Lab 5: Connecting IoT to the Cloud with Microsoft Azure

Group Members:

- Member 1: Khalil Ahmad
- Member 2: Apala Pramanik
- Member 3: Audrey Vazzana
- Member 4: Raoul

Table of Contents

1. Introduction

In this lab, we utilized Azure IoT Central to simulate a cloud-based IoT application. Azure IoT Central is a FaaS that simplifies device management and telemetry data collection. Our goal was to create an IoT Central application, configure virtual devices to send telemetry data, and visualize this data on a custom dashboard. Each group member created and deployed their own virtual device to the cloud.

2. Development Process

In this section, we document the steps taken during the development of our IoT Central integration and highlight the challenges we faced along the way.

1. Project Planning and Initial Setup

The initial phase of development involved researching the Azure IoT Central platform and setting up the environment. The main goals were:

- Set up Azure IoT Central.
- Create a virtual device to simulate telemetry data transmission.
- Develop Python scripts for device-to-cloud communication.

Resources that helped during this phase:

- Azure IoT Central Documentation: Provided guidance on how to configure the IoT Central application and set up device credentials.
- Open-source projects and Python IoT libraries: These were referenced for implementing device command handling and telemetry data transmission.

2. Coding and Device Implementation

Each team member created a Python script to simulate a virtual device using the `iotc` Python library. Below were the key steps:

- **IoT Central connection**: We used the scope ID, device ID, and device key from our IoT Central application to connect each device.
- **Command Handling**: The devices were designed to receive a `SendData` command from IoT Central. When the command was received, telemetry data for temperature, wind speed, and humidity was generated and sent to the cloud.
- **Telemetry Data**: In addition to command-based data transmission, devices sent telemetry data at regular intervals (every 60 seconds).

3. Challenges and Troubleshooting

Several challenges arose during development, and the following troubleshooting techniques were used:

- **IoT Central Connection Issues**: At times, devices failed to connect due to incorrect credentials. The error was traced back to a misconfiguration in the device key or scope ID. Double-checking credentials resolved this issue.
- **Telemetry Data Format**: Initially, telemetry data was not displayed properly on the IoT Central dashboard. We discovered that the data format being sent was incorrect, and converting numerical data to strings resolved the problem.
- **Command Execution Failures**: While testing command execution, some devices failed to respond to the `SendData` command. This was resolved by ensuring that the `on_commands`

function was properly registered and called upon receiving commands.

4. Testing and Validation

After resolving the major issues, the devices were tested thoroughly to ensure they could:

- Connect to IoT Central.
- Respond to commands.
- Send telemetry data at regular intervals.

We used the IoT Central dashboard to monitor the devices in real-time and validate the data being received.

5. Lessons Learned

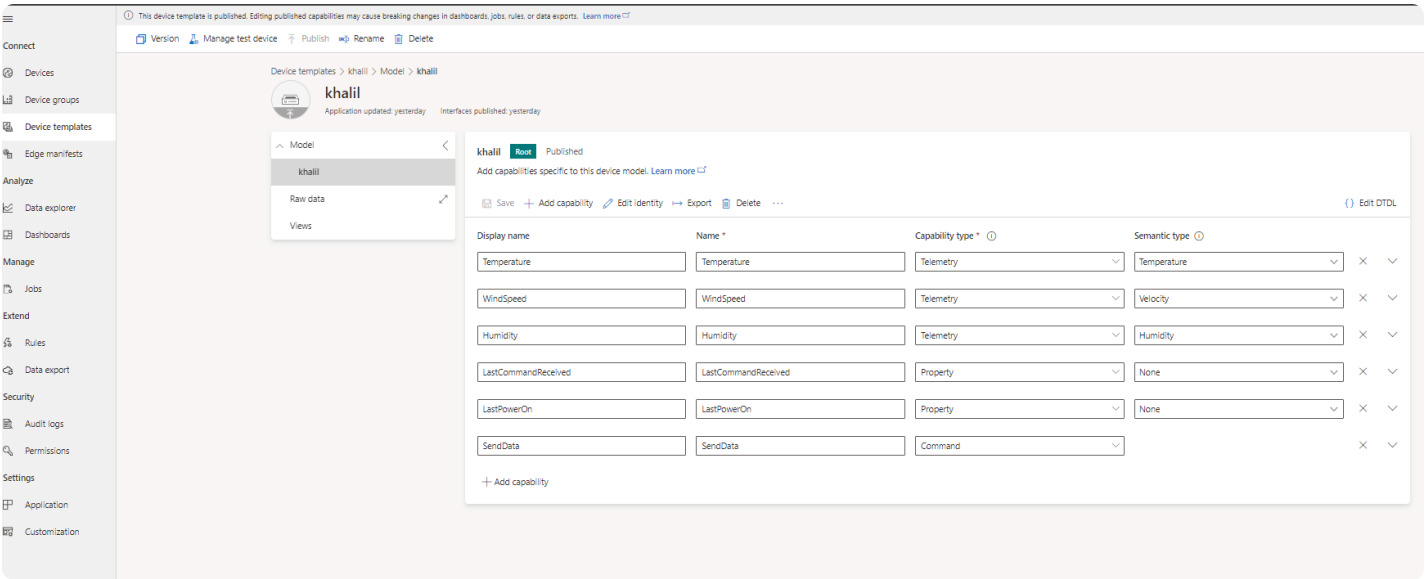
Through the development process, we gained valuable insights into the complexities of IoT integration. Azure IoT Central simplifies many tasks, but ensuring accurate device communication and telemetry data requires careful attention to detail.

3. Device Configuration

While the rest of the code remains same, the IoT central credentials are different for each member. So, we only change the following 3 lines of code and replace the credentials with our own.

Images of device configurations here

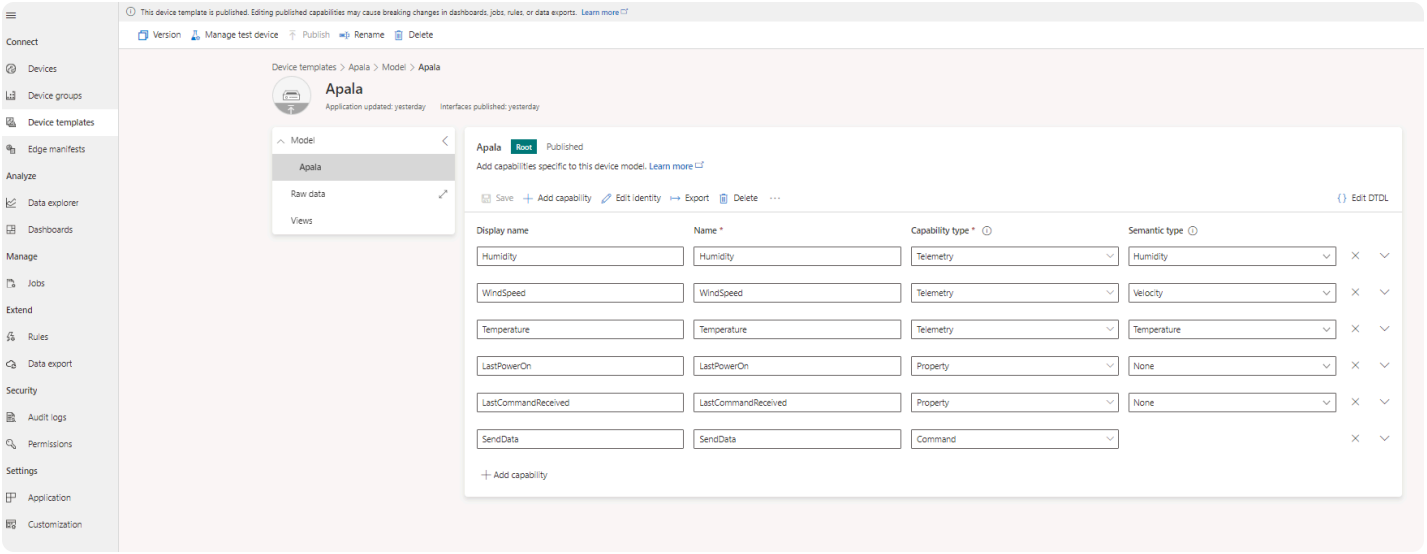
Khalil device configuration on Azure IOT Central



Khalil IOT Central credentials

```
# Replace these with your IoT Central credentials
scopeId = '0ne00CE8964'  # From lab 5.txt
device_id = '26znol6fzh7'  # From lab 5.txt
device_key = 'qoCaBmM9hRi7QJhw4HKIM9t68Od55cv2VhC5/NqNzWE='  # From lab 5.txt

```

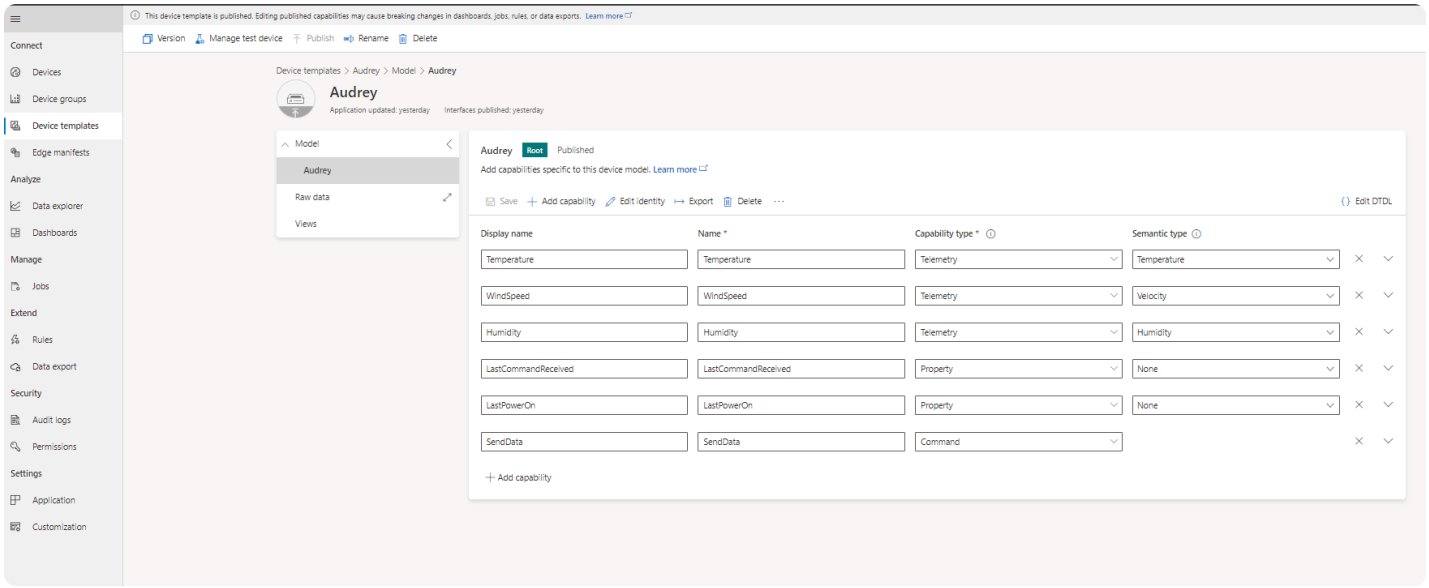
Apala device configuration on Azure IOT Central



Apala IOT Central credentials

```
# Replace these with your IoT Central credentials
scopeId = '0ne00CE8964'  # From lab 5.txt
device_id = '2mt69me4w6q'  # From lab 5.txt
device_key = '9PRKb8z6zBhPddc3X/iiVxo8J266+yvJpPZ1LRNTp2g='  # From lab 5.txt

```

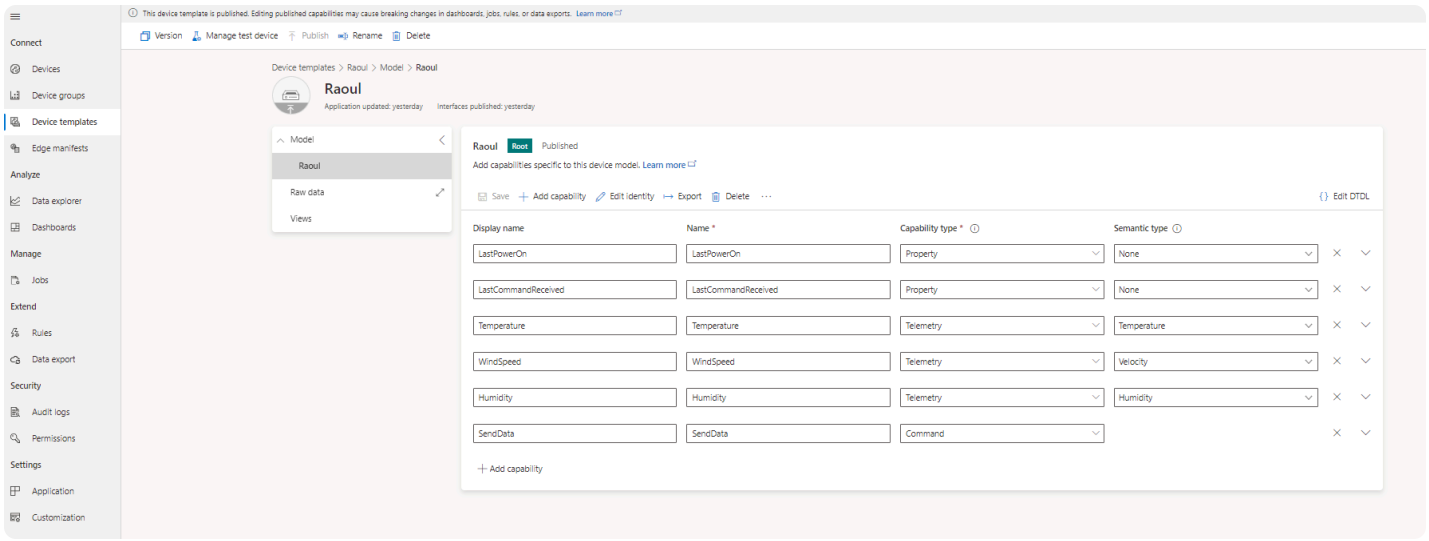
Audrey device configuration on Azure IOT Central



Audrey IOT Central credentials

```
# Replace these with your IoT Central credentials
scopeId = '0ne00CE8964'  # From lab 5.txt
device_id = 'rb20scn64p'  # From lab 5.txt
device_key = 'a+4I6iBCu9fonE08b2PAf6Ej+YgYeFI2G+rFnbwe1jE='  # From lab 5.txt
```

Raoul device configuration on Azure IOT Central



Raoul IOT Central credentials

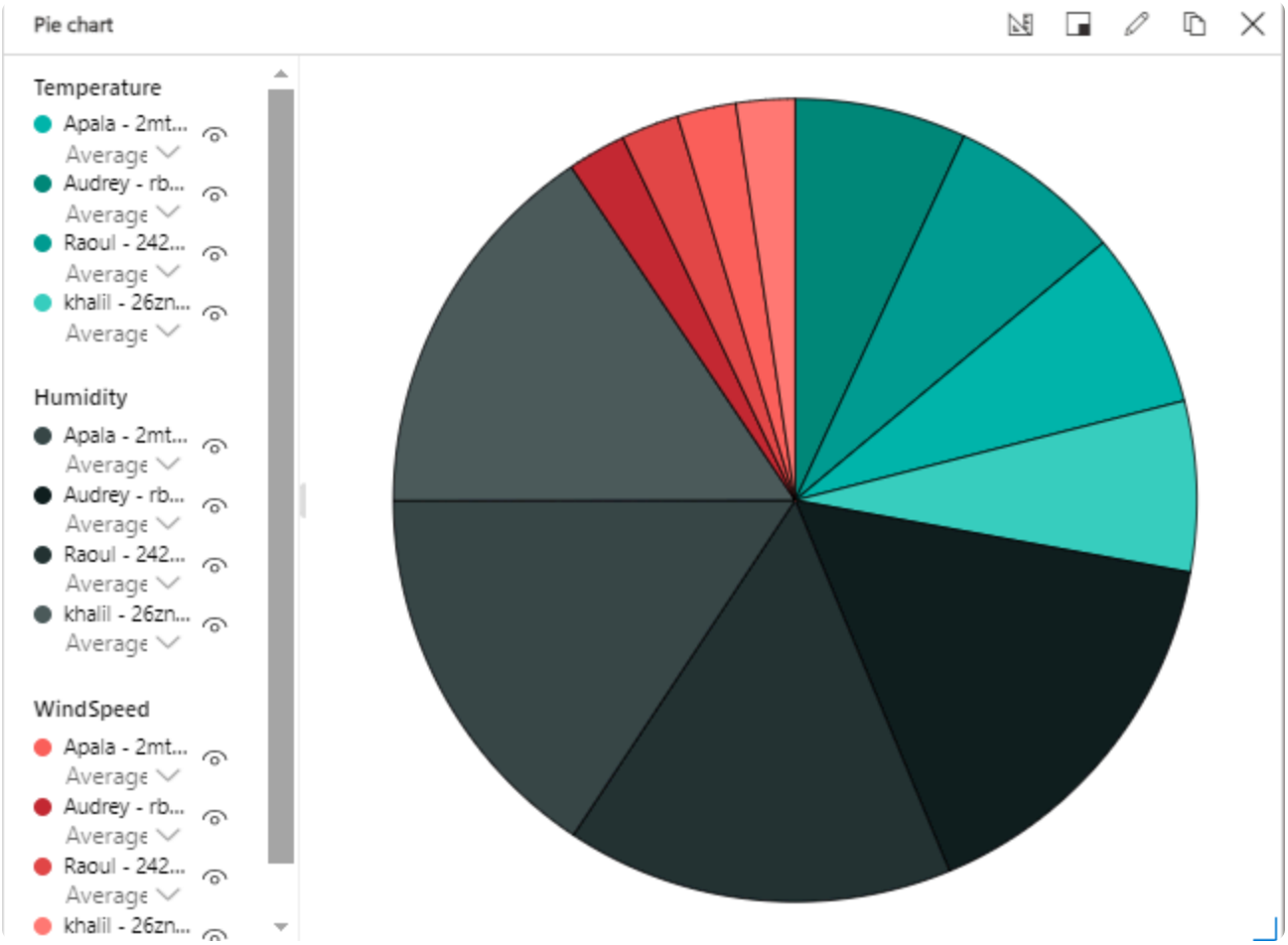
```
# Replace these with your IoT Central credentials
scopeId = '0ne00CE8964'  # From lab 5.txt
device_id = '242baqqe14v'  # From lab 5.txt
device_key = 'ntBkjFjjNA35KlewRc8aurAf7625u3xvimsiO8InBC0='  # From lab 5.txt
```

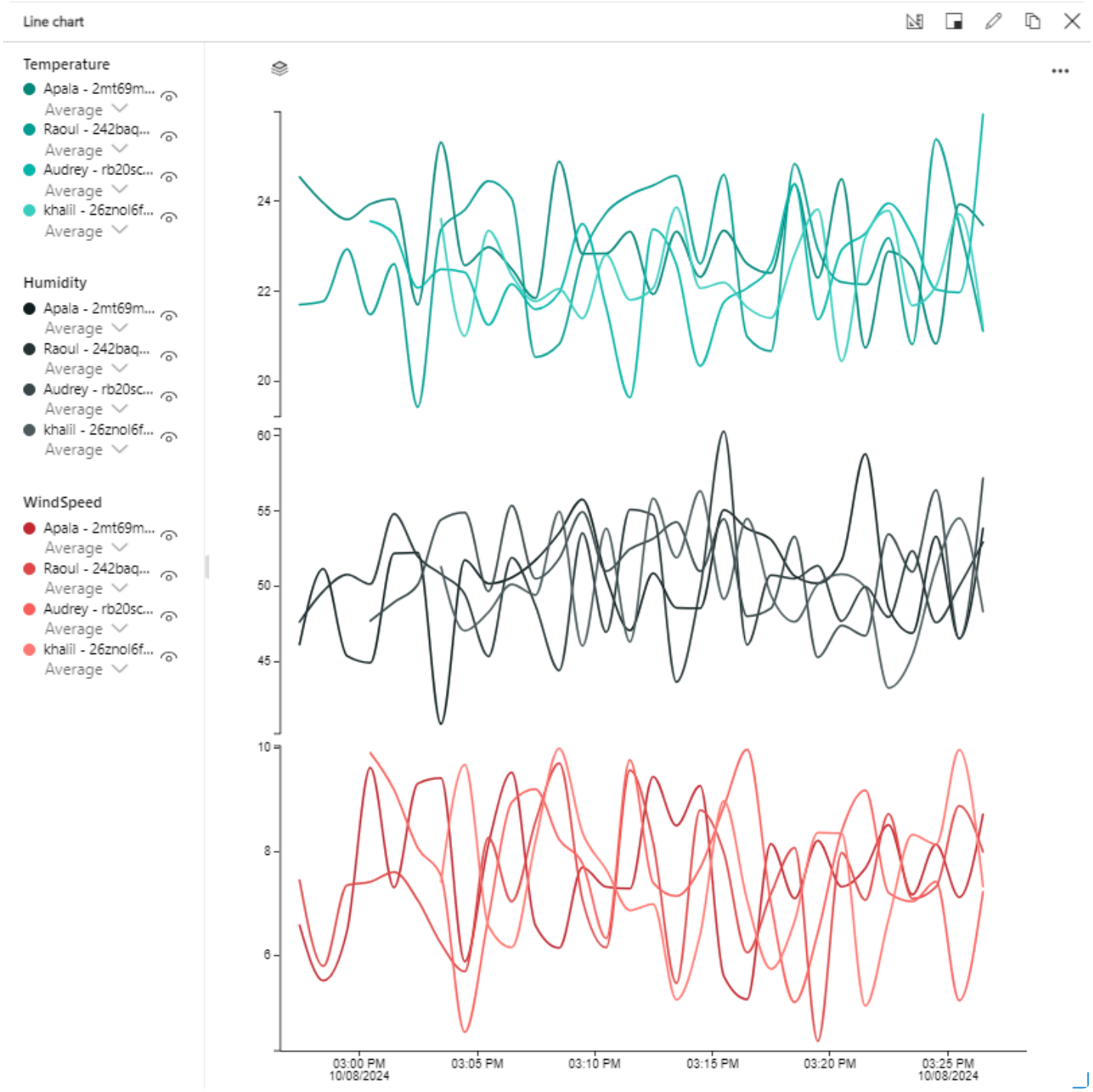
4. Dashboard Creation

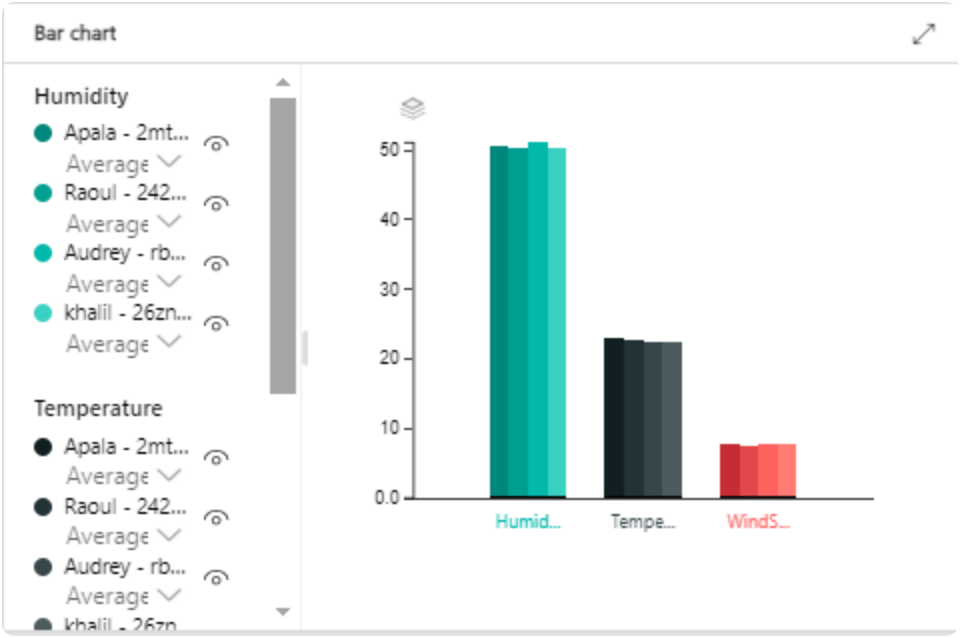
We created a custom dashboard to visualize the telemetry data from all devices. The following charts were included:

1. **Pie Chart**: Visualizes temperature, humidity and windspeed readings from all 4 devices as a pie chart.
2. **Line Chart**: Visualizes temperature, humidity and windspeed readings from all 4 devices as a line chart.
3. **Bar Chart**: Visualizes temperature, humidity and windspeed readings from all 4 devices as a bar chart.

Dashboard screenshots here


Pie chart
Temperature
Apala - 2mt...
Average
Audrey - rb...
Average
Raoul - 242...
Average
khalil - 26zn...
Average
Humidity
Apala - 2mt...
Average
Audrey - rb...
Average
Raoul - 242...
Average
khalil - 26zn...
Average
WindSpeed
Apala - 2mt...
Average
Audrey - rb...
Average
Raoul - 242...
Average
khalil - 26zn...


Line chart
Temperature
Apala - 2mt69m...
Average
Raoul - 242baq...
Average
Audrey - rb20sc...
Average
khalil - 26znol6f...
Average
Humidity
Apala - 2mt69m...
Average
Raoul - 242baq...
Average
Audrey - rb20sc...
Average
khalil - 26znol6f...
Average
WindSpeed
Apala - 2mt69m...
Average
Raoul - 242baq...
Average
Audrey - rb20sc...
Average
khalil - 26znol6f...
Average
24
22
20
60
55
50
45
10
8
6
03:00 PM
10/08/2024
03:05 PM
03:10 PM
03:15 PM
03:20 PM
03:25 PM
10/08/2024



5. Virtual Device Implementation

Each group member implemented their virtual device using the `iotc` Python library. Below is the code for one of the devices:

1. IoT Central Credentials

While the rest of the code remains same, the IoT central credentials are different for each member. So, we only change the following 3 lines of code and replace the credentials with our own. The scope ID, device ID, and device key are set with values obtained from a configuration file (`lab 5.txt`). These credentials allow the device to securely connect to IoT Central.

```
scopeId = '0ne00CE8964'  # From lab 5.txt
device_id = '26znol6fzh7'  # From lab 5.txt
device_key = 'qoCaBmM9hRi7QJhw4HKIM9t68Od55cv2VhC5/NqNzWE='  # From lab 5.txt
```

2. Command Handling Function

This function handles incoming commands sent from IoT Central. If the `SendData` command is received, the device generates and sends telemetry data (temperature, wind speed, and humidity) and updates a property (`LastCommandReceived`).

```
def on_commands(command: Command):
    print(f"Command received: {command.name}")

    if command.name == "SendData":
        print("Sending data as per the SendData command")
        iotc.send_telemetry({
            'Temperature': str(random.uniform(15.0, 30.0)),
            'WindSpeed': str(random.uniform(0.0, 15.0)),
            'Humidity': str(random.uniform(30.0, 70.0))
        })
        iotc.send_property({
            "LastCommandReceived": time.time()
        })
        print("Telemetry and LastCommandReceived property sent")

    command.reply()
```

3. IoT Central Client Initialization

The IoT Central client is initialized with the device credentials and the connection type is set to use a device key. This establishes the device's connection to IoT Central.

```
iotc = IoTCClient(
    device_id,
    scopeId,
    IOTCConnectType.IOTC_CONNECT_DEVICE_KEY,
    device_key
)

iotc.connect()
```

4. Setting Up Command Listener

Once the connection is established, a listener is set up to handle incoming commands. The `on_commands` function is passed to the command event handler.

```
iotc.on(IOTCEvents.IOTC_COMMAND, on_commands)
```

5. Sending Device Properties

Upon powering up the device, it sends an initial property (`LastPowerOn`) to IoT Central, which contains the timestamp of when the device was last powered on.

```
iotc.send_property({
    "LastPowerOn": time.time()
})

print("Device connected and LastPowerOn property sent")
```

6. Telemetry Data Generation and Transmission

In a continuous loop, telemetry data for temperature, wind speed, and humidity is randomly generated and sent to IoT Central every 60 seconds. The `is_connected()` method ensures that the loop runs only when the device is connected.

```
while iotc.is_connected():
    temperature = random.uniform(15.0, 30.0)
    wind_speed = random.uniform(0.0, 15.0)
    humidity = random.uniform(30.0, 70.0)

    iotc.send_telemetry({
        'Temperature': str(temperature),
        'WindSpeed': str(wind_speed),
        'Humidity': str(humidity)
    })

    print(f"Telemetry sent - Temperature: {temperature}, Wind Speed: {wind_speed},

    time.sleep(60)
```

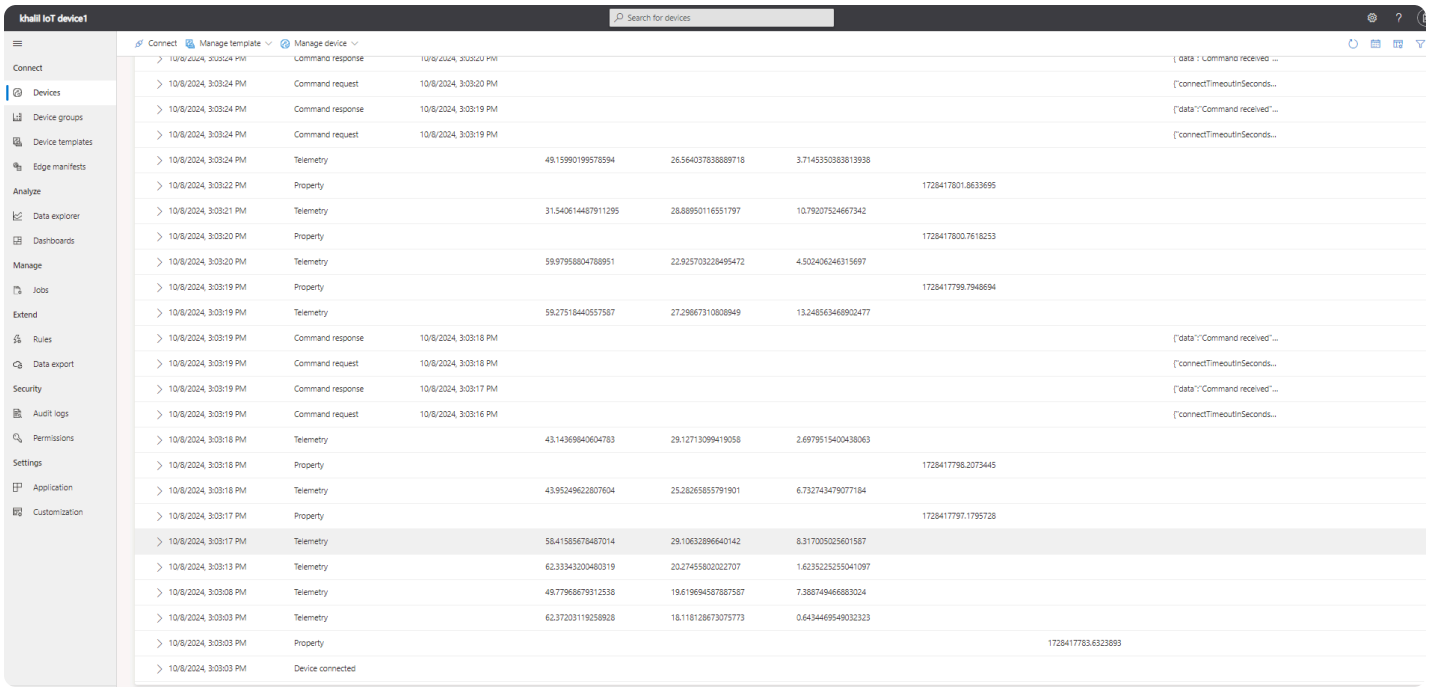
6. Results

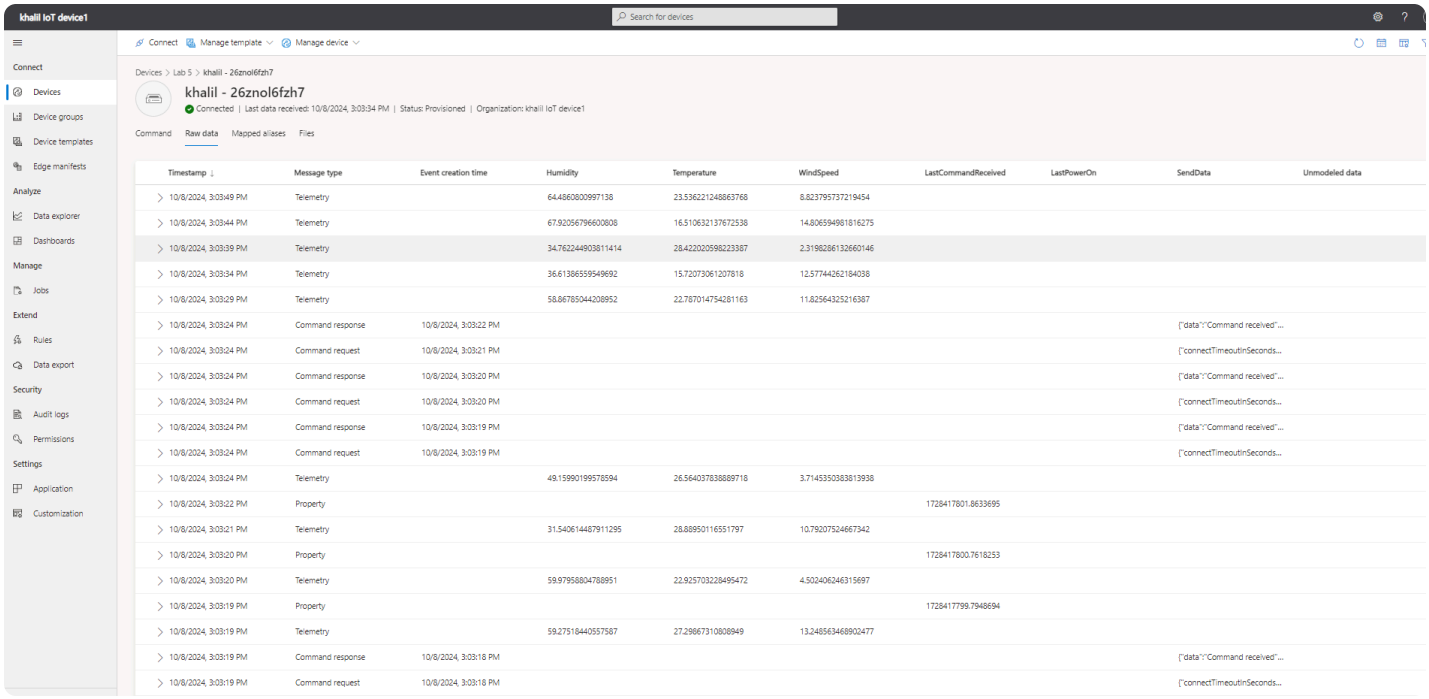
- All devices successfully sent telemetry data to IoT Central.
- The commands from the IoT Central were executed on the virtual devices.
- We monitored the data in real-time using the dashboard, observing telemetry values and device status.

Images of data on Azure IoT Central here

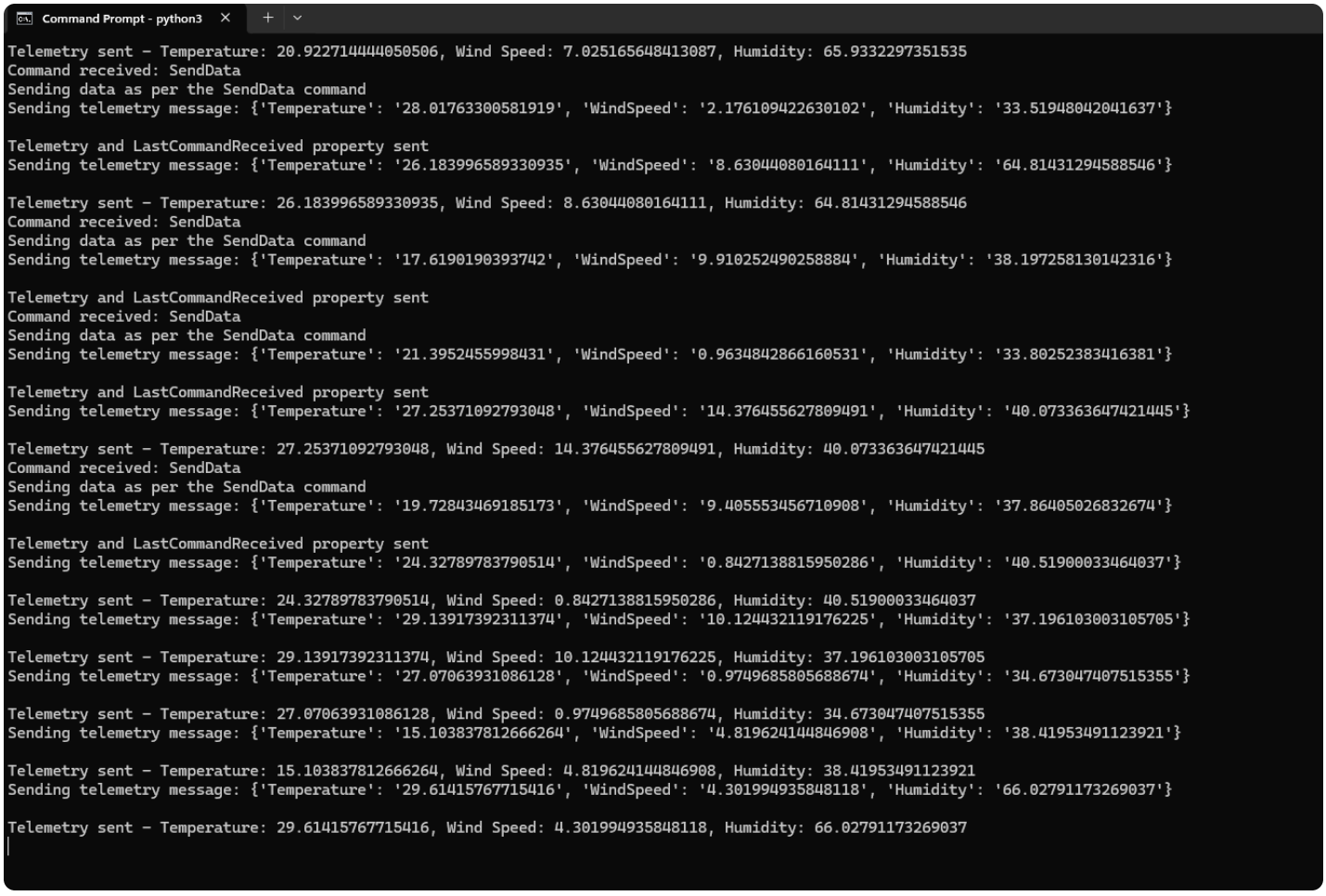
Khalil's Data

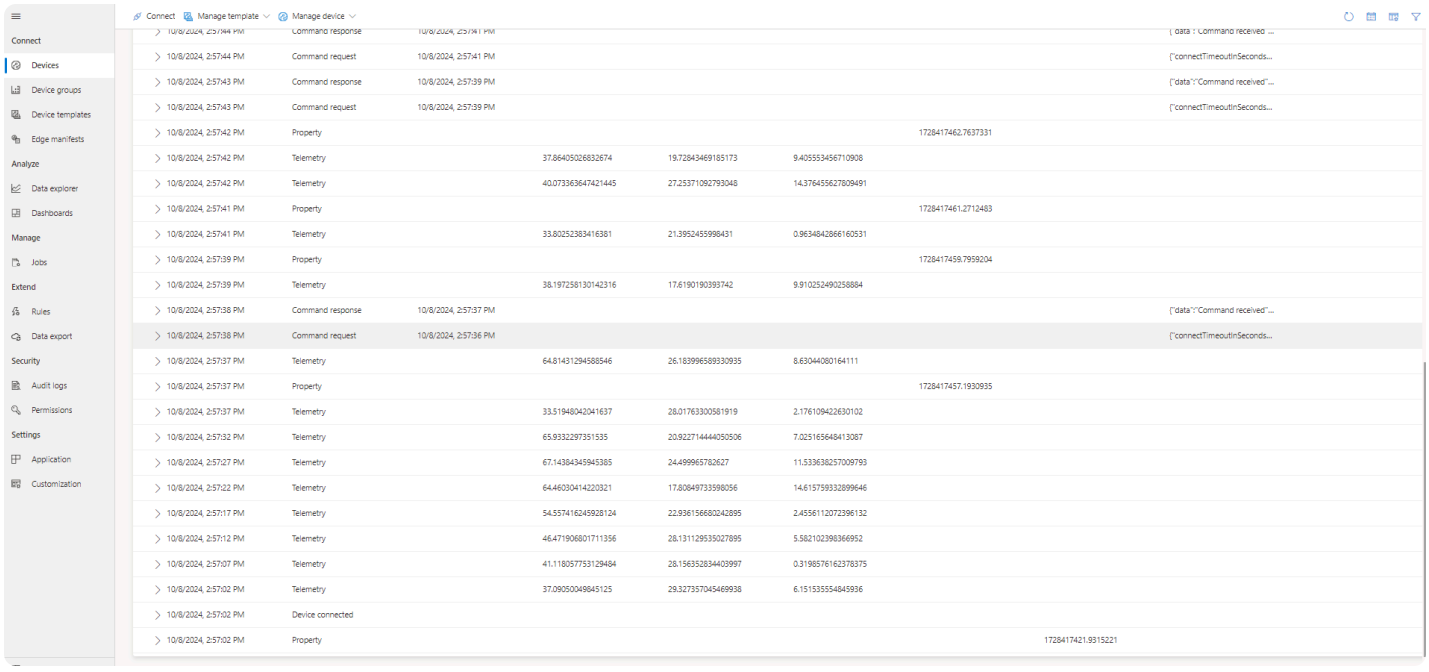
```
PS C:\Users\kahmad2\desktop\lab5>
PS C:\Users\kahmad2\desktop\lab5>
PS C:\Users\kahmad2\desktop\lab5>
PS C:\Users\kahmad2\desktop\lab5>
PS C:\Users\kahmad2\desktop\lab5> python3 khalil.py
Syncing property '$version'
Device connected and LastPowerOn property sent
Sending telemetry message: {'Temperature': '18.118128673075773', 'WindSpeed': '0.6434469549032323', 'Humidity': '62.37203119258928'}

Telemetry sent - Temperature: 18.118128673075773, Wind Speed: 0.6434469549032323, Humidity: 62.37203119258928
Sending telemetry message: {'Temperature': '19.619694587887587', 'WindSpeed': '7.388749466883024', 'Humidity': '49.77968679312538'}

Telemetry sent - Temperature: 19.619694587887587, Wind Speed: 7.388749466883024, Humidity: 49.77968679312538
Sending telemetry message: {'Temperature': '20.27455802022707', 'WindSpeed': '1.6235225255041097', 'Humidity': '62.33343200480319'}

Telemetry sent - Temperature: 20.27455802022707, Wind Speed: 1.6235225255041097, Humidity: 62.33343200480319
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '29.106328966401 42', 'WindSpeed': '8.317005025601587', 'Humidity': '58.41585678487014'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '25.28265855791901', 'WindSpeed': '6.7327434790771 84', 'Humidity': '43.95249622807604'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '29.127130994190 58', 'WindSpeed': '2.6979515400438063', 'Humidity': '43.14369840604783'}

Telemetry sent - Temperature: 29.12713099419058, Wind Speed: 2.6979515400438063, Humidity: 43.14369840604783
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '27.29867310808949', 'WindSpeed': '13.2485634689024 77', 'Humidity': '59.27518440557587'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '22.925703228495472', 'WindSpeed': '4.5024062463156 97', 'Humidity': '59.979588047889 51'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '28.889501165517 97', 'WindSpeed': '10.792075246673 42', 'Humidity': '31.5406144879112 95'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '26.564037838889718', 'WindSpeed': '3.7145350383813938', 'Humidity': '49.15990199578594'}

Telemetry sent - Temperature: 26.564037838889718, Wind Speed: 3.7145350383813938, Humidity: 49.15990199578594
Sending telemetry message: {'Temperature': '22.787014754281163', 'WindSpeed': '11.825643252163 87', 'Humidity': '58.867850442089 52'}

Telemetry sent - Temperature: 22.787014754281163, Wind Speed: 11.825643252163 87, Humidity: 58.867850442089 52
```

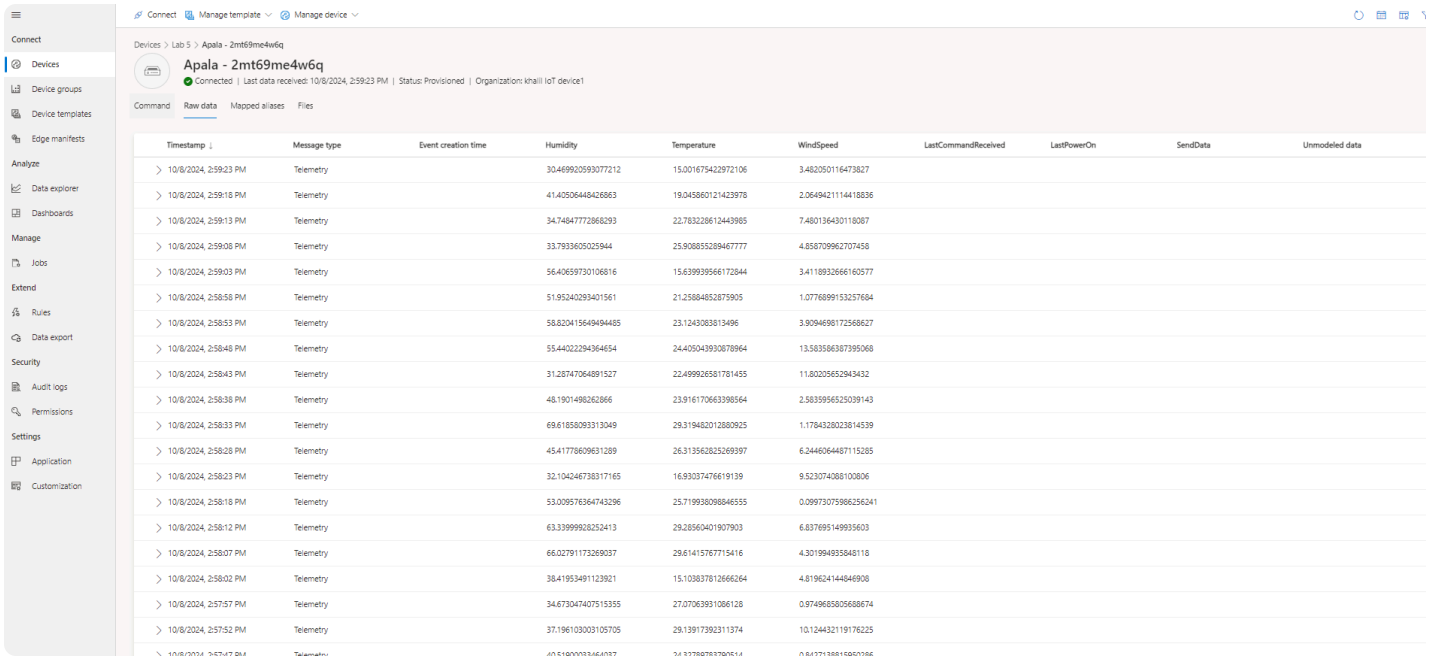



khalil IoT device1
Search for devices
Connect
Manage template
Manage device
Devices > Lab 5 > khalil - 26znol6fzh7
khalil - 26znol6fzh7
Connected | Last data received: 10/8/2024, 3:03:34 PM | Status: Provisioned | Organization: khalil IoT device1
Command
Raw data
Mapped aliases
Files
Connect
Devices
Device groups
Device templates
Edge manifests
Analyze
Data explorer
Dashboards
Manage
Jobs
Extend
Rules
Data export
Security
Audit logs
Permissions
Settings
Application
Customization
Timestamp
Message type
Event creation time
Humidity
Temperature
WindSpeed
LastCommandReceived
LastPowerOn
SendData
Unmodeled data
10/8/2024, 3:03:49 PM Telemetry 64.486080997138 23.536221248863768 8.823795737219454
10/8/2024, 3:03:44 PM Telemetry 67.92056796600808 16.510632137672538 14.806594981816275
10/8/2024, 3:03:39 PM Telemetry 34.762244903811414 28.422020598223387 2.3196286132660146
10/8/2024, 3:03:34 PM Telemetry 36.61386559549692 15.720730612078 18 12.57744262184038
10/8/2024, 3:03:29 PM Telemetry 58.86785044208952 22.787014754281163 11.82564325216387
10/8/2024, 3:03:24 PM Command response 10/8/2024, 3:03:22 PM {"data":"Command received"...
10/8/2024, 3:03:24 PM Command request 10/8/2024, 3:03:21 PM {"connectTimeoutInSeconds...
10/8/2024, 3:03:24 PM Command response 10/8/2024, 3:03:20 PM {"data":"Command received"...
10/8/2024, 3:03:24 PM Command request 10/8/2024, 3:03:20 PM {"connectTimeoutInSeconds...
10/8/2024, 3:03:24 PM Command response 10/8/2024, 3:03:19 PM {"data":"Command received"...
10/8/2024, 3:03:24 PM Command request 10/8/2024, 3:03:19 PM {"connectTimeoutInSeconds...
10/8/2024, 3:03:24 PM Telemetry 49.15990199578594 26.564037838689718 3.7145350383813938
10/8/2024, 3:03:22 PM Property 1728417801.8633695
10/8/2024, 3:03:21 PM Telemetry 31.540614487911295 28.88950116551797 10.79207524667342
10/8/2024, 3:03:20 PM Property 1728417800.7618253
10/8/2024, 3:03:20 PM Telemetry 59.97958804788951 22.925703228495472 4.502406246315697
10/8/2024, 3:03:19 PM Property 1728417799.7948694
10/8/2024, 3:03:19 PM Telemetry 59.27518440557587 27.29867310808949 13.248563468902477
10/8/2024, 3:03:19 PM Command response 10/8/2024, 3:03:18 PM {"data":"Command received"...
10/8/2024, 3:03:19 PM Command request 10/8/2024, 3:03:18 PM {"connectTimeoutInSeconds...

Apala's Data

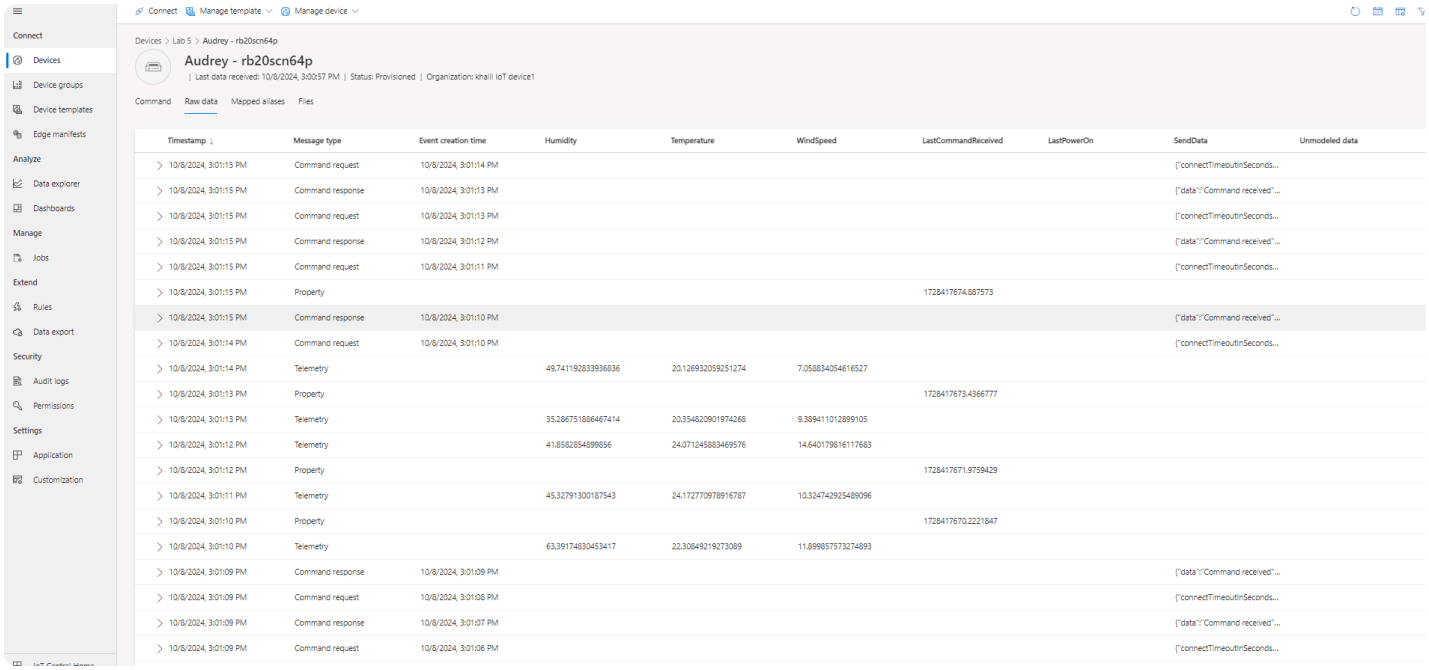

Command Prompt - python3
Telemetry sent - Temperature: 20.922714444050506, Wind Speed: 7.025165648413087, Humidity: 65.9332297351535
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '28.01763300581919', 'WindSpeed': '2.176109422630102', 'Humidity': '33.51948042041637'}
Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '26.183996589330935', 'WindSpeed': '8.63044080164111', 'Humidity': '64.81431294588546'}
Telemetry sent - Temperature: 26.183996589330935, Wind Speed: 8.63044080164111, Humidity: 64.81431294588546
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '17.6190190393742', 'WindSpeed': '9.910252490258884', 'Humidity': '38.197258130142316'}
Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '21.3952455998431', 'WindSpeed': '0.9634842866160531', 'Humidity': '33.80252383416381'}
Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '27.253710927930048', 'WindSpeed': '14.376455627809491', 'Humidity': '40.0733636474214445'}
Telemetry sent - Temperature: 27.253710927930048, Wind Speed: 14.376455627809491, Humidity: 40.0733636474214445
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '19.72843469185173', 'WindSpeed': '9.4055534567109O8', 'Humidity': '37.864050268326745'}
Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '24.327897837905l4', 'WindSpeed': '0.8427138815950286', 'Humidity': '40.519000334640375'}
Telemetry sent - Temperature: 24.327897837905l4, Wind Speed: 0.8427138815950286, Humidity: 40.519000334640375
Sending telemetry message: {'Temperature': '29.139173923113745', 'WindSpeed': '10.124432119176225', 'Humidity': '37.196103003105705'}
Telemetry sent - Temperature: 29.139173923113745, Wind Speed: 10.124432119176225, Humidity: 37.196103003105705
Sending telemetry message: {'Temperature': '27.07063931086128', 'WindSpeed': '0.97496858056886745', 'Humidity': '34.673047407515355'}
Telemetry sent - Temperature: 27.07063931086128, Wind Speed: 0.97496858056886745, Humidity: 34.673047407515355
Sending telemetry message: {'Temperature': '15.103837812666264', 'WindSpeed': '4.8196241448469085', 'Humidity': '38.41953491123921'}
Telemetry sent - Temperature: 15.103837812666264, Wind Speed: 4.8196241448469085, Humidity: 38.41953491123921
Sending telemetry message: {'Temperature': '29.61415767715416', 'WindSpeed': '4.301994935848118', 'Humidity': '66.02791173269037'}
Telemetry sent - Temperature: 29.61415767715416, Wind Speed: 4.301994935848118, Humidity: 66.02791173269037


Connect
Manage template
Manage device
Devices
Device groups
Device templates
Edge manifests
Analyze
Data explorer
Dashboards
Manage
Jobs
Extend
Rules
Data export
Security
Audit logs
Permissions
Settings
Application
Customization



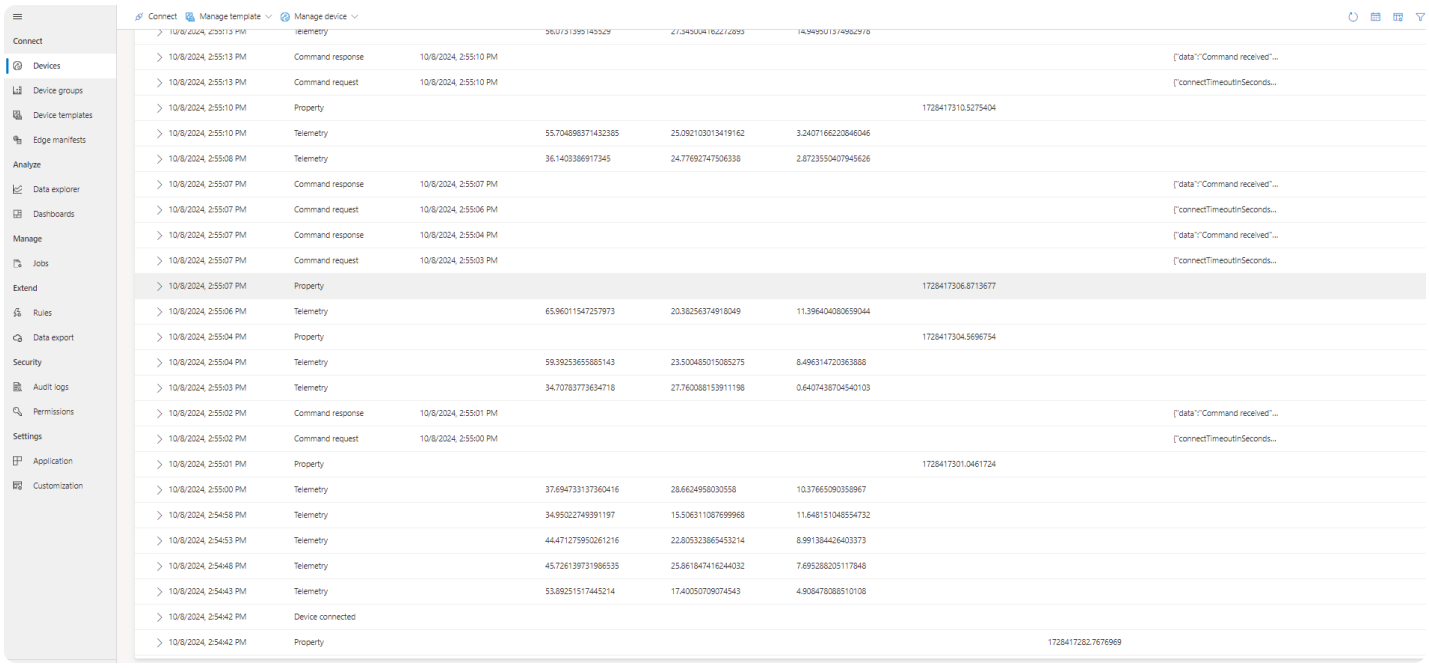
Audrey's Data

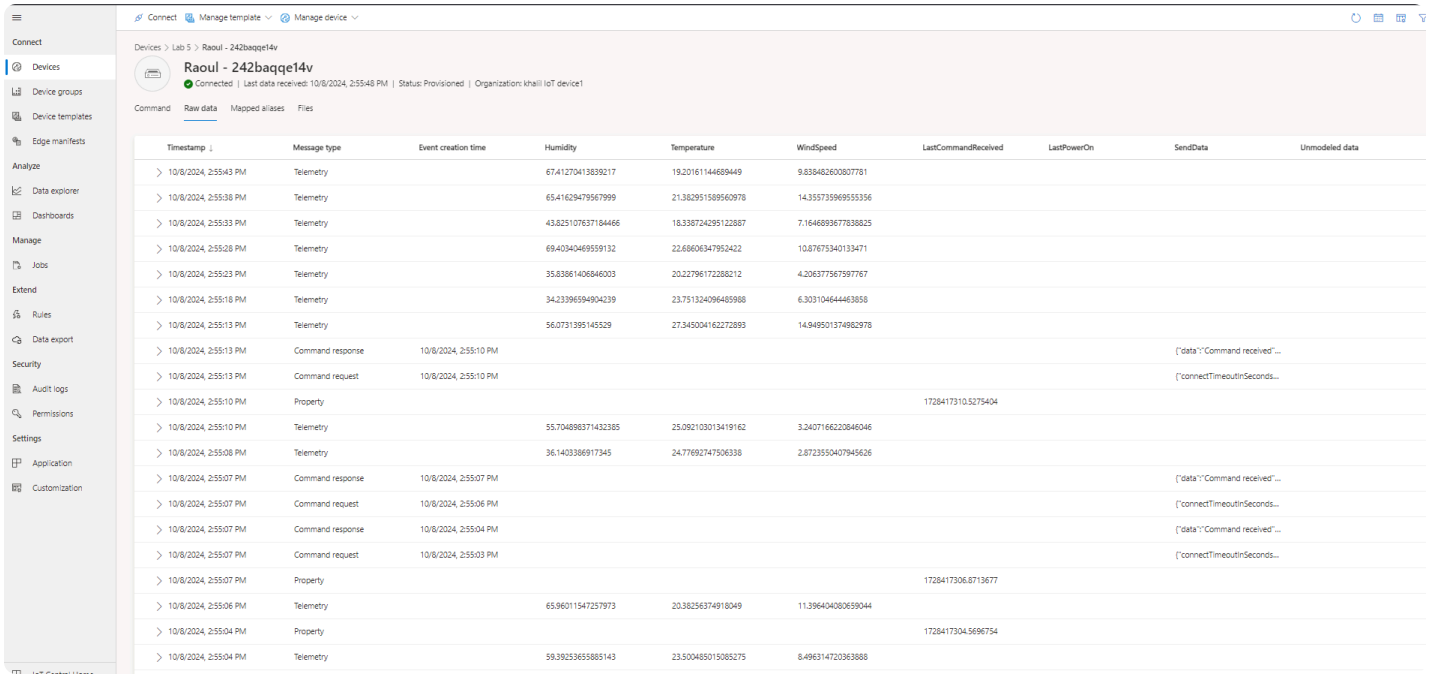
```
Device connected and LastPowerOn property sent
Sending telemetry message: {'Temperature': '27.98933687513839', 'WindSpeed': '2.0257362784817836', 'Humidity': '52.09094794649 64'}

Telemetry sent - Temperature: 27.98933687513839, Wind Speed: 2.0257362784817836, Humidity: 52.09094794649 64
Sending telemetry message: {'Temperature': '26.486423607108517', 'WindSpeed': '12.939817200198133', 'Humidity': '39.79806240853122'}

Telemetry sent - Temperature: 26.486423607108517, Wind Speed: 12.939817200198133, Humidity: 39.79806240853122
Sending telemetry message: {'Temperature': '16.17942569491931', 'WindSpeed': '14.7191016612728 41', 'Humidity': '51.10547122064736'}

Telemetry sent - Temperature: 16.17942569491931, Wind Speed: 14.7191016612728 41, Humidity: 51.10547122064736
Sending telemetry message: {'Temperature': '17.70794605377311', 'WindSpeed': '10.355354320592664', 'Humidity': '48.93687247310278'}

Telemetry sent - Temperature: 17.70794605377311, Wind Speed: 10.355354320592664, Humidity: 48.93687247310278
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '29.6269807061409 34', 'WindSpeed': '11.1620510296457 01', 'Humidity': '37.27593142362505'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '17.93767890552671', 'WindSpeed': '12.3986649281964 74', 'Humidity': '53.67527106150339'}

Telemetry sent - Temperature: 17.93767890552671, Wind Speed: 12.3986649281964 74, Humidity: 53.67527106150339
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '26.77807002655427', 'WindSpeed': '8.722287330985994', 'Humidity': '42.105739979908755'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '22.30849219273089', 'WindSpeed': '11.8998575732748 93', 'Humidity': '63.39174830453417'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '24.172770978916787', 'WindSpeed': '10.3247429254890 96', 'Humidity': '45.32791300187543'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '24.071245883469576', 'WindSpeed': '14.6401798161176 83', 'Humidity': '41.8582854899856'}

Telemetry sent - Temperature: 24.071245883469576, Wind Speed: 14.6401798161176 83, Humidity: 41.8582854899856
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '20.3548209019742 68', 'WindSpeed': '9.3894110128991 05', 'Humidity': '35.2867518864674 14'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '20.1269320592512 74', 'WindSpeed': '7.0588340546165 27', 'Humidity': '49.7411928339368 36'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '29.80845927068396', 'WindSpeed': '2.8332755127906544', 'Humidity': '58.96881184363305'}
```



Raouls's Data

```
Sending telemetry message: {'Temperature': '28.6624958030558', 'WindSpeed': '10.37665090358967', 'Humidity': '37.6947331373604l6'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '27.760088153911198', 'WindSpeed': '0.6407438704540103', 'Humidity': '34.70783773634718'}

Telemetry sent - Temperature: 27.760088153911198, Wind Speed: 0.6407438704540103, Humidity: 34.70783773634718
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '23.500485015085275', 'WindSpeed': '8.496314720363888', 'Humidity': '59.39253655885143'}

Telemetry and LastCommandReceived property sent
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '20.382563749180l9', 'WindSpeed': '11.3964040806590l4', 'Humidity': '65.960115472579l3'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '24.776927475063l8', 'WindSpeed': '2.8723550407945626', 'Humidity': '36.1403386917345'}

Telemetry sent - Temperature: 24.776927475063l8, Wind Speed: 2.8723550407945626, Humidity: 36.1403386917345
Command received: SendData
Sending data as per the SendData command
Sending telemetry message: {'Temperature': '25.092103013419162', 'WindSpeed': '3.2407166220846046', 'Humidity': '55.704898371432385'}

Telemetry and LastCommandReceived property sent
Sending telemetry message: {'Temperature': '27.3450041622728l3', 'WindSpeed': '14.9495013749829l8', 'Humidity': '56.0731395145529'}

Telemetry sent - Temperature: 27.3450041622728l3, Wind Speed: 14.9495013749829l8, Humidity: 56.0731395145529
Sending telemetry message: {'Temperature': '23.751324096485988', 'WindSpeed': '6.3031046444463858', 'Humidity': '34.233965949042l9'}

Telemetry sent - Temperature: 23.751324096485988, Wind Speed: 6.3031046444463858, Humidity: 34.233965949042l9
Sending telemetry message: {'Temperature': '20.22796172288212', 'WindSpeed': '4.2063777567597767', 'Humidity': '35.83861406846003'}

Telemetry sent - Temperature: 20.22796172288212, Wind Speed: 4.2063777567597767, Humidity: 35.83861406846003
Sending telemetry message: {'Temperature': '22.68606347952422', 'WindSpeed': '10.87675340133471', 'Humidity': '69.40340469559132'}

Telemetry sent - Temperature: 22.68606347952422, Wind Speed: 10.87675340133471, Humidity: 69.40340469559132
Sending telemetry message: {'Temperature': '18.338724295122887', 'WindSpeed': '7.1646893677838825', 'Humidity': '43.825107637184466'}

Telemetry sent - Temperature: 18.338724295122887, Wind Speed: 7.1646893677838825, Humidity: 43.825107637184466
Sending telemetry message: {'Temperature': '21.382951589560978', 'WindSpeed': '14.355735969555356', 'Humidity': '65.41629479567999'}

Telemetry sent - Temperature: 21.382951589560978, Wind Speed: 14.355735969555356, Humidity: 65.41629479567999
Sending telemetry message: {'Temperature': '19.201611446894l9', 'WindSpeed': '9.838482600807781', 'Humidity': '67.41270413839217'}

Telemetry sent - Temperature: 19.201611446894l9, Wind Speed: 9.838482600807781, Humidity: 67.41270413839217
Sending telemetry message: {'Temperature': '26.508170116768877', 'WindSpeed': '6.8284333895316465', 'Humidity': '39.93400896253121'}

Telemetry sent - Temperature: 26.508170116768877, Wind Speed: 6.8284333895316465, Humidity: 39.93400896253121
Sending telemetry message: {'Temperature': '29.393518036074802', 'WindSpeed': '0.434929818057354l4', 'Humidity': '57.70640244707l8'}

Telemetry sent - Temperature: 29.393518036074802, Wind Speed: 0.434929818057354l4, Humidity: 57.70640244707l8
```





7. Conclusion

This lab allowed us to understand how IoT devices connect to the cloud using Azure IoT Central. We were able to create virtual devices, send telemetry data, and respond to cloud commands. The simplicity of the IoT Central platform made it easier to manage multiple devices and analyze the data.

8. Appendix: Code

Python script with proper documentation.

```
import random
import time
from datetime import datetime
from iotc.models import Command
from iotc import IoTCClient, IOTCConnectType, IOTCEvents

# Replace these with your IoT Central credentials
scopeId = '0ne00CE8964'  # From lab 5.txt
device_id = '26znol6fzh7'  # From lab 5.txt
device_key = 'qoCaBmM9hRi7QJhw4HKIM9t68Od55cv2VhC5/NqNzWE='  # From lab 5.txt

# Function to handle incoming commands
def on_commands(command: Command):
    print(f"Command received: {command.name}")

    if command.name == "SendData":
        # Send data when the "SendData" command is received
        print("Sending data as per the SendData command")
        iotc.send_telemetry({
            'Temperature': str(random.uniform(15.0, 30.0)),
            'WindSpeed': str(random.uniform(0.0, 15.0)),
            'Humidity': str(random.uniform(30.0, 70.0))
        })
        # Update the LastCommandReceived property
        iotc.send_property({
            "LastCommandReceived": time.time()
        })
        print("Telemetry and LastCommandReceived property sent")

    # Acknowledge the command
    command.reply()

# Initialize IoTC client
iotc = IoTCClient(
    device_id,
    scopeId,
    IOTCConnectType.IOTC_CONNECT_DEVICE_KEY,
    device_key
)

iotc.connect()

# Set up the command listener
iotc.on(IOTCEvents.IOTC_COMMAND, on_commands)

# Send the LastPowerOn property once the device is powered on
iotc.send_property({
    "LastPowerOn": time.time()
```

```
    })

    print("Device connected and LastPowerOn property sent")

    # Start sending telemetry data every 60 seconds
    while iotc.is_connected():
        # Generate and send telemetry data for temperature, wind speed, and humidity
        temperature = random.uniform(15.0, 30.0)  # Simulated temperature in Celsius
        wind_speed = random.uniform(0.0, 15.0)    # Simulated wind speed in km/h
        humidity = random.uniform(30.0, 70.0)     # Simulated humidity in percentage

        # Send telemetry data
        iotc.send_telemetry({
            'Temperature': str(temperature),
            'WindSpeed': str(wind_speed),
            'Humidity': str(humidity)
        })

        print(f"Telemetry sent - Temperature: {temperature}, Wind Speed: {wind_speed},

        # Wait for 60 seconds before sending the next telemetry data
        time.sleep(5)
```