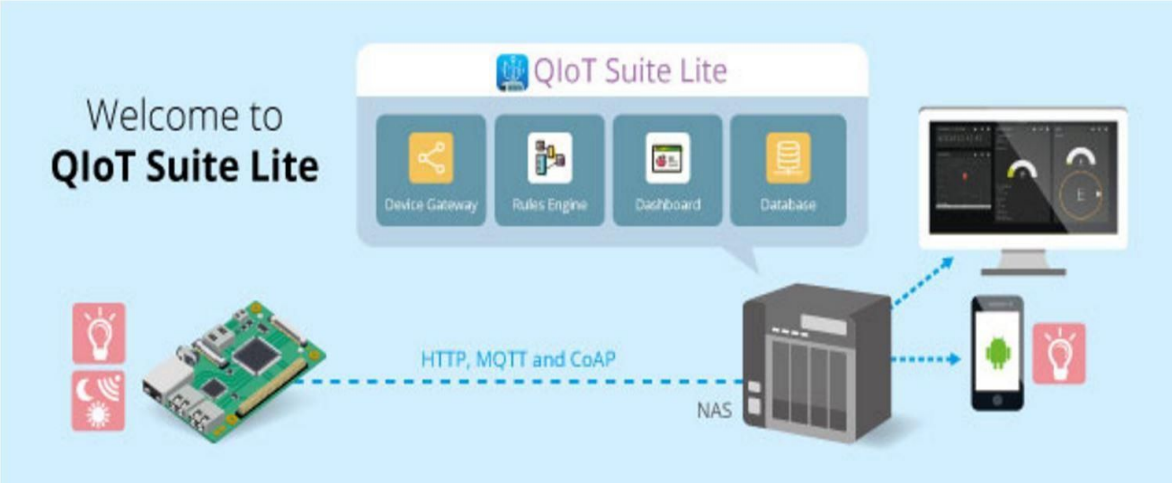
Get started with Arduino Yun (Python)

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.

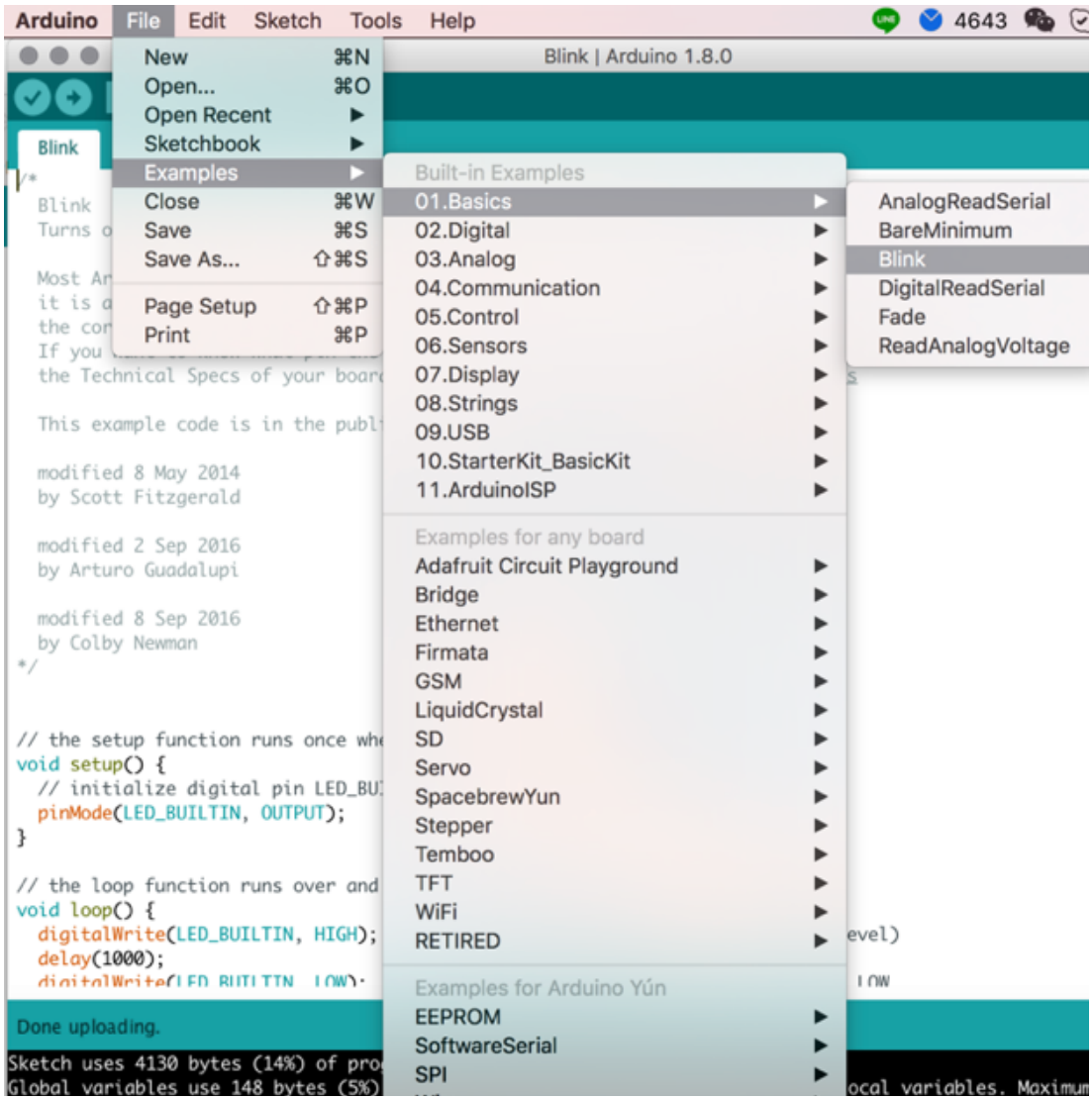
1.1 Download Arduino IDE

- 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

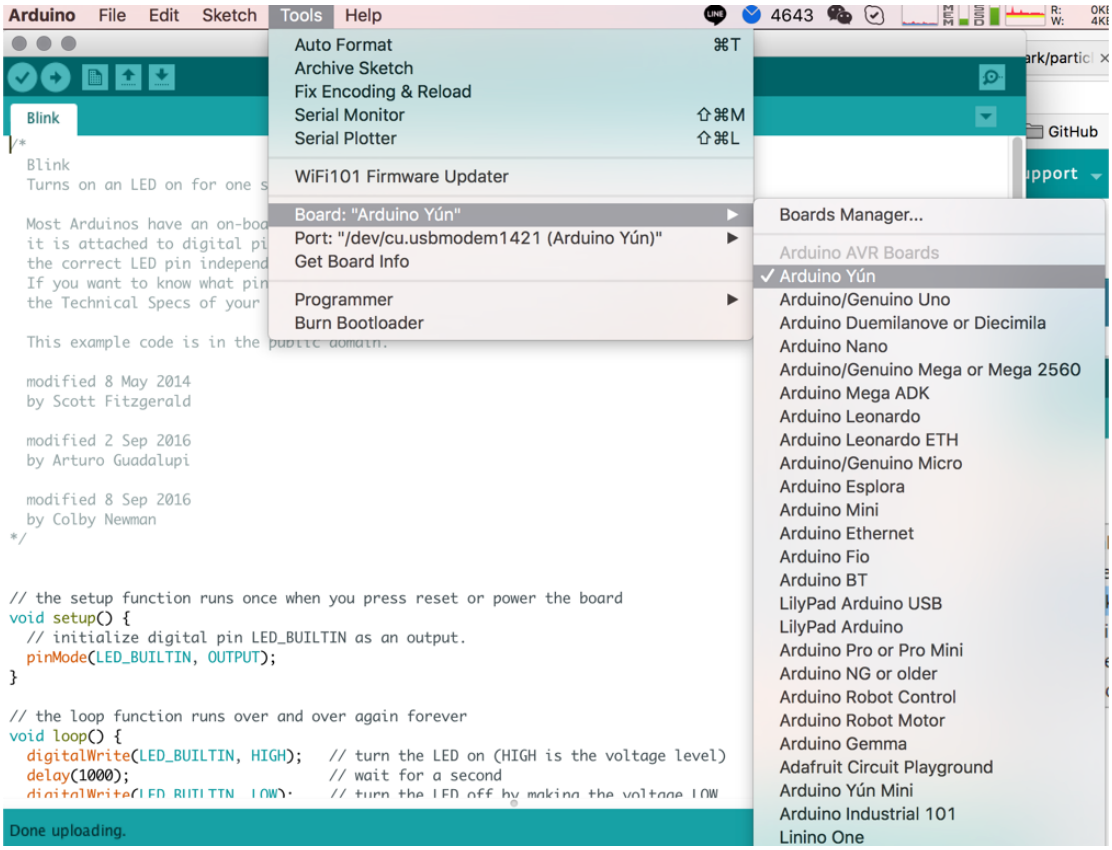
1.2 Configure and test your device

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

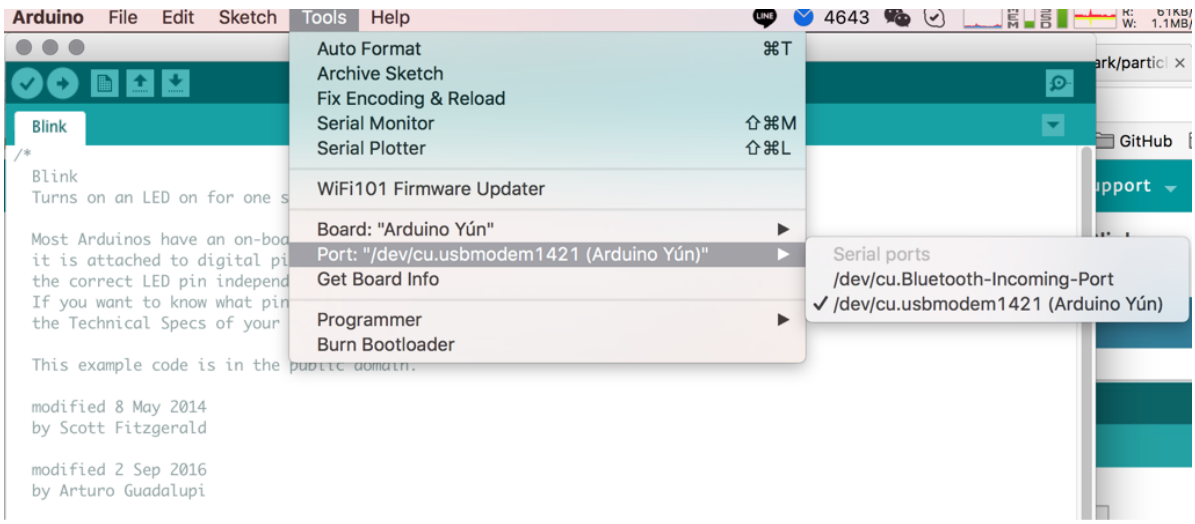
- Connect Arduino Yun to your PC with USB.
- Open your Arduino IDE and find Blink example.



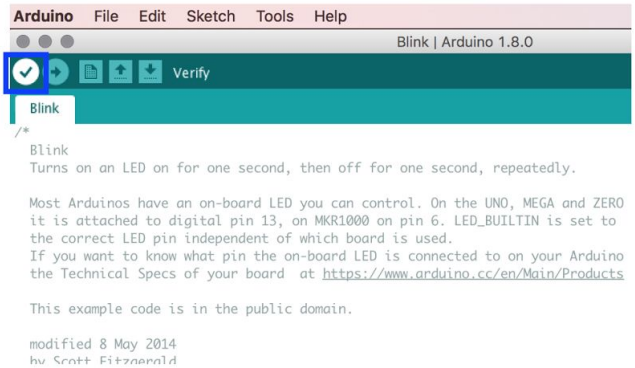
- Configure your board:



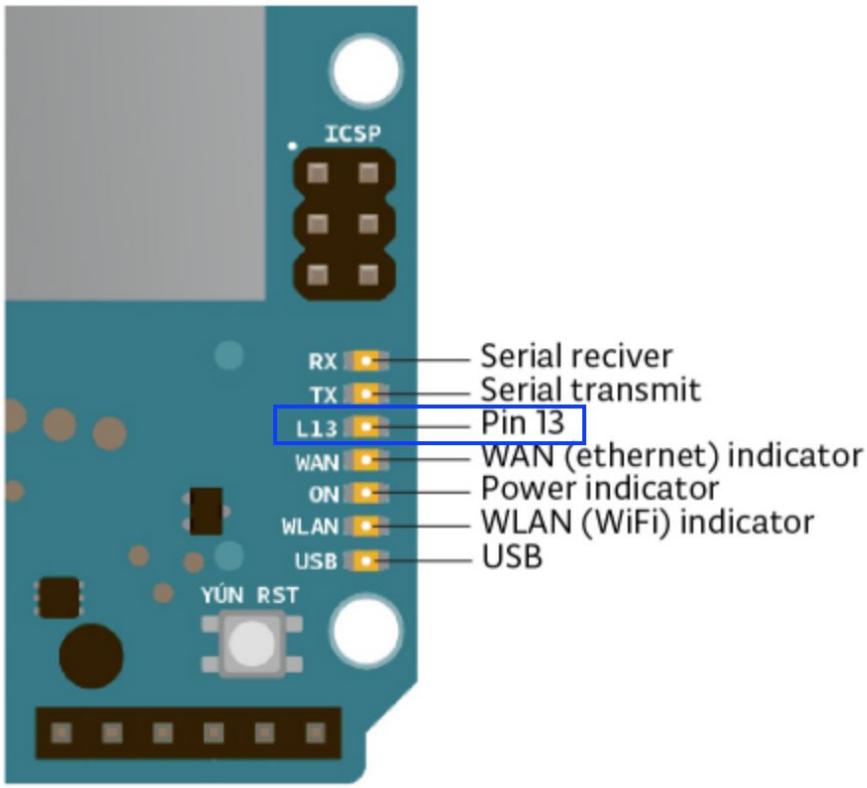
- Configure your port:



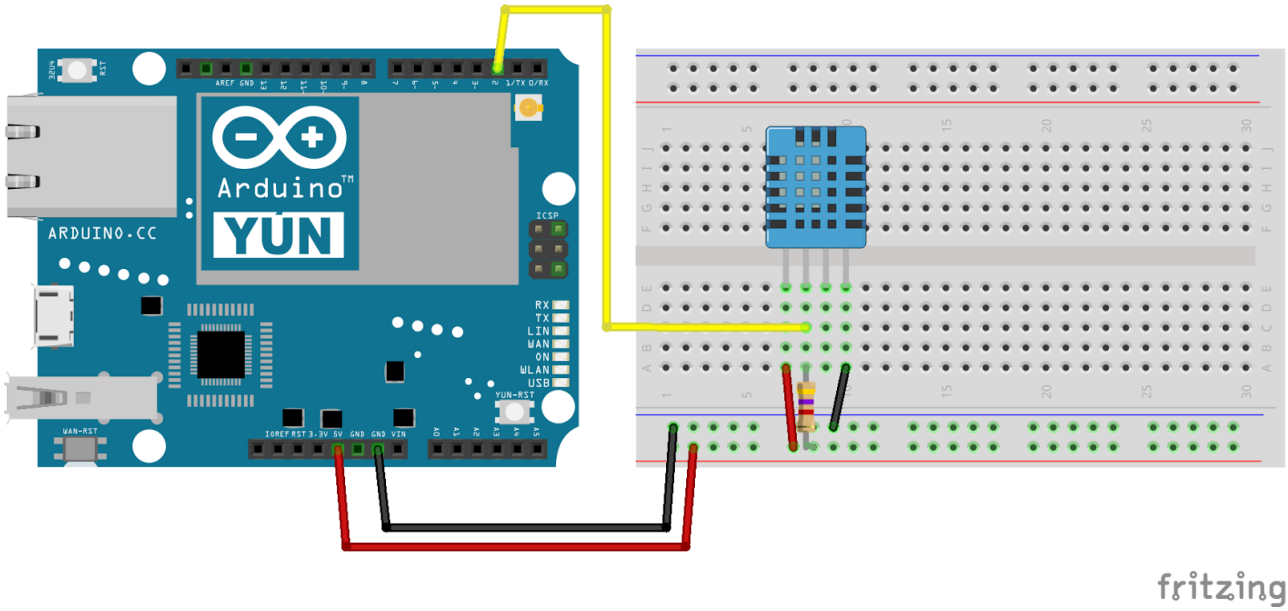
- Verify and upload your Blink example to Arduino Yun



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

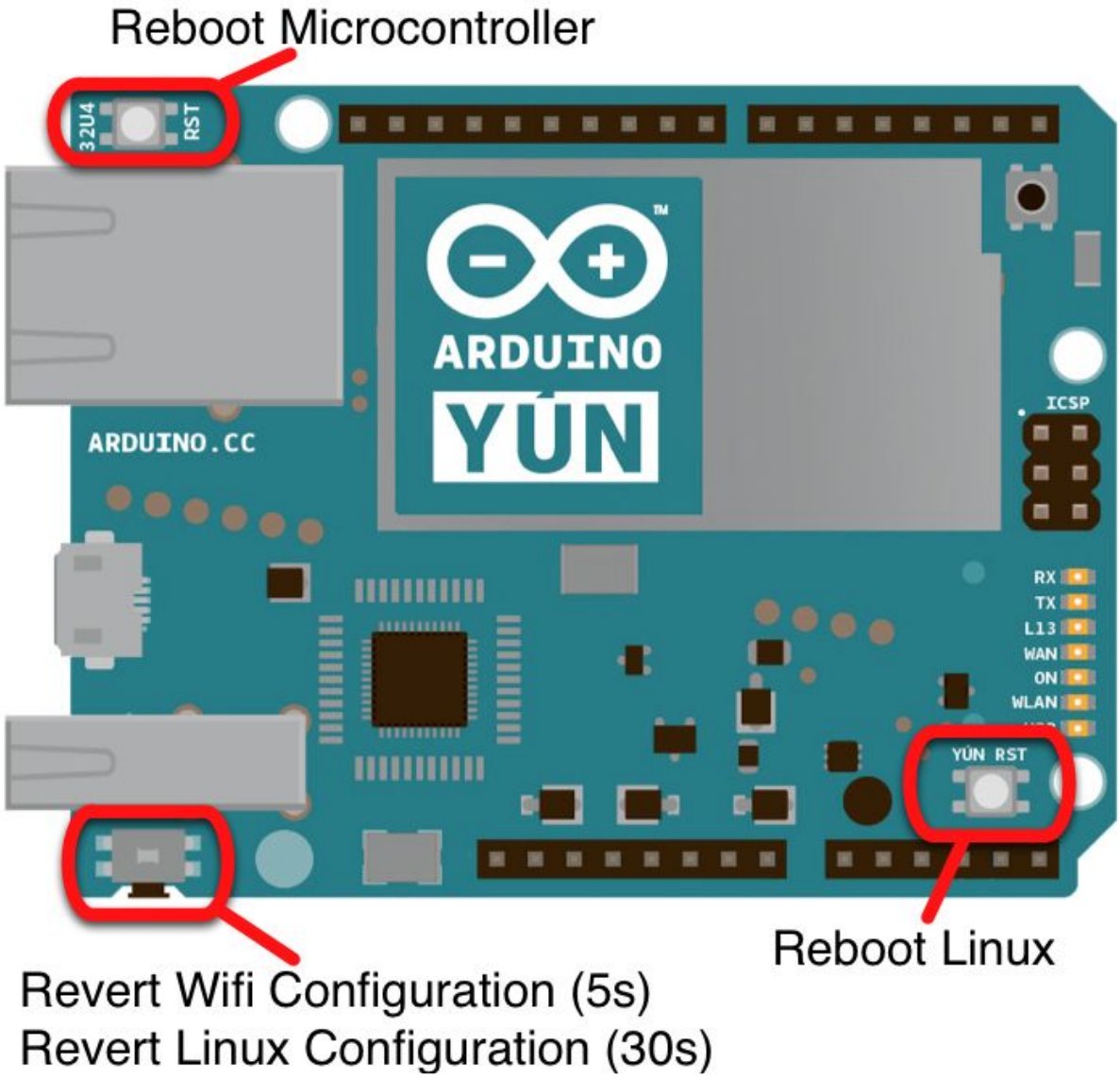


- Connect your DHT11 sensor to “**Ground**”, “**5V**”, “**Pin 2(Digital)**” on Arduino Yun respectively.

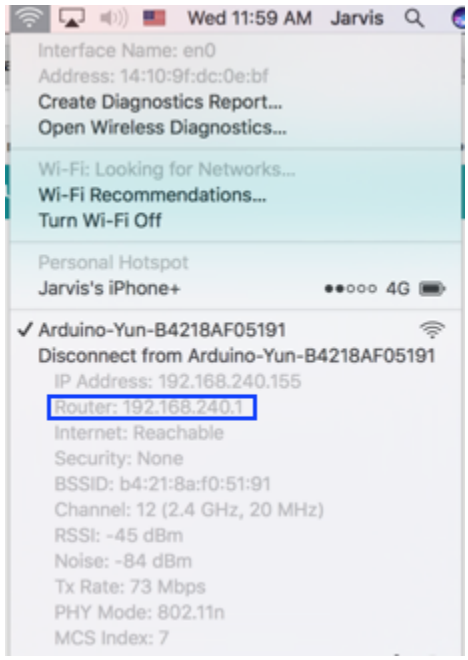


1.3 Get IP address of your Arduino Yun

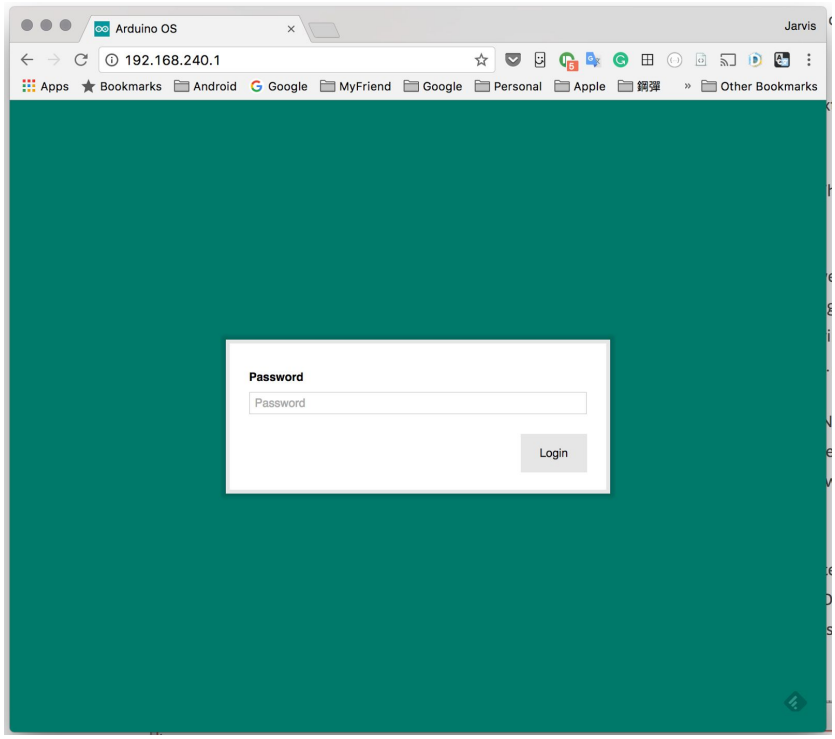
- Press Wifi reset button (need to press 5 seconds)



- 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”.

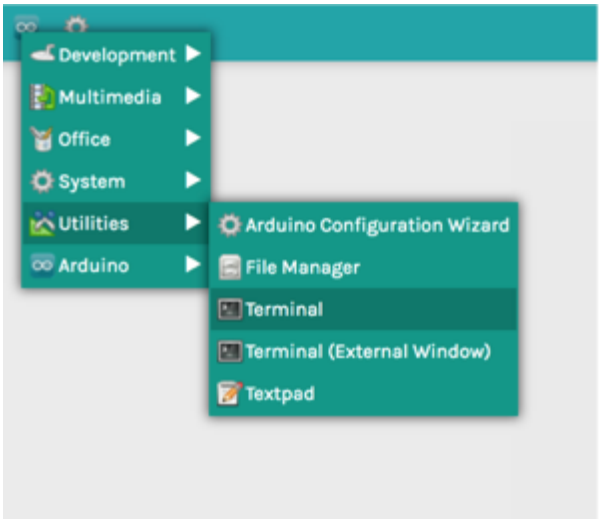


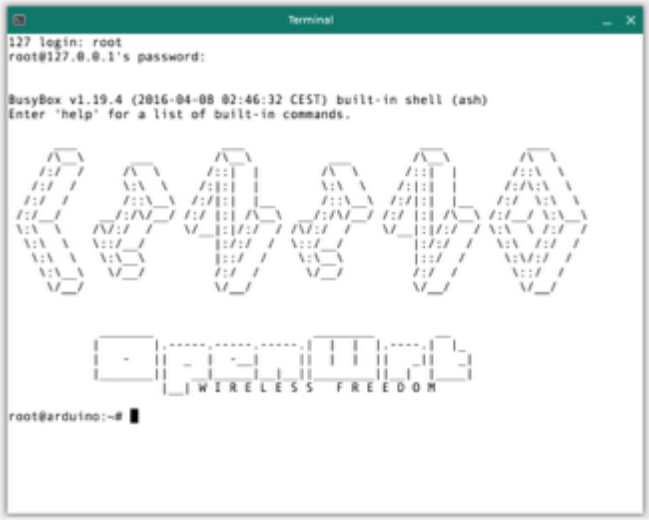
- Complete your Board setting and link your Arduino Yun to the router you are using. Skip “API setting” and click on save to complete.

- 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”.

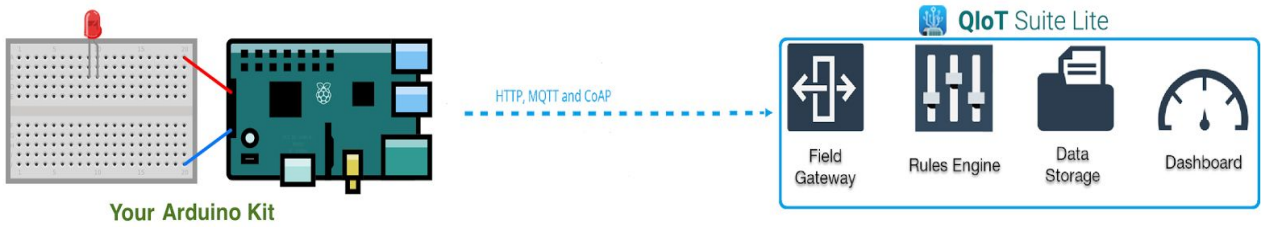




- Install PIP and relatives library by tying in the following commands.
  root@arduino:~# **opkg update**
  root@arduino:~# **opkg install distribute**
  root@arduino:~# **opkg install python-openssl**
  root@arduino:~# **easy_install pip**

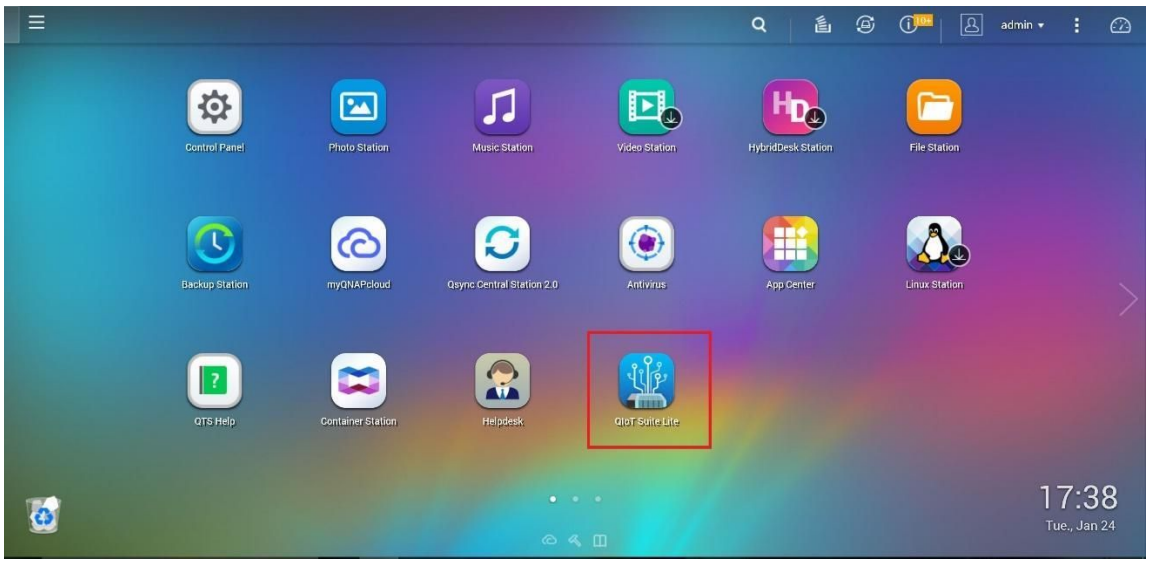
Lesson 2: Create your device in QIoT Suite Lite

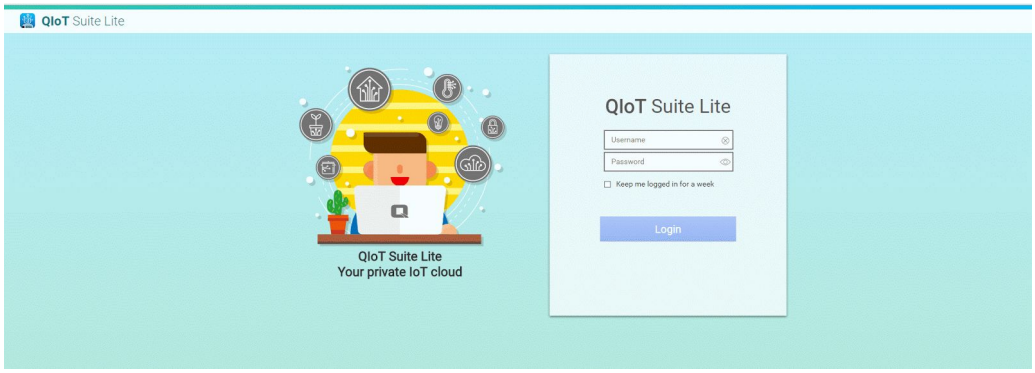
In this lesson, you provision your QNAP QIoT Suite Lite software, and create your first device in QIoT Suite Lite.



2.1 Install QIoT Suite Lite

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





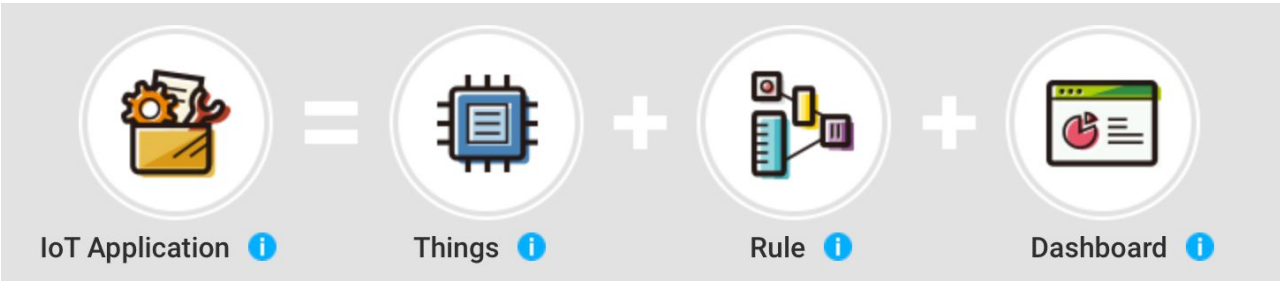
2.2 Create a new IoT application

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.

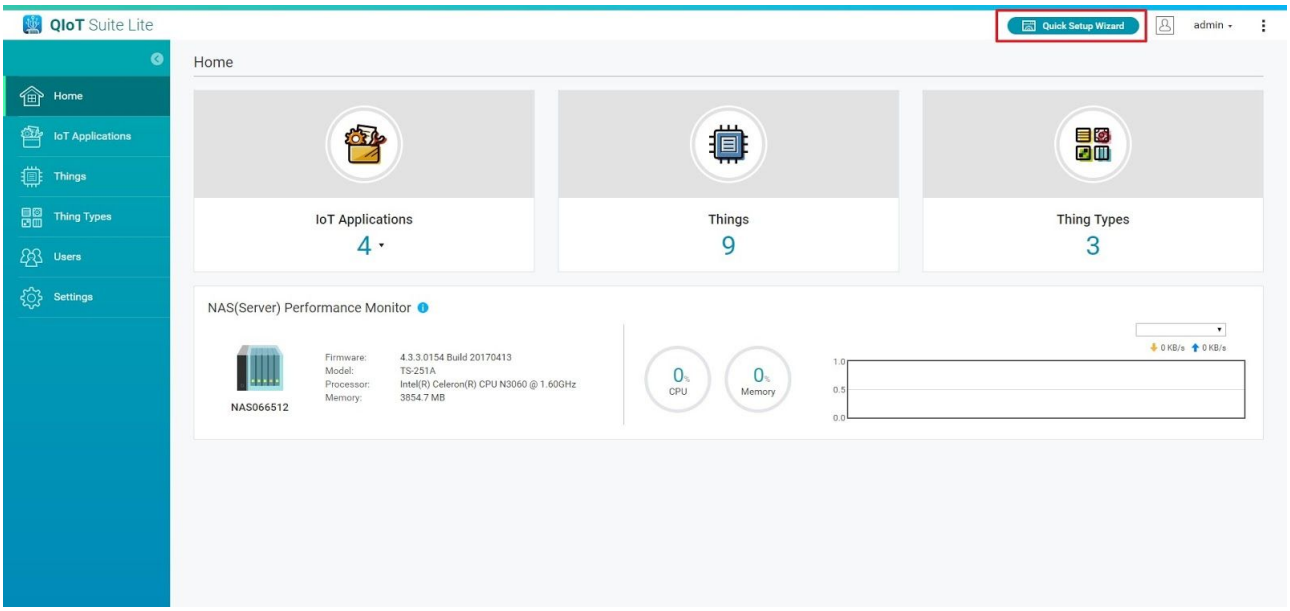
The Rule makes it possible to build IoT applications that gather, process, analyze and act on data generated by connected devices based on business rules you define. A rule can apply to data from one or many devices, and it can take one or many actions in parallel.

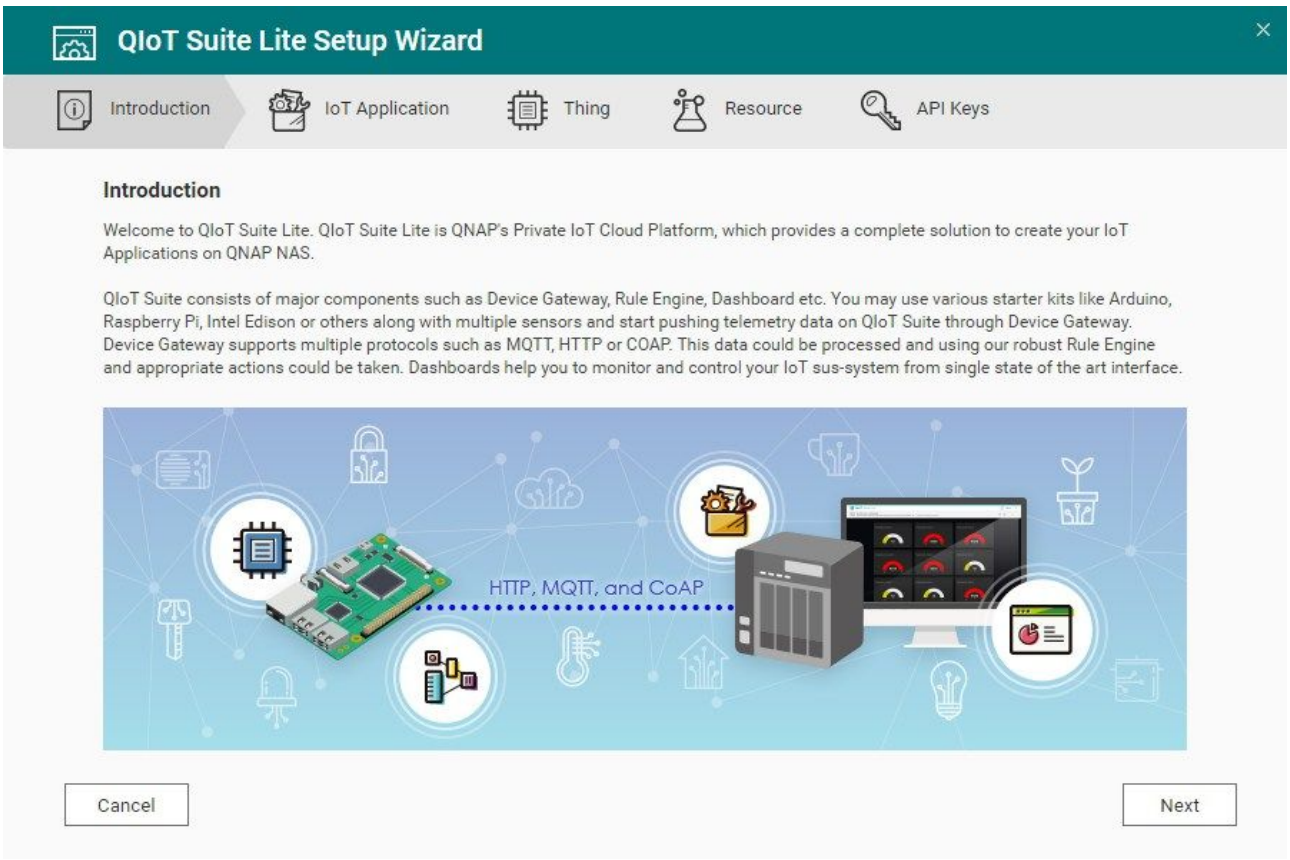
With Dashboard, you can turn your data processing efforts into analytics and reports that provide real-time insights into your business.

All these elements provide user a complete IoT Application environment.

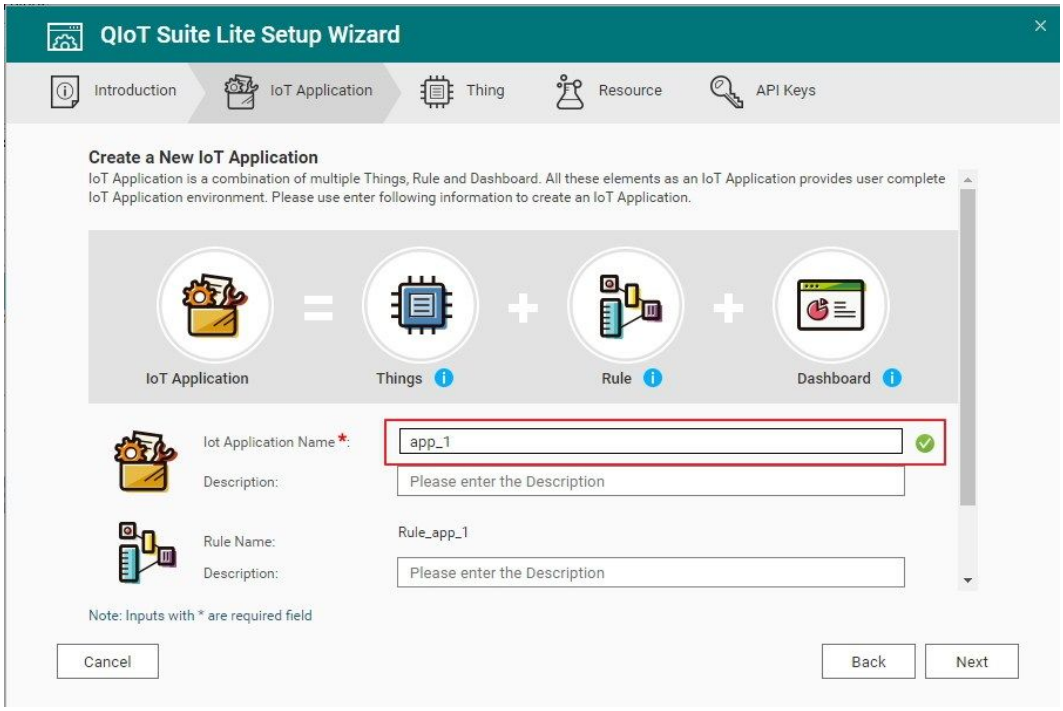


- If this is your first time to use QIoT Suite Lite, QIoT Suite Lite provides a wizard to help you quick setup a IoT application.
- Click “Quick Setup Wizard” to start quick setup wizard . After you read the QIoT Suite Lite introduction in pop window, click Next button.

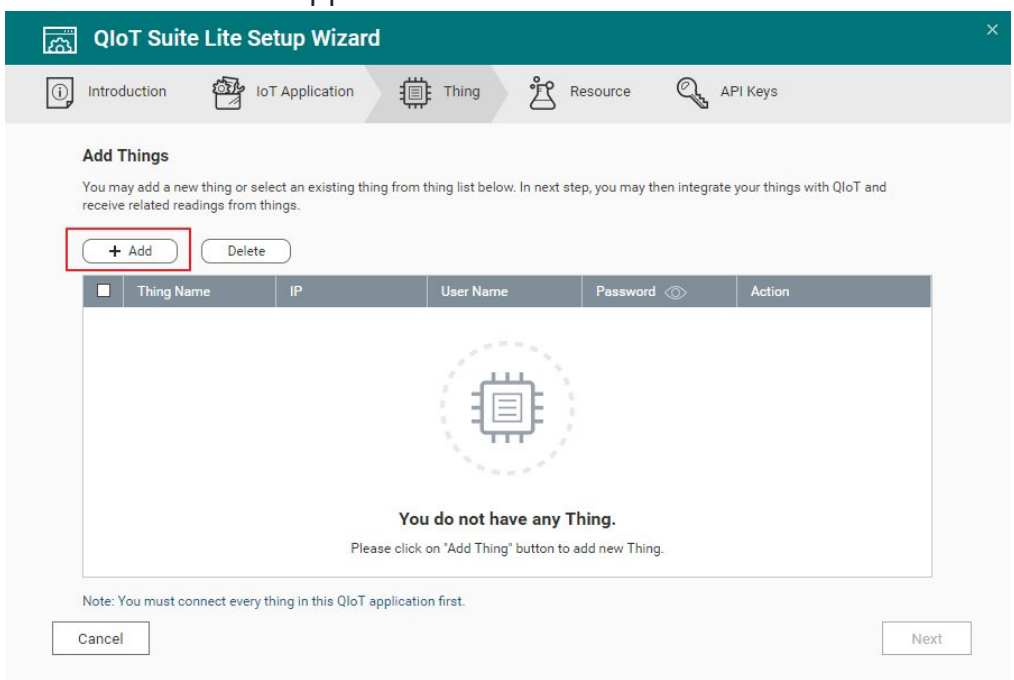

QIoT Suite Lite
Quick Setup Wizard
admin
Home
IoT Applications
Things
Thing Types
Users
Settings
Home
IoT Applications
4
Things
9
Thing Types
3
NAS(Server) Performance Monitor
NAS066512
Firmware: 4.3.3.0154 Build 20170413
Model: TS-251A
Processor: Intel(R) Celeron(R) CPU N3060 @ 1.60GHz
Memory: 3854.7 MB
0%
CPU
0%
Memory
0 KB/s
0 KB/s
1.0
0.5
0.0


QIoT Suite Lite Setup Wizard
Introduction
IoT Application
Thing
Resource
API Keys
Introduction
Welcome to QIoT Suite Lite. QIoT Suite Lite is QNAP's Private IoT Cloud Platform, which provides a complete solution to create your IoT Applications on QNAP NAS.
QIoT Suite consists of major components such as Device Gateway, Rule Engine, Dashboard etc. You may use various starter kits like Arduino, Raspberry Pi, Intel Edison or others along with multiple sensors and start pushing telemetry data on QIoT Suite through Device Gateway. Device Gateway supports multiple protocols such as MQTT, HTTP or COAP. This data could be processed and using our robust Rule Engine and appropriate actions could be taken. Dashboards help you to monitor and control your IoT sus-system from single state of the art interface.
HTTP, MQTT, and CoAP
Cancel
Next

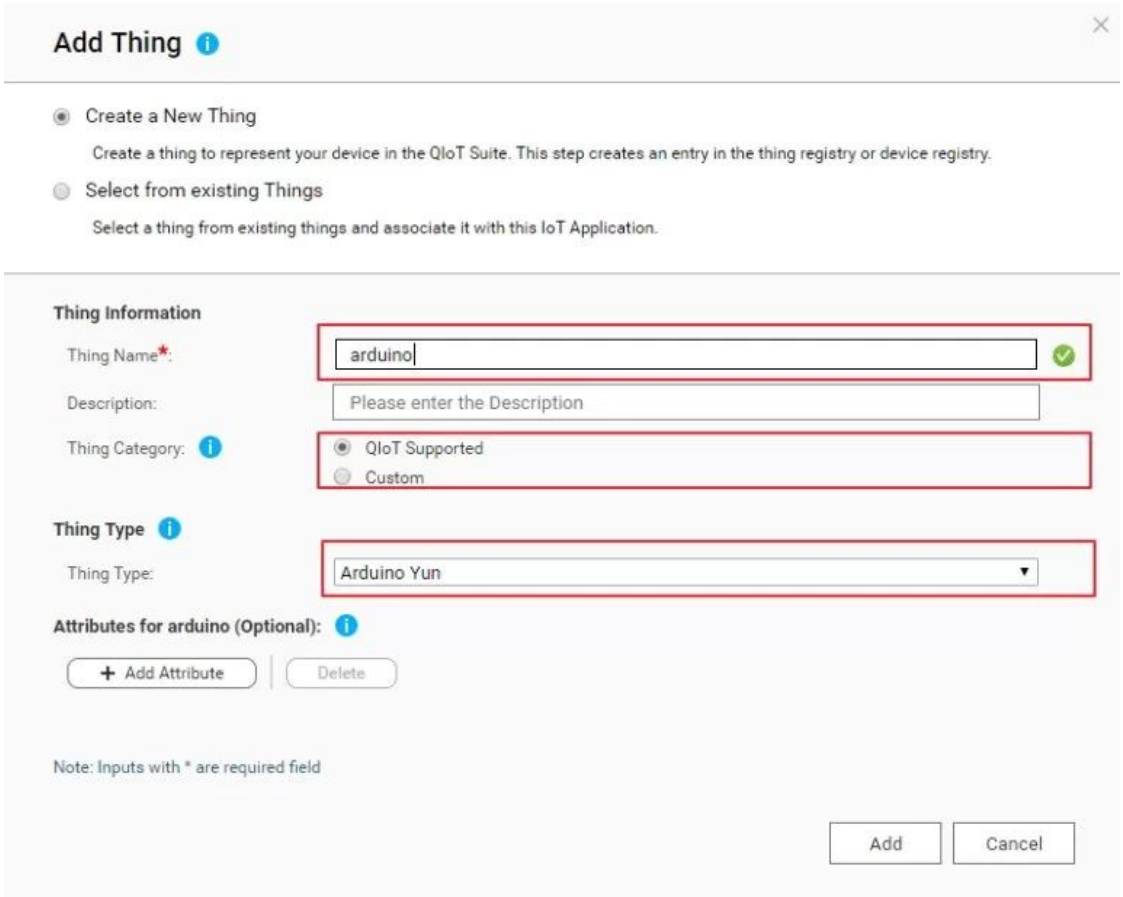
- Create a New IoT application
  - Enter IoT Application name, e.g., app_1.
  - Rule name and Dashboard name will be generated automatically based on the name of IoT Application you fill in.
  - Click “Next” to complete create a new IoT application



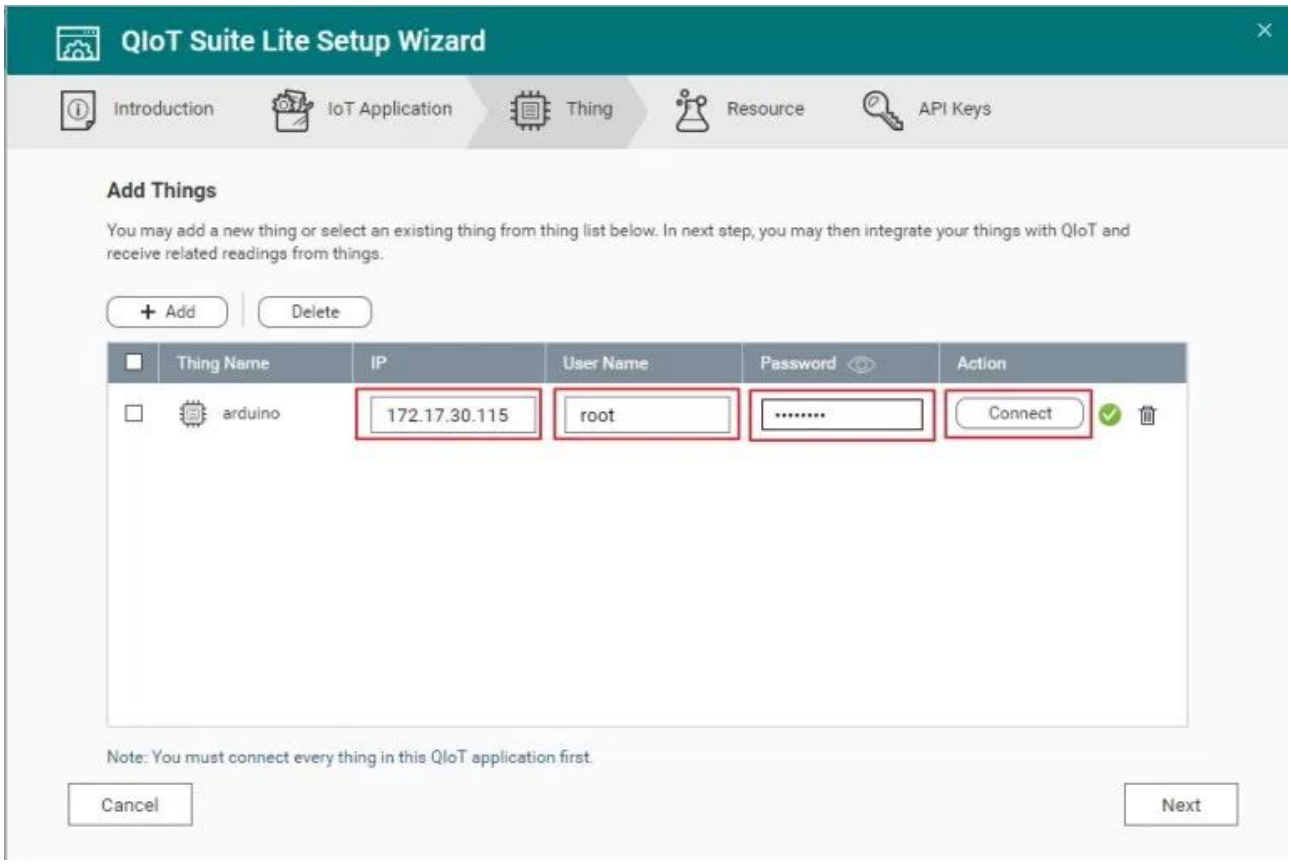
- Click “+Add” to add this application’s device.



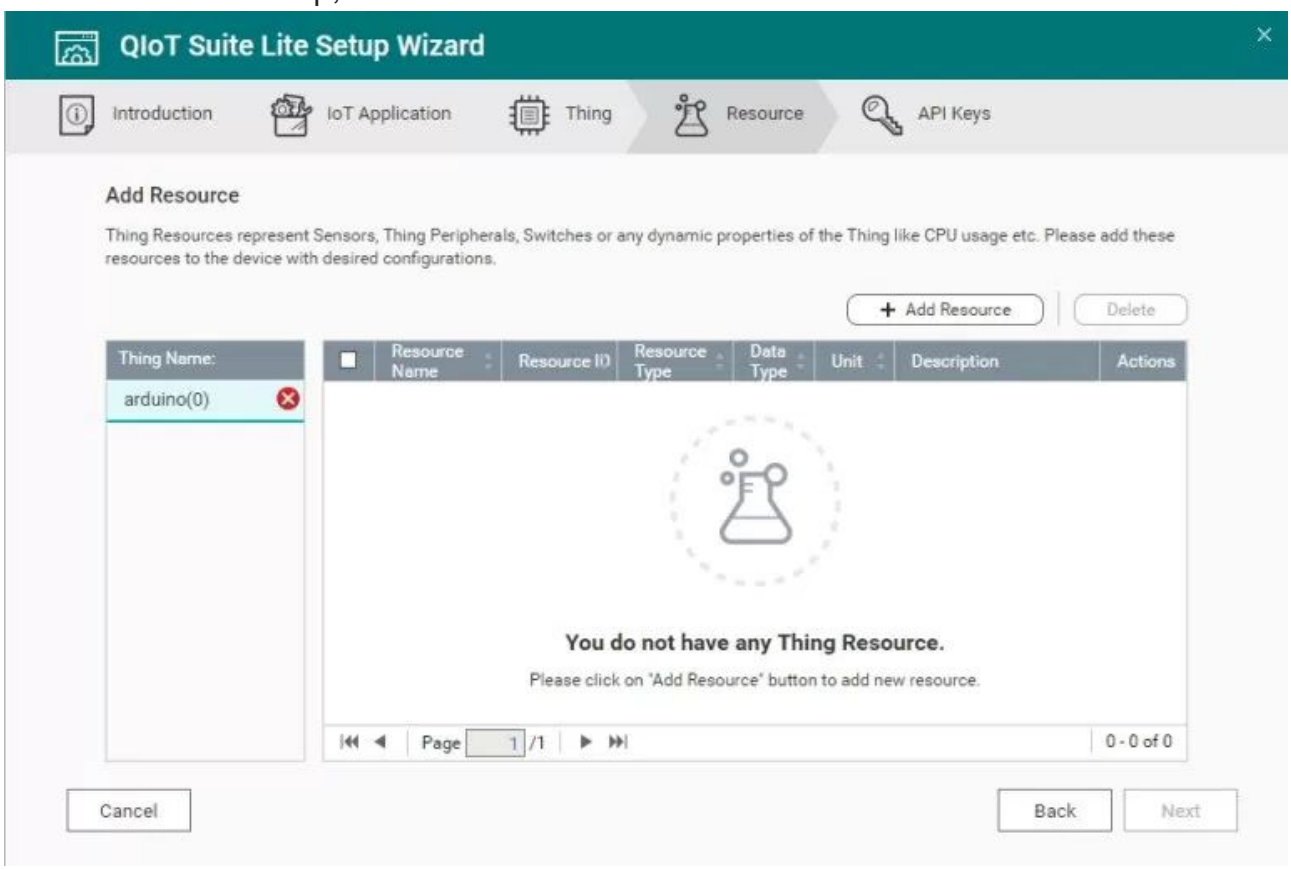
- After click “+Add ”,In “Add Thing” pop window:
  - Set device’s name (e.g., arduino).
  - QIot Suite Lite now support Arduino Yun,Raspberry Pi,and Edison ,so you can select “QIoT Supported” thing category and select “Thing Type” is “Raspberry Pi”
  - (optional) add attribute to device’s detail information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



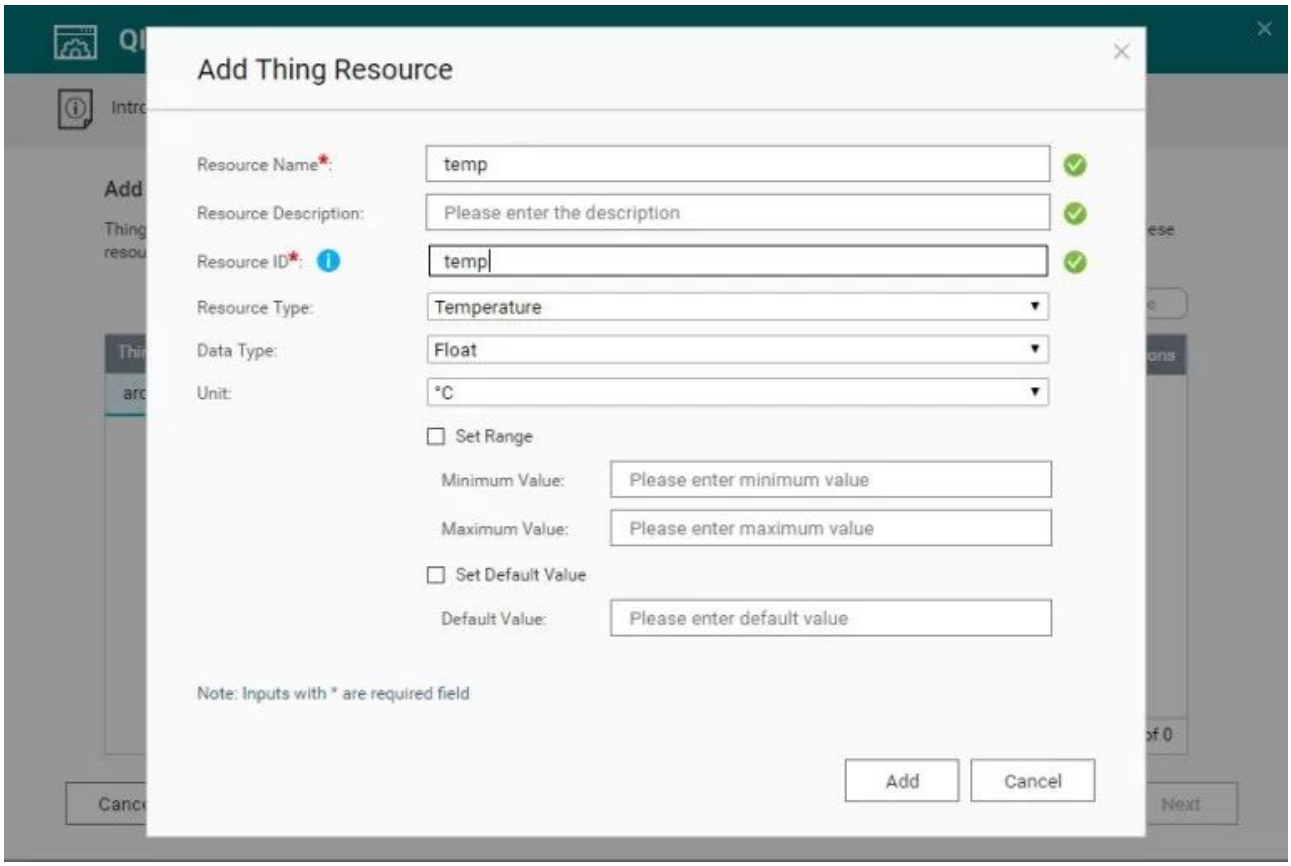
- Please provide IP address, user name, and password of your device, then click “Connect” .After waiting test connection between your device and QIot Suite Lite success, you could click “Next” to next step.



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

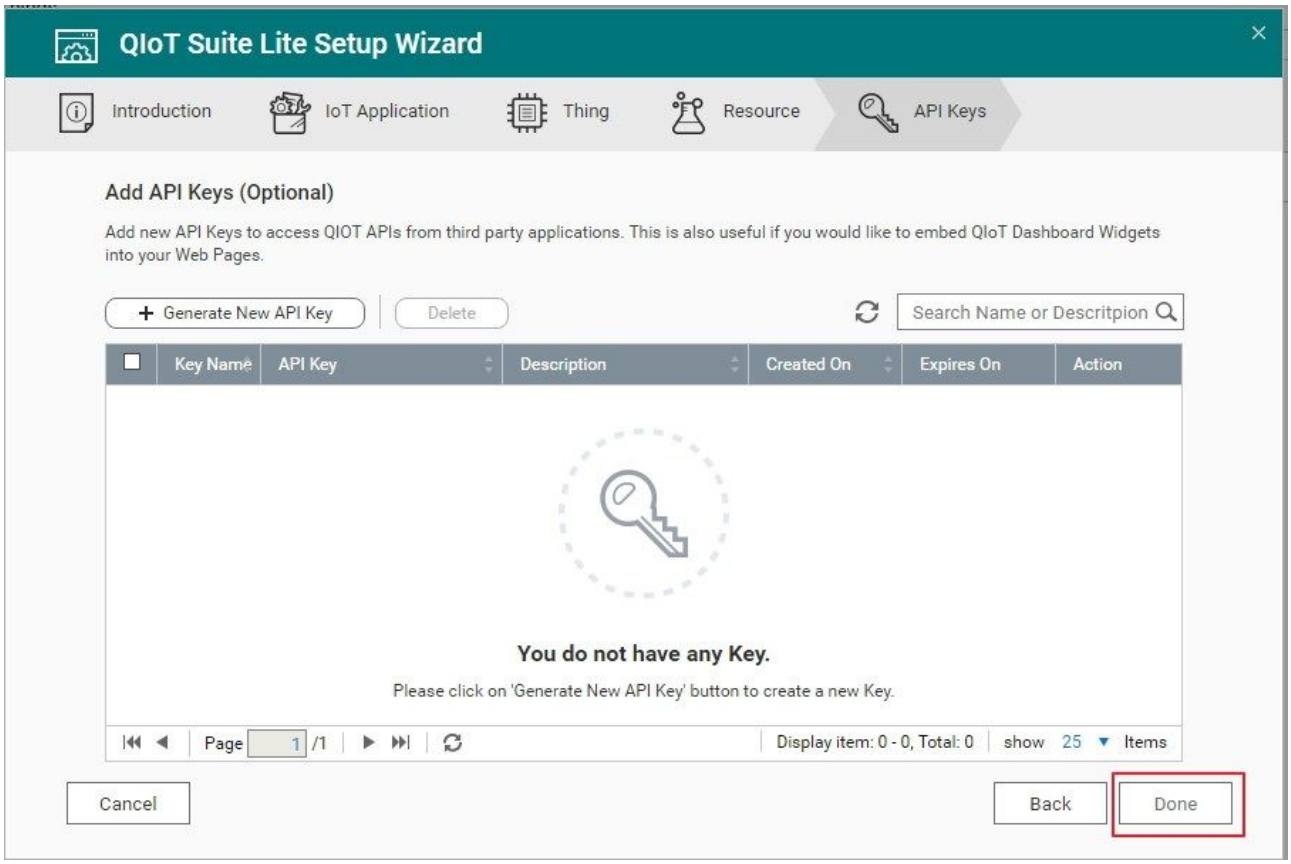


- After click “Add Resource”,In “Add Thing Resource” pop window:
    - Set resource’s name (e.g., temp).
    - Set resource’s id.Resource id will be used to create a topic in the QIoT broker.This ID should be unique for the device and no duplicates should be allowed for the same device
    - And set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

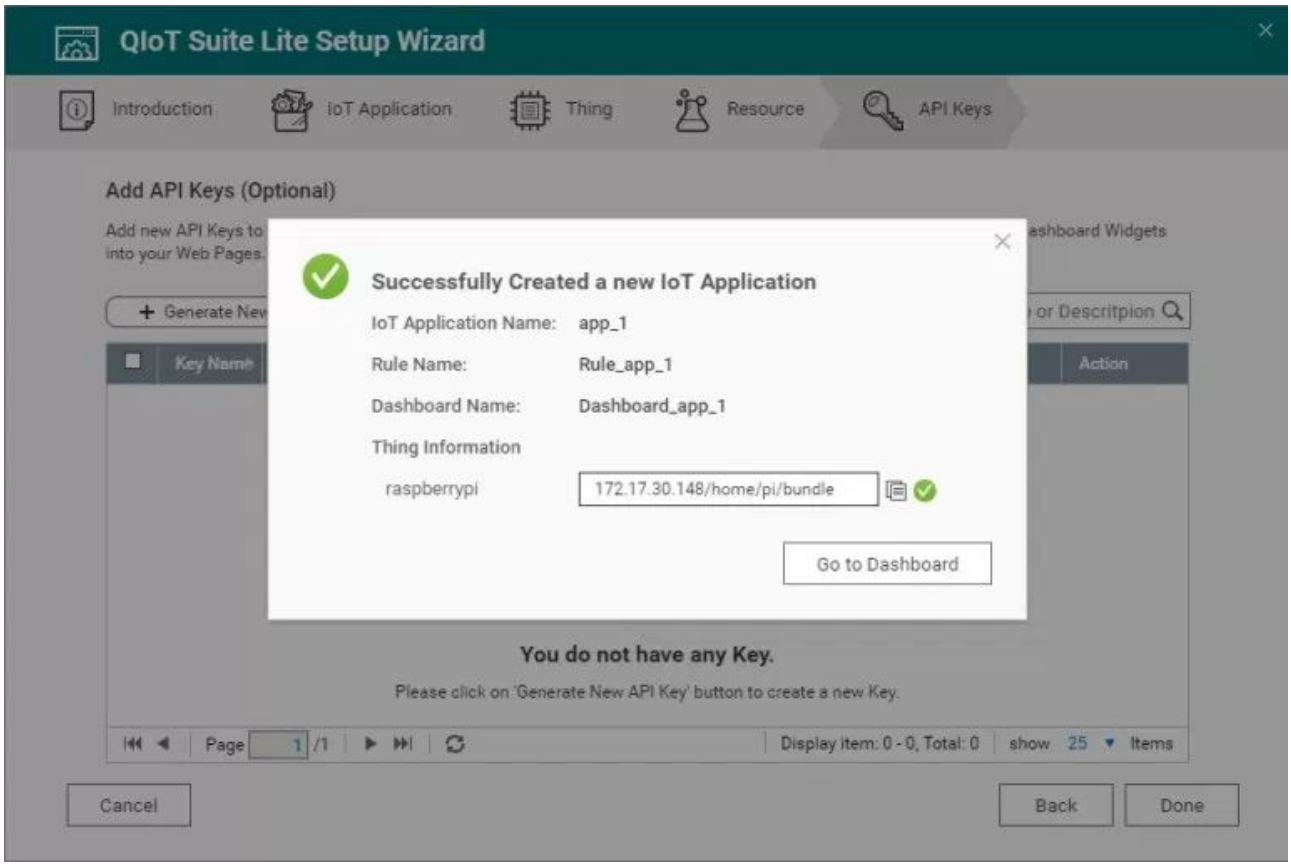


- Click Next after you add all the resources (sensors) on your device.

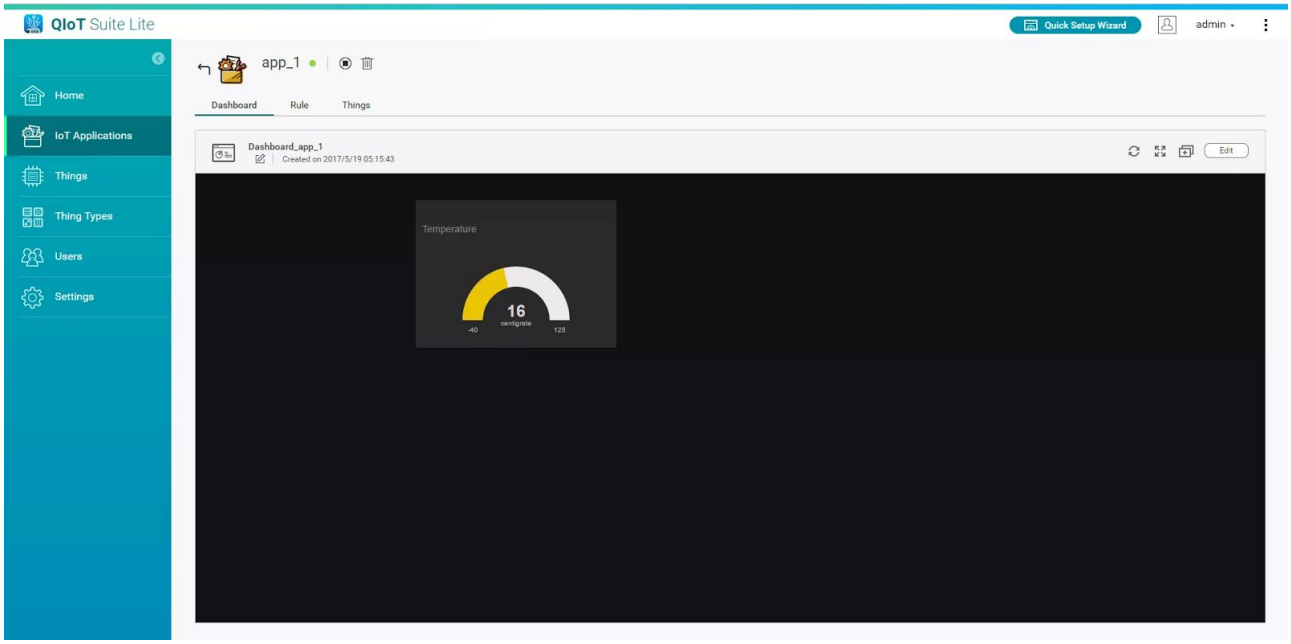
- In “API Keys” step :
    - If you would like to embed QIoT dashboard widgets into your web pages or access QIoT APIs from third party applications.You could click “generate new API Key” to create API key or click “done” start deploy sample code to your device.



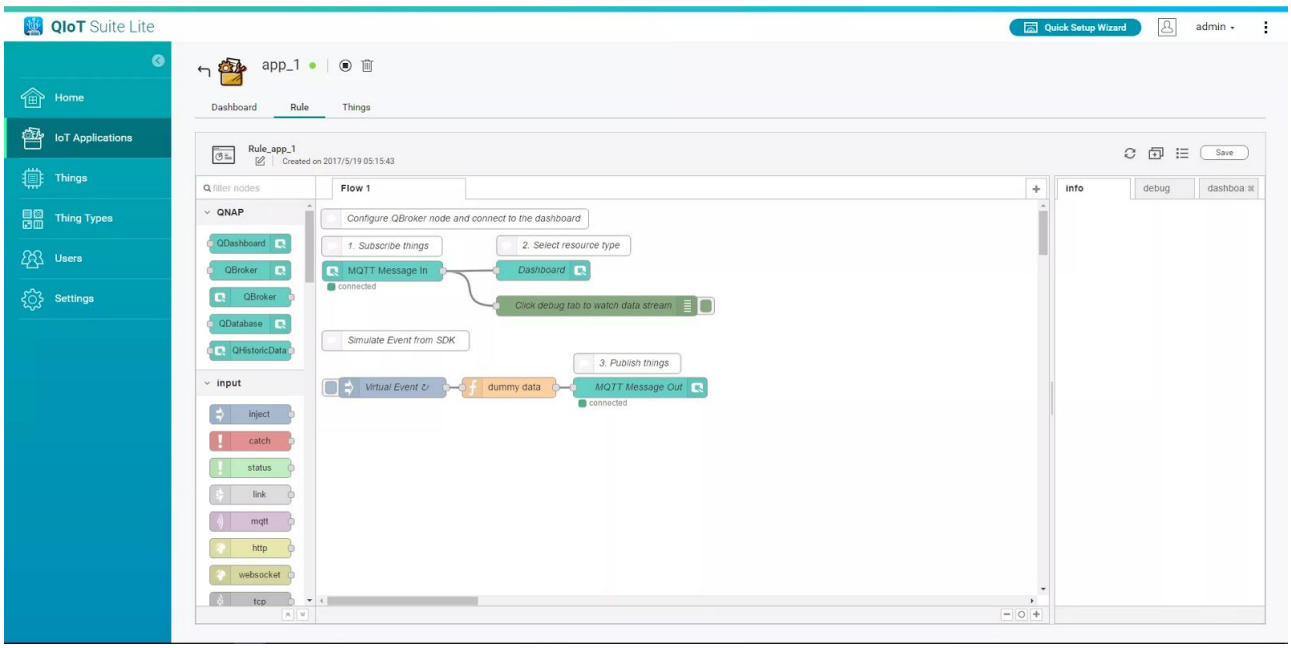
- For a while, sample codes and related files (certificate, resource information) have been upload to the specified path on your device.



- 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.

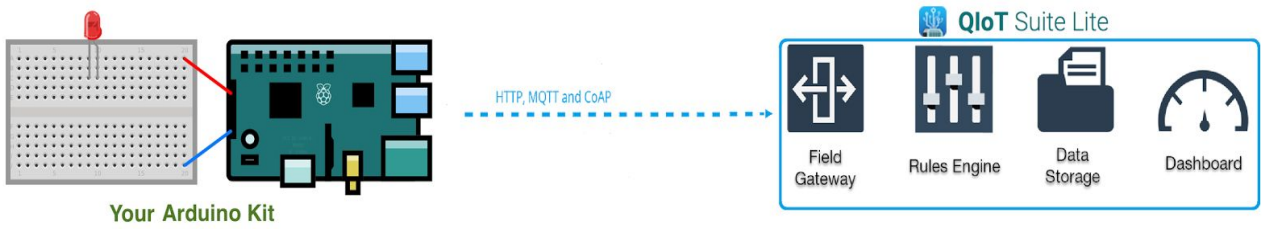


- Select “Rule” tab, you could to define the flow or rule about how to process the data sent from the device, and how to present in dashboard.



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 Arduino Yun 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).

- Install sample code dependency,enter command as following
  root@arduino:~# **cd /home/{{user}}/bundle**
  root@arduino:~/home/root/bundle# **pip install paho-mqtt**

- 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.

```
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_i("temp", str(random.randint(0, 41)))
    """
        or publish by resource topic
        TODO: you could replace "qiot/things/admin/edison/temp" by any Topic form QI
```

- Run the sample application.

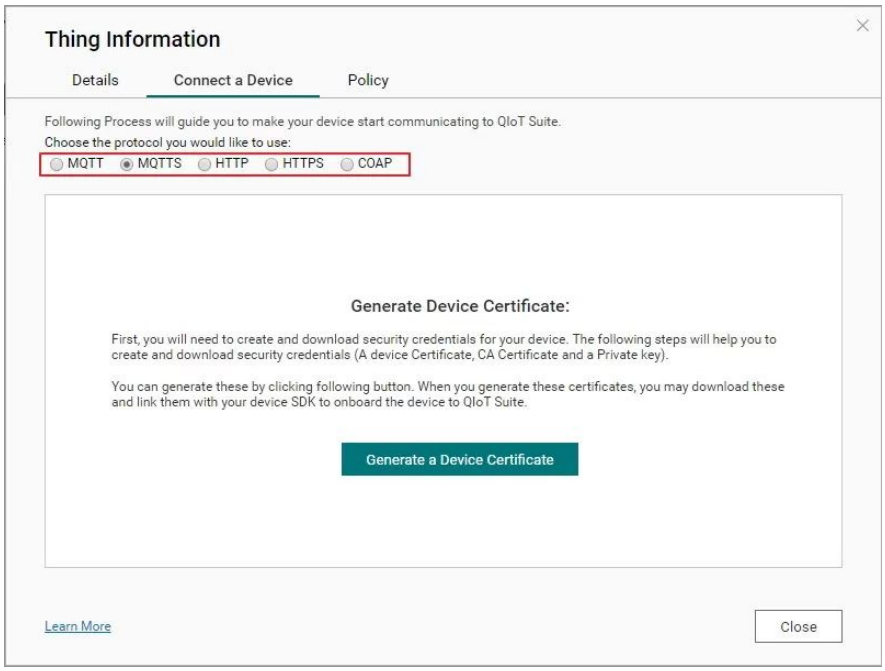
```
root@arduino:~/home/root/bundle# pyton mqtt.py
```

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

```
root@arduino:/home/root/bundle# python mqtt.py
new mqtt protocal
CLIENT_CERT path :/home/root/bundle/ssl/3048642017-05-16_10-31-51_certificate.pem
PRIVATE_CERT exists or not :True
Use MQTTS
USER_NAME : 0cf76869-43d5-49ec-95a6-197e74798ed8 USER_PASS : r:6723da30e376545f341ceb10a0d13b05
finish setup
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":39}
connect ready
connection ready
client ready
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":31}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":40}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":39}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":9}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":3}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":31}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":35}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":34}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":8}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":22}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":20}
NOW TOPIC_NAME :qiot/things/admin/arduino/temp MESSAGE : {"value":5}
```

3.2 Another protocol

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



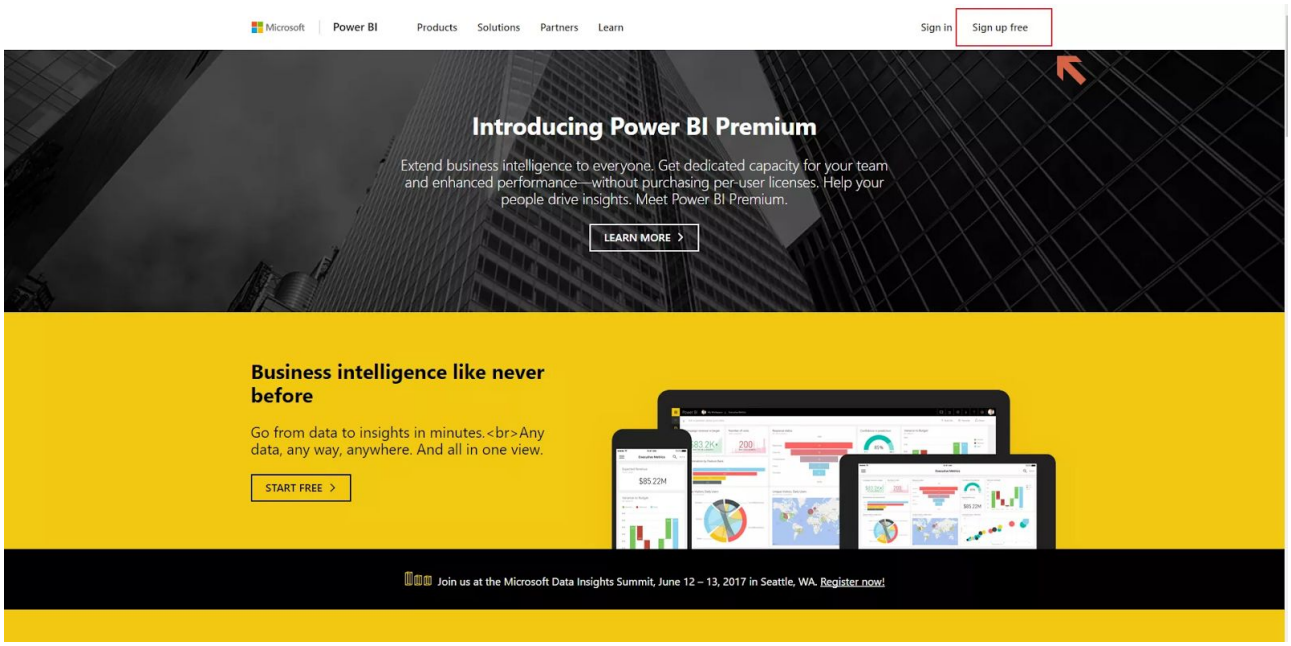
- SSH to your device , and input command as following.

```
root@arduino:~/home/root/bundle# cd /home/{{user}}/bundle

// mqtt(dont' need certificate,just put JSON file to "res" folder):
root@arduino:~/home/root/bundle# pip install paho-mqtt
root@arduino:~/home/root/bundle# python mqtt.py

// http
root@arduino:~/home/root/bundle# pip install requests
root@arduino:~/home/root/bundle# python http.py

// https
root@arduino:~/home/root/bundle# pip install requests
root@arduino:~/home/root/bundle# python https.py

// coap
root@arduino:~/home/root/bundle# pip install coapthon
root@arduino:~/home/root/bundle# python coap.py
```

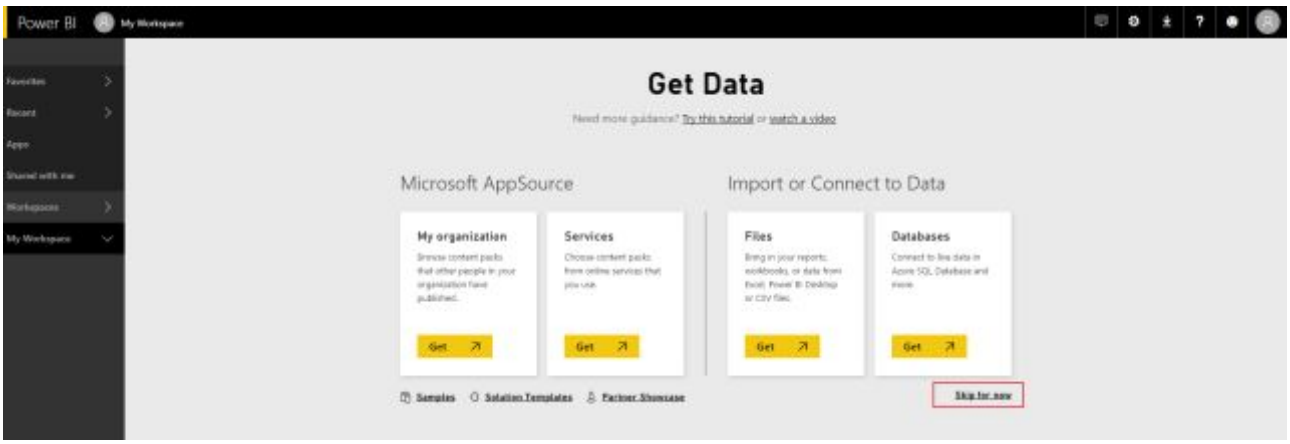
Lesson 4: Integrate Power BI

4.1 Get your first Power BI account

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

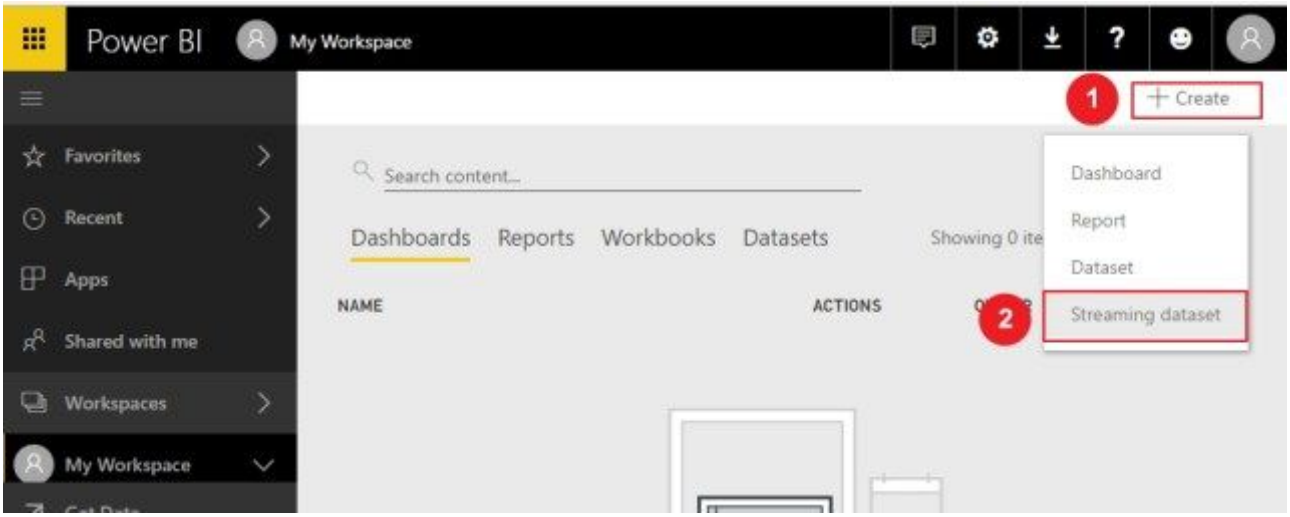


- After a sequence of registration, the page will lead you to below page,you can press “My workspace”, and “skip for now” button appear. You could click “skip for now “ to start create dataset.

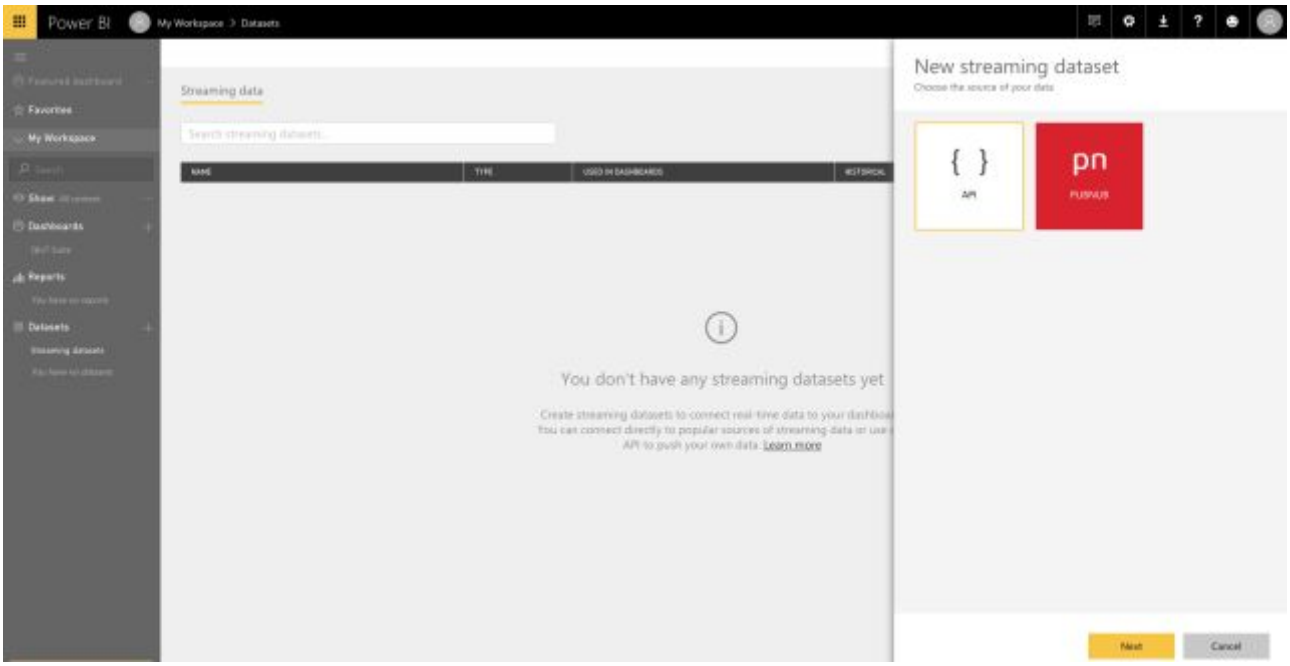


4.2 Setup your streaming dataset API

- Create “Datasets”
  - Click “Create” in scrren upper right corner
  - And then click “Streming 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. QIot 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 *

temp

Values from stream *

temp	Number
max	Number
mix	Number
Enter a new value name	Text

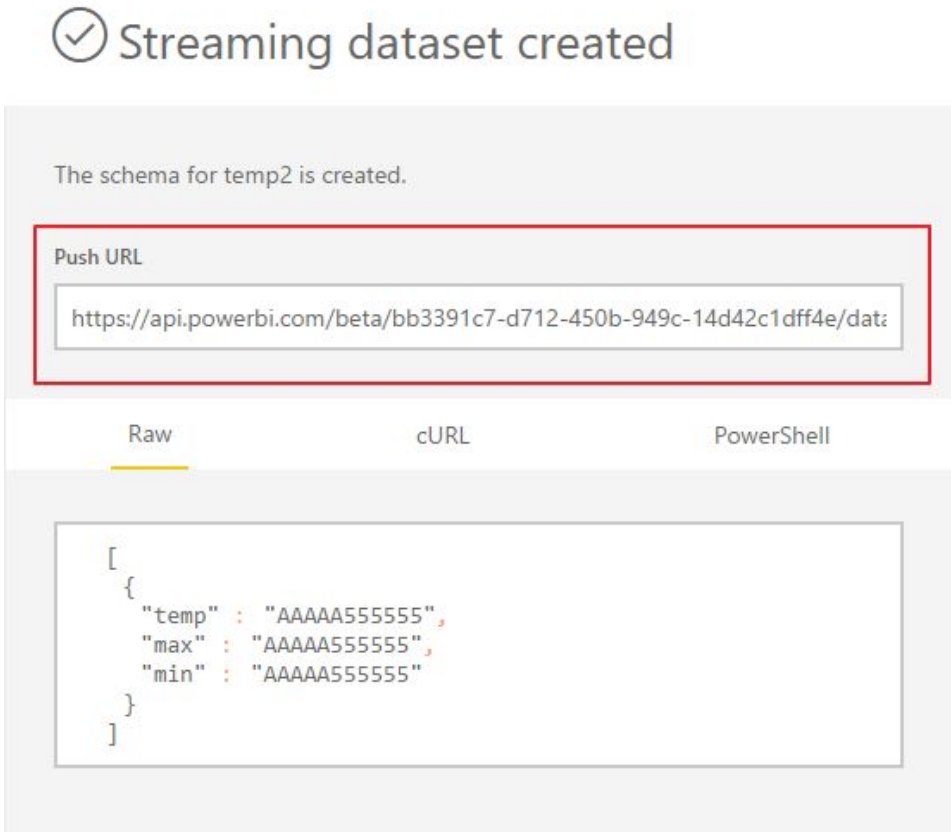
```
[
  {
    "temp" : 98.6,
    "max" : 98.6,
    "mix" : 98.6
  }
]
```

Historic data analysis

On

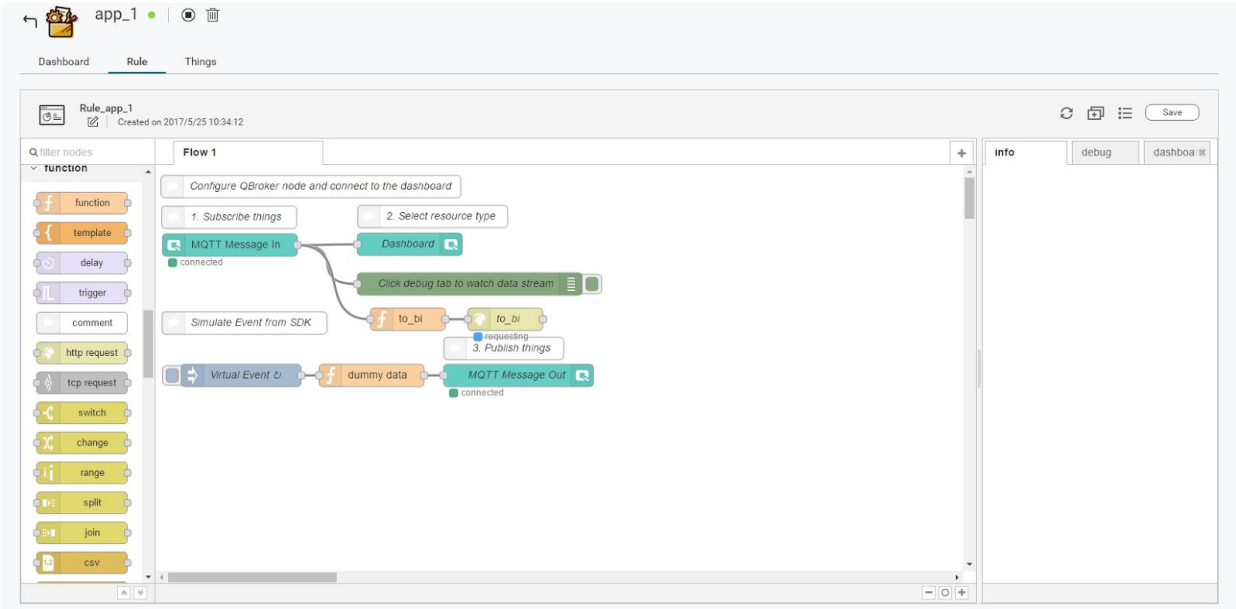
Back Done Cancel

- 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.



4.3 Configure Node-RED's nodes in IoT application

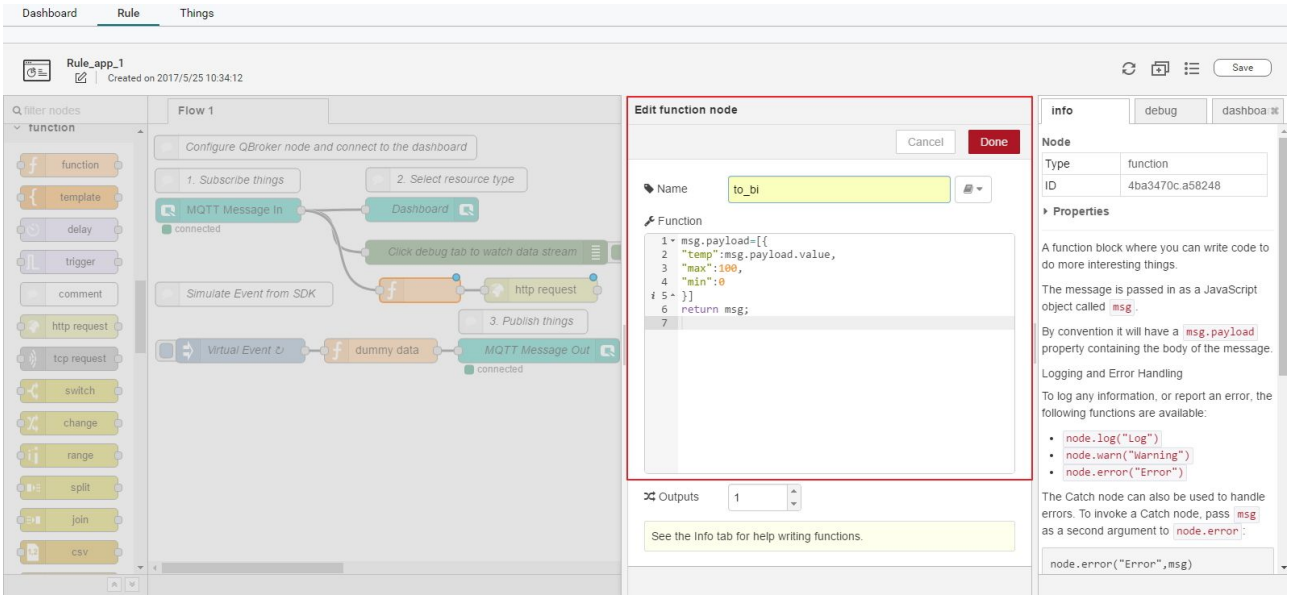
- Create IoT application in QIoT suite.
- The following one is your first node-red flow, and then you can start to create your own IoT flow. more node-red information can be found in "Node-Red".



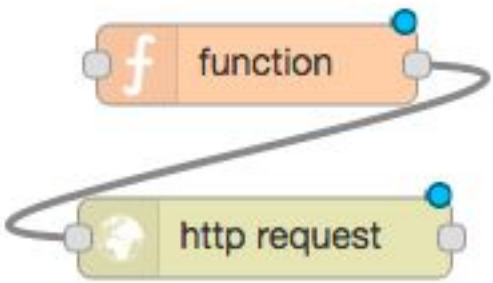
- Before you start to push live data to Power BI.
- We need a “function” node to convert IoT data to streaming data dataset. Here you can replace msg.payload to your JSON dataset.

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

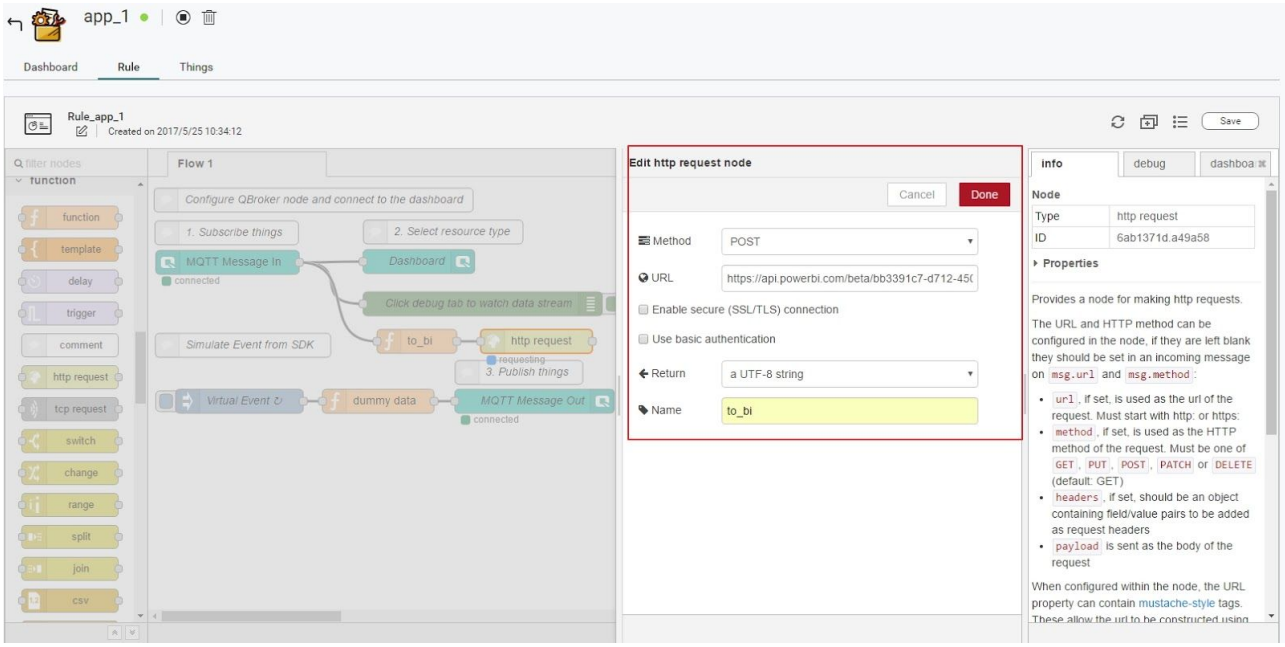
```
msg.payload=[{
        "temp":msg.payload.value,
        "max":100,
        "min":0
}]
return msg;
```



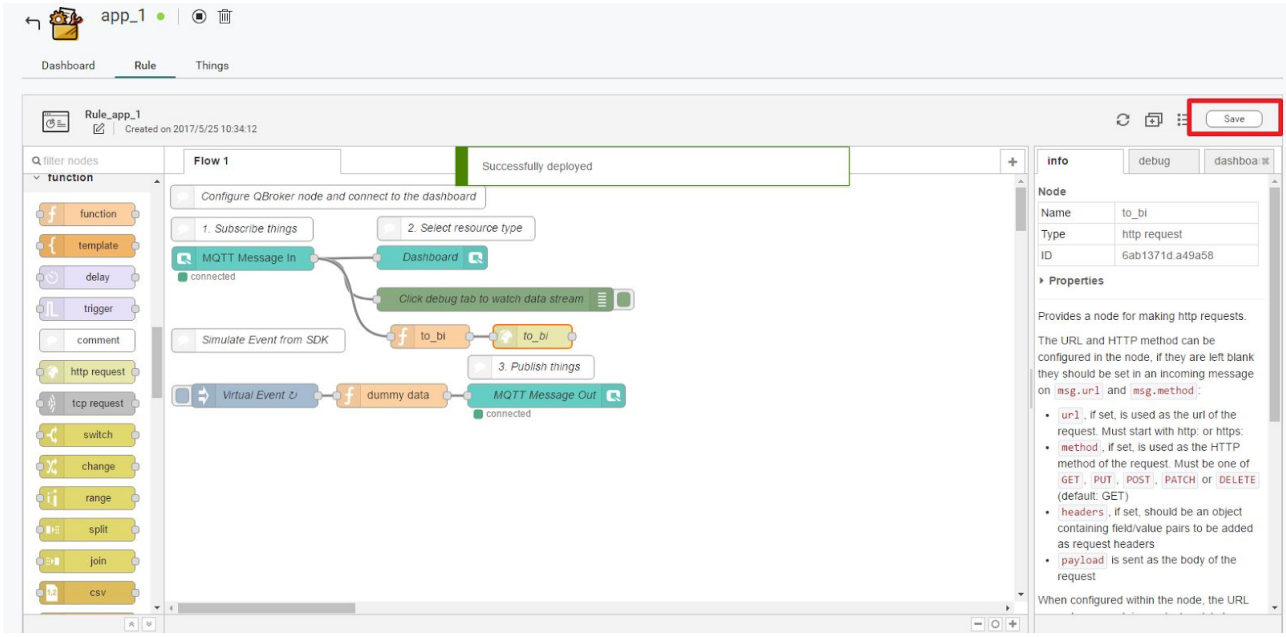
- We need a “http request” node to help us to push live data to Power BI. Just drop and drag “http request” node and connect to tail of “function“ node.



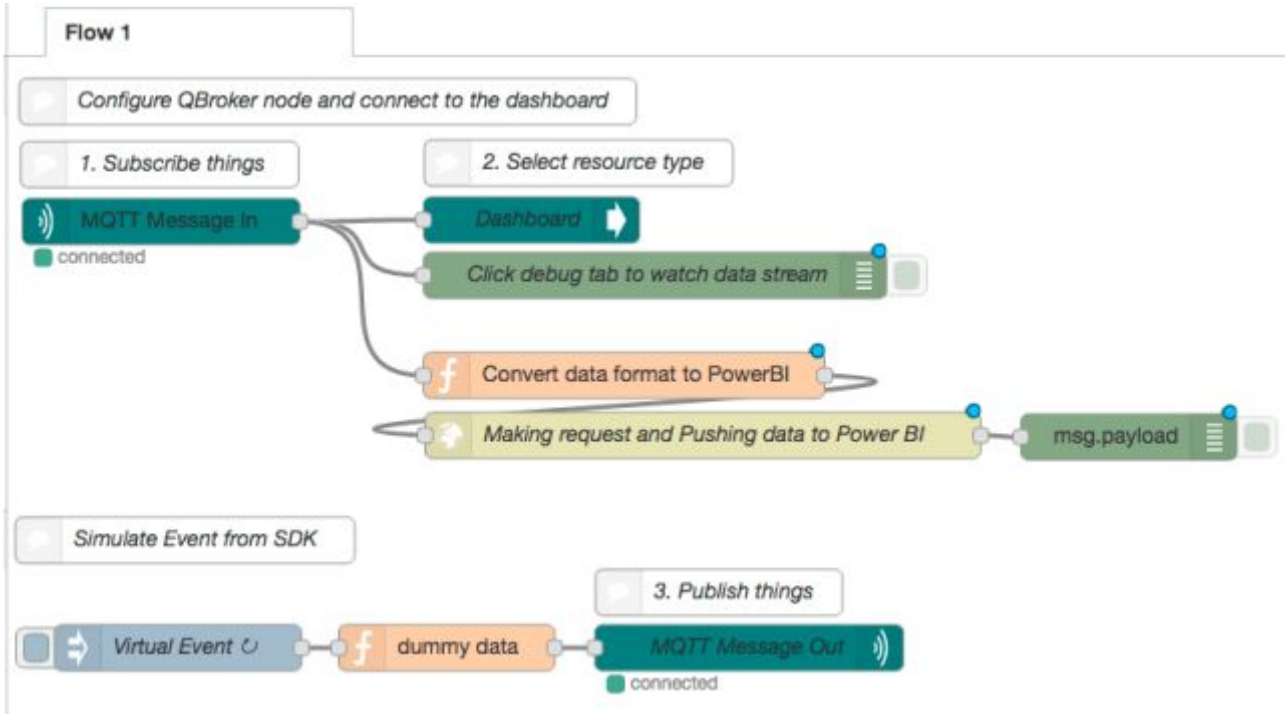
- Double click http request node, copy and paste REST API URL that you got from Power BI console, and set http method to POST.



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

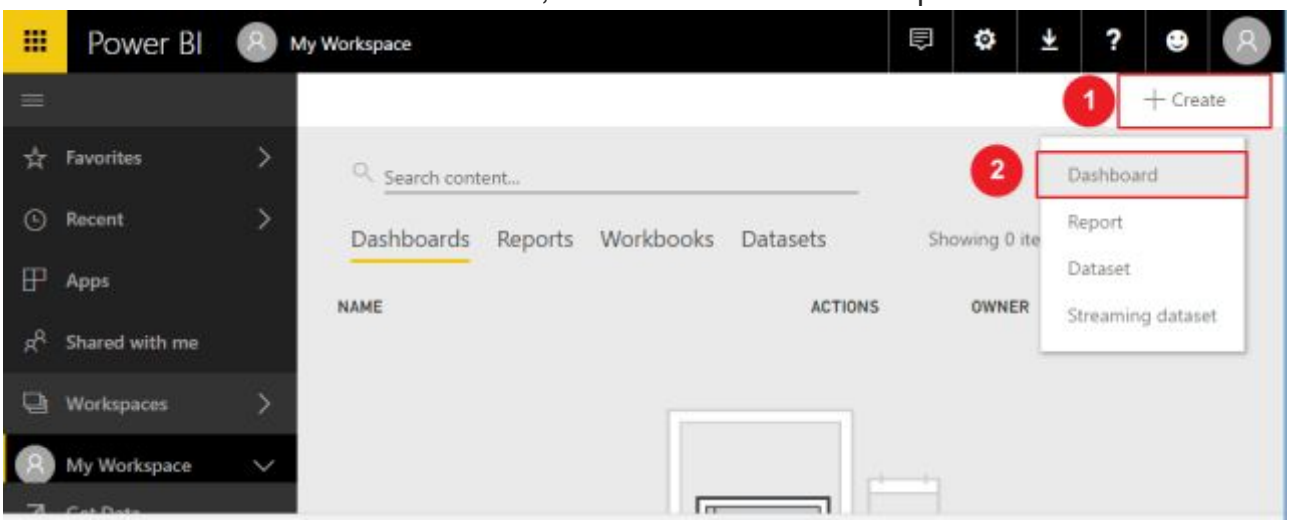


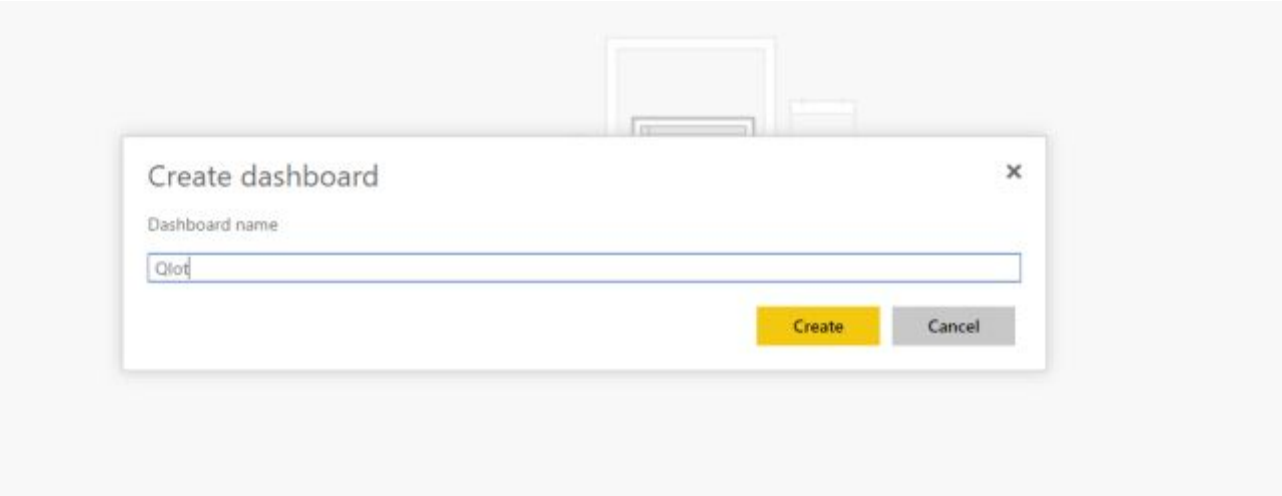
- Finally, your node-red flow will look like below one.



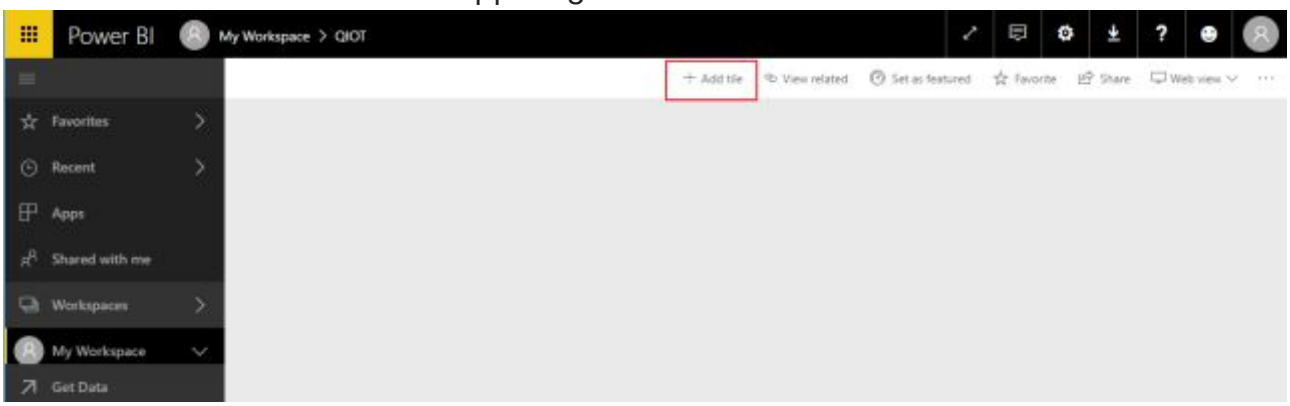
4.4 Add tile to display real-time data

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

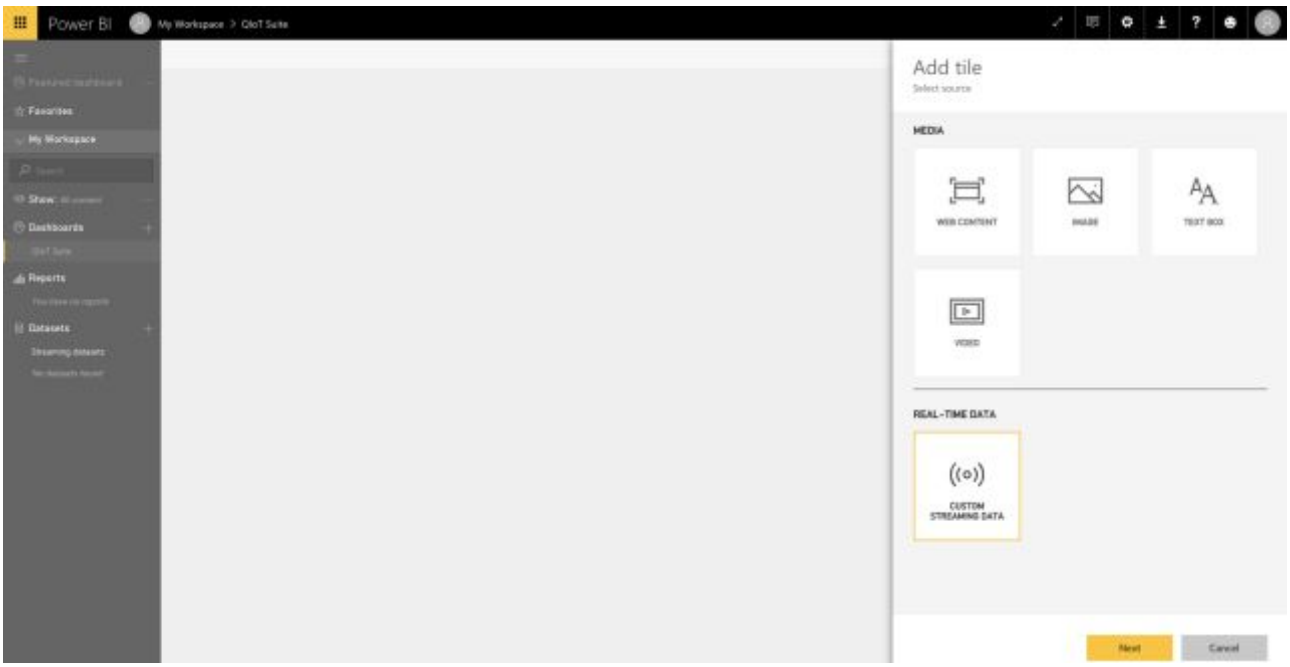




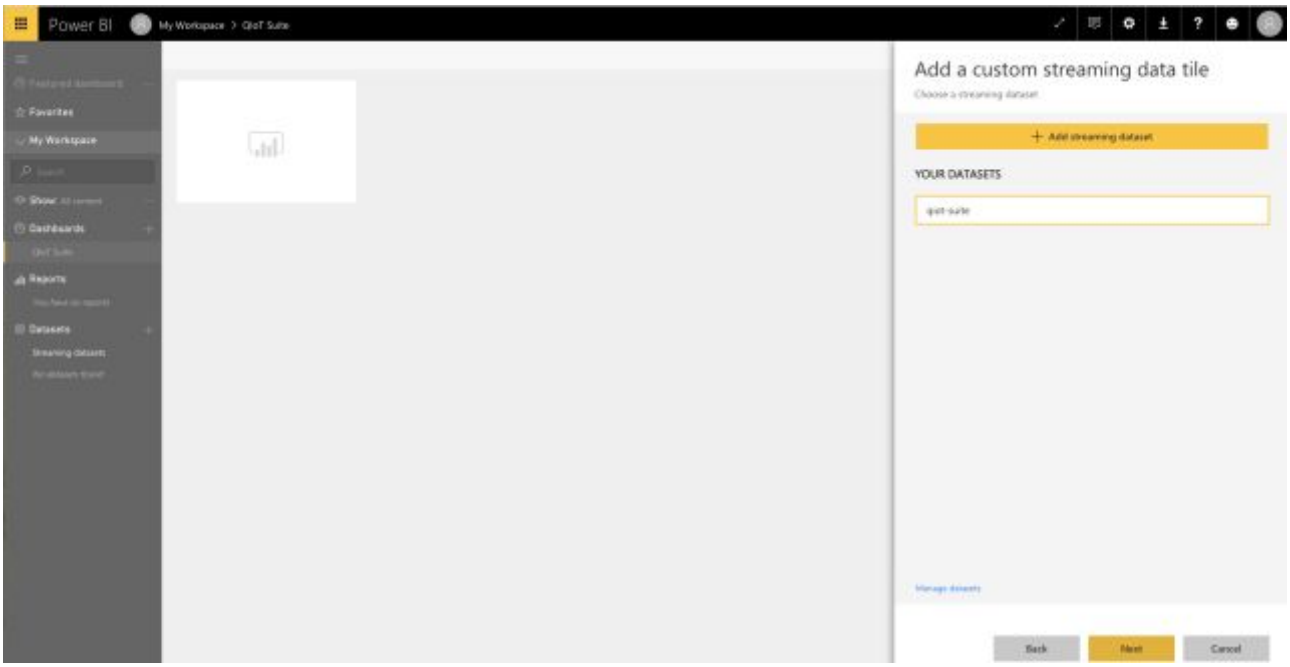
- Click “Add tile” in screen upper right corner



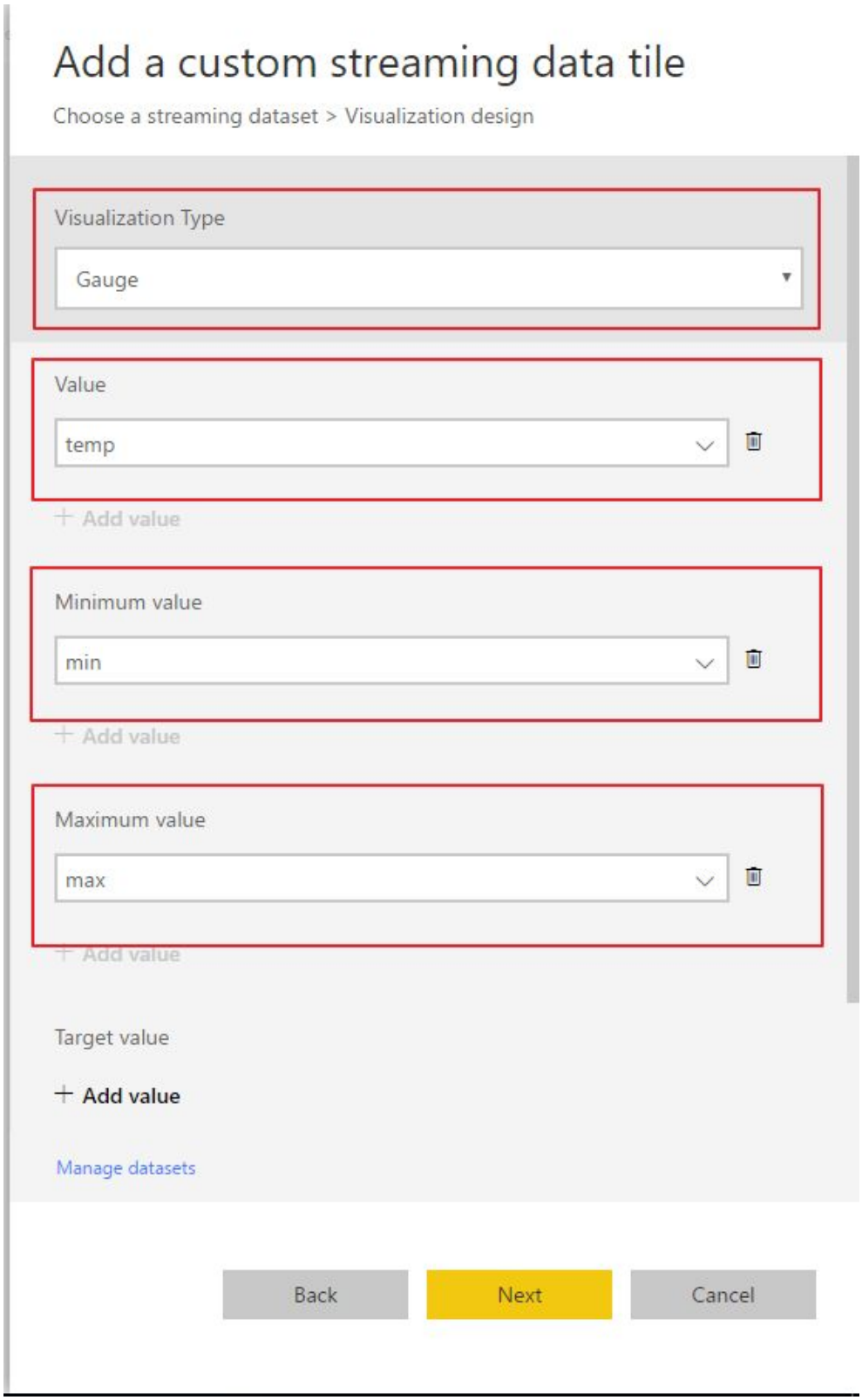
- Select “CUSTOM STREAMING DATA” and then select the “Next” button.



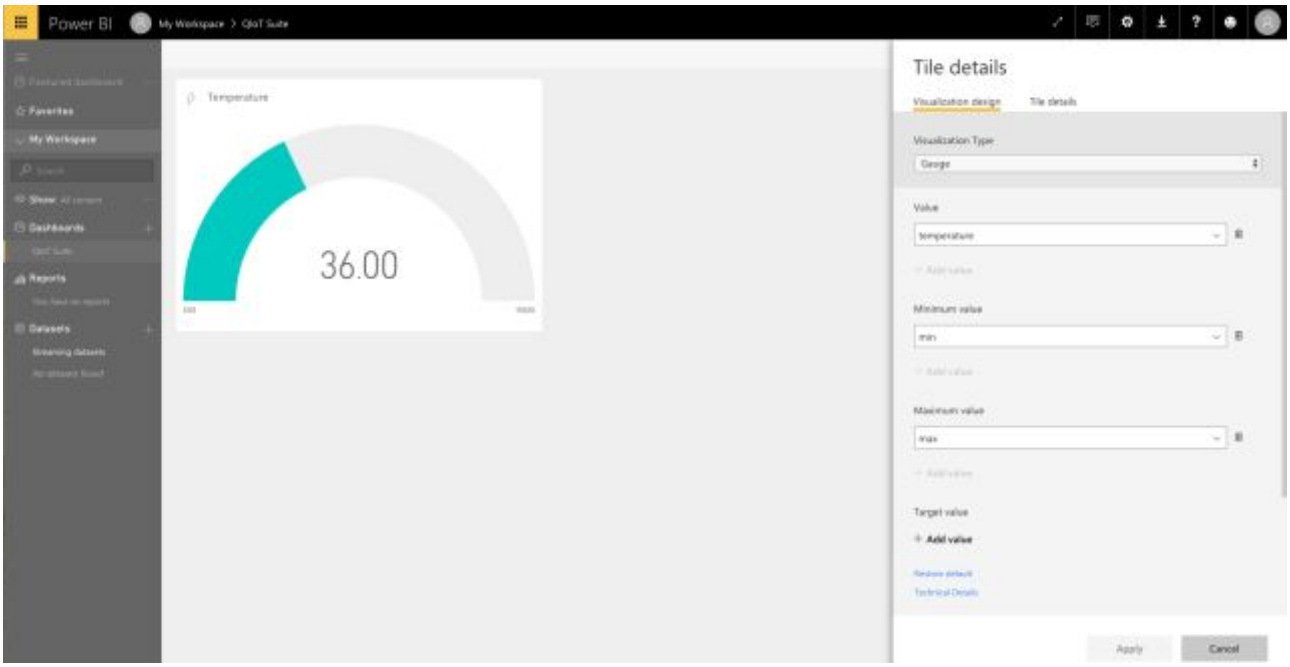
- Select datasets and then click the “Next” button.



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



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


Power BI
My Workspace > QoT Suite
Favorites
My Workspace
Dashboards
Reports
Datasets
Temperature
36.00
Tile details
Visualization design
Tile details
Visualization Type
Gauge
Value
temperature
Minimum value
min
Maximum value
max
Target value
Add value
Apply
Cancel

Appendix

QNAP QIoT Startkit Sample Code Introduction

- GitHub: Sample Code
- Sample Code Structure

```
qnap-qiot-sdks/
    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	{ "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>"resources": [# resource des
	{
	"resourcename": "adf", # resource name
	"resourceid": "dfadf", # resource id
	"resourcetypename": "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>