CS437: Internet of Things
Lab 4

Name: Michael Edukonis
NETID: meduk2
Late Days Used: 0
Video Link: https://youtu.be/QtMbw1Q8Muo
Gitlab repository: https://gitlab.engr.illinois.edu/meduk2/lab4
Team Members: N/A

March 31, 2023

CS437 Internet of Things
Lab 4 – Cloud Infrastructure Notes

1. Build the Cloud
    a. AWS account was setup
        i. 6 “Things” created each with their own TLS credentials
            1. One GreenGrass Core device – Ubuntu Linux Desktop
            2. Five emulated “vehicles”
                a. Each vehicle has it’s data to be transmitted in a separate CSV file.
                b. Each vehicle has a MQTT client to listen for messages for self only
                c. Each vehicle has their own certificate and key to authenticate with AWS via Transport Layer Security.

2. Data Inference
    a. report_CO2.py is the renamed vehicle emulator. Script will increment and process each vehicle’s data.
        i. Use Pandas to read in and process CSV data file.
            1. Will only transmit 3 columns (timestep_time, vehicle_CO2, and vehicle_id)
            2. Convert from CSV to JSON format.
        ii. Connect to AWS IOT and send MQTT message.
            1. Topic will change based on vehicle data.
            2. Also acts as a receive client but I also have 5 separate clients that will be running for demonstration purposes to show that each vehicle only receives it’s own messages.
        iii. Setup database for storage and further analysis
            1. Utilized DynamoDB in AWS – missed the part about sagemaker until the end.
            2. Installed jupyter on linux machine and successful at pulling data from DynamoDB and plotting

3. Setup AWS channel, pipeline, datastore, dataset in AWS analytics and tied to sagemaker Jupyter notebook.
    a. Successfully plotted data visualization

iv. *Lambda function to analyze emission levels (not on Greengrass – see note below)
    1. Looking at CO2 only per lab spec.
    2. Lambda will:
        a. Analyze all data rows for the car's data.
        b. Calculate the row that has the highest CO2 level
            i. Findings will include the other attributes for that row as well.
        c. Report back findings appropriately.
            i. via MQTT message
            ii. store to DynamoDB table already setup

3. Use IOT Analytics to Visualize the Data
    a. retrieve_data.py will query the dynamodb database.
        i. Pandas and Matplotlib will plot a bar chart
    b. Code and resulting chart will be handled in a Jupyter notebook
    c. Originally missed sagemaker/pipeline instruction in lab document but I was able to set these up and plot graph using AWS tools (Analytics/pipeline, sagemaker, Jupyter notebook)

4. Extra Credit
    a. N/A

All pertinent code is posted to UIUC gitlab (link above).

*Lambda is posted on AWS IOT. The video will show that although Greengrass is running on the linux core machine and reports as "healthy", I was unable to deploy Greengrass group settings (including the lambda) to the core device. The deployment status is stuck at the "In progress" state and there is nothing in the logs that would indicate a problem. In order to complete the specifications of the lab on time, it was decided to not utilize Greengrass and post the lambda on AWS.