University of Sfax
Higher Institute of Computer Science and Multimedia
Module : Internet Of Things
Students : FEKI Emna,BZIOUICH Wiem & KHARRAT Sahar

Computer Science Department
Academic Year: 2025/2026
Classroom : TIINFO-GL TD1-TP1

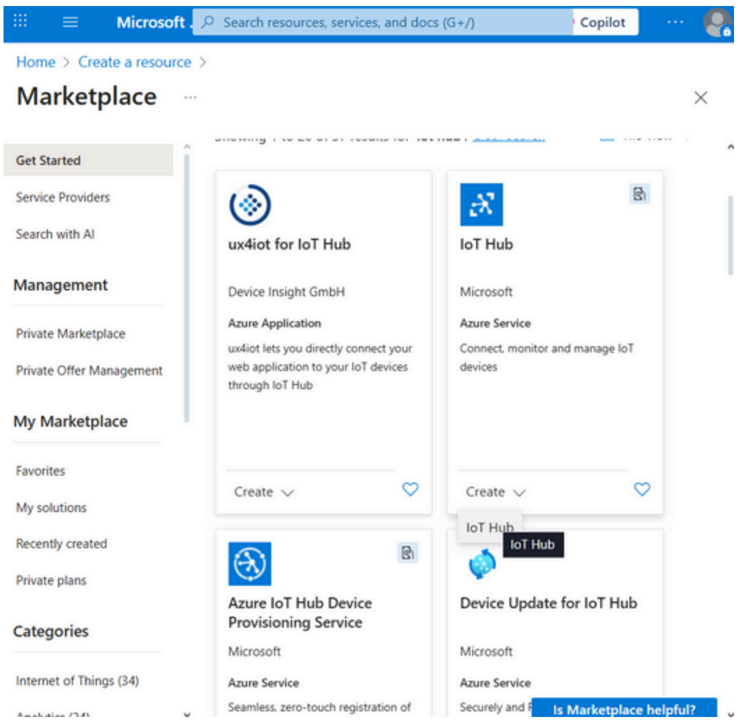
IoT Workflow with Azure IoT Hub – Cloud Platform Demonstration

Objectives :

- Showcase a complete IoT data flow
- Demonstrate real-time data visualization
- Implement downlink control

Part 1 : Azure IoT Hub Setup

Step 1.1 :Creation of an IOT Hub

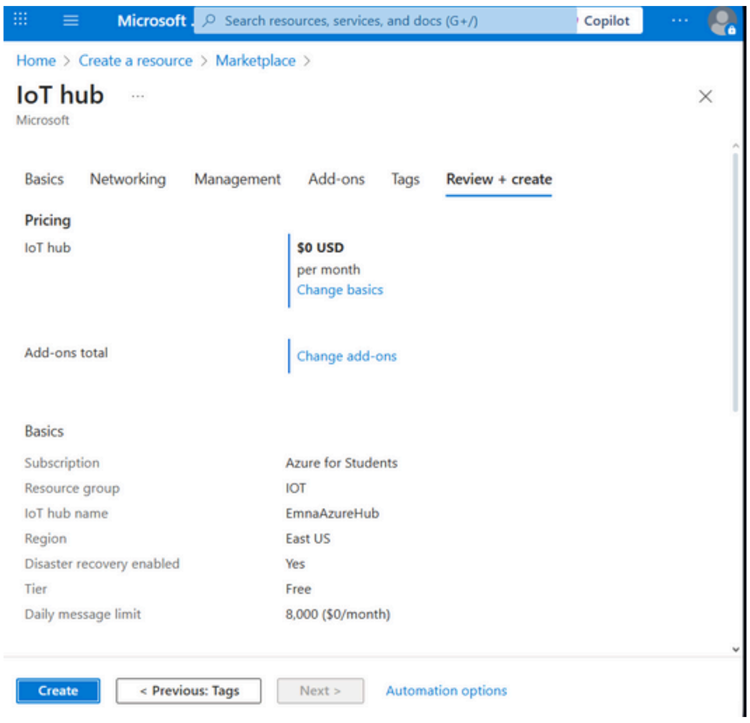


ux4iot for IoT Hub
Device Insight GmbH
Azure Application
ux4iot lets you directly connect your web application to your IoT devices through IoT Hub

IoT Hub
Microsoft
Azure Service
Connect, monitor and manage IoT devices

Azure IoT Hub Device Provisioning Service
Microsoft
Azure Service
Seamless, zero-touch registration of

Device Update for IoT Hub
Microsoft
Azure Service
Securely and



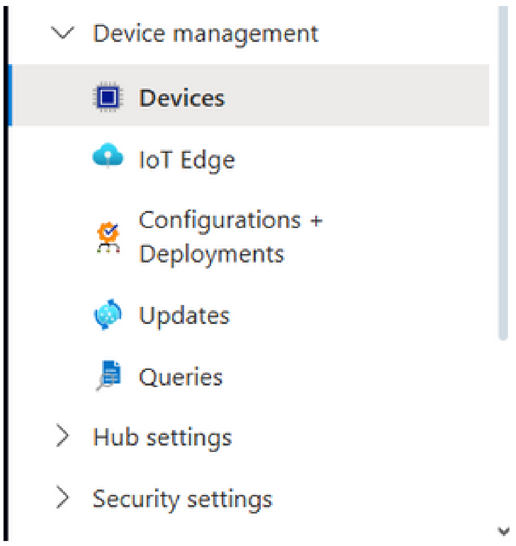
Pricing

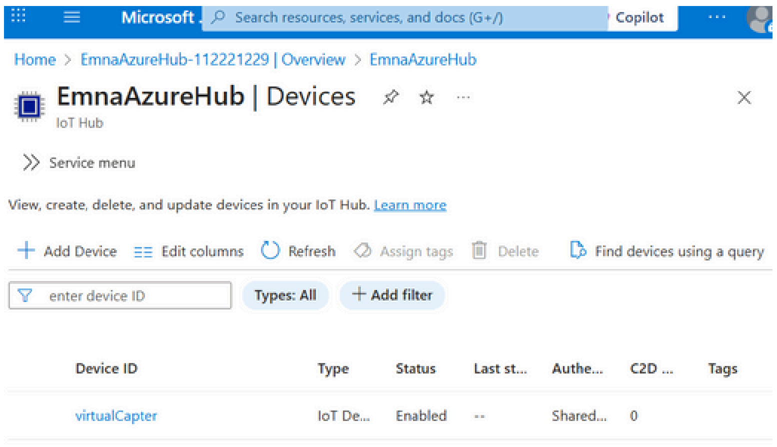
IoT hub	$0 USD per month
Add-ons total	

Basics

Subscription	Azure for Students
Resource group	IOT
IoT hub name	EmnaAzureHub
Region	East US
Disaster recovery enabled	Yes
Tier	Free
Daily message limit	8,000 ($0/month)

Step 1.2 :Creation of an IOT device





Home > EmnaAzureHub-112221229 | Overview > EmnaAzureHub

EmnaAzureHub | Devices

IoT Hub

Service menu

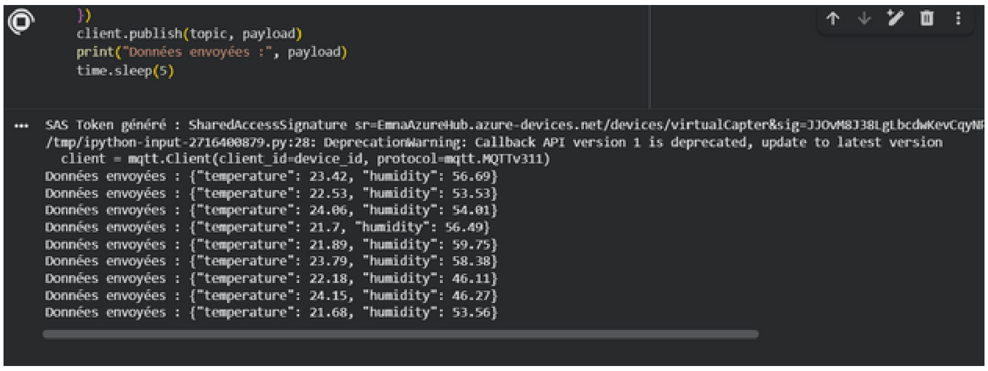
View, create, delete, and update devices in your IoT Hub. Learn more

Device ID	Type	Status	Last st...	Authe...	C2D ...	Tags
virtualCapter	IoT De...	Enabled	--	Shared...	0	

Part 2: Sensor Simulation (Python)

Step2.1 : Create a script Python to send data to the device

```
!pip install azure-iot-device

# Import libraries
from azure.iot.device import IoTHubDeviceClient, Message
import random
import time

# === Azure IoT Hub Device Connection String ===
CONNECTION_STRING = "HostName=EmnaAzureHub.azure-devices.net;DeviceId=virtualCapter;
 SharedAccessKey=6IaXnVPqEi4VHIqVFczDb65UTv63isVw21p1DDOJn/o="

# Create IoT Hub client
client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)

# Connect to IoT Hub
print("Connecting to Azure IoT Hub...")
client.connect()
print("Connected successfully!")

# Send simulated telemetry
print("Sending telemetry data...")
for i in range(10):  # Send 10 messages
    temperature = round(random.uniform(20, 30), 2)
    humidity = round(random.uniform(40, 60), 2)
    payload = f'{{"temperature": {temperature}, "humidity": {humidity}}}'
    message = Message(payload)
    client.send_message(message)
    print(f"Message {i+1} sent: {payload}")
    time.sleep(5)  # Wait 5 seconds between messages
```

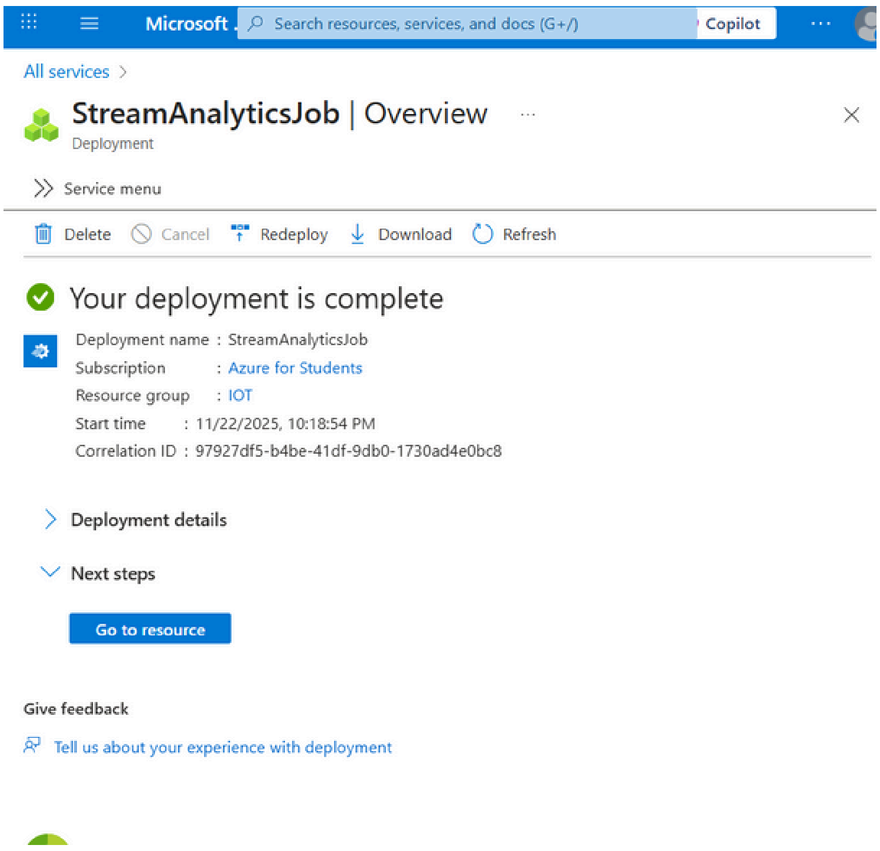


```
    })
    client.publish(topic, payload)
    print("Données envoyées :", payload)
    time.sleep(5)
```

```
SAS Token généré : SharedAccessSignature sr=EmnaAzureHub.azure-devices.net/devices/virtualCapter&sig=JJOvM8J38LgLbcdwKevCqyN
/tmp/ipython-input-2716400879.py:28: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client(client_id=device_id, protocol=mqtt.MQTTv311)
Données envoyées : {"temperature": 23.42, "humidity": 56.69}
Données envoyées : {"temperature": 22.53, "humidity": 53.53}
Données envoyées : {"temperature": 24.06, "humidity": 54.01}
Données envoyées : {"temperature": 21.7, "humidity": 56.49}
Données envoyées : {"temperature": 21.89, "humidity": 59.75}
Données envoyées : {"temperature": 23.79, "humidity": 58.38}
Données envoyées : {"temperature": 22.18, "humidity": 46.11}
Données envoyées : {"temperature": 24.15, "humidity": 46.27}
Données envoyées : {"temperature": 21.68, "humidity": 53.56}
```

Part 3: Stream Analytics Job

Step3.1 : Create a steam job



StreamAnalyticsJob | Overview

Deployment

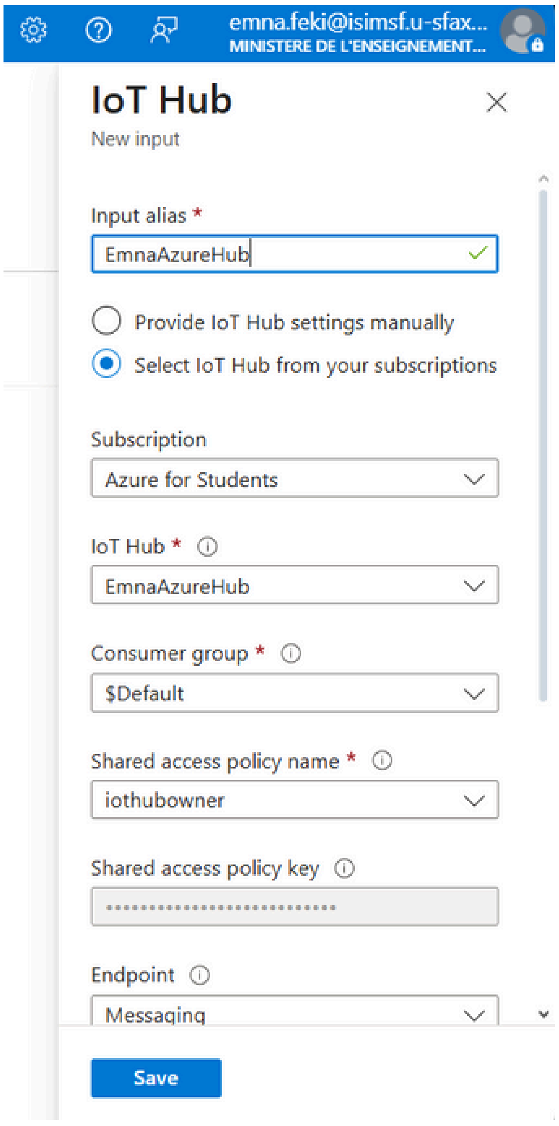
Your deployment is complete

Deployment name : StreamAnalyticsJob
Subscription : Azure for Students
Resource group : IOT
Start time : 11/22/2025, 10:18:54 PM
Correlation ID : 97927df5-b4be-41df-9db0-1730ad4e0bc8

Give feedback

Tell us about your experience with deployment

Step 3.2 :Creation of an input and output for the job


emna.feki@isimsf.u-sfax...
MINISTERE DE L'ENSEIGNEMENT...

IoT Hub
New input

Input alias *
EmnaAzureHub

Provide IoT Hub settings manually [✓ Select IoT Hub from your subscriptions]

Subscription
Azure for Students

IoT Hub *
EmnaAzureHub

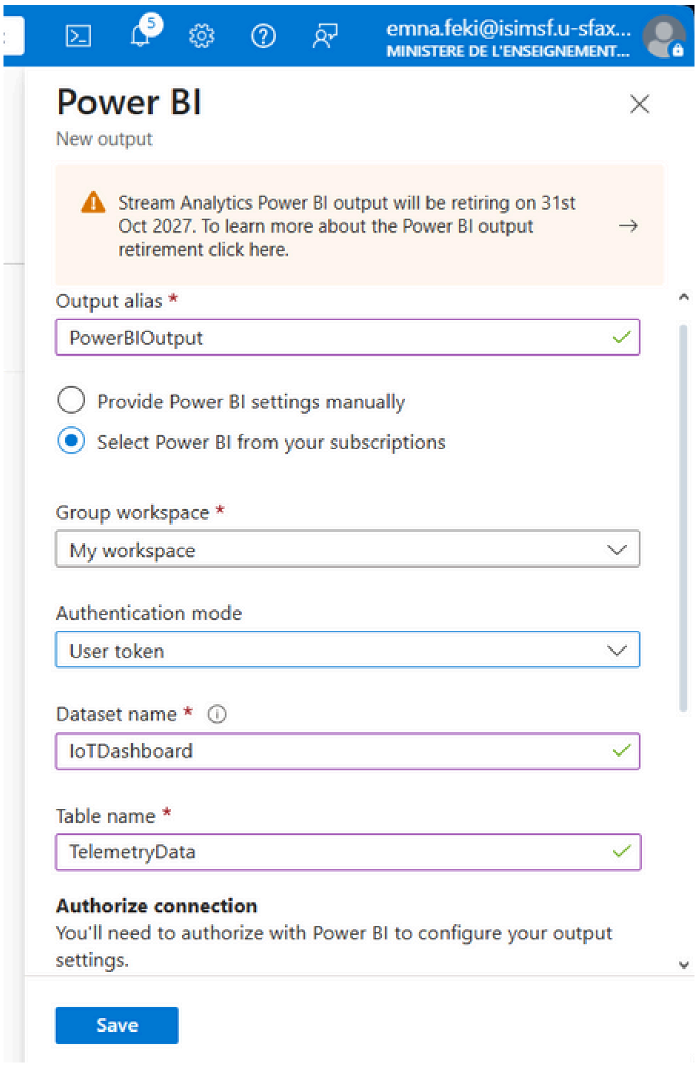
Consumer group *
$Default

Shared access policy name *
iothubowner

Shared access policy key
••••••••••••••••••••••••••••

Endpoint
Messaging

Save


emna.feki@isimsf.u-sfax...
MINISTERE DE L'ENSEIGNEMENT...

Power BI
New output

Stream Analytics Power BI output will be retiring on 31st Oct 2027. To learn more about the Power BI output retirement click here.

Output alias *
PowerBIOutput

Provide Power BI settings manually [✓ Select Power BI from your subscriptions]

Group workspace *
My workspace

Authentication mode
User token

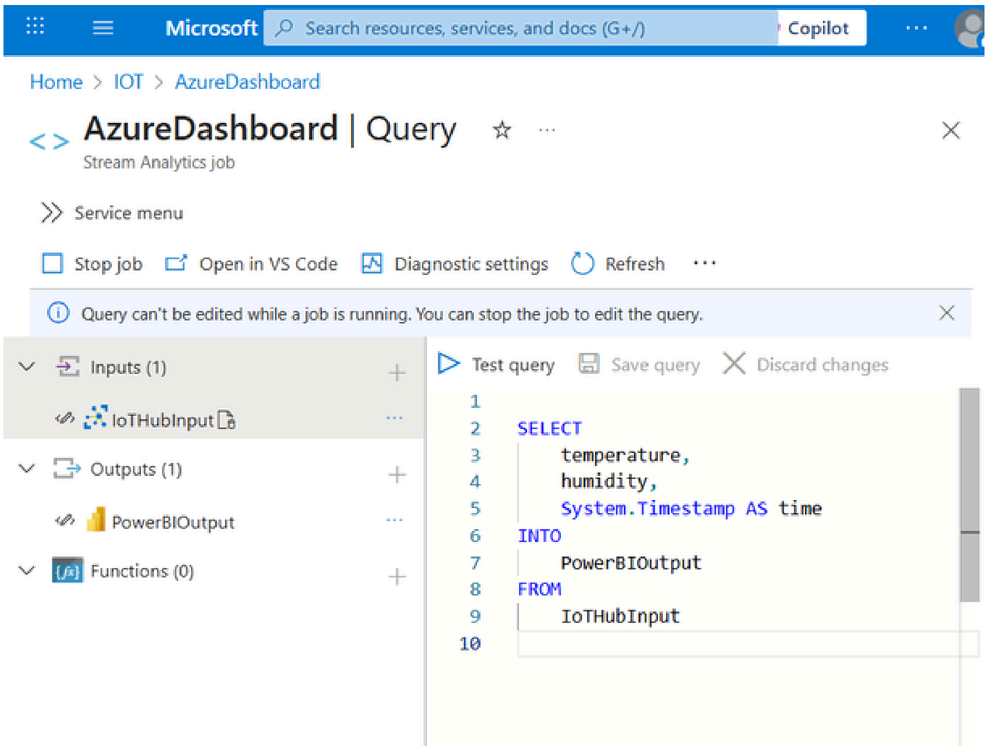
Dataset name *
IoTDashboard

Table name *
TelemetryData

Authorize connection
You'll need to authorize with Power BI to configure your output settings.

Save

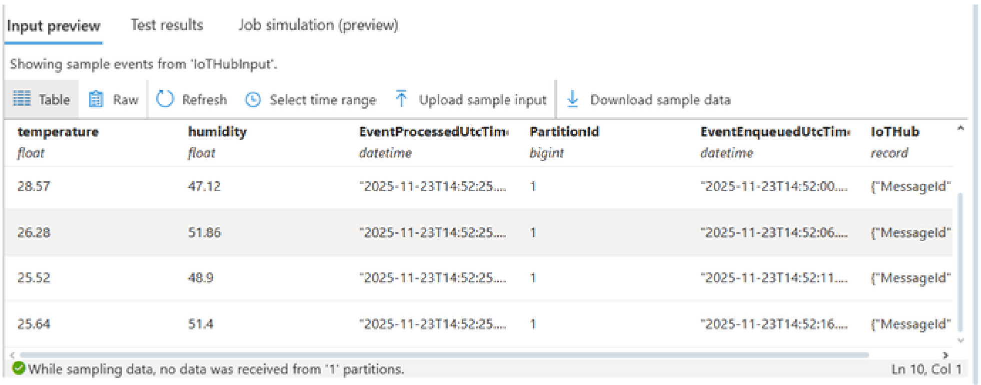
Step 3.3 :Create a query and start the job



Query can't be edited while a job is running. You can stop the job to edit the query.

```
SELECT
    temperature,
    humidity,
    System.Timestamp AS time
INTO
    PowerBIOutput
FROM
    IoTHubInput
```

Step 3.4 :Getting data from the script



Showing sample events from 'IoTHubInput'.

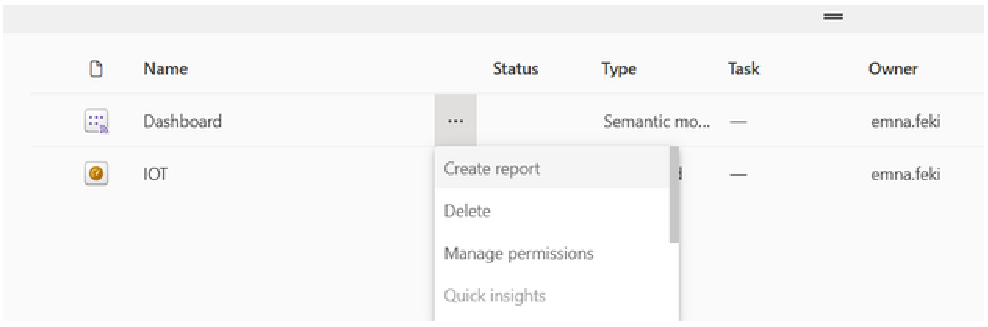
temperature *float*	humidity *float*	EventProcessedUtcTim *datetime*	PartitionId *bigint*	EventEnqueuedUtcTim *datetime*	IoTHub *record*
28.57	47.12	"2025-11-23T14:52:25....	1	"2025-11-23T14:52:00....	{"MessageId"
26.28	51.86	"2025-11-23T14:52:25....	1	"2025-11-23T14:52:06....	{"MessageId"
25.52	48.9	"2025-11-23T14:52:25....	1	"2025-11-23T14:52:11....	{"MessageId"
25.64	51.4	"2025-11-23T14:52:25....	1	"2025-11-23T14:52:16....	{"MessageId"

While sampling data, no data was received from '1' partitions.

Ln 10, Col 1

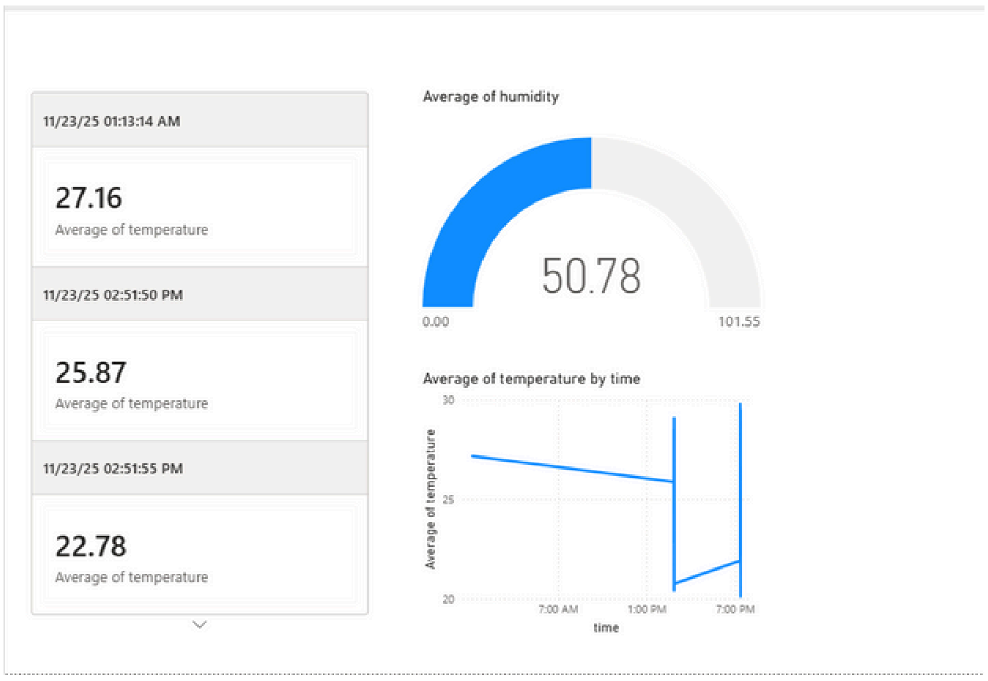
Part 4: Power BI Dashboard

Step 4.1 : Create a report for the DB already created in the output



Name	Status	Type	Task	Owner
Dashboard	...	Semantic mo...	—	emna.feki
IOT			—	emna.feki

Step 4.2 :Display Data



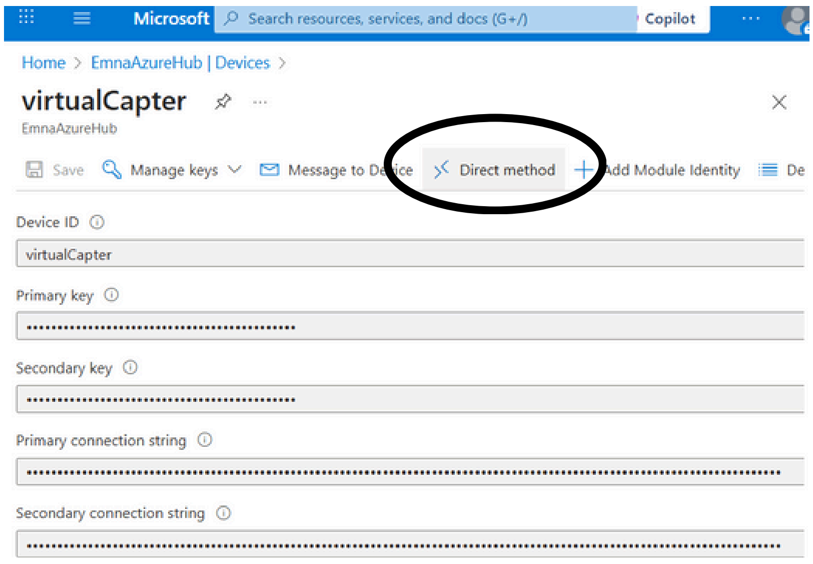
11/23/25 01:13:14 AM	27.16 Average of temperature
11/23/25 02:51:50 PM	25.87 Average of temperature
11/23/25 02:51:55 PM	22.78 Average of temperature

Average of humidity

50.78

Part 5: Downlink control

Step 5.1 : Create a direct method to the virtual device



virtualCapter

EmnaAzureHub

Device ID: virtualCapter

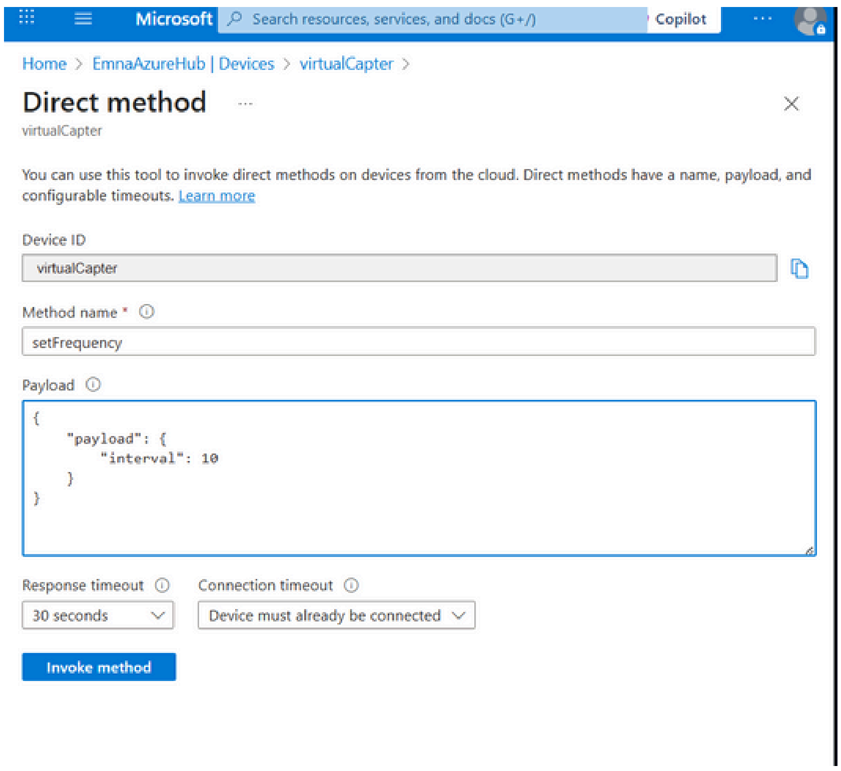
Primary key: ••

Secondary key: ••

Primary connection string: ••

Secondary connection string: ••

Step 5.2 :Code of the method


Microsoft
Search resources, services, and docs (G+/)
Copilot
Home > EmnaAzureHub | Devices > virtualCapter >
Invoke method

Direct method

virtualCapter

You can use this tool to invoke direct methods on devices from the cloud. Direct methods have a name, payload, and configurable timeouts. Learn more

Device ID: virtualCapter

Method name *: setFrequency

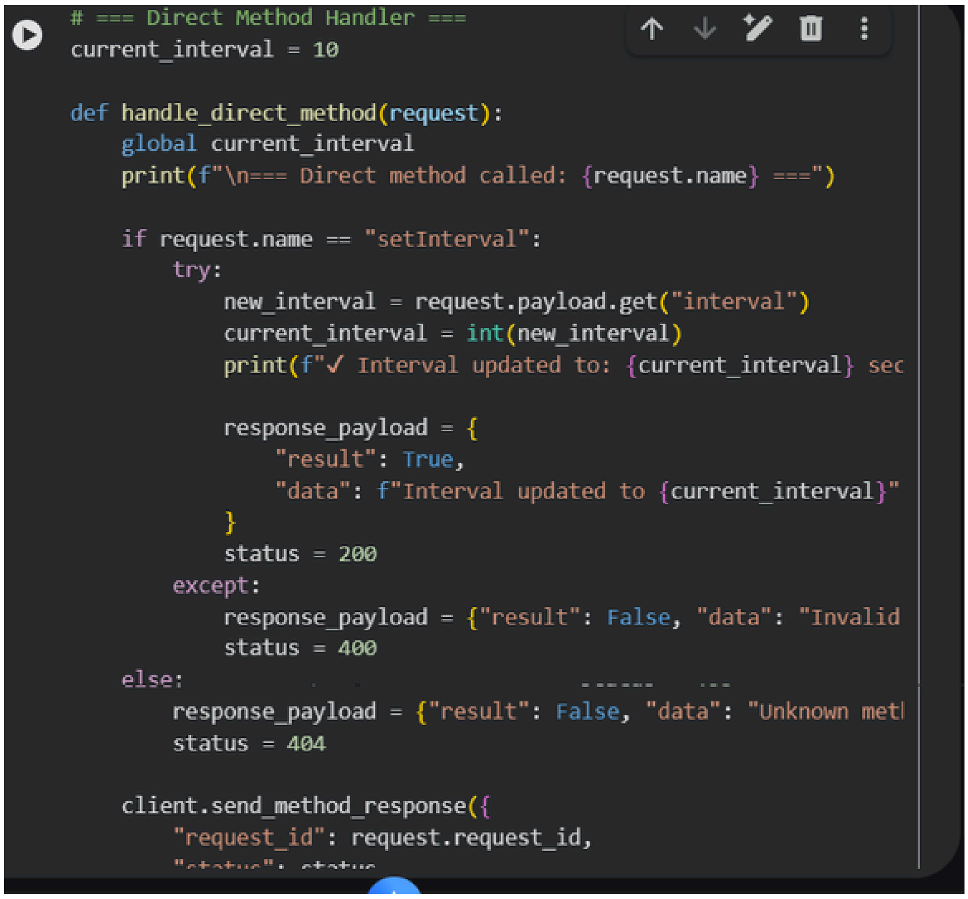
Payload:

```
{
    "payload": {
        "interval": 10
    }
}
```

Response timeout: 30 seconds

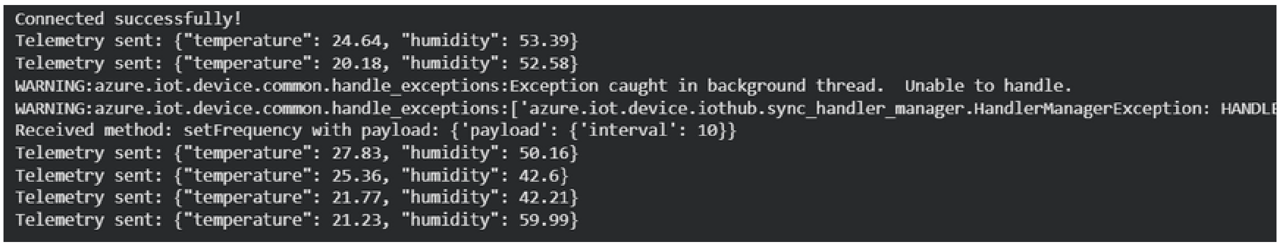
Connection timeout: Device must already be connected

Step 5.3: Edit the script by adding a method handler



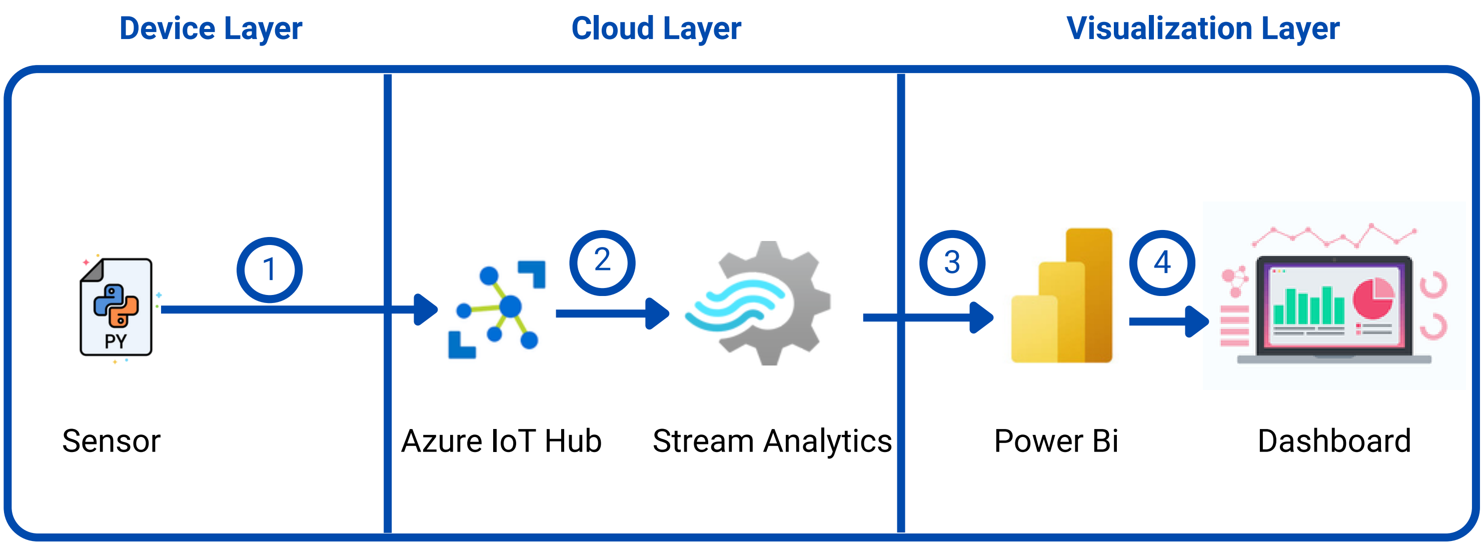
```
# === Direct Method Handler ===
current_interval = 10

def handle_direct_method(request):
    global current_interval
    print(f"\n=== Direct method called: {request.name} ===")

    if request.name == "setInterval":
        try:
            new_interval = request.payload.get("interval")
            current_interval = int(new_interval)
            print(f"✓ Interval updated to: {current_interval} sec

            response_payload = {
                "result": True,
                "data": f"Interval updated to {current_interval}"
            }
            status = 200
        except:
            response_payload = {"result": False, "data": "Invalid
            status = 400
    else:
        response_payload = {"result": False, "data": "Unknown met
        status = 404

    client.send_method_response({
        "request_id": request.request_id,
        "status": status
```

Step 5.4: The script get the method



```
Connected successfully!
Telemetry sent: {"temperature": 24.64, "humidity": 53.39}
Telemetry sent: {"temperature": 20.18, "humidity": 52.58}
WARNING:azure.iot.device.common.handle_exceptions:Exception caught in background thread.  Unable to handle.
WARNING:azure.iot.device.common.handle_exceptions:['azure.iot.device.iothub.sync_handler_manager.HandlerManagerException: HANDLE
Received method: setFrequency with payload: {'payload': {'interval': 10}}
Telemetry sent: {"temperature": 27.83, "humidity": 50.16}
Telemetry sent: {"temperature": 25.36, "humidity": 42.6}
Telemetry sent: {"temperature": 21.77, "humidity": 42.21}
Telemetry sent: {"temperature": 21.23, "humidity": 59.99}
```

Part 6: Architecture of the workflow



1. The Python script simulates a virtual sensor that sends temperature and humidity telemetry to Azure IoT Hub using the MQTT protocol, authenticated via the device connection string.

2. Azure IoT Hub receives the MQTT messages and forwards them as an input stream to the Azure Stream Analytics job for real-time processing.

3. The Stream Analytics job applies a query to transform and aggregate the IoT data, then outputs the processed dataset to Power BI for visualization.

4. Power BI consumes the dataset and renders a real-time dashboard with interactive charts, including a temperature line chart, a humidity gauge, and a summary card.