National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Devshree Ethape
Student ID: 23417196

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Devshree Ethape
Student ID:	23417196
Programme:	Cloud Computing
Year:	2025-2026
Module:	MSc Research Project
Supervisor:	Rashid Mijumbi
Submission Due Date:	28/01/2026
Project Title:	Unified Multi-Feed Anomaly Detection for AWS Lambda
Word Count:	10864
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Devshree Ethape
Date:	28th January, 2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Devshree Ethape
Student ID: 23417196

1 Introduction

This setup guide describes the way the AWS cloud resources were configured to create a real-time anomaly detection pipeline of AWS Lambda. The system discovers CloudWatch metrics, streams them to a Kinesis stream, processes them with Apache Flink in a Zeppelin notebook and generates alerts over Amazon SNS in case abnormal behaviour is observed. Everything is on AWS controlled services, and thus there are no servers or containers to be maintained. This manual will set out to explain the configuration of each service, the interaction of the components and how the system can be brought into deployment or how it can be replicated.

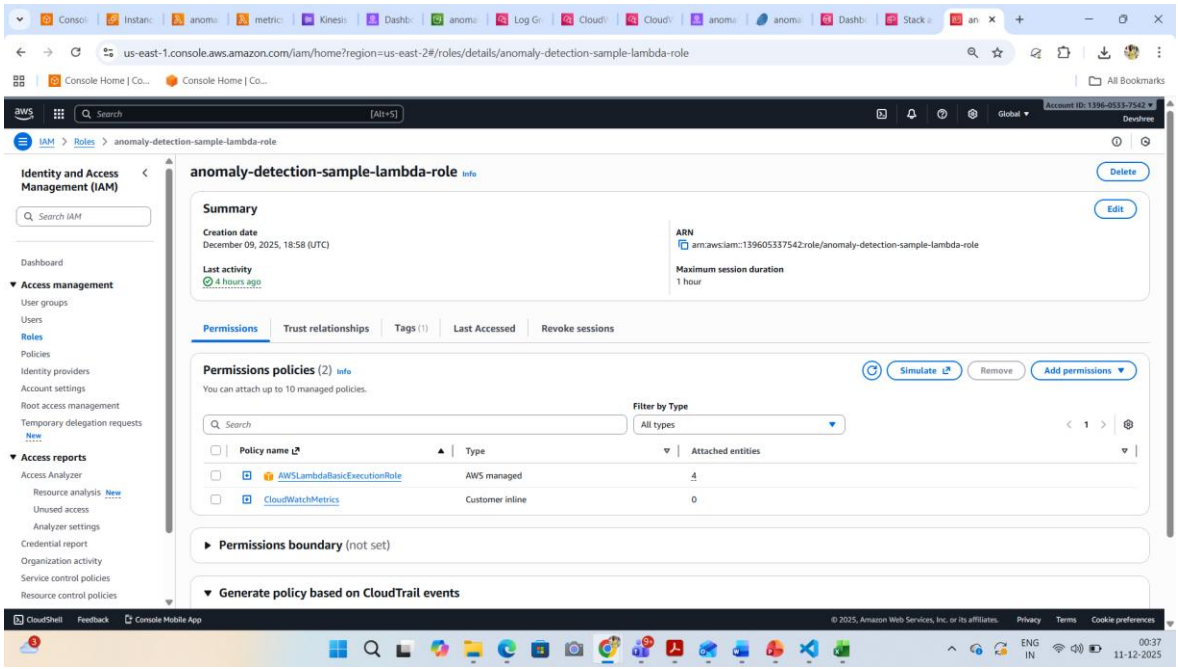
2 System Requirements and Architecture Overview

2.1 AWS Services Required

To deploy this system, the following AWS services must be enabled:

- **AWS Lambda** for execution of metric extraction and alerting.
- **Amazon CloudWatch** to store and make available Duration telemetry.
- **Amazon Kinesis Data Streams** to transport telemetry and anomaly outputs.
- **Amazon EventBridge to** scheduled Trigger for Metrics Collection
- **Amazon S3 Storage** for Raw Metrics and Anomaly Events
- **Amazon Managed Service for Apache Flink** for analytics and threshold detection.
- **Amazon SNS** to deliver email or SMS alerts to operators.
- **AWS Identity and Access Management (IAM)** for secure role assignment.

The pipeline is based on event-driven architecture to make sure that messaging, decoupling, and horizontal scaling are efficient. CloudWatch measurements serve as the input, Kinesis as the streaming bus, Flink as the transformation and detecting unit and SNS as the communication tool.



2.2 Developer Tools and Access

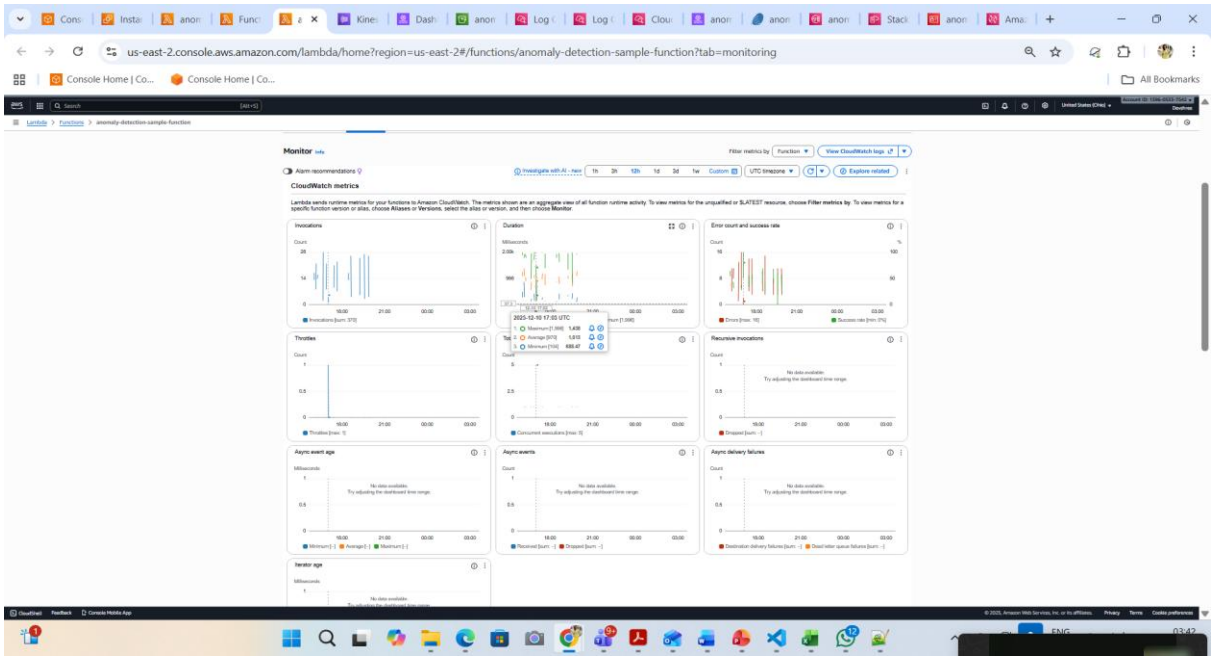
To configure and maintain the system, the following are recommended:

- AWS Console access
- AWS CLI (optional but recommended)
- Python 3.9 or later for Lambda functions
- Text editor or IDE (e.g., VSCode)
- Basic knowledge of IAM configuration and AWS services

3 Configuration Steps

3.1 Configure CloudWatch Metrics

Duration metric is automatically published in AWS Lambda into CloudWatch. There is no need to create a set up to create metrics, but the user can choose to have enhanced monitoring to be more accurate. Measurements will be stored in CloudWatch as default retention policies unless otherwise (Gregor, 2024).



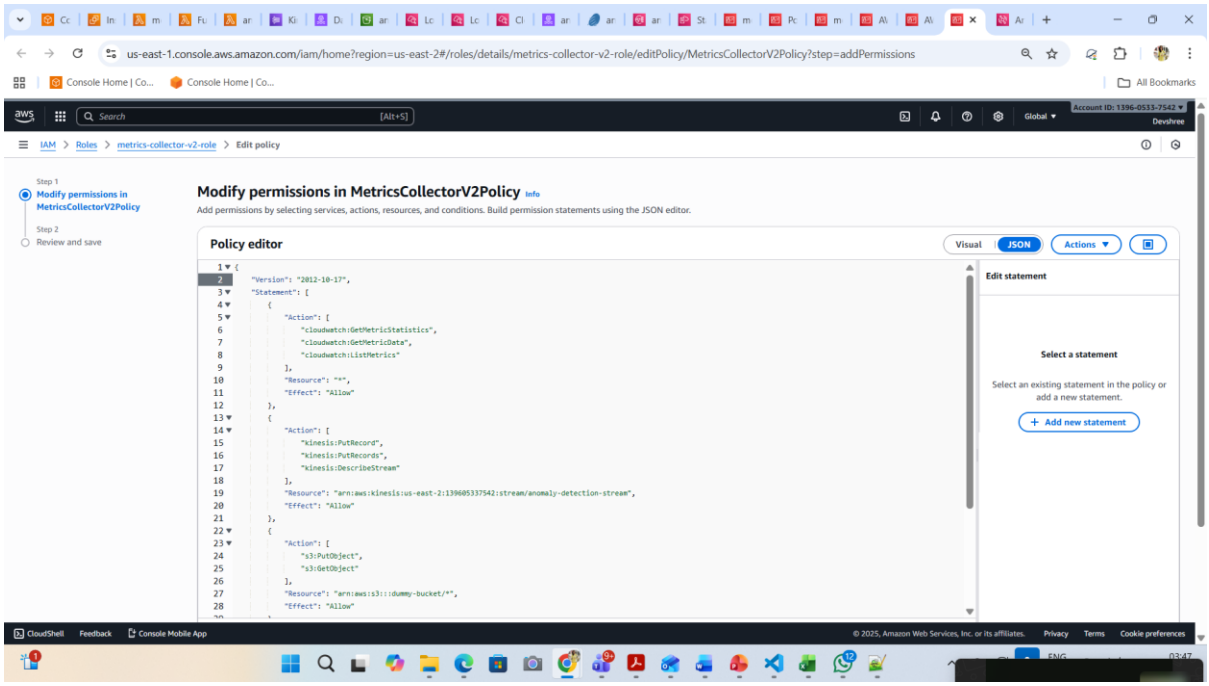
3.2 Create the Metric Extraction Lambda

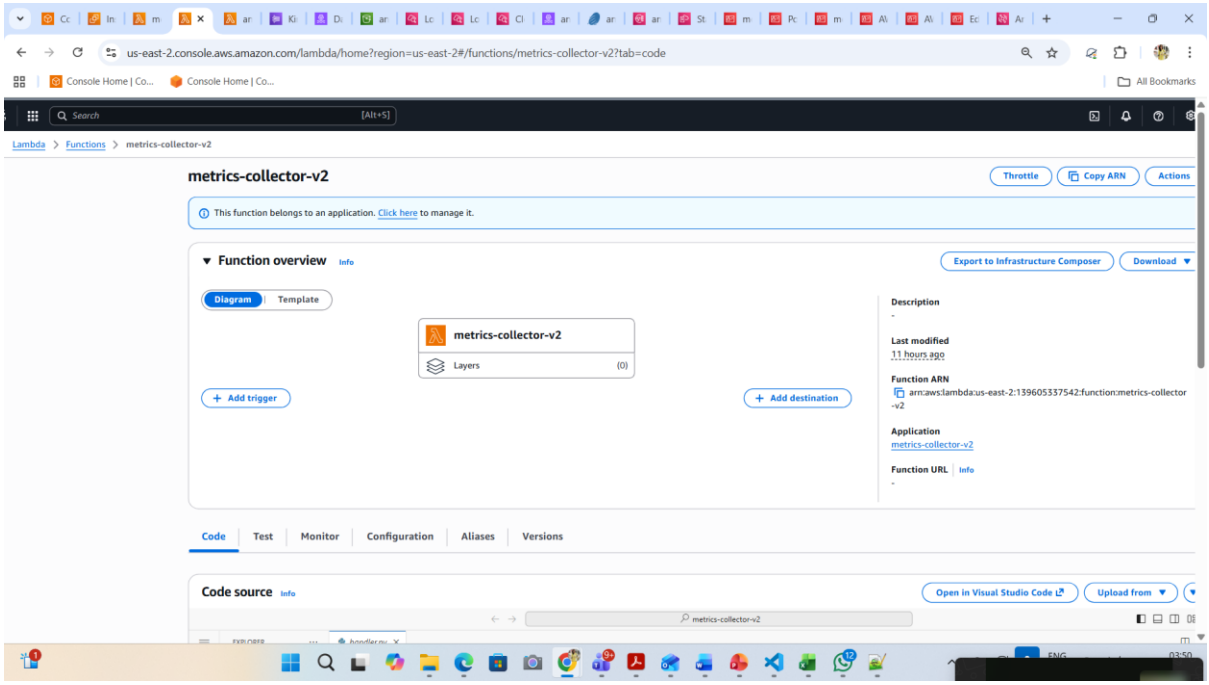
A Lambda function will have to be developed that will query CloudWatch, retrieve the latest values of the most recent execution durations, and insert them to the first Kinesis stream.

Steps:

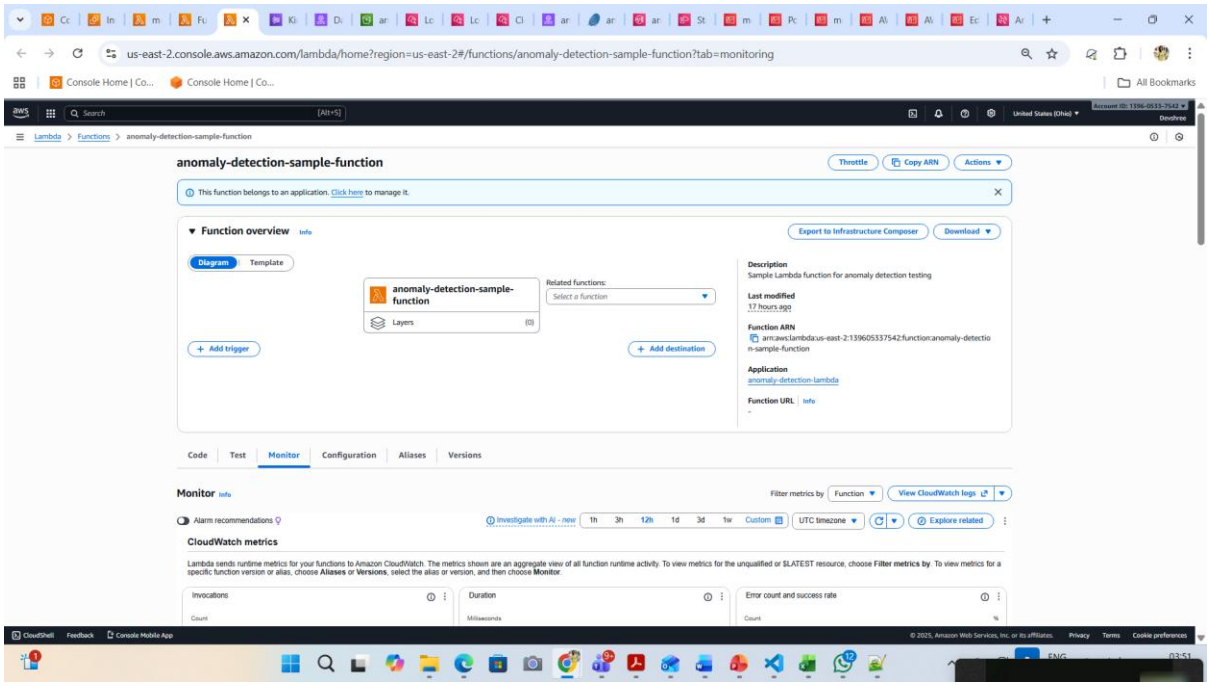
1. Navigate to **AWS Lambda → Create Function**
2. Choose **Author from Scratch**
3. Name the function: metrics-collector-v2
4. Select Python 3.9 Runtime
5. Create or assign an IAM role with permissions to:
    - cloudwatch:GetMetricStatistics
    - kinesis:PutRecord
6. Insert the Python code (as supplied in Appendices of the main report)
7. Deploy the function
8. Add a **CloudWatch Scheduled Trigger** to invoke every 60 seconds

This function converts duration metrics into JSON objects and pushes them to the Kinesis Raw Metrics Stream.





This Lambda-function gathers CloudWatch metrics of the observed function. It produces a metric record in the form of a small JSON and writes it to Kinesis stream after every two minutes. The fields of every record include invocation count, errors, error rate, duration, throttles and concurrency. The two functions send logs to CloudWatch as part of their respective log groups.



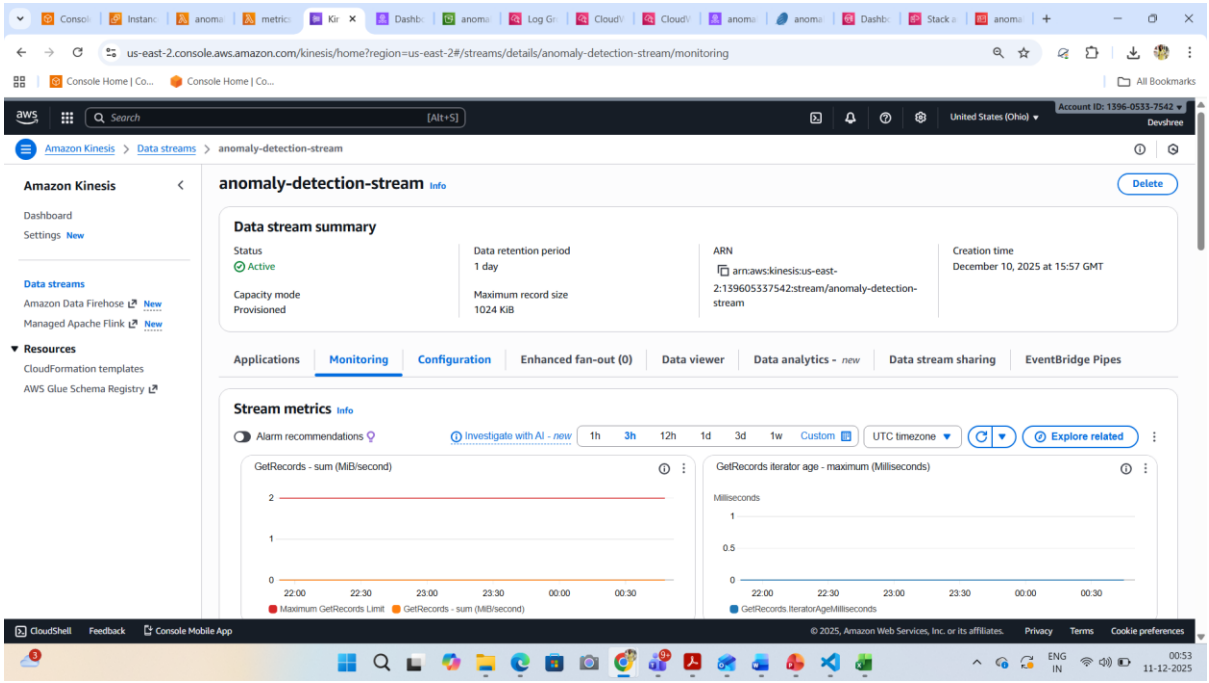
This is the monitored function in the project and is Lambda. It executes Python code in simple code to produce executions of different lengths with some errors. Such executions generate actual CloudWatch metrics which can be analyzed by the detection system.

3.3 Create Kinesis Data Streams

Kinesis stream is required:

- **anomaly-detection-stream**

  Select Scaling with Traffic Spikes On-Demand. Configure default encryption and give the IAM role with which Flink is connected access to read and write both streams.



3.1.1 Environment Variables Used in the System

The two Lambda functions in the project also utilize environment variables to evade configuration information hard-coding. This enables the system to be brought up to date or even extended without changing the application code.

Variable	Purpose	Example Value
KINESIS_STREAM_NAME	Name of the stream where metrics are published	anomaly-detection-stream
TARGET_FUNCTION_NAME	Lambda function that is being monitored	anomaly-detection-sample-function
RAW_EVENTS_BUCKET	S3 bucket for storing raw and anomaly records	anomaly-detection-raw-events-139605337542
SNS_TOPIC_ARN	SNS topic for sending alert notifications	arn:aws:sns:us-east-2:139605337542:anomaly-detection-alerts
AWS_REGION	Region of deployment	us-east-2

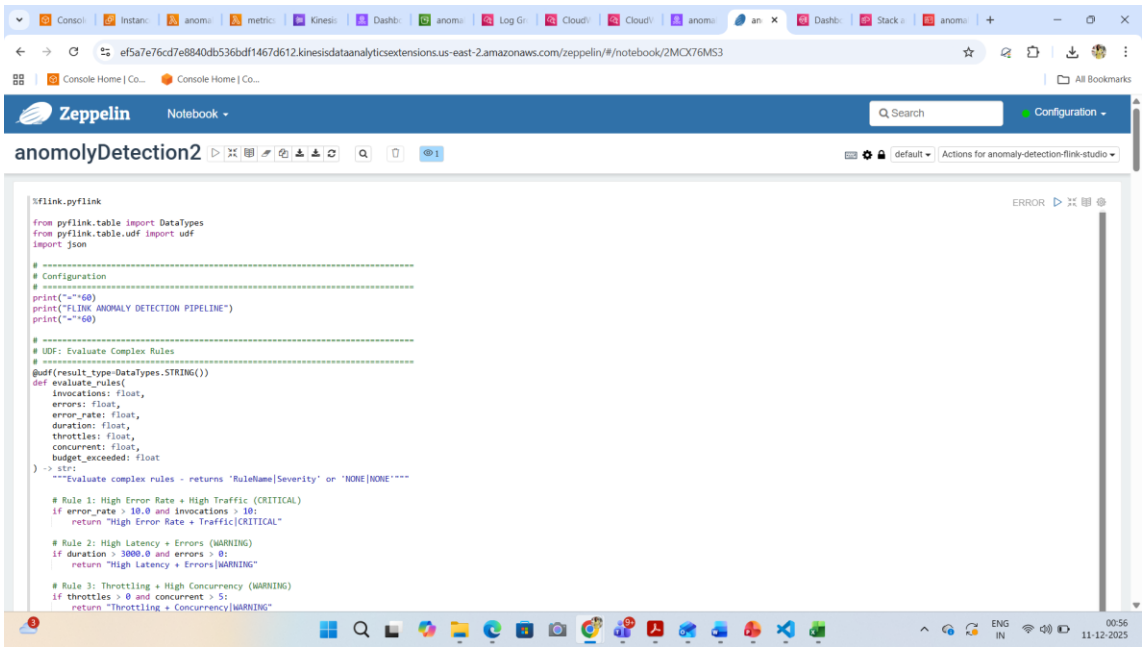
3.4 Configure Apache Flink Analytics (Zeppelin)

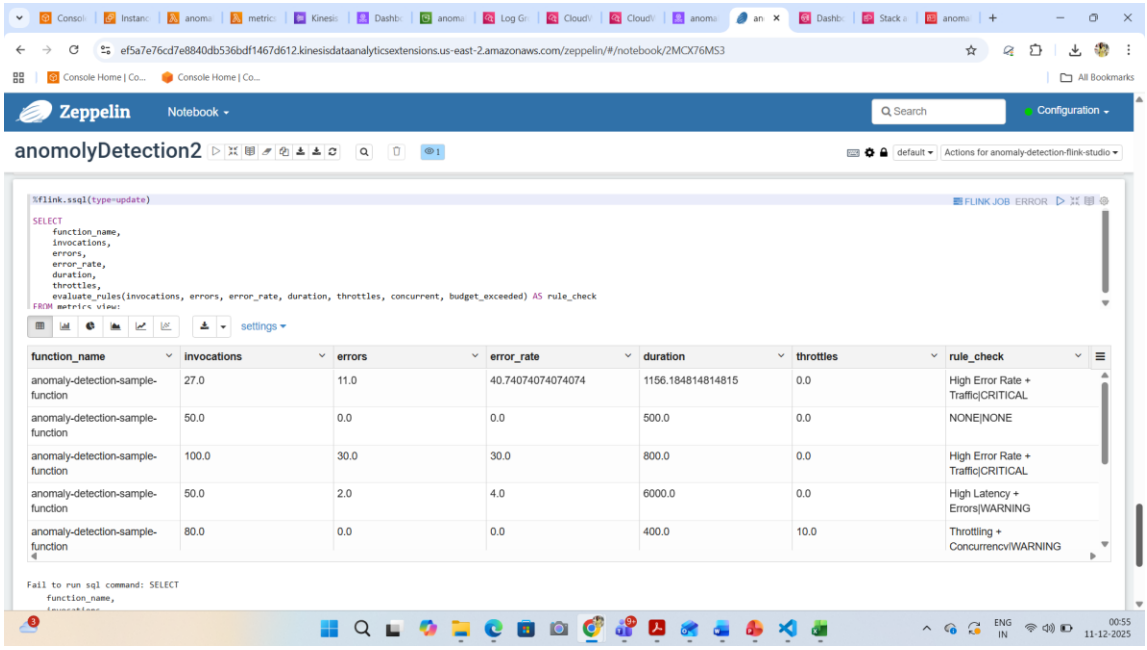
The anomaly detection logic logic is performed by SQL which is processed by Amazon Managed Service of Apache Flink. Once a studio notebook session has been created, use the raw and anomaly tables mapped to the corresponding streams and run SQL which applies threshold logic (duration > 2000ms by default).

Upon the completion, Flink processes streaming data repeatedly and directs anomalies to the specific anomaly stream. Such reasoning can be replaced by machine learning inference in the future to aid in multi-feed detection (Nguyen, Elmroth and Bhuyan, 2025).

Apache Flink notebook was also improved with the better processing by providing the option of embedding custom Python UDFs into the analytics layer. Three functions have been added, evaluate_rules, checking each metric with various rules including high error rates, abnormal latency, throttling, or potential Denial-of-Wallet behaviour and provides a simple output of RuleNameSeverity, send_sns_alert, which pushes real-time alerts directly out of Flink to the SNS topic anomaly-detection-alerts including the name of the rule, the rule triggered, the severity and the

key metrics and store_s3 which stores anomaly records as JSON files in the S3 path anomaly-detection-raw- The revised pipeline reads the events on Kinesis and customizes the rule engine, emitting alerts and storing anomalies to be sent only when there is a need, and printing the outcomes in Zeppelin. This simplified process renders the system more encompassing, effective and workable.





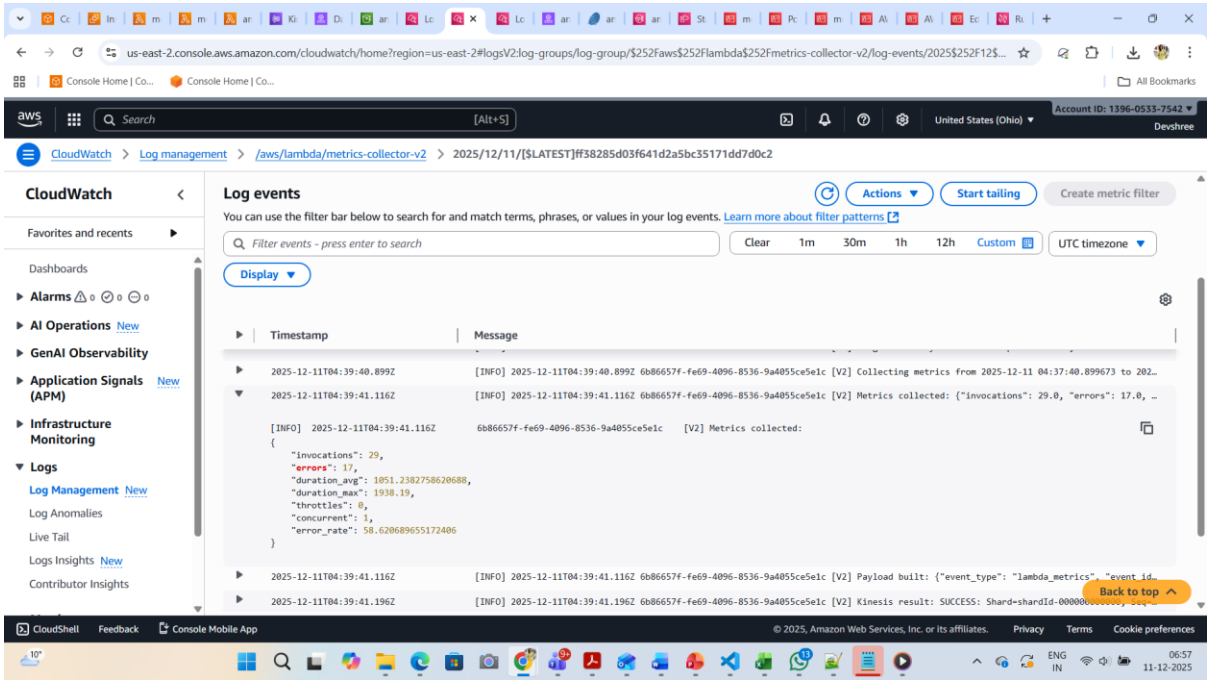
3.5 Amazon CloudWatch Metrics

CloudWatch gathers performance data of the observed Lambda such as:

- Invocation count
- Average and maximum duration.
- Error count and error rate
- Throttles
- Concurrent executions

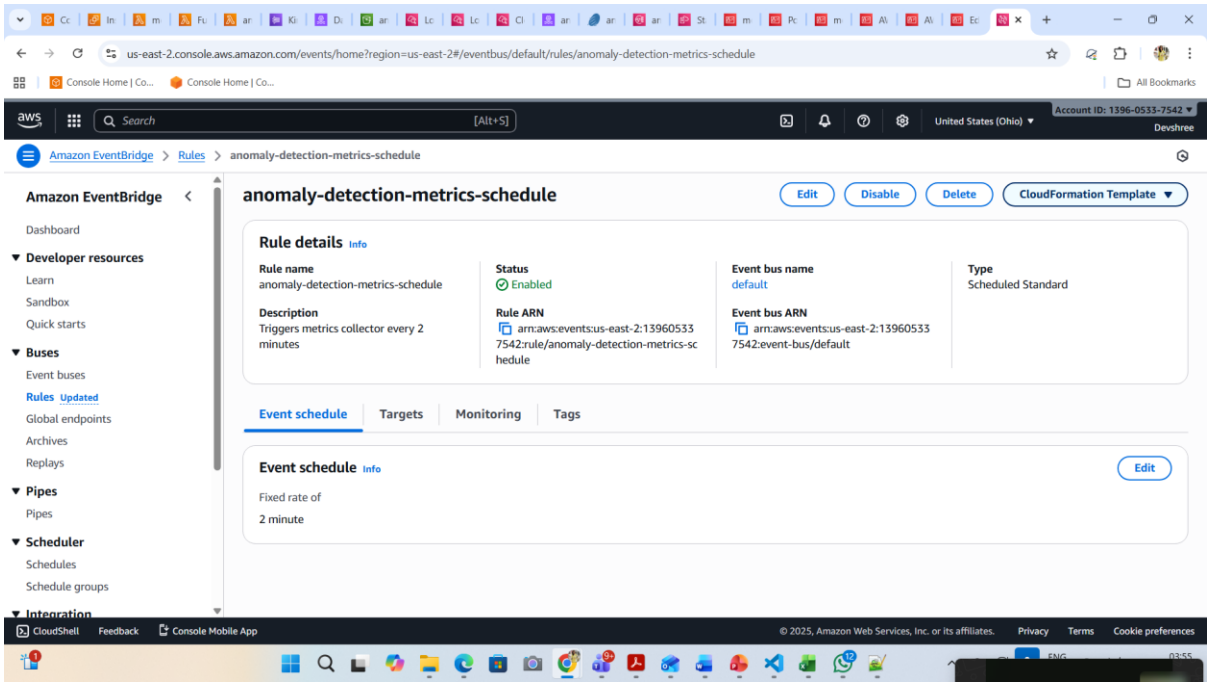
The metrics collector, Lambda, queries these values with the CloudWatch API and transforms them into JSON then sends them to Kinesis.

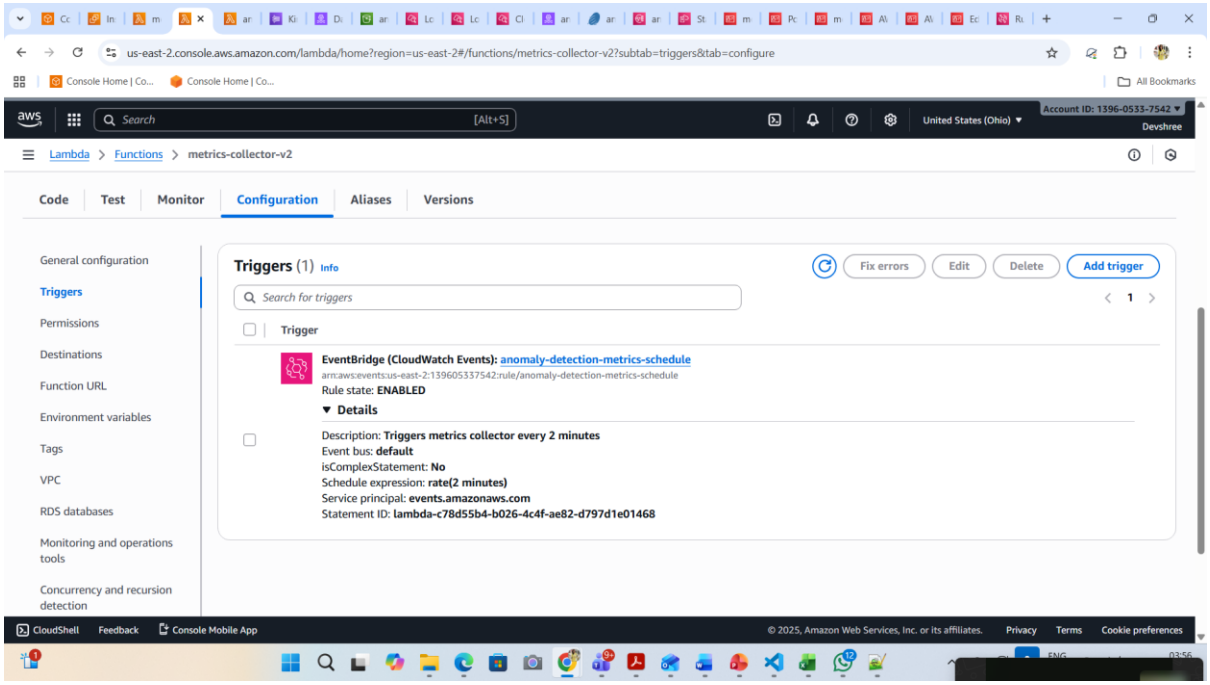
These runtime logs of both Lambda functions are also stored in CloudWatch logs and are useful during debugging.



3.6 Amazon EventBridge — Scheduled Trigger for Metric Collection

EventBridge is deployed to have an automatic trigger of the Metrics Collector Lambda after every two minutes. This will make the CloudWatch metrics collected automatically and without human intervention. The timely aspect of the scheduled rule is the time-driven component of the whole anomaly detection pipeline.





3.7 Create the SNS Notification System

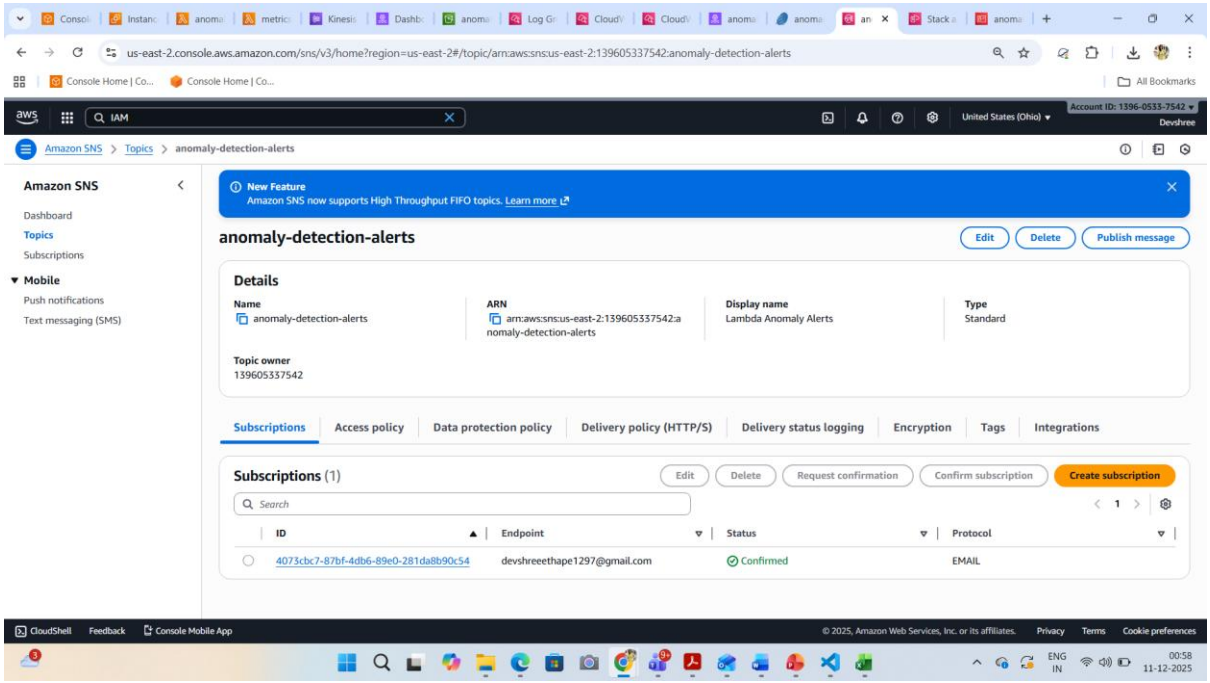
Click on the SNS Topic, and create an SNS Topic called lambda-anomaly-alerts and subscribe an email address and confirm it through inbox. Install the second Lambda function (lambdaAnomalyToSNS) which will read the anomaly stream and post messages about the detected anomalies in the form of alert messages to SNS (Escaleira et al., 2025).

Configure IAM permissions permitting:

- Reading from lambda-anomaly-stream

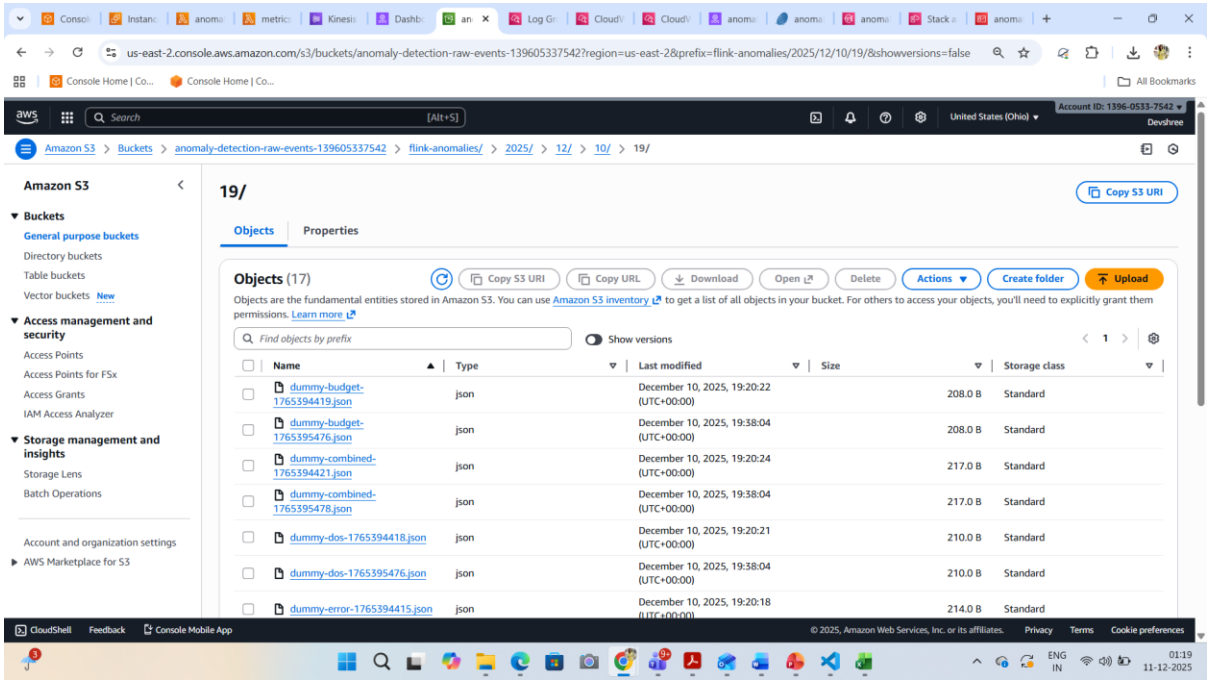
- Publishing to SNS Topics

SNS becomes the final communication endpoint for operators.



3.8 Creation of Amazon S3 Storage for Raw Metrics and Anomaly Events

All the raw events in the form of metrics that the Metrics Collector Lambda gathers and the records of anomalies that the Flink application produces are stored in Amazon S3. The logs act as a historical record to be analyzed and validated to be used as machine-learning extensions in the future.



4 Testing and Validation

The experiment was done by using a combination of both manual invocations and Flink notebook output.

4.1 Lambda Testing

The application of the monitored feature created actual CloudWatch metrics.

Because metrics had been published in Kinesis, invocation of the collector Lambda ensured this.

4.2 Kinesis Stream Validation

The Kinesis console indicated continuous incoming records when being tested.

4.3 Flink Notebook Validation

The results of the print were:

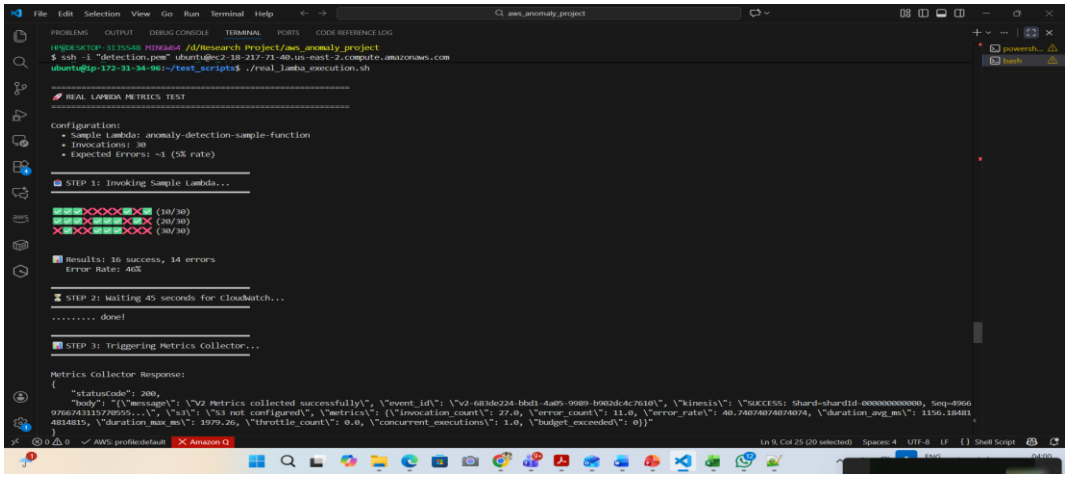
- raw metric values
- triggered rule results
- SNS publish status
- S3 storage status

4.4 SNS Notification Testing

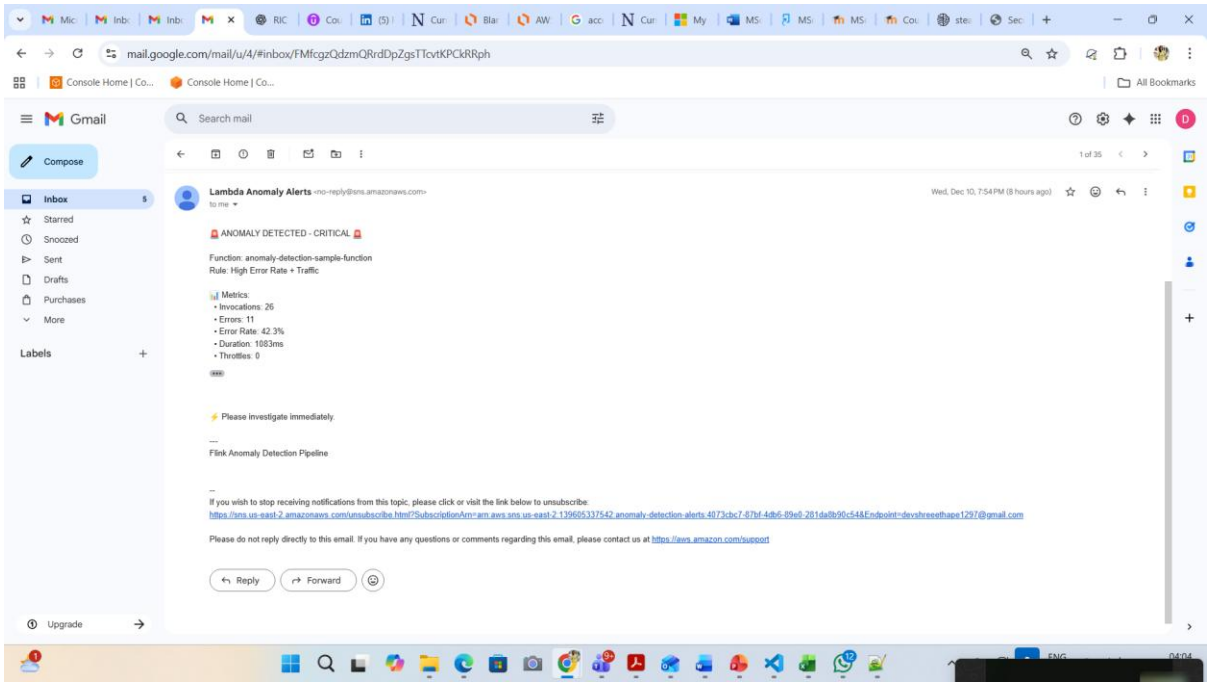
The verified SNS subscription was able to receive email alerts.

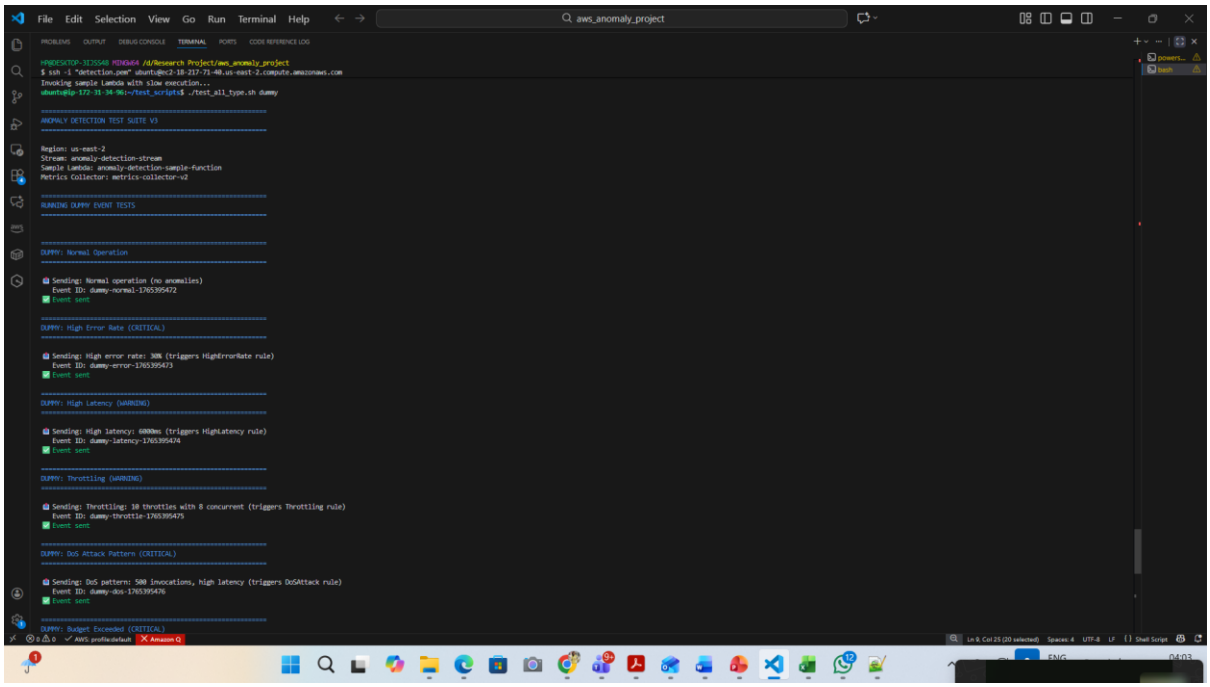
4.5 S3 Storage Testing

Every anomaly would generate a new JSON file in the appropriate folder structure, which proved that the pipeline reliably stored the results.

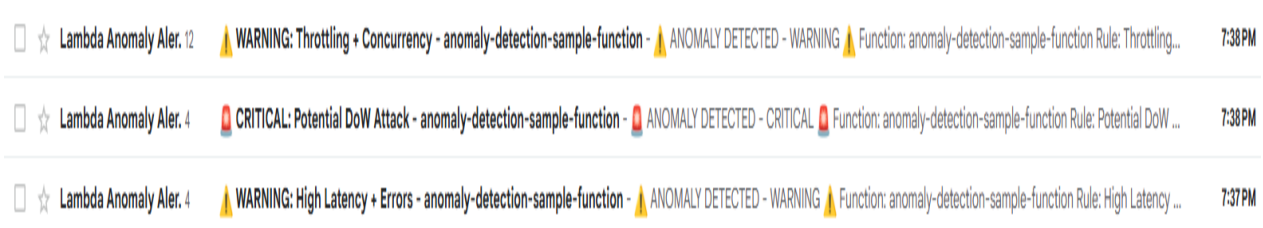


Test with real lambda function metrics:-





Test with simulated dummy lambda metrics:-



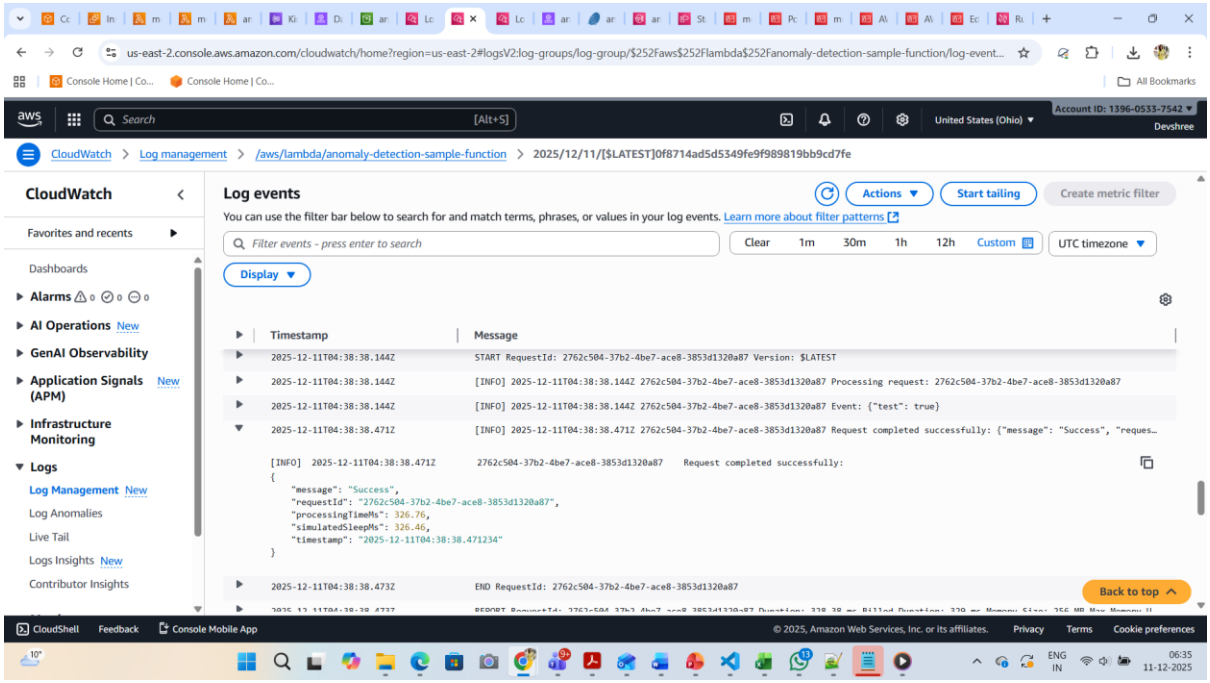
4.6 CloudWatch Log Groups to debugging.

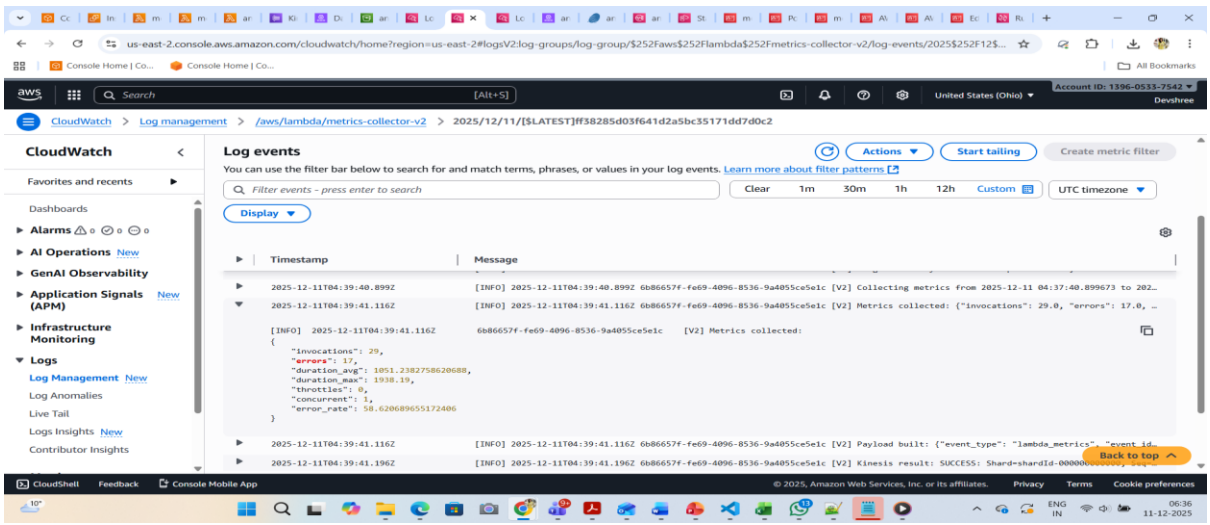
Lambas are automatically configured to have log groups created by AWS. Such records will be necessary in the process of confirming that the pipeline is functioning properly. Some of the groups of logs that were produced during the deployment are as follows:

- /aws/lambda/anomaly-detection-sample-function
- /aws/lambda/metrics-collector-document
- aws/lambda/lambdaAnomalyToSNS

The log groups help verify:

- Whether the metric collection Lambda is timely or not.
- Success of metrics publication to Kinesis.
- Any mistakes during de-coding payloads SNS publishing failures
- System anomalies passing through the system.
- Troubleshooting tool used in the development was cloudwatch logs.





5 Security Configuration and IAM

The principle of least privilege is adhered to by the system. The individual resources, including extractor Lambda, Flink analytics and alerting Lambda, should be given a dedicated IAM role with only the necessary permissions.

Security expectations:

- No wildcard permissions (*:*:*) should be used.
- IAM roles should be reviewed post-deployment.
- SNS and Kinesis encryption remain enabled by default.
- When production data is used, implement CloudTrail monitoring.

Future deployments should consider managed keys, log access controls, and compliance measures for data governance.

6 Maintenance and Operational Tasks

After deployment, the system requires minimal maintenance. The main operational tasks include:

- Restarting Flink jobs if terminated
- Updating threshold values based on new performance baselines
- Adding new Lambda functions as additional feeds
- Reviewing SNS logs to ensure delivery
- Monitoring CloudWatch for sustained behavioural deviation trends

Future expansion may include dynamic thresholding, integration with Route 53 health checks, or visual dashboards to improve situational awareness (Amazon Web Services, 2023).

7 Troubleshooting Guide

Common issues include:

- No anomalies detected → Verify extractor Lambda and Kinesis stream data.
- Alerts not received → Confirm SNS subscription or check for IAM publish restrictions.
- Flink SQL fails to execute → Confirm stream write permissions and region match.

8 Conclusion

This configuration manual describes the entire configuration of the AWS parts that were involved in the anomaly detection system. The last pipeline is based on managed services with the ability to expand automatically and work without human control. The system gathers metrics, analyzes anomalies on the basis of rules, emits the alert and retains anomaly data in an efficient way. The setup is scalable and can be extended to additional components e.g. machine-learning models or monitoring of multiple Lambda functions.

References

Gregor, S. (2024) 'Design science research and the co-creation of project value', *International Journal of Project Management*, [online]. Available at: https://www.sciencedirect.com/science/article/pii/S0263786324000267 [Accessed 29 Nov. 2025].

Nguyen, C., Elmroth, E. & Bhuyan, M. (2025) 'Silent Failures in Stateless Systems: Rethinking Anomaly Detection for Serverless Computing', *arXiv preprint*, July. Available at: https://arxiv.org/abs/2507.04969 [Accessed 29 Nov. 2025].

Escaleira, P. et al. (2025) 'A systematic review on security mechanisms for serverless computing', *Concurrency and Computation: Practice and Experience*, [online]. Available at: https://link.springer.com/article/10.1007/s10586-025-05371-4 [Accessed 29 Nov. 2025].

Amazon Web Services (2023) *Real-time time series anomaly detection for streaming applications on Amazon Managed Service for Apache Flink*. Available at: https://aws.amazon.com/blogs/big-data/real-time-time-series-anomaly-detection-for-streaming-applications-on-amazon-managed-service-for-apache-flink/ (Accessed: 08 December 2025).