users/claudiohong/Downloads/DHT11/ino/DHT11-Bridge-Demo.ino
/Users/claudiohong/Downloads/QNAPIoT_Test/QNAPIoT_Test.ino
22 Arduino Yun on /dev/cu.usbmodem1411
```

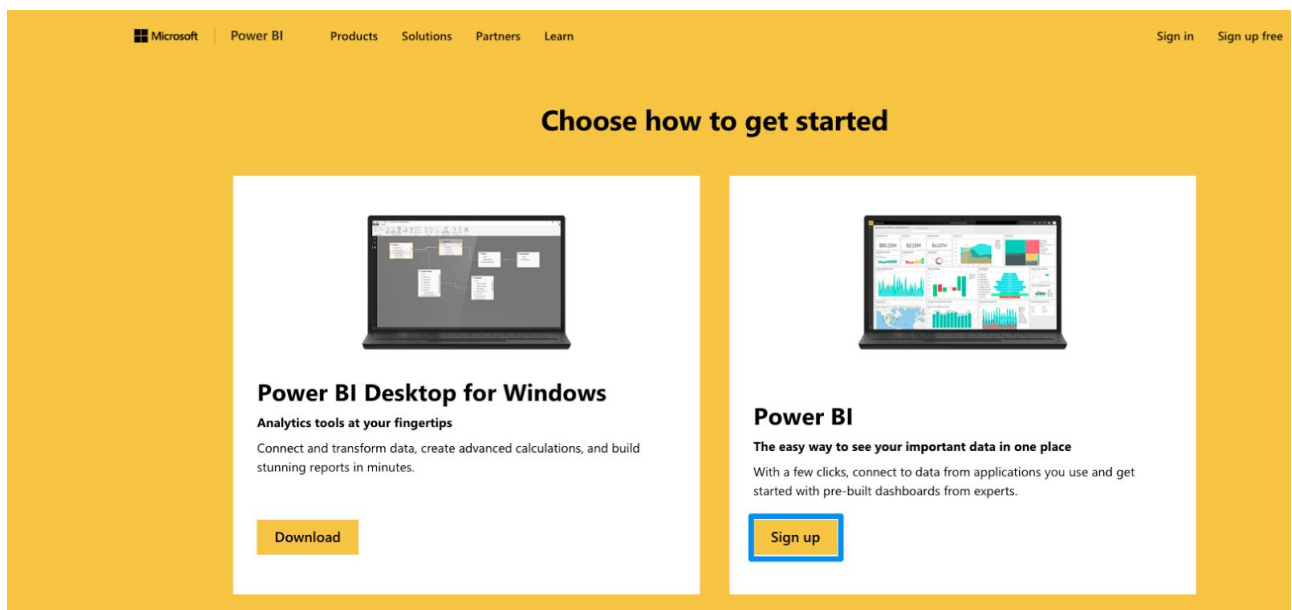
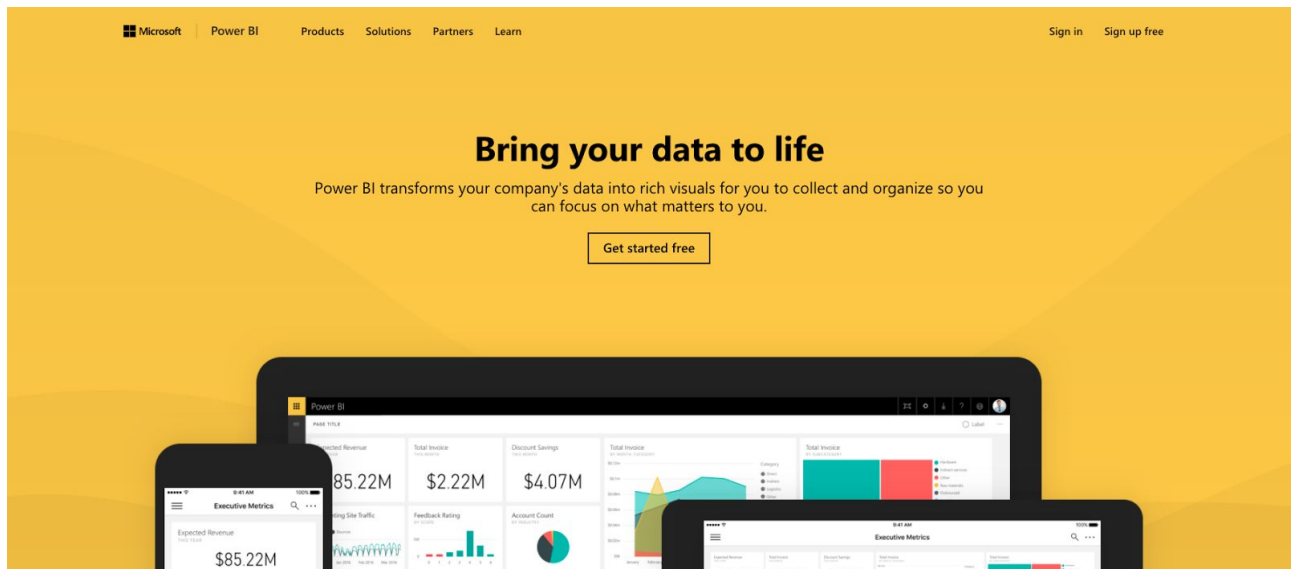
- Run the dht11 sample application.

```
root@arduino:~# python dht11.py
```

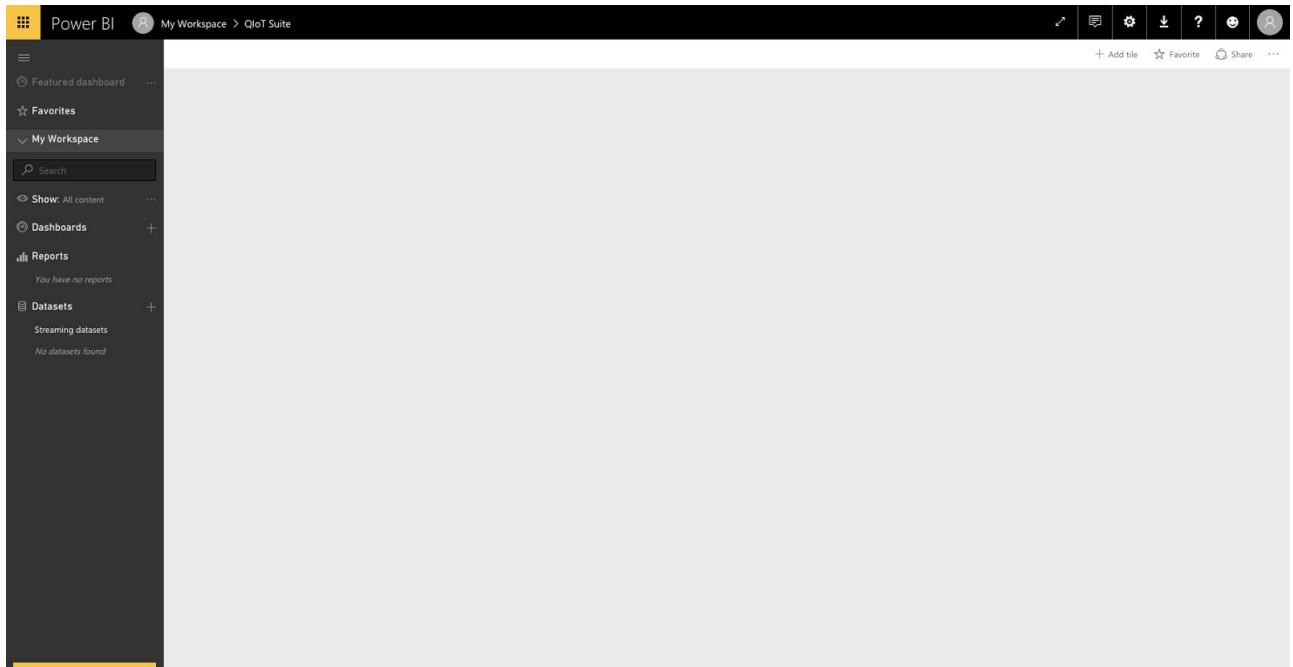
Lesson 4: Integrate Power BI

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.

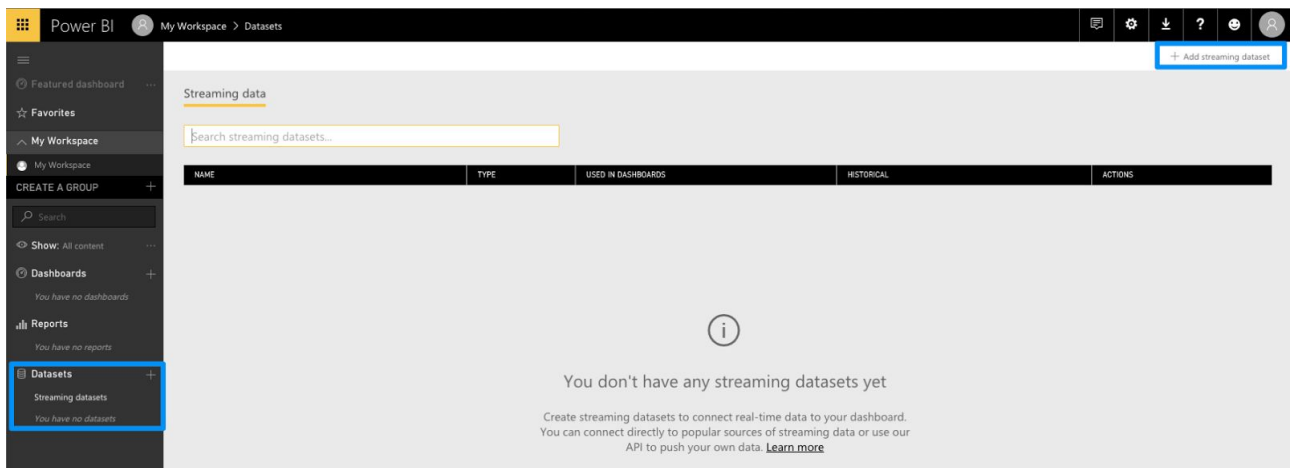


- After a sequence of registration, the page will lead you to below page

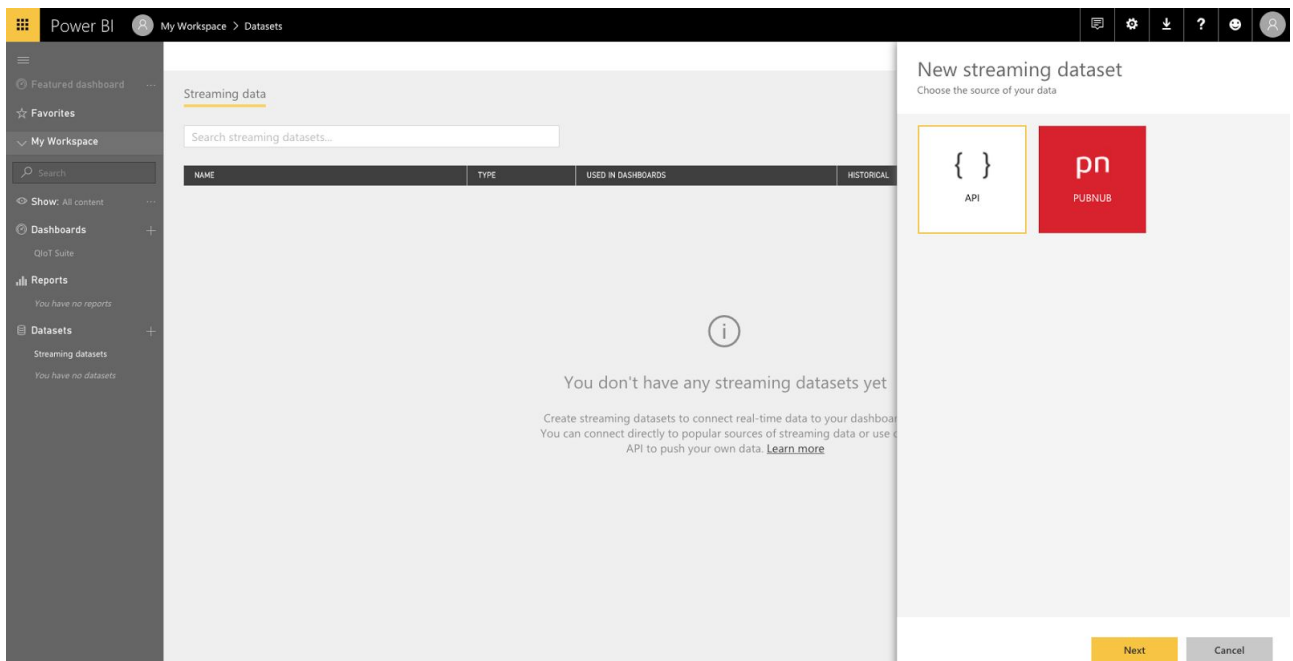


4.2 Setup your streaming dataset API

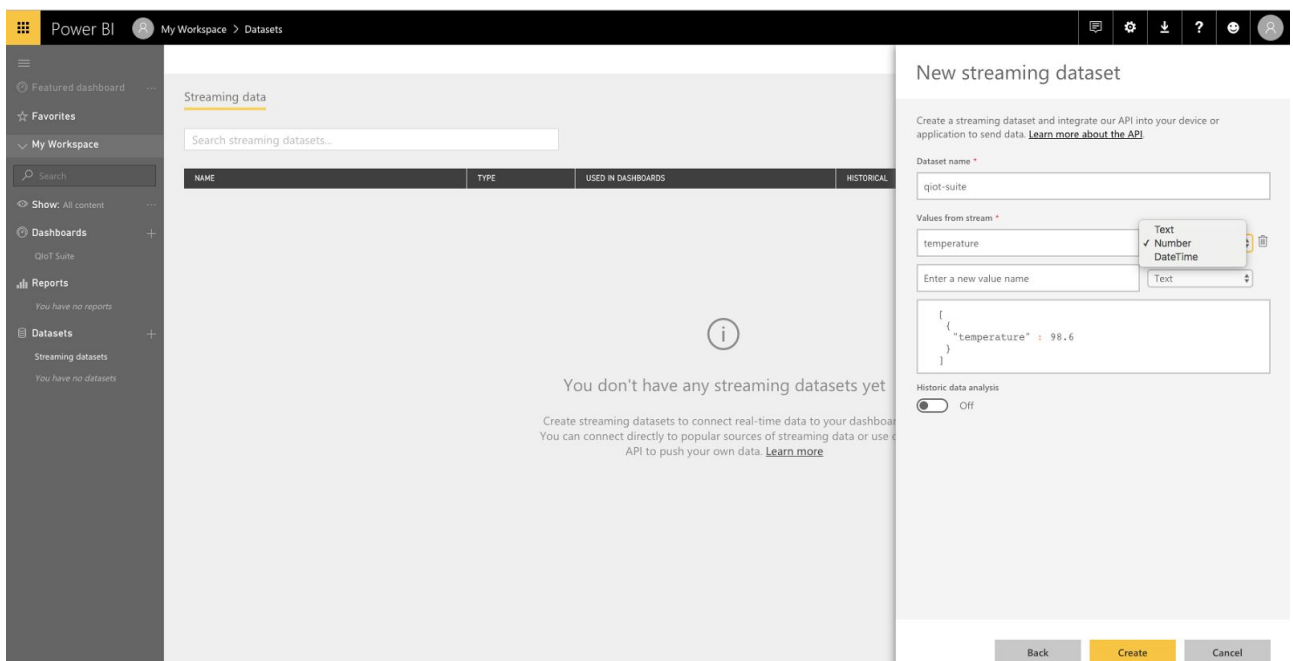
- Go to “**Datasets -> Streaming datasets**” section at left menu, and click the “**+ Add streaming dataset**” button at right-top corner.



- 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

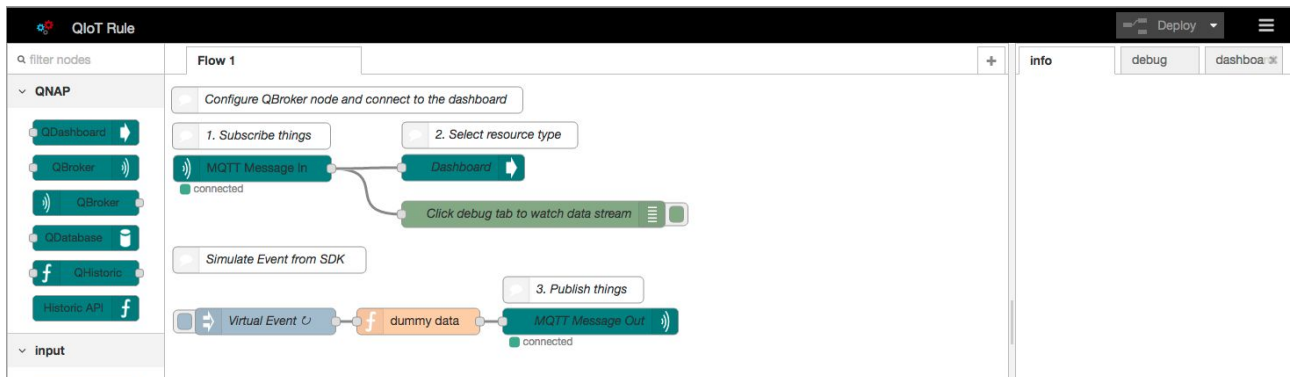
`https://api.powerbi.com/beta/bb3391c7-d712-450b-949c-14d42c1dff4e/datasets/qiot-suite`

Raw cURL PowerShell

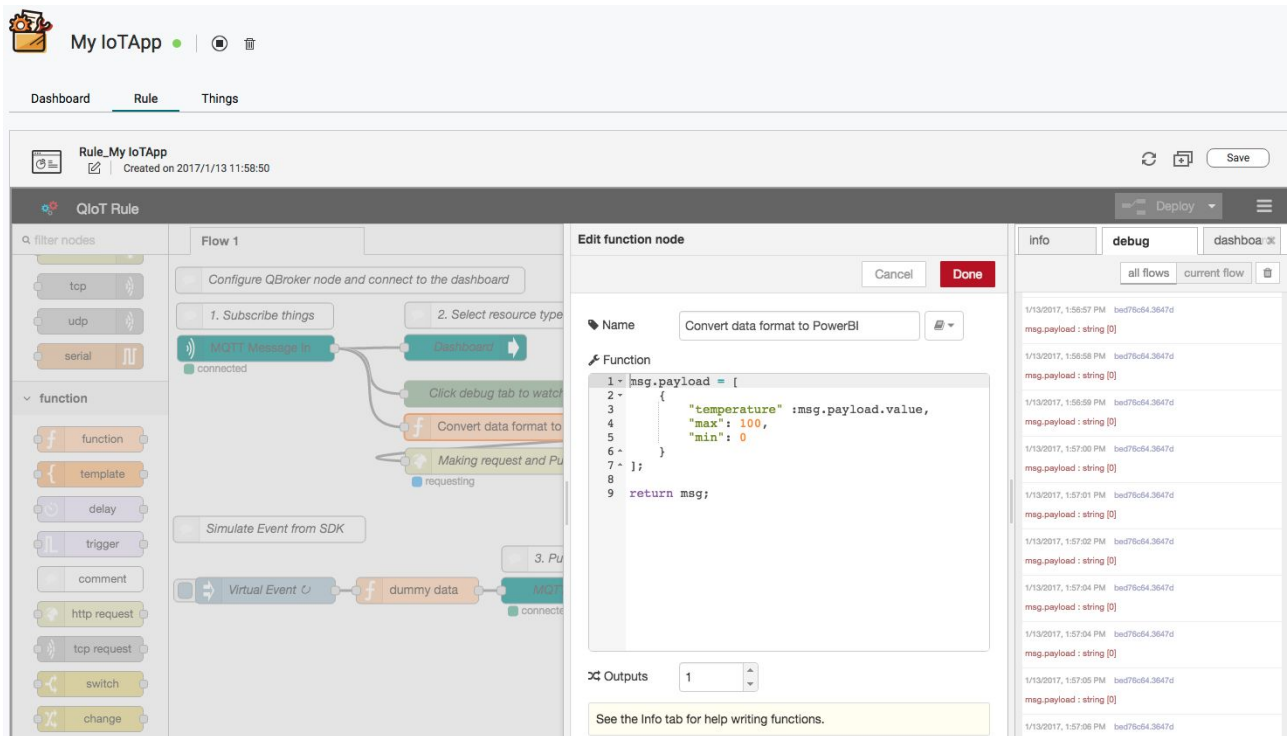
```
{
  "temperature": 98.6,
  "min": 98.6,
  "max": 98.6
}
```

4.3 Configure Node-RED's nodes in IoT application

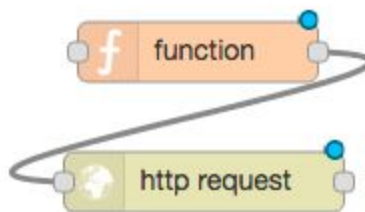
- Create IoT application in QIoT suite. The following one is your first node-red flow, and then you can start to create your own IoT flow. more node-red information can be found in “<https://nodered.org/>”.



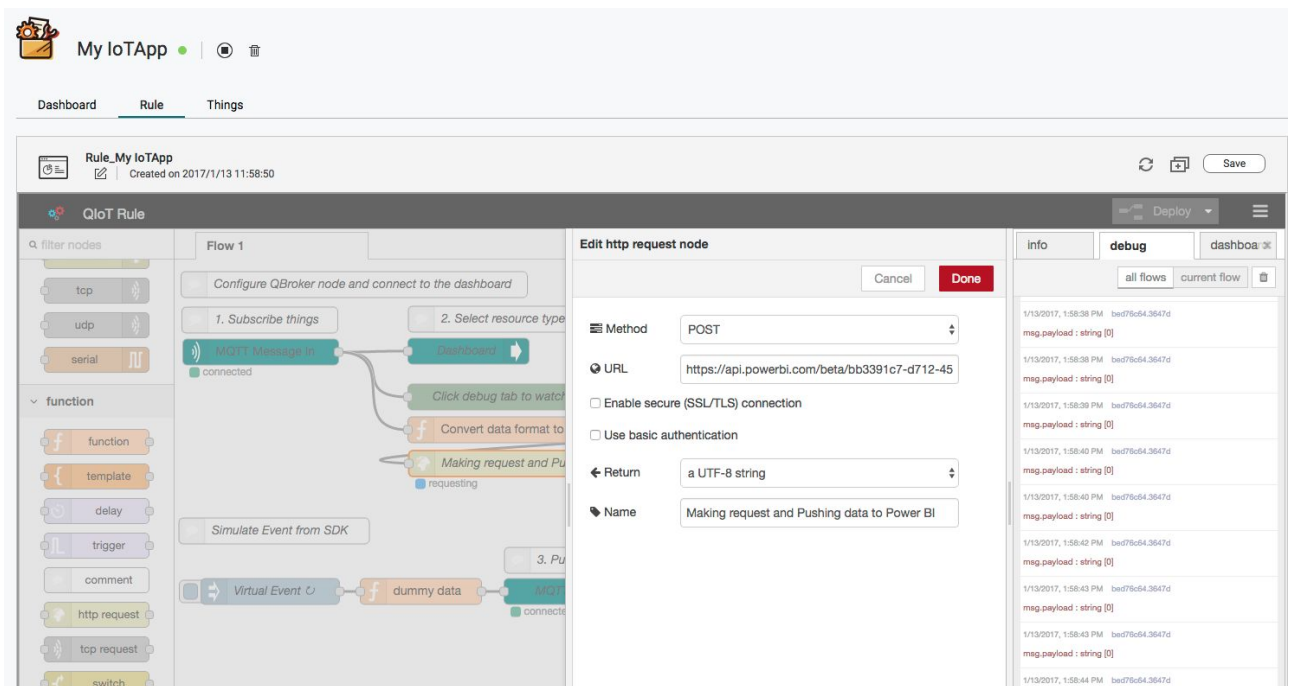
- Before you start to push live data to Power BI. We need a “function” node to convert IoT data to streaming data dataset. Here you can replace `msg.payload` to your JSON dataset.



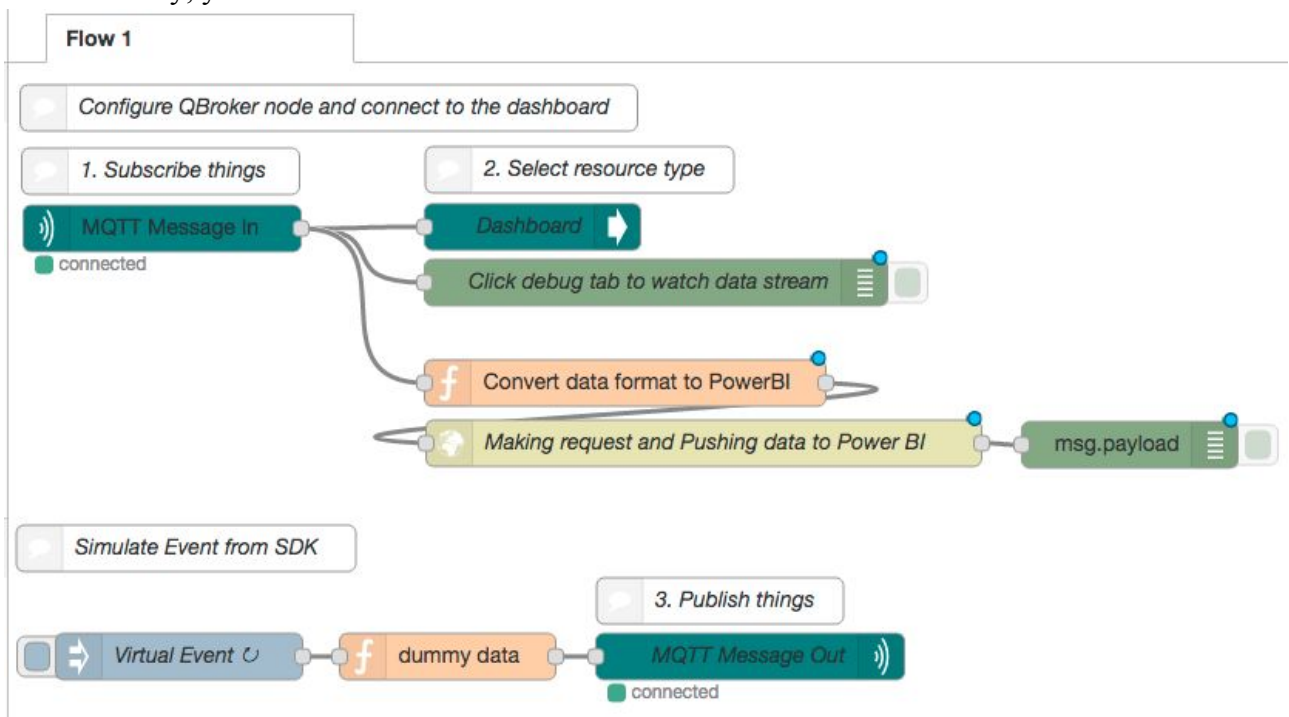
- We need a “http request” node to help us to push live data to Power BI. Just drop and drag “http request” node and connect to tail of “function” node.



- 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.

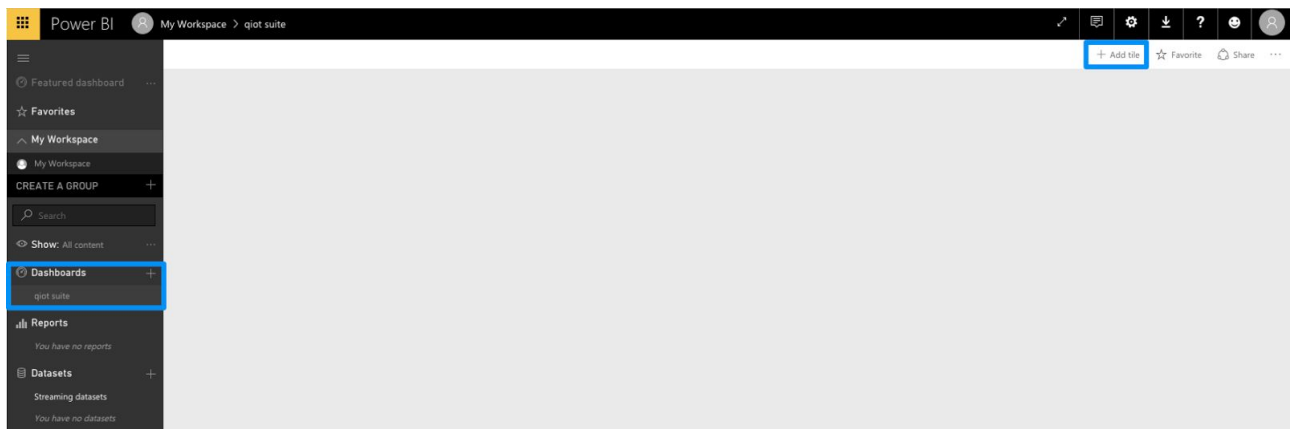


- Finally, your node-red flow will look like below one.

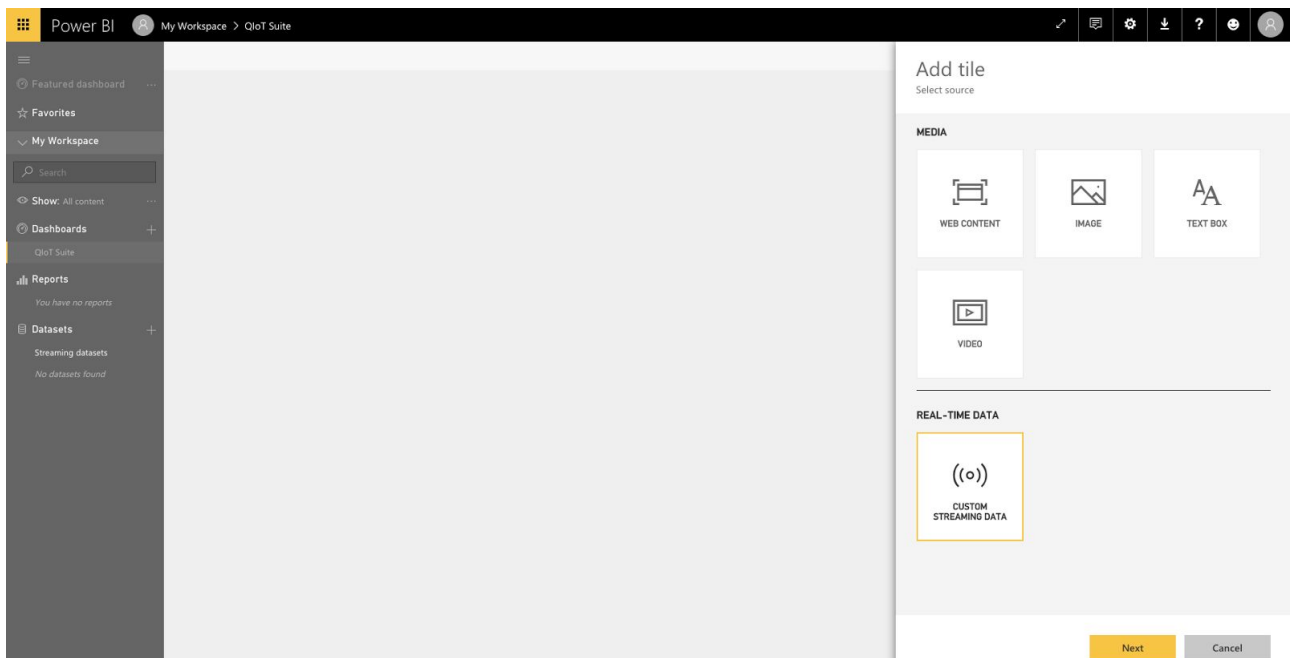


4.4 Add tile to display real-time data

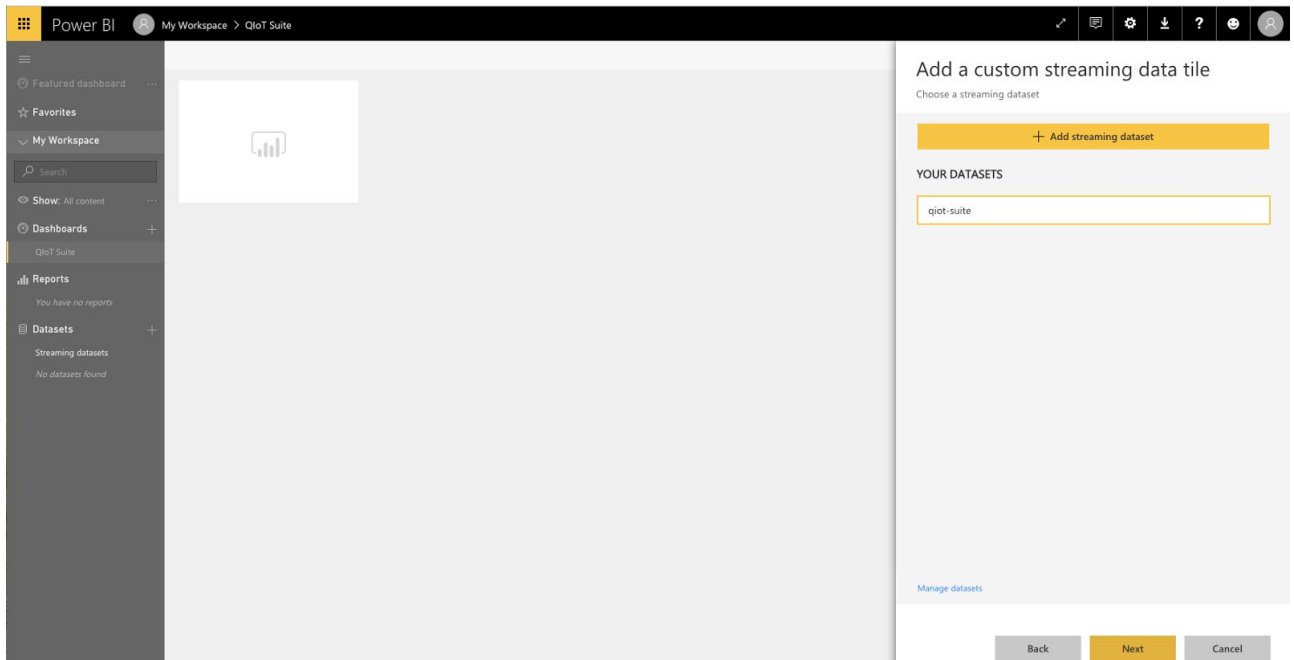
- Go to “**Dashboards +**” to create your first dashboard, then click the “**+ Add tile**” button to configure a widget.



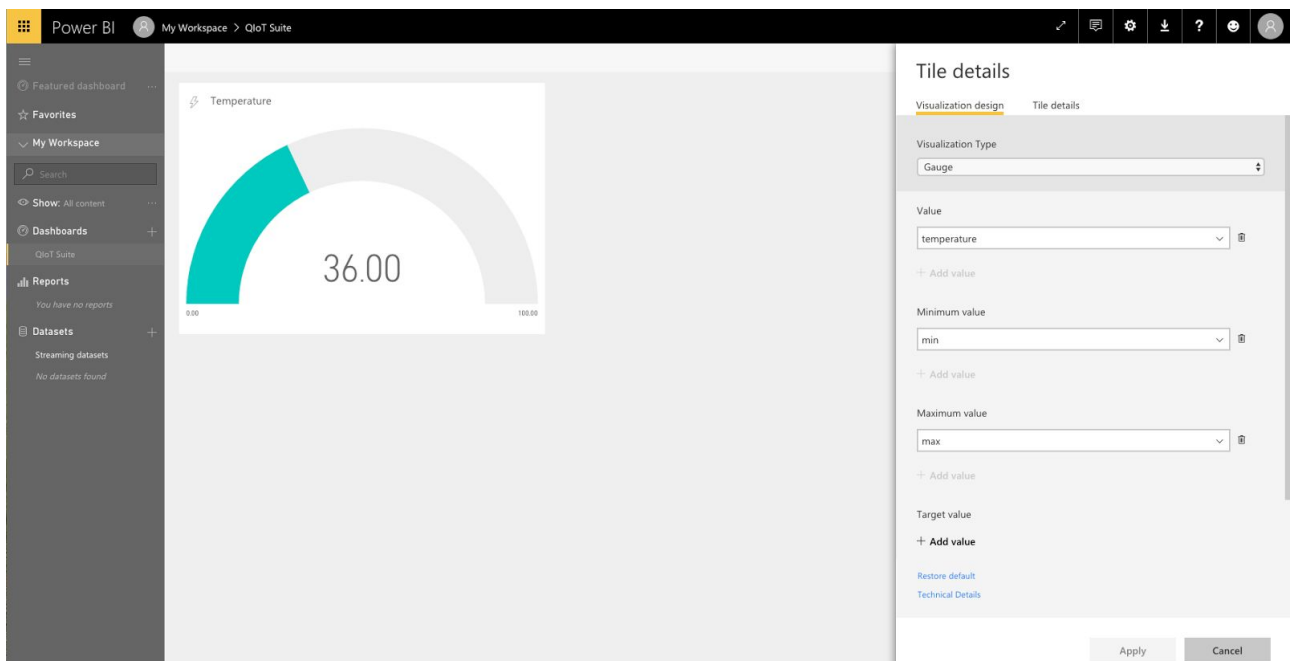
- Select “CUSTOM STREAMING DATA” and then select the “Next” button.



- 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/
  python/                                # python program language
  device/                                # arduino-yun...
    arduino-yun/
      examples/
        lib/                             # QIoT command Lib
        res/                             # QIoT resourceinfo.json folder
        ssl/                             # QIoT certificate files folder.
        mqtt.py                          # sample code - mqtt/mqtts publish
        http.py                           # sample code - http post
        https.py                          # sample code - https post
        coap.py                           # sample code - coap postt
        mqtt_subscribe.py                 # sample code - mqtt/mqtts subscribe
        http_get.py                       # sample code - http get
        https_get.py                      # sample code - https get
        coap_observe.py                   # sample code - coap get
  nodejs/                                # node.js program language
    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 "clientId": "adfa_1491561635", # thing Id "username": "00477f86-425b-49de-8590-xx", # username "password": "r:2825dedfb012969e1dfb6adb8", # password } </pre>

	<pre> "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>