



Connect S210X Sensors to Azure IoT Central User Guide

Version: v1.0.0



Table of Contents

1. SenseCAP S210X Sensors	1
2. SenseCAP & Node-RED	2
2.1.1 Node-RED	2
2.1.2 Install Node.Js	2
2.1.3 Installing Node-RED with npm	3
2.2 Get the SenseCAP API	4
2.3 Node-RED Configuration	5
2.3.1 Add a new mqtt-broker node	5
2.3.2 Add Topic	8
2.3.3 Add debug node	9
3. SenseCAP & Node_RED & Azure IoT Central	10
3.1 Microsoft Azure IoT Central Configuration	11
3.1.1 Log in to Azure IoT Central	11
3.1.2 Create a new application	13
3.1.3 Create a Device Template	14
3.1.4 Create a Device	15
3.2 Node-RED Configuration	16
3.2.1 Install Azure IoT Paletts	16
3.2.2 Configure the Azure IoT Central node	17
3.2.3 Configure the function node	18
3.3 Data Presentation	20
3.3.1 Raw data	20
3.3.2 Dashboard	21
4. Tech Support	24

1. SenseCAP S210X Sensors

SenseCAP S210X Sensors are with the IP66 rating, $-40 \sim +85C^{\circ}$ operating temperature and a built-in 19Ah high-capacity battery, combined with the devices' low power consumption, the series can operate in harsh outdoor environments for up to 10 years with a range of up to 10km.



In this tutorial, we will introduce how to connect the SenseCAP S210X Sensors to the Microsoft Azure IoT Central via Node-RED

2. SenseCAP & Node-RED

This chapter, the first in a series, walks you through installing and using Node-red and calling the SenseCAP API to connect to Node-RED.

This chapter is to make it easier for our users to connect data from the SenseCAP platform to various other Paas platforms for more in-depth data processing.

2.1.1 Node-RED

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.



2.1.2 Install Node.Js

To install Node-RED locally you will need a supported version of Node.js. Node-RED currently recommends [Node 14.x LTS](#).

2.1.3 Installing Node-RED with npm

To install Node-RED you can use this npm command that comes with Node.js:

```
sudo npm install -g --unsafe-perm node-red
```

**Note:**

If you are using Windows, do not start the command with “sudo”.

Once installed as a global module you can use this command to start Node-RED in your terminal:

```
nore-red
```

```
jessie@JessiedeMacBook-Air ~ % node-red
8 Sep 09:28:19 - [info]

Welcome to Node-RED
=====

8 Sep 09:28:19 - [info] Node-RED version: v3.0.2
8 Sep 09:28:19 - [info] Node.js version: v18.7.0
8 Sep 09:28:19 - [info] Darwin 21.6.0 arm64 LE
8 Sep 09:28:20 - [info] Loading palette nodes
8 Sep 09:28:20 - [warn] -----
8 Sep 09:28:20 - [warn] [node-red-node-rbe/rbe] 'rbe' already registered by module
node-red
8 Sep 09:28:20 - [warn] -----
8 Sep 09:28:20 - [info] Settings file : /Users/jessie/.node-red/settings.js
8 Sep 09:28:20 - [info] Context store : 'default' [module=memory]
8 Sep 09:28:20 - [info] User directory : /Users/jessie/.node-red
8 Sep 09:28:20 - [warn] Projects disabled : editorTheme.projects.enabled=false
8 Sep 09:28:20 - [info] Flows file : /Users/jessie/.node-red/flows.json
8 Sep 09:28:20 - [info] Server now running at http://127.0.0.1:1880/
8 Sep 09:28:20 - [warn]
```

Then you can then access the Node-RED editor by pointing your browser at <http://localhost:1880>.

2.2 Get the SenseCAP API

Before proceeding to this section, make sure you have bound your S210x device in the SenseCAP console.



Note:

If you haven't bound the device, please check [S210X Sensors User Guide](#) first

Log in to the [SenseCAP console](#). In the drop-down bar to the right of the user name at the top of the dashboard, we can find the **Organization Information**, please select it to get the **Organization ID**

Organization Information – Displaying the details of the current organization account.

Organization Information (The information cannot be modified. If you need to modify it please email to: sensecap@seed.cc)

Organization Name: Jessie
 User Account: [redacted]
 Email: [redacted]
 Telephone: null
 Official Website: null
Organization Id: 42425274000000000000
 Country: Armenia

Then, we also need to get the API key for SenseCAP. Please click on **Security -> Access API keys** on the left side of the dashboard. Then Create an Access Key.

Security / Access API keys – Use access keys to make secure REST or HTTP Query protocol requests to SENSECAP service APIs, so keep them safe.

Organization Id: 42425274000000000000

Create Access Key

Label	API ID	Access Level	API Status	Creation Time
	SYKYC...	<input type="radio"/> Read Only <input checked="" type="radio"/> Full Access	Active (effect at 2022-08-18 10:10:03)	2022-08-18 10:10:03

Click on the **API ID** you created and you will get her **Access API keys**, please copy it and the **Organization ID**, we will use them in the later steps.

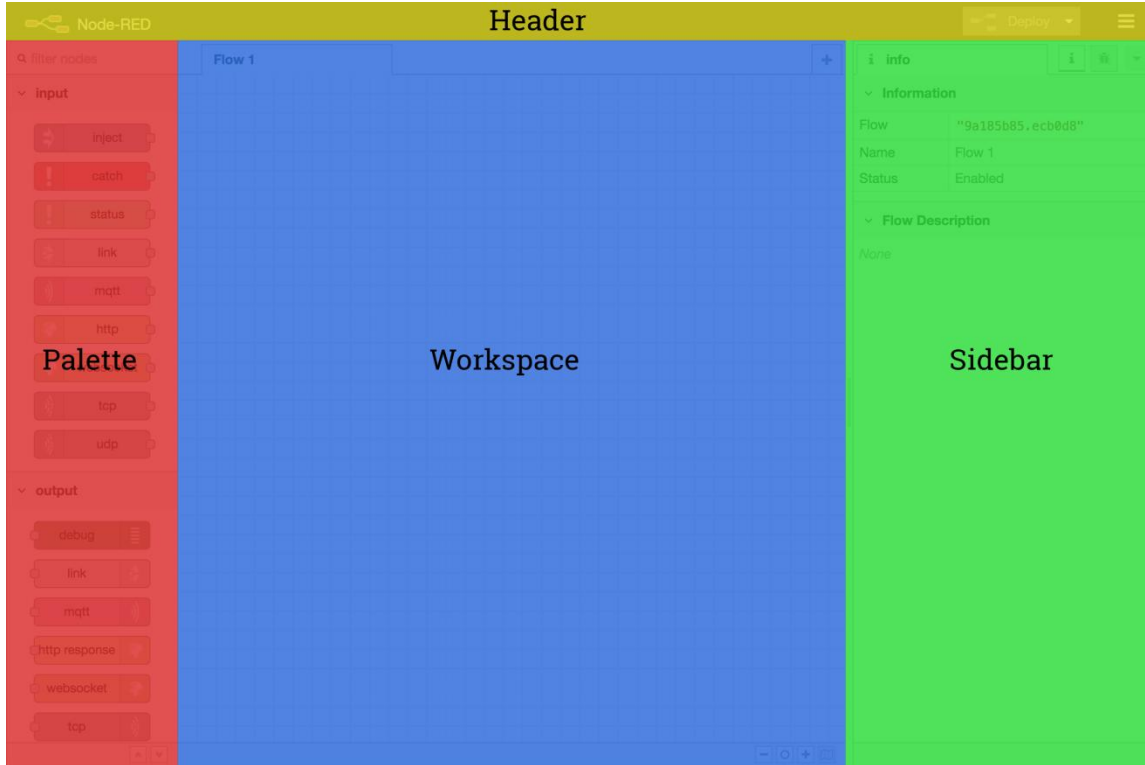
Access API keys

API ID: SYKYC...

Access API keys: [redacted]

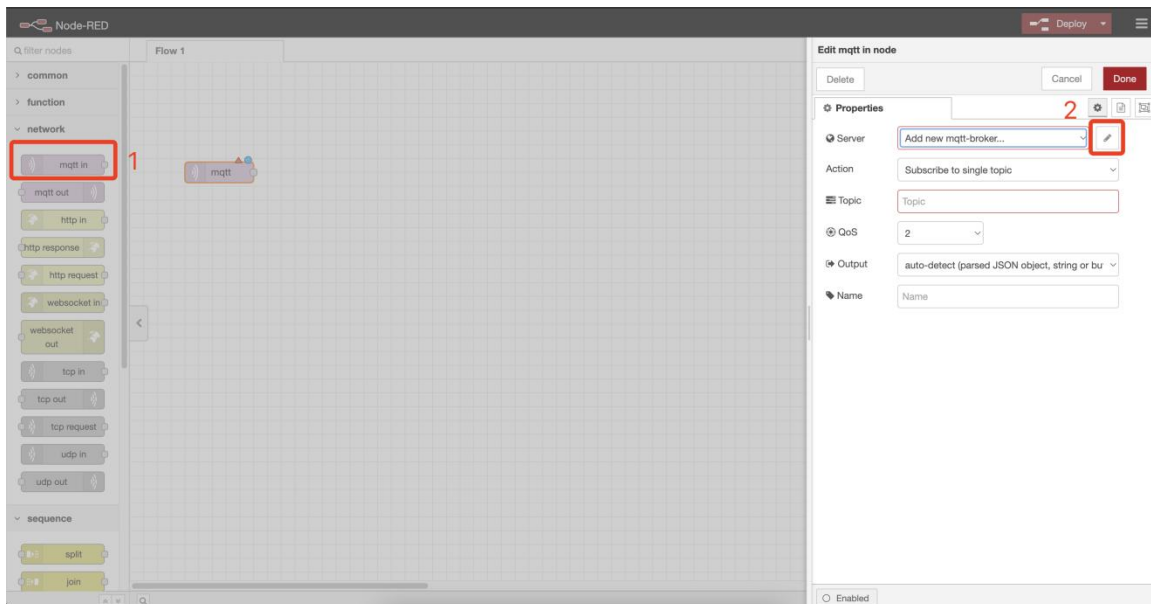
Close

2.3 Node-RED Configuration



2.3.1 Add a new mqtt-broker node

Drag out an **mqtt in** node, double-click it to enter the configuration page, then click the edit button after **Add new mqtt-broker**.



The configuration of **mqtt-broker** is required to be filled out as follows:

Server: openstream.api.sensecap.sseed.cc

Port: 1883

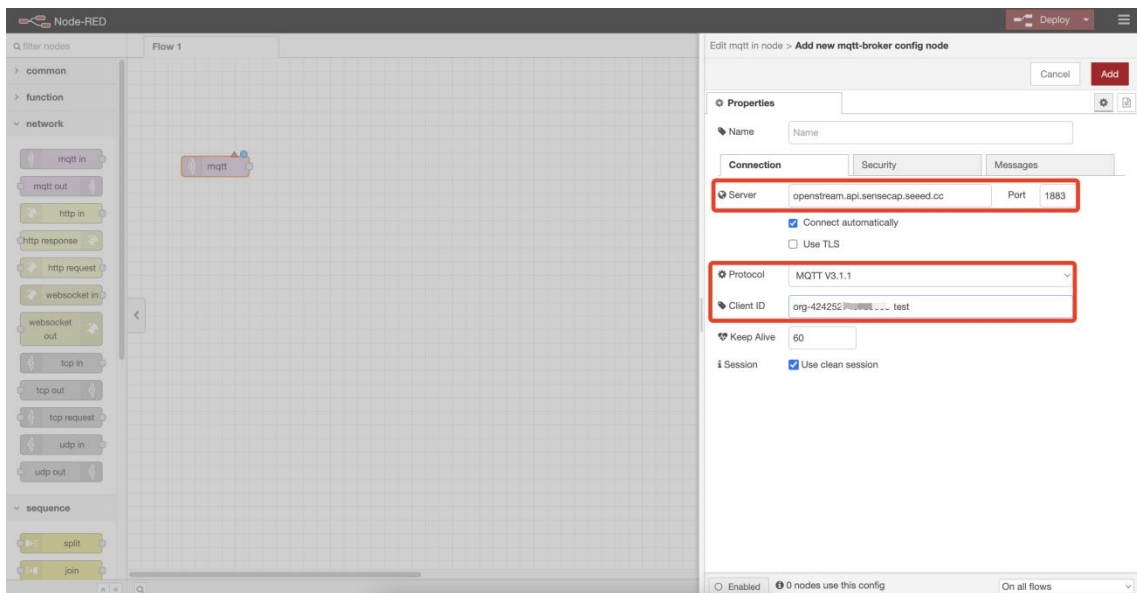
Protocol: MQTT V3.1.1

Client ID: **org-<Organization ID><Random ID>**

Organization ID: Obtained from your **Organization information**

Random ID: Use your own randomly generated numbers and lowercase letters.

Example:org-43243***23-test

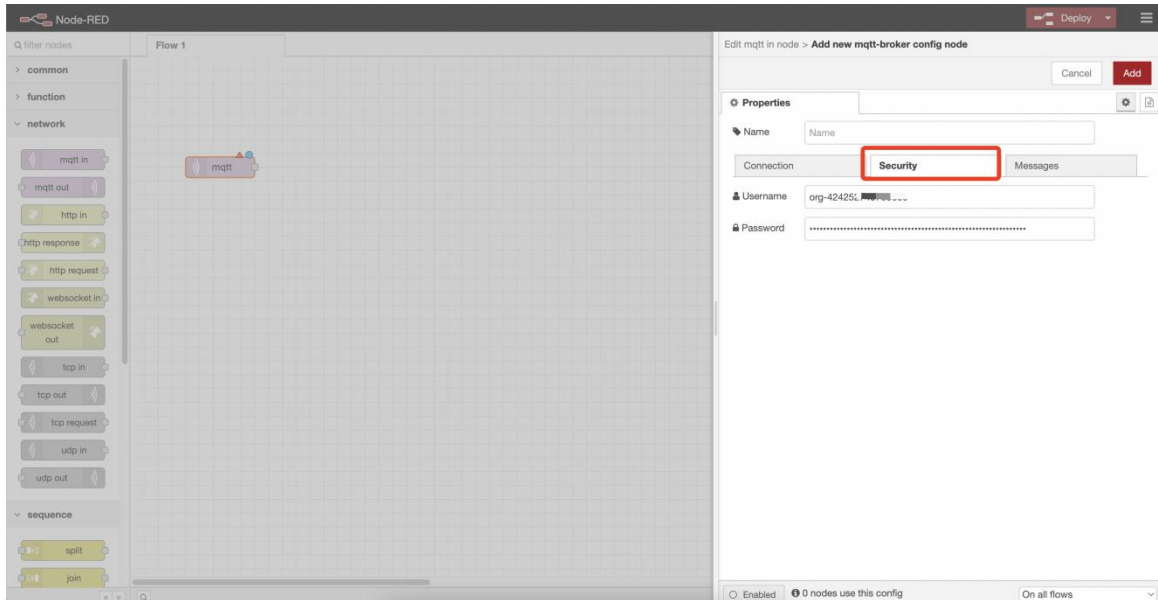


Then we fill in the **Security** options field with the Username and Password:

Username format: **org-*<Organization ID>***

Organization ID: Your organization ID we have obtained it before

Password: Fill in the **Access API keys** that we obtained before.



2.3.2 Add Topic

Topic: Configuring a topic in a specific format determines the kind of device and data type to be received.

Topic format:

/device_sensor_data/<OrgID>/<DeviceEUI>/<Channel>/<Reserved>/<Measurement ID>

OrgID	You can find the id on your organization information
DeviceEUI	You can find EUI on the Device Basic Properties or device label
Channel	A physical interface on the device to connect to the sensor, default:1
Reserved	Reserved field
MeasurementID	Please check measurement_list

⚠ Note:

"+" indicates that this field has no filter conditions and can match all.
 "/+/"+/+/" means to listen to all "<DeviceEUI>", "<Channel>", "<Reserved>",
 "<MeasurementID>"

Example: /device_sensor_data/424988****44/2CF7F***0002/+/+/+

This topic means receiving all remote sensing data of the current device.

The screenshot shows the Node-RED interface with a flow editor on the left and a configuration panel on the right. The configuration panel is titled 'Edit mqtt in node' and contains the following fields:

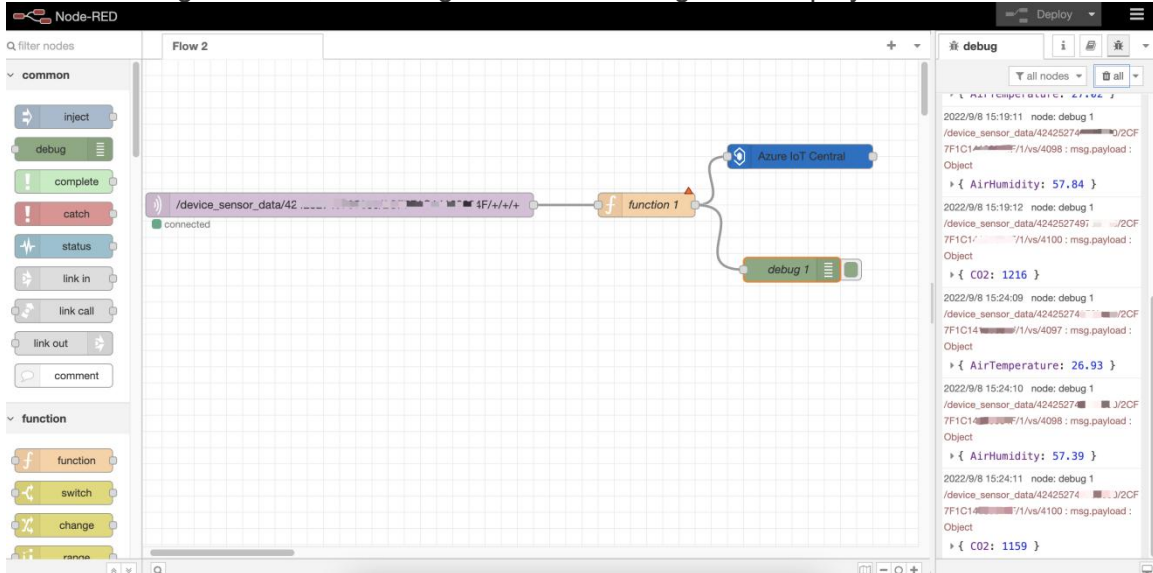
- Delete:** A button to delete the node.
- Cancel:** A button to cancel the configuration.
- Done:** A button to save the configuration.
- Properties:** A section containing:
 - Server:** A dropdown menu showing 'org-4242527...-test@openstrea'.
 - Action:** A dropdown menu showing 'Subscribe to single topic'.
 - Topic:** A text input field containing '/device_sensor_data/424253...'. This field is highlighted with a red box.
 - QoS:** A dropdown menu showing '2'.
 - Output:** A dropdown menu showing 'auto-detect (parsed JSON object, string or bu'.
 - Name:** A text input field containing 'Name'.
- Enabled:** A checkbox at the bottom of the panel.

2.3.3 Add debug node

Drag out a **debug** node, connect to the **mqtt-in** node, then click **Deploy**

After the deployment is successful, you will see "**Connected**" under the **mqtt in** building block, the data reporting interval is determined by the sensor which we connected.

After receiving the data, the debug window on the right will display the raw data.



The screenshot shows the Node-RED web interface. On the left, the 'common' node palette includes 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The 'function' palette includes 'function', 'switch', 'change', and 'range'. The main workspace, titled 'Flow 2', contains a flow starting with an MQTT input node labeled '/device_sensor_data/42...', followed by a function node labeled 'function 1', and then an 'Azure IoT Central' node. A 'debug 1' node is connected to the output of the function node. Below the MQTT node, a green indicator shows 'connected'. On the right, the 'debug' window displays the following raw data:

```
2022/9/8 15:19:11 node: debug 1
/device_sensor_data/42425274.../2CF
7F1C1.../1/s/4098 : msg.payload :
Object
  > { AirHumidity: 57.84 }

2022/9/8 15:19:12 node: debug 1
/device_sensor_data/424252749f.../2CF
7F1C1.../1/s/4100 : msg.payload :
Object
  > { CO2: 1216 }

2022/9/8 15:24:09 node: debug 1
/device_sensor_data/42425274.../2CF
7F1C14.../1/s/4097 : msg.payload :
Object
  > { AirTemperature: 26.93 }

2022/9/8 15:24:10 node: debug 1
/device_sensor_data/4242527.../2CF
7F1C1.../1/s/4098 : msg.payload :
Object
  > { AirHumidity: 57.39 }

2022/9/8 15:24:11 node: debug 1
/device_sensor_data/42425274.../2CF
7F1C1.../1/s/4100 : msg.payload :
Object
  > { CO2: 1159 }
```

3. SenseCAP & Node_RED & Azure IoT Central

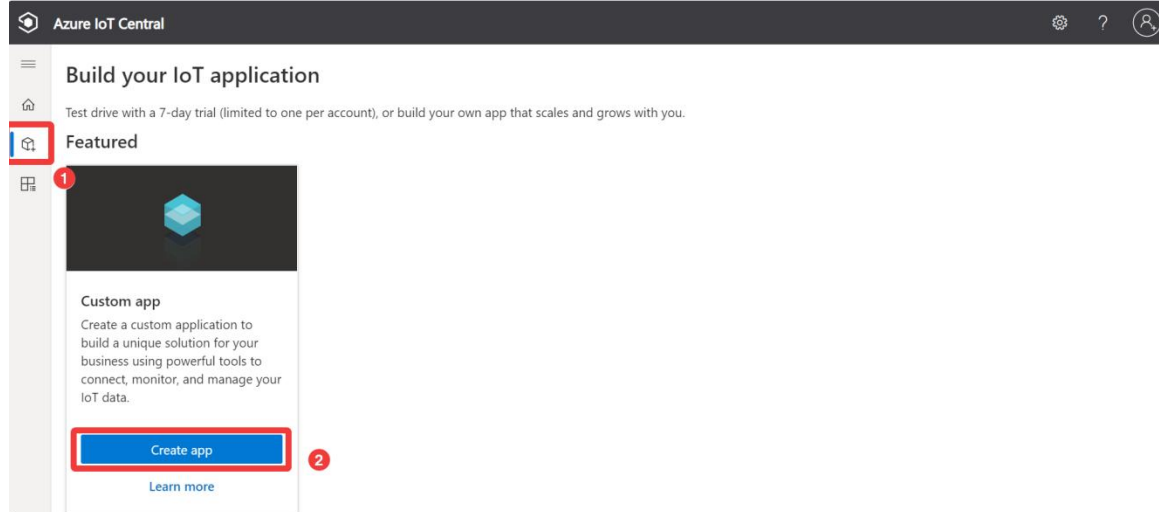
[Microsoft Azure IoT Central](#) is a fully managed global IoT SaaS (software as a service) solution that makes it easy to connect, monitor and manage your IoT assets at scale. It is highly secure, scales with your business as it grows, ensures that your investments are repeatable and integrates with your existing business apps. It also bridges the gap between your business applications and IoT data. Finally it offers centralized management to reconfigure and update your devices.

The content of this chapter will continue to use the Node-RED introduced earlier and facilitate the management of the S210X Sensor suite in Microsoft Azure IoT Central through the use of Node-RED.

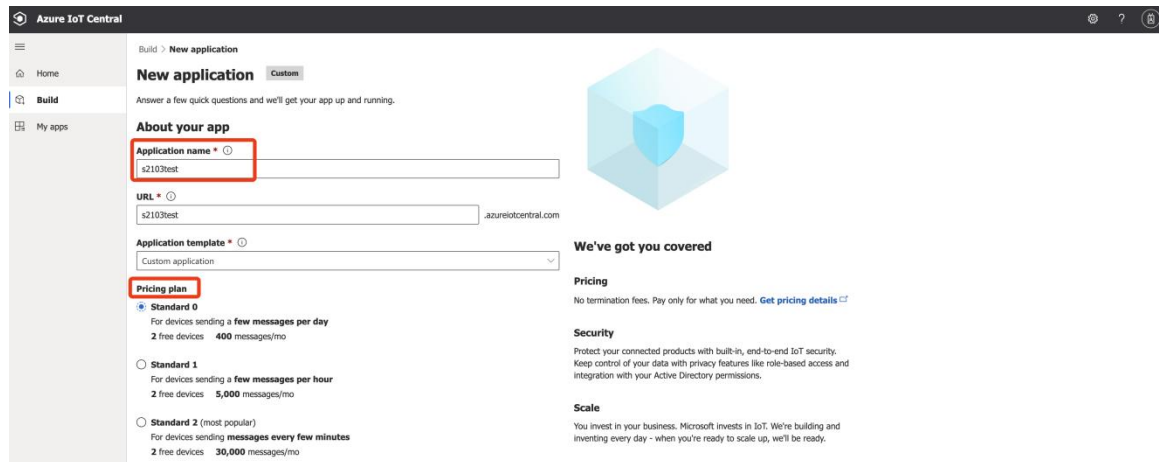
3.1 Microsoft Azure IoT Central Configuration

3.1.1 Log in to Azure IoT Central

Please visit [Azure IoT Central](#) website, click **Build** from the navigation menu on the left, and click **Custom apps**.

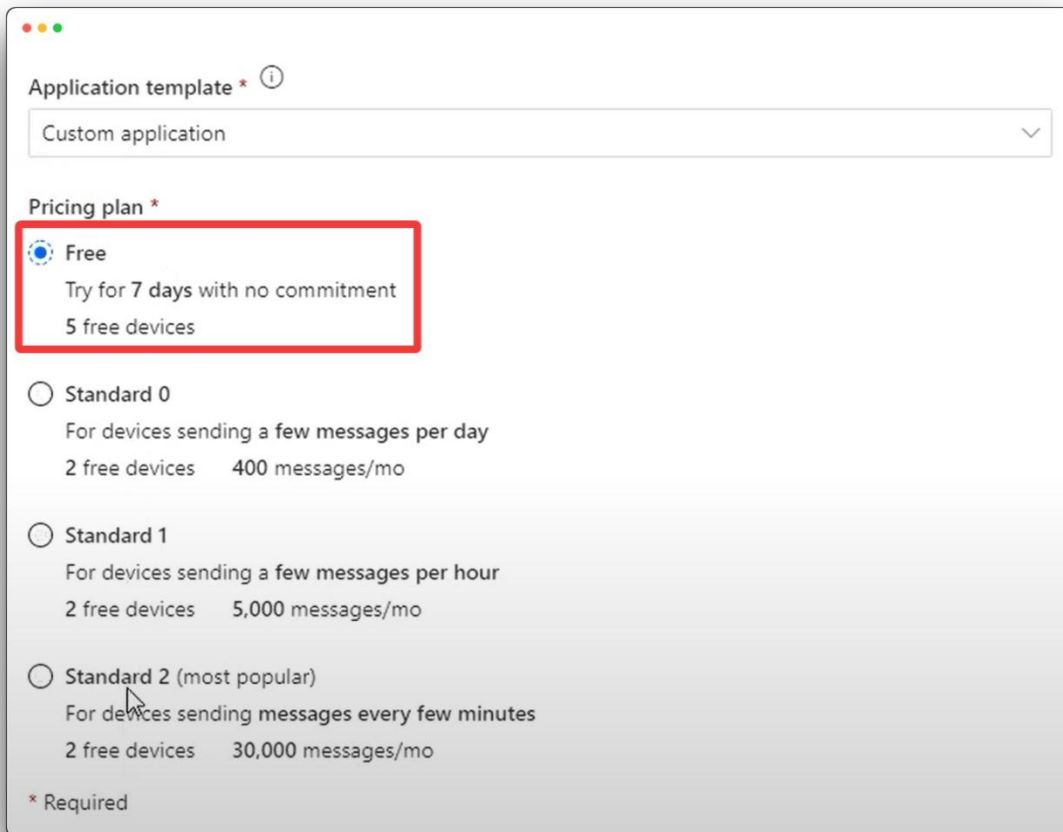


Fill in the **Application name** and choose the **Pricing plan**. Application URL will be created automatically when you fill in the application name.



 **Note:**

Note: If you are a new user of Azure IoT Central, we recommend that you select Free as this will not consume your fees.



Application template * ⓘ
Custom application

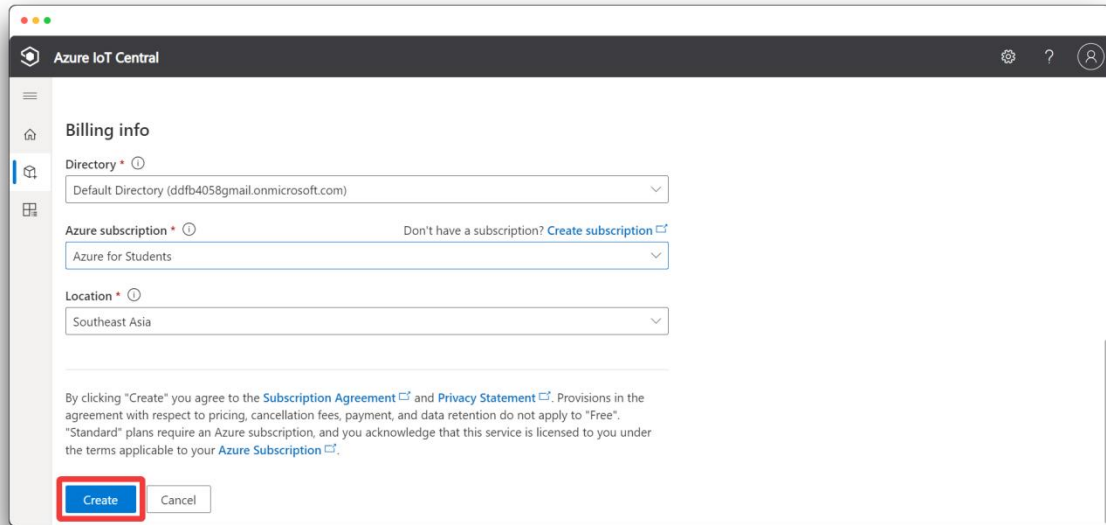
Pricing plan *

- Free
Try for 7 days with no commitment
5 free devices
- Standard 0
For devices sending a few messages per day
2 free devices 400 messages/mo
- Standard 1
For devices sending a few messages per hour
2 free devices 5,000 messages/mo
- Standard 2 (most popular)
For devices sending messages every few minutes
2 free devices 30,000 messages/mo

* Required

3.1.2 Create a new application

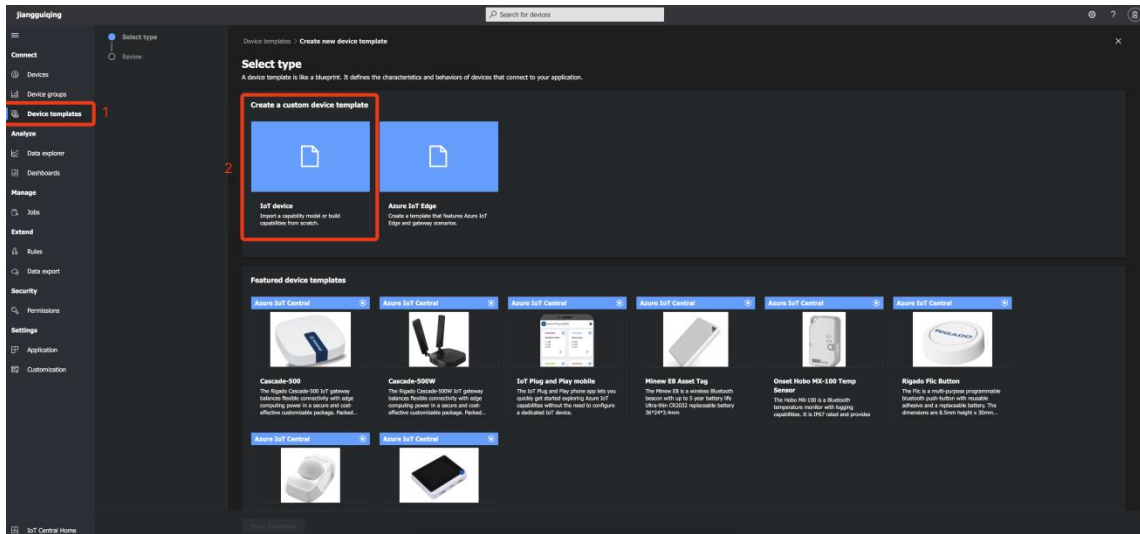
Click **Create** to create the new application. Now you have successfully set up Azure IoT Central!



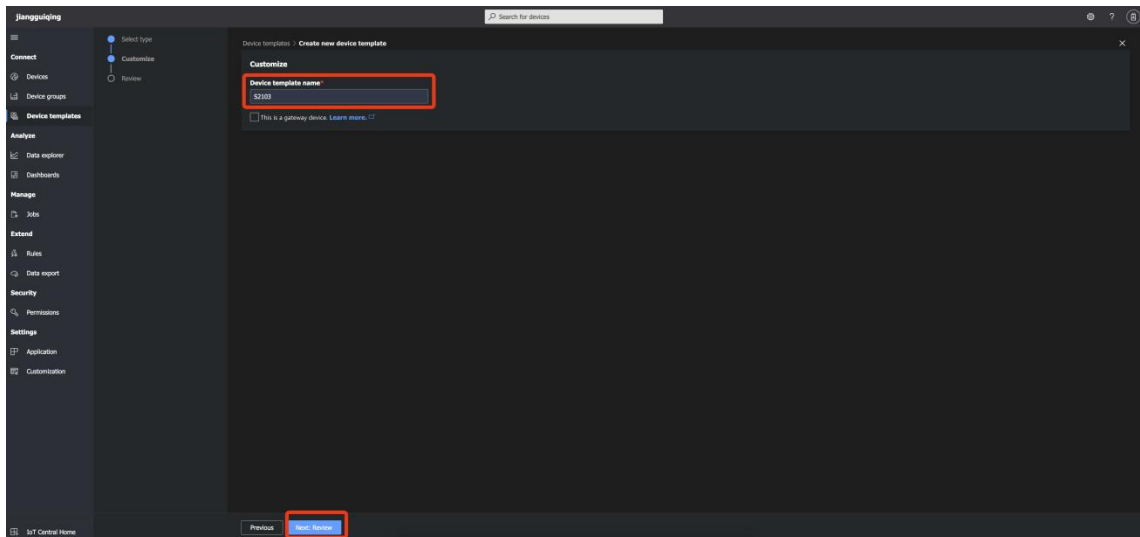
The screenshot shows the 'Billing info' section of the Azure IoT Central interface. It contains three dropdown menus: 'Directory' (Default Directory (ddf4058gmail.onmicrosoft.com)), 'Azure subscription' (Azure for Students), and 'Location' (Southeast Asia). Below these fields is a disclaimer text: 'By clicking "Create" you agree to the [Subscription Agreement](#) and [Privacy Statement](#). Provisions in the agreement with respect to pricing, cancellation fees, payment, and data retention do not apply to "Free". "Standard" plans require an Azure subscription, and you acknowledge that this service is licensed to you under the terms applicable to your [Azure Subscription](#).' At the bottom, there are two buttons: 'Create' (highlighted with a red box) and 'Cancel'.

3.1.3 Create a Device Template

Create a new device template by clicking on **Device templates** in the left-hand menu bar.

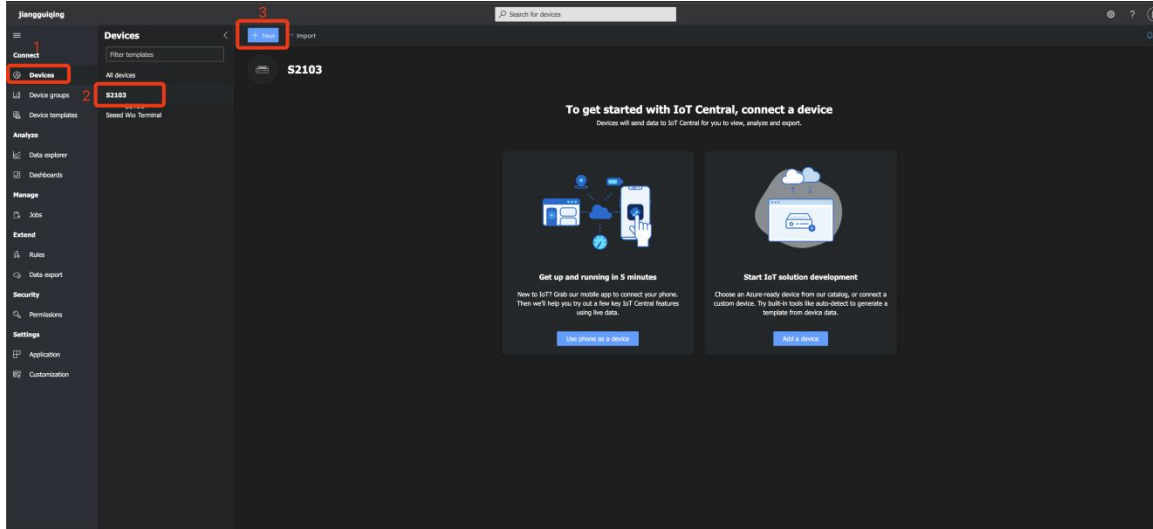


Name your device template and click **create**



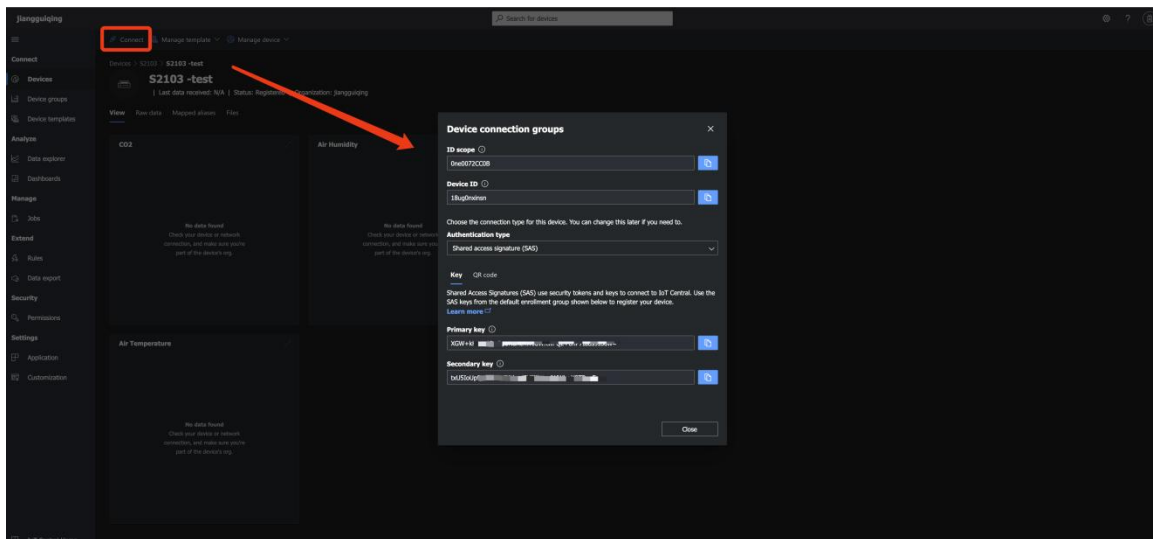
3.1.4 Create a Device

Click on **Devices** -> **S2103** under the left menu bar.



Once you have created the device, you will see the device we have just created under **Device**, please tap into the device and click on the **Connect** button in the top left corner.

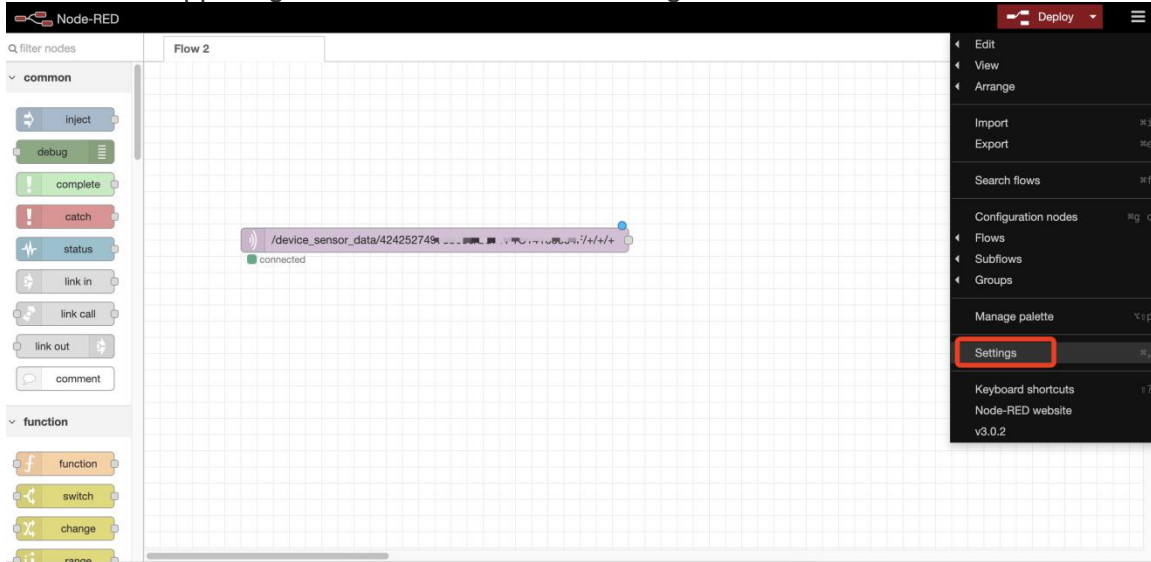
Please make a note of this information, which we will use in the next steps.



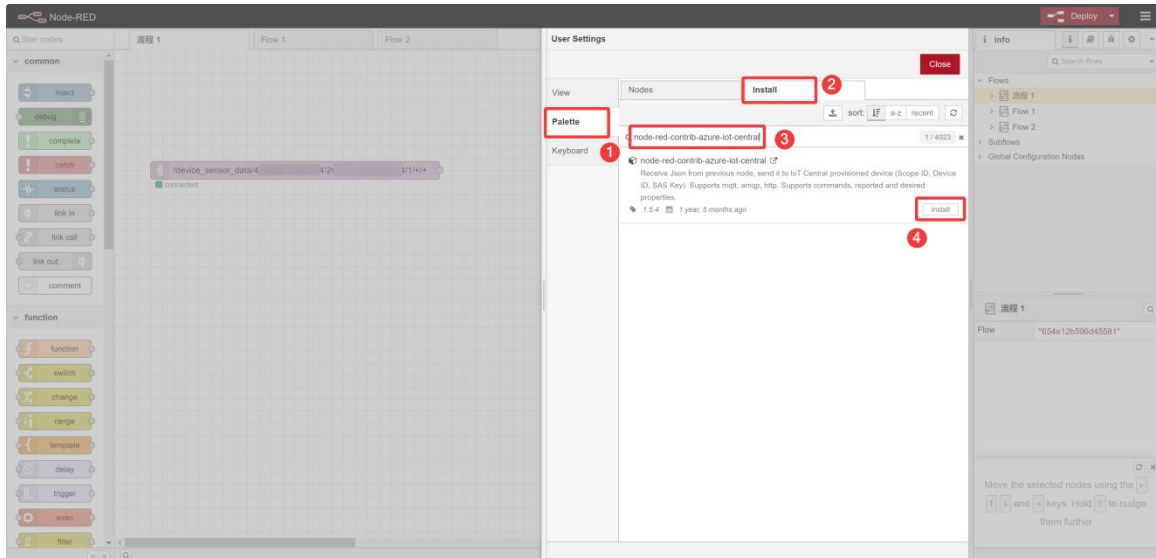
3.2 Node-RED Configuration

3.2.1 Install Azure IoT Paletts

Click on the upper-right menu bar and select Settings



Search and install "node-red-contrib-azure-iot-central" in the Paletts - Install



3.2.2 Configure the Azure IoT Central node

Drag out the **Azure IoT Central** node from the **function** bar on the left, double-click it to enter the configuration page, then click the edit button to edit **Azure IoT Central** node

The configuration is required to be filled out as follows:

Transport: MQTT

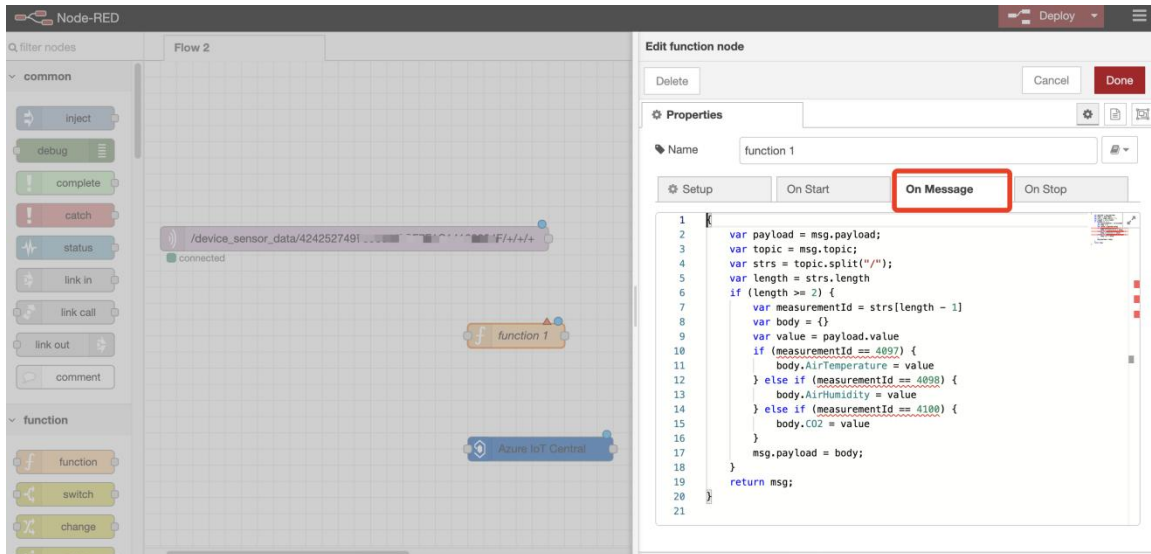
Authentication: SAS

Scope ID/Device ID/Primary Key: We have obtained it before

3.2.3 Configure the function node

Data reporting to Azure IoT Central needs to follow a specific data format, so it's necessary to add a function building block to process the data format.

Drag out the **function** node from the function bar on the left, double-click it to enter the edit page, then copy the code to **On Message**.

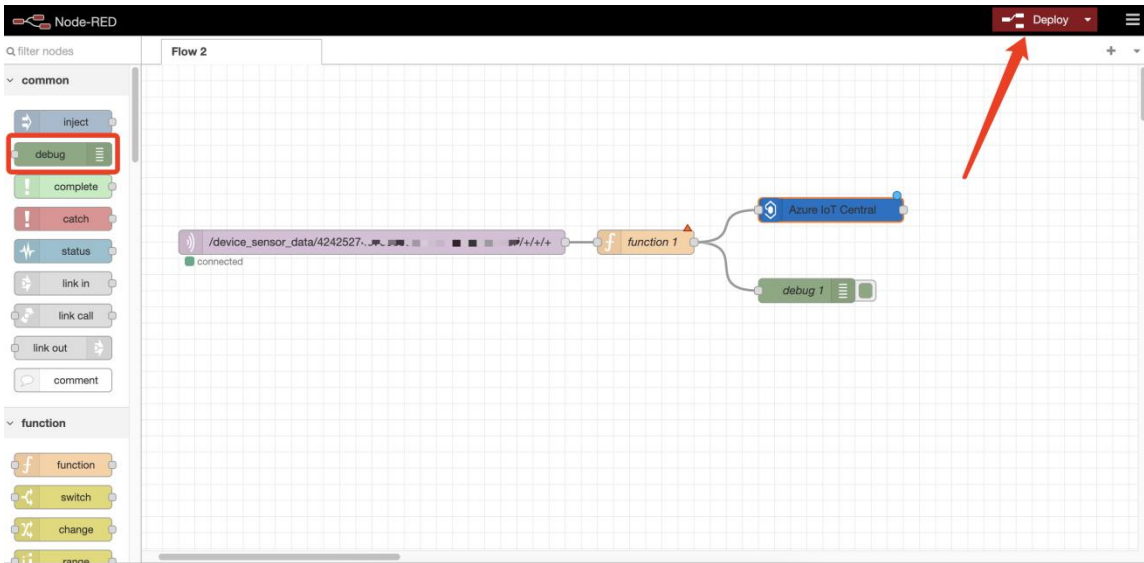


```

{
  var payload = msg.payload;
  var topic = msg.topic;
  var strs = topic.split("/");
  var length = strs.length
  if (length >= 2) {
    var measurementId = strs[length - 1]
    var body = {}
    var value = payload.value
    if (measurementId == 4097) {
      body.AirTemperature = value
    } else if (measurementId == 4098) {
      body.AirHumidity = value
    } else if (measurementId == 4100) {
      body.CO2 = value
    }
    msg.payload = body;
  }
  return msg;
}

```

If you want to see the logging information of the data, you can add a debug node after the **function** node.



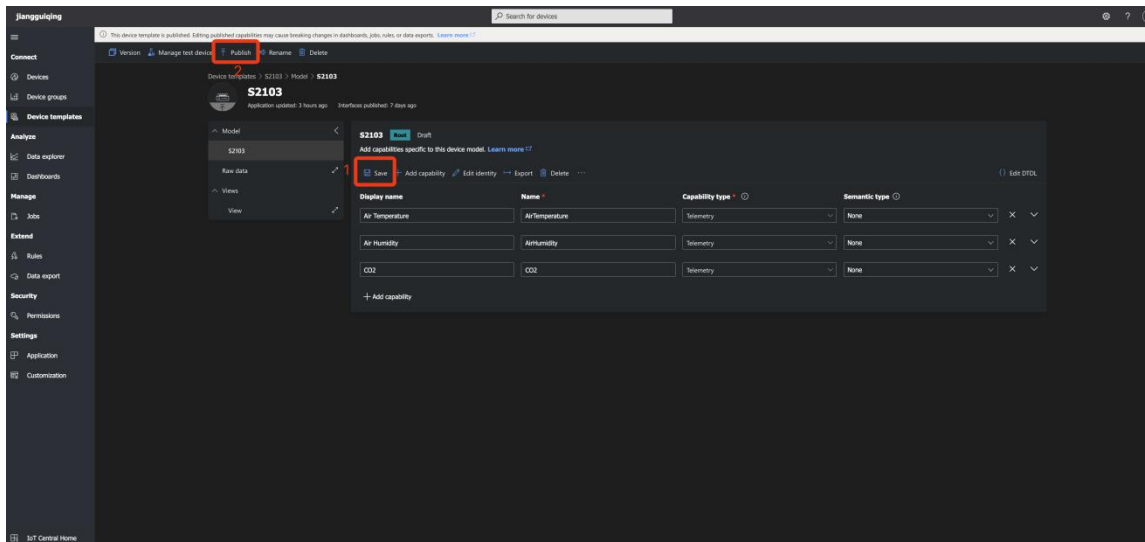
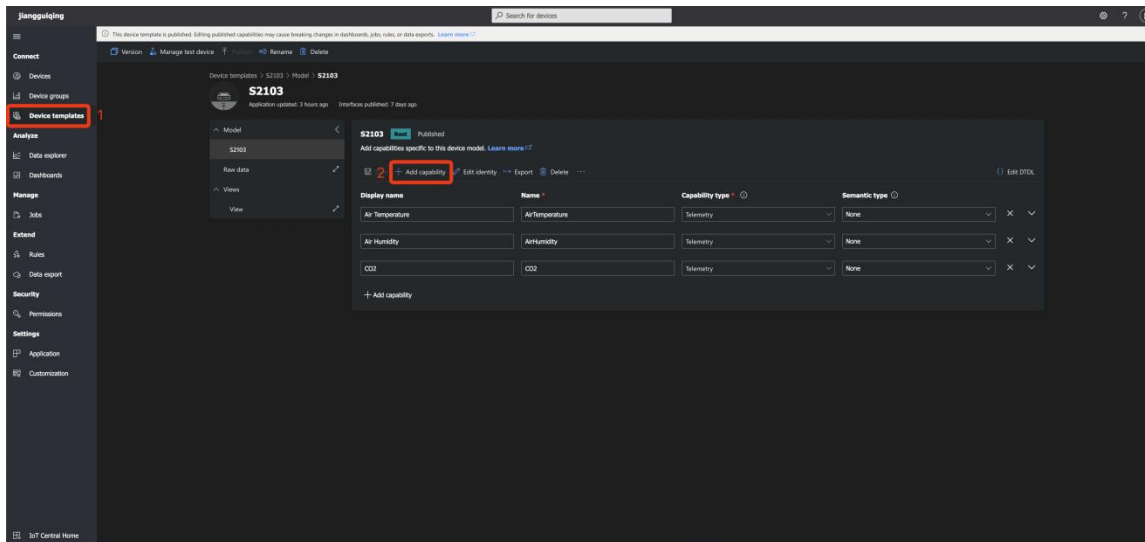
Once the S210X Sensor starts powering up and working and starts sending data to the SenseCAP PaaS server, then we can check the data on Azure IoT Central.

3.3 Data Presentation

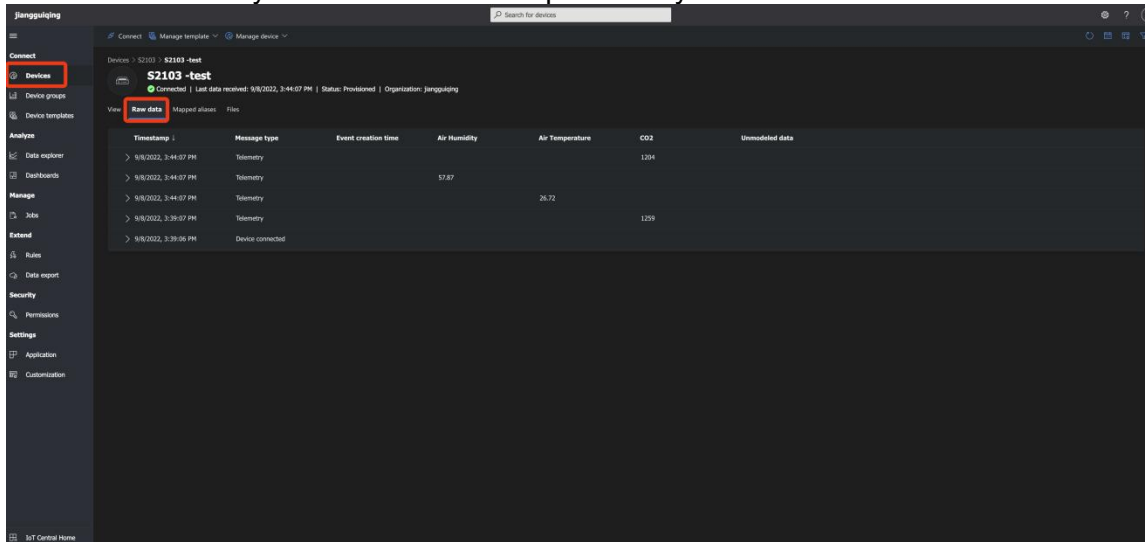
The data visible in the **Raw data** column are placed in **Unmodeled data**, so we need to parse the data according to the code above.

3.3.1 Raw data

Add the capability that you need, then click **save** and **publish**



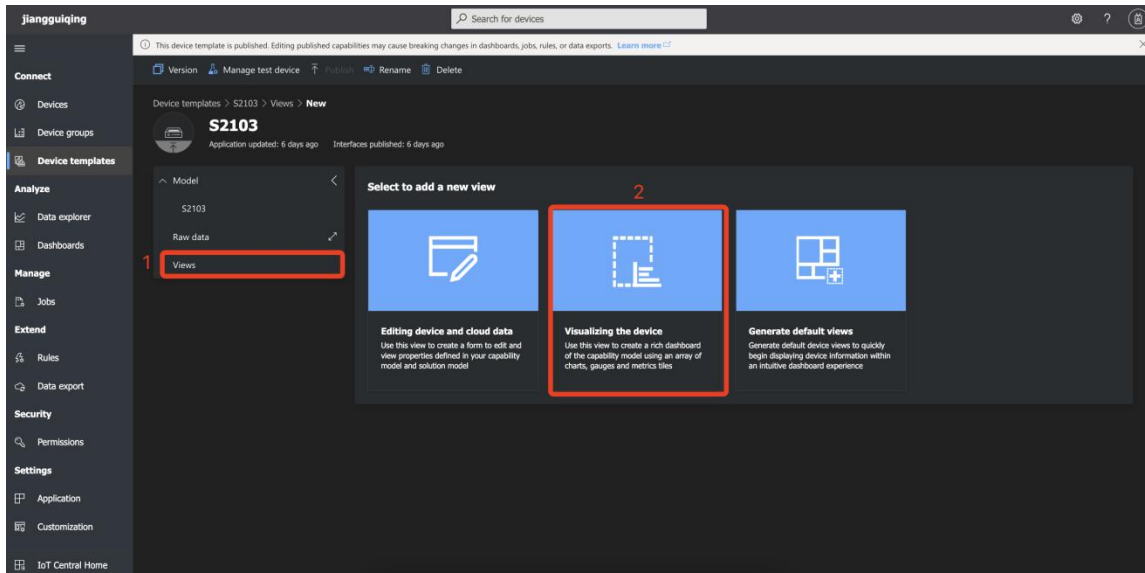
Then we can clearly check the raw data uploaded by the sensor.



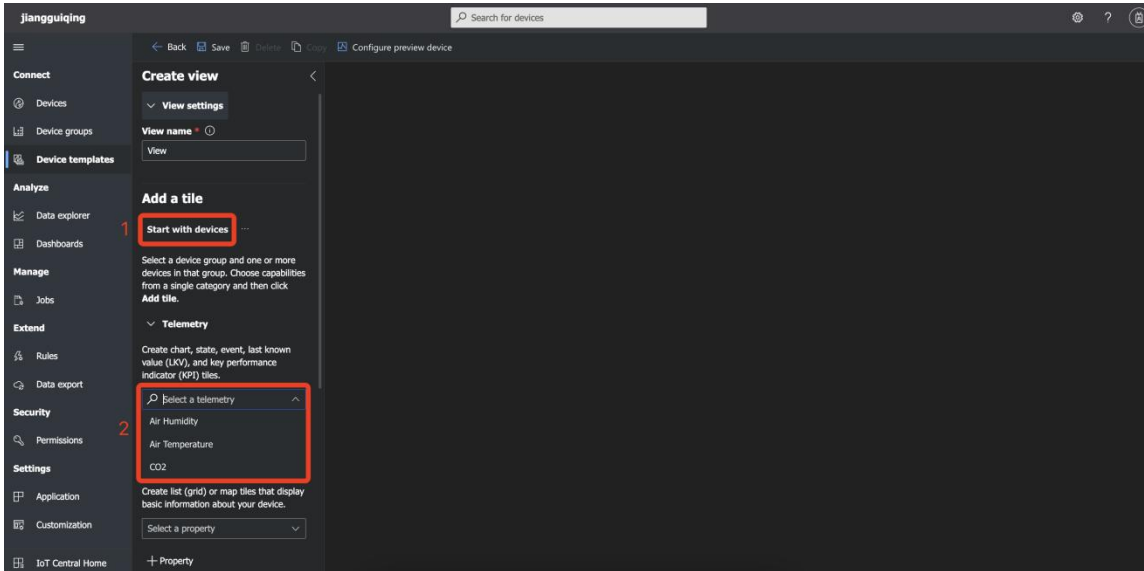
3.3.2 Dashboard

If you want to enrich your data dashboard page, you can also configure it to be displayed in Overview.

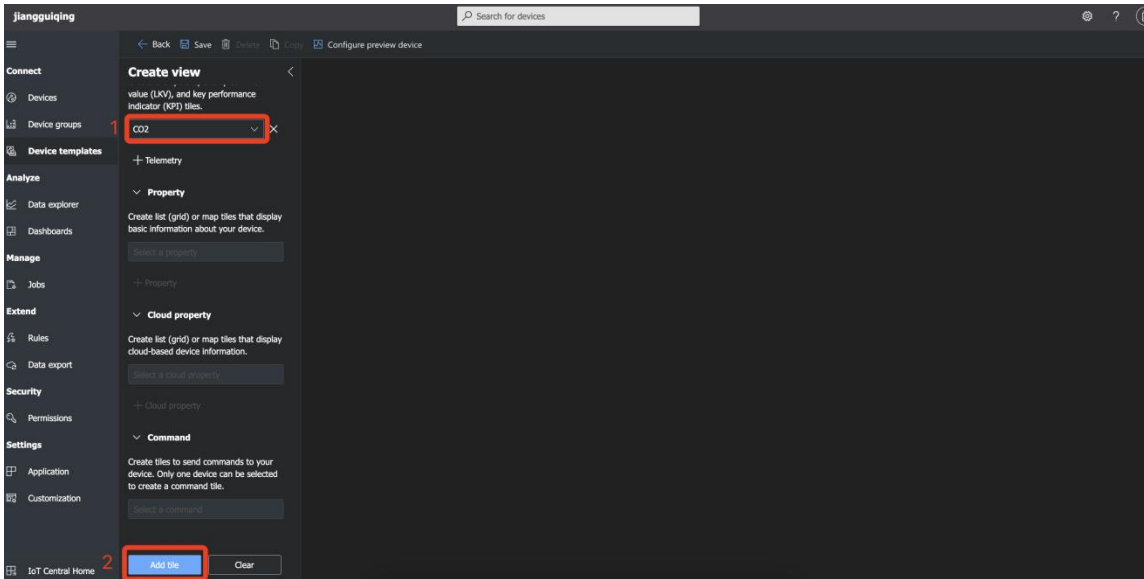
Click on **Overview** on the left navigation menu.



Collapse starts with devices drop-down menu and selects the telemetry that you want to visualize.

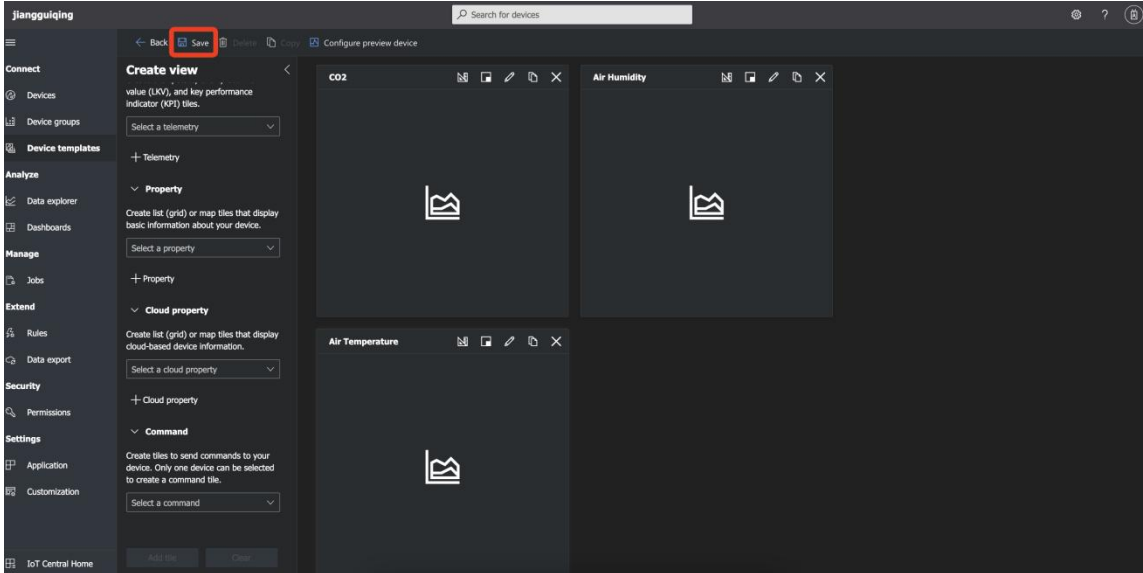


Click Add tile and you will see the tile added to the Azure IoT Central Dashboard.

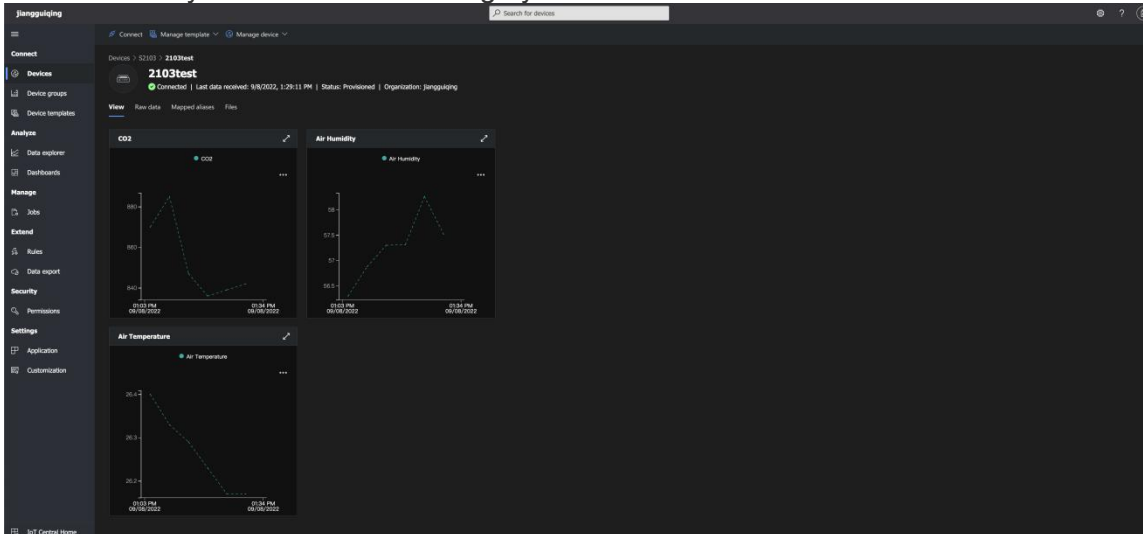


So next, customize your sensor data monitoring dashboard to your liking!

After finishing your changes, just click on **save** and **publish**



You can view your sensor data through your custom dashboard now!



4. Tech Support

Support is provided Monday to Friday, from 09:00 to 18:00 GMT+8. Due to different time zones, we cannot offer live support. However, your questions will be answered as soon as possible in the before-mentioned schedule.

Provide as much information as possible regarding your enquiry (product models, accurately describe your problem and steps to replicate it etc.) and send a mail to: sensecap@seeed.cc