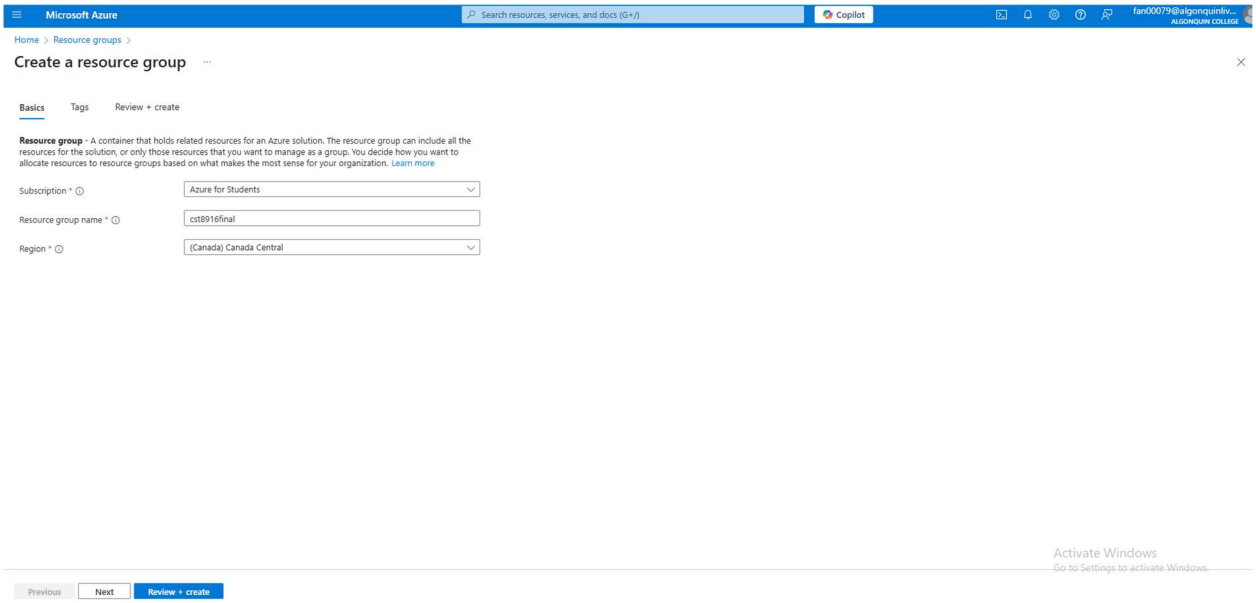
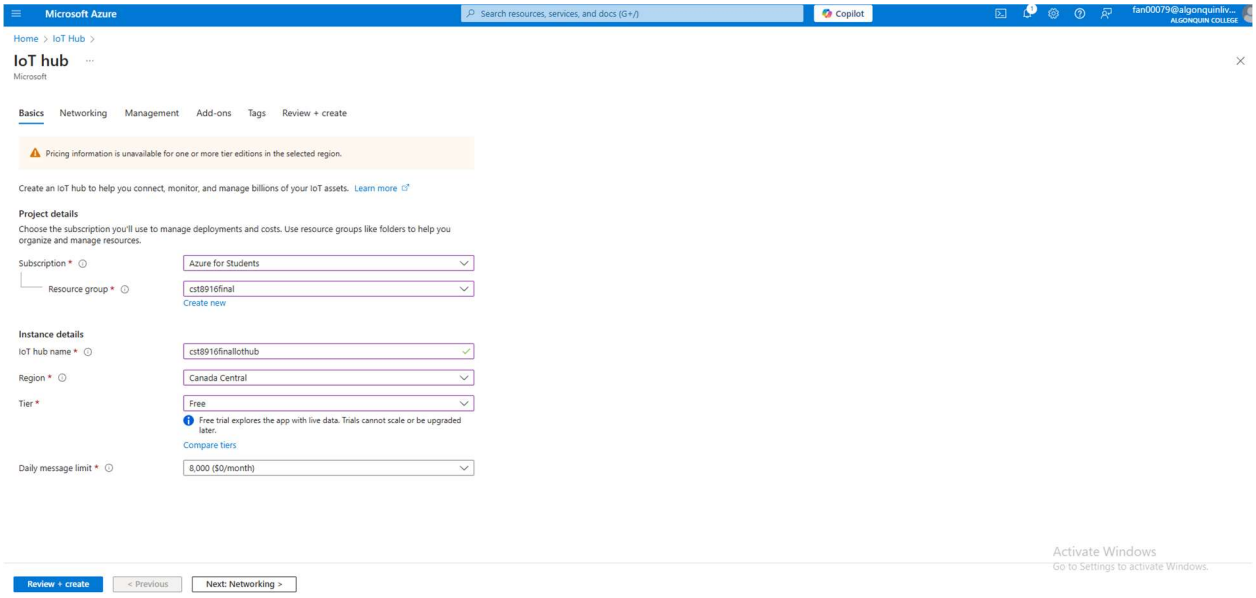
Cst8916final

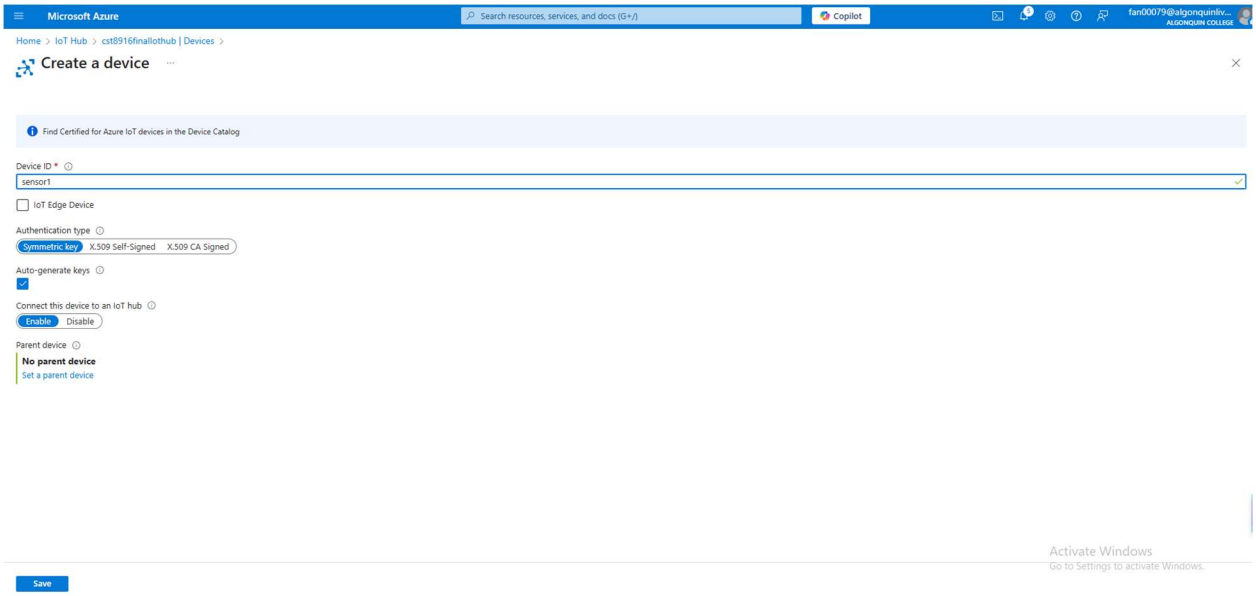
Create resource group

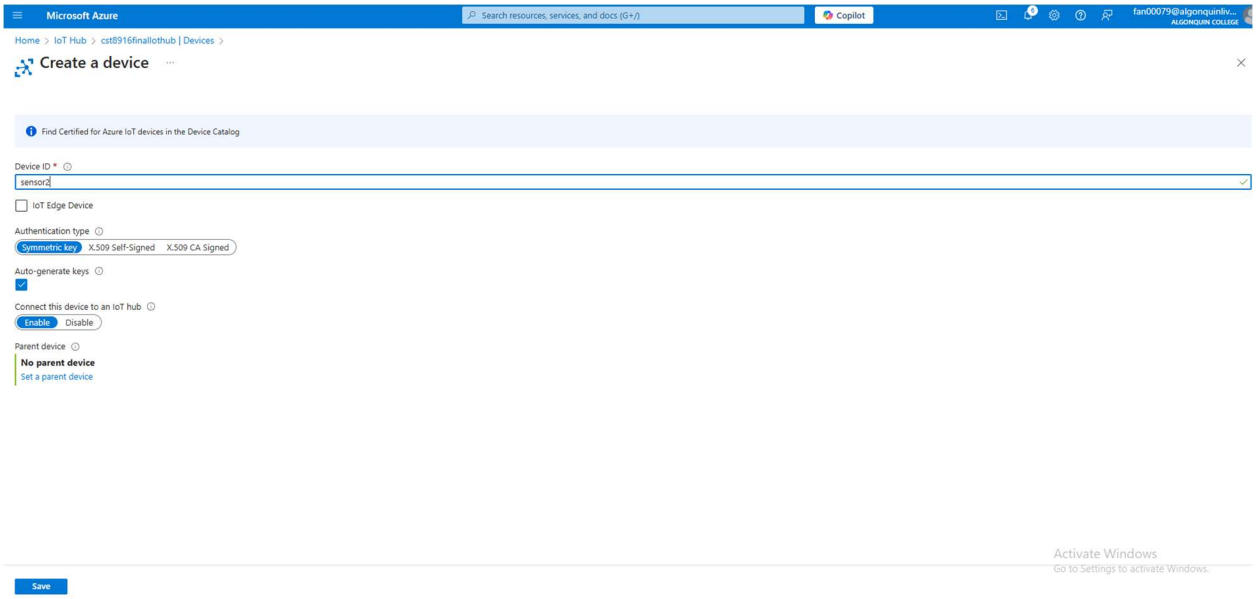


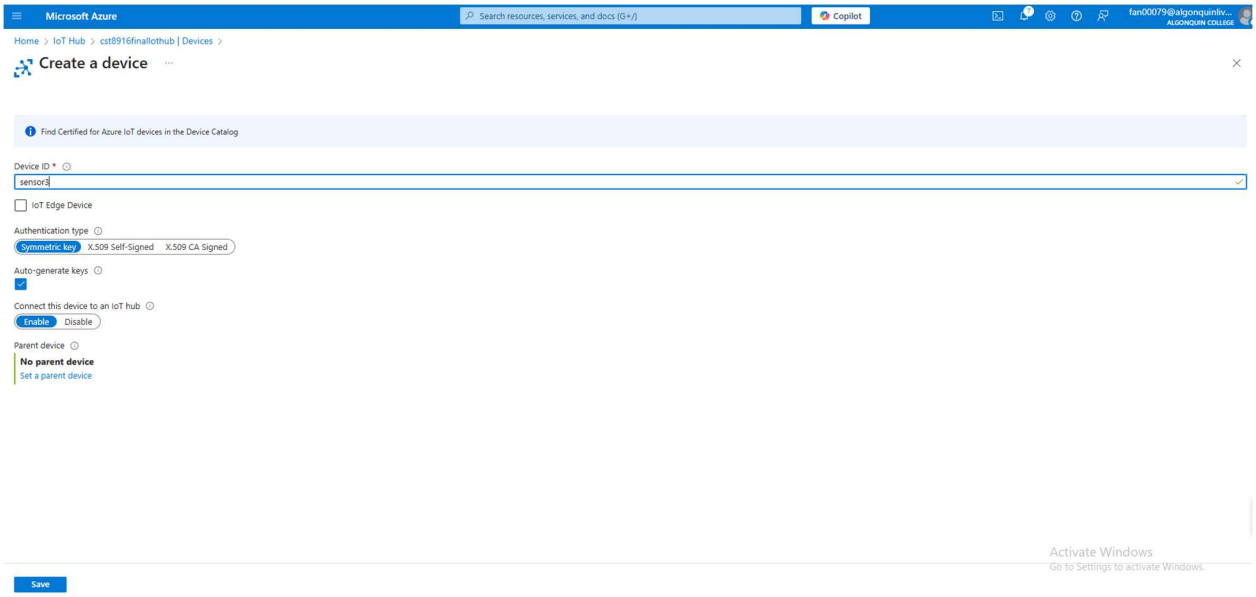
Create lot hub



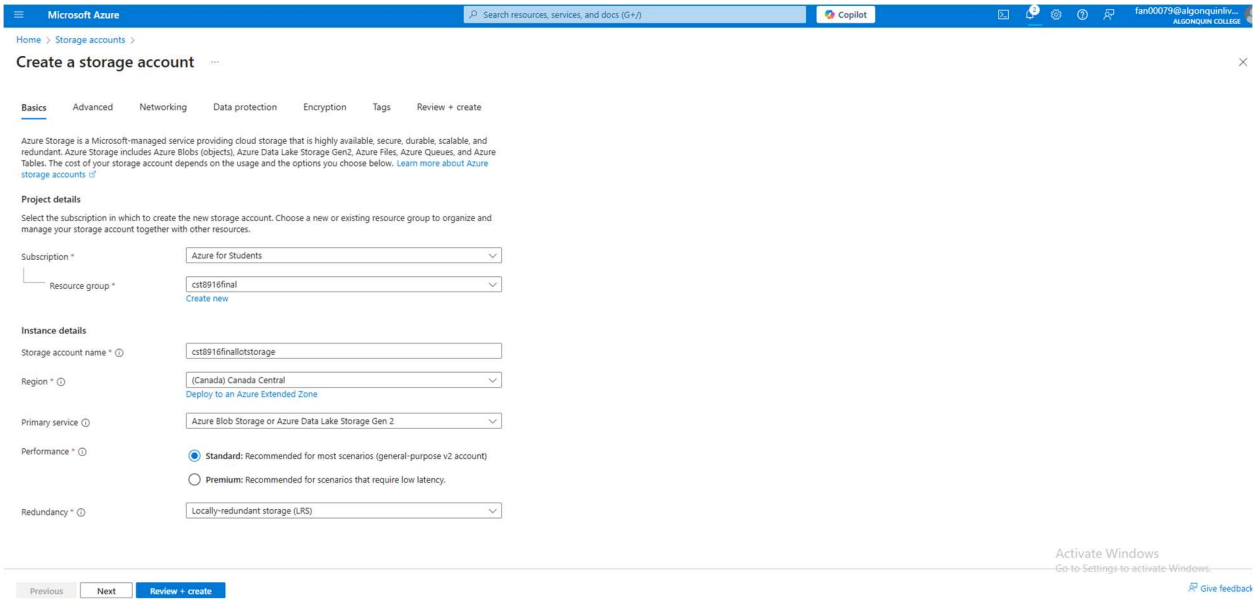
Create device


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
fan00079@algonquinliv...
ALGONQUIN COLLEGE
Home > IoT Hub > cst8916finallothub | Devices >
Create a device
Find Certified for Azure IoT devices in the Device Catalog
Device ID *
sensor1
IoT Edge Device
Authentication type
Symmetric key
X.509 Self-Signed
X.509 CA Signed
Auto-generate keys
Connect this device to an IoT hub
Enable
Disable
Parent device
No parent device
Set a parent device
Activate Windows
Go to Settings to activate Windows.
Save

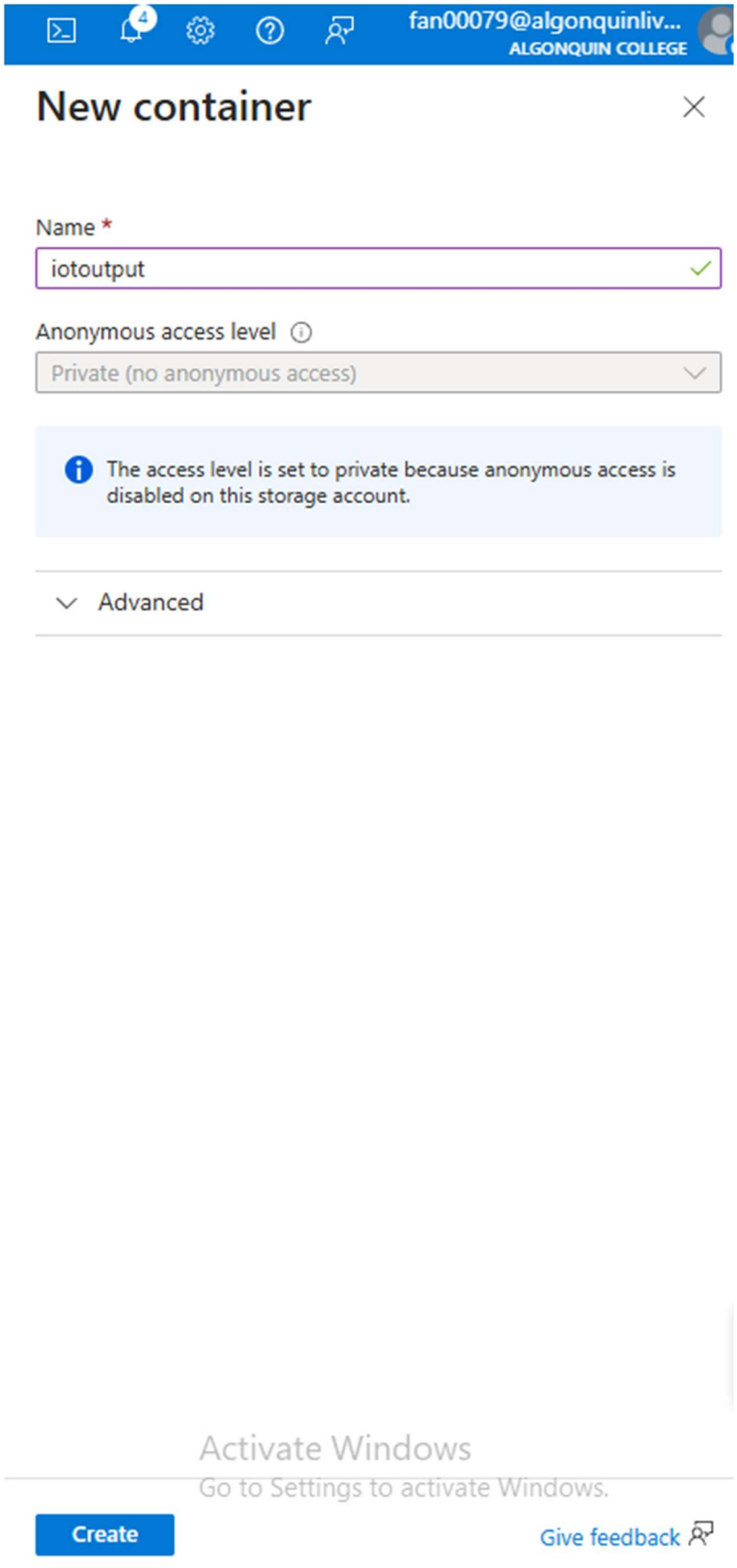

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
fan00079@algonquinliv...
ALGONQUIN COLLEGE
Home > IoT Hub > cst8916finallothub | Devices >
Create a device
Find Certified for Azure IoT devices in the Device Catalog
Device ID *
sensor2
IoT Edge Device
Authentication type
Symmetric key
X.509 Self-Signed
X.509 CA Signed
Auto-generate keys
Connect this device to an IoT hub
Enable
Disable
Parent device
No parent device
Set a parent device
Activate Windows
Go to Settings to activate Windows.
Save



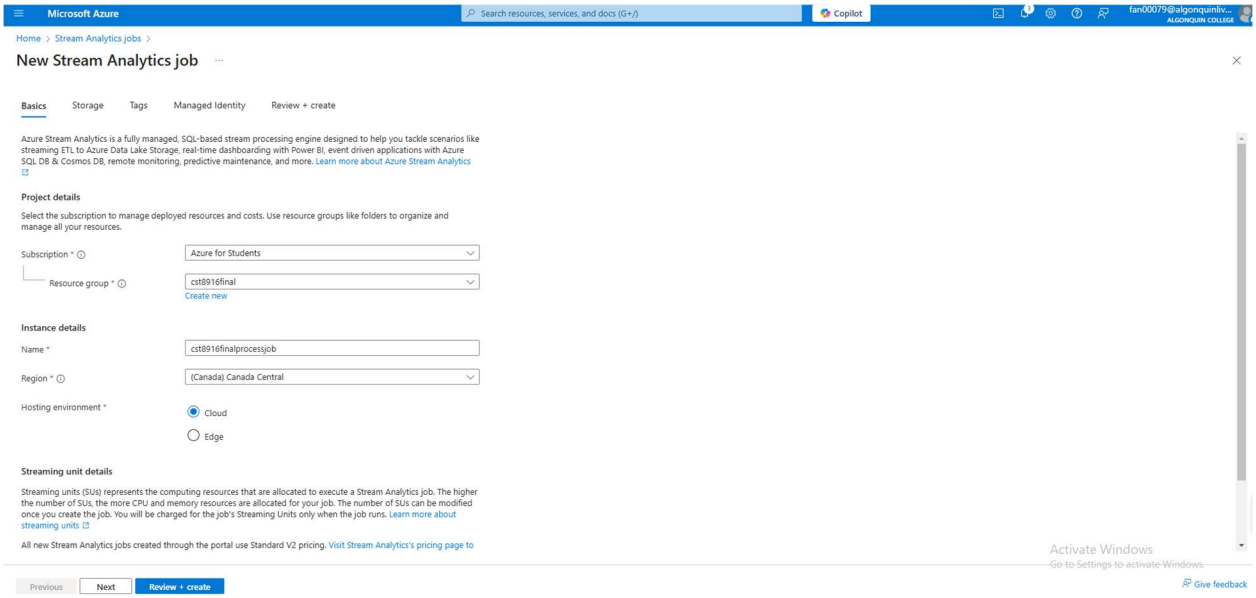
Create storage for lot



Create iot output container


fan00079@algonquinliv...
ALGONQUIN COLLEGE
New container
Name *
iotoutput
Anonymous access level
Private (no anonymous access)
The access level is set to private because anonymous access is disabled on this storage account.
Advanced
Activate Windows
Go to Settings to activate Windows.
Create
Give feedback

Create stream analytics job



Create sensor python code

sensor1.py > ...

```python
import time
import random
from datetime import datetime, timezone
from azure.iot.device import IoTHubDeviceClient, Message

CONNECTION_STRING = "HostName=cst8916finallothub.azure-devices.net;DeviceId=sensor1;SharedAccessKey=2zjeT7Sq0ZTI5OqIkkMGaZzHGQ0A8Mm0Z3YPQm+8dNg="

def get_telemetry():
    return {
        "location": "Dow's Lake",
        "iceThickness": random.randint(20, 40),  # cm
        "surfaceTemperature": round(random.uniform(-10, 0), 1),  # °C
        "snowAccumulation": random.randint(0, 15),  # cm
        "externalTemperature": round(random.uniform(-15, 5), 1),  # °C
        "timestamp": datetime.now(timezone.utc).isoformat()
    }

def main():
    client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)

    print("Sending telemetry to IoT Hub...")
    try:
        while True:
            telemetry = get_telemetry()
            message = Message(str(telemetry))
            client.send_message(message)
            print(f"Sent message: {message}")
            time.sleep(10)
    except KeyboardInterrupt:
        print("Stopped sending messages.")
    finally:
        client.disconnect()

if __name__ == "__main__":
    main()
```

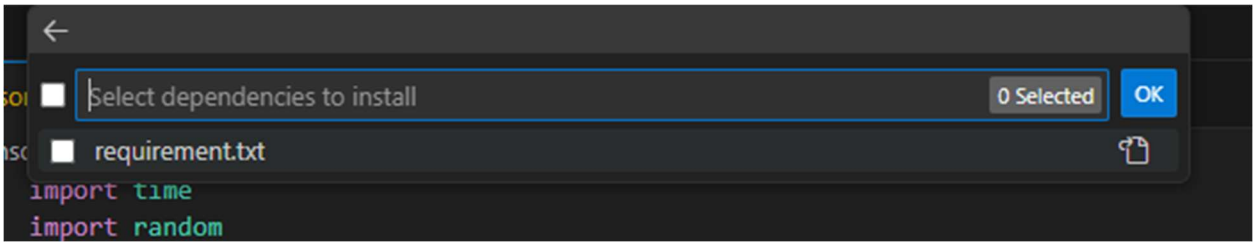
sensor2.py > main

```python
import time
import random
from datetime import datetime, timezone
from azure.iot.device import IoTHubDeviceClient, Message

CONNECTION_STRING = "HostName=cst8916finallothub.azure-devices.net;DeviceId=sensor2;SharedAccessKey=NeQr8fVvhRdEaS+B2NQSzX3QkWq3GbbahZrN8yMTTS8="

def get_telemetry():
    return {
        "location": "Fifth Avenue",
        "iceThickness": random.randint(20, 40),  # cm
        "surfaceTemperature": round(random.uniform(-10, 0), 1),  # °C
        "snowAccumulation": random.randint(0, 15),  # cm
        "externalTemperature": round(random.uniform(-15, 5), 1),  # °C
        "timestamp": datetime.now(timezone.utc).isoformat()
    }

def main():
    client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)

    print("Sending telemetry to IoT Hub...")
    try:
        while True:
            telemetry = get_telemetry()
            message = Message(str(telemetry))
            client.send_message(message)
            print(f"Sent message: {message}")
            time.sleep(10)
    except KeyboardInterrupt:
        print("Stopped sending messages.")
    finally:
        client.disconnect()

if __name__ == "__main__":
    main()
```

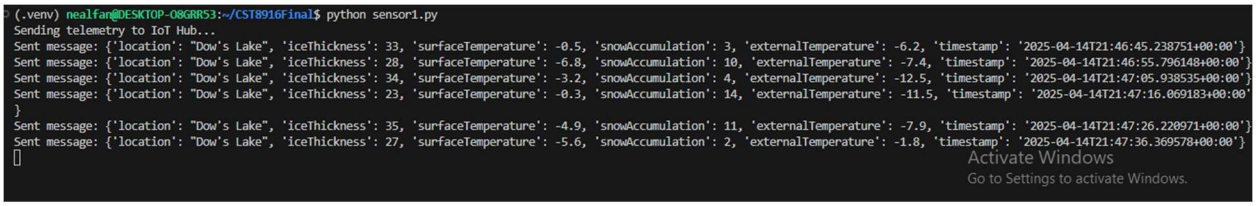
sensor3.py > ...

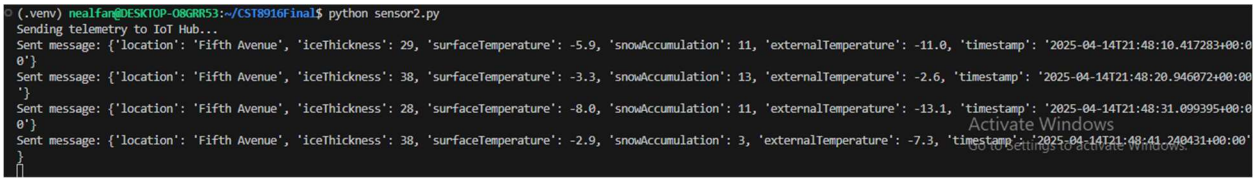
```python
import time
import random
from datetime import datetime, timezone
from azure.iot.device import IoTHubDeviceClient, Message

CONNECTION_STRING = "HostName=cst8916finallothub.azure-devices.net;DeviceId=sensor3;SharedAccessKey=FFjeHcwuPAtkn0XhVgUeJvCv5KbaaKqeT0k+iIev+p8="

def get_telemetry():
    return {
        "location": "NAC",
        "iceThickness": random.randint(20, 40),  # cm
        "surfaceTemperature": round(random.uniform(-10, 0), 1),  # °C
        "snowAccumulation": random.randint(0, 15),  # cm
        "externalTemperature": round(random.uniform(-15, 5), 1),  # °C
        "timestamp": datetime.now(timezone.utc).isoformat()
    }

def main():
    client = IoTHubDeviceClient.create_from_connection_string(CONNECTION_STRING)

    print("Sending telemetry to IoT Hub...")
    try:
        while True:
            telemetry = get_telemetry()
            message = Message(str(telemetry))
            client.send_message(message)
            print(f"Sent message: {message}")
            time.sleep(10)
    except KeyboardInterrupt:
        print("Stopped sending messages.")
    finally:
        client.disconnect()

if __name__ == "__main__":
    main()
```

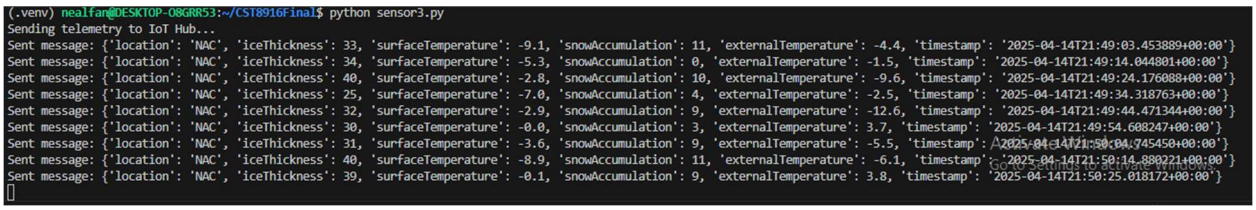
Create Python Environment



Run example output







Create stream analytics job input

IoT Hub

New input

Input alias *

iotinput

Provide IoT Hub settings manually

Select IoT Hub from your subscriptions

Subscription

Azure for Students

IoT Hub *

cst8916finallothub

Consumer group *

$Default

Shared access policy name *

iothubowner

Shared access policy key

••••••••••••••••••••••••••••

Endpoint

Messaging

Partition key

Event serialization format *

JSON

Encoding

UTF-8

Save

Create stream analytics job output

Blob storage/ADLS Gen2

New output

Output alias *

iotoutput

- ( ) Provide Blob storage/ADLS Gen2 settings manually
- (•) Select Blob storage/ADLS Gen2 from your subscriptions

Subscription

Azure for Students

Storage account *

cst8916finallotstorage

Container *

- ( ) Create new
- (•) Use existing

iotoutput

Authentication mode

Connection string

Storage account key

••••••••••••••••••••••••••

Event serialization format *

JSON

Format

Line separated

Encoding

UTF-8

Write mode *

- ( ) Once, when all results for the time partition are available.

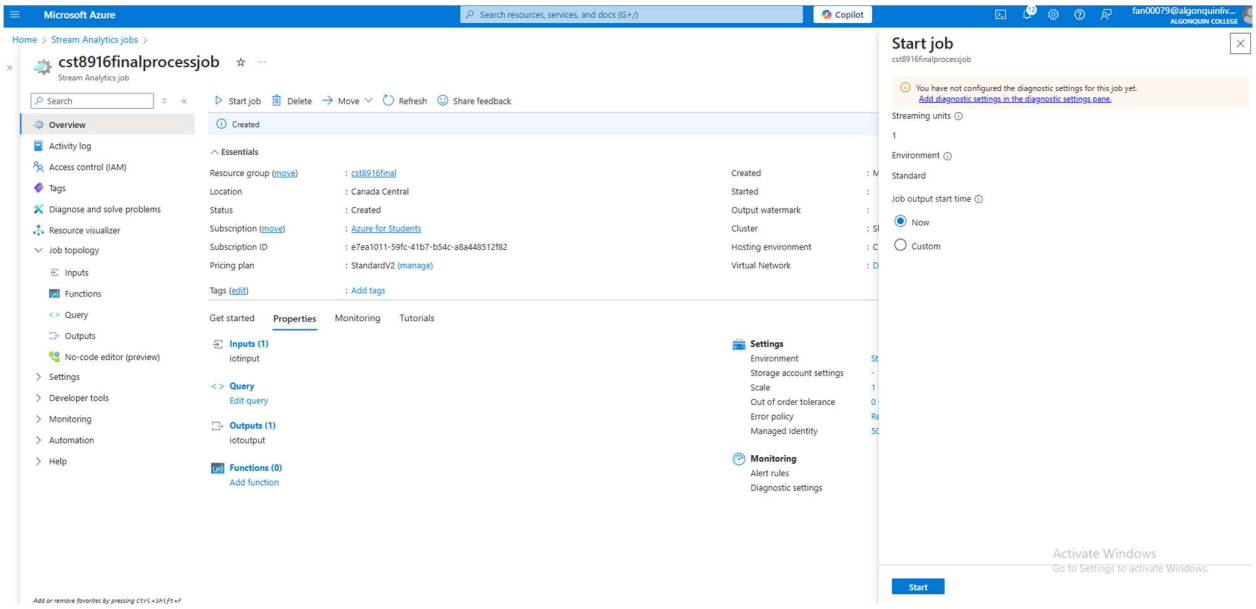
Save

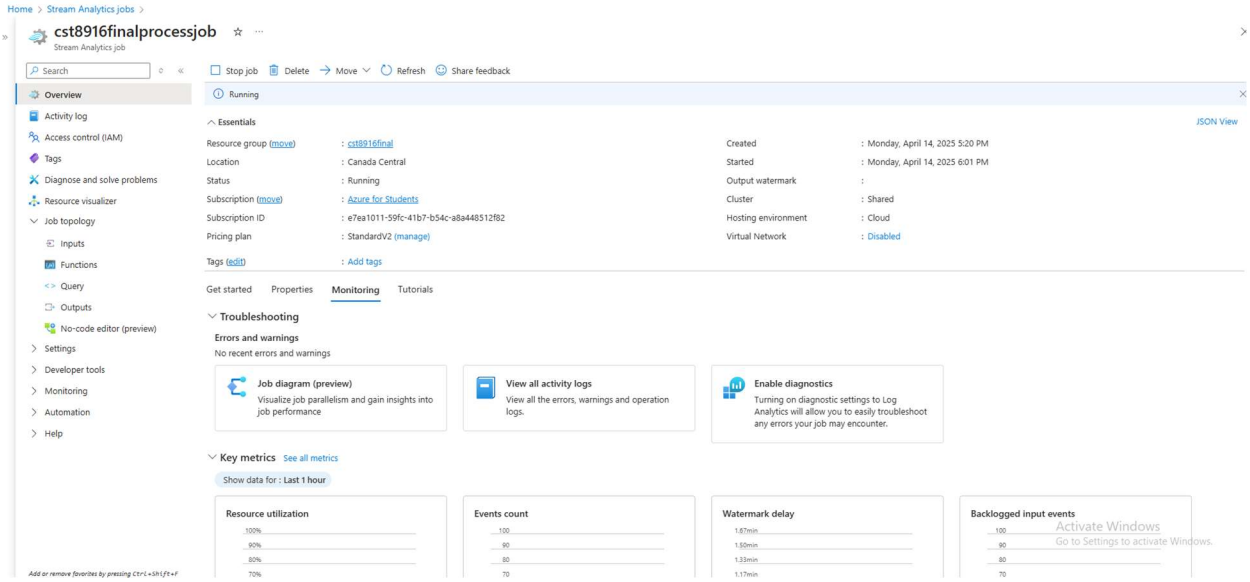
Query

Test query Save query Discard changes

```
SELECT
    location AS Location,
    AVG(CAST(iceThickness AS float)) AS AvgIceThickness,
    MAX(CAST(snowAccumulation AS float)) AS MaxSnowAccumulation,
    System.Timestamp AS EventTime
INTO
    [iotoutput]
FROM
    [iotinput]
GROUP BY
    location, TumblingWindow(minute, 5)
```

Start stream analytics job




Home > Stream Analytics jobs >
cst8916finalprocessjob
Stream Analytics job
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Job topology
Inputs
Functions
Query
Outputs
No-code editor (preview)
Settings
Developer tools
Monitoring
Automation
Help
Add or remove favorites by pressing Ctrl+Shift+F
Stop job
Delete
Move
Refresh
Share feedback
Running
Essentials
JSON View
Resource group (move) : cst8916final
Location : Canada Central
Status : Running
Subscription (move) : Azure for Students
Subscription ID : e7ea1011-59fc-41b7-b54c-a8a448512f82
Pricing plan : StandardV2 (manage)
Tags (edit) : Add tags
Created : Monday, April 14, 2025 5:20 PM
Started : Monday, April 14, 2025 6:01 PM
Output watermark :
Cluster : Shared
Hosting environment : Cloud
Virtual Network : Disabled
Get started
Properties
Monitoring
Tutorials
Troubleshooting
Errors and warnings
No recent errors and warnings
Job diagram (preview)
Visualize job parallelism and gain insights into job performance
View all activity logs
View all the errors, warnings and operation logs.
Enable diagnostics
Turning on diagnostic settings to Log Analytics will allow you to easily troubleshoot any errors your job may encounter.
Key metrics
See all metrics
Show data for : Last 1 hour
Resource utilization
100%
90%
80%
70%
Events count
100
90
80
70
Watermark delay
1.67min
1.50min
1.33min
1.17min
Backlogged input events
100
90
80
70
Activate Windows
Go to Settings to activate Windows.