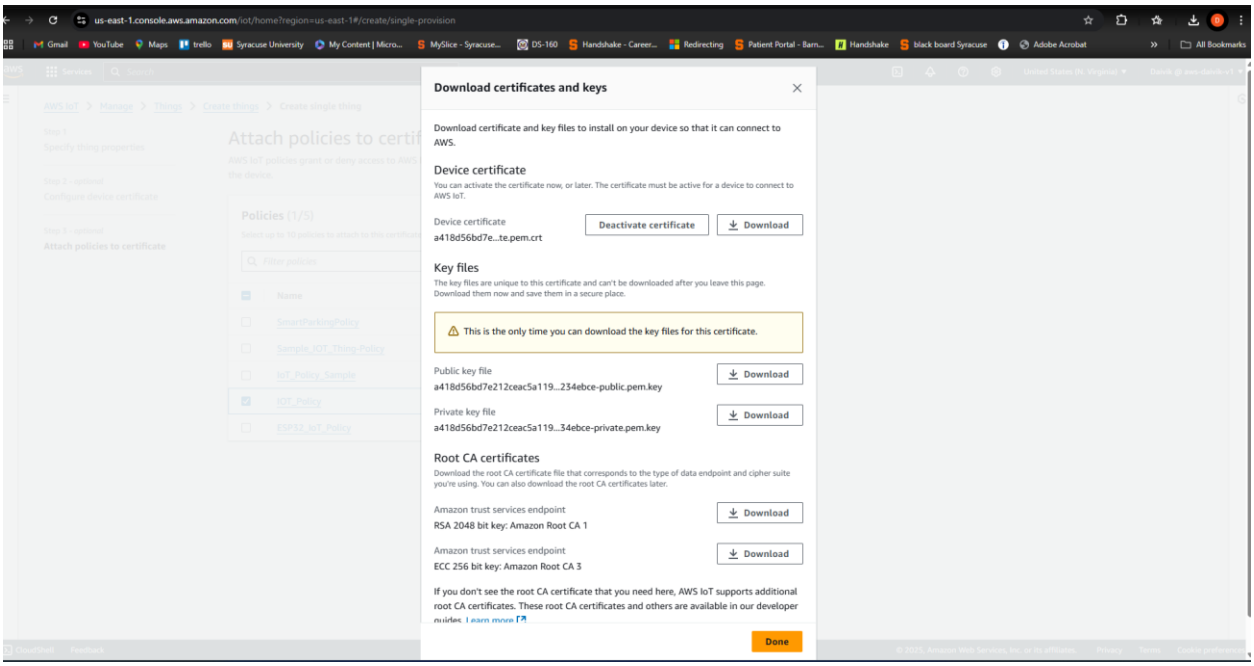
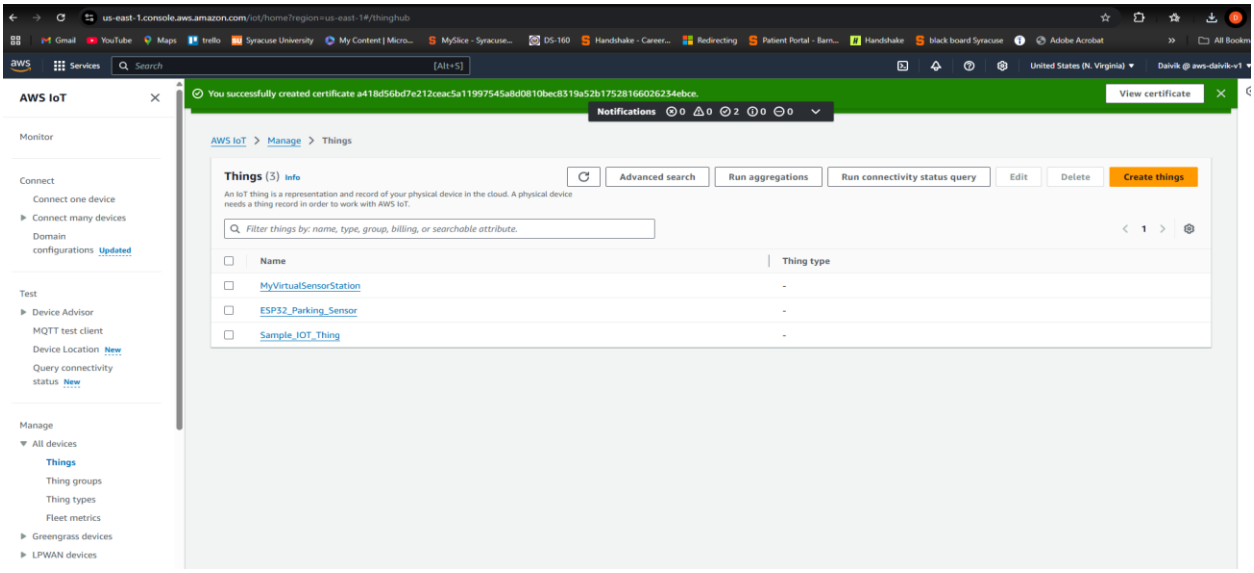
CIS600 Internet of Things: Application Development

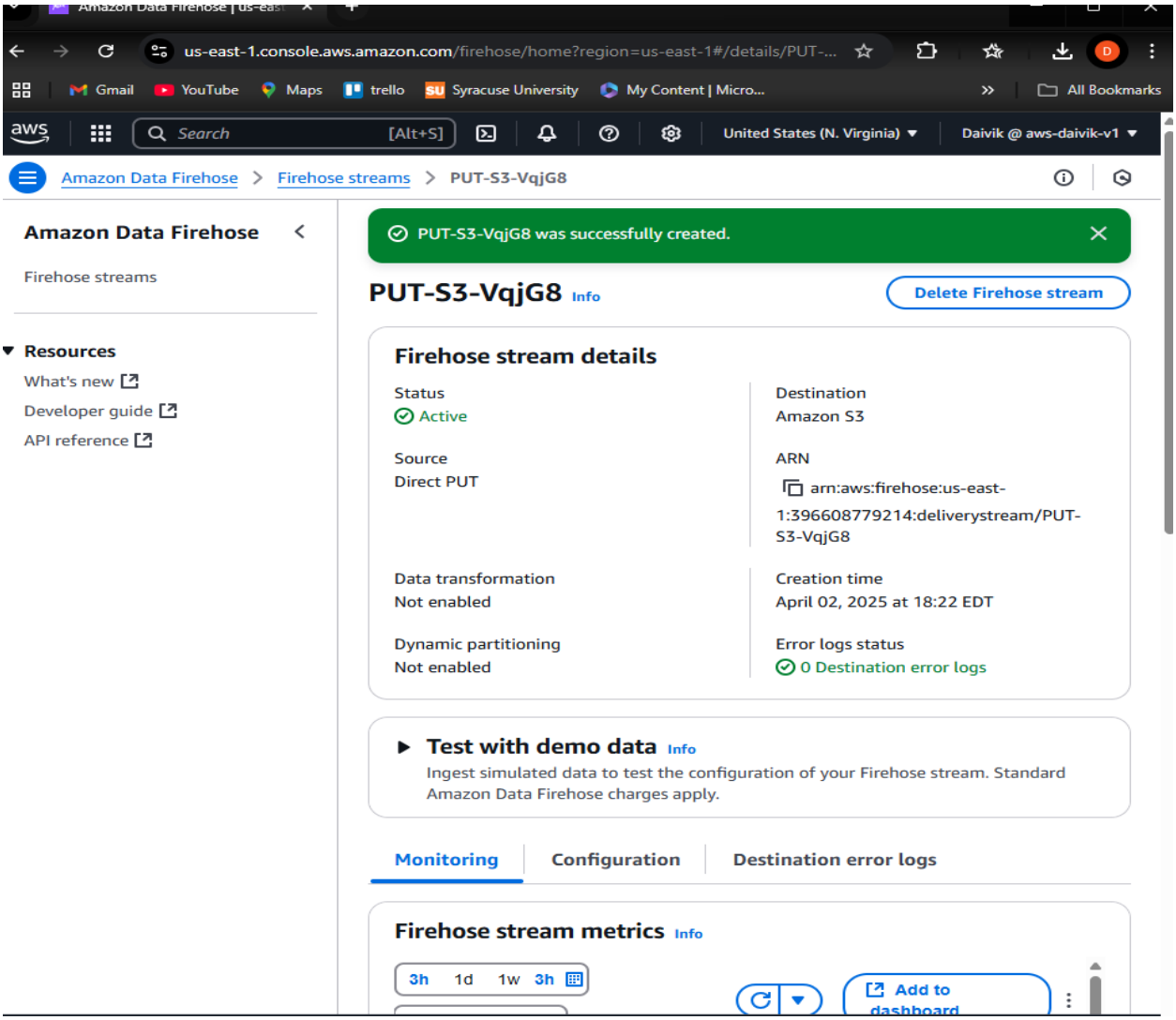
Step-by-Step Implementation:

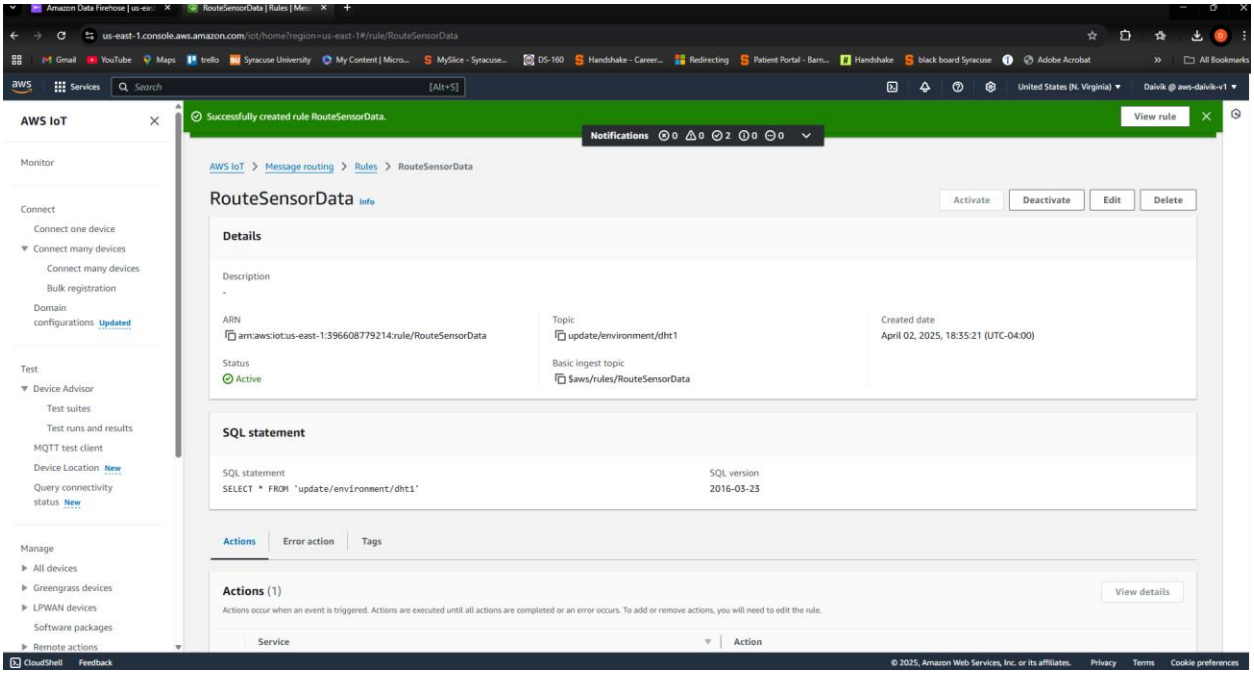
- Python MQTT Simulator
  - ✓ Developed a Python script (mqtt_aws_simulator.py) to simulate environmental sensor data (temperature, humidity, and CO_2).
  - ✓ Set up secure MQTT communication using paho-mqtt and AWS IoT certificates.
  - ✓ Generated random sensor data and published to the topic: update/environment/dht1.
- AWS IoT Core Configuration
  - ✓ Created a Thing and downloaded certificates (certificate, private key, public key, root CA).
  - ✓ Used endpoint from AWS IoT Core (*.amazonaws.com) and configured it in the simulator.
- Test Connection
  - ✓ Used AWS IoT Core's MQTT test client to verify data publishing on the topic.
  - ✓ Successfully saw live messages appearing in the test client window.
- Create AWS IoT Rule
  - ✓ Created a rule named RouteSensorData.
    SQL Query:
    SELECT * FROM 'update/environment/dht1'
  - ✓ The rule was configured to route messages to the AWS IoT Analytics pipeline.
- IoT Analytics Setup
  - ✓ Channel: Created to collect raw MQTT data.
  - ✓ Pipeline: Connected the channel to a data store.
  - ✓ Data Store: Configured to store incoming JSON data.
  - ✓ Dataset: Created a SQL query to retrieve and preview sensor data.
- Testing and Monitoring
  - ✓ Monitored message ingestion in AWS IoT Analytics → Channels.
  - ✓ Verified data pipeline and successful dataset generation.

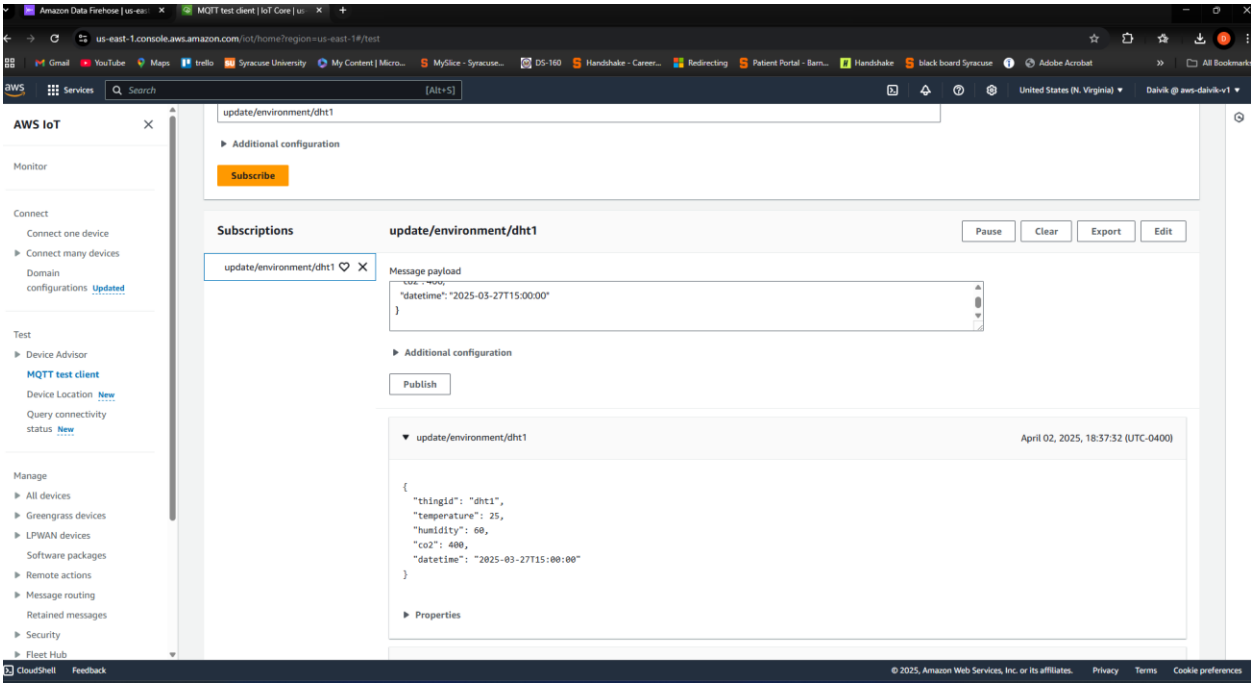
Screenshots of Output:

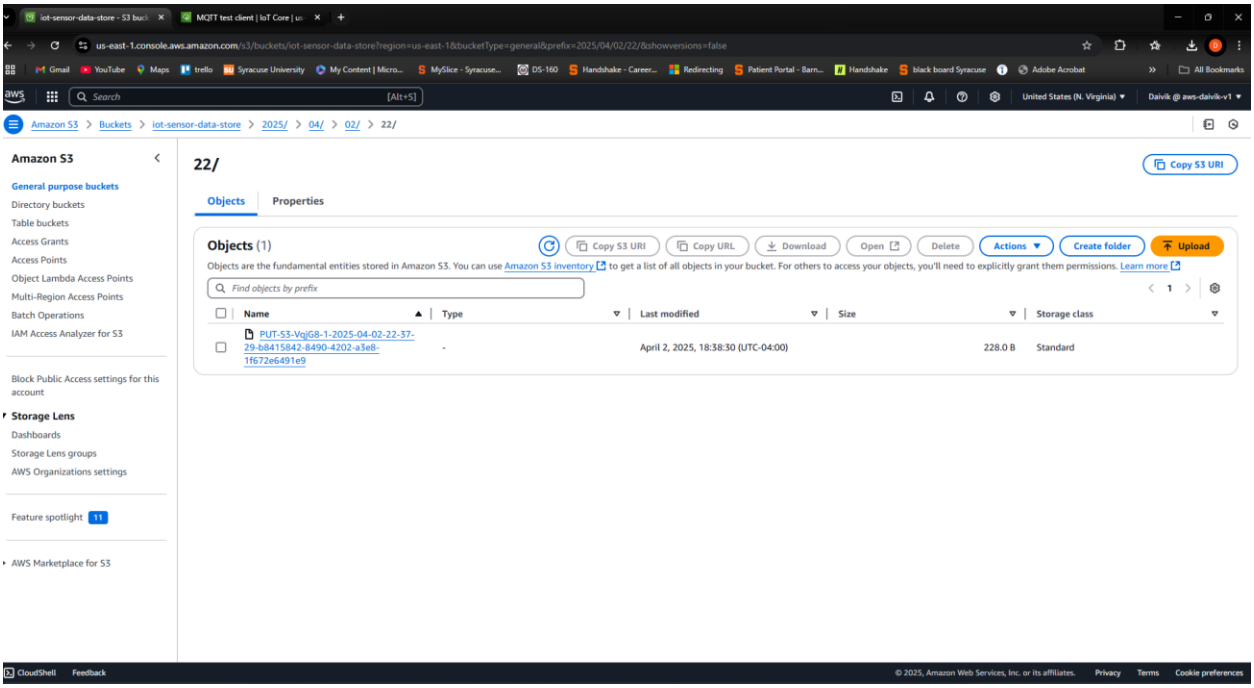


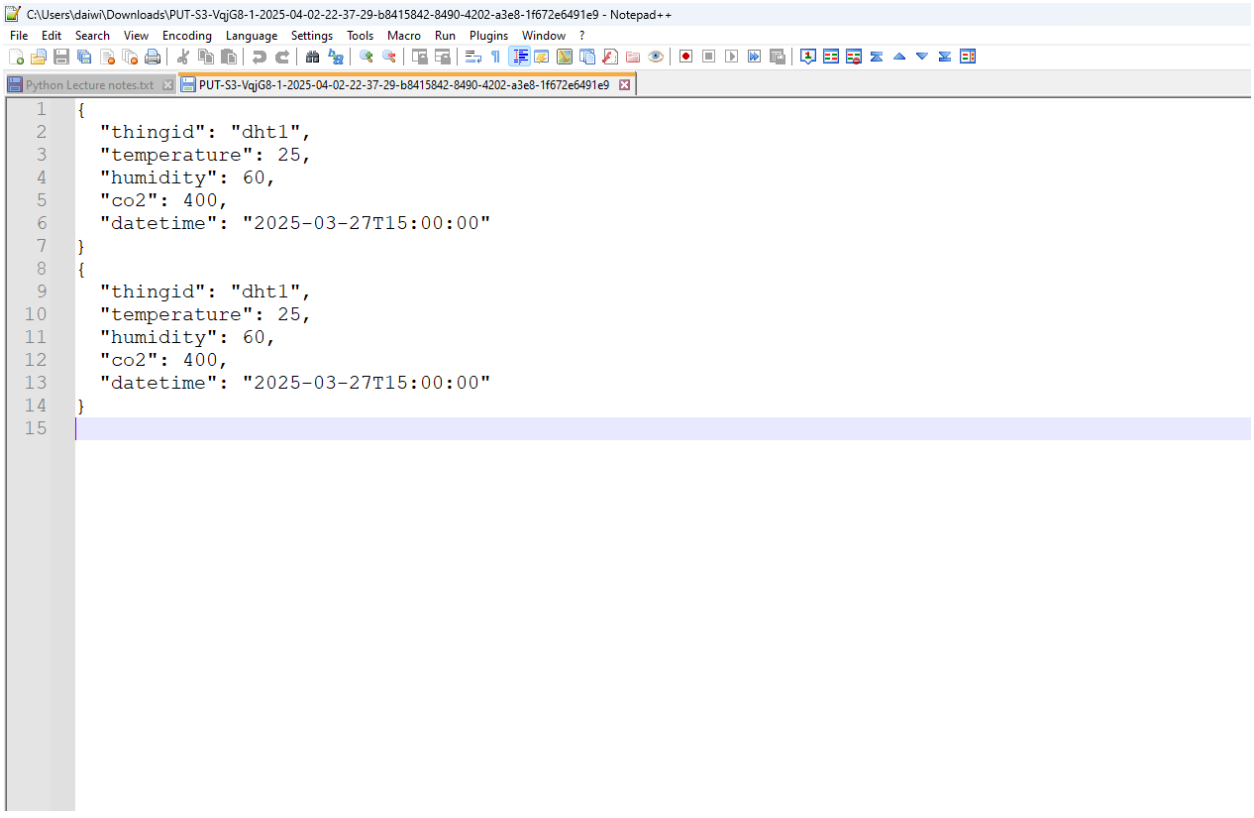

AWS IoT
You successfully created certificate a418d56bd7e212ceac5a11997545a8d0810bec8319a52b17528166026234ebce.
View certificate
Notifications
AWS IoT > Manage > Things
Things (3) Info
An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.
Advanced search
Run aggregations
Run connectivity status query
Edit
Delete
Create things
Filter things by: name, type, group, billing, or searchable attribute.
Name
Thing type
MyVirtualSensorStation
ESP32_Parking_Sensor
Sample_IOT_Thing
Monitor
Connect
Connect one device
Connect many devices
Domain configurations Updated
Test
Device Advisor
MQTT test client
Device Location New
Query connectivity status New
Manage
All devices
Things
Thing groups
Thing types
Fleet metrics
Greengrass devices
LPWAN devices

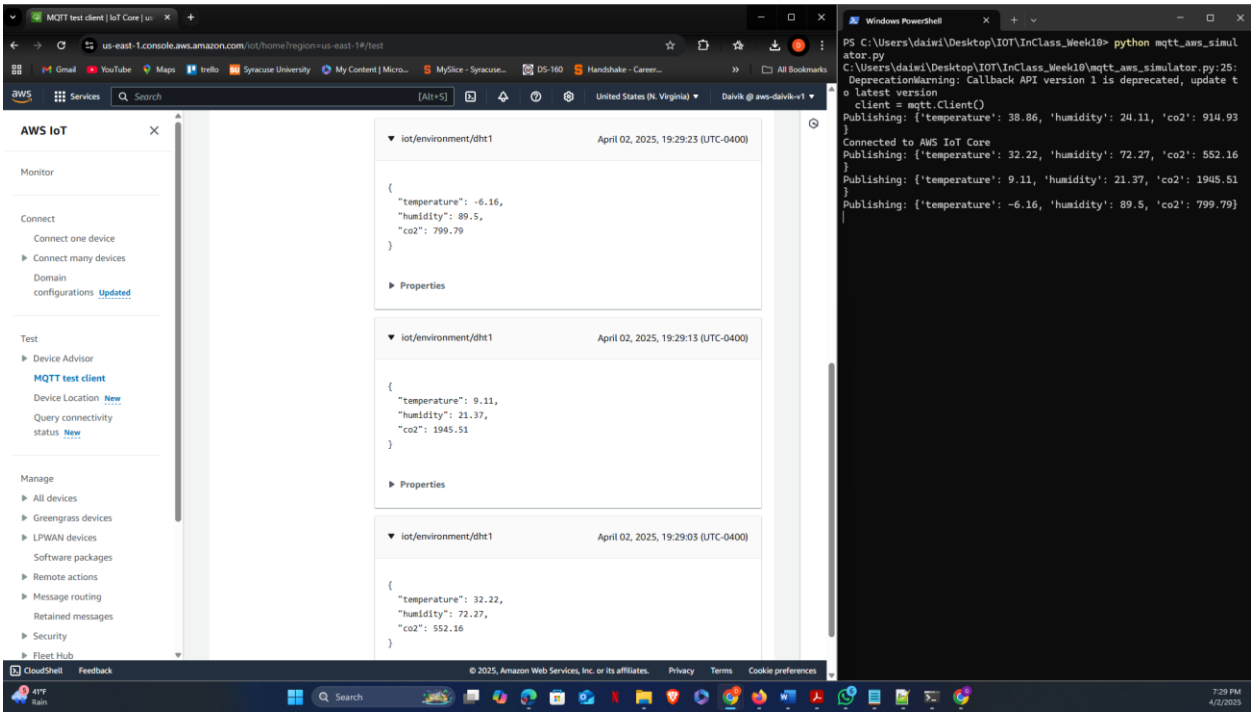

us-east-1.console.aws.amazon.com/firehose/home?region=us-east-1#/details/PUT-...
Gmail
YouTube
Maps
trello
Syracuse University
My Content | Micro...
All Bookmarks
Search
[Alt+S]
United States (N. Virginia)
Daivik @ aws-daivik-v1
Amazon Data Firehose > Firehose streams > PUT-S3-VqjG8
Amazon Data Firehose
Firehose streams
Resources
What's new
Developer guide
API reference
PUT-S3-VqjG8 was successfully created.
PUT-S3-VqjG8 Info
Delete Firehose stream
Firehose stream details
Status
Active
Source
Direct PUT
Destination
Amazon S3
ARN
arn:aws:firehose:us-east-1:396608779214:deliverystream/PUT-S3-VqjG8
Data transformation
Not enabled
Dynamic partitioning
Not enabled
Creation time
April 02, 2025 at 18:22 EDT
Error logs status
0 Destination error logs
Test with demo data Info
Ingest simulated data to test the configuration of your Firehose stream. Standard Amazon Data Firehose charges apply.
Monitoring
Configuration
Destination error logs
Firehose stream metrics Info
3h
1d
1w
3h
Add to dashboard


Successfully created rule RouteSensorData.
View rule
AWS IoT > Message routing > Rules > RouteSensorData
RouteSensorData Info
Activate
Deactivate
Edit
Delete
Details
Description
ARN
arn:aws:iot:us-east-1:396608779214:rule/RouteSensorData
Topic
update/environment/dht1
Created date
April 02, 2025, 18:35:21 (UTC-04:00)
Status
Active
Basic ingest topic
$aws/rules/RouteSensorData
SQL statement
SQL statement
SELECT * FROM 'update/environment/dht1'
SQL version
2016-03-23
Actions
Error action
Tags
Actions (1)
View details
Service
Action


update/environment/dht1
Additional configuration
Subscribe
Subscriptions
update/environment/dht1
Pause
Clear
Export
Edit
update/environment/dht1
Message payload
"datetime": "2025-03-27T15:00:00"
Additional configuration
Publish
update/environment/dht1
April 02, 2025, 18:37:32 (UTC-0400)
"thingid": "dht1",
"temperature": 25,
"humidity": 60,
"co2": 400,
"datetime": "2025-03-27T15:00:00"
Properties


Amazon S3 > Buckets > iot-sensor-data-store > 2025/ > 04/ > 02/ > 22/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
22/
Copy S3 URI
Objects
Properties
Objects (1)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more
Find objects by prefix
Name
Type
Last modified
Size
Storage class
PUT-S3-VqjG8-1-2025-04-02-22-37-29-b8415842-8490-4202-a3e8-1f672e6491e9
April 2, 2025, 18:38:30 (UTC-04:00)
228.0 B
Standard
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


C:\Users\daiwi\Downloads\PUT-S3-VqjG8-1-2025-04-02-22-37-29-b8415842-8490-4202-a3e8-1f672e6491e9 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Python Lecture notes.txt
PUT-S3-VqjG8-1-2025-04-02-22-37-29-b8415842-8490-4202-a3e8-1f672e6491e9
{
"thingid": "dht1",
"temperature": 25,
"humidity": 60,
"co2": 400,
"datetime": "2025-03-27T15:00:00"
}
{
"thingid": "dht1",
"temperature": 25,
"humidity": 60,
"co2": 400,
"datetime": "2025-03-27T15:00:00"
}



Reflection:

I obtained practical experience simulating real-time data from virtual sensors using AWS IoT services during this assignment. Setting up a secure connection with MQTT using AWS IoT Core was one of the first difficulties I encountered. I had some issues with wrong locations and mismatched certificate files, which at first resulted in connection failures. It seemed like a huge victory when I was able to connect and publish sensor data after troubleshooting and fixing those problems.

I gained a thorough grasp of how AWS manages streaming IoT data by developing the IoT Analytics pipeline, which includes the channel, pipeline, data store, and dataset. It was fascinating to observe how each part contributed to the processing and storing of data as well as how insights could be extracted from the dataset using SQL queries. I felt more assured that the system was functioning flawlessly after testing the data flow with AWS's MQTT client and viewing real-time updates.

Overall, this assignment improved my understanding of cloud-based IoT systems and how various AWS services work together. It was a terrific learning experience that increased both my technical abilities and confidence in creating scalable IoT solutions.

GitHub Repository:

https://github.com/Daivik-Gangappa/aws-iot-mqtt-simulator-2