🚨 GuardDuty Root Account Misuse Alerting System

A lightweight security automation solution that detects root account activity using Amazon GuardDuty and instantly alerts administrators via Amazon SNS using a Lambda-based notification workflow.

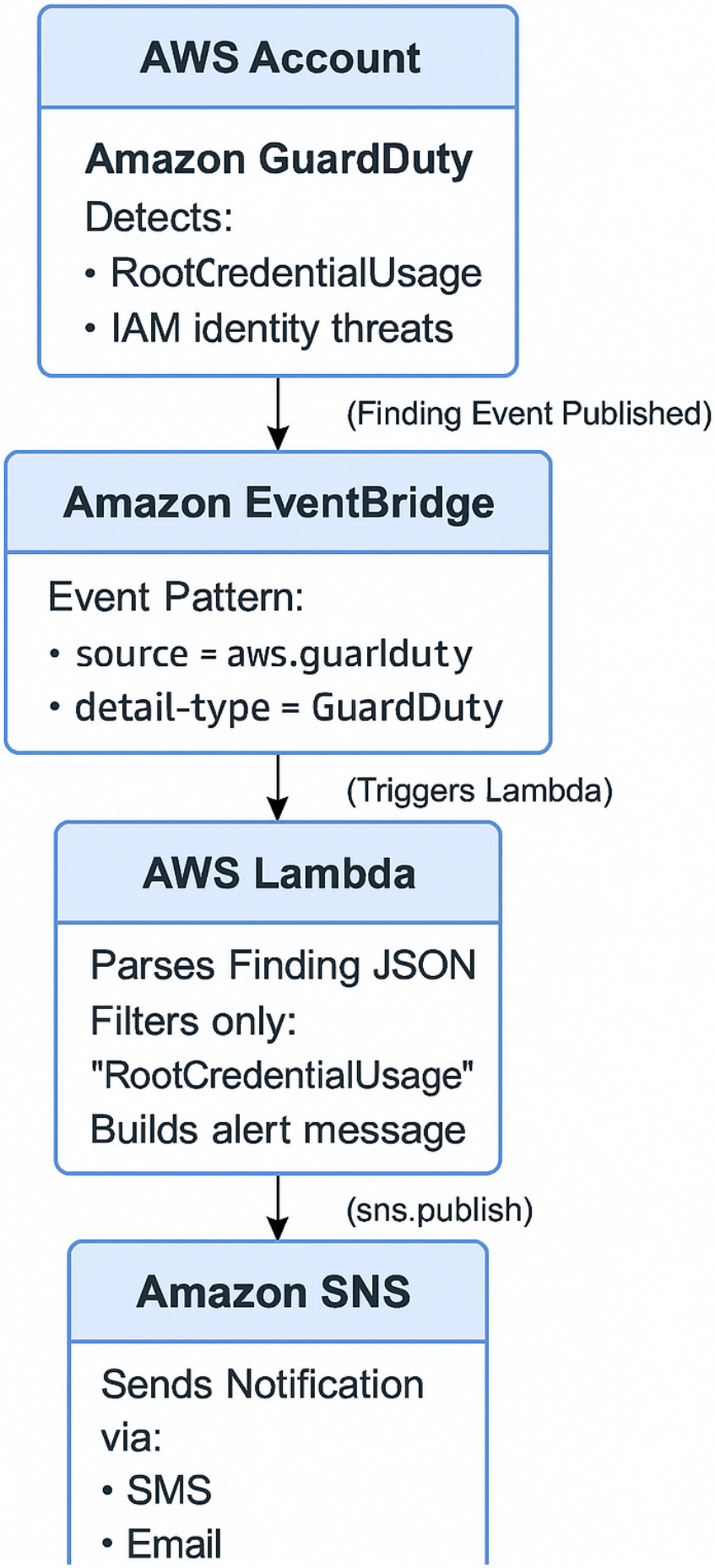
🔧 **Services Used**

- **Amazon GuardDuty**
- **Amazon EventBridge**
- **AWS Lambda (Python)**
- **Amazon SNS**

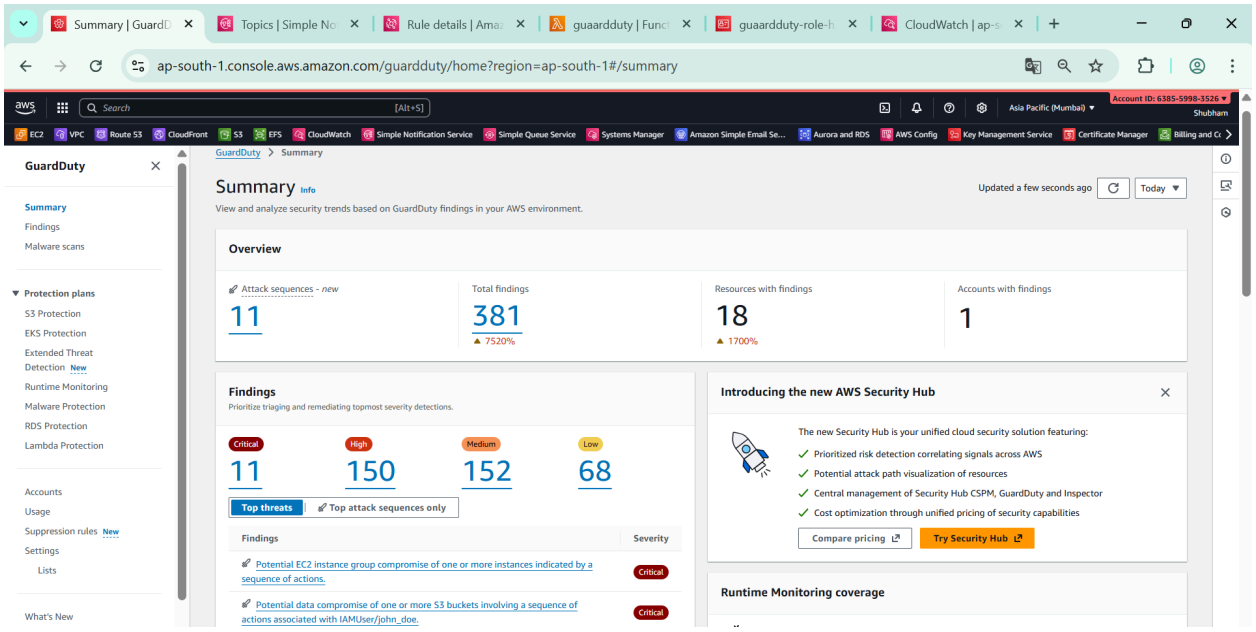
Project Highlights

- **Implemented real-time detection of AWS root account usage using Amazon GuardDuty.**
- **Designed an EventBridge → Lambda → SNS automation pipeline to immediately alert administrators.**

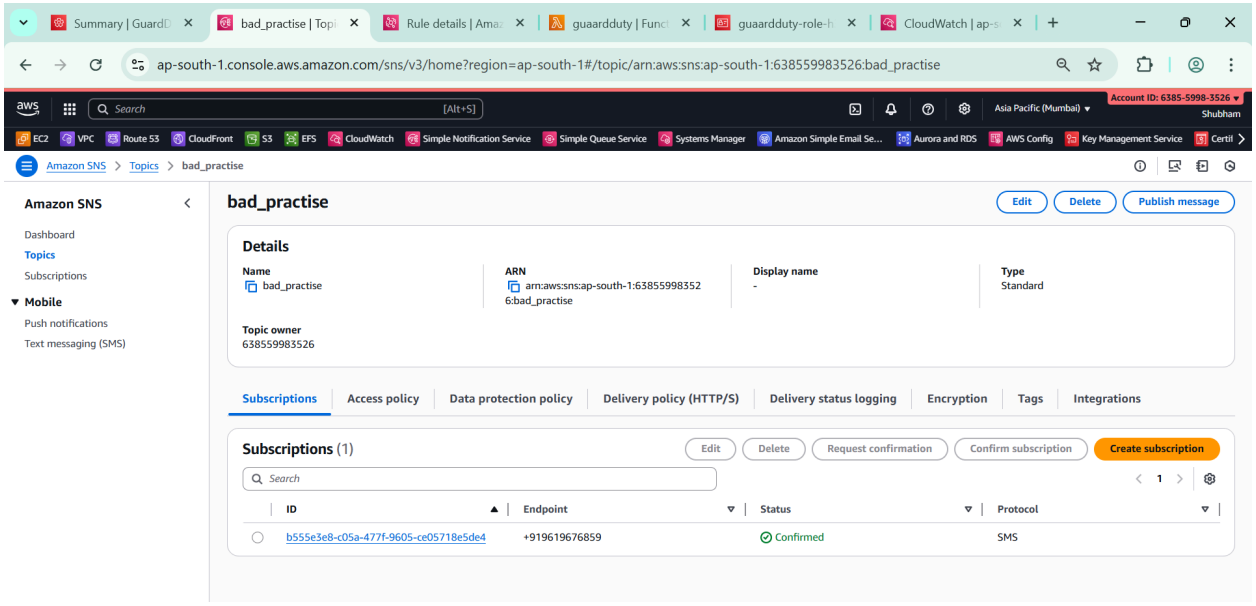
- **Built a serverless, cost-effective, fully automated security workflow with no infrastructure to maintain.**
- **Parsed and enriched GuardDuty findings to extract details like:**
    - **Finding type**
    - **Source IP**
    - **Region**
    - **Event timestamp**
- **Delivered alerts via SMS and email using Amazon SNS for rapid incident response.**
- **Built event pattern filters to match only high-risk identity misuse findings.**
- **Ensured compliance with AWS Security Best Practices by notifying stakeholders about dangerous root usage.**
- **Validated system behavior using sample GuardDuty findings and EventBridge test events.**


AWS Account
Amazon GuardDuty
Detects:
• RootCredentialUsage
• IAM identity threats
(Finding Event Published)
Amazon EventBridge
Event Pattern:
• source = aws.guarlduty
• detail-type = GuardDuty
(Triggers Lambda)
AWS Lambda
Parses Finding JSON
Filters only:
"RootCredentialUsage"
Builds alert message
(sns.publish)
Amazon SNS
Sends Notification
via:
• SMS
• Email

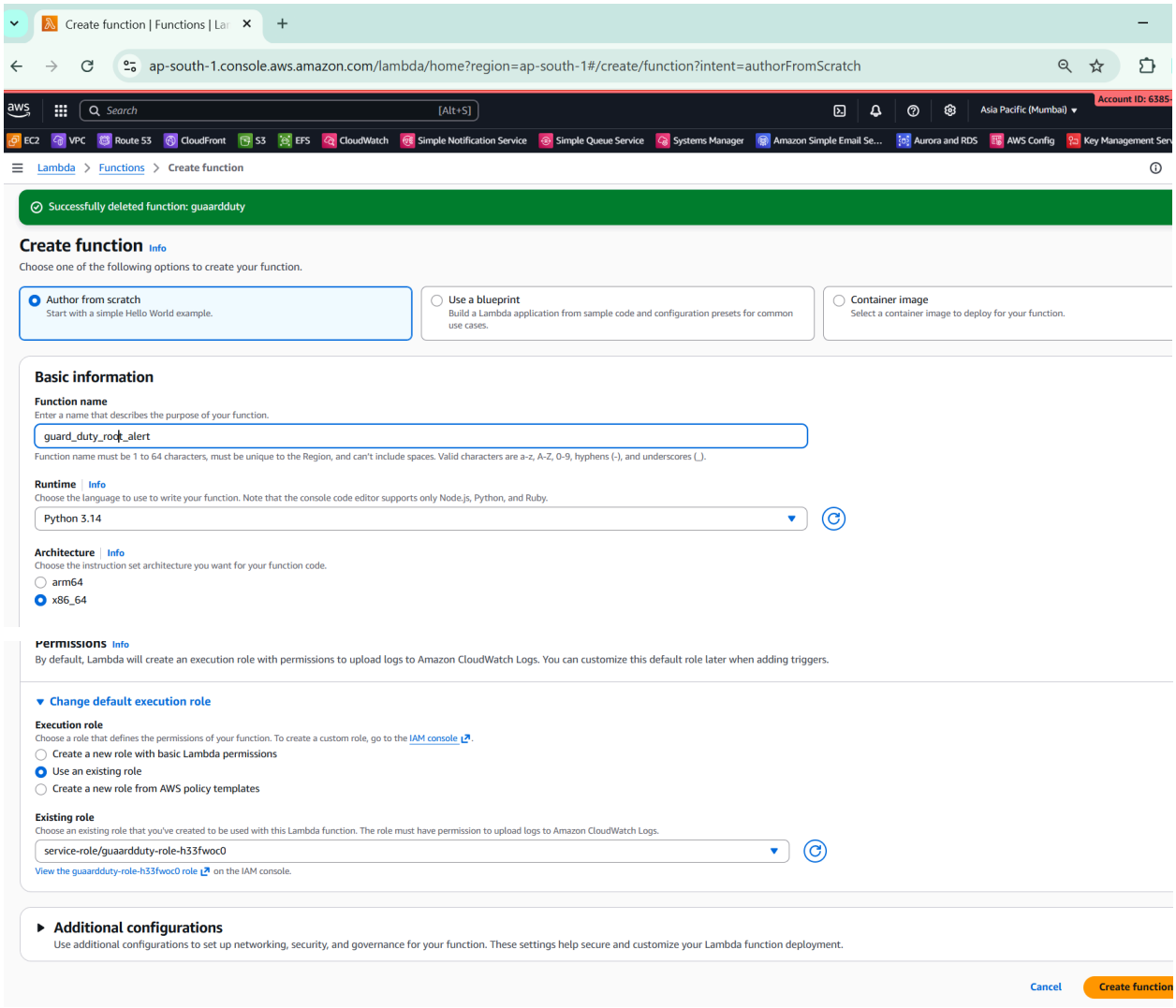
Step 1) We will enable Guard duty for our AWS account.



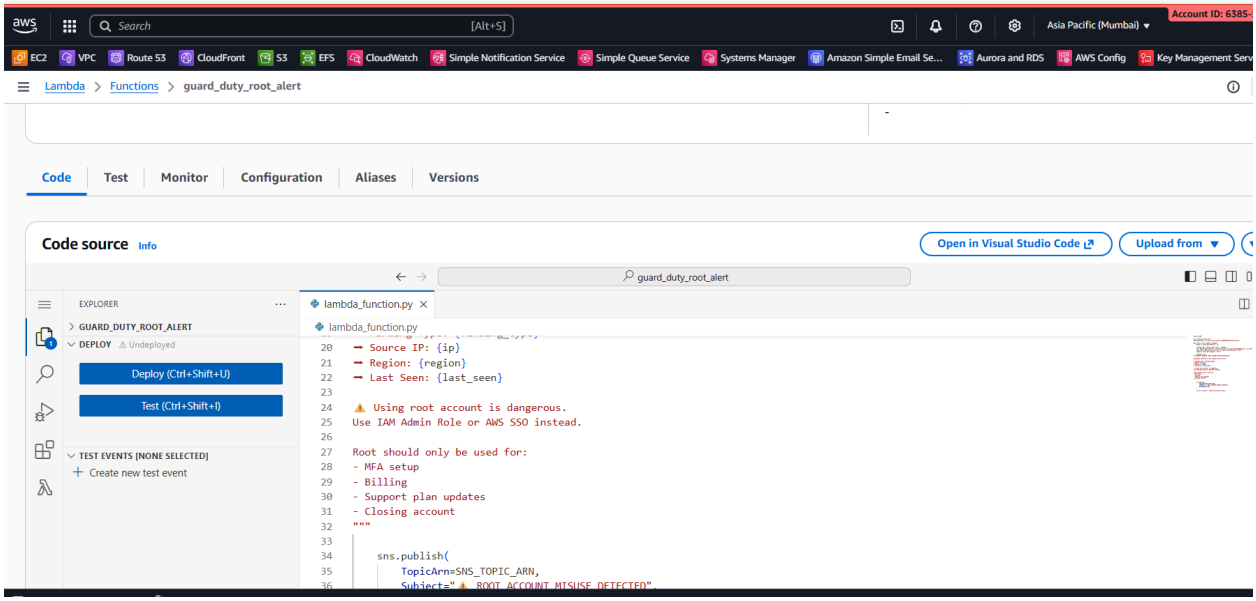
Step 2) We will create SNS topic and add subscriptions to that topic



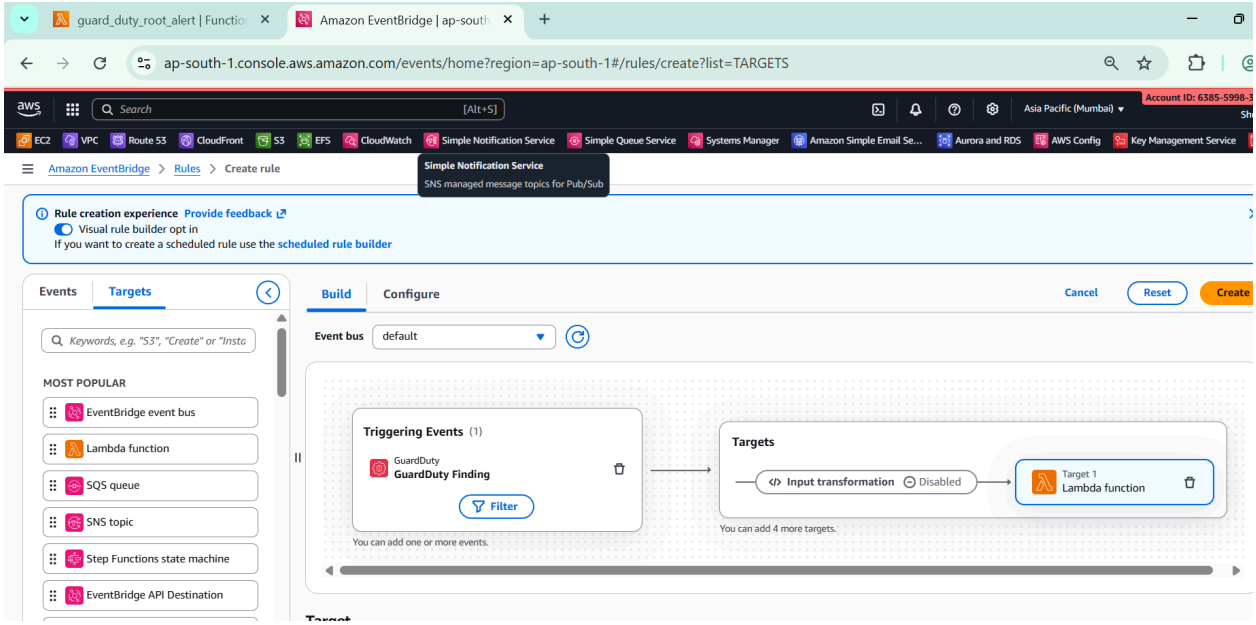
Step 3) Now we will create Lambda function to automate trigger on any findings based on root account.



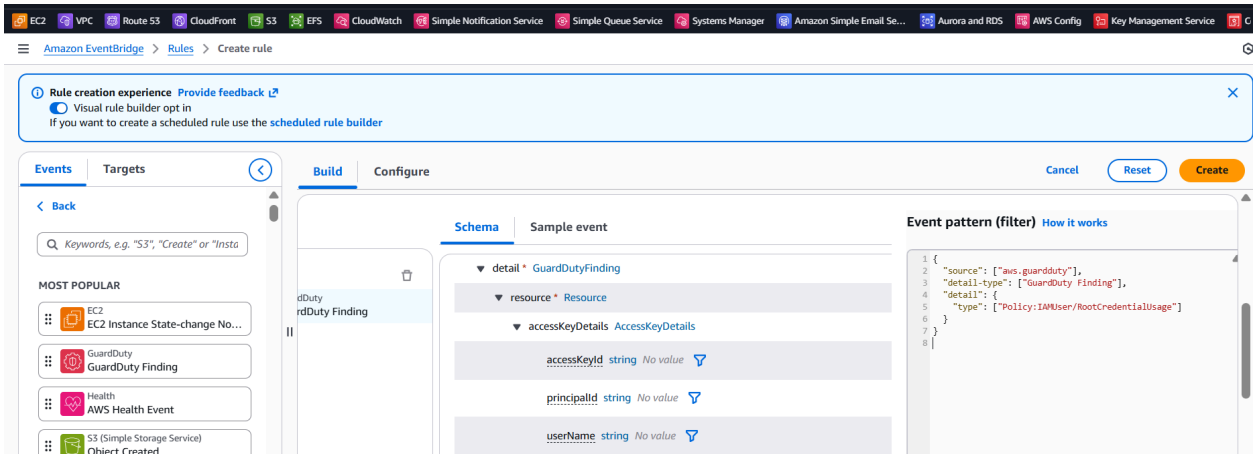
Step 4) We will plac the code python.



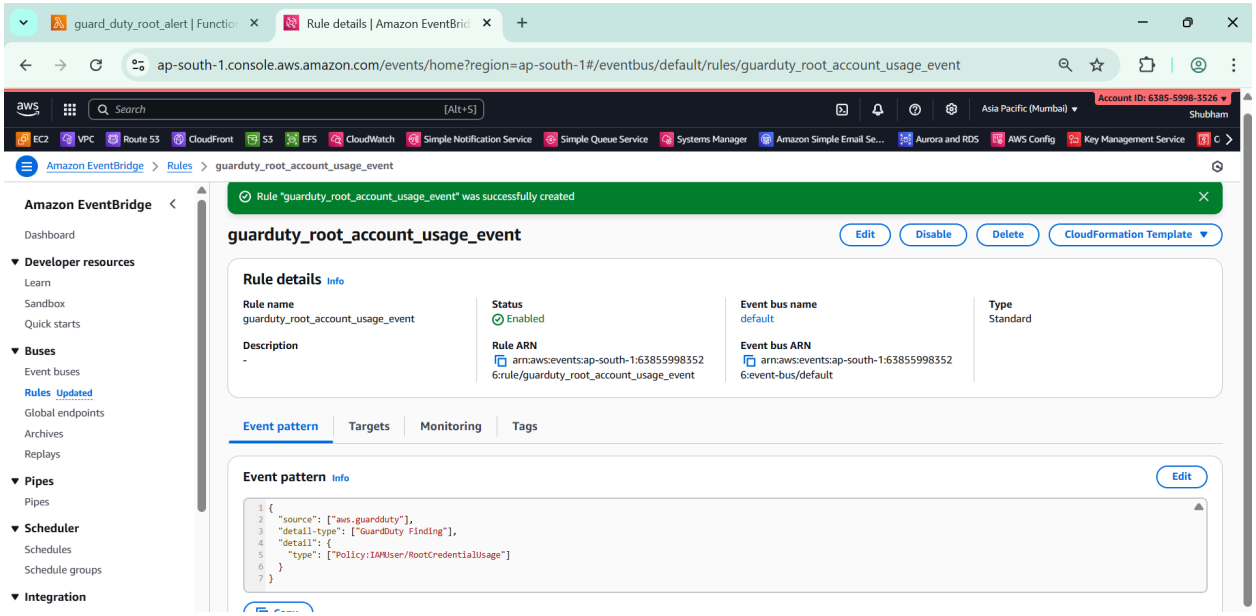
Step 5) We will now create Aws eventbridge rule and add event trigger and target as lambda function.



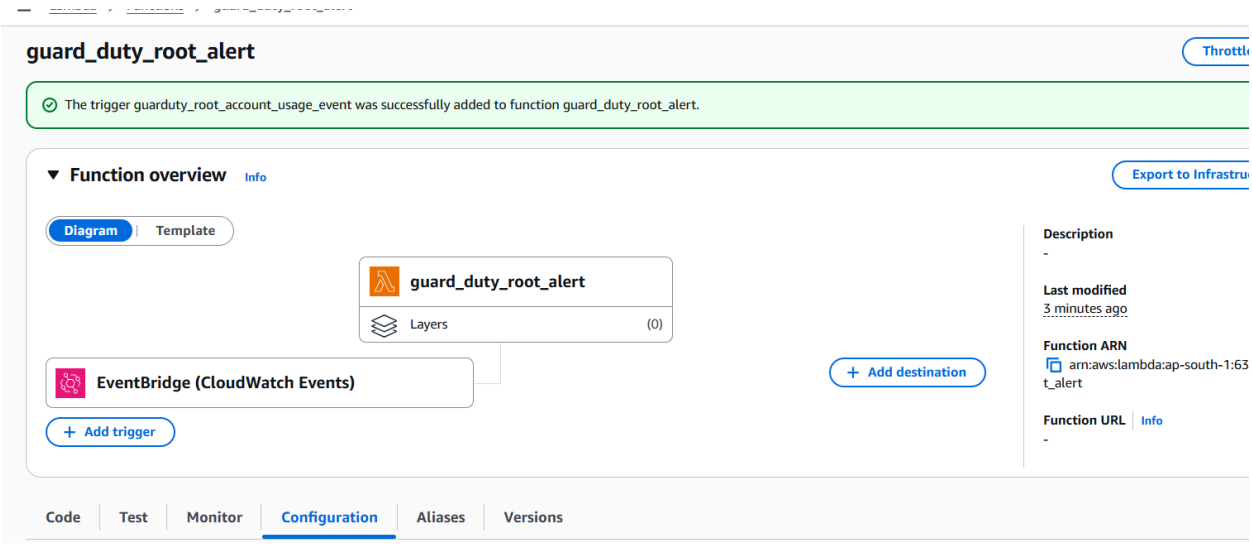
Step 6) We will enter event rule filter now.



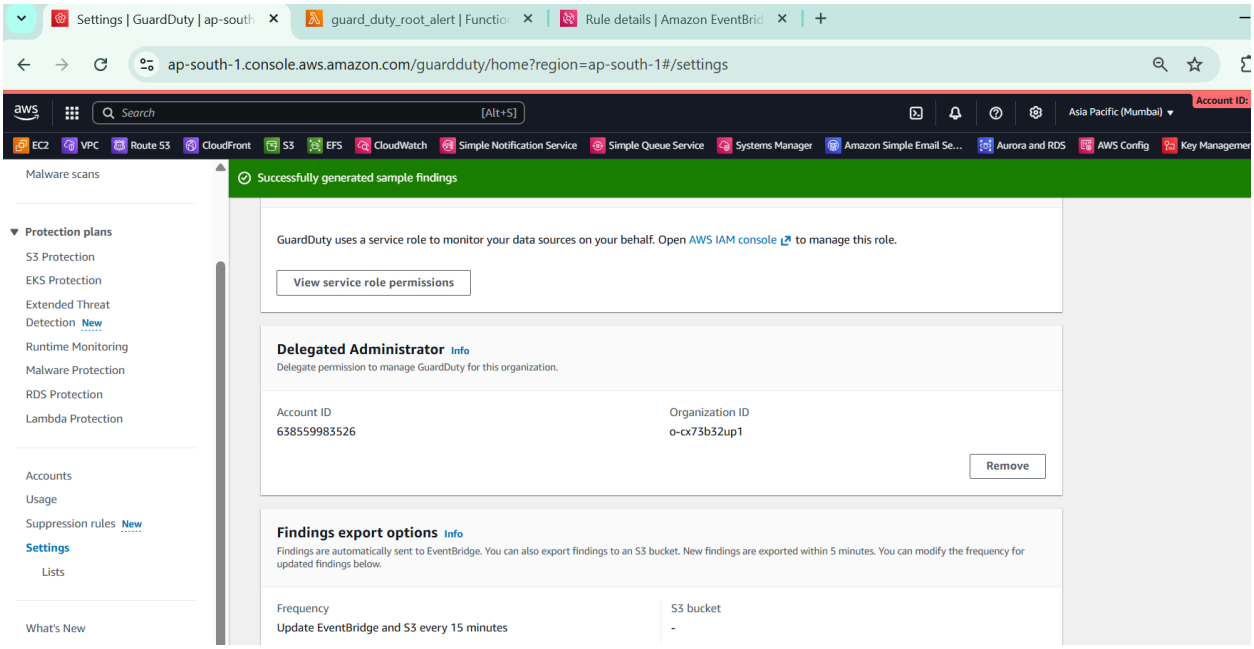
Step 7) We can see the rule is created successfully.



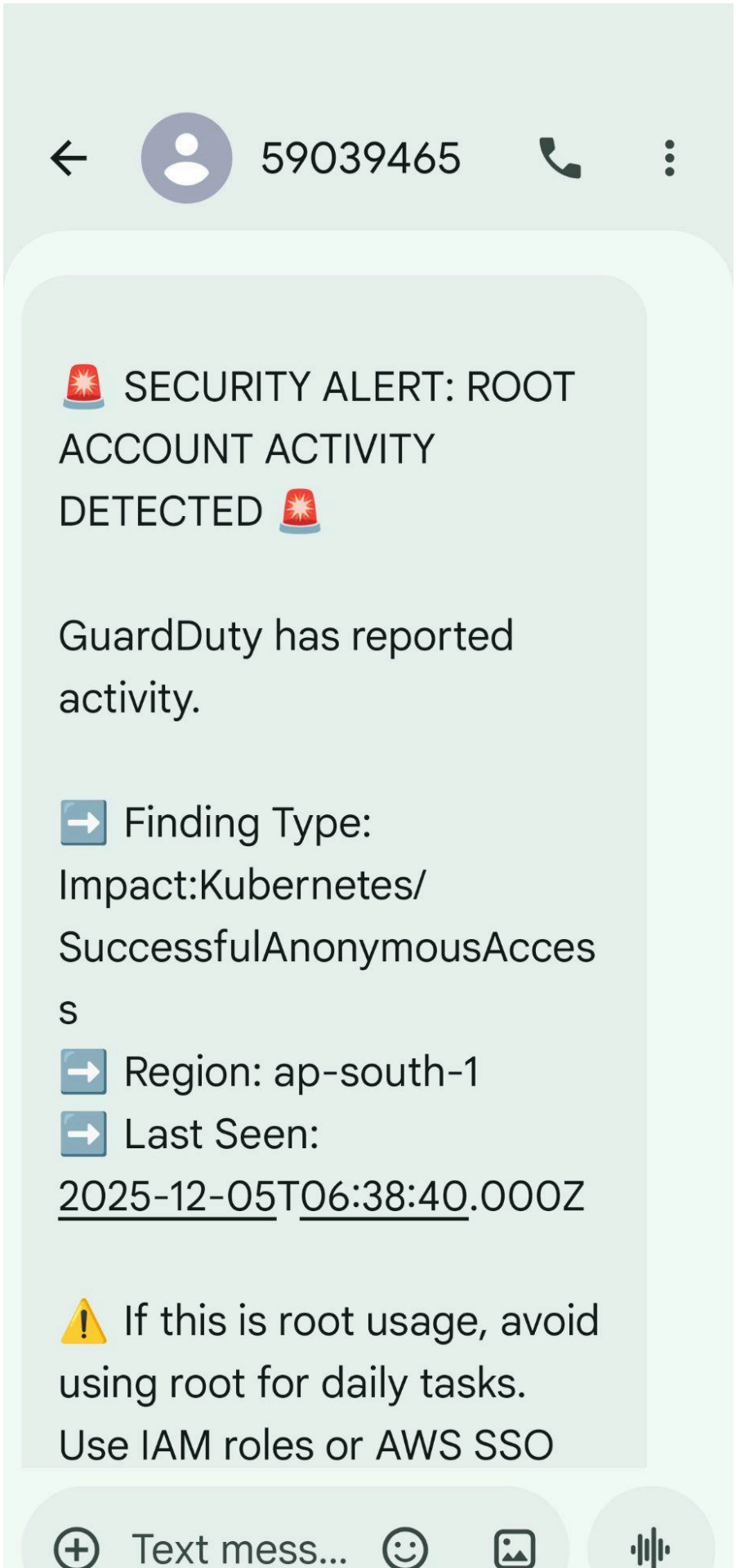
Step 8) We will add this event as trigger for Lambda function as show below



Step 9) Now we will run generate sample findings in aws guardduty to simulate fake threats.



Step 10) We can see we received notification once threat detected.


59039465
🚨 SECURITY ALERT: ROOT ACCOUNT ACTIVITY DETECTED 🚨
GuardDuty has reported activity.
➡️ Finding Type: Impact:Kubernetes/SuccessfulAnonymousAccess
➡️ Region: ap-south-1
➡️ Last Seen: 2025-12-05T06:38:40.000Z
⚠️ If this is root usage, avoid using root for daily tasks.
Use IAM roles or AWS SSO
Text mess...

This project successfully demonstrates an automated security response workflow using AWS GuardDuty, EventBridge, Lambda, and SNS. By detecting root account misuse in real time and instantly alerting administrators through SMS/Email, it significantly strengthens the security posture of the AWS account.
This automation ensures faster incident response, reduces human error, and enforces best practices for identity and access management across cloud environments.