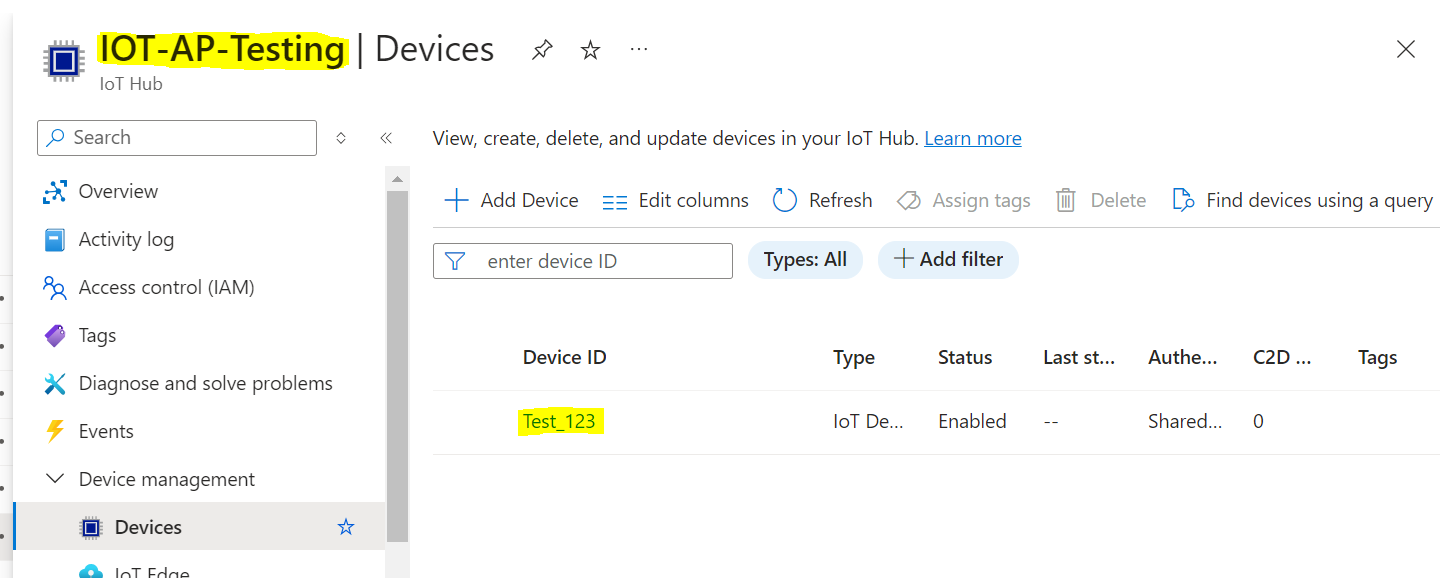
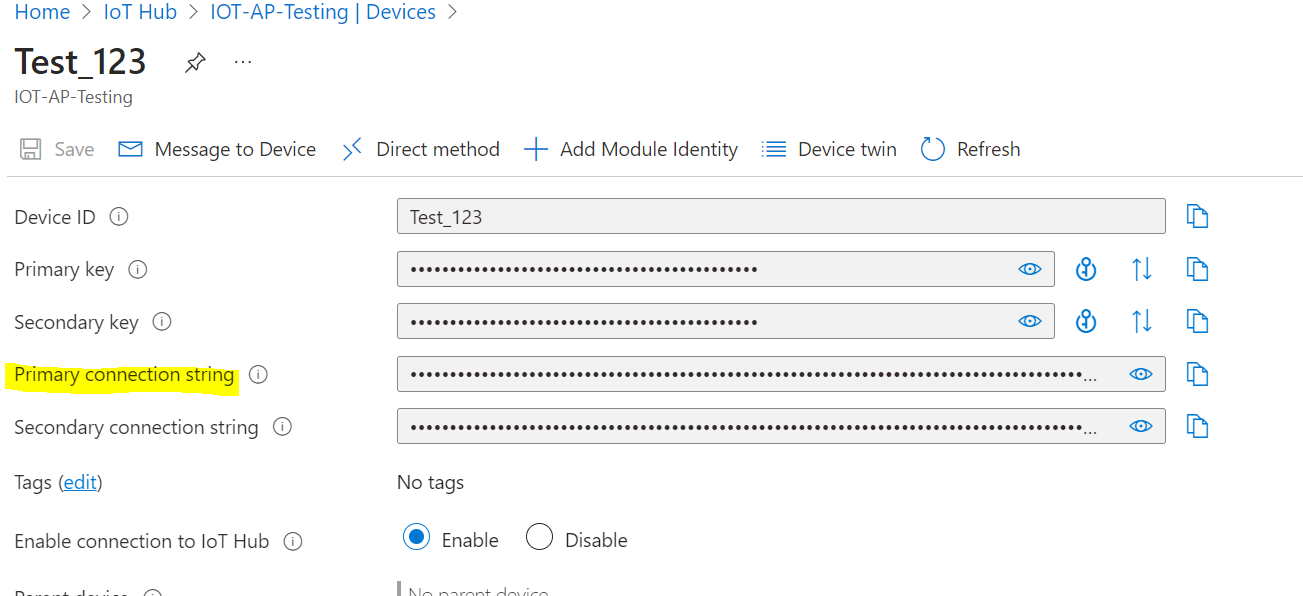
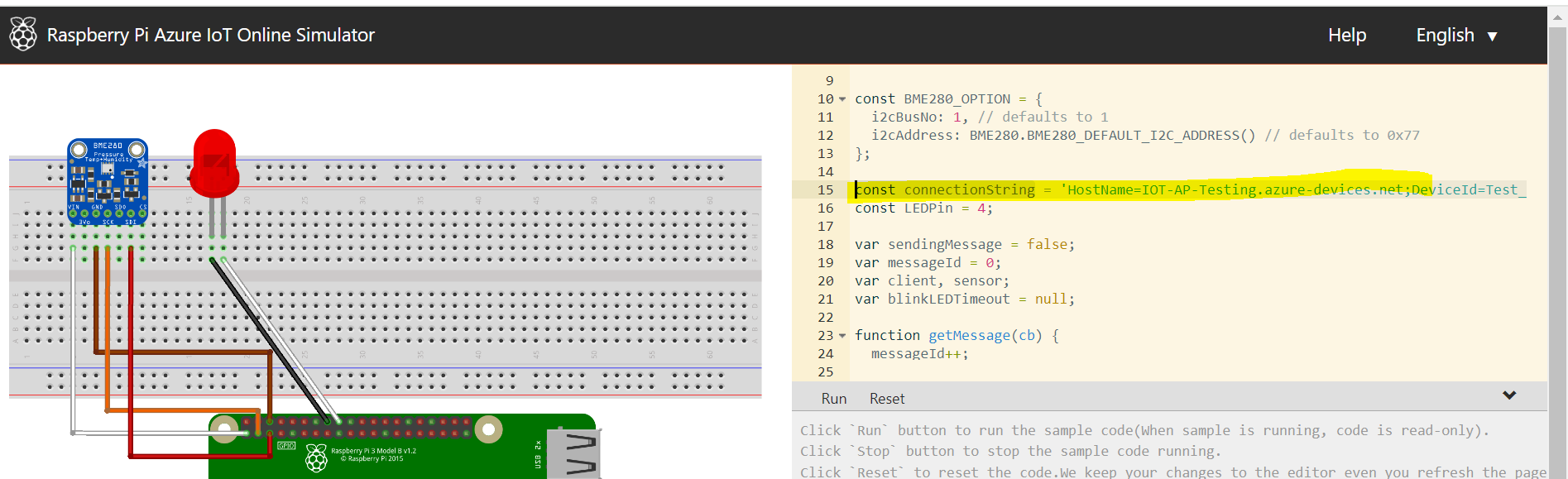
Low Frequency flow setup:

1. Create an IOT hub, and inside it creates an IOT device – which receives data through Raspberry pi simulator.

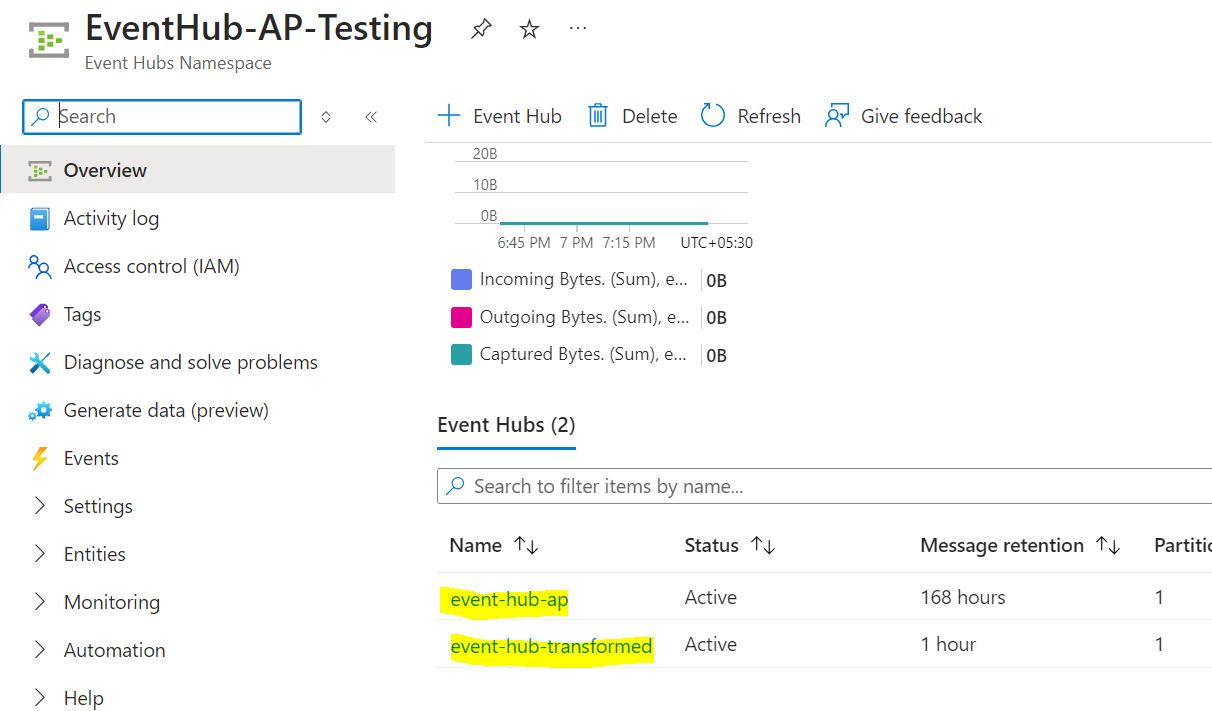


1. Use the Primary connection string of the IOT hub created, and paste it in “const connectionString” of Raspberry Pi simulator in order to simulate the data , which will flow to IOT hub .

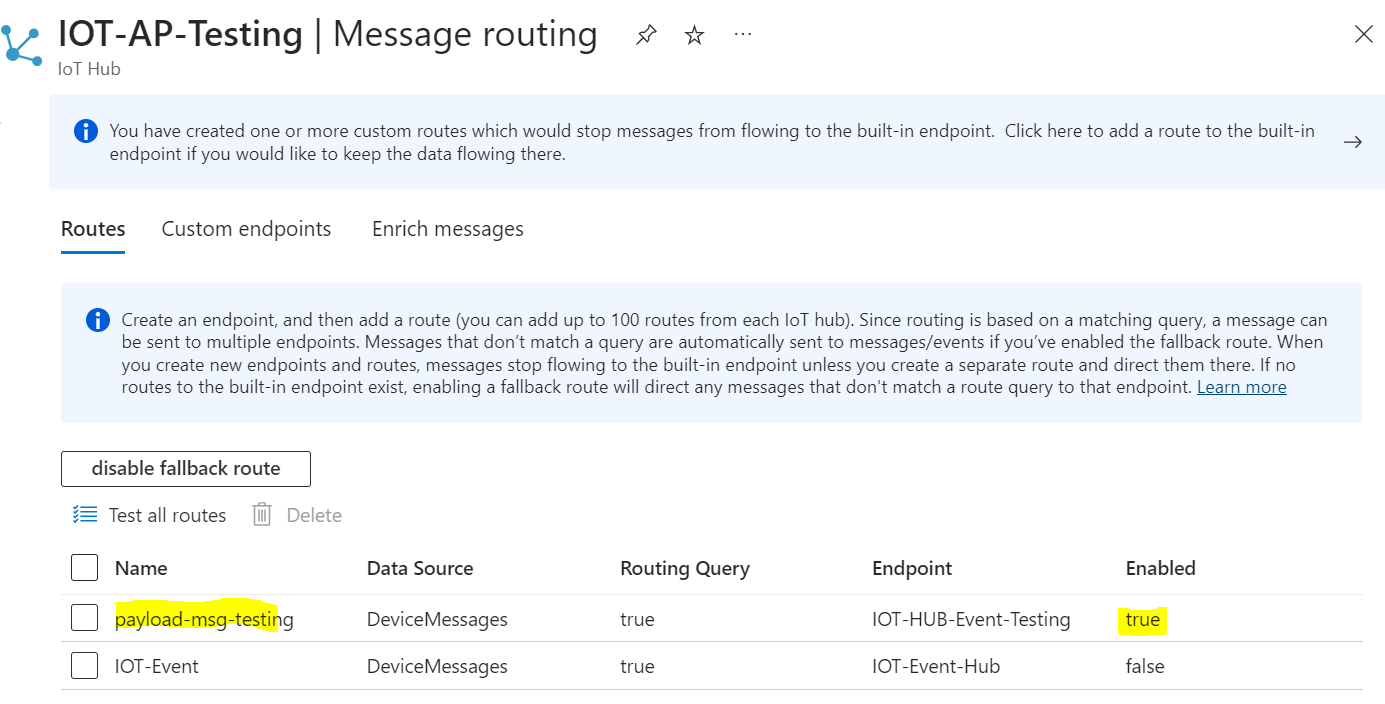


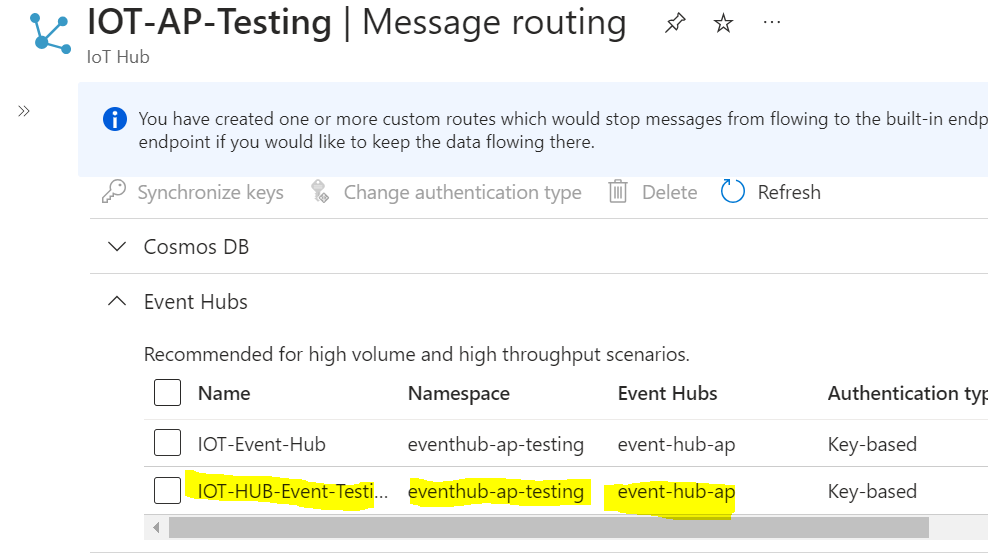


1. Create the 2 event hub, first event hub to received the data from IOT hub and second event hub in order to receive the transformed data from function app as below.

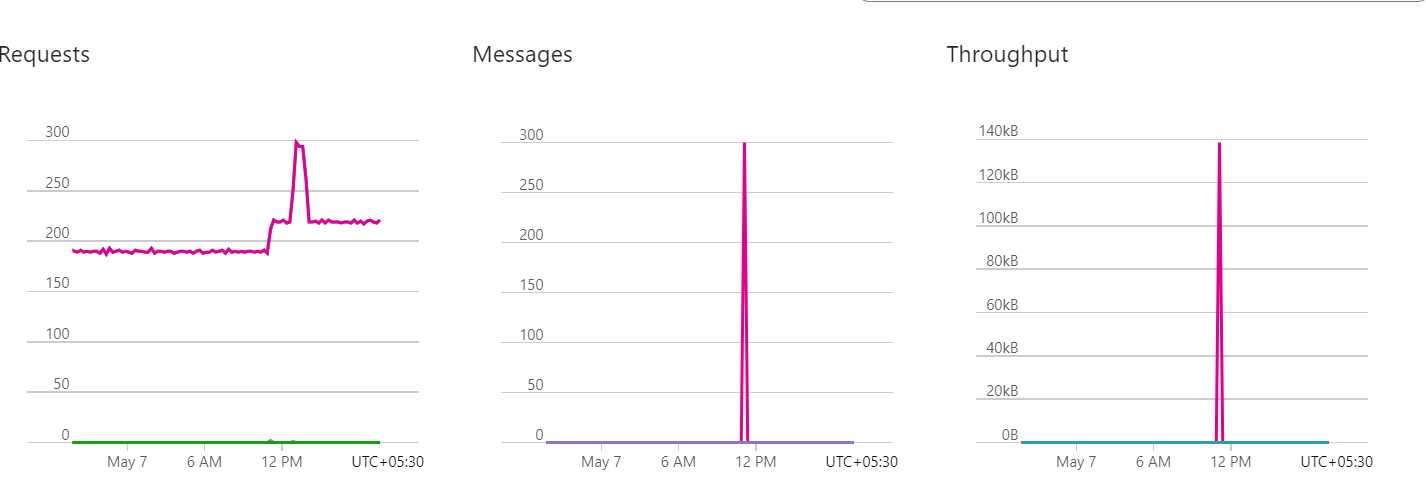


1. In order to establish the connection between IOT to Event hub , we need to (IOT hub à Message routing ) create routes and custom endpoints , which should contain the relevant information i.e “event hub namespace” name and “event hub” name of the event hub to which we want to redirect the messages to.





1. After the messages start redirecting from IOT hub to Event hub, below is the image will be seen in event hub in order to verify the messages that are being sent to event hub.



1. Create a Function app, which will push the data from event hub to another event hub using the below code.

Code :

import azure.functions as func

import logging

import os

import json

from azure.eventhub import EventHubProducerClient, EventData

app = func.FunctionApp()

@app.event\_hub\_message\_trigger(arg\_name="azeventhub", event\_hub\_name="event-hub-ap",

                               connection="SOURCE\_CONN\_STR")

def fn\_sample(azeventhub: func.EventHubEvent):

    logging.info('Python EventHub trigger processed an event: %s',

                azeventhub.get\_body().decode('utf-8'))

    # Decode the message body from bytes to string

    message\_body = azeventhub.get\_body().decode('utf-8')

    # Deserialize the JSON data

    event\_data = json.loads(message\_body)

    # Log the output

    logging.info(json.dumps(event\_data, indent=4))

    logging.info(f"output for first event hub is {json.dumps(event\_data, indent=4)}")

    try:

        # Send the data to another Event Hub

        target\_event\_hub = os.environ["DESTINATION\_EVENTHUB\_NAME"]

        connection\_str = os.environ["DESTINATION\_CONN\_STR"]

        producer\_client = EventHubProducerClient.from\_connection\_string(connection\_str)

        event\_data\_batch = producer\_client.create\_batch()

        event\_data\_batch.add(EventData(message\_body))

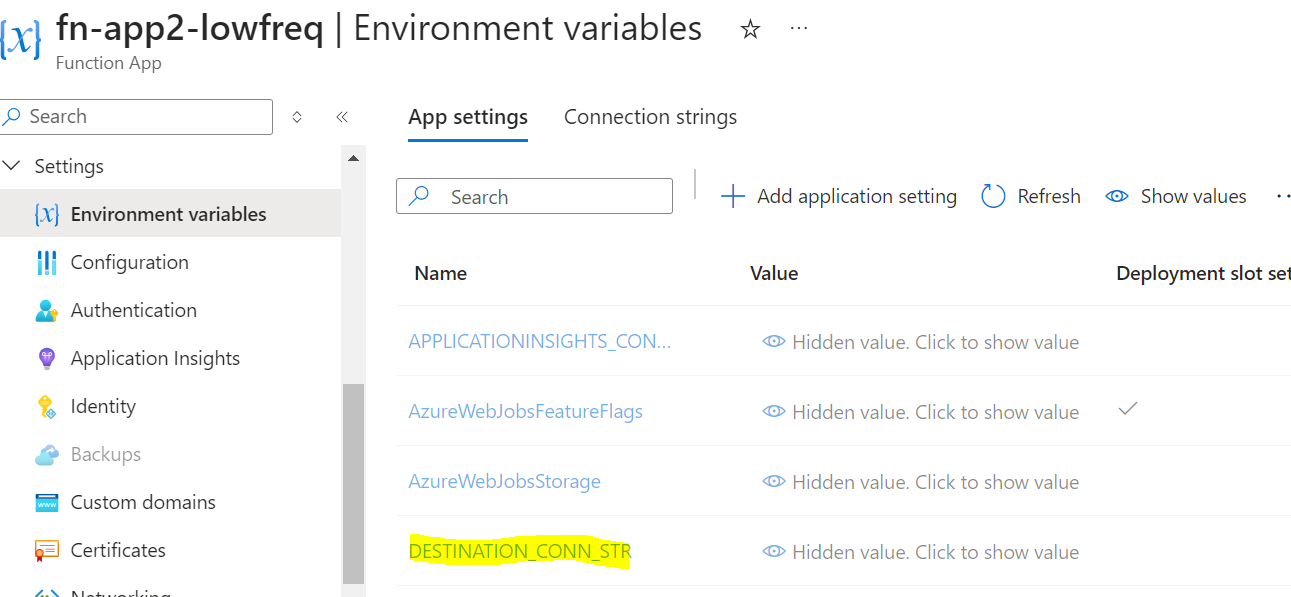
        producer\_client.send\_batch(event\_data\_batch, eventhub\_name=target\_event\_hub)

        logging.info("Sent data to destination Event Hub: %s", target\_event\_hub)

    except Exception as e:

        logging.error("Error sending data to destination Event Hub: %s", e)

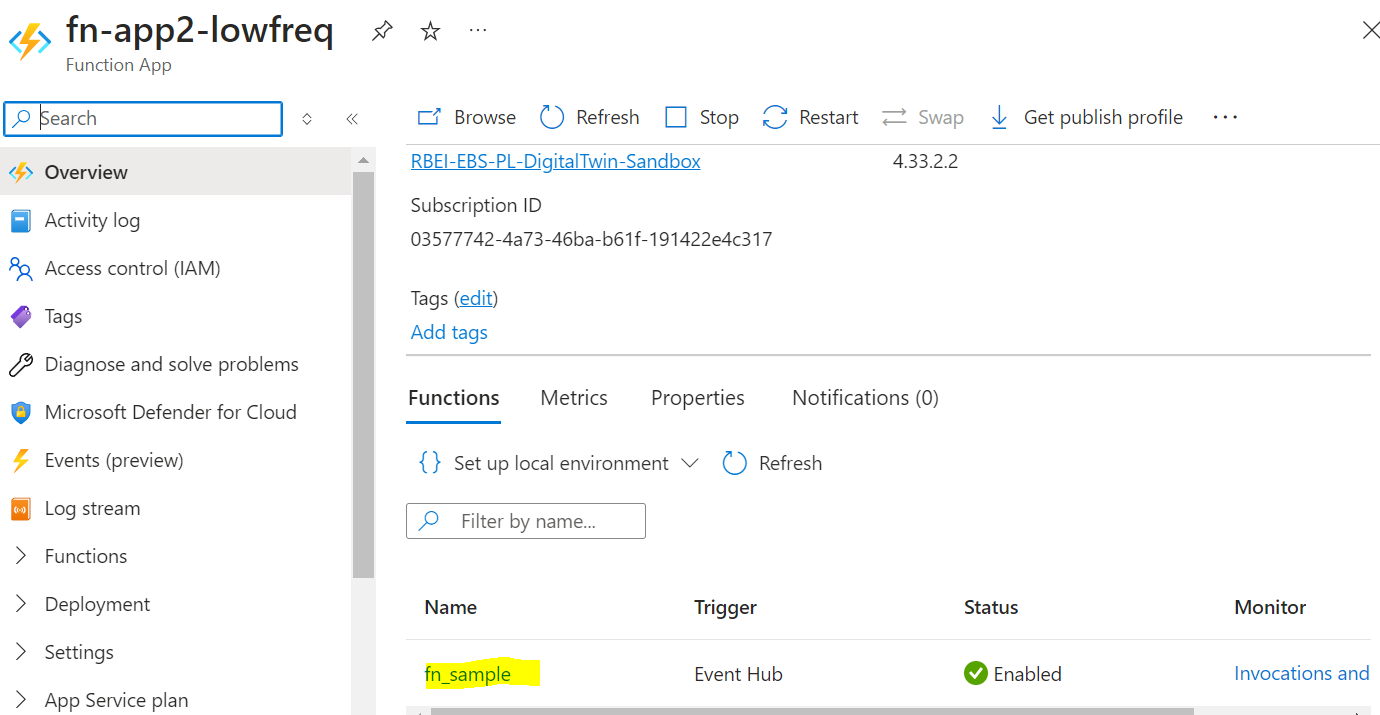
Note : the code is written in FastAPI , and also the connection string of the destination event hub are stored in environment variable in function app.



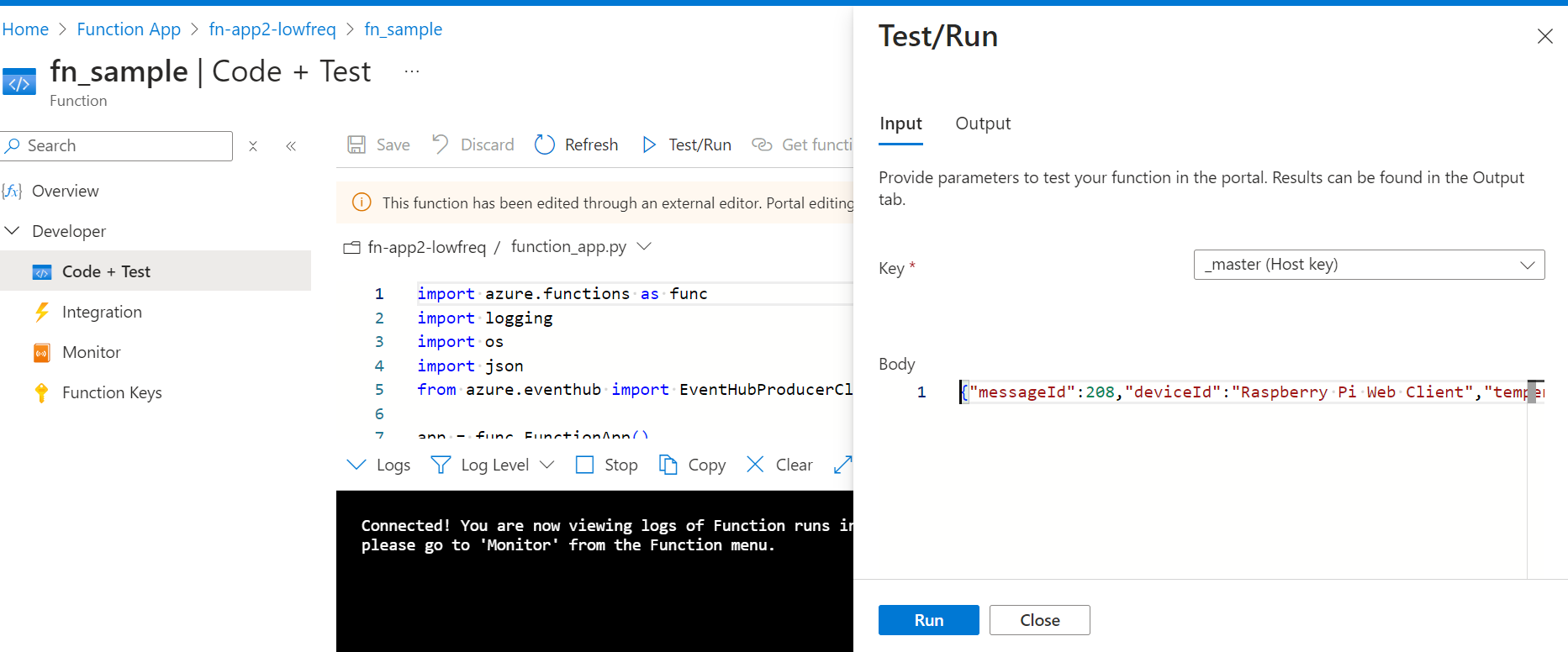
1. After the code has been written in Visual studio, the code needs to be deployed using visual studio to functions.

Workspace -à deploy to function app.

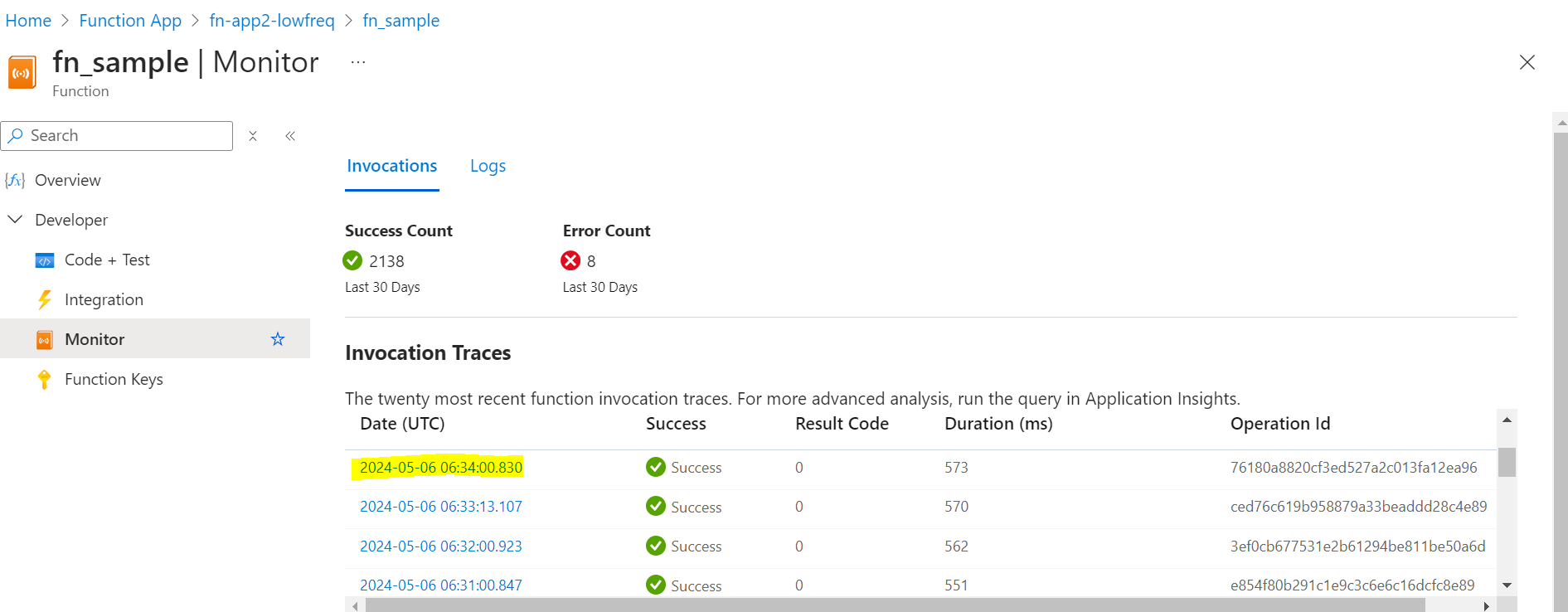
After the deployment is successful the function will appear in azure portal with updated code as below.



Can verify the code and test it , by giving a sample Json input to it and running it .

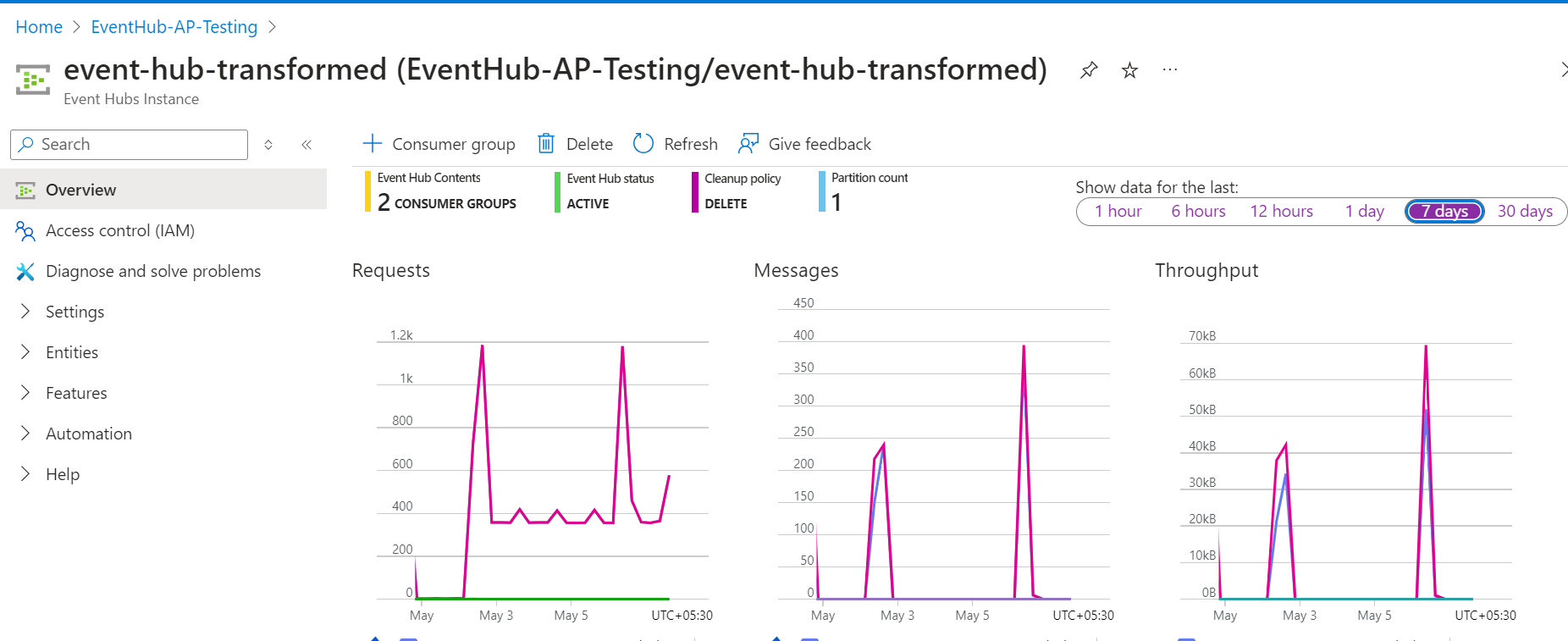


Can monitor the output and verify for any errors in the code by clicking on the monitor tab of the function app.



Note: since the function app is created as EventHub based trigger, the function app will be triggered every time whenever a new message arrives in EventHub.

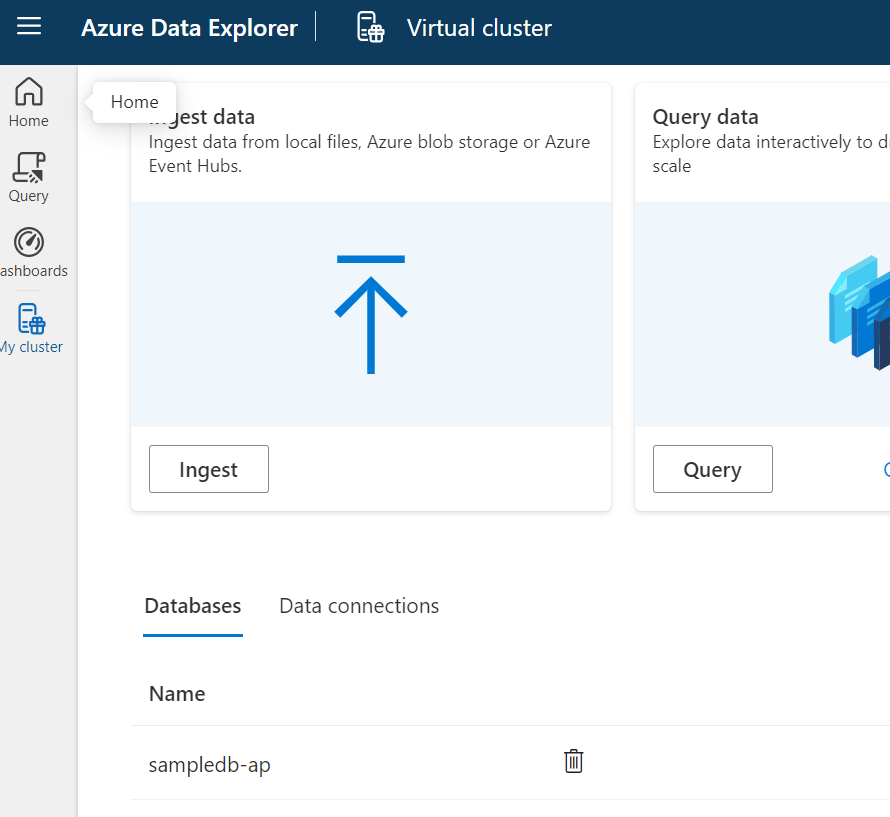
1. After the function app triggers and does the required transformation, the data is pushed into new event hub as below.

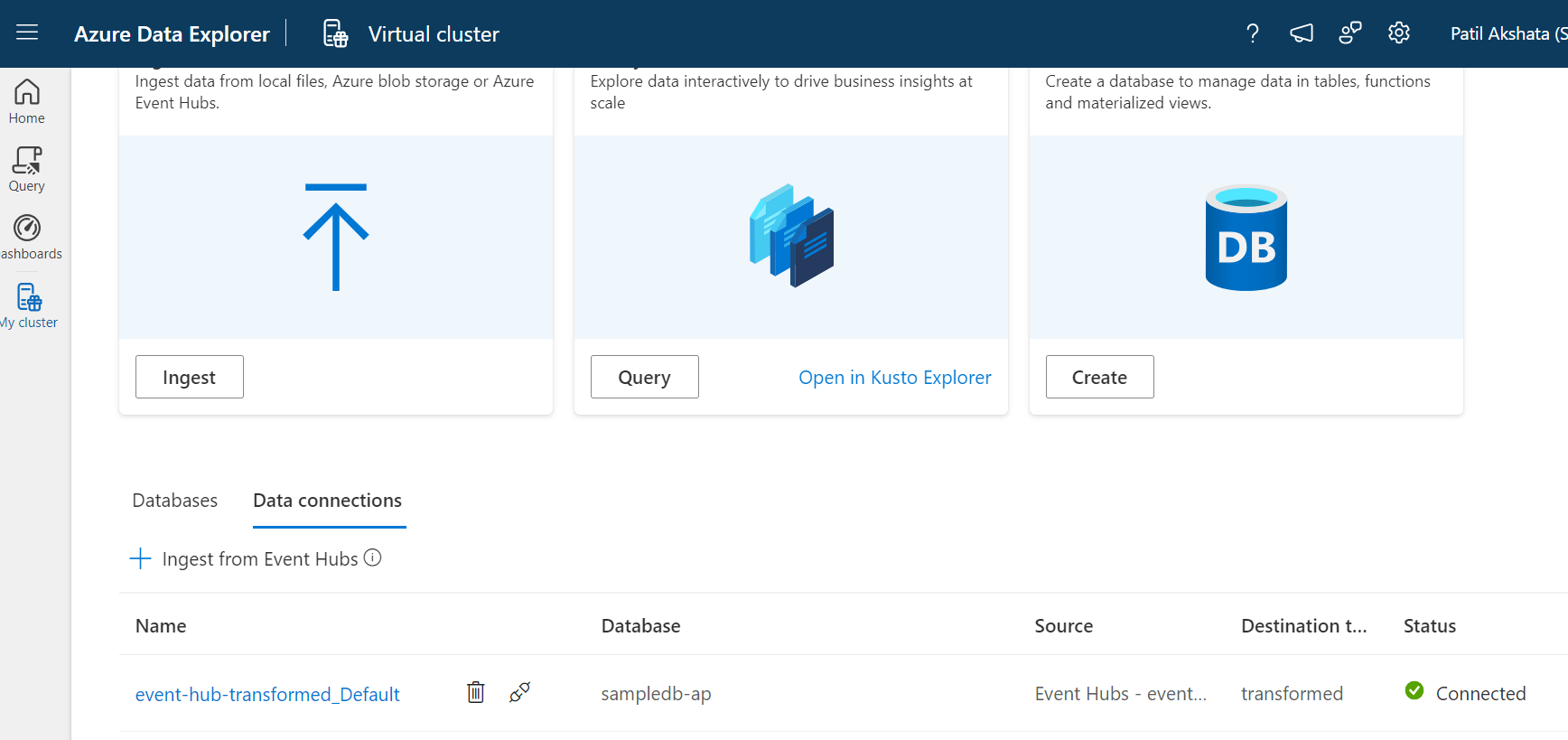


1. After the data is pushed to new event hub, it needs to be pushed to ADX to store the data .

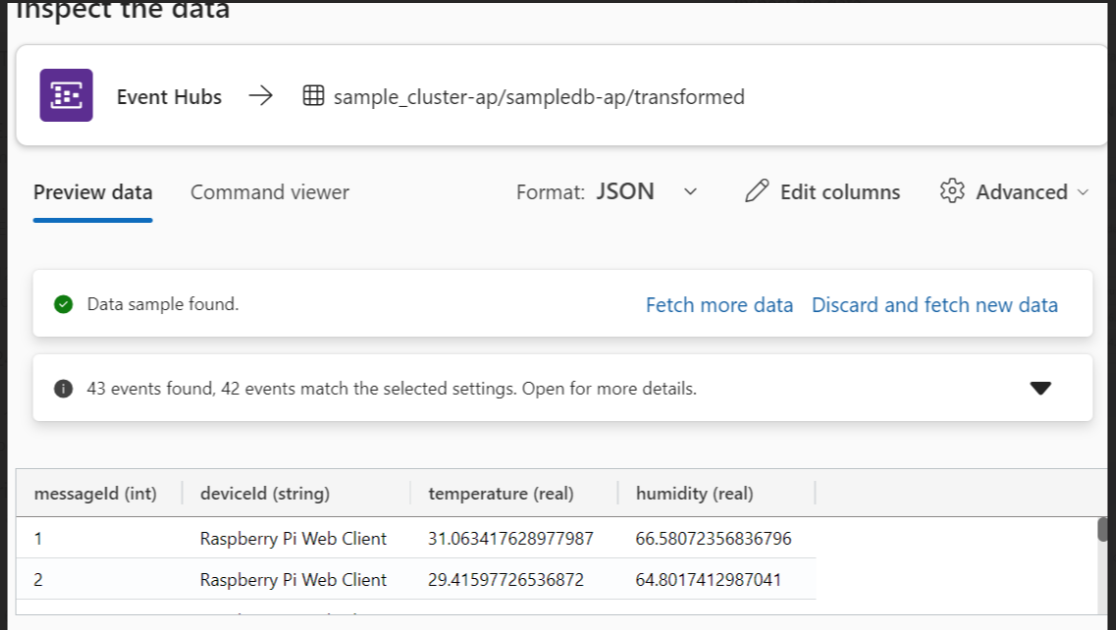
Create a cluster with the required information, and open the ADX data explorer (<https://dataexplorer.azure.com/clusters/kvc-nbqswgzk0xru09yp0v.northeurope/databases/adx_database>)

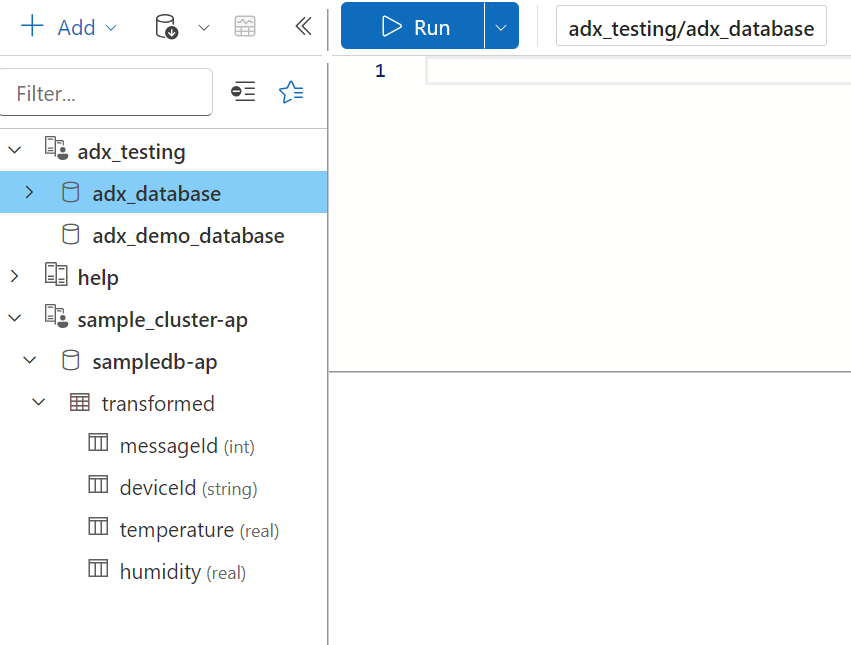
and establish a connection to new event hub and create a database and table as below.



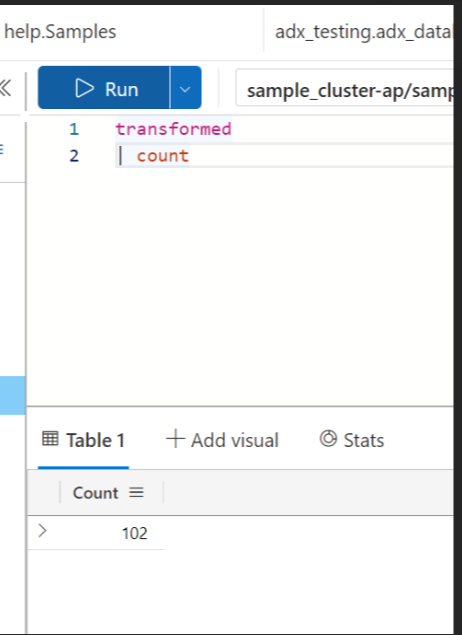


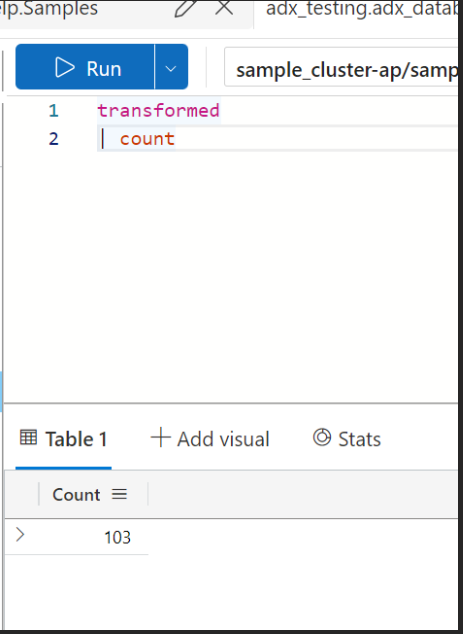
After the connection to event hub is successful, we can see few data getting fetched as below:





1. Keep monitoring and verifying the count from ADX and there will be an increase in count of the messages as data flows from raspberry pi -à IOT HUB à EVENT HUB à FUNCTION APP -à NEW EVENT HUB -à ADX as below.





**Summary:**

RaspberryPI à IOT Hub à Event Hub (Sender) à Azure Function App à Event Hub ( Receiver) à ADX