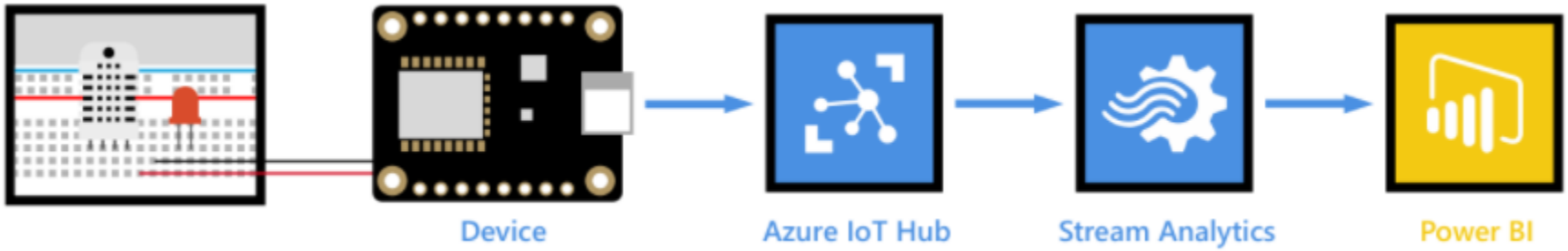
Connect Your Device to the Microsoft Azure IoT Hub

Azure IoT Hub & Azure Stream Analytics & Power BI

Arduino MKR WIFI 1010 to Azure IoT Hub

1. Connect a Arduino MKR WIFI 1010 board securely to Azure IoT Hub using an MQTT client.
2. Create a consumer group on your IoT hub.
3. Create and configure an **Azure Stream Analytics** job to read temperature telemetry from your consumer group and send it to **Power BI**.
4. Create a report of the temperature data in **Power BI** and share it to the web*.



https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi

What is the cloud?

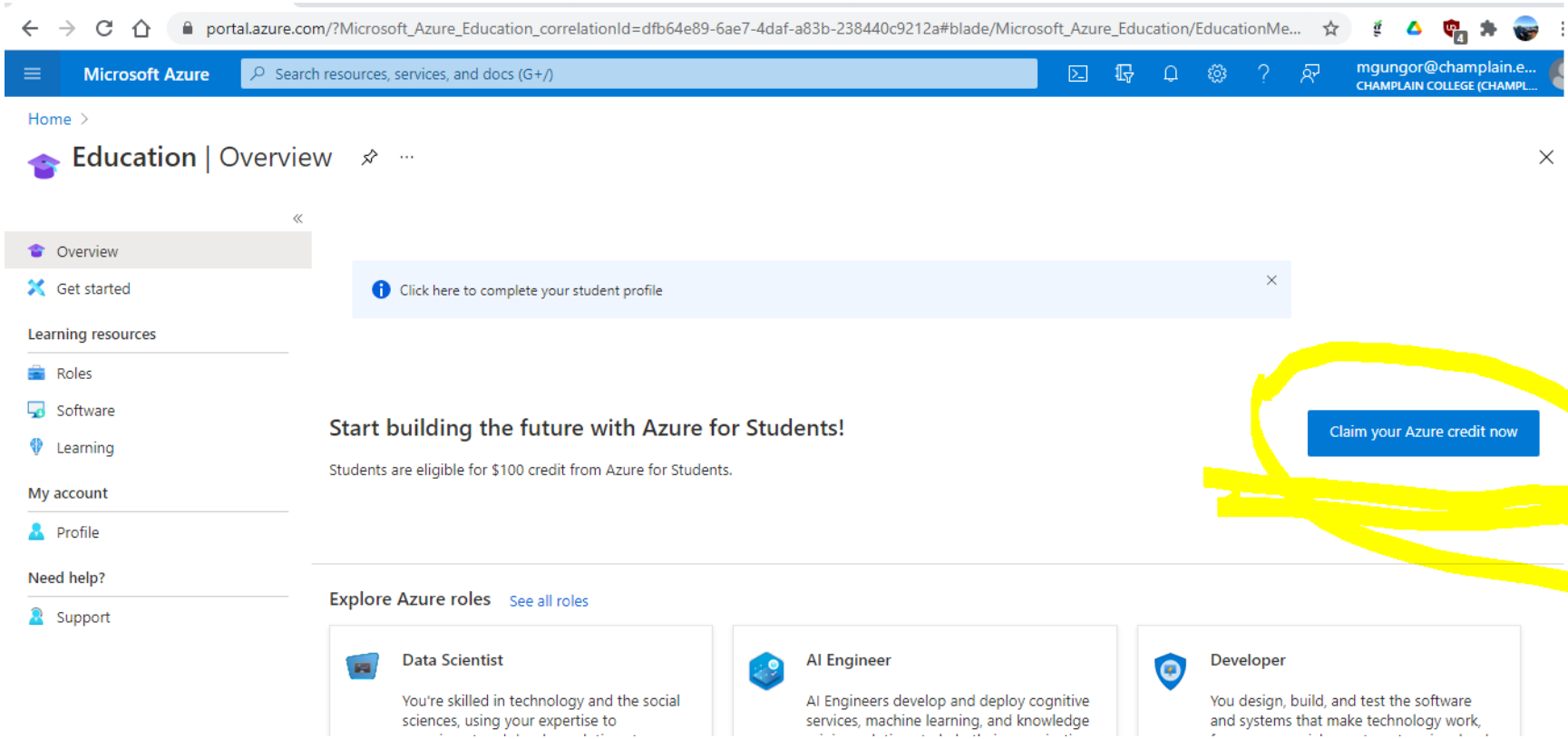
- Before the cloud, when a company wanted to provide services to their employees (such as databases or file storage), or to the public (such as websites), they would build and run a data center.
- This ranged from a room with a small number of computers, to a building with many computers.
  - Buying computers
  - Hardware maintenance
  - Power and cooling
  - Networking
  - Security, including securing the building and securing the software on the computers
  - Software installation and updates
- This could be very expensive, require a wide range of skilled employees, and be very slow to change when needed.

A case: An online store needed to plan for a busy holiday season
Buying computers just for those moments, and sitting idle till the next

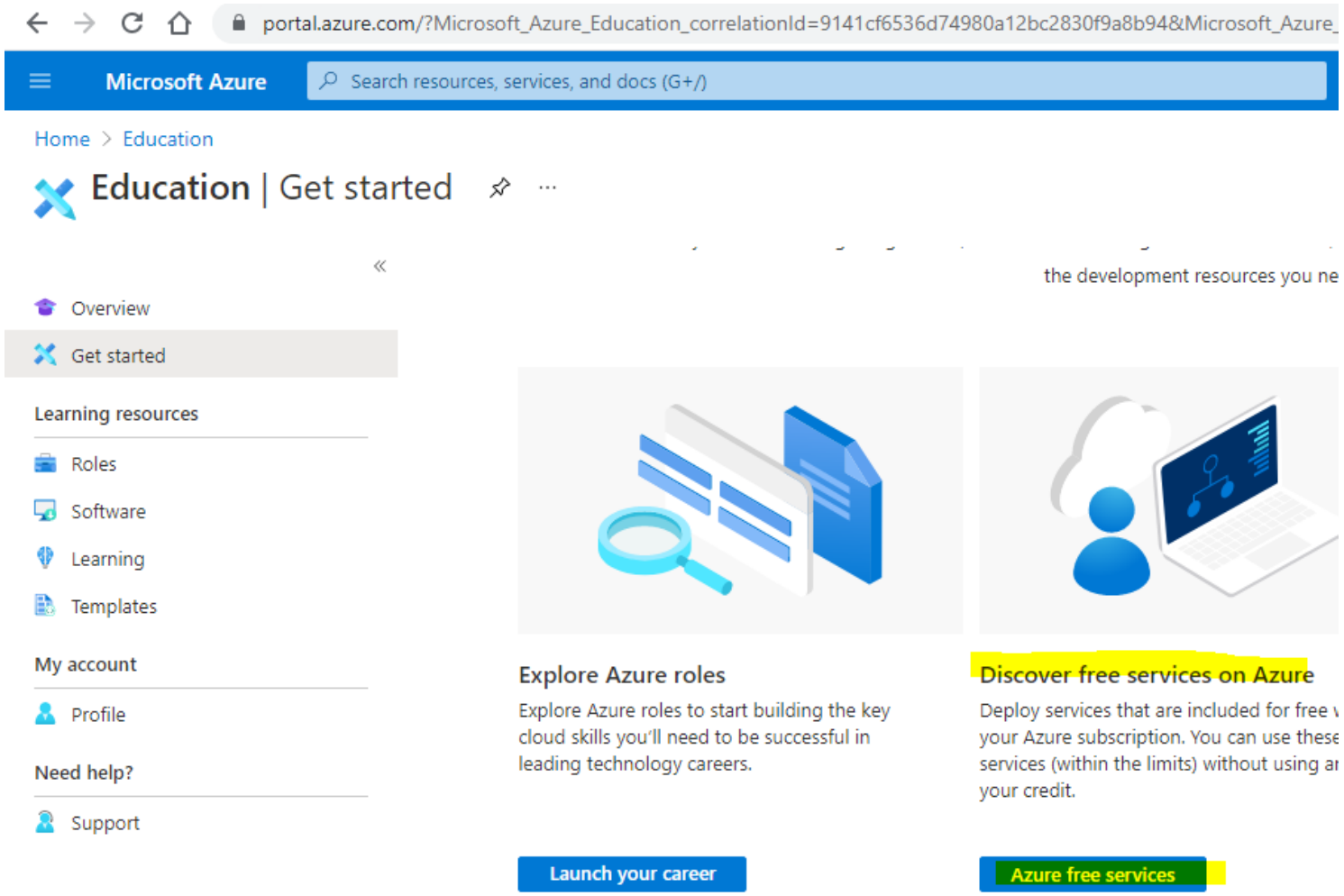
Setting Up Azure Account & Registering our device

Create a Cloud Subscription

- https://azure.microsoft.com/en-us/free/students/

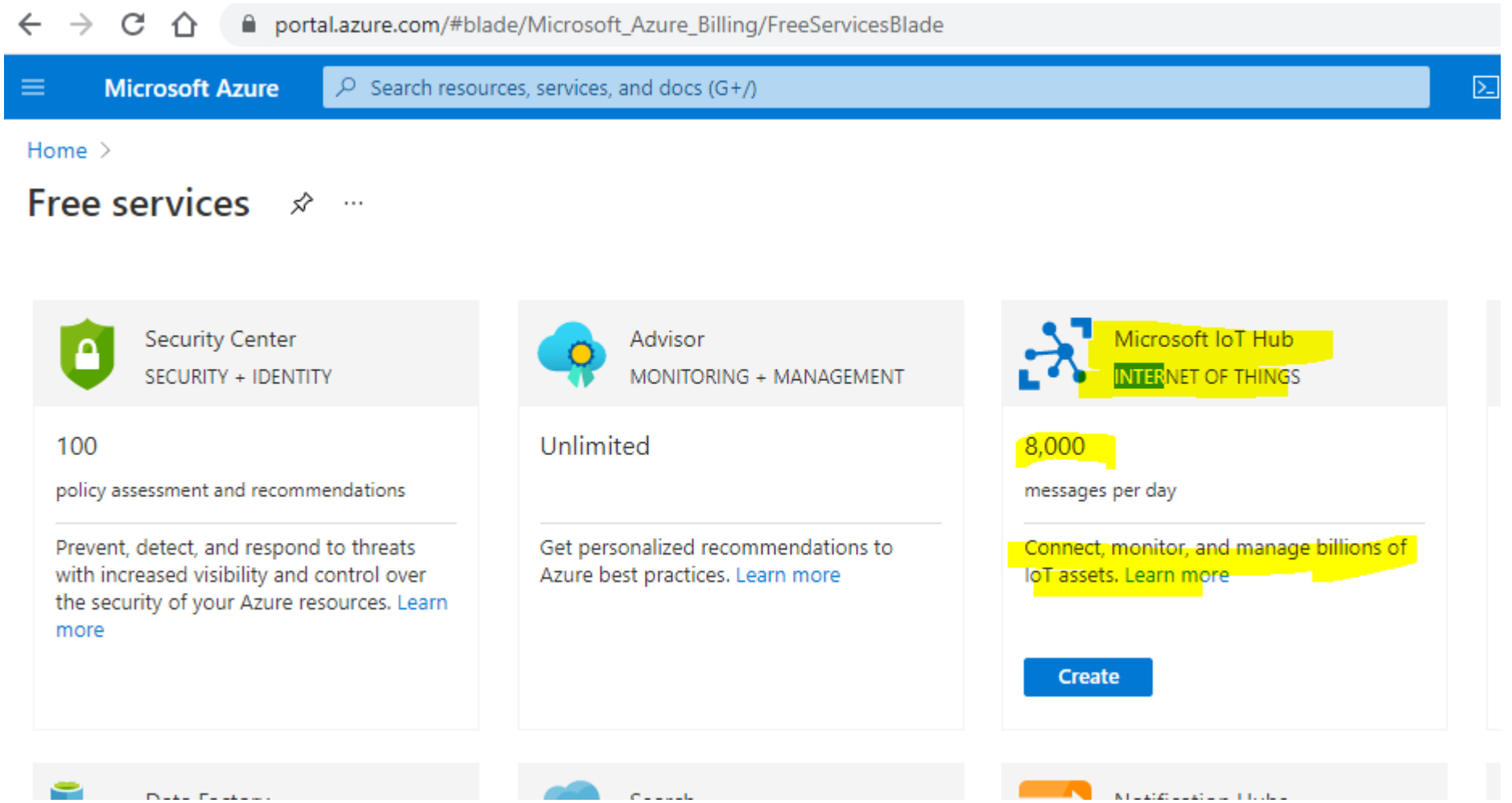


Azure free services



https://azureforeducation.microsoft.com/devtools

Create Microsoft IoT Hub



https://azureforeducation.microsoft.com/devtools

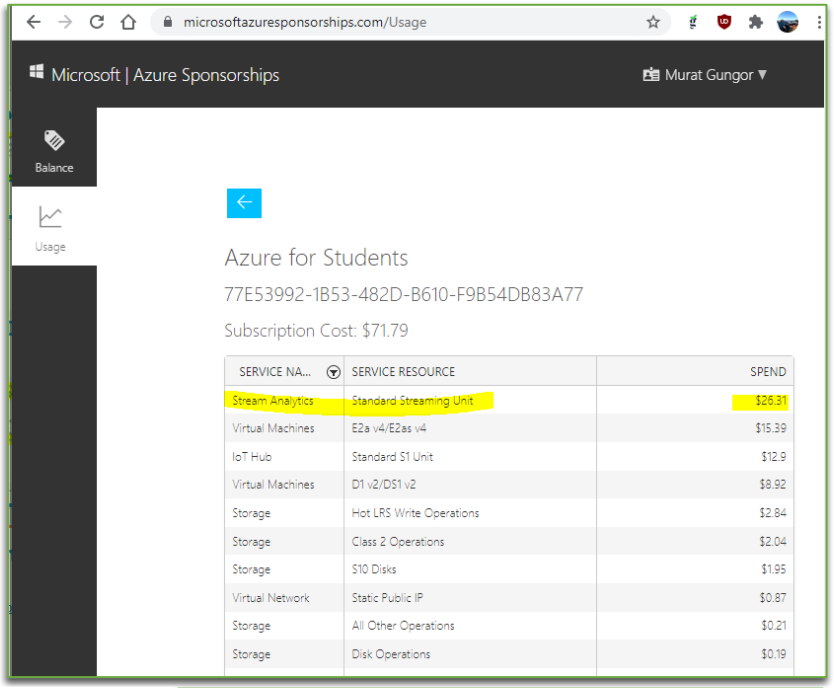
How to Check Your Account Balance

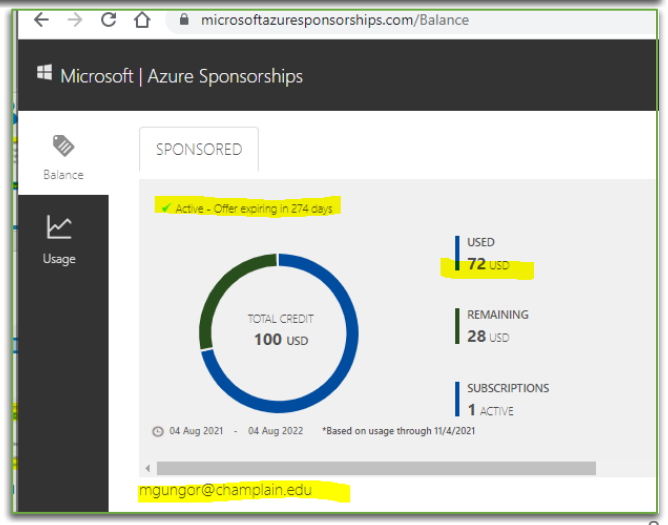
Microsoft | Azure Sponsorships

- https://www.microsoftazuresponsorships.com

- Check Your Balance
  - Get the most out of your sponsored account

- Usage Details
  - Drill down into the usage statistics of your subscriptions

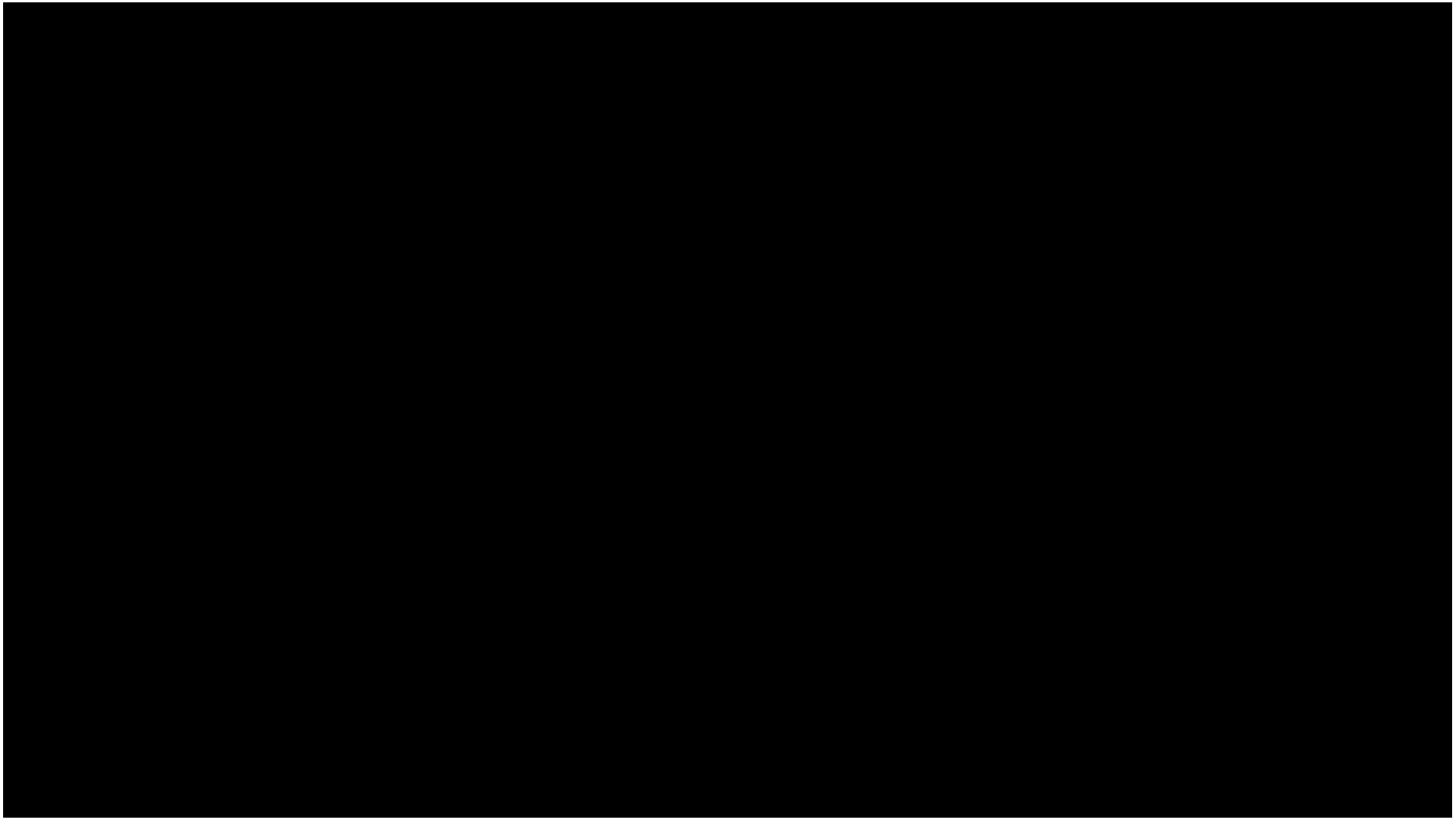




Connect your Device to the IoT service

- Once the hub is created,
  your IoT device can connect to it.
- Only added devices can connect to a service,
  so you will need to register your device first.

What is Azure IOT Hub?



https://youtu.be/eEIPnYLj73A

[Optional] – install the Azure CLI (Command Line Interface)

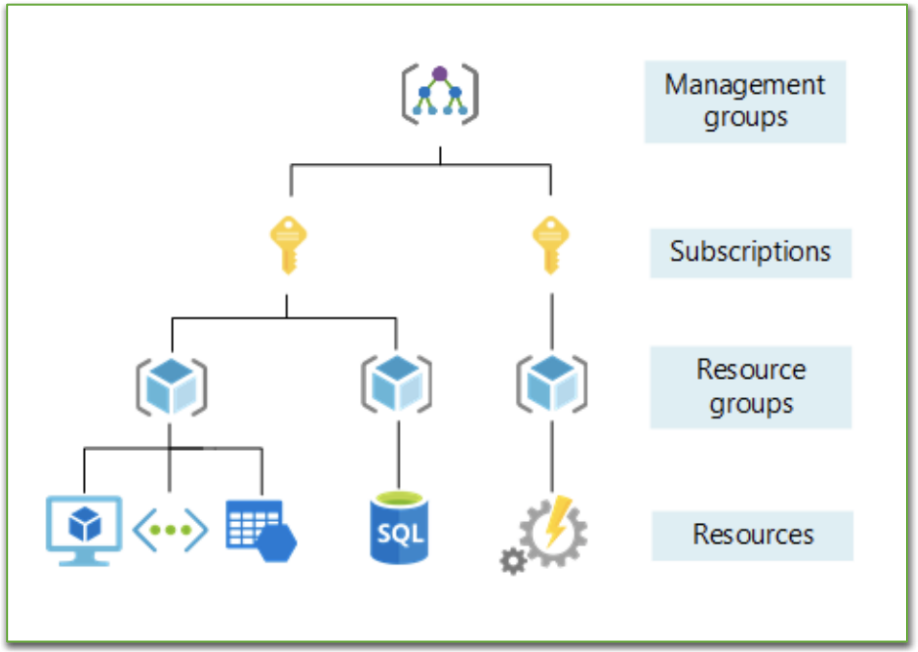
- Azure Command-Line Interface
  - https://docs.microsoft.com/en-us/cli/azure/
- Steps to Create an IoT service in the cloud
  - https://github.com/microsoft/IoT-For-Beginners/blob/main/2-farm/lessons/4-migrate-your-plant-to-the-cloud/README.md#task---install-the-azure-cli

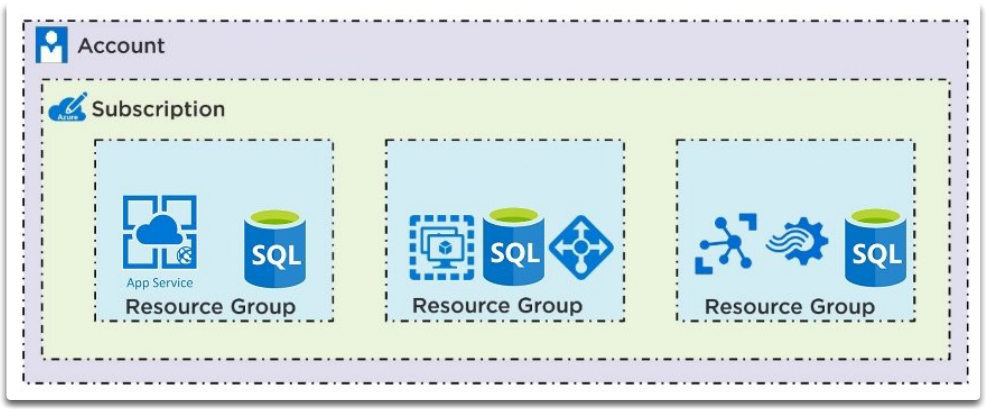
Create IoT Hub and Resource Group

What is Resource Group?

- Azure services, such as **IoT Hub** instances, virtual machines, databases, or AI services, are referred to as **resources**.

- **Every resource** has to **live inside** a **Resource Group**, a logical grouping of one or more resources.





- [Optional] If you like to user Azure CLI follow the steps at the link below
  - https://github.com/microsoft/IoT-For-Beginners/blob/main/2-farm/lessons/4-migrate-your-plant-to-the-cloud/README.md#task---create-a-resource-group

Name your IoT Hub and Resource Group

This will take several minutes
be patient!

Mine took ~4 Minutes

… Deployment is in progress

IoT hub

Microsoft

Basics | Networking | Management | Tags | Review + create

Create an IoT hub to help you connect, monitor, and manage billions of your IoT assets. Learn more

Project details

Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription * : Azure for Students

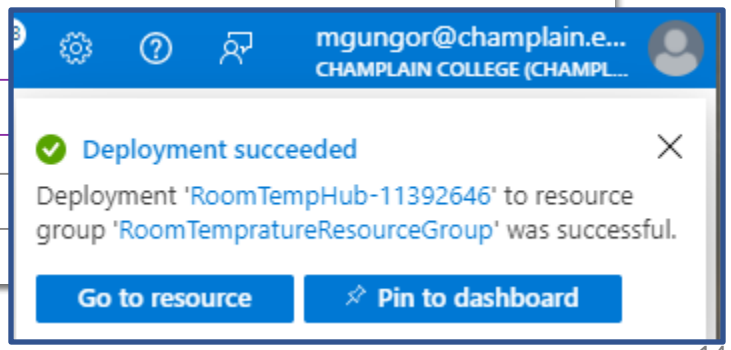
Resource group * : RoomTempratureResourceGroup

Create new

Instance details

IoT hub name * : RoomTempHub

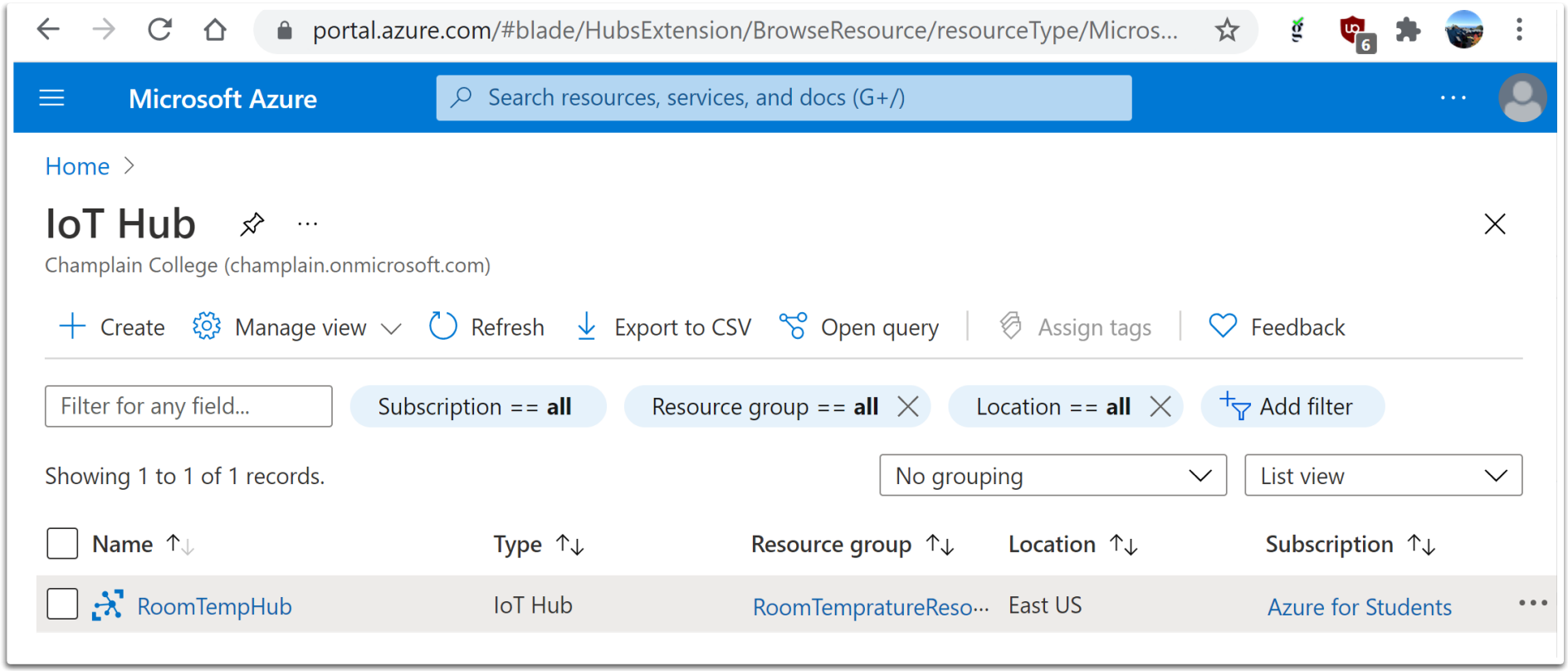
Region * : East US



Completed! IoT Hub

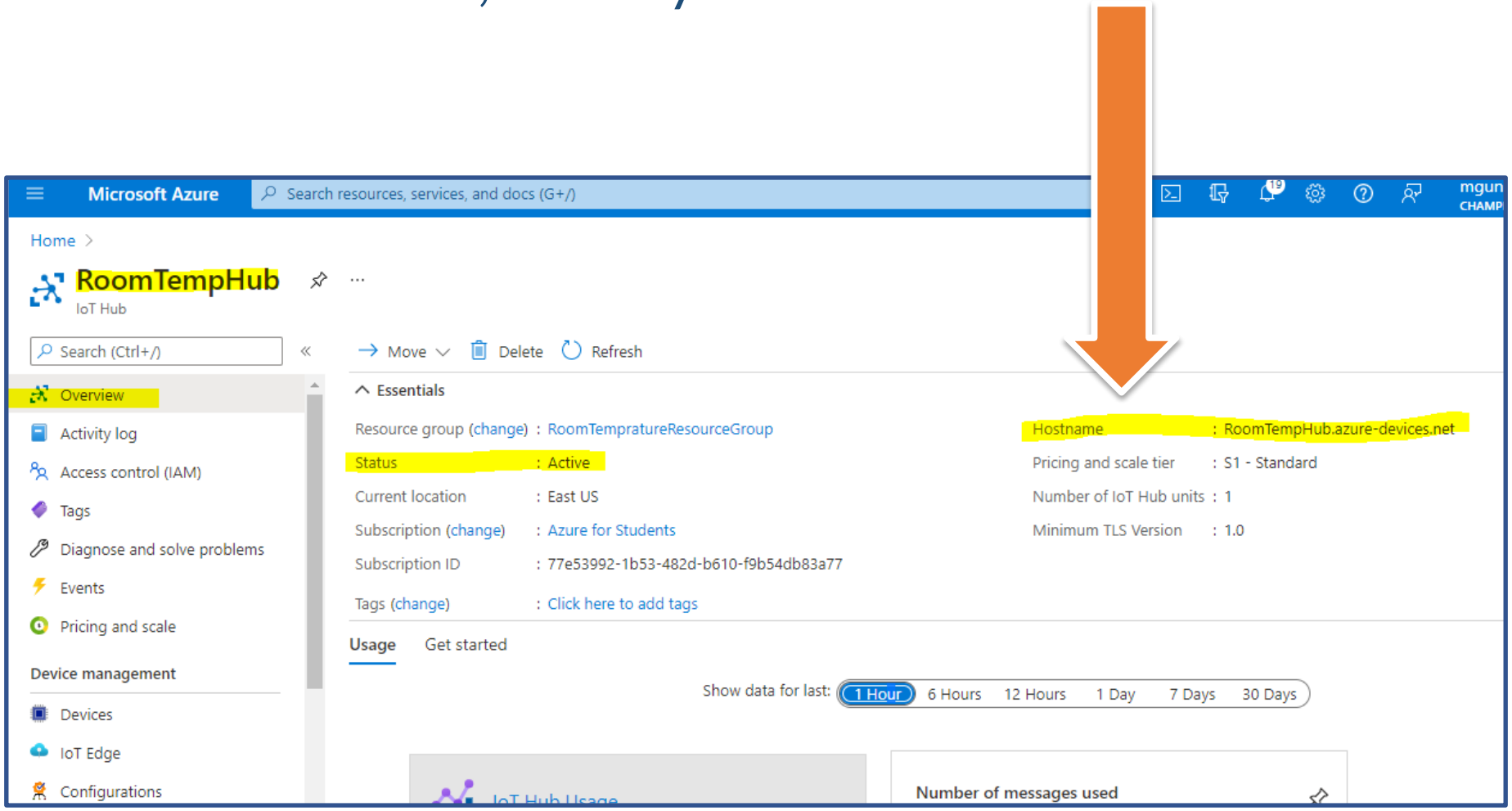
Deployment succeeded

- You can now **have** an IoT Hub in your resource group.



Navigate to your hub and look for the Hostname

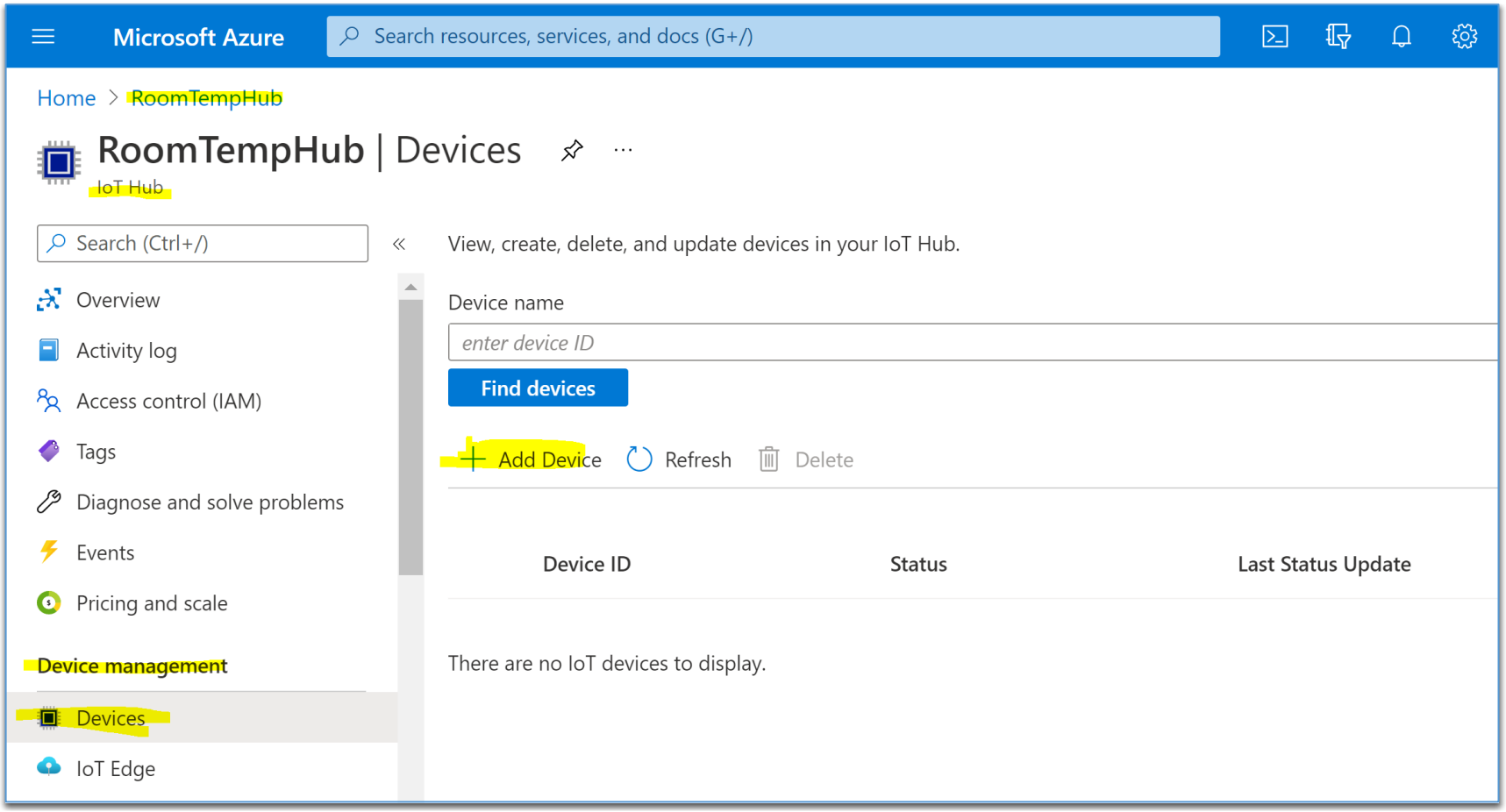
- The Hostname, this is your **broker**!

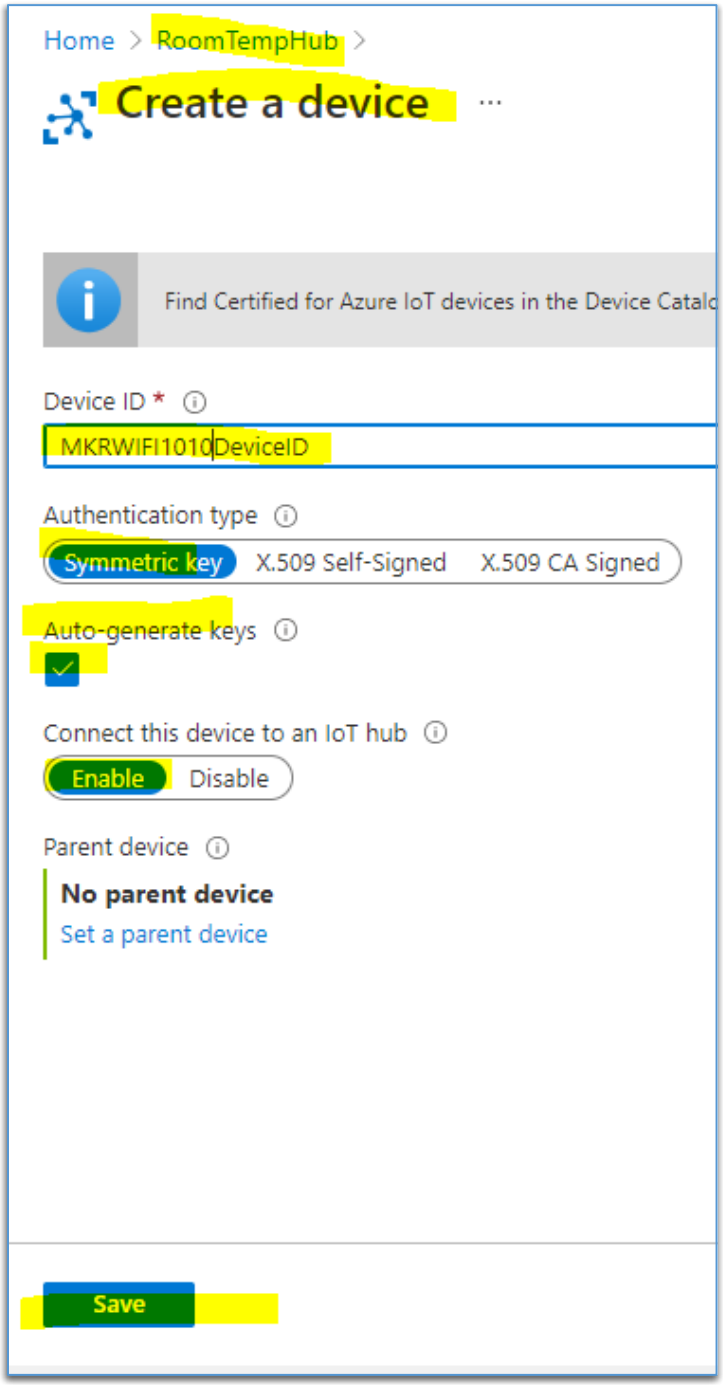


Add Device

Task – Add Device to IoT Hub

Next we need to provision a device, look for “Devices” along the left side and select “Add Device”



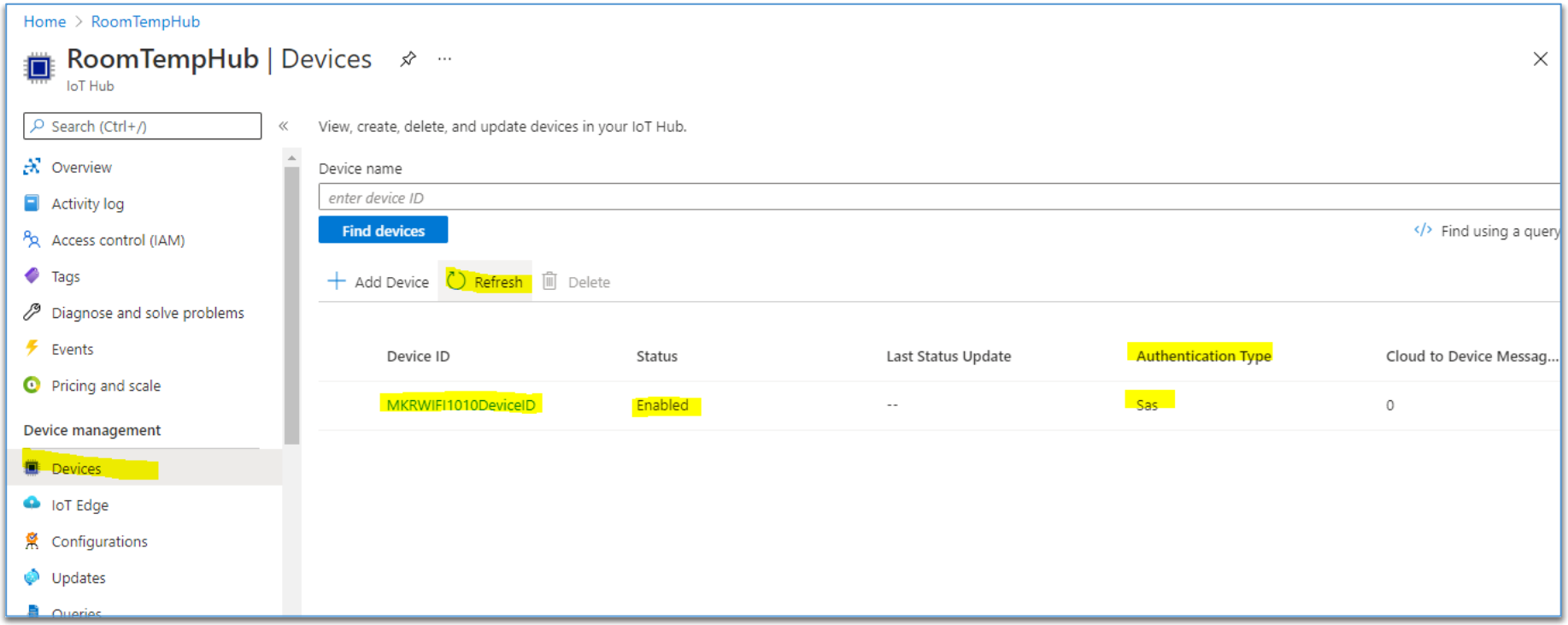


Adding a Device

- Name the device,
- Leave all other settings as default; authentication as “Symmetric key” and
- Leave “Auto-generate keys” checked,
- “Connect this device to an IoT Hub” enabled
- and save.
- Your Device ID is the **MKRWIFI1010DeviceID**.

Device Added!

- Refresh to see the newly added device
- Your Device ID is the **MKRWIFI1010DeviceID**



Coding Part 1
Sending text message

“arduino_secrets.h”

- This contains information for connecting to
  - Wi-Fi Info => WE GOT THIS
  - MQTT broker => WE GOT THIS
  - Device ID => WE GOT THIS
  - Device Pass => WE NEED THIS

platformio.ini | main.cpp | arduino_secrets.h | Ex

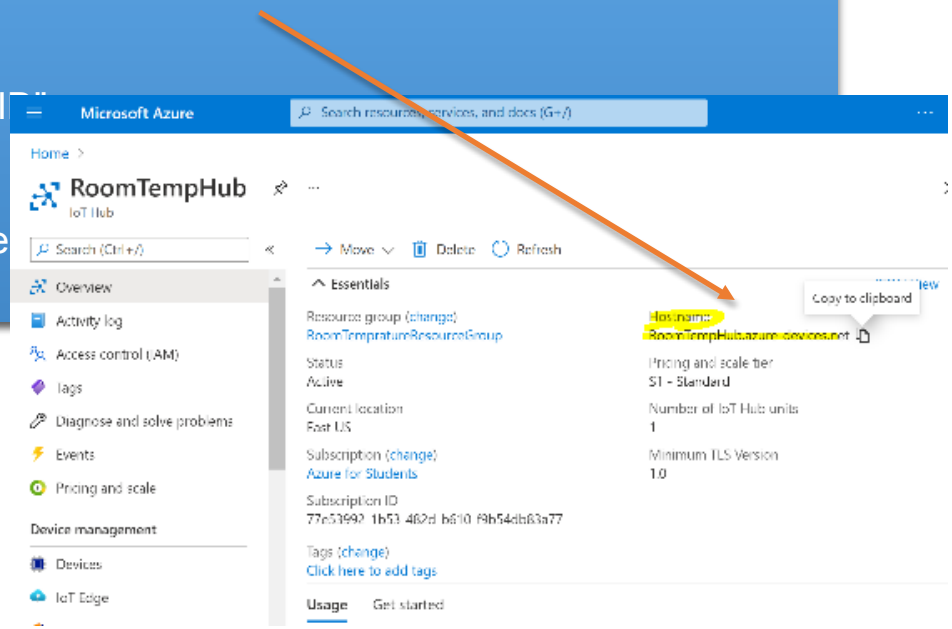
2021Nov2Azure > src > main.cpp > ...

```
24
25    #include "arduino_secrets.h"
26
27    /////// Enter your sensitive data in arduino_secrets
28    const char ssid[]        = SECRET_WIFI_SSID;
29    const char pass[]        = SECRET_WIFI_PASS;
30    const char broker[]      = SECRET_BROKER;
31    String     deviceId      = SECRET_DEVICE_ID;
32    String     devicePass    = SECRET_DEVICE_PASSWORD;
33
```

Partial My: arduino_secrets.h

- Following is the secrets file content
- Except WIFI info the rest should be different in your file

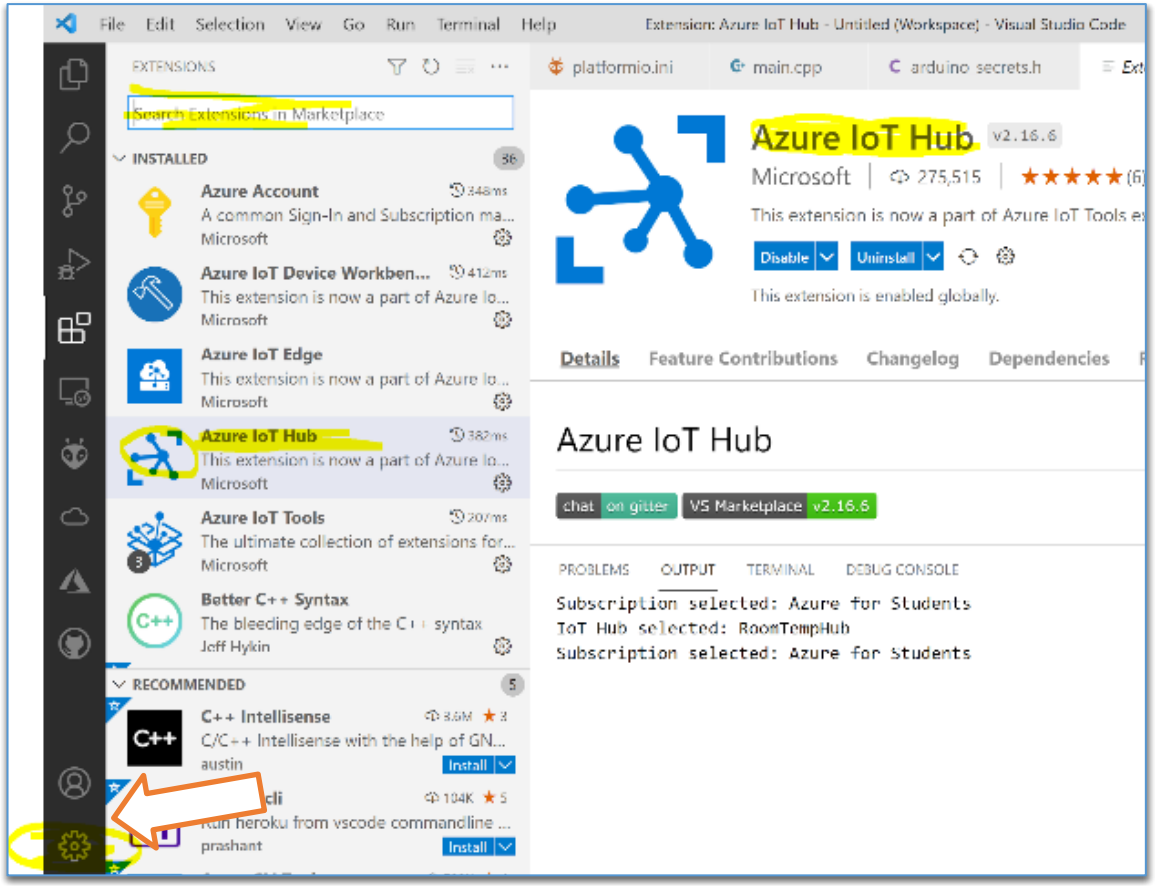
```
// Fill in  your WiFi networks SSID and password
#define SECRET_WIFI_SSID      "ChamplainPSK"
#define SECRET_WIFI_PASS      "letusdare"

// Fill in the hostname of your Azure IoT Hub broker
#define SECRET_BROKER    "RoomTempHub.azure-devices.net"

// Fill in the device id
#define SECRET_DEVICE_ID  "MKRWIFI1010DeviceI
// Fill in the device passoword
#define SECRET_DEVICE_PASSWORD "SharedAcce
```



Shared Access Signature (SAS) as Device Password

Install – Azure IoT Hub Extension

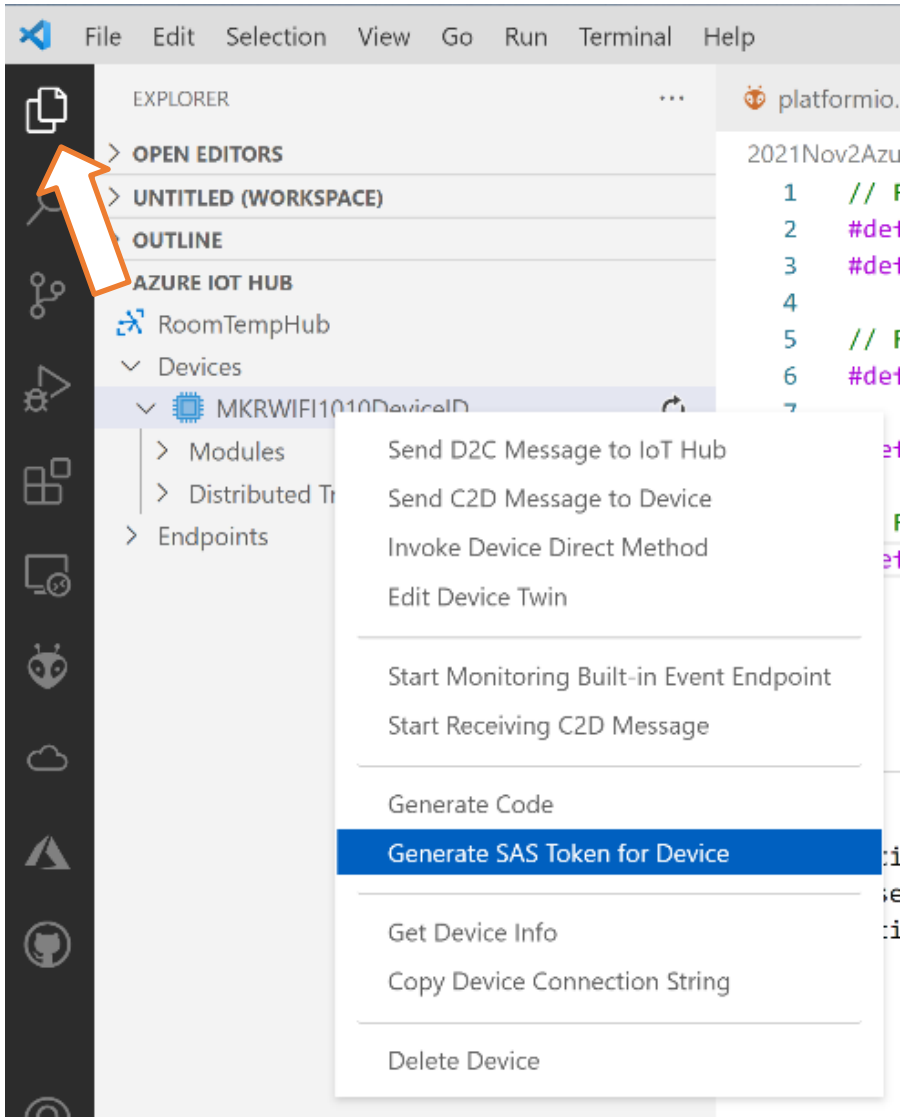
- We will use this extension to generate Shared Access Signature (SAS) as Device Password
- Installing Steps: Visual Studio Code / Manage / Extensions (Ctrl + Shift + X)



Once installed

- Login to Azure using VS Code,
- Then in the **Explorer** pane you'll be able to select **Azure IoT Hub**,
- Locate
  - your Hub,
  - the Device you created,
- Right click and Generate SAS Token

Creating the Device Password



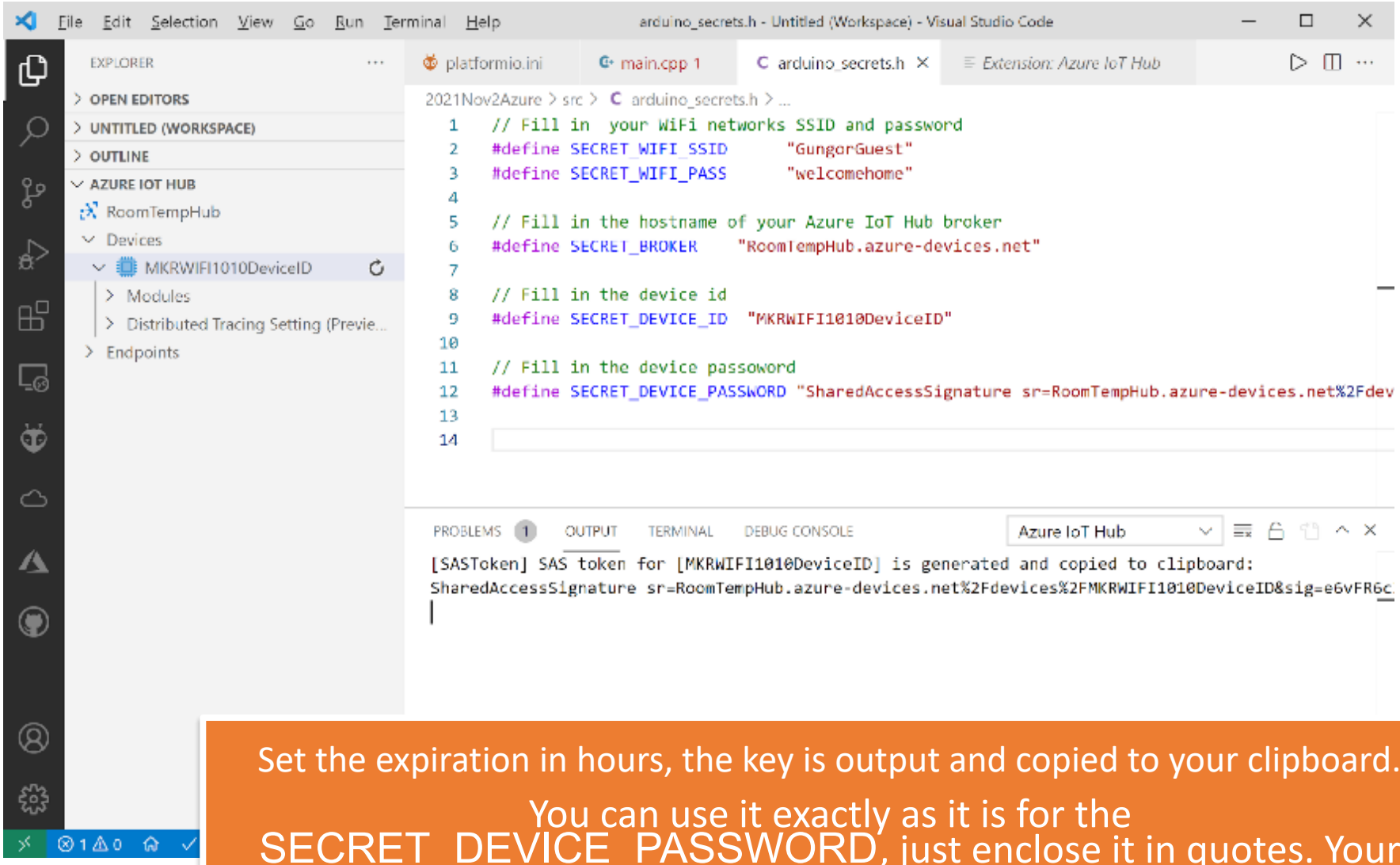
Click on

- Explorer
- AZURE IOT HUB
- Devices
- Right Click on a device and click **Generate SAS Token for Device**

SAS:

Shared Access Signature as Device Password

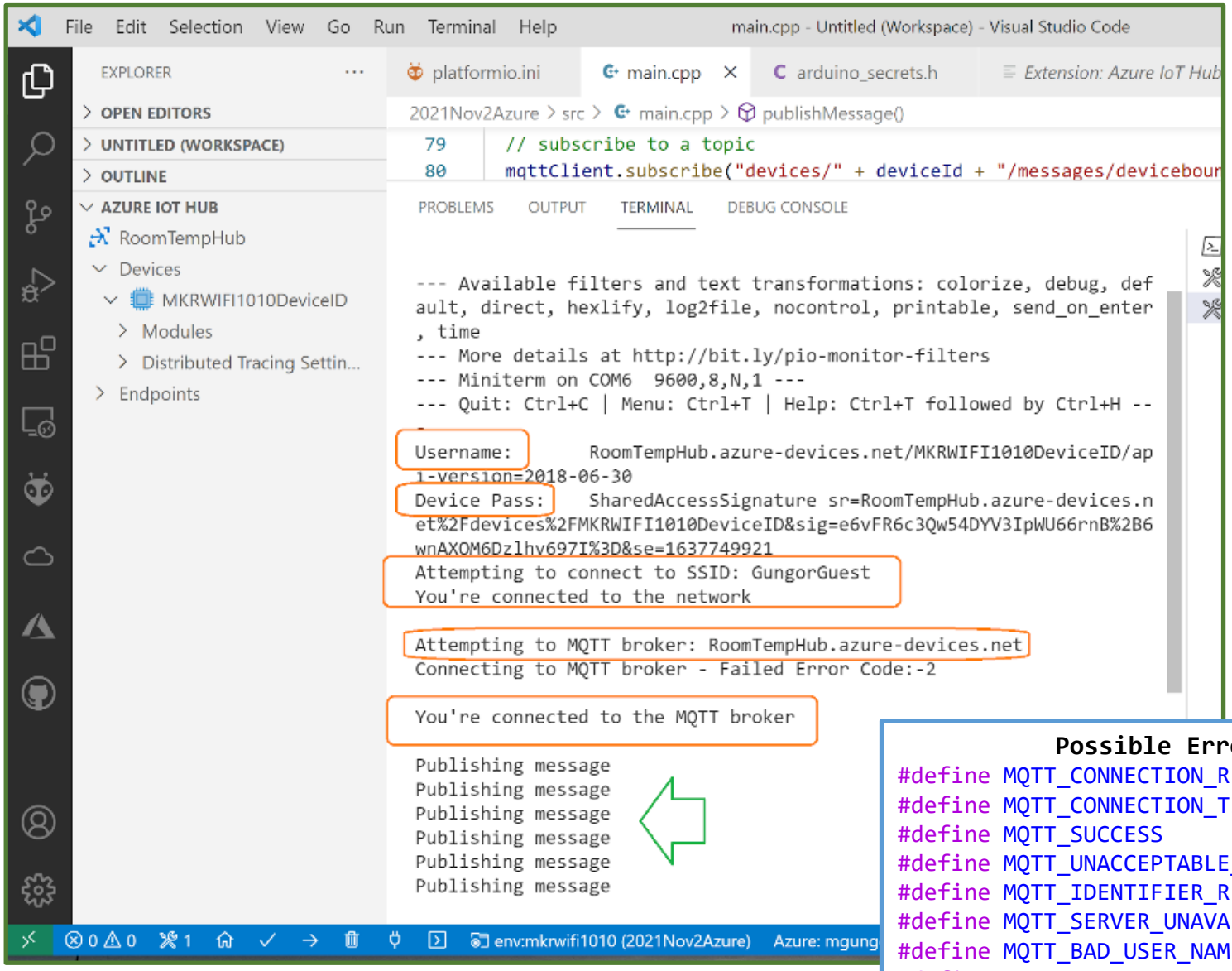
Generated SAS Token as Device Password



Set the expiration in hours, the key is output and copied to your clipboard.

You can use it exactly as it is for the SECRET_DEVICE_PASSWORD, just enclose it in quotes. Your code should include "SharedAccessSignature sr=", do not omit this.

Running of the application

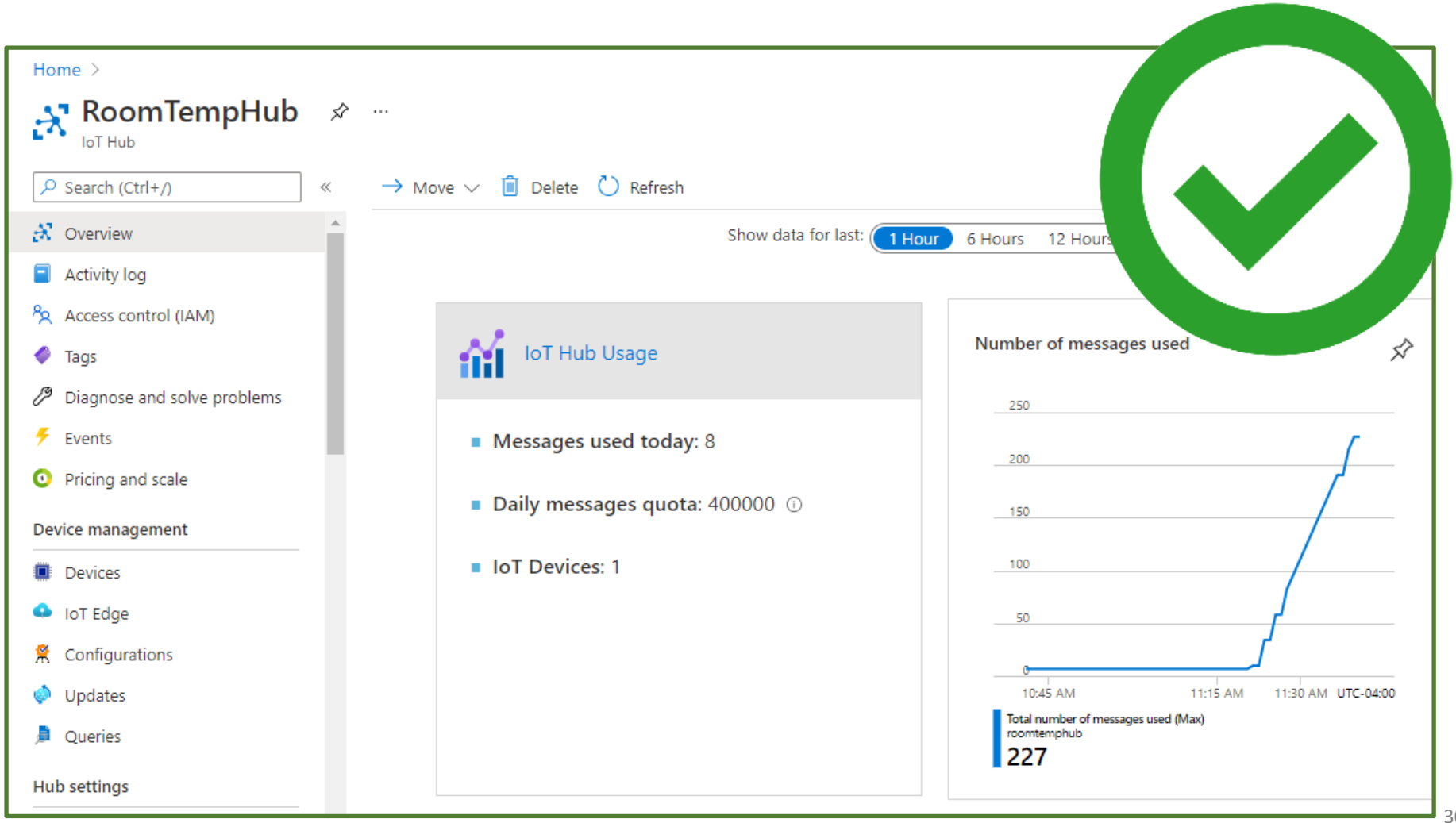


Possible Error Messages

```
#define MQTT_CONNECTION_REFUSED            -2
#define MQTT_CONNECTION_TIMEOUT            -1
#define MQTT_SUCCESS                        0
#define MQTT_UNACCEPTABLE_PROTOCOL_VERSION  1
#define MQTT_IDENTIFIER_REJECTED            2
#define MQTT_SERVER_UNAVAILABLE             3
#define MQTT_BAD_USER_NAME_OR_PASSWORD      4
#define MQTT_NOT_AUTHORIZED                 5
```

View from the IoT Hub

- Yes! We are now sending some data to Azure!

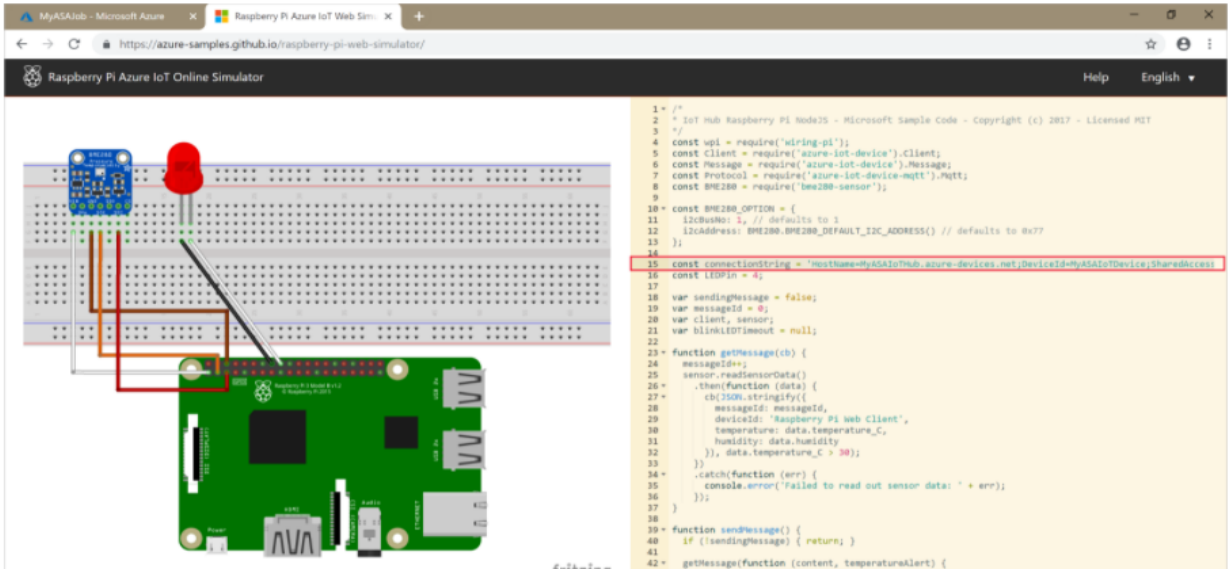


[In case]Troubleshooting:
Run the IoT simulator to make sure you can connect

- Link: https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-quick-create-portal

Run the IoT simulator

1. Open the Raspberry Pi Azure IoT Online Simulator.
2. Replace the placeholder in Line 15 with the Azure IoT Hub device connection string you saved in a previous section.
3. Click **Run**. The output should show the sensor data and messages that are being sent to your IoT Hub.



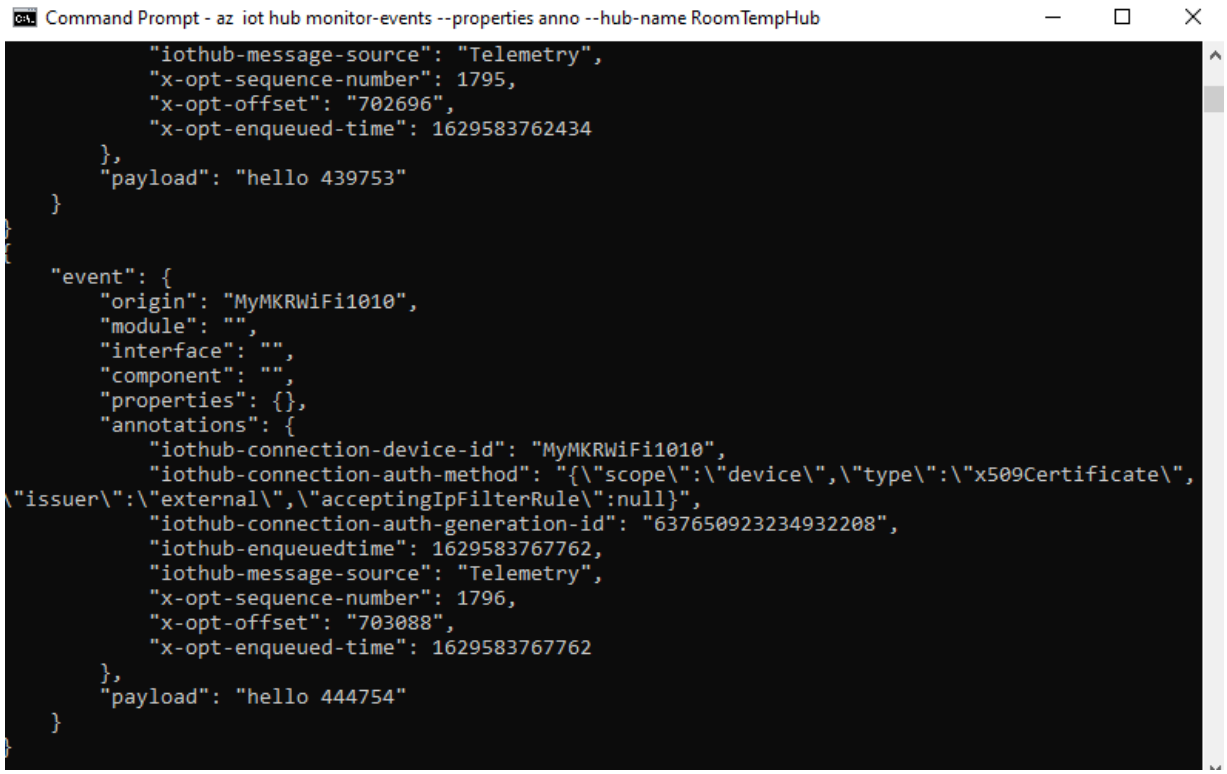
1.Open the Raspberry Pi Azure IoT Online Simulator.

[Optional] Monitor Events via Azure CLI

[Optional]
Monitor Messages via Command Line Interface

- Run the following command in your command prompt or terminal to monitor messages sent to your IoT Hub:

```
az iot hub monitor-events --properties anno --hub-name <hub_name>
```



The time values in the annotations are in UNIX time, representing the number of seconds since midnight on 1st January 1970.

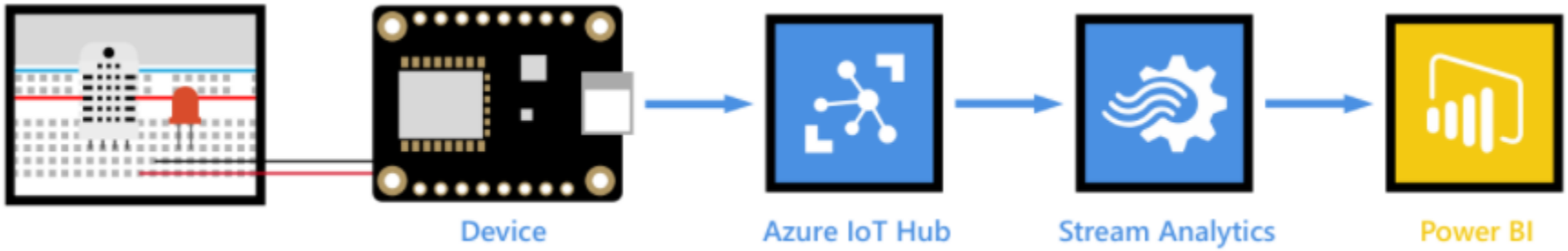
Exit the event monitor when you are done.

https://www.epochconverter.com/

Where are we now?

Arduino MKR WIFI 1010 to Azure IoT Hub

- Connect a Arduino MKR WIFI 1010 board securely to Azure IoT Hub using an MQTT client. => **COMPLETED**
- Create a consumer group on your IoT hub.
- Create and configure an **Azure Stream Analytics** job to read temperature telemetry from your consumer group and send it to **Power BI**.
- Create a report of the temperature data in **Power BI** and share it to the web.



https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi

Visualize real-time sensor data from Azure IoT Hub

Reference
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi

Coding Part 2

Sending
temperature and humidity telemetry

Coding Side – Sending Telemetry Data

main.cpp | platformio.ini | arduino_secrets.h

2021Nov2Azure > src > main.cpp > ...

```
25
26  #include <Arduino_MKRIoTCarrier.h> // IOT CARRIER
27  #include <ArduinoJson.h> // JSON
28
29  #include "arduino_secrets.h"
30
31  /////// Enter your sensitive data in arduino_secrets.h
32  const char ssid[]        = SECRET_WIFI_SSID;
33  const char pass[]        = SECRET_WIFI_PASS;
34  const char broker[]      = SECRET_BROKER;
35  String     deviceId      = SECRET_DEVICE_ID;
36  String     devicePass    = SECRET_DEVICE_PASSWORD;
37
38  WiFiClient    wifiClient;            // Used for the TCP socket connection
39  BearSSLClient sslClient(wifiClient); // Used for SSL/TLS connection, integrate
40  MqttClient    mqttClient(sslClient);
41
42  // IOT Carrier
43  MKRIoTCarrier carrier;
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
Publishing message      26.57 :Temperature <=> Humidity :38.88
{"messageId":280,"deviceId":"MKR WiFI 1010 Client","temperature":26.5662,"humidity"
:38.88189}
Publishing message      26.60 :Temperature <=> Humidity :38.86
{"messageId":281,"deviceId":"MKR WiFI 1010 Client","temperature":26.60282,"humidity
":38.85979}
Publishing message      26.57 :Temperature <=> Humidity :38.88
{"messageId":282,"deviceId":"MKR WiFI 1010 Client","temperature":26.5662,"humidity"
```


platformio.ini


arduino_secrets.h


main.cpp

- This time we are using JSON format to send telemetry messages.
- publishTemperatureHumidity() function added
- IoT Carrier code changes added.

Forming JSON Message to Send

```
void publishTemperatureHumidity() {
  Serial.print("Publishing message\t");

  // Read Sensor Data
  float temperature = carrier.Env.readTemperature();
  float humidity = carrier.Env.readHumidity();
  …
  DynamicJsonDocument doc(1024);
  doc["messageId"] = meesageId++;
  doc["deviceId"] = "MKR WiFI 1010 Client";
  doc["temperature"] = temperature;
  doc["humidity"] = humidity;

  string telemetry;
  serializeJson(doc, telemetry);

  Serial.println(telemetry.c_str());

  // send message, the Print interface can be used to set the message contents
  mqttClient.beginMessage("devices/" + deviceId + "/messages/events/");
  mqttClient.print(telemetry.c_str());
  mqttClient.endMessage();
}
```

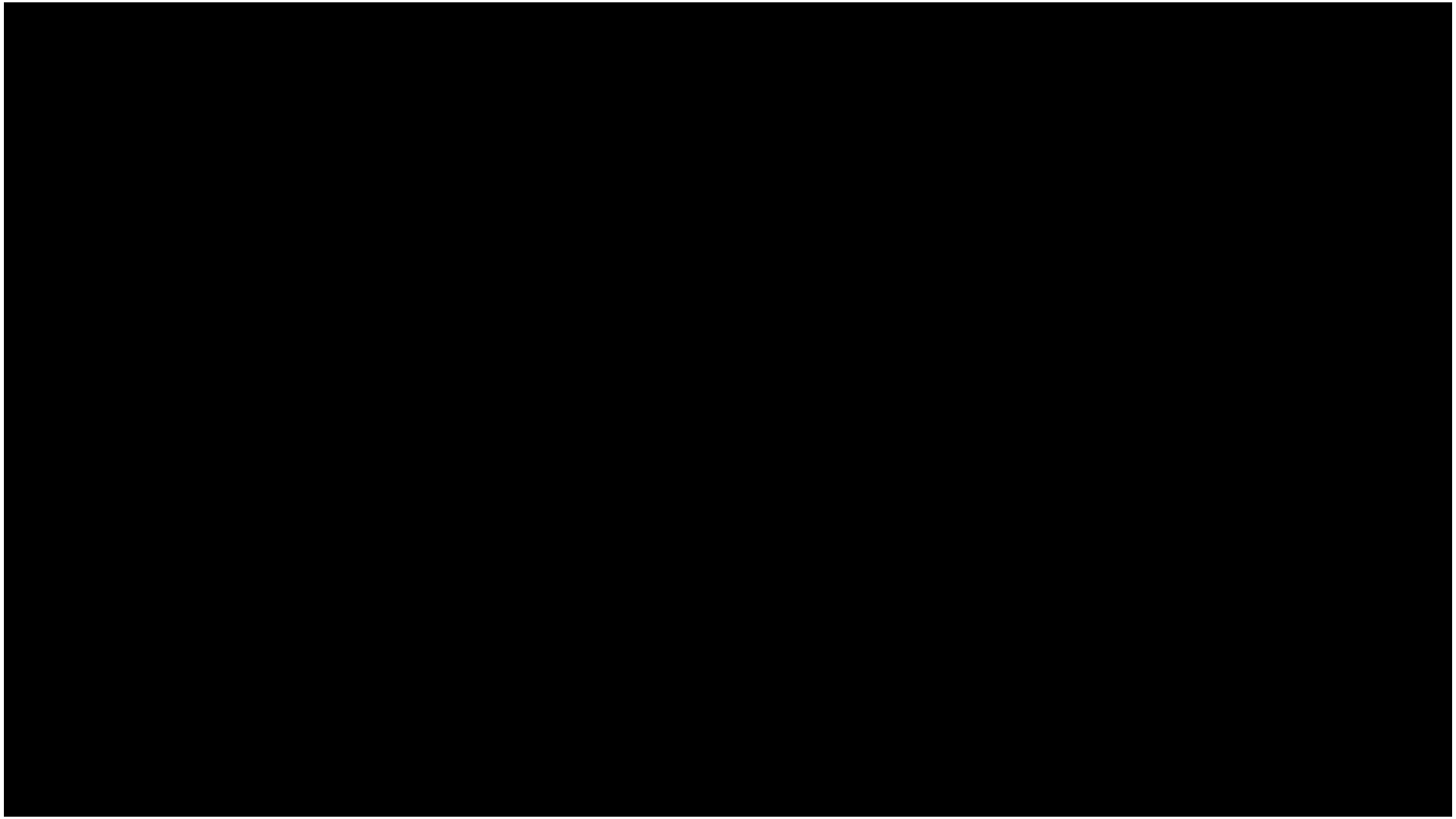
Reading telemetry from sensors

Forming JSON payload

Sending the message

QuickStart:
Create a Stream Analytics job by using the Azure portal

What Is Azure Stream Analytics?



https://www.youtube.com/watch?v=po-R-XsSZnM

Visualize real-time sensor data

- To visualize real-time sensor data that your **Azure IoT hub** receives.
- To do so, you configure an **Azure Stream Analytics** job to consume the data from IoT Hub
- and route it to a dataset in **Power BI**.

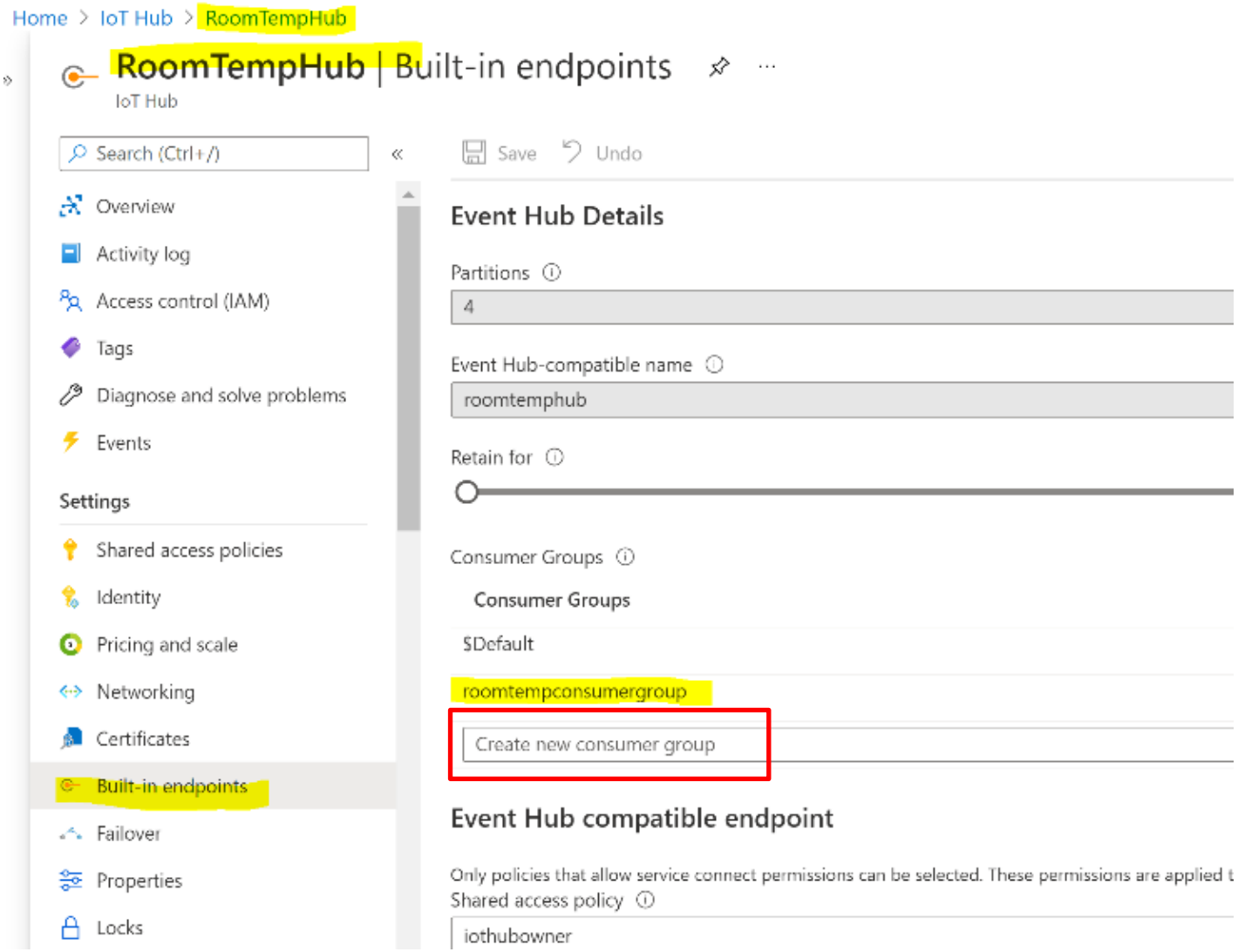
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi

Add a consumer group to your IoT hub

To add a consumer group to your IoT hub, follow these steps:

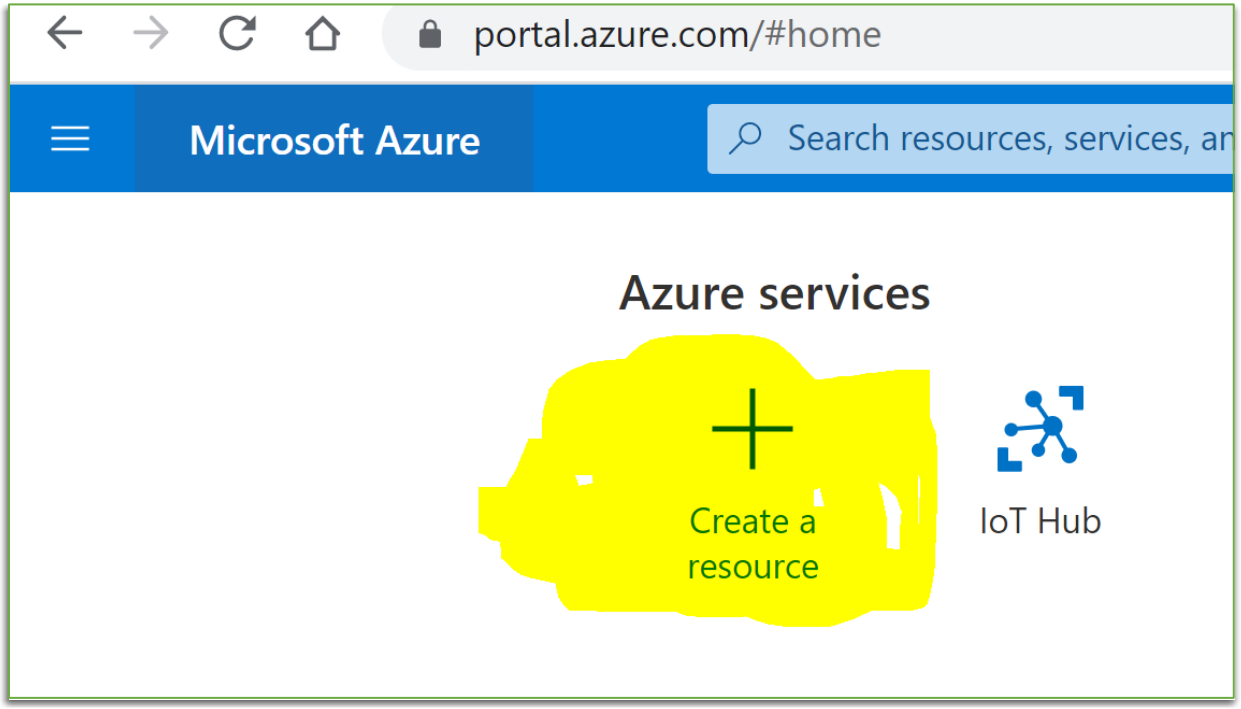
1. In the Azure portal, open your IoT hub.
2. On the left pane, select **Built-in endpoints**. Enter a name for your new consumer group in the text box under **Consumer groups**.
3. Click anywhere outside the text box to save the consumer group.

Create a new Consumer Group

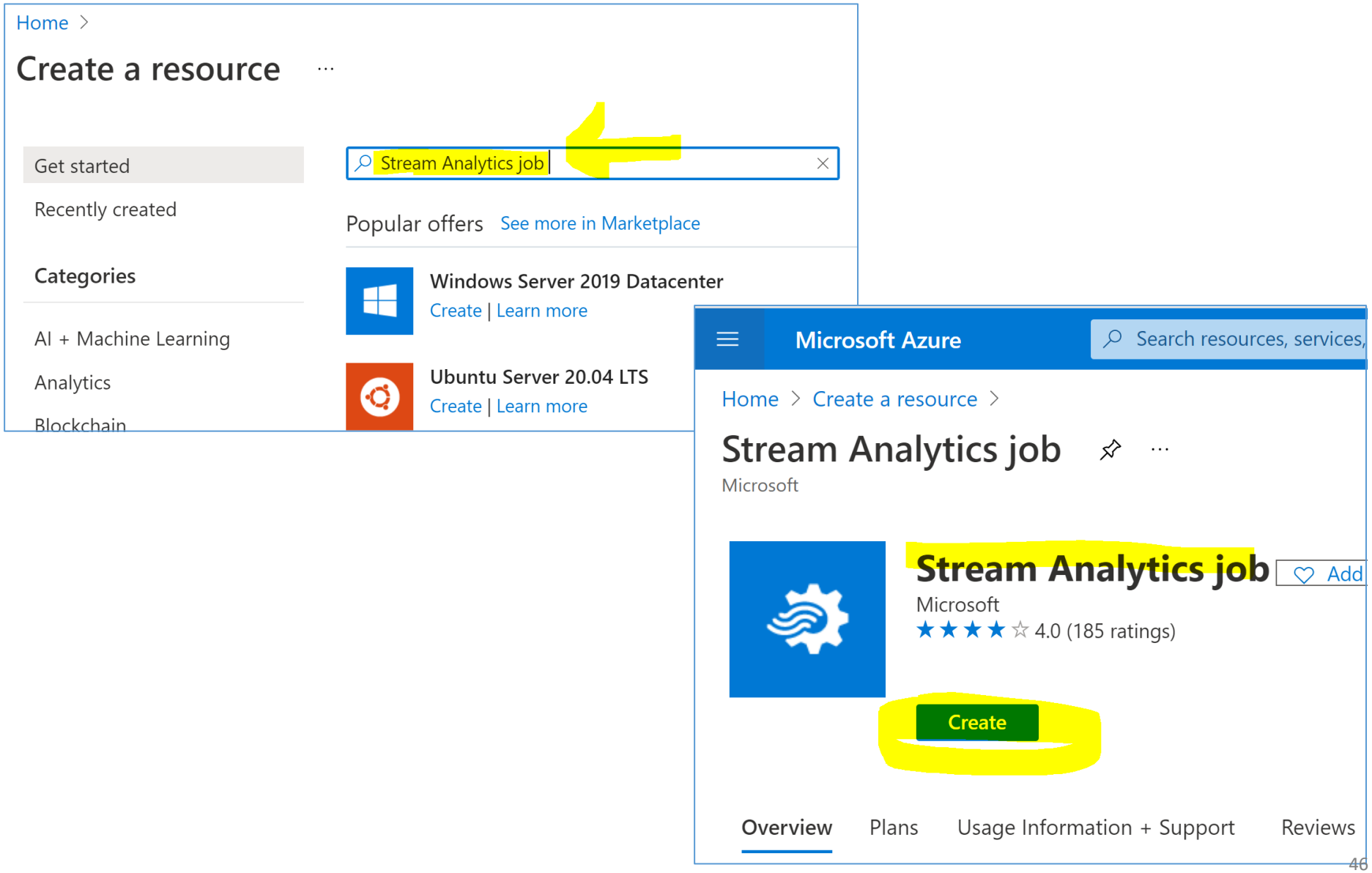


Create, configure, and run a Stream Analytics job

1. In the Azure portal, select **Create a resource**. Type *Stream Analytics Job* in the search box and select it from the drop-down list. On the **Stream Analytics job** overview page, select **Create**

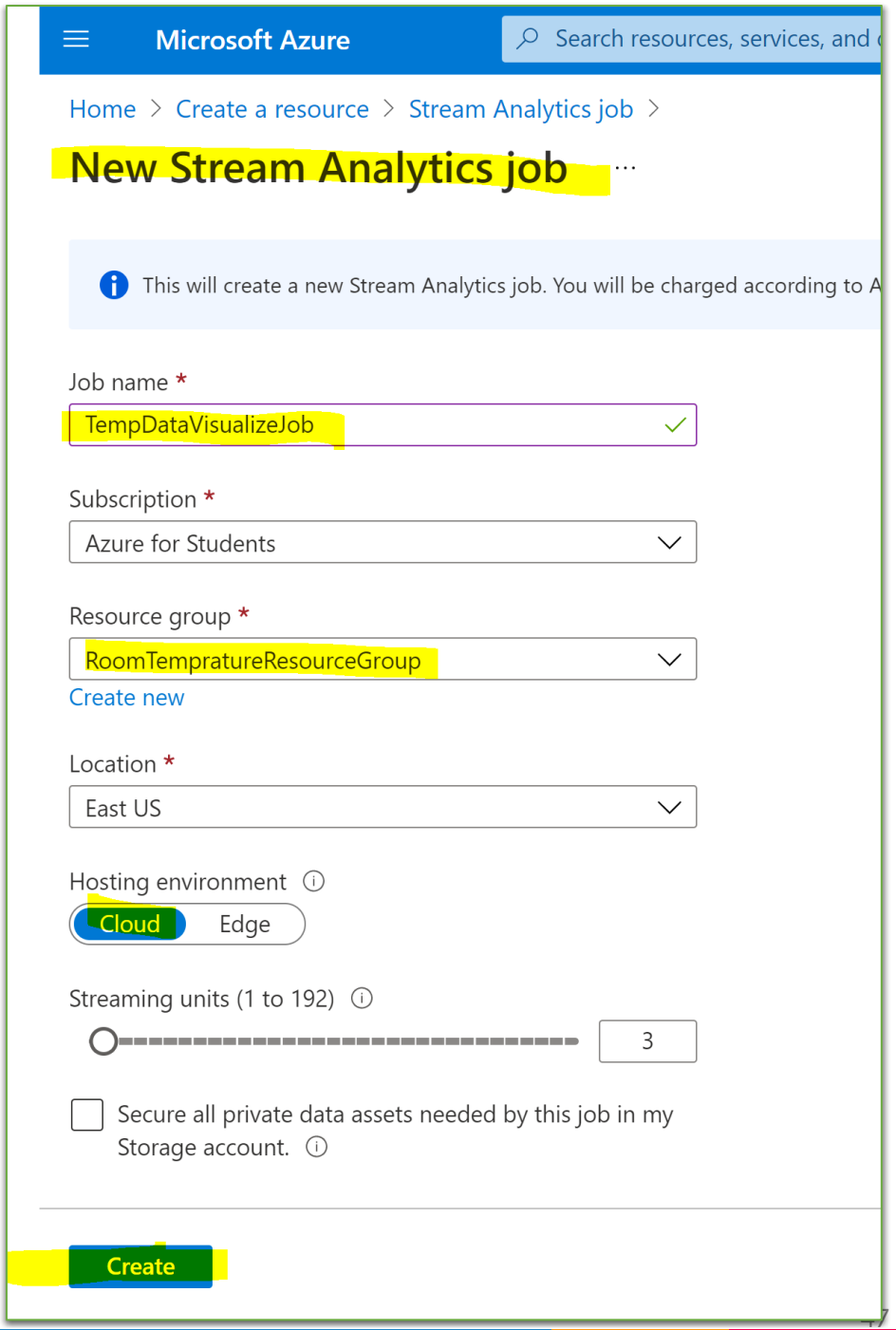


Search: Stream Analytics job

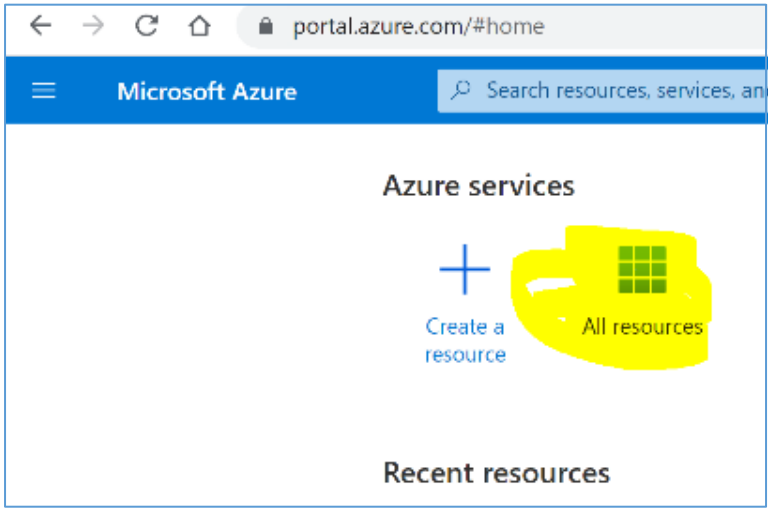


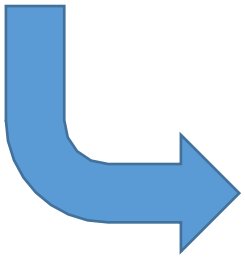
Next

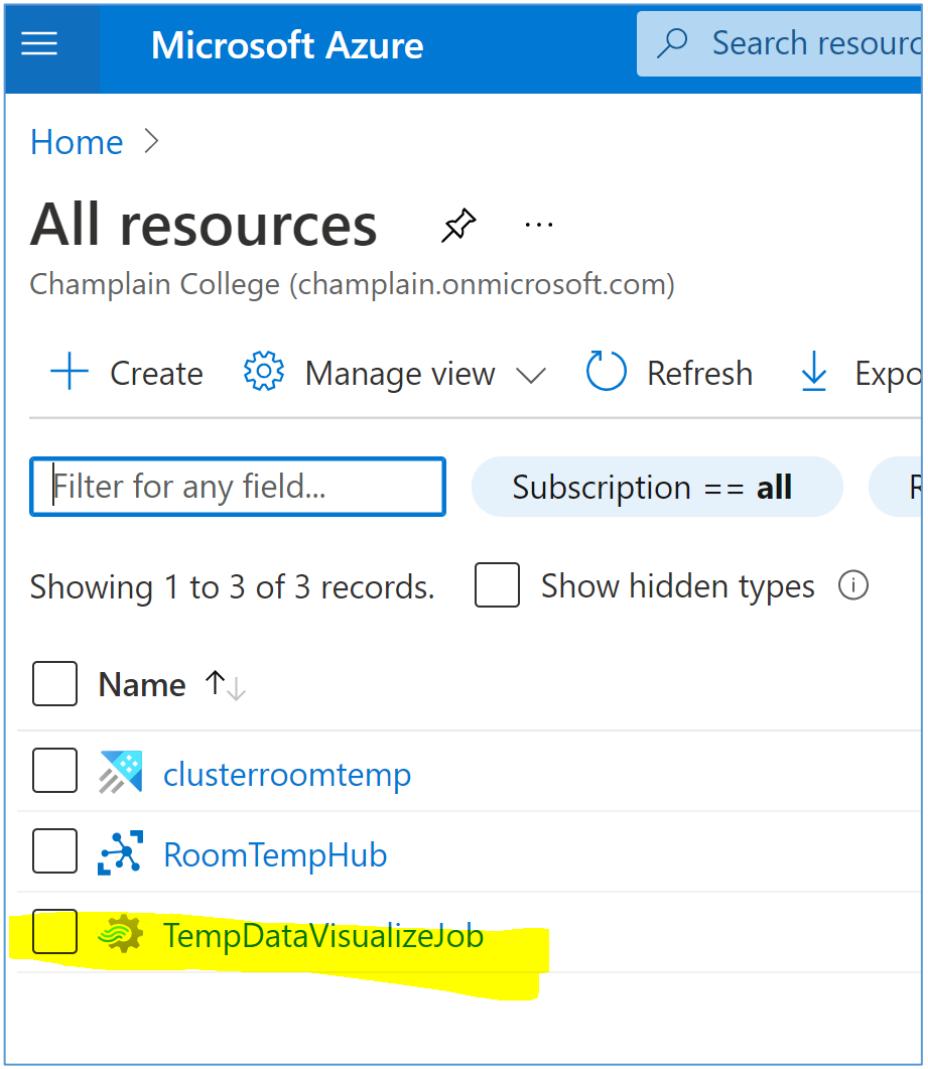
Be Patient
Deployment is in progress



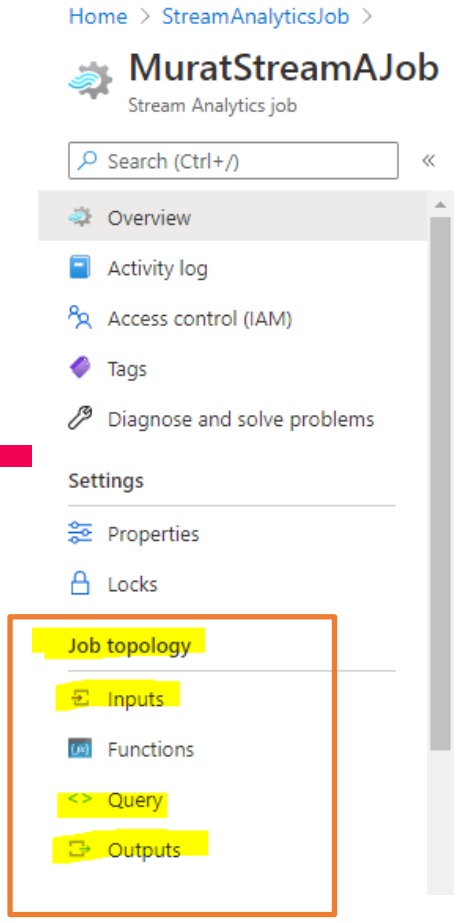
Done! Check it out:
You have now the Stream Analytics job



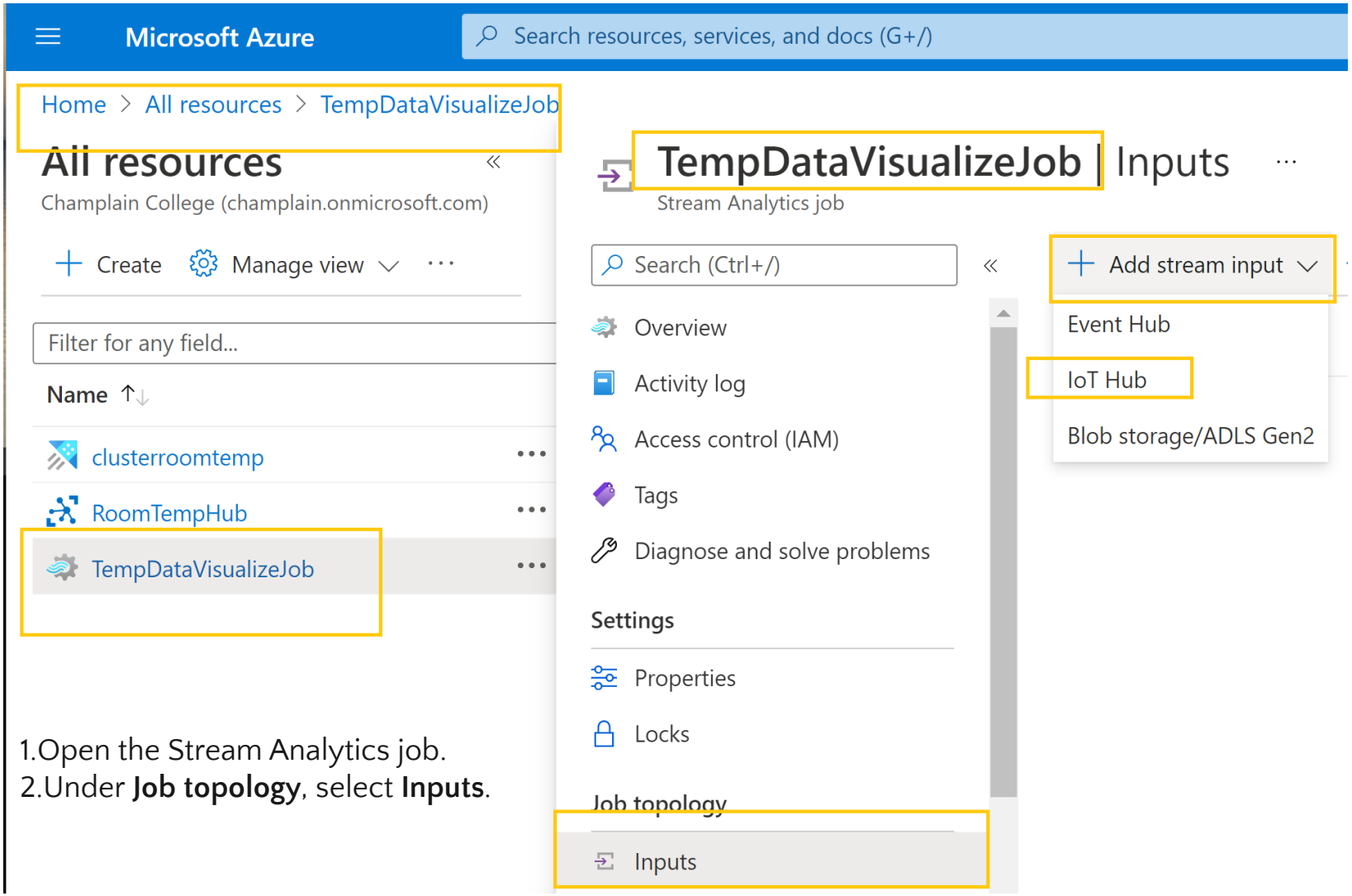




Add Stream Input & Add Stream Output Add Query



Select **Add stream input**, then select **IoT Hub** from the drop-down list

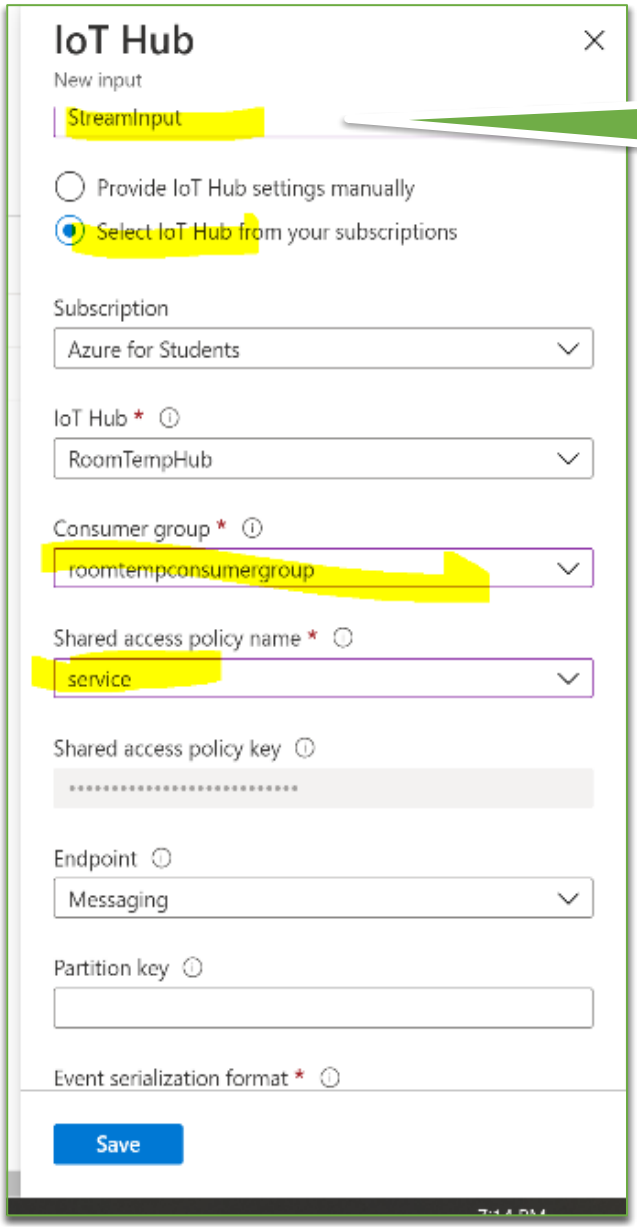


1. Open the Stream Analytics job.
2. Under **Job topology**, select **Inputs**.

1

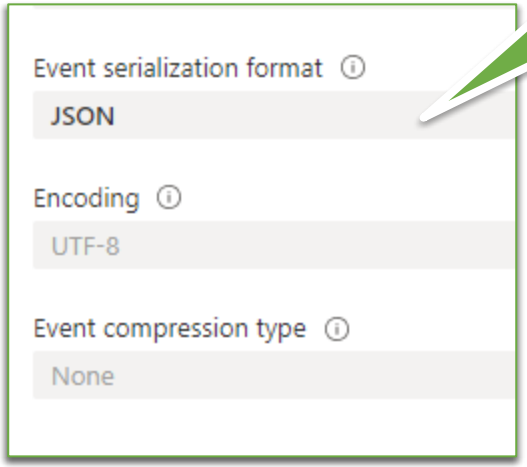
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi#add-an-input-to-the-stream-analytics-job

Add Stream Input



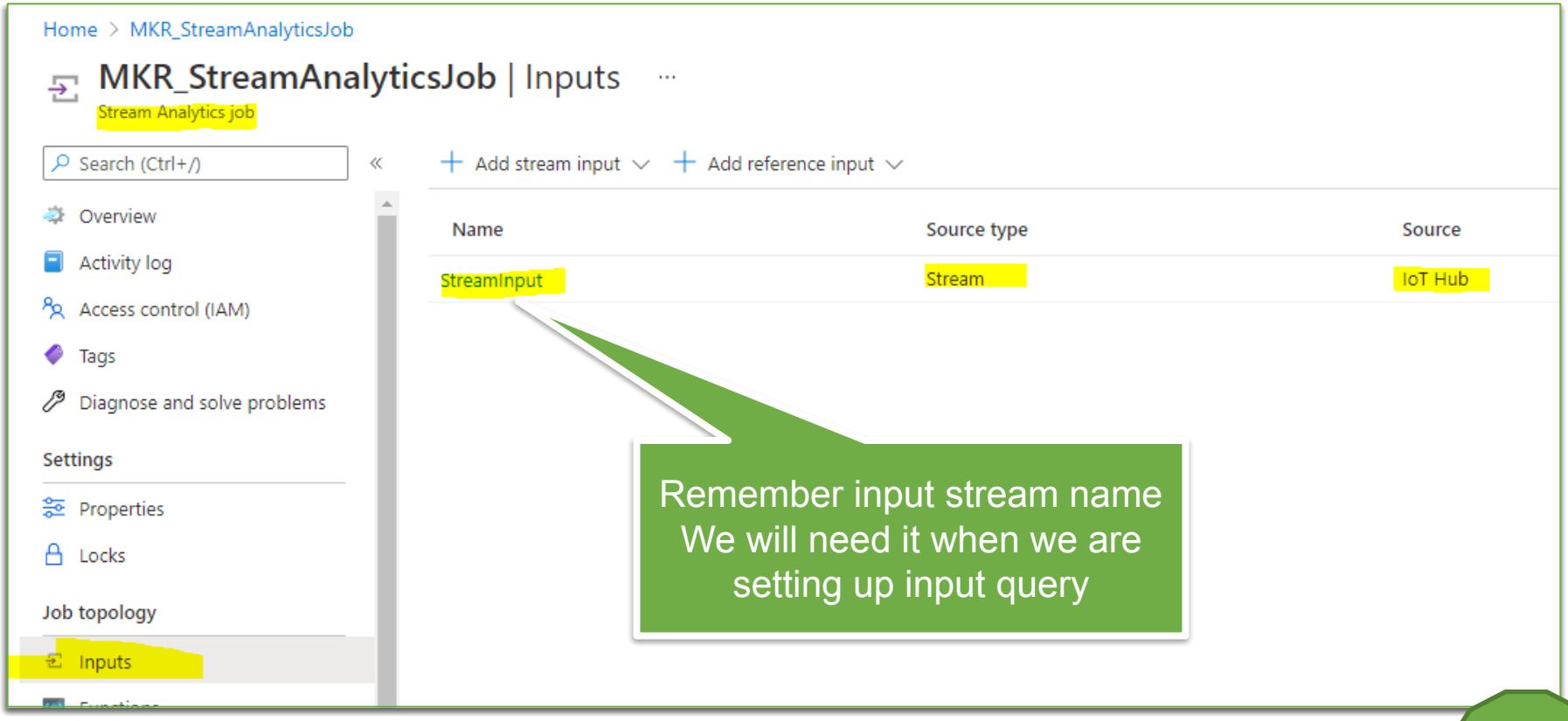
Remember input stream name
We will need it when we are setting up input query

Our code updated to send telemetry as JSON format



2

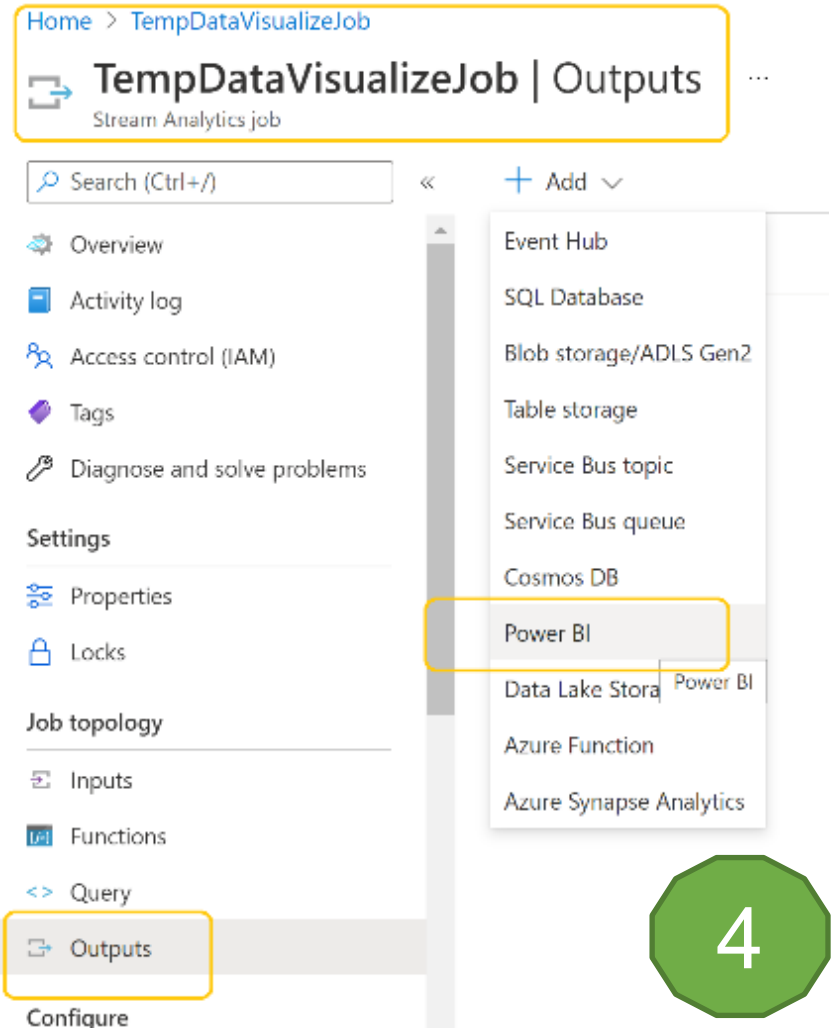
Done! Stream Input Added



3

Add a Stream Output to the Stream Analytics job

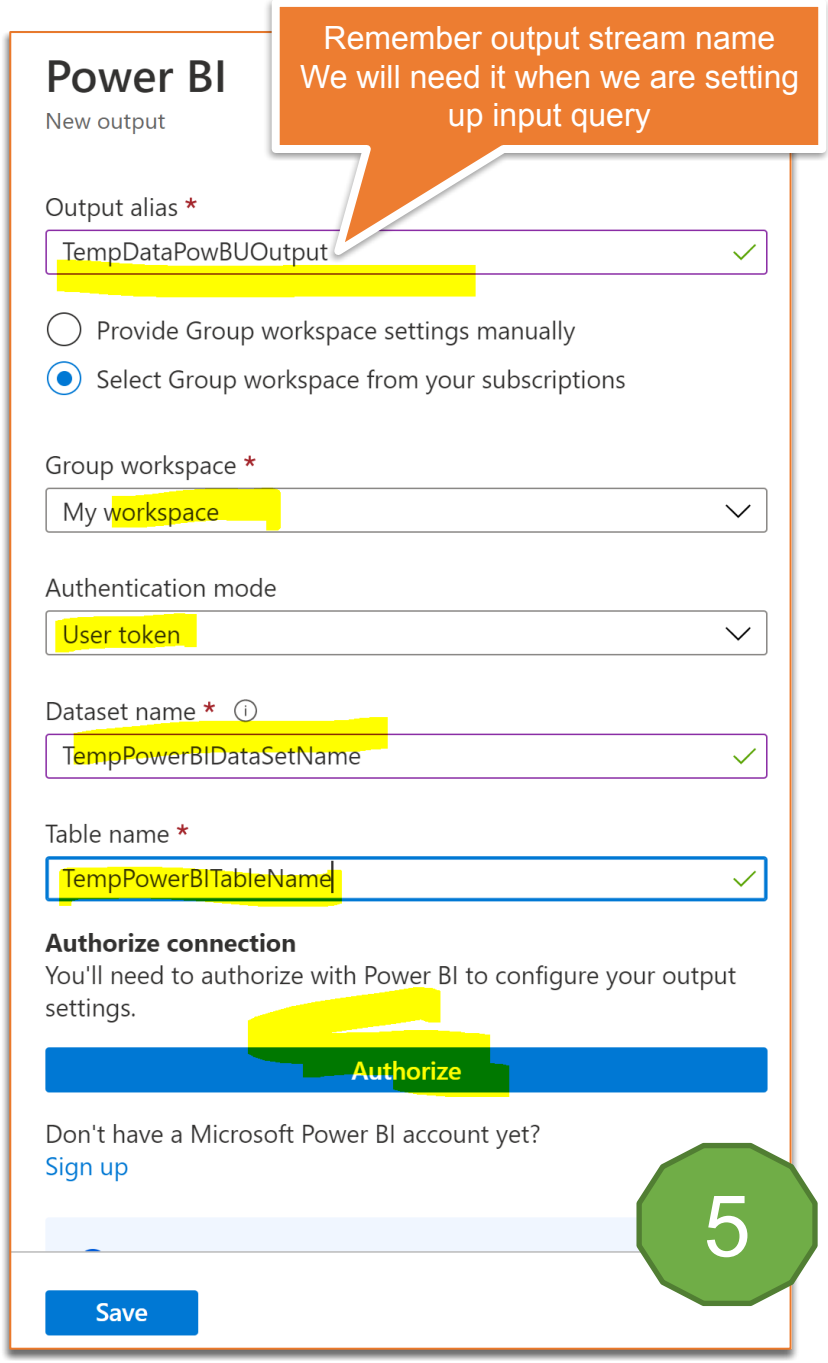
- Under **Job topology**, select **Outputs**.
- In the **Outputs** pane, select **Add**, and then select **Power BI** from the drop-down list.
- On the **Power BI – New output** pane, select **Authorize** and follow the prompts to sign in to your Power BI account.



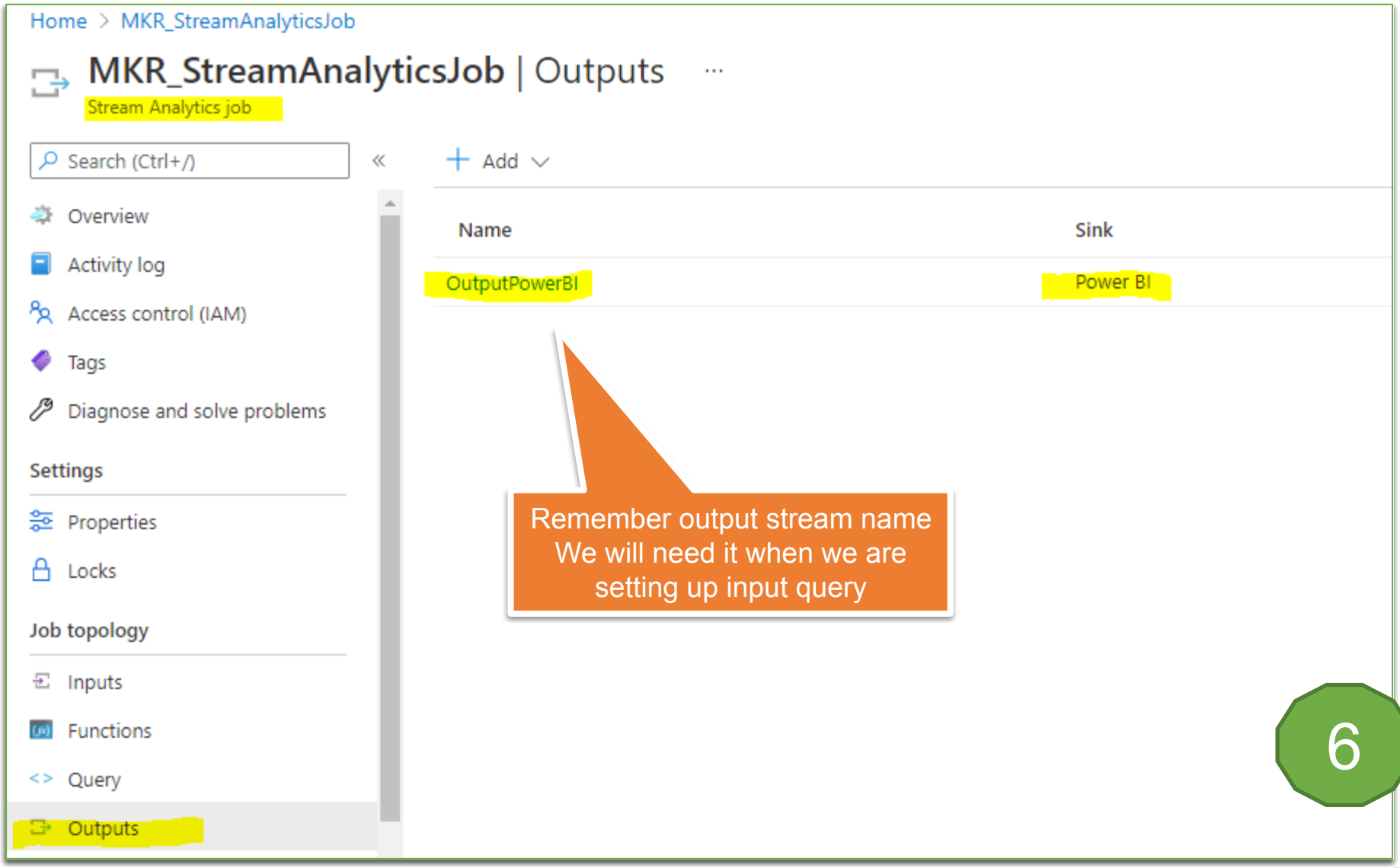
Add a Stream Output

After you've signed in to Power BI, enter the following information:

- **Output alias**:
  A unique alias for the output.
- **Group workspace**: Select your target group workspace.
- **Dataset name**:
  Enter a dataset name.
- **Table name**:
  Enter a table name.
- **Authentication mode**:
  Leave at the default.

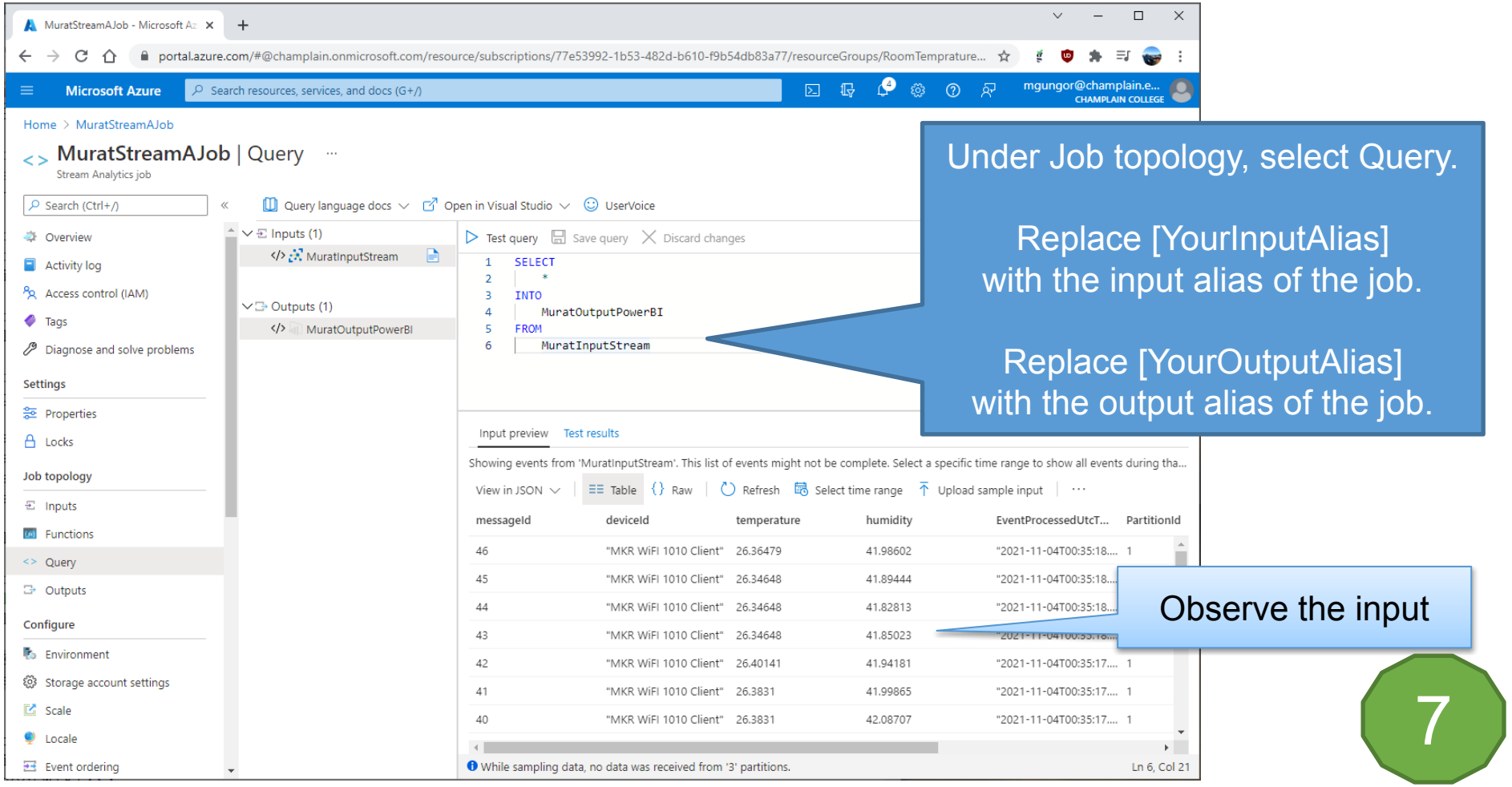


Done! Stream Output Added



Query – FROM and INTO update

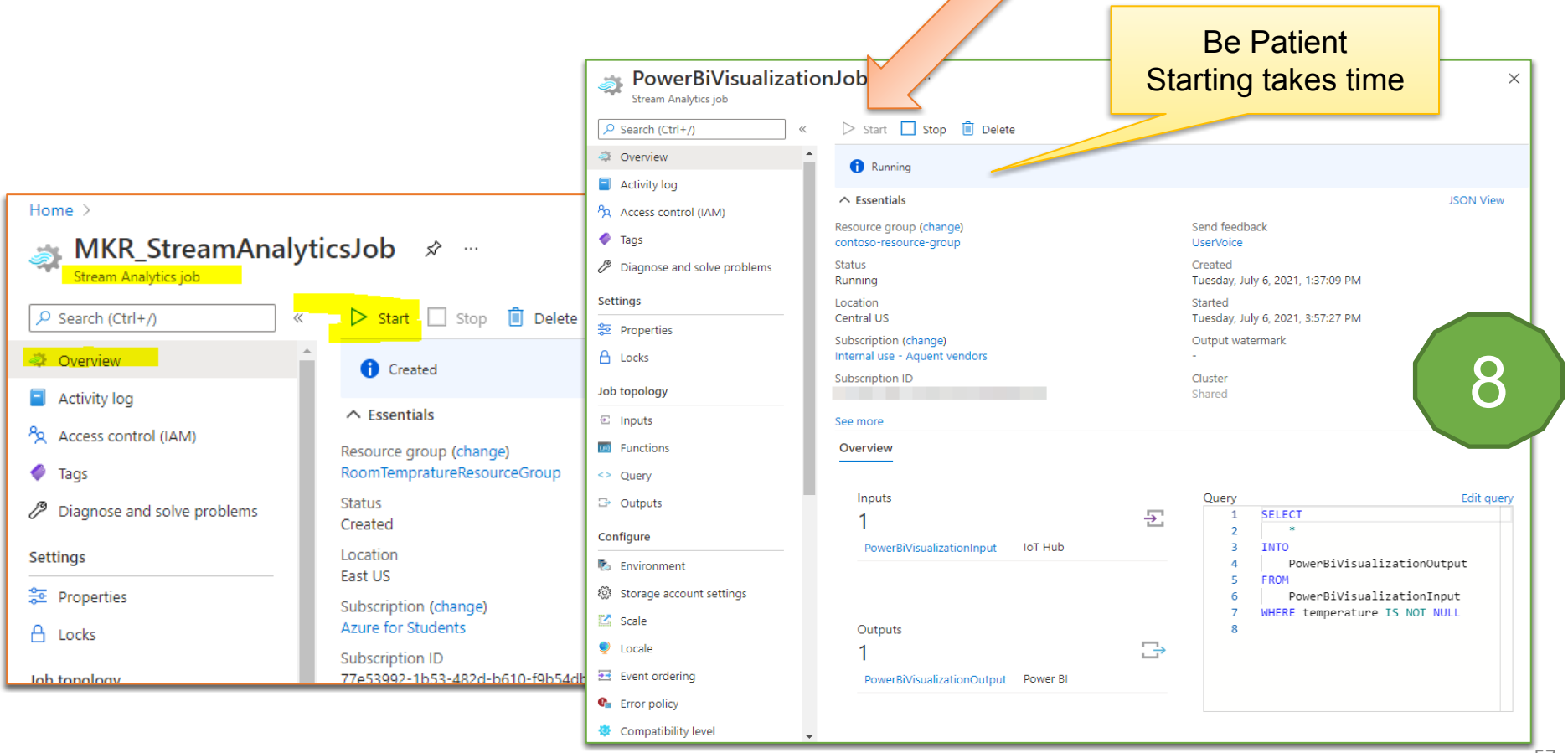
- Test Query



https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi#configure-the-query-of-the-stream-analytics-job

START the Stream Analytics job

- In the Stream Analytics job, select **Overview**, then select **Start** > **Now** > **Start**. Once the job successfully starts, the job status changes from **Stopped** to **Running**.



Power BI is a business analytics service by Microsoft.

Business Intelligence (BI)

PowerBI

Create Power BI report to visualize the data

What is Power BI?

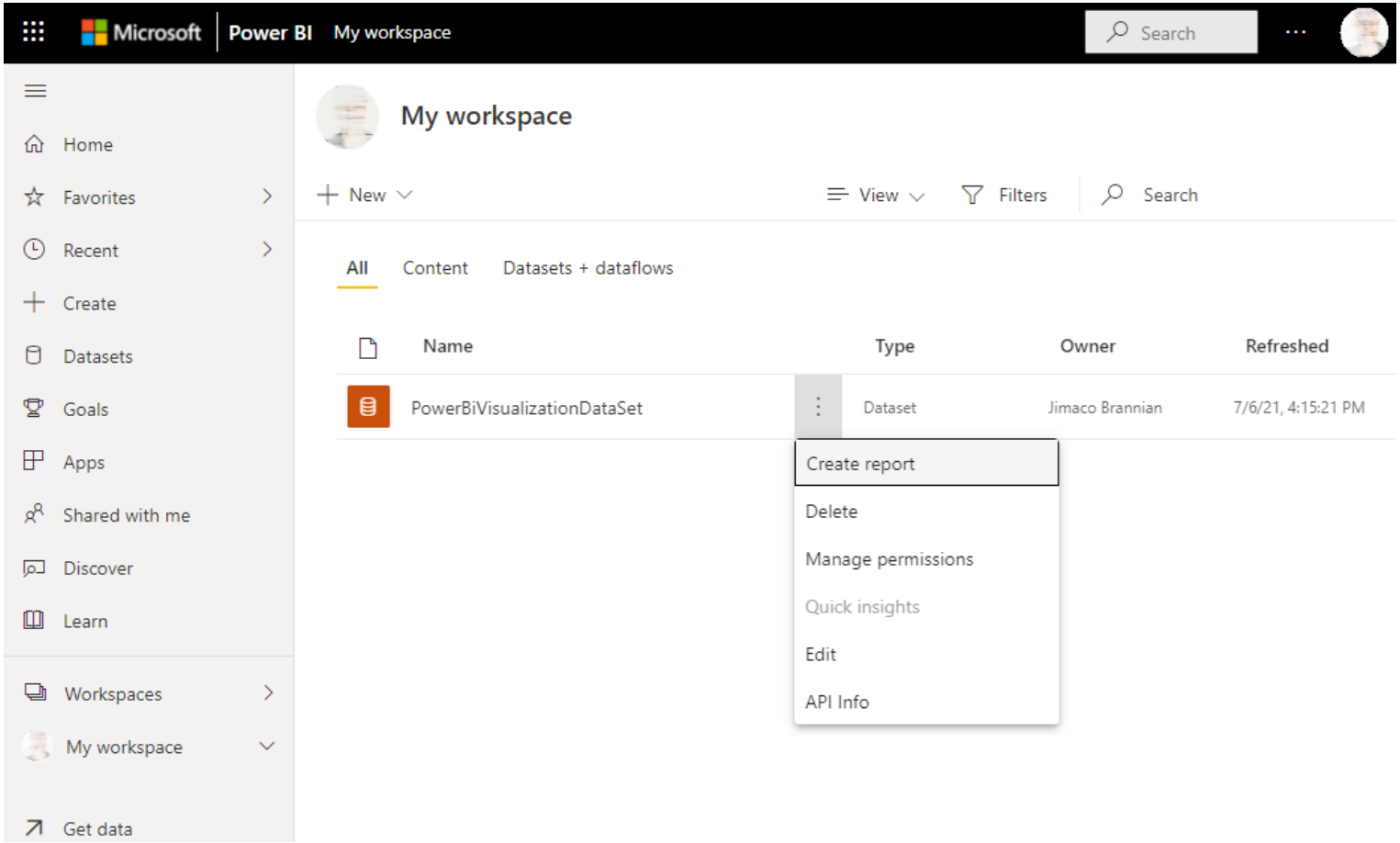
https://youtu.be/yKTSLffVGbk

Create and publish a Power BI report to visualize the data

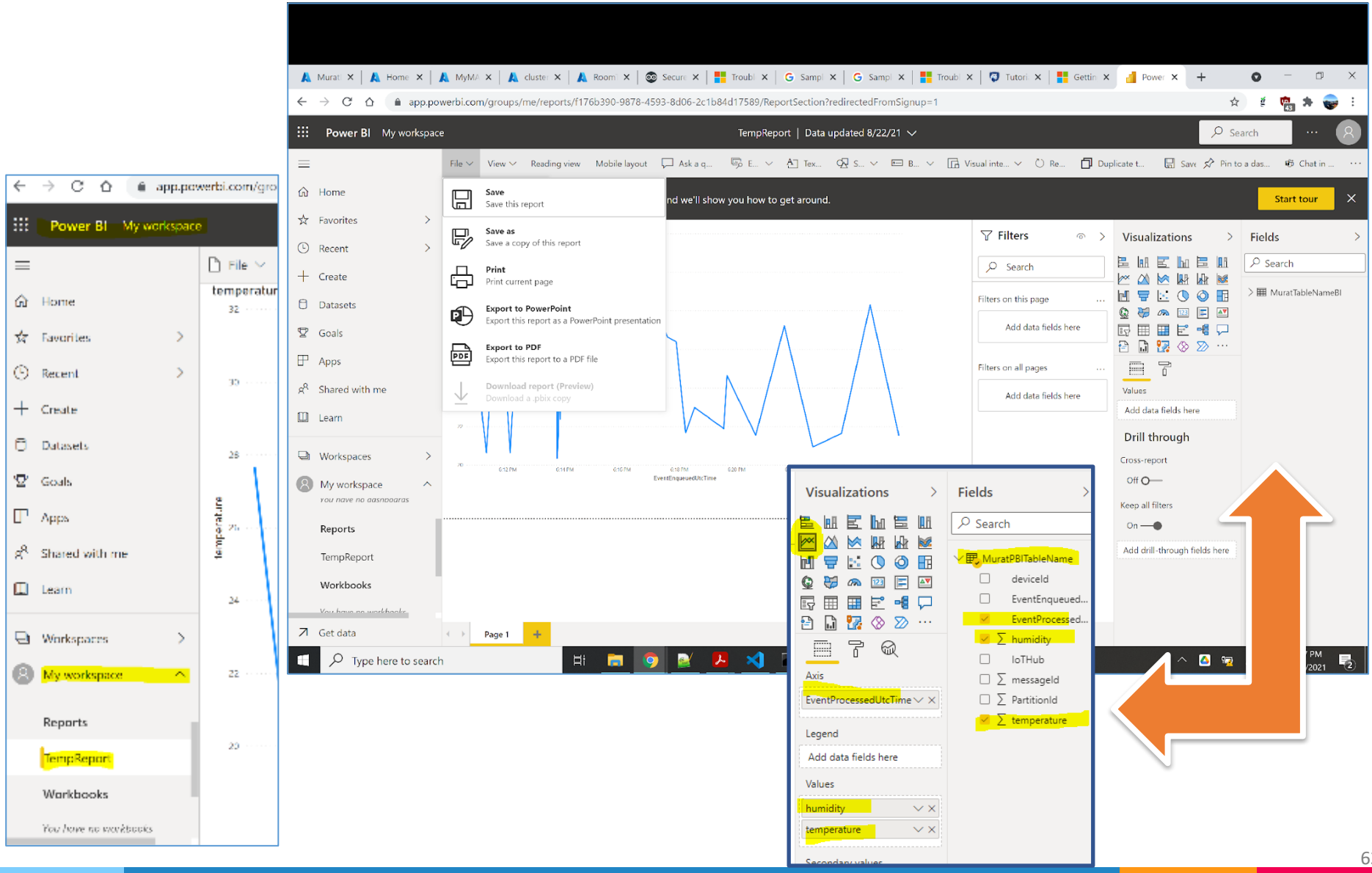
1. Make sure the client app is running on your device.
2. Sign in to your Power BI account and select **Power BI service** from the top menu.
   - https://powerbi.microsoft.com/
3. Select the workspace you used from the side menu, **My Workspace**.
4. Under the **All** tab or the **Datasets + dataflows** tab, you should see the dataset that you specified when you created the output for the Stream Analytics job.

https://powerbi.microsoft.com/

- Hover over the dataset you created, select **More options** menu (the three dots to the right of the dataset name), and then select **Create report**.



Power BI report to visualize the data



Cleanup resources

Cleaning Up Resources

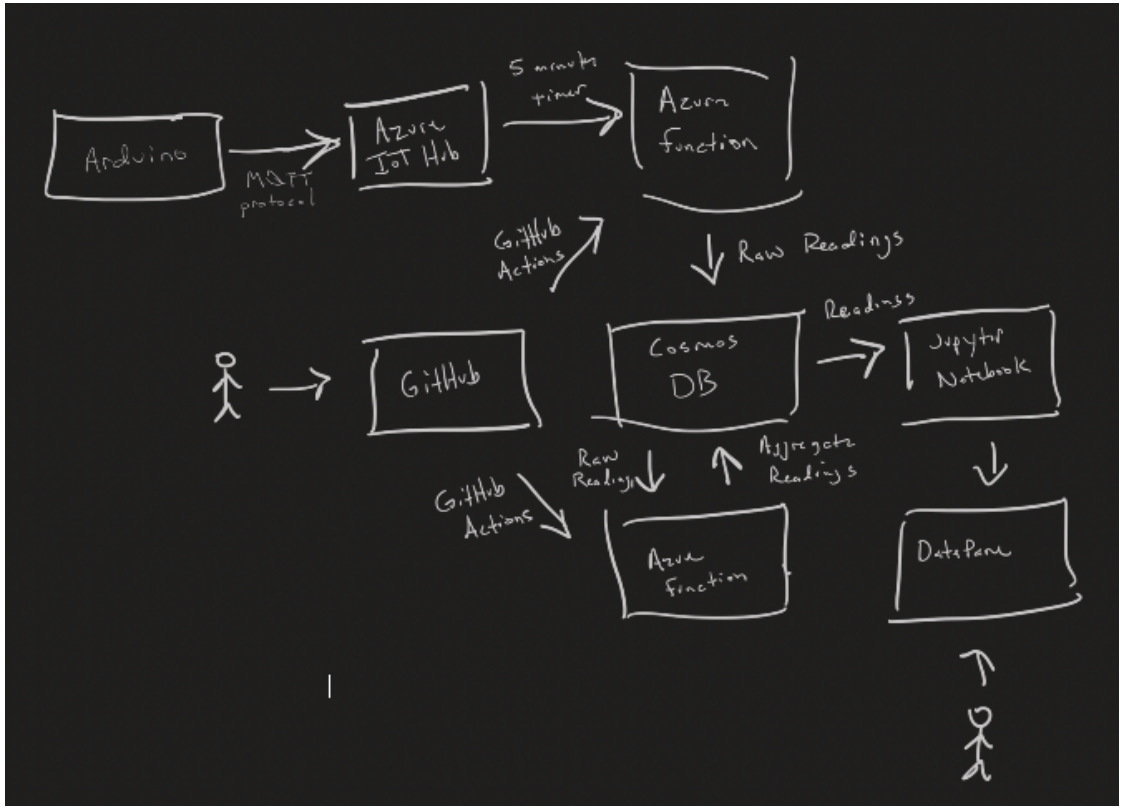
- You've created a resource group, an IoT hub, a Stream Analytics job, and a dataset in Power BI.
- If you don't need the IoT hub or the other resources you created any longer, you can delete the resource group in the portal.
- Clean up Power BI resources as well too

More Detail:
https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi#cleanup-resources

Want to Explore More

Good Tutorials:

- How can we save telemetry an Azure SQL database?
  - https://www.mssqltips.com/sqlservertip/6335/how-to-capture-iot-data-in-azure/
- IoT Azure Pipeline
  - https://paul-bruffett.medium.com/iot-azure-pipeline-9725ac2b6a00



More Application

- Visualize real-time sensor data from your Azure IoT hub in a web application
  - https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-web-apps
- Tutorial: Implement IoT spatial analytics by using Azure Maps
  - https://docs.microsoft.com/en-us/azure/azure-maps/tutorial-iot-hub-maps
- Arduino IoT Cloud Google Sheets Integration
  - https://create.arduino.cc/projecthub/Arduino_Genuino/arduino-iot-cloud-google-sheets-integration-71b6bc

Some Terminology

Communicate with IoT Hub

- Device to cloud (D2C) messages
  - – these are messages sent from a device to IoT Hub, such as telemetry. They can then be read off the IoT Hub by your application code.
- Cloud to device (C2D) messages
  - - these are messages sent from application code, via an IoT Hub to an IoT device
- Direct method requests
  - - these are messages sent from application code via an IoT Hub to an IoT device to request that the device does something, such as control an actuator. These messages require a response so your application code can tell if it was successfully processed.
- Device twins
  - - these are JSON documents kept synchronized between the device and IoT Hub, and are used to store settings or other properties either reported by the device, or should be set on the device (known as desired) by the IoT Hub.

Extra Information:
Device Provisioning Service

Azure IoT Hub
Device Provisioning Service (DPS)

- Link IoT Device Provisioning Service and IoT hub so that the IoT Hub Device Provisioning Service can register devices to that hub
  - How to link an IoT hub
- Set up allocation policy, which is a IoT Hub Device Provisioning Service setting that determines how devices are assigned to an IoT hub
  - How to set up an allocation policy
- Add the device's unique security artifacts to the Device Provisioning Service
  - How to enroll device
- Start the device to allow your client application to start the registration with your Device Provisioning service
  - How to start and verify a device

QuickStart: Set up the IoT Hub Device Provisioning Service with the Azure portal

- Link the IoT hub and your Device Provisioning Service
  - https://docs.microsoft.com/en-us/azure/iot-dps/quick-setup-auto-provision#link-the-iot-hub-and-your-device-provisioning-service
- Azure IoT Hub Device Provisioning Service (DPS) Documentation
  - https://docs.microsoft.com/en-us/azure/iot-dps/quick-create-simulated-device-x509

IoT Hub DPS Demo

- View a demo of an IoT device being automatically provisioned, in a secure and scalable way, with IoT Hub Device Provisioning Service. You'll understand how you can provision thousands of IoT devices with zero-touch.
- Click to watch the video
  - https://azure.microsoft.com/en-us/resources/videos/iot-hub-dps-demo/
  - **Quickstart: Set up the IoT Hub Device Provisioning Service with Azure CLI**
  - https://docs.microsoft.com/en-us/azure/iot-dps/quick-setup-auto-provision-cli

References

- IoT Azure Pipeline
  - https://paul-bruffett.medium.com/iot-azure-pipeline-9725ac2b6a00
- Securely Connecting an Arduino NB 1500 to Azure IoT Hub © CC BY
  - https://create.arduino.cc/projecthub/Arduino_Genuino/securely-connecting-an-arduino-nb-1500-to-azure-iot-hub-af6470
- Securely Connecting an Arduino MKR WiFi 1010 to AWS IoT Core
  - https://create.arduino.cc/projecthub/Arduino_Genuino/securely-connecting-an-arduino-mkr-wifi-1010-to-aws-iot-core-a9f365
- Quickstart: Create a Stream Analytics job by using the Azure portal
  - https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-quick-create-portal
- Tutorial: Visualize real-time sensor data from Azure IoT Hub using Power BI
  - https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-power-bi
- Some more in the notes section of these slides.

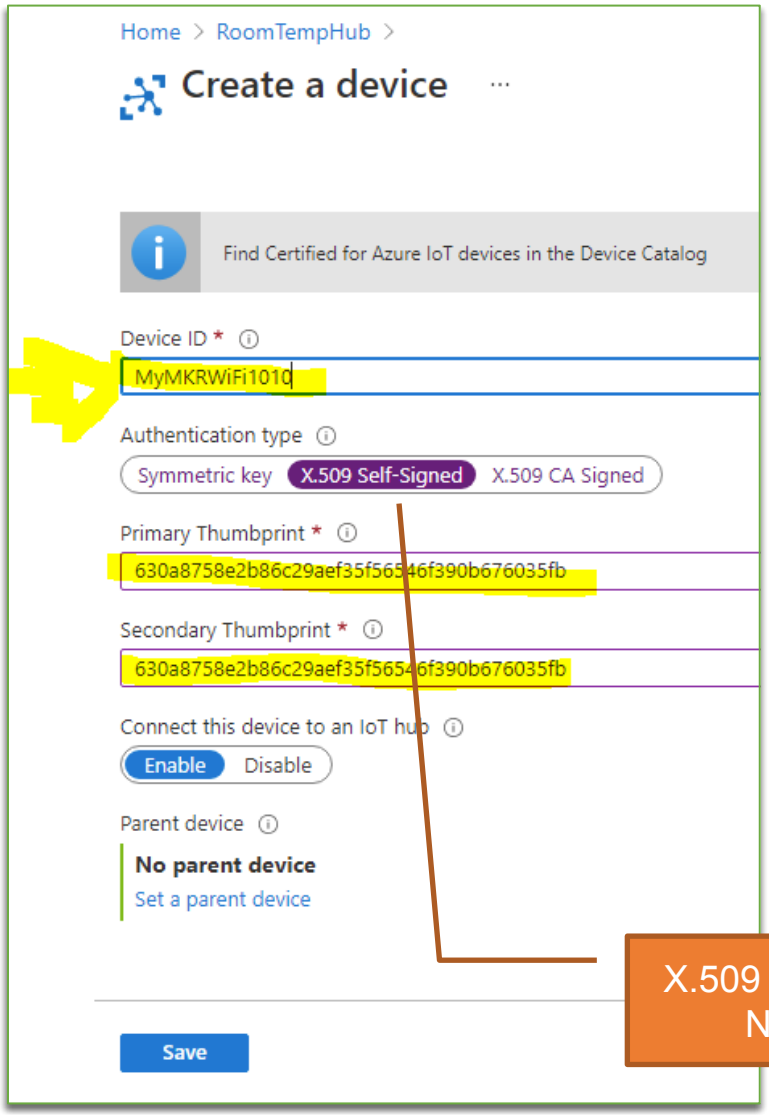
Thanks!

Any questions?

Let us dare!

These could be useful Most likely you will not need for this class

Adding a device PAUSE!
First we need device certification



- IoT device connects to your IoT Hub using Certification

Pause Here!

- To complete this we need to jump to next steps to create our certification

Creating Certification for your Arduino

X.509 certificates for authentication

- Azure IoT Hub allows devices that connect using the MQTT protocol and use X.509 certificates for authentication.

- We'll use a sketch to generate a self signed **X.509 Certificate** on the board and then add the SHA1 of this certificate to Azure IoT Hub portal.

- Every Arduino MKR board with on-board connectivity is equipped with a Microchip crypto element.

  This crypto element can be used to securely generate and store a 256-bit ECC (Elliptic Curve Cryptography) key.

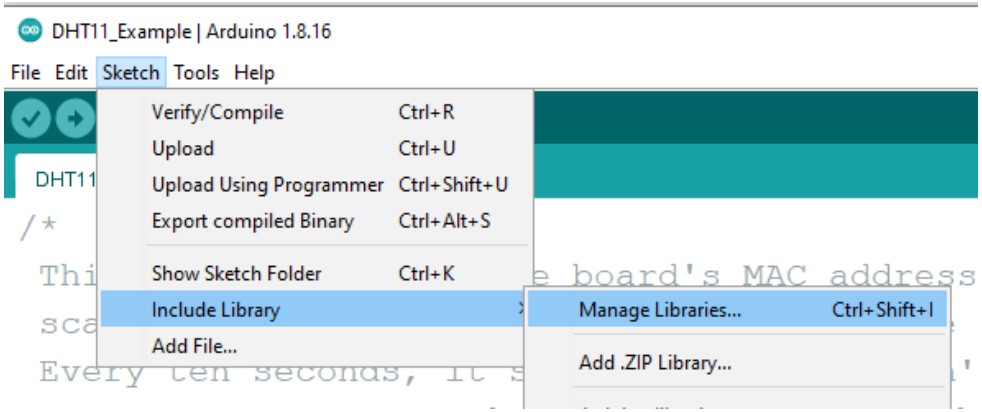
Libraries to Install

We will install the Arduino libraries that will be used, using the Arduino IDE's library manager.

Open the *Sketch -> Include Library -> Manage Libraries*... menu, search for and individually install each of the following libraries:

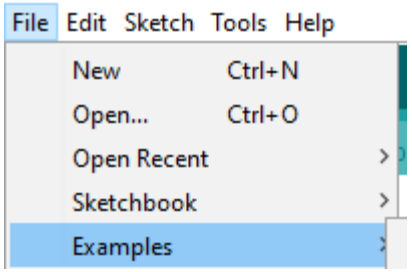
Libraries

1. ArduinoBearSSL
2. ArduinoECCX08
3. ArduinoMqttClient
4. Arduino Cloud Provider Examples

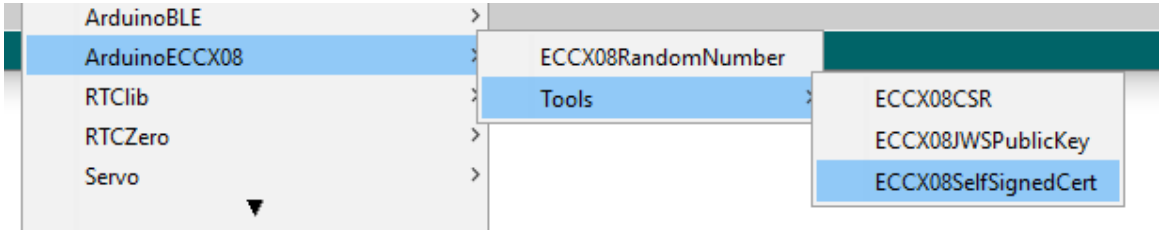


X.509 certificates for authentication

The self signed certificate can be generated using an example sketch from the ArduinoECCX08 library.



- "Open" the sketch in the Arduino IDE using the
  File \
  Examples \
  ArduinoECCX08 \
  Tools \
  ECCX08SelfSignedCert



- Click the "Upload" button to build and upload the sketch to your board, then open the Serial Monitor.

- After this, you will be prompted for information to include in the self signed certificate, including the issue year, month, day and hour of the certificate and the expiry period in years.
- For this tutorial we'll be using
- **slot 0** to generate and store the private key used to sign the self signed certificate (slots 1 to 4 can be used to generate and store additional private keys if needed) - then
- **slot 8** will be used to store the issue and expiry date of the certificate along with its signature.

Serial Port – Enter the data requested

ECCX08 Serial Number = 012385338ACB4748EE

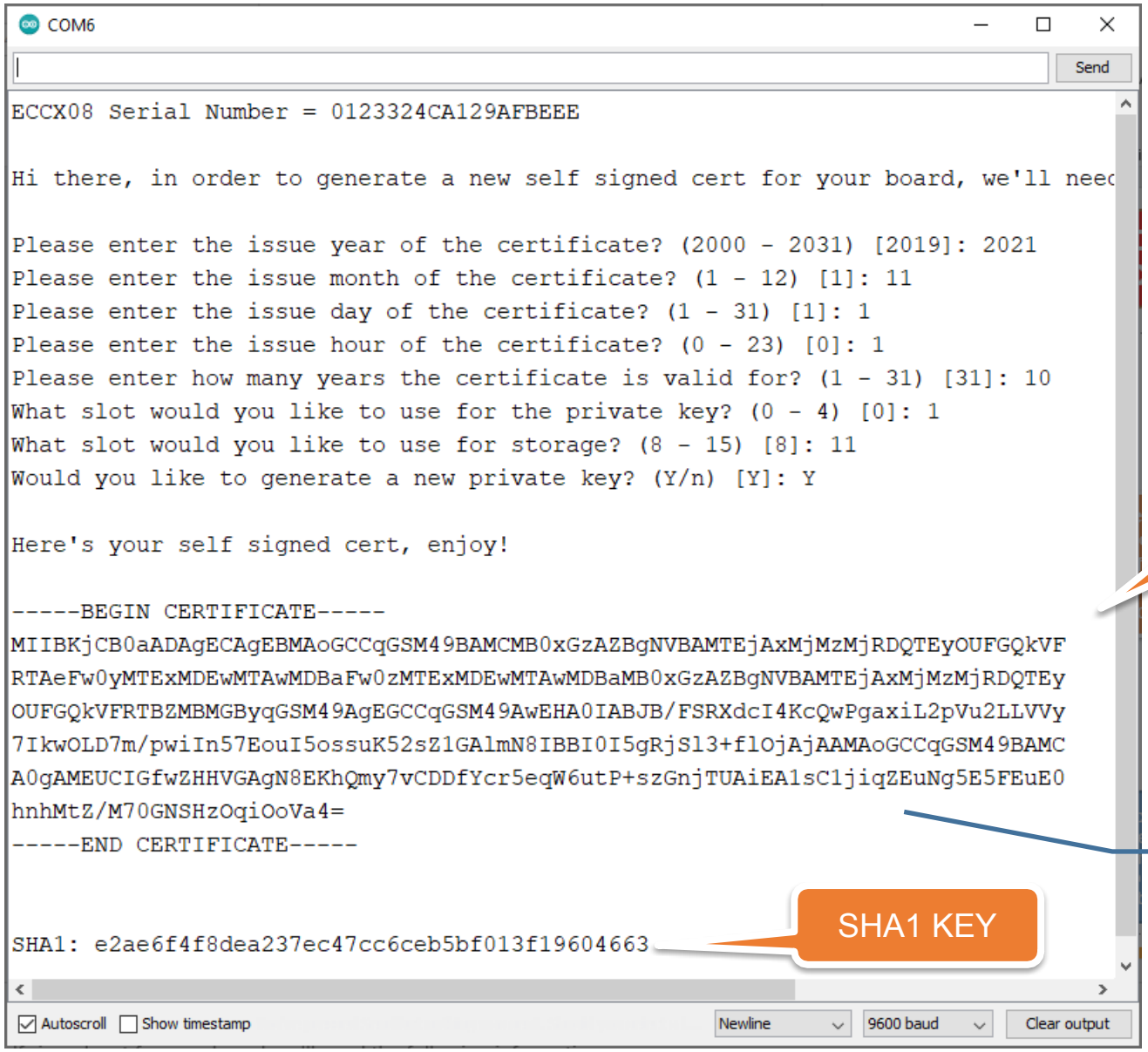
Hi there, in order to generate a new self signed cert for your board, we'll need the following information ...

Please enter the issue year of the certificate? (2000 - 2031) [2019]: **2021**
Please enter the issue month of the certificate? (1 - 12) [1]: **8**
Please enter the issue day of the certificate? (1 - 31) [1]: **18**
Please enter the issue hour of the certificate? (0 - 23) [0]: **11**
Please enter how many years the certificate is valid for? (1 - 31) [31]: **11**
What slot would you like to use for the private key? (0 - 4) [0]: **0**
What slot would you like to use for storage? (8 - 15) [8]: **8**
Would you like to generate a new private key? (Y/n) [Y]: **Y**

Here's your self signed cert, enjoy!

Serial Monitor – Executing

DON'T CLOSE



Copy everything
Save into a text file
for further
processing
Use Notepad++

SHA1 KEY

Note: Since the private key is generated inside the crypto element it never leaves the device and is stored securely and cannot be read.

Sample Run Output

```
-----BEGIN CERTIFICATE-----
MIIBKjCB0aADAgECAgEBMAoGCCqGSM49BAMCMB0xGzAZBgNVBAMTEjAxMjM4NTMzOEFDQjQ3NDhF
RTAeFw0yMTA4MTgxMTAwMDBaFw0zMjA4MTgxMTAwMDBaMB0xGzAZBgNVBAMTEjAxMjM4NTMzOEFD
QjQ3NDhFRTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABDJ7ddEvDvp7eVK6gp2dCyGNMN0N0re0
WpBofjzOMcOYbWGTiOZeQtoUghi4lle3avK1Im9WV67/0nlysTgeSFWjAjAAMAoGCCqGSM49BAMC
A0gAMEUCIQCObICMKzAMcASi0maa/P9hW8UXvsurk2whrljxQEbdVgIgeFrzdnFjWhJR7BFyKBek
TlqW0q7BYfTSQkD3/YIg7cg=DONT
-----END CERTIFICATE-----
```

From Your Serial Port:

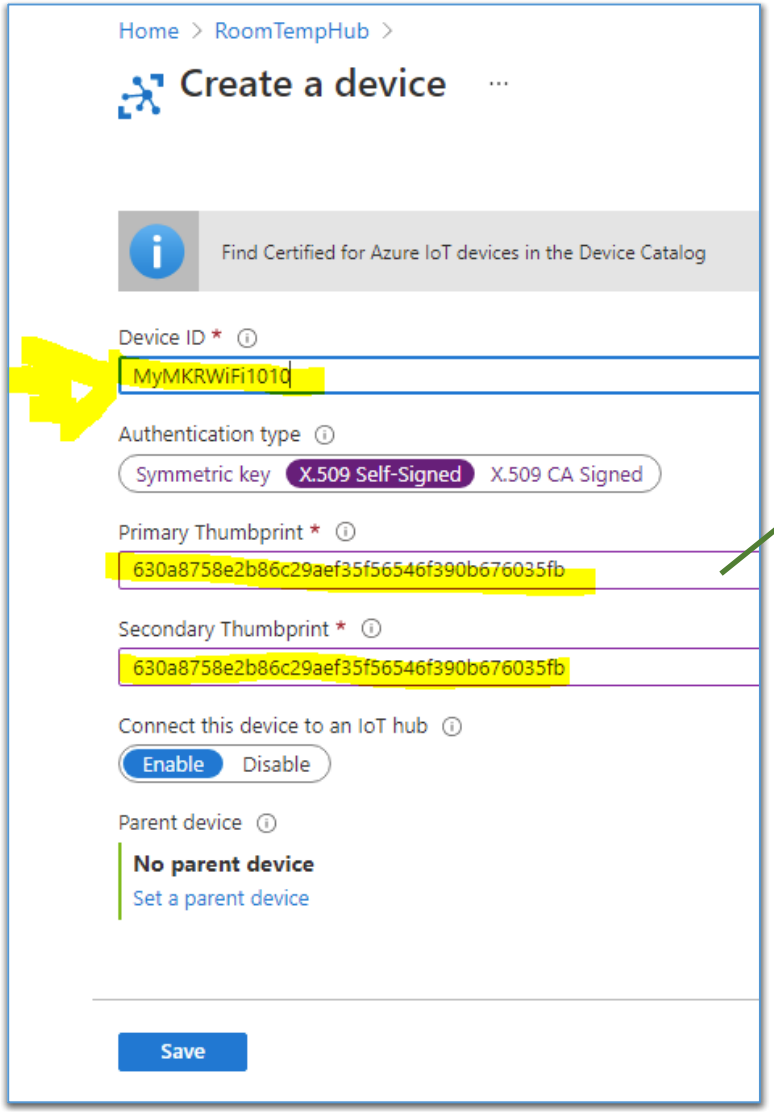
Save the above text in orange color to **ECCX08cert.pem**
(You can name the file anything but keep the extension)

SHA1: **630a8758e2b86c29aef35f56546f390b676035DONTfb**

We need this blue key while adding our device to Azure Cloud

Go back to Azure to Add our Device

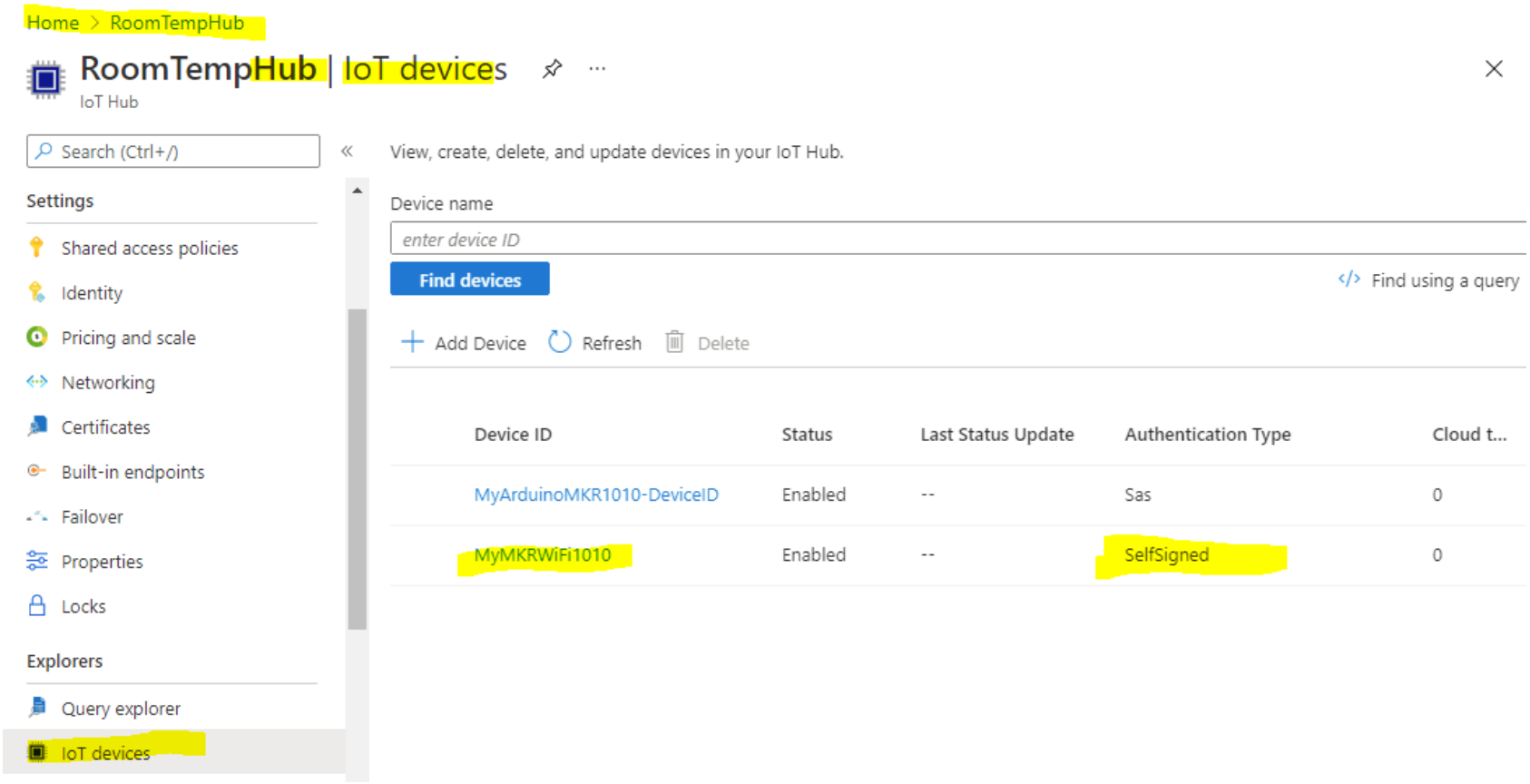
Create a device



- When your IoT device connects to your IoT Hub using Certification

SHA1 KEY

Device Added to Azure Cloud



Code Part at Arduino

Open This Example Code

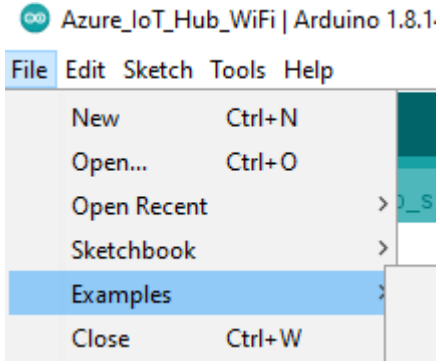
Open

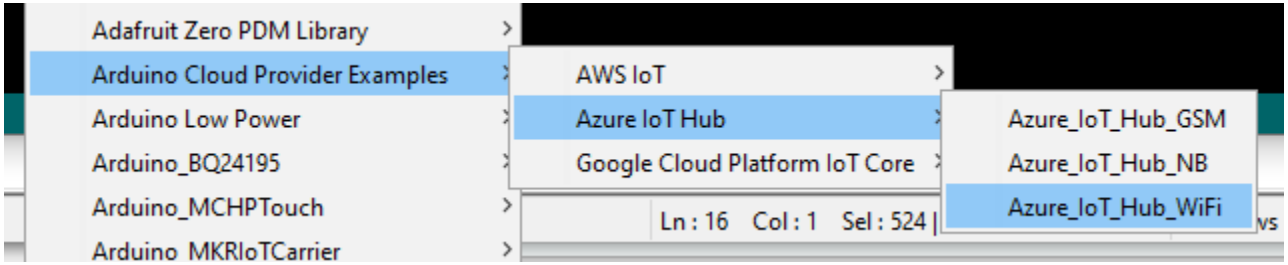
Azure Examples /

Arduino Cloud Provider Examples /

Azure IoT Hub /

Azure IoT Hub WiFi



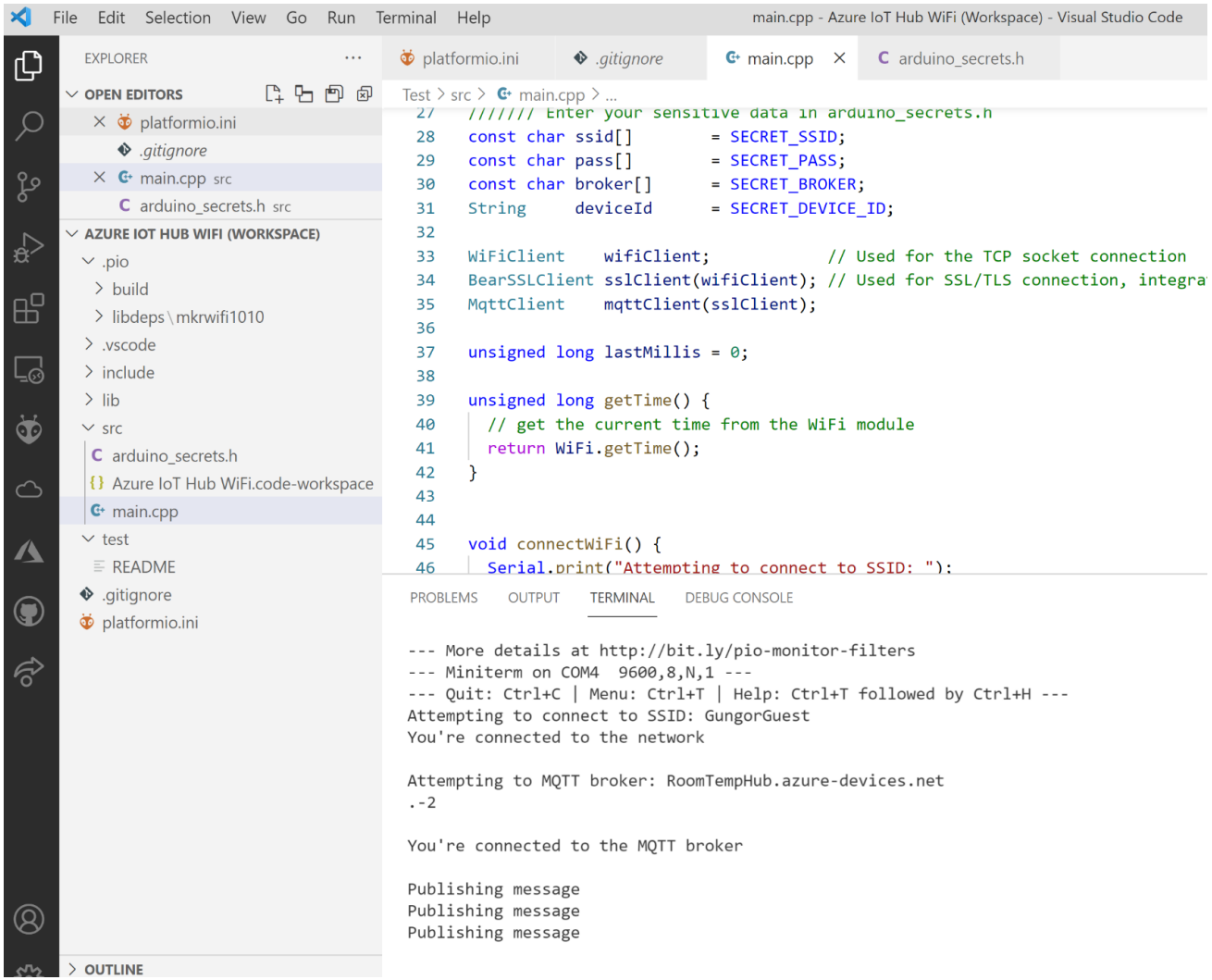


Visual Studio Code

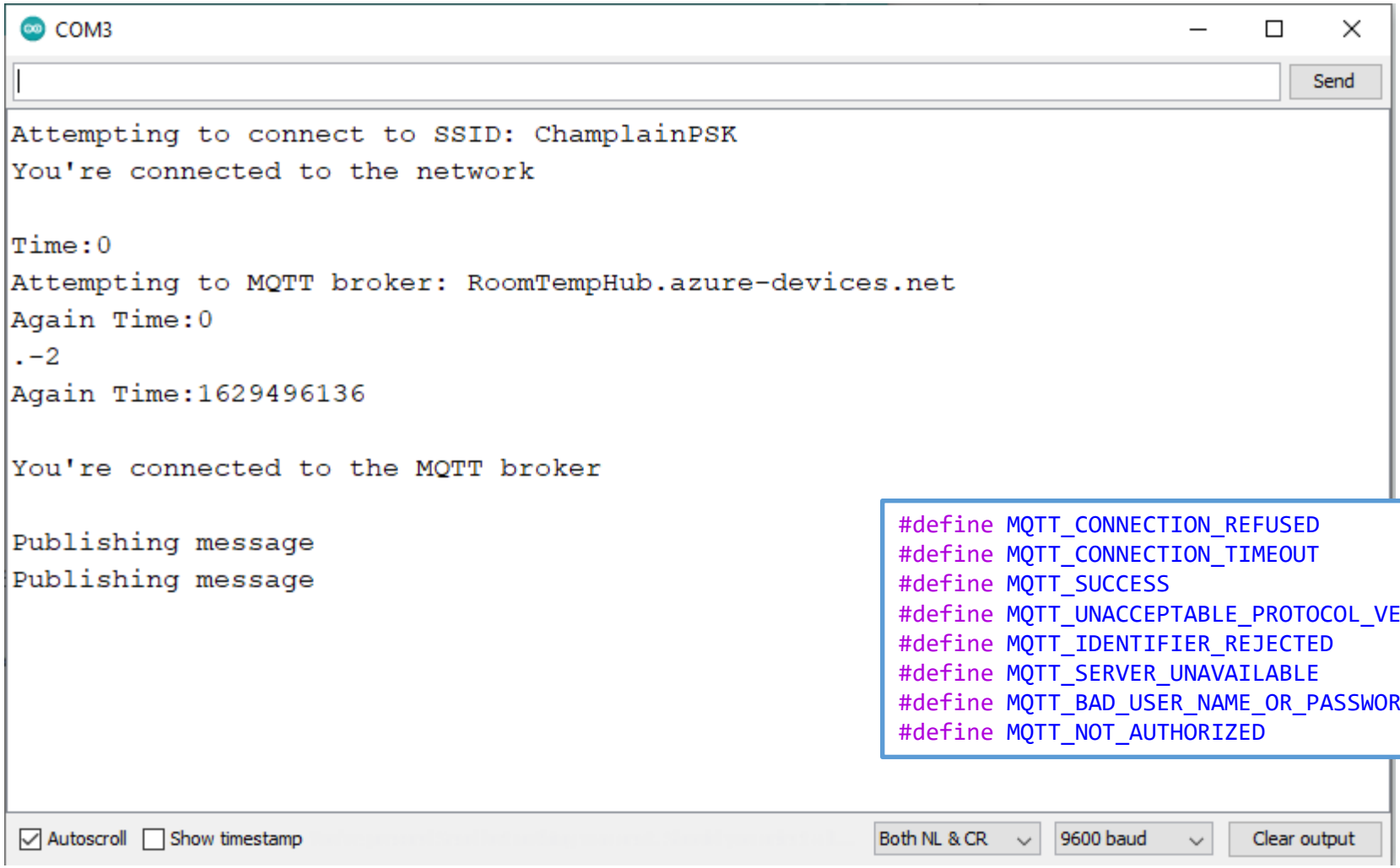

platformio.ini


main.cpp


arduino_secrets.h



My Sample Run at Serial Monitor




```
#define MQTT_CONNECTION_REFUSED            -2
#define MQTT_CONNECTION_TIMEOUT            -1
#define MQTT_SUCCESS                        0
#define MQTT_UNACCEPTABLE_PROTOCOL_VERSION  1
#define MQTT_IDENTIFIER_REJECTED            2
#define MQTT_SERVER_UNAVAILABLE             3
#define MQTT_BAD_USER_NAME_OR_PASSWORD      4
#define MQTT_NOT_AUTHORIZED                 5
```

Using JSON format

Publishing message

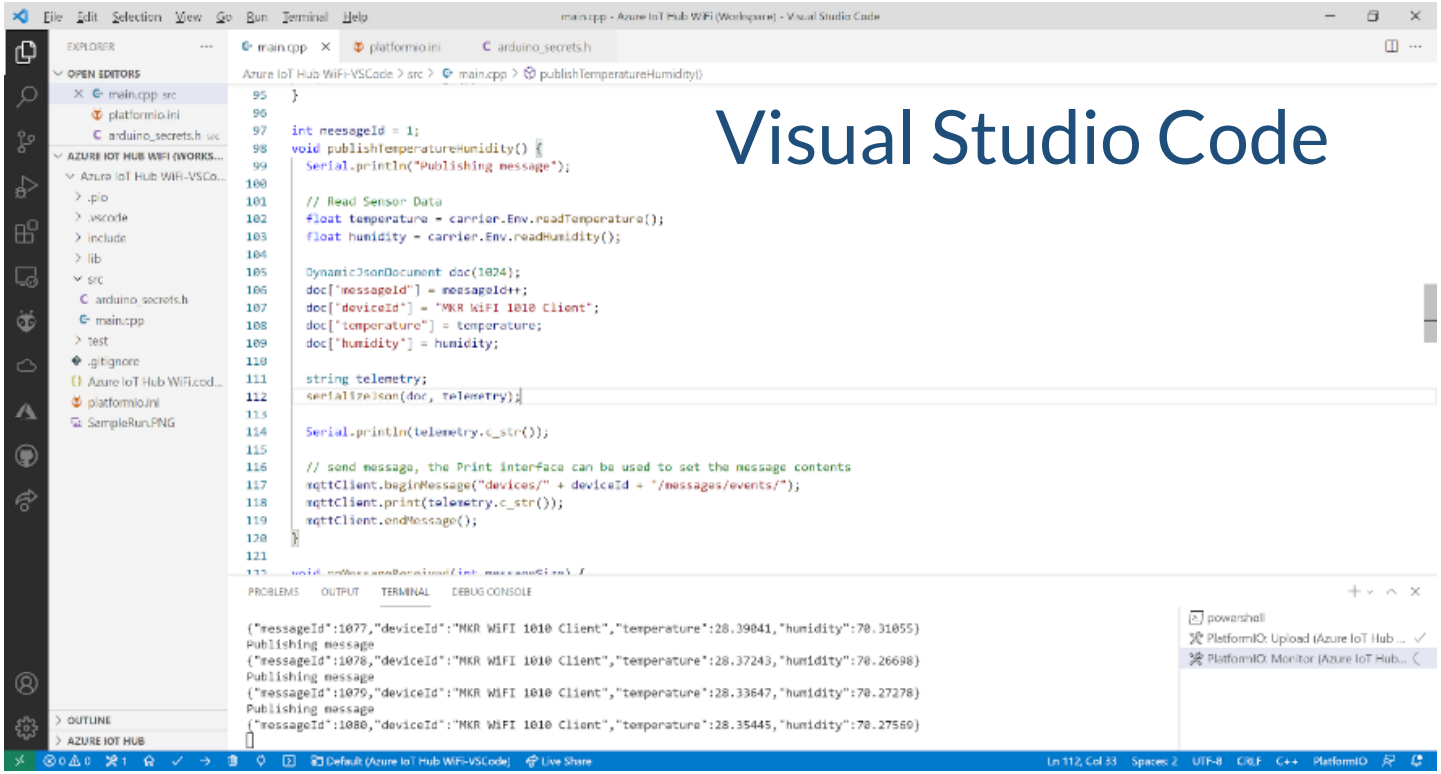
```
{
        "messageId":968,
        "deviceId":"MKR WiFI 1010 Client",
        "temperature":28.24657,
        "humidity":70.85092
}
```

platformio.ini

arduino_secrets.h

main.cpp

Visual Studio Code



Azure services

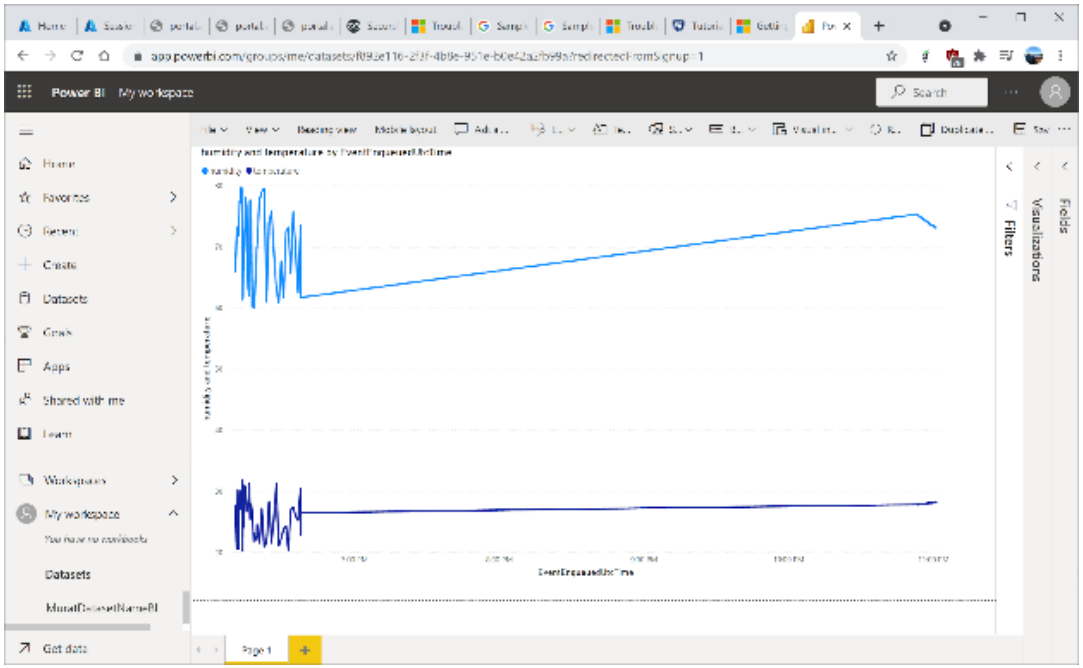
Create a resource

Power BI Embedded

All resources

Recent resources

Name
MuratIoTHub
MuratStreamAnalysJob
MuratResourceGroup



Azure Settings