AME 598 : Programming for IoT

Final Project Report

IoT-Based Factory/Plant Monitoring and Control System

Faculty

Tejaswi Linge Gowda

Submitted By

Ameya Ajitraj Shahane
Tejaswini Ashok Walunjkar

05 December 2023

1. INTRODUCTION

In conventional industrial environments, the absence of onsite real-time monitoring and proactive alert systems to relevant stakeholders contributes to operational inefficiencies and delays in addressing critical issues. The lack of a centralized data collection and analysis platform poses significant challenges for industries, hindering their ability to promptly identify potential equipment failures or process anomalies. This reactive approach not only jeopardizes operational continuity but also leads to unexpected downtime resulting in escalation in maintenance costs and compromising overall safety protocols.

The integration of a robust real-time monitoring system is paramount to overcoming these challenges. By employing novel data collection and analysis techniques, the resultant proactive operational model will enhance industrial performance. Real-time monitoring enables the swift detection of irregularities thus allowing timely intervention to prevent potential equipment failures and operational disruptions. This paradigm shift towards a proactive approach enhances efficiency, minimizes downtime, and ultimately reduces maintenance costs. Embracing these technological advancements not only streamlines industrial processes but also ensures a sustainable and secure operational framework for the future.

2. PROJECT STATEMENT

IIoT-Based Factory Monitoring and Control System aims to enhance operational efficiency and preemptive maintenance in an industrial setting by leveraging dashboard monitoring of sensor data. In this project, sensor devices deployed across the industrial environment continuously collect and transmit data to a centralized server for real time monitoring. Additionally, critical sensor data trigger points employ Simple Notification Service (SNS) to issue timely notifications to relevant stakeholders, enabling swift response.

3. APPROACH

The proposed approach comprises installation of a strong industrial monitoring system that uses strategically placed sensors to collect real-time data in the industrial setting. These sensors function as data gathering nodes, securely transferring information to a centralized server suited for data management. To enable easy collation and retrieval, the server employs modern data handling techniques. Real-time data analysis is used to improve the systems' capabilities by implementing complex algorithms aimed for anomaly identification and SNS for quick redressal in response to vital sensor data . Users can customize alert thresholds to tailor notifications based on specific characteristics, ensuring a prompt and targeted response to emergent situations. To give users a full insight, This proposed method includes developing of a demonstration kit, replicating a real-world industrial process plant, along with an IoT System pipeline and user-friendly dashboards. These dashboards will provide user friendly monitoring interfaces to analyze large data easily.

4. METHODOLOGY

The Internet of Things (IoT) refers to the network of interconnected devices that communicates and shares data over the internet. In industrial automation, IoT plays a crucial role by enabling seamless integration of physical devices, sensors, and systems to collect, analyze, and exchange data as shown below in fig.1. This facilitates real-time monitoring, control, and automation of industrial processes, for improved efficiency, decision making and reduced operational costs.

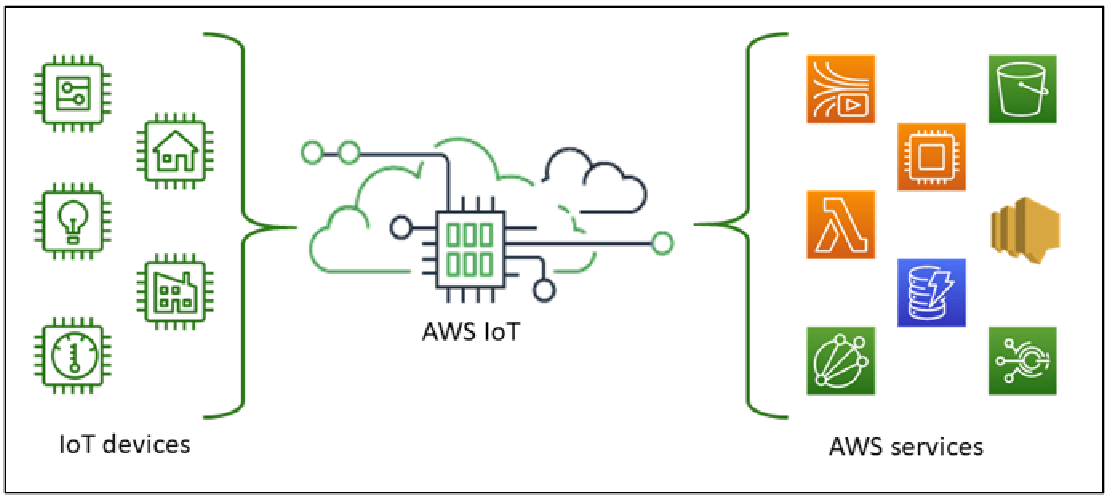


Fig1. AWS IoT core system

5. AWS IoT SYSTEM OVERVIEW

Amazon Web Services (AWS) is a cloud computing platform offered by Amazon, providing a wide array of scalable and cost-effective services. From computing power to storage and databases, AWS empowers businesses and individuals to build, deploy, and scale applications globally, transforming the way technology is utilized and managed. For this specific application, the AWS subsystems explored include:

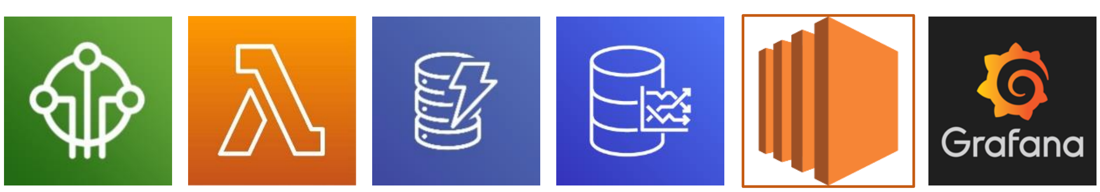


Fig2. AWS Services Symbols

a) IoT Core:

AWS IoT Core is a powerful cloud service by Amazon Web Services, designed for managing IoT devices at large scale. It ensures secure connectivity between devices and the cloud, supporting various communication protocols. Security is a paramount feature, with support for device authentication, access control through IAM, and encryption using AWS KMS. The device registry centralizes device information, and the Rules Engine processes data based on user-defined conditions, enabling actionable insights. Device Shadows provide a virtual representation of device states, ensuring application interaction even when devices are offline. Seamless integration with other AWS services, scalability, global deployment options, and robust logging and monitoring capabilities make AWS IoT Core a comprehensive solution for organizations managing and deriving value from large-scale IoT ecosystems.

b) Lambda Function:

Amazon Web Services (AWS) Lambda is a serverless compute service offering seamless execution of code without the need for infrastructure management. Developers can upload their functions, written in various supported languages, and Lambda automatically handles scaling, deployment, and resource allocation. With Lambda, users only pay for the compute time consumed by their code, making it a cost-effective solution. It supports event-driven architectures, responding to events from various AWS services or custom sources. The service enables the creation of microservices, allowing modular and scalable application development. Additionally, Lambda integrates smoothly with other AWS services, enhancing flexibility and facilitating the creation of dynamic, responsive applications.

c) DynamoDB:

Amazon DynamoDB is a fully managed NoSQL database service within the Amazon Web Services (AWS) ecosystem. Known for its seamless scalability, DynamoDB offers high-performance and low-latency access to applications' data. It accommodates various use cases, from web applications to gaming, with automatic and instant scalability to handle fluctuating workloads. DynamoDB is designed to deliver consistent single-digit millisecond performance at any scale, while its robust security features ensure data integrity. With features like automatic backups, encryption at rest and in transit, and flexible indexing, DynamoDB empowers developers to build applications that demand fast and reliable access to vast amounts of data.

d) Timestream:

Amazon Timestream is a fully managed, serverless time-series database service by AWS, designed for handling large-scale and high-performance time-series data. It enables users to effortlessly ingest, process, and query data generated over time, making it ideal for applications like IoT, industrial telemetry, and DevOps monitoring. Its purpose-built architecture supports frequent data updates and complex queries with millisecond latency. Timestream also integrates seamlessly with other AWS services, allowing users to build comprehensive solutions for real-time analytics, visualization, and actionable insights from time-series data.

e) AWS Elastic cloud Compute EC2:

Amazon Elastic Compute Cloud (EC2) is a web service offered by Amazon Web Services (AWS) that provides scalable compute capacity in the cloud. EC2 allows users to rent virtual servers, known as instances, on a pay-as-you-go basis. Users can choose from a variety of instance types based on their specific computing needs, including instances optimized for compute, memory, storage, or accelerated computing. EC2 instances can run a wide range of operating systems and support various applications. Users have full control over their instances, including the ability to start, stop, terminate, and scale capacity based on demand. Additionally, EC2 instances can be configured with security groups, network settings, and storage options, offering flexibility and control over the computing environment.

f) AWS Managed Grafana:

AWS Managed Grafana is a powerful cloud-based observability solution provided by Amazon Web Services (AWS). This managed service seamlessly integrates with popular AWS data sources, enabling users to effortlessly analyze and visualize operational data in real-time. By automating Grafana deployment, scaling, and maintenance tasks, AWS Managed Grafana simplifies the management overhead, allowing users to focus on extracting actionable insights from their data. It supports various AWS services, offering a centralized platform for monitoring, troubleshooting, and optimizing performance across diverse applications.

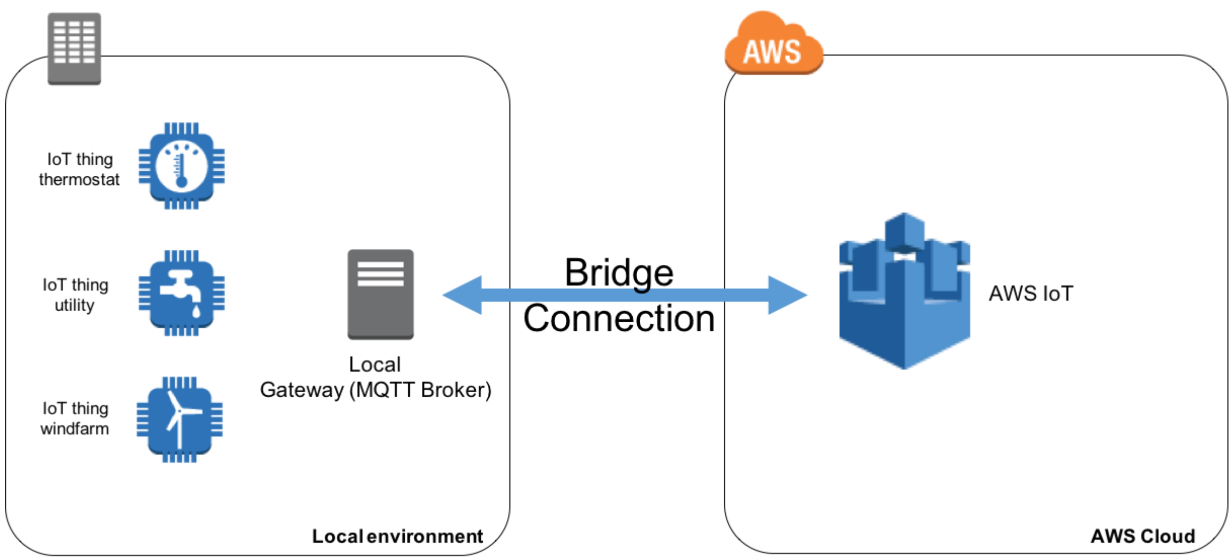


Fig 3. Device-Cloud Diagram

6. SETUP

The project was divided into two parts: hardware setup and software setup. The independent methodologies used greatly aided in the timely development, deployment, and debugging, culminating in the project's successful integration and completion.

a) Simulation/Software:

Due to delays in collecting the necessary project components, which surpassed the originally anticipated deadline, we created a simple Python code to produce sensor data at random. This simulated data was used for development of the server handler, allowing us to continue working on the project uninterrupted. This method allowed us to prevent delays and keep the project rolling until the actual components arrived. Meanwhile, the integrated data streaming and handling channels of the IoT pipeline and AWS Services were investigated. We were able to construct a solid pipeline for the final system by taking a step-by-step approach to pinging data to each AWS service. We moved on to the actual implementation without further disturbance to the project's deadline once we had obtained all of the necessary components.

b) Hardware Plant Setup:

A two-tank fluid system was developed using a do-it-yourself (DIY) approach, which included making and soldering electronic circuits, assembling of piping system, 3D printing components, and sensor integration. The soldering electronics portion entails the construction of electronic components such as voltage regulators, sensor and controllers to monitor and control fluid levels in the two tanks. Piping system integration is a critical part of the process, with a focus on secure connections and seamless fluid flow between the two tanks. We employed 3D printed solutions for manufacturing bulkheads, connectors which were not available on a commercially off the shelf basis. This enabled us to ensure precise fitment of various elements for the creation of this fluid system. From electrical component housing to customized connectors, 3D printing provided us with rapid deployment and execution of this project.

7. CHALLENGES

What is the design gap? What you set out to do vs. what you ended up ACTUALLY doing?

Our envisioned plan aimed for seamless integration of AWS services and solutions; however, resource constraints lead us to think out of the box for a simplified implementation. The challenges encountered revealed the expanse of AWS to us. While the original goal emphasized extensive real-time data processing, the final execution focused on prioritizing core functionalities due to time constraints. Despite this design gap, we achieved results aligning with key objectives to deliver a functional solution. These challenges highlight the importance of adaptability and emphasize finding alternatives for subsequent enhancements.

a) AWS Permissions and Accessibility:

Navigating through the complexity of AWS permissions in the project proved difficult, necessitating meticulous administration and configuration. The challenge entailed precise definition of permissions for various AWS services and components, while striking a balance between usefulness and security requirement. The resolution involves a thorough revision of the permission structure, with an emphasis on secure and restricted accessibility. This optimization has effectively addressed data security issues while ensuring desired access to numerous components of the IoT system.

b) Logistical Time :

Due to prolonged lead times for delivery of crucial components, we decided to implement a unique solution in the form of a Python code that simulates sensor data during the waiting period. This ensured continual development and consistent progress on the creation of this system. Working together on controllables, flexible planning, and quick adaptation resulted in a smooth changeover upon arrival of all hardware components. The creation of simulation model maintained our momentum, contributing to the successful project completion.

8. RESULTS

Systematic implementation of algorithm development and hardware configuration led to effective use of the demo kit. The screenshot given below shows the system in active state.

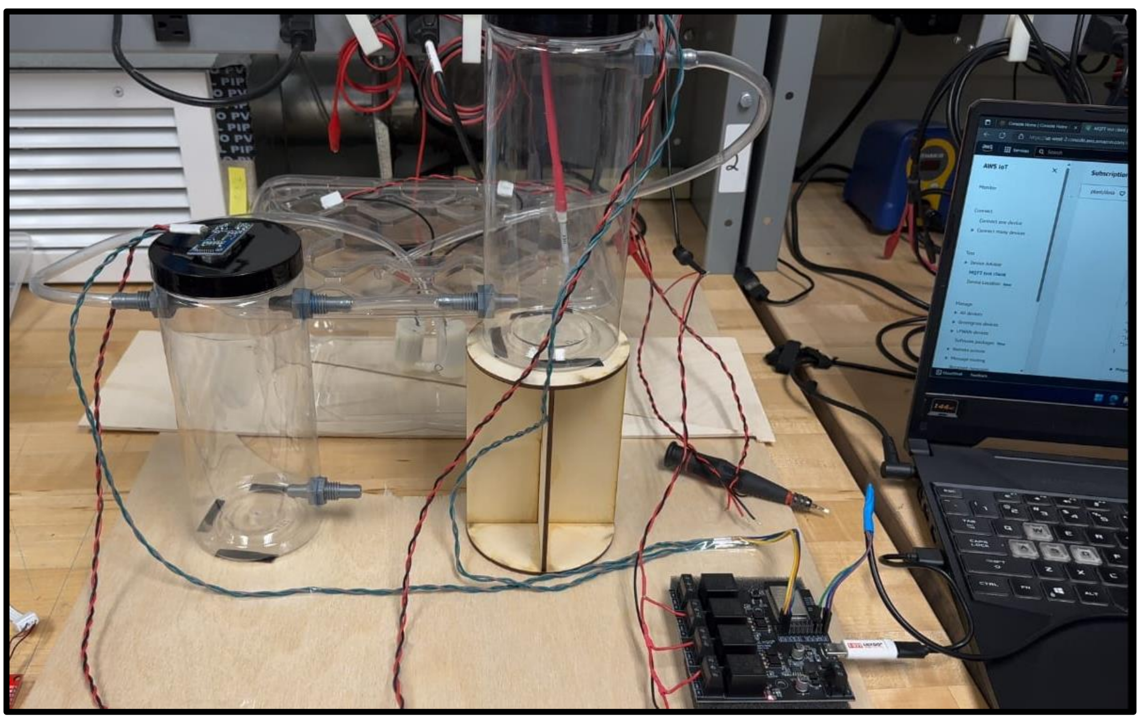


Fig 4. Physical Plant Setup

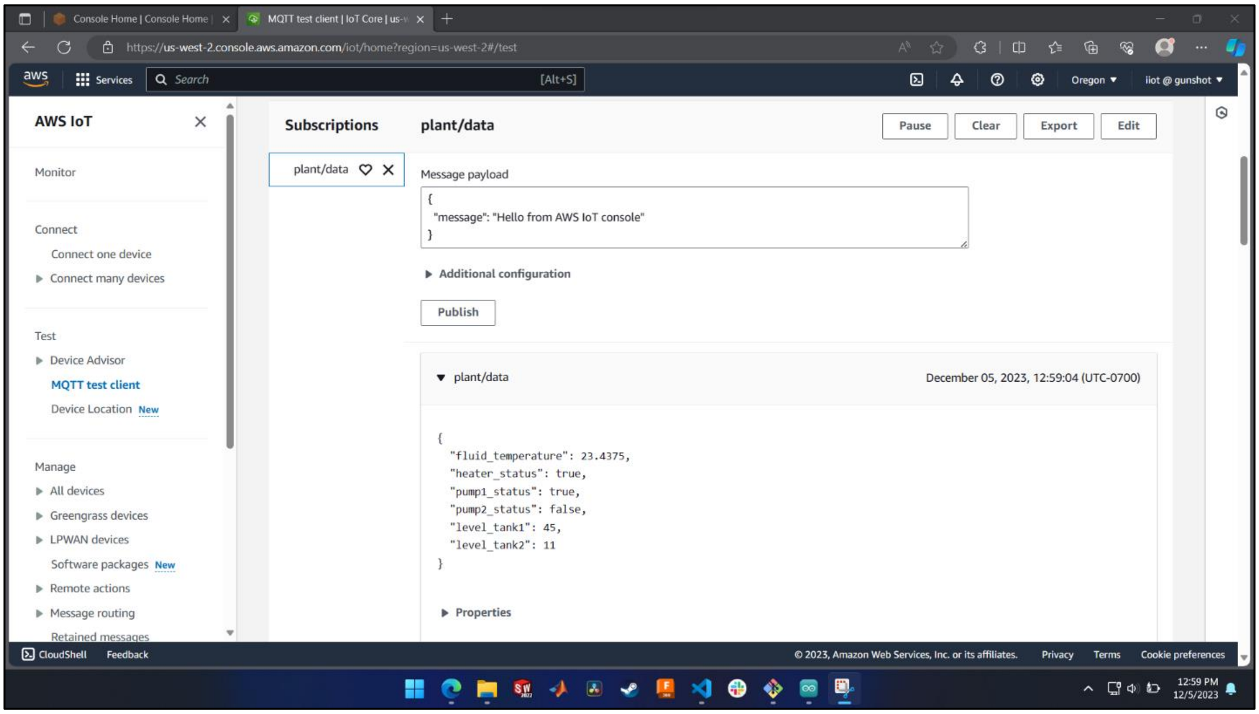


Fig 5. AWS IoT core receiving message from plant in real time

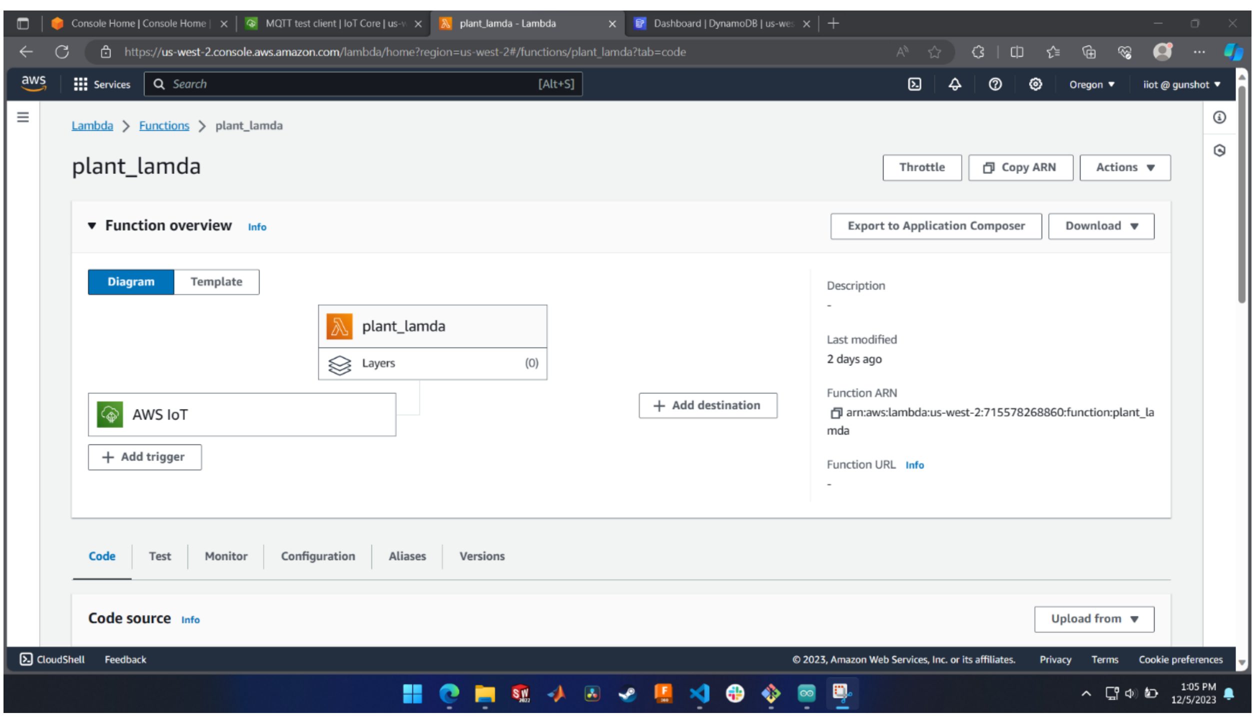
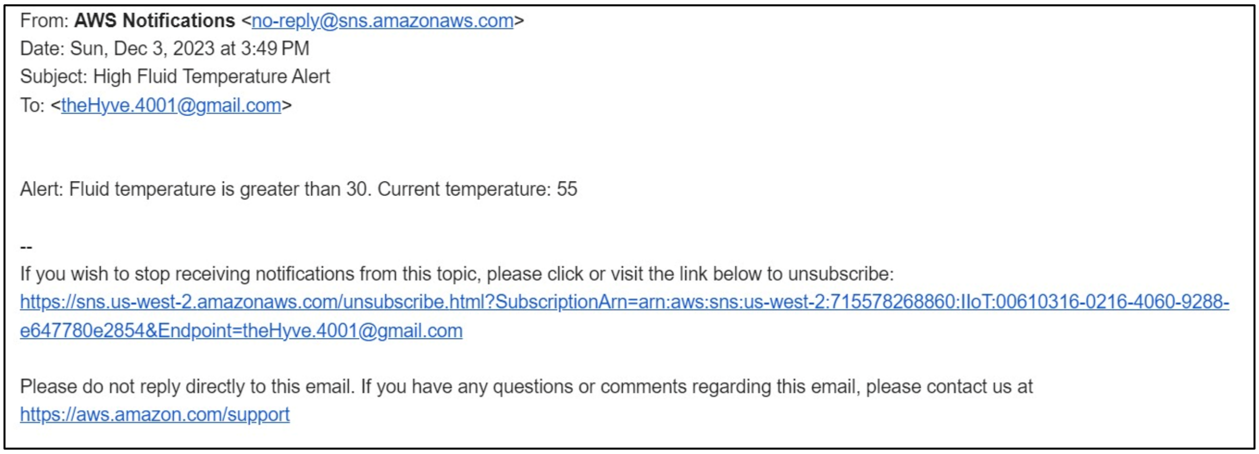


Fig 6. AWS Lambda function receives and processes data for storage



From: **AWS Notifications** <no-reply@sns.amazonaws.com>
Date: Sun, Dec 3, 2023 at 3:49 PM
Subject: High Fluid Temperature Alert
To: <theHyve.4001@gmail.com>

Alert: Fluid temperature is greater than 30. Current temperature: 55

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
https://sns.us-west-2.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-west-2:715578268860:IIoT:00610316-0216-4060-9288-e647780e2854&Endpoint=theHyve.4001@gmail.com

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at https://aws.amazon.com/support

Fig 7. AWS Lambda function also sends mail alert notifications (SNS)

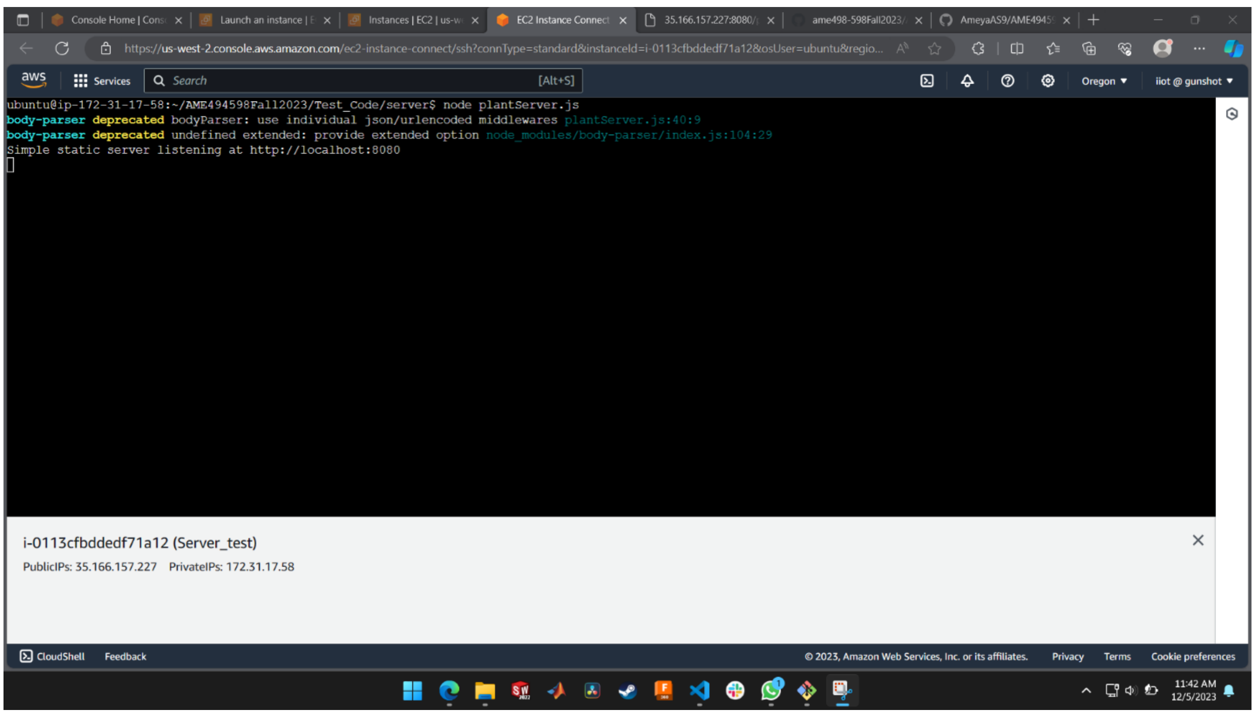


Fig 8. AWS EC2 server running to handle requests

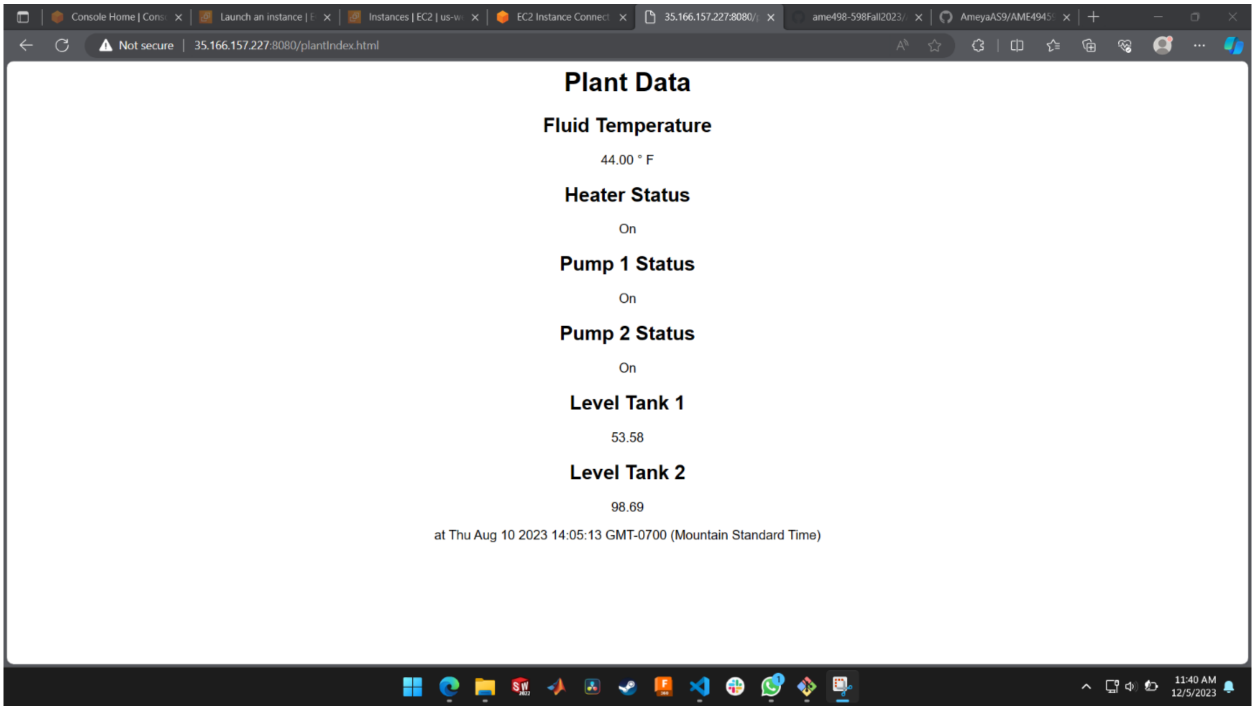


Fig 9. Dashboard running showing real time data

9. CONCLUSION

By combining data simulation model and hardware integration our IIoT solution has overcome challenges related to logistical delays and AWS configuration complexities. This project has honed our technical proficiency by the exposure to diverse fields of operation and a steep learning curve. By utilizing essential AWS services like Lambda, Timestream, DynamoDB, Grafana, and IoT Core, an advanced ecosystem was created, providing the foundation for an effective and proactive industrial monitoring system. The challenges have enlightened us about the vastness of the industrial IoT systems and concurrently enhanced our approach to problem-solving as a team. We acknowledge the invaluable direction and support provided by our faculty members, Tejaswi Linge Gowda and Assegid Kidane, without whom the project would not have been possible.