Enterprise Continuous Monitoring and Threat Detection on AWS
Author: Adedayo
Specialization: Cloud Security & Threat Detection
Platform: Amazon Web Services (AWS)

1. Introduction

This document describes the design and implementation of a centralized monitoring and threat detection framework in an AWS environment.

I implemented this solution to improve visibility into API activity, network traffic, security misconfigurations, and threat indicators, while enabling rapid incident response.

2. Objectives

The primary objectives were to:

- Enable continuous security monitoring
- Detect suspicious IAM and API activity
- Identify network-based threats
- Centralize security findings
- Automate low-severity remediation
- Improve incident response readiness

3. Security Monitoring Architecture

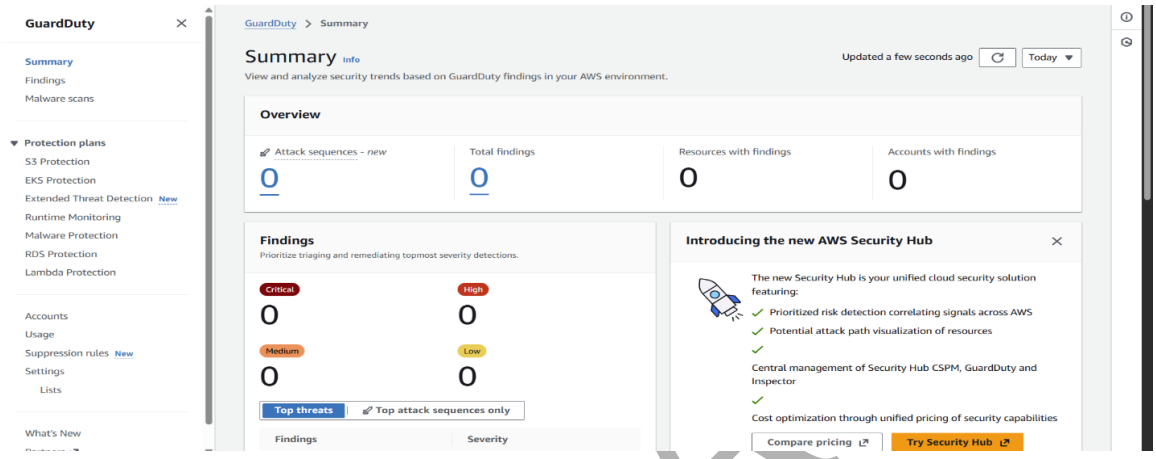
The monitoring framework was built using the following AWS services:

- Amazon GuardDuty
- AWS CloudTrail
- AWS Security Hub
- VPC Flow Logs
- Amazon CloudWatch
- Amazon SNS
- AWS Lambda

All security events and findings were centralized for analysis and response.

4. GuardDuty Configuration

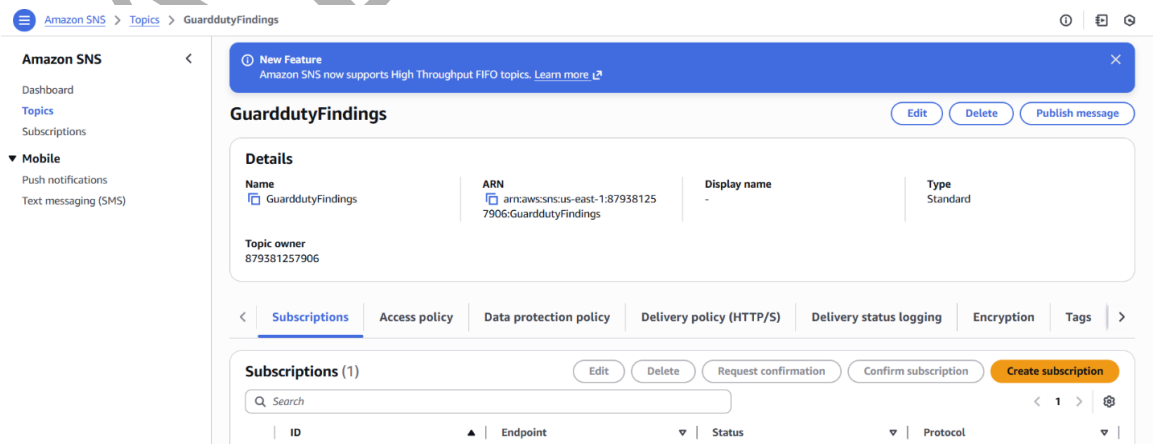
I enabled Amazon GuardDuty to monitor:



- CloudTrail logs
- VPC Flow Logs
- DNS activity

GuardDuty was configured to generate findings for suspicious behavior and compromised credentials.

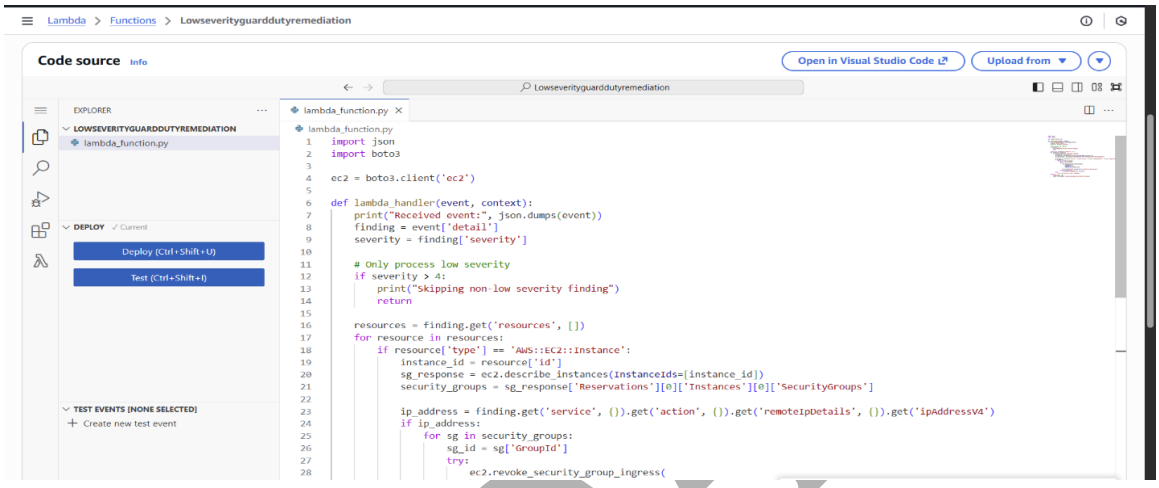
An SNS topic was created to distribute GuardDuty alerts to the security team.



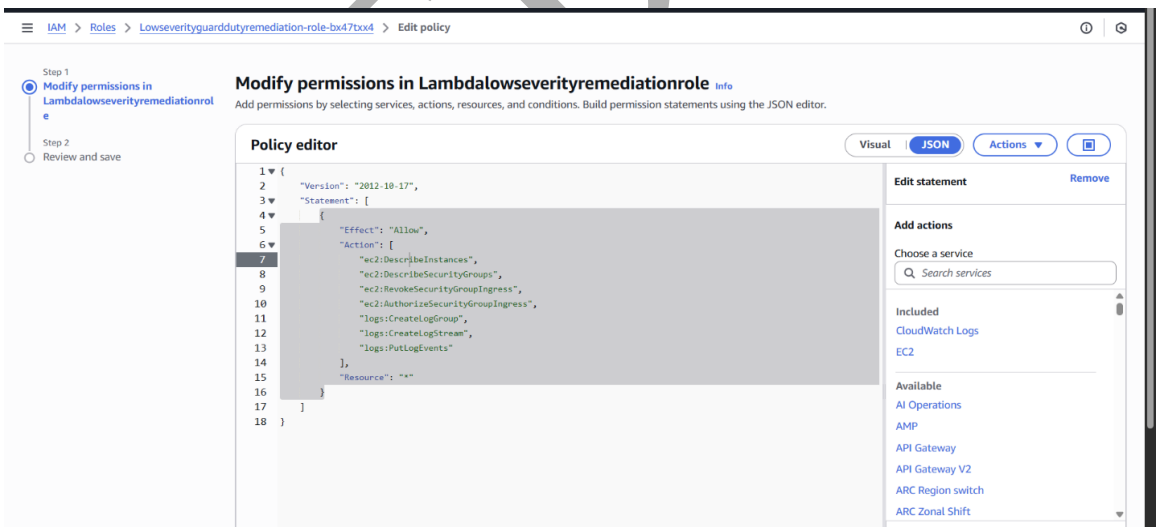
Sample findings were generated to validate alert delivery.

5. Automated Remediation for Low-Severity Findings

I implemented automated remediation for low-severity findings using AWS Lambda.



A dedicated IAM role was created with permissions to modify tagged security groups.



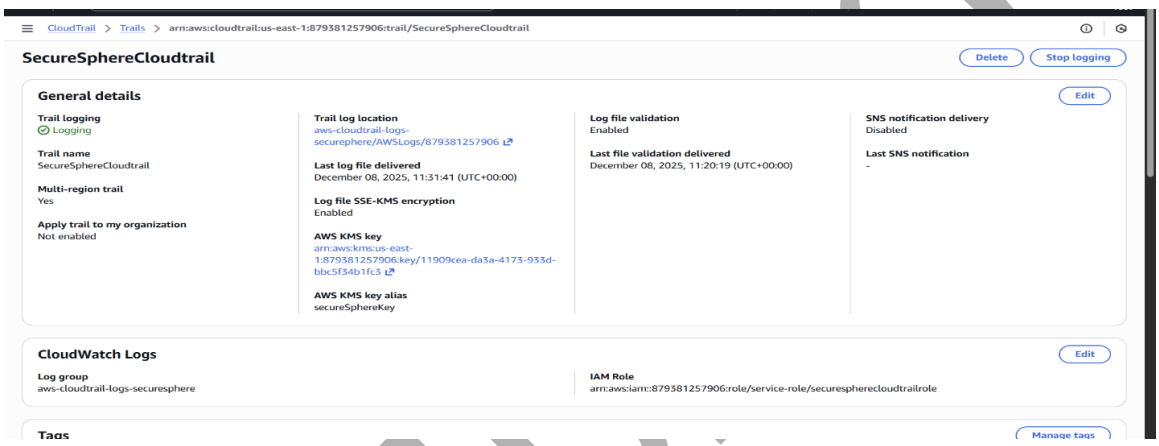
In production environments, permissions would be restricted to specific ARNs and tagged resources to enforce least privilege.

A Lambda function was deployed and triggered using EventBridge when low-severity findings occurred.

6. CloudTrail and API Monitoring

I configured a centralized CloudTrail trail to capture all AWS API activity.

Logs were forwarded to CloudWatch Logs for near real-time analysis.



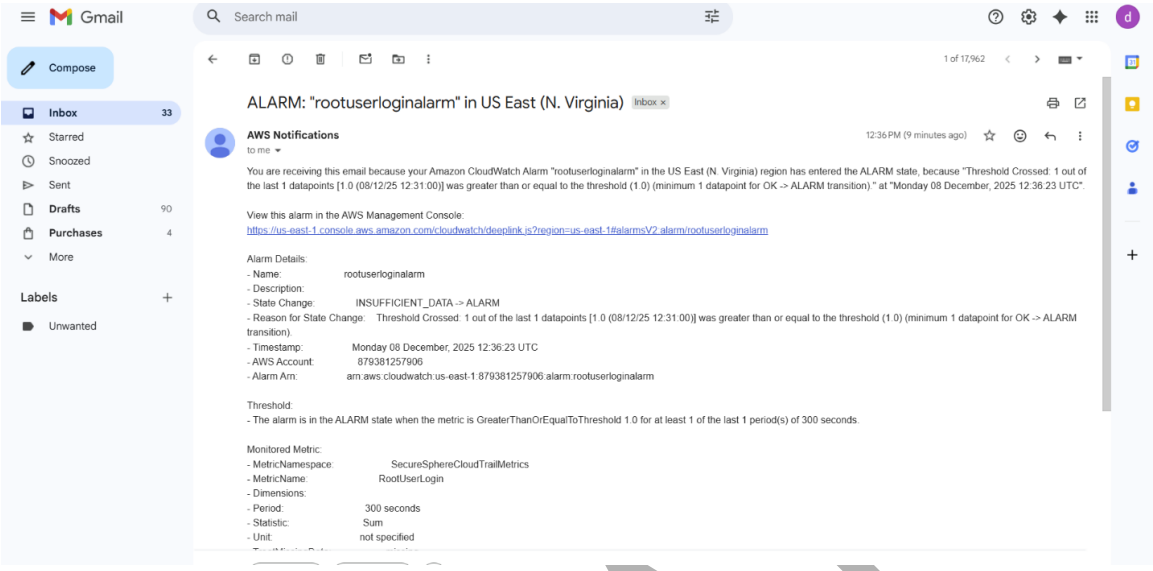
SNS notifications were configured to alert the security team when suspicious events were detected.

7. CloudWatch Metric Filters and Alarms

Metric filters were created for critical security events, including:

- Unauthorized API calls
- Console login without MFA
- Root account usage
- IAM policy changes
- CloudTrail configuration changes
- AWS Config changes
- S3 bucket policy changes

Each metric was linked to a CloudWatch alarm and SNS notification.



Manual test events were generated to validate metric activation and alert delivery.

8. Security Hub Integration

AWS Security Hub was enabled to aggregate findings from multiple services, including:

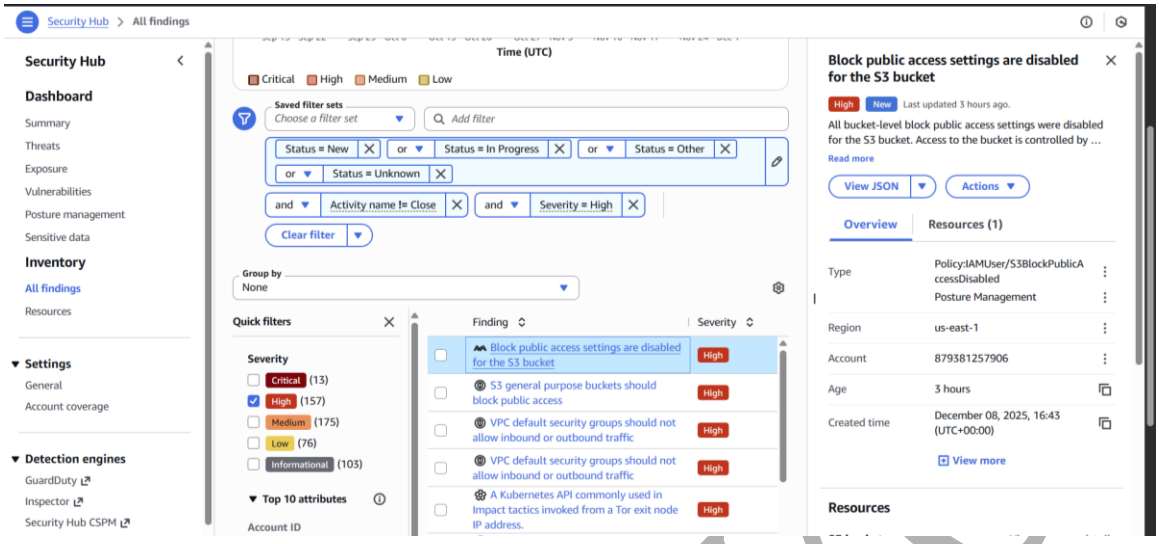
- AWS Config
- GuardDuty
- Inspector
- Macie

Security standards were reviewed and findings were prioritized based on risk level.

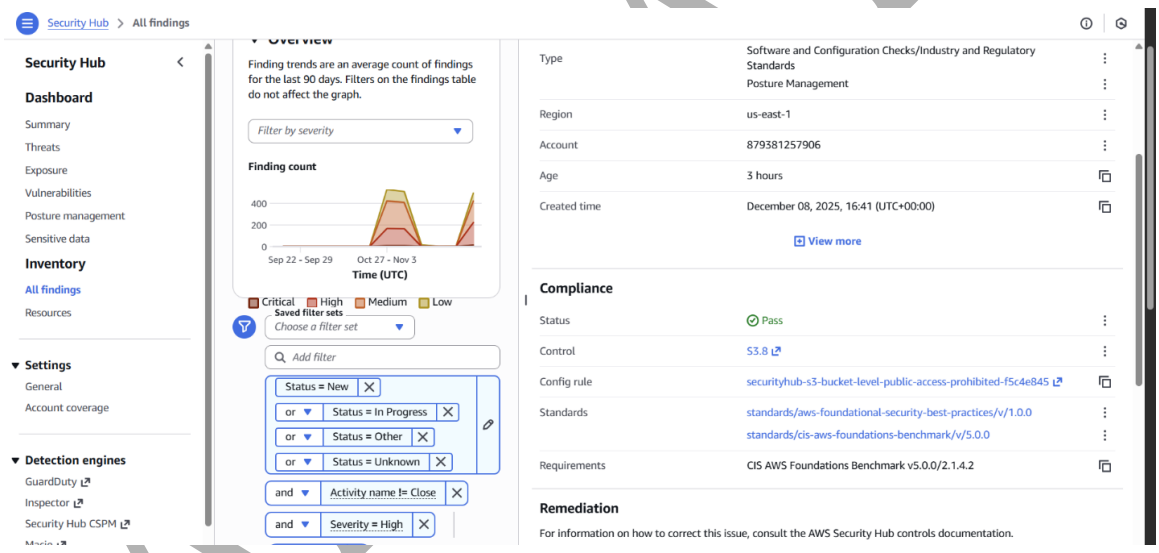
9. S3 Security Monitoring and Remediation

A test S3 bucket was created with public access enabled to validate detection capabilities.

GuardDuty and Security Hub detected the misconfiguration.



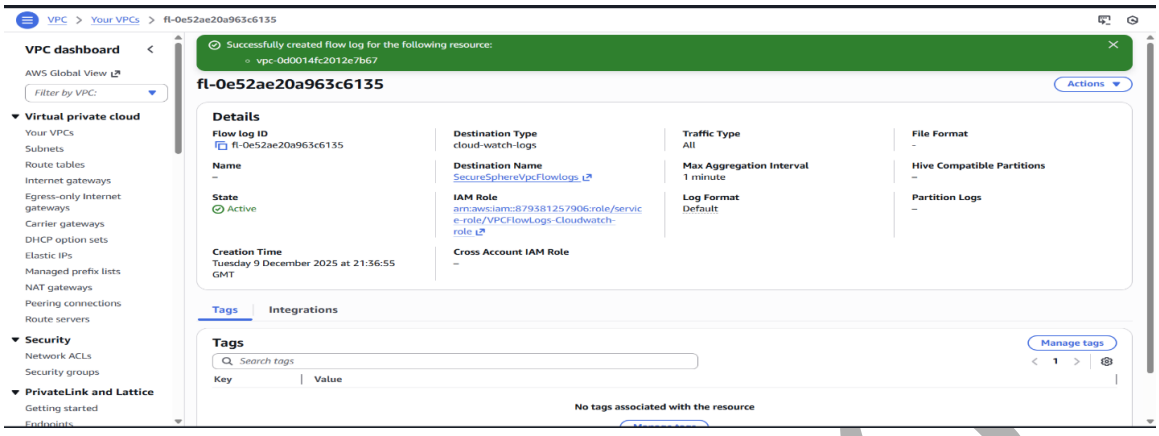
Public access was removed to remediate the high severity finding.



Server access logging was enabled to improve audit visibility.

10. Network Security and VPC Flow Logs

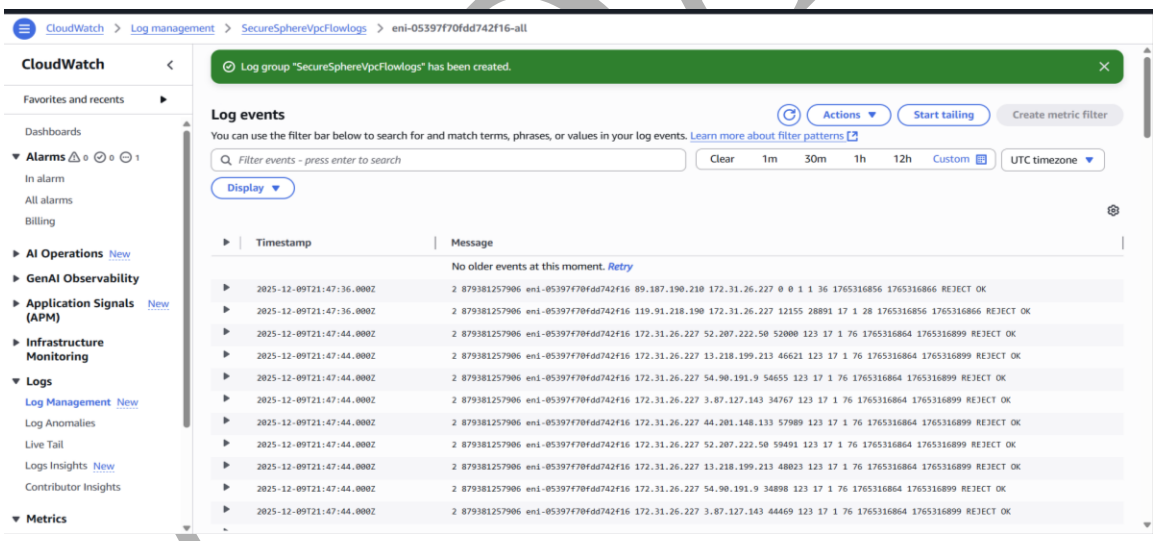
VPC Flow Logs were enabled for all production VPCs.



Logs were sent to CloudWatch Log Groups for monitoring and analysis.

An EC2 instance was used to generate network traffic for validation.

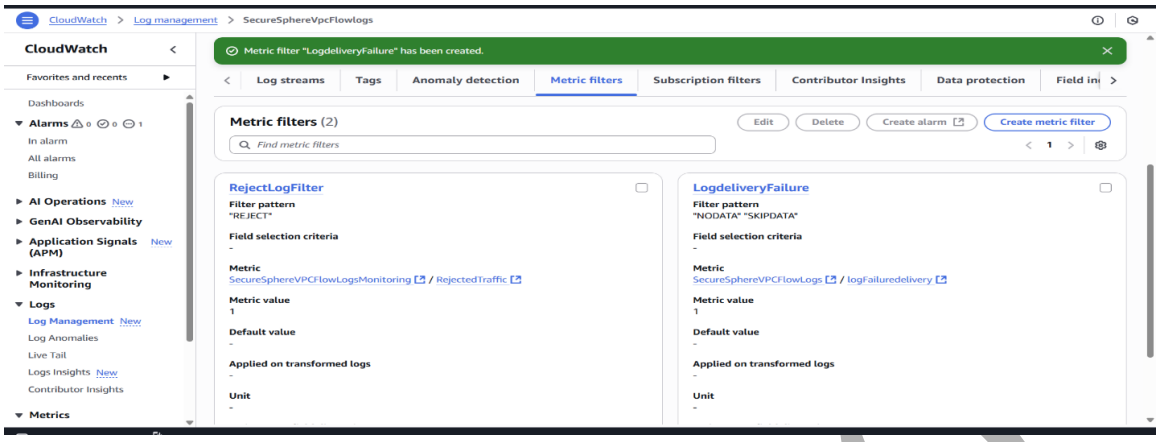
Both ACCEPT and REJECT traffic was verified in log streams.



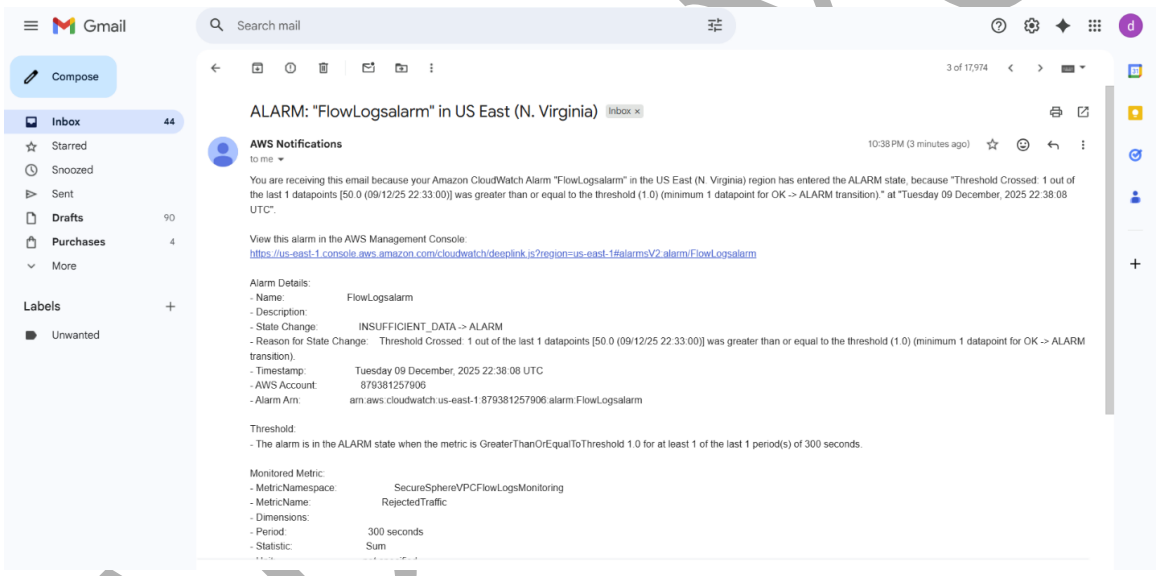
11. Network Monitoring Alerts

Metric filters were created for:

- Rejected traffic events
- Log delivery failures



CloudWatch alarms were configured to notify the security team of network anomalies.



12. Documentation and Governance

All configurations, alerting workflows, and remediation procedures were documented.

This documentation supported audits, incident response, and knowledge transfer.

13. Outcomes and Impact

This implementation delivered the following improvements:

- Centralized threat detection
- Faster incident response
- Improved network visibility
- Automated remediation
- Reduced security misconfigurations
- Enhanced compliance monitoring

14. Conclusion

This monitoring and threat detection framework provides comprehensive visibility into cloud security risks.

Through centralized logging, automated response, and continuous analysis, the environment supports secure and resilient operations.