**Task 22: Cost Optimization Through Automated Resource Management in AWS**

**Objective:**

The purpose of this runbook is to outline the steps to automate the lifecycle management of AWS resources, including EC2 instances, EBS volumes, and snapshots, while leveraging CloudWatch, Lambda, and SNS to optimize costs.

**Components Used:**

**EC2 Instance**: A virtual machine in AWS that you create for running applications. I have launched (Using CFT) an EC2 instance as part of the process.

**EBS Volume**: Elastic Block Store (EBS) provides persistent block storage for EC2 instances. I've created and attached an EBS volume to my EC2 instance (Using CFT) to store data.

**Snapshot**: A point-in-time copy of an EBS volume. I created a snapshot of my EBS volume to preserve its state, ensuring that the data is backed up and can be restored or replicated.

**Cloud Formation**: An AWS service used for automating the deployment and management of infrastructure. I've used Cloud Formation templates to create and configure my EC2 instance and EBS volume.

**CloudWatch**: AWS monitoring service that collects and tracks metrics for AWS resources. I have set up a CloudWatch alarm to monitor the CPU utilization of the EC2 instance and trigger actions based on the threshold.

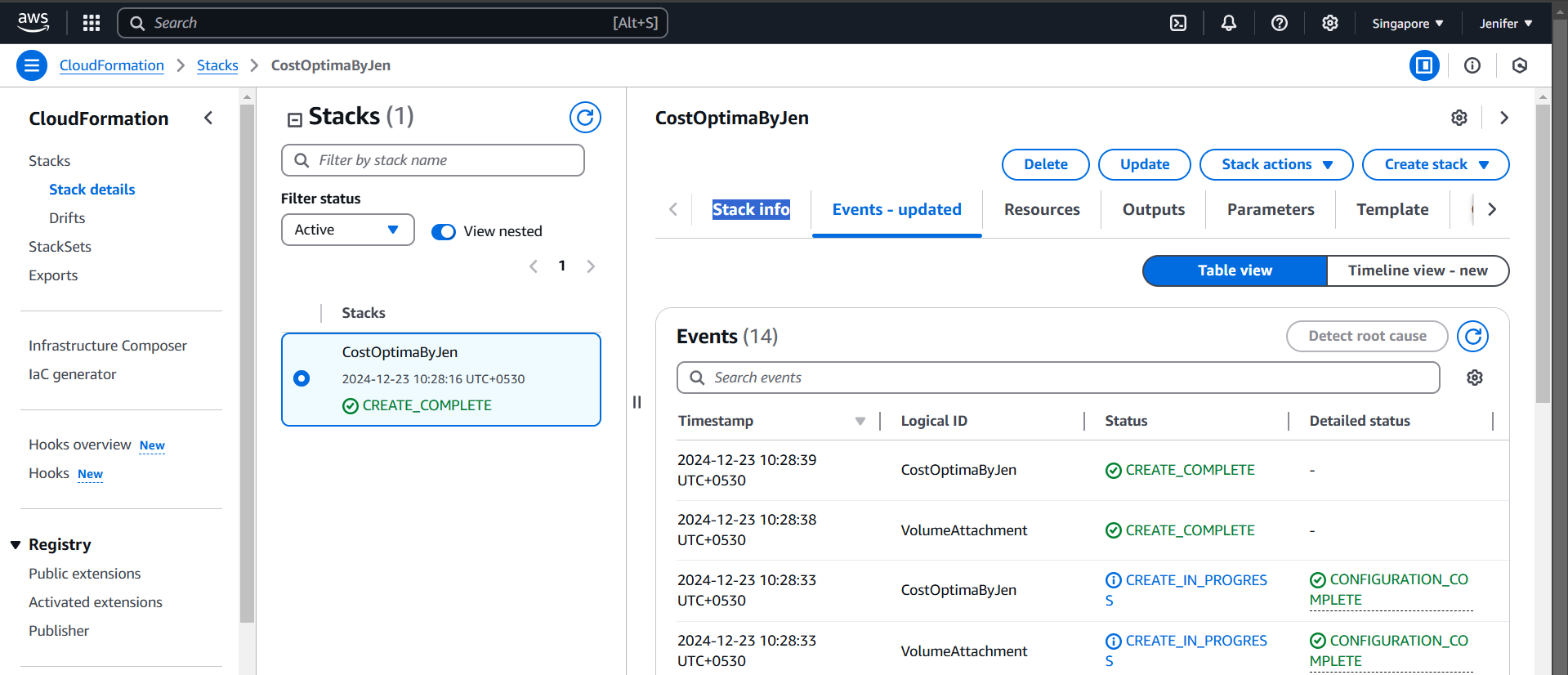
**SNS (Simple Notification Service)**: An AWS service for sending notifications. I've configured SNS to send an email when the CloudWatch alarm is triggered, as well as for Lambda function notifications.

**Lambda**: A serverless compute service that lets you run code in response to events. I've created a Lambda function to delete the EC2 instance, EBS volume, and stale SNS subscription once the CloudWatch alarm condition is met.

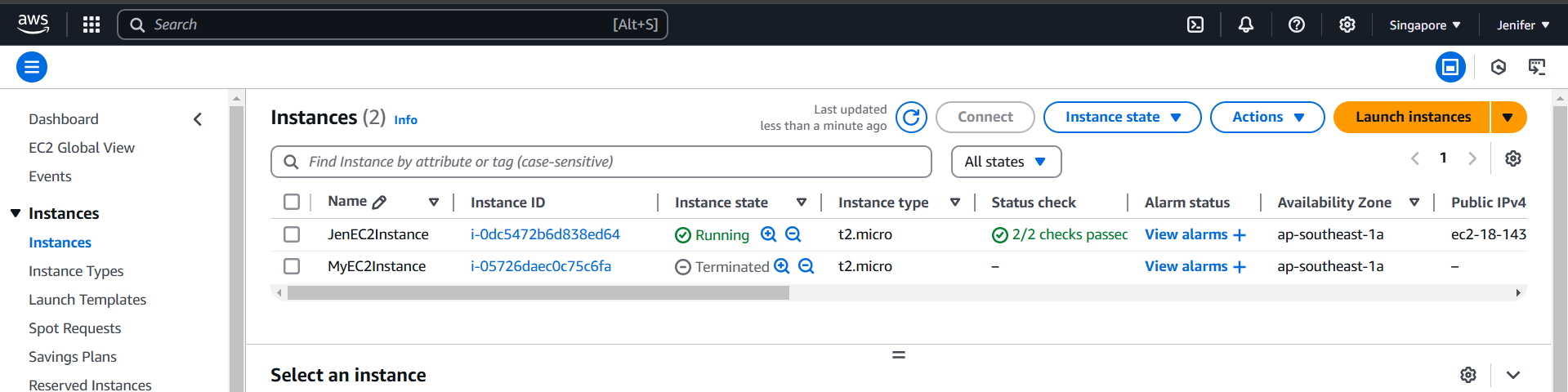
**SNS Subscription**: The mechanism that allows you to receive notifications from SNS. I've set up an email subscription for receiving alerts related to the CloudWatch alarm and Lambda function results.

**I. Launching Resources Using CFT**

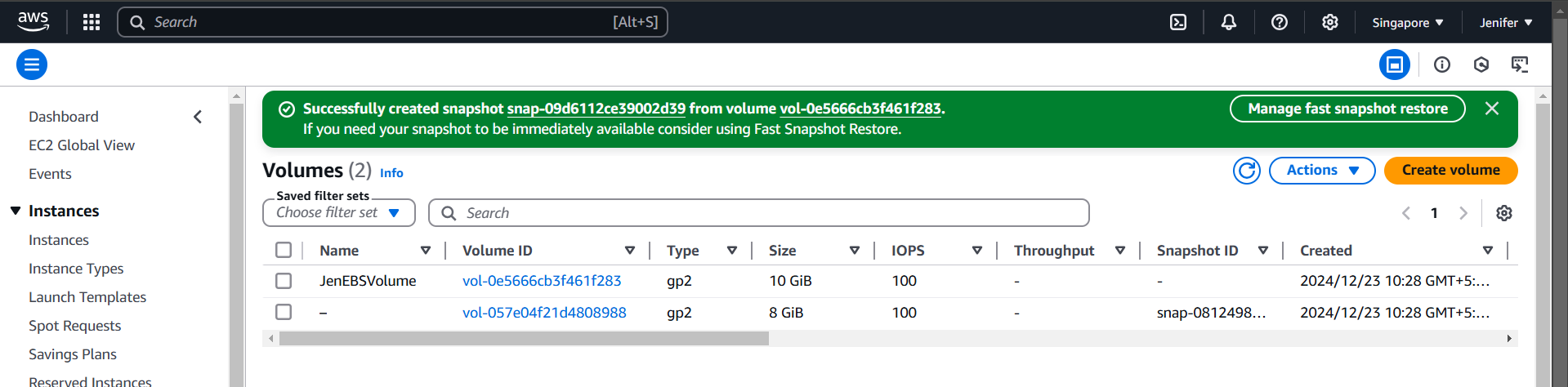
1. A Stack Created for Launching the Resources



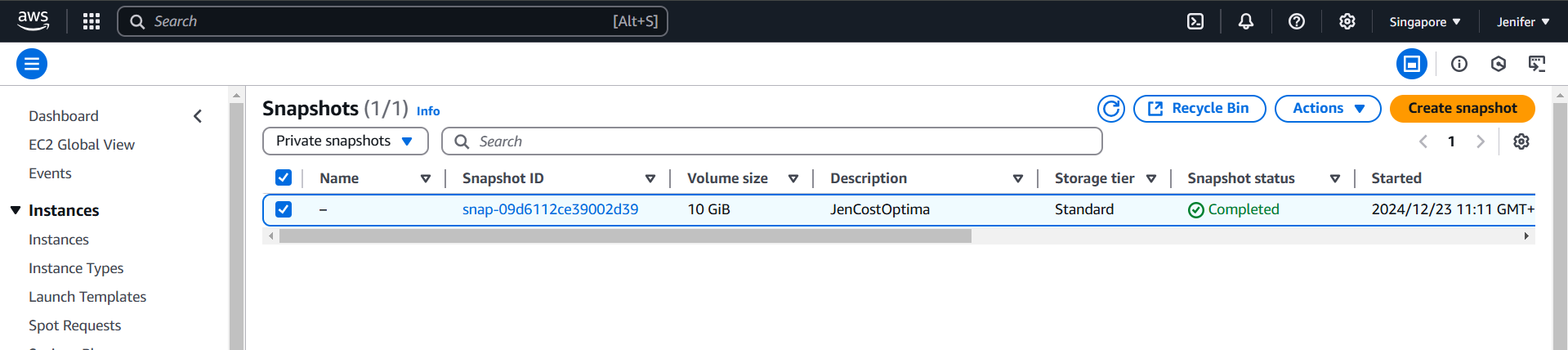
1. EC2 Instance Got Created



1. EBS Volume Got Created and Attached to the EC2 Instance

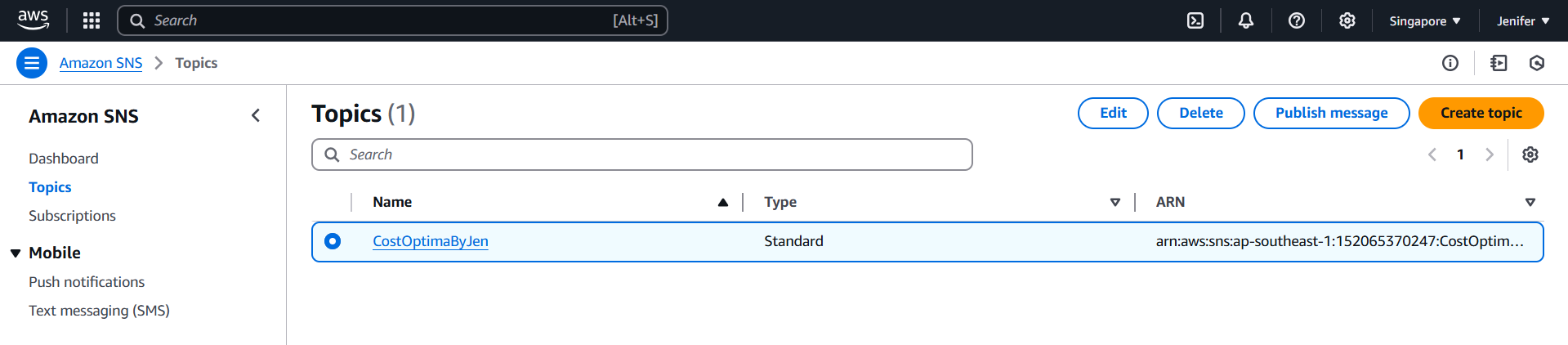


1. I have created a snapshot for the EBS Volume

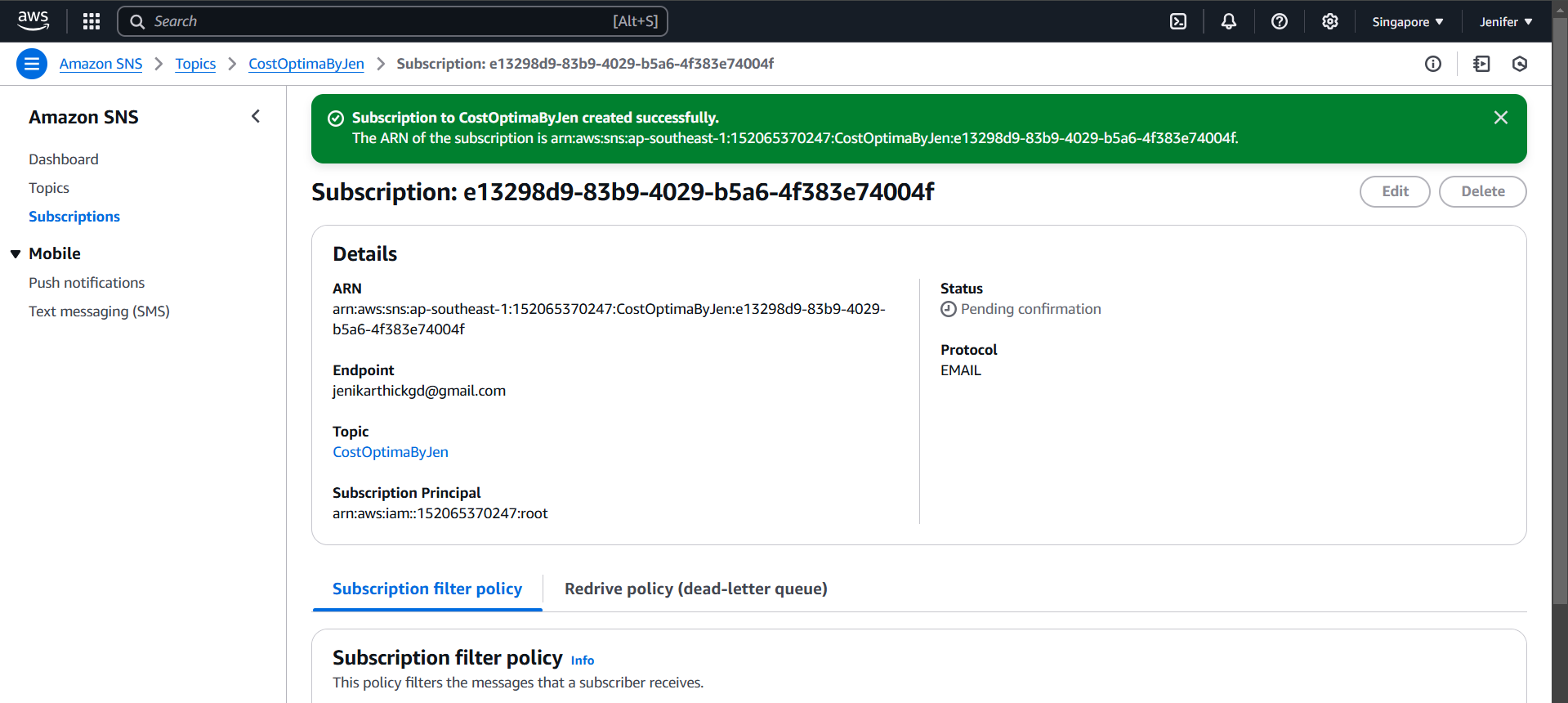


**II. SNS Configuration for Triggering Email to the Provided Mail ID**

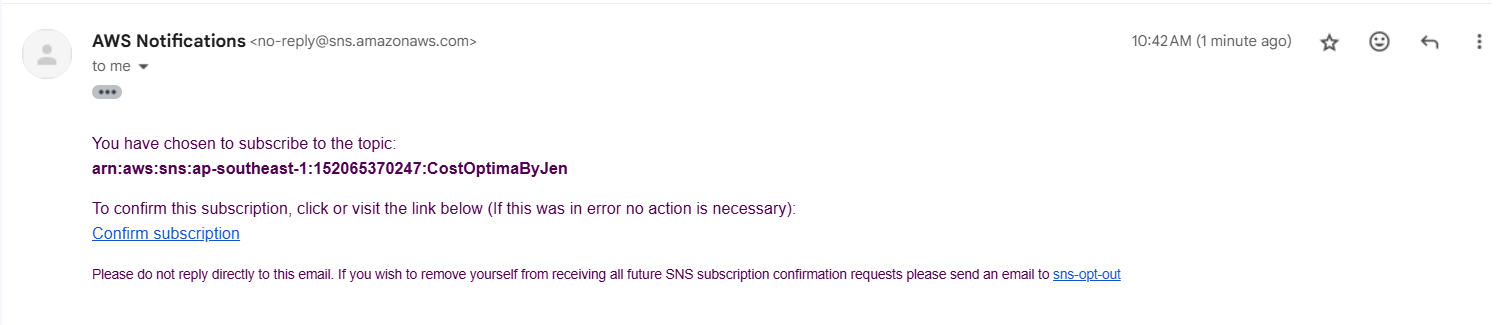
1. An SNS Topic has Created

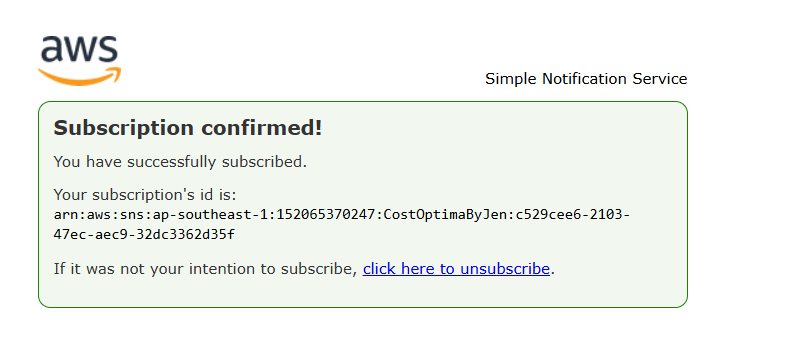


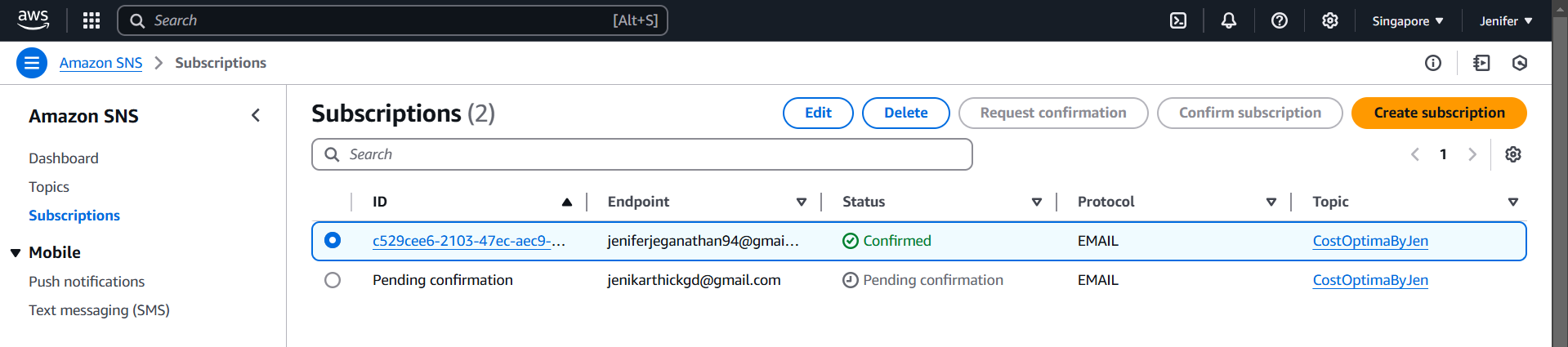
1. The Subscription for the Topic has Created



1. Subscription has confirmed

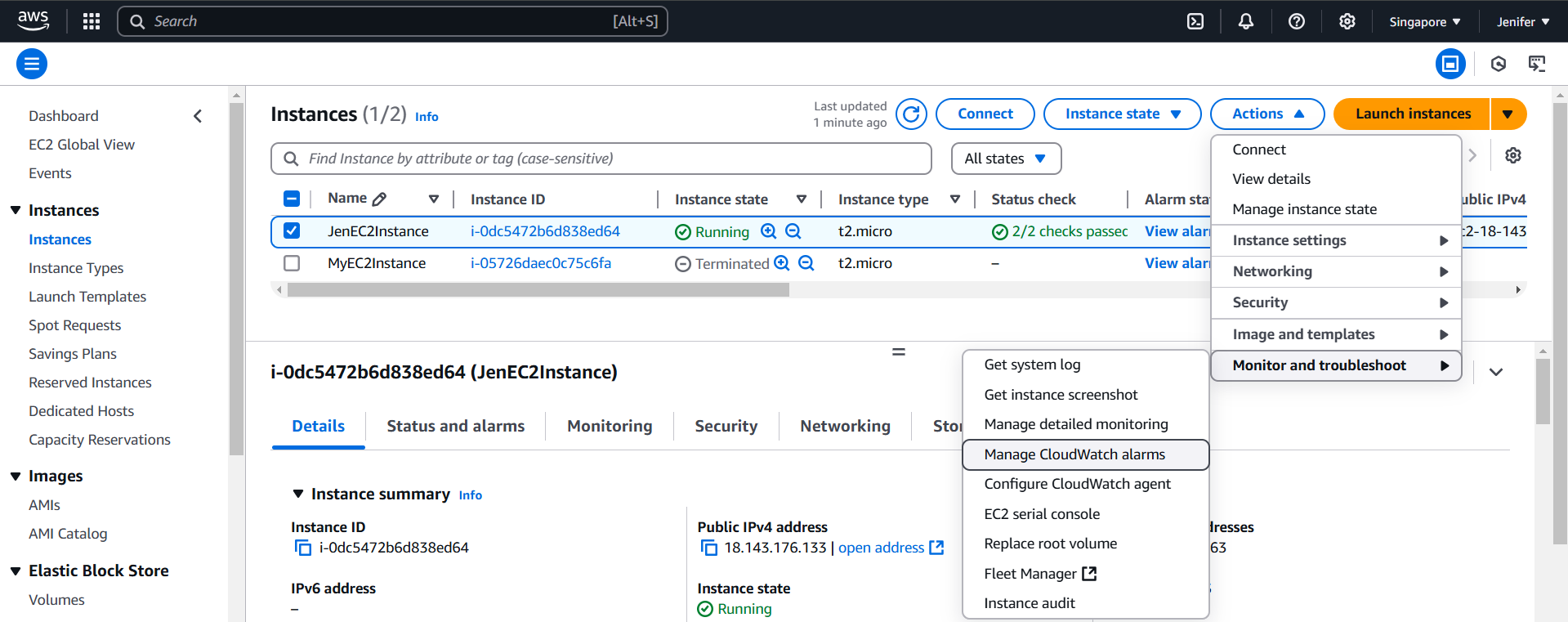




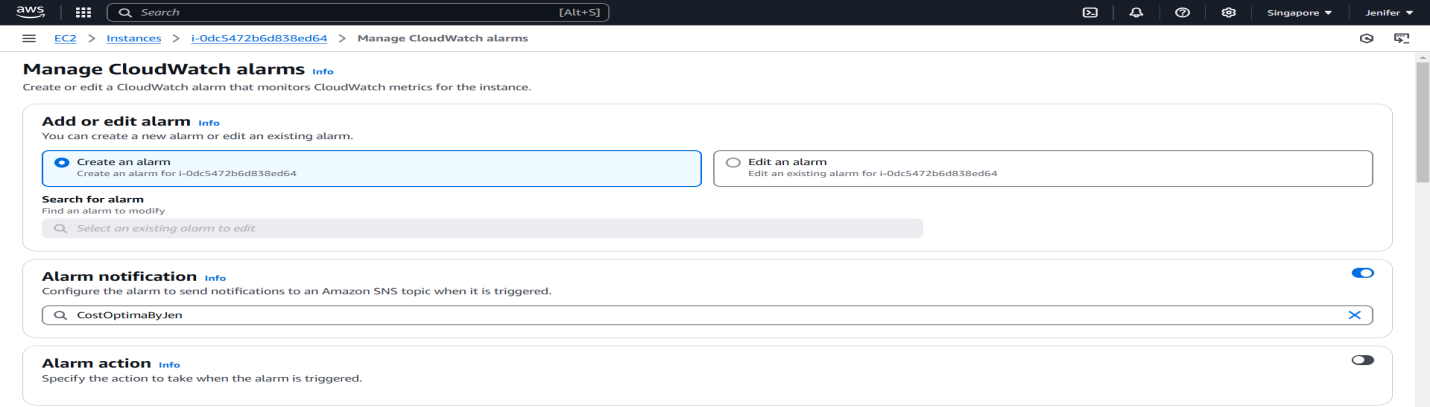


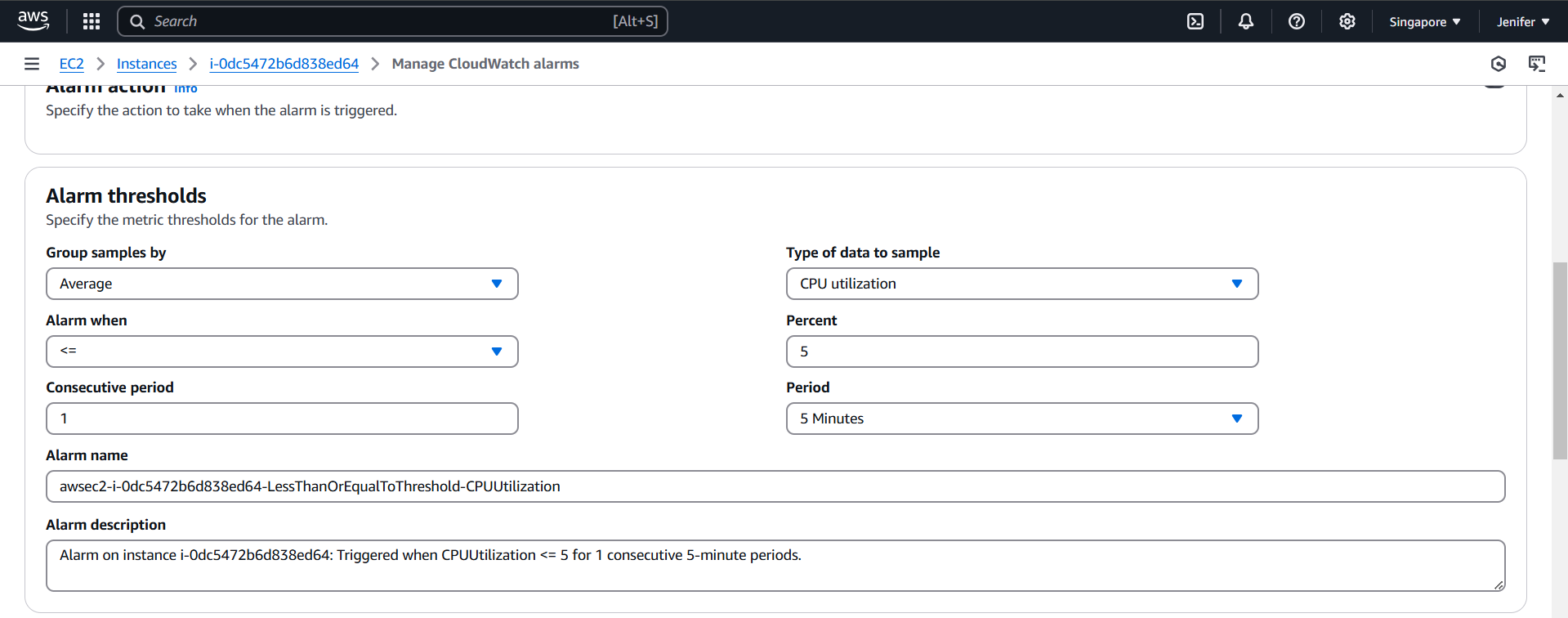
**III. Configuring CloudWatch Alarm to set the Threshold**

1. Directed to the appropriate path

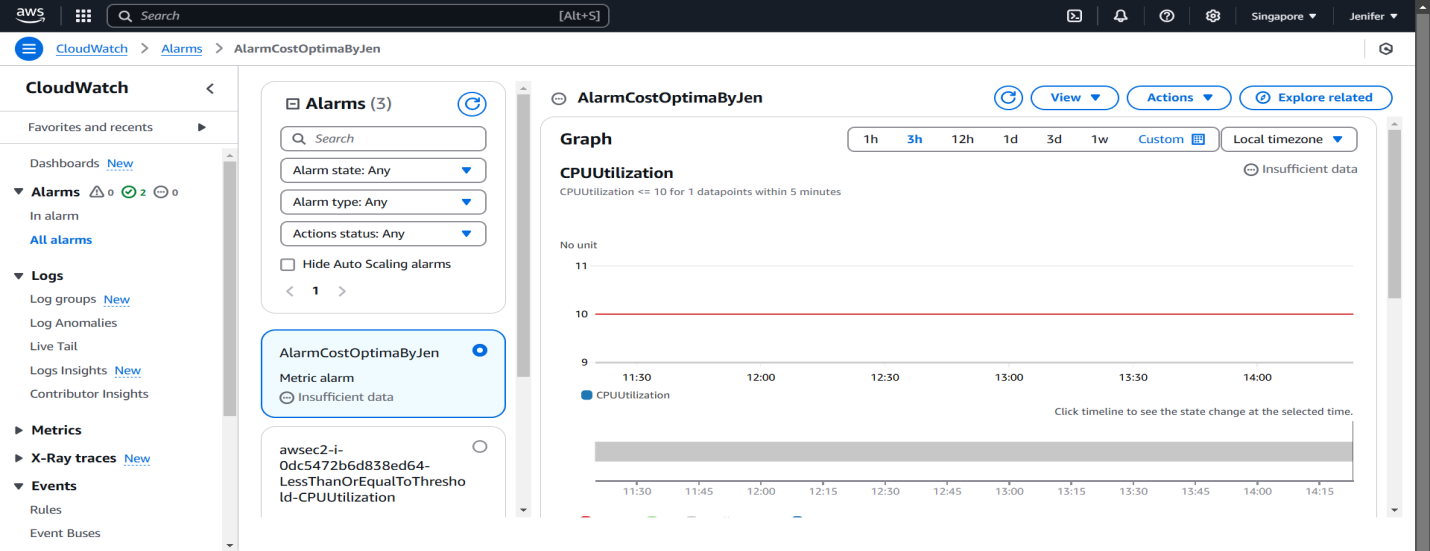


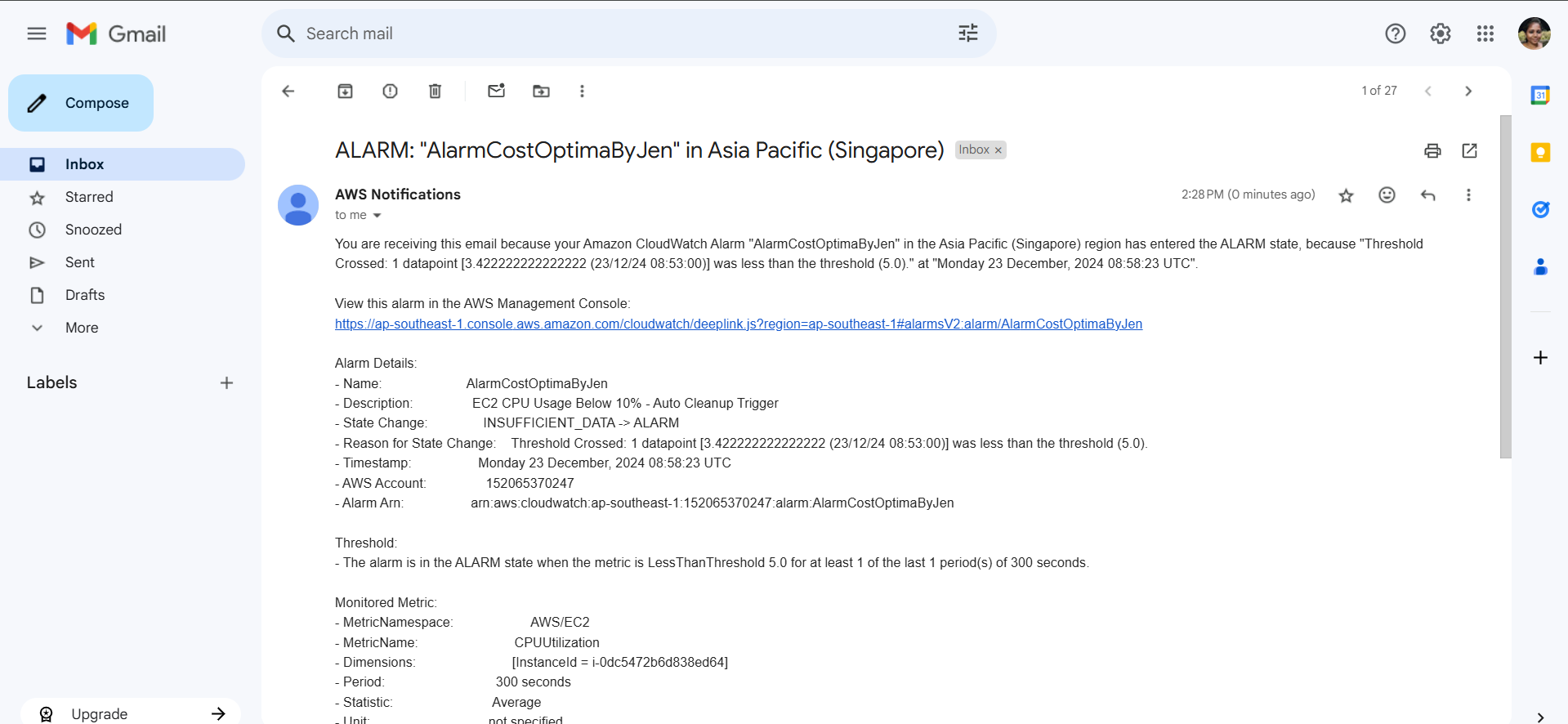
1. Setting Up the Alarm Threshold



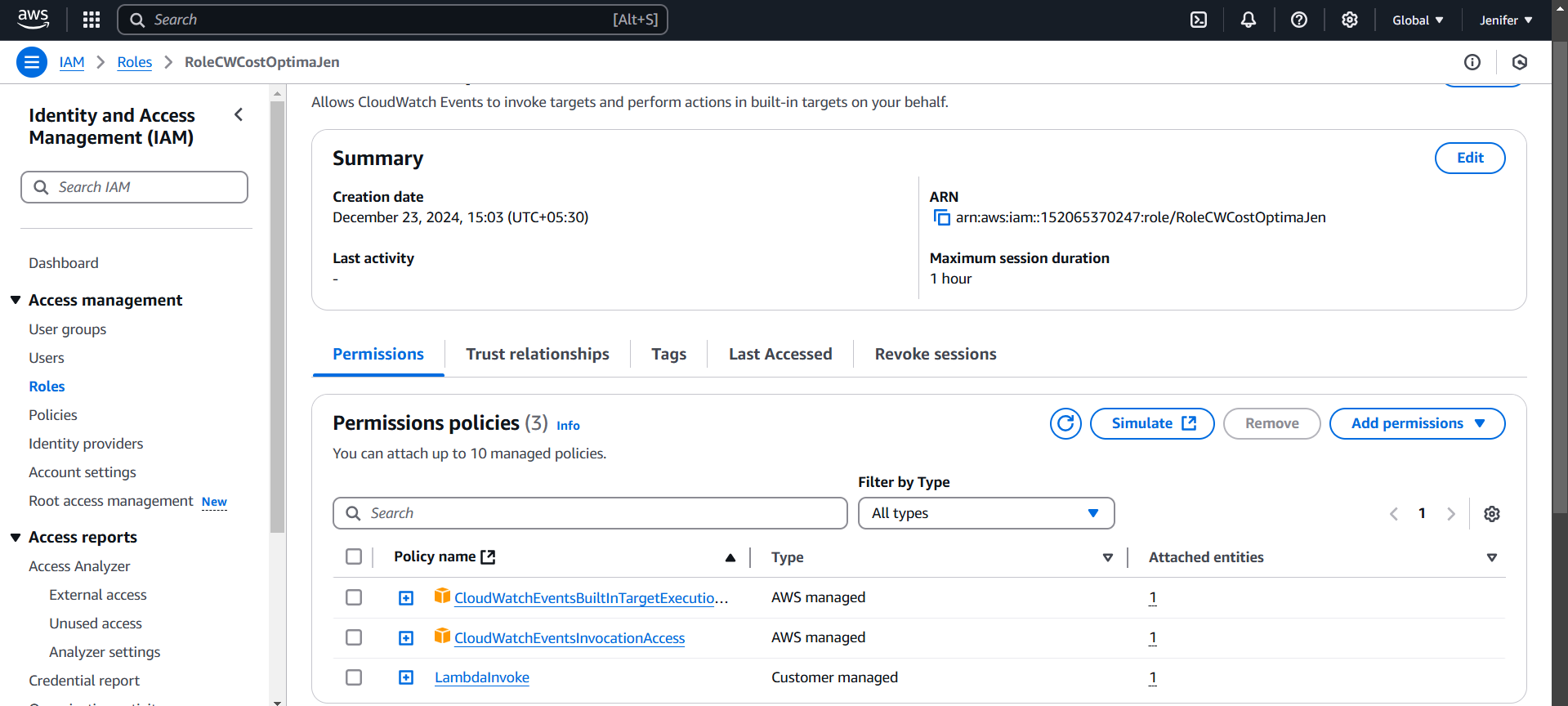


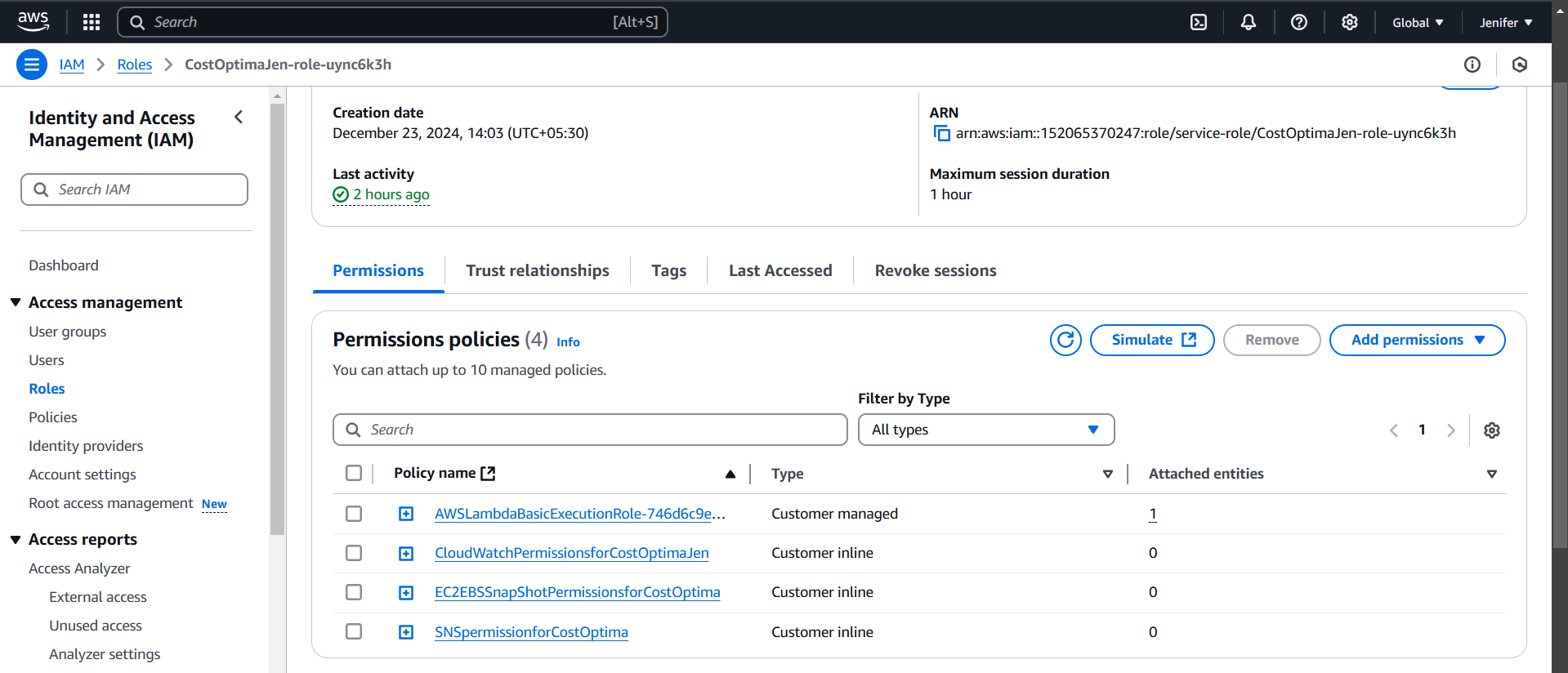
1. The Alarm got Created & CloudWatch Sent the Email Via SNS Configuration





**IV. IAM Role Creation for Lambda & CloudWatch to Communicate each other**





Needed Roles:

For Lambda:

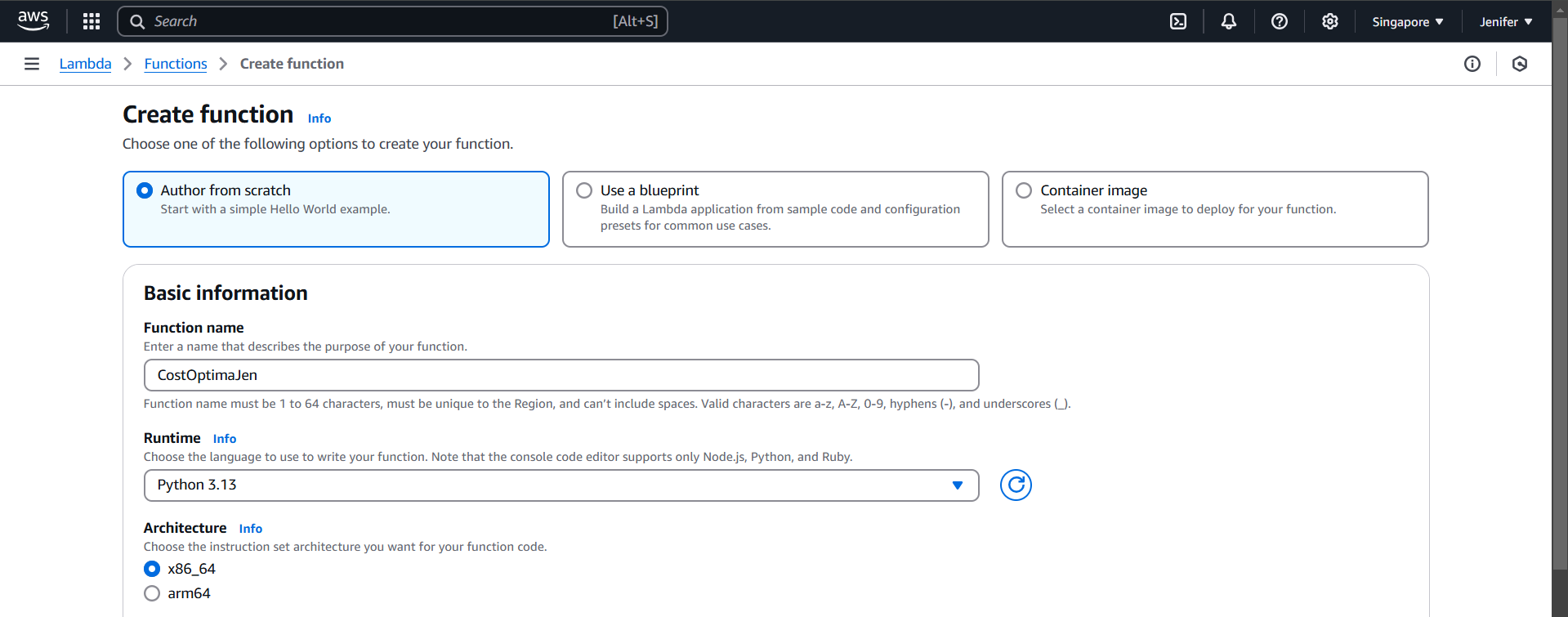
* "ec2:TerminateInstances"
* "ec2:DescribeInstances"
* "ec2:DescribeVolumes"
* "ec2:DescribeSnapshots"
* "ec2:DeleteVolume"
* "ec2:DeleteSnapshot"
* "ec2:DetachVolume"
* "sns:Publish"
* “logs:CreateLogGroup”,
* "logs:CreateLogStream”,
* "logs:PutLogEvents”

For CloudWatch:

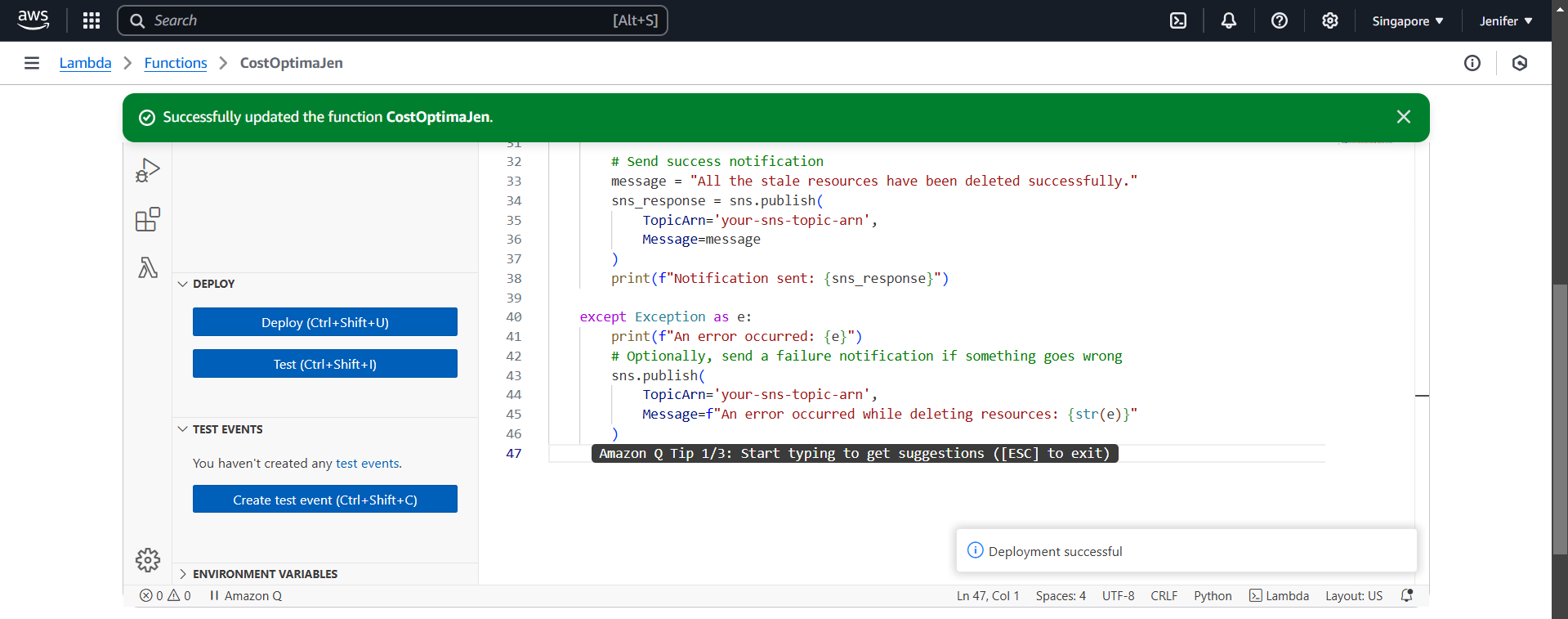
* “lambda:InvokeFunction”

**V. Lambda Function Configurations**

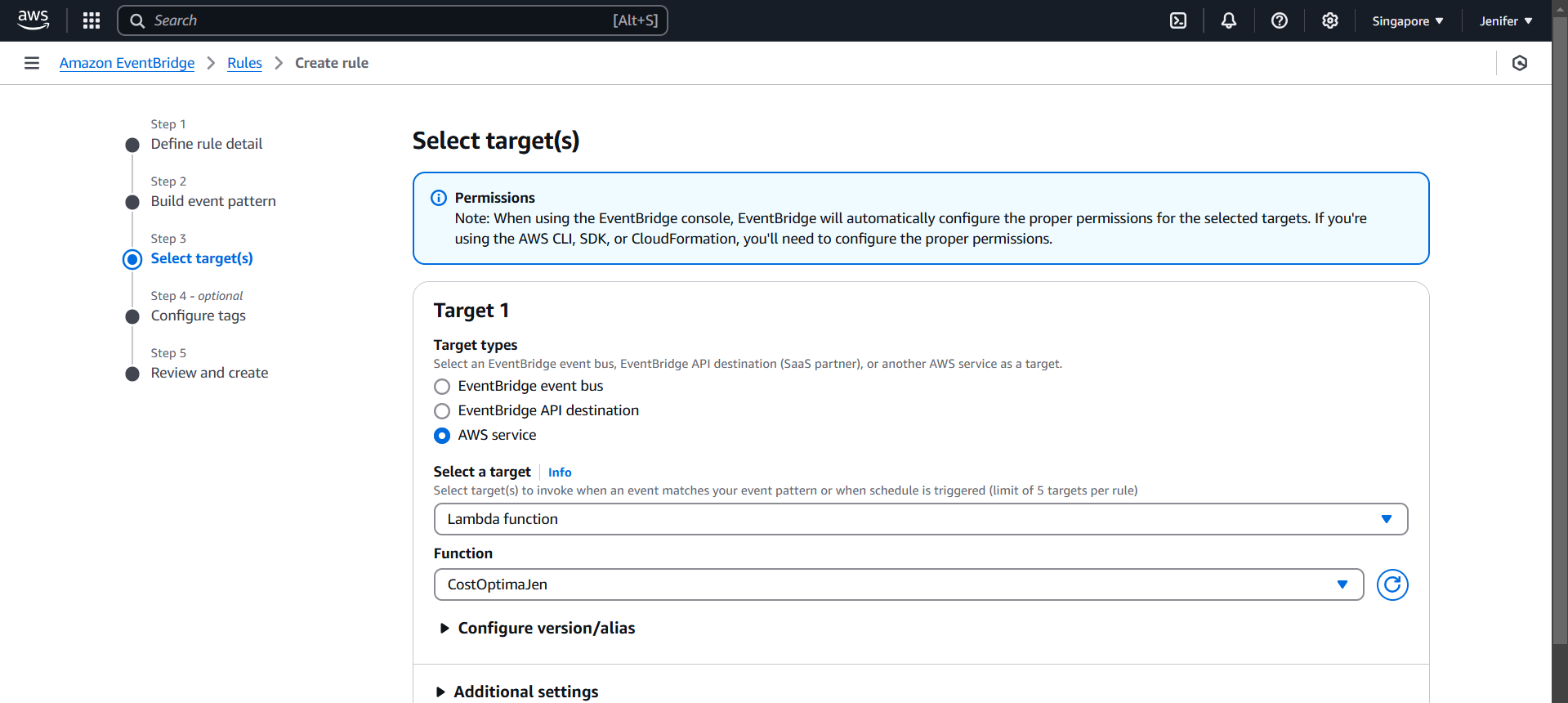
1. Basic Configuration

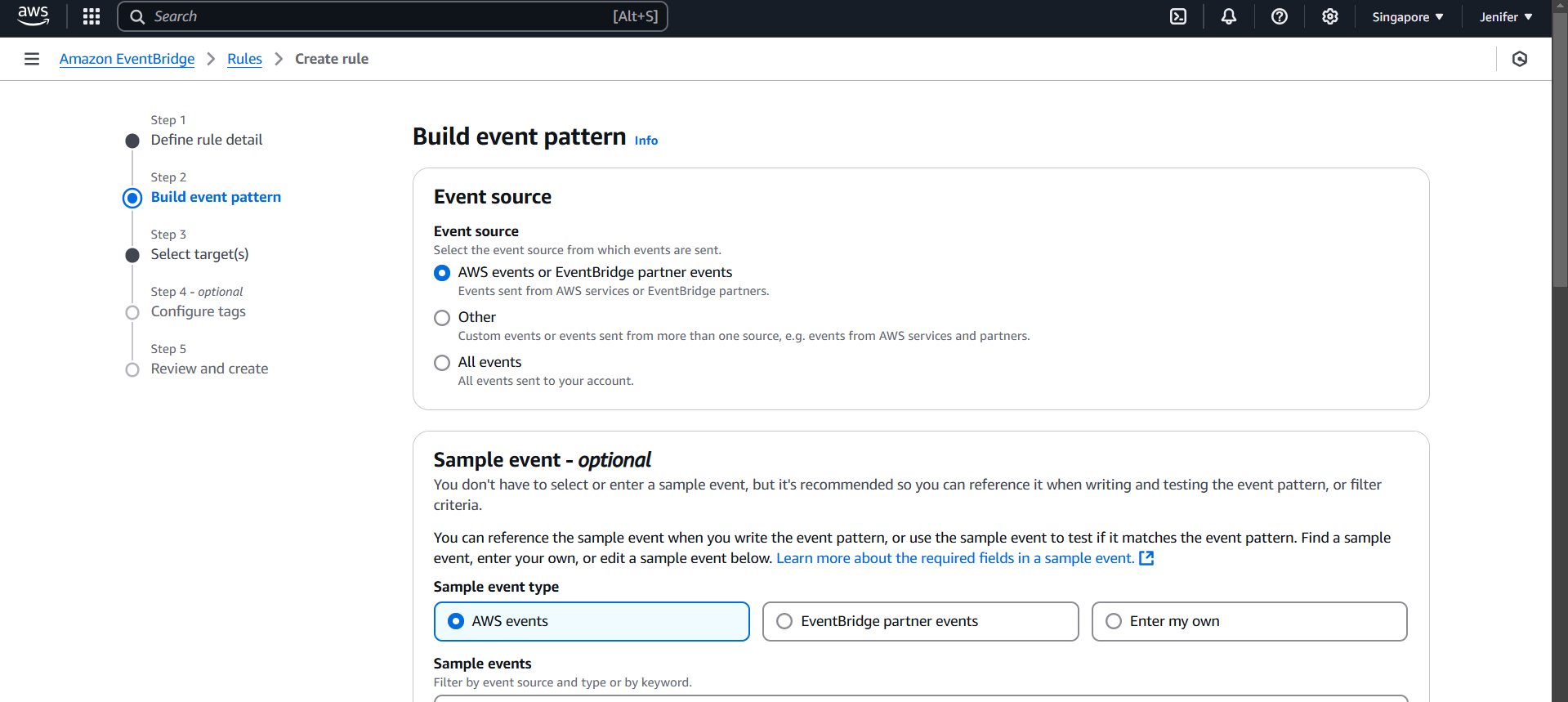


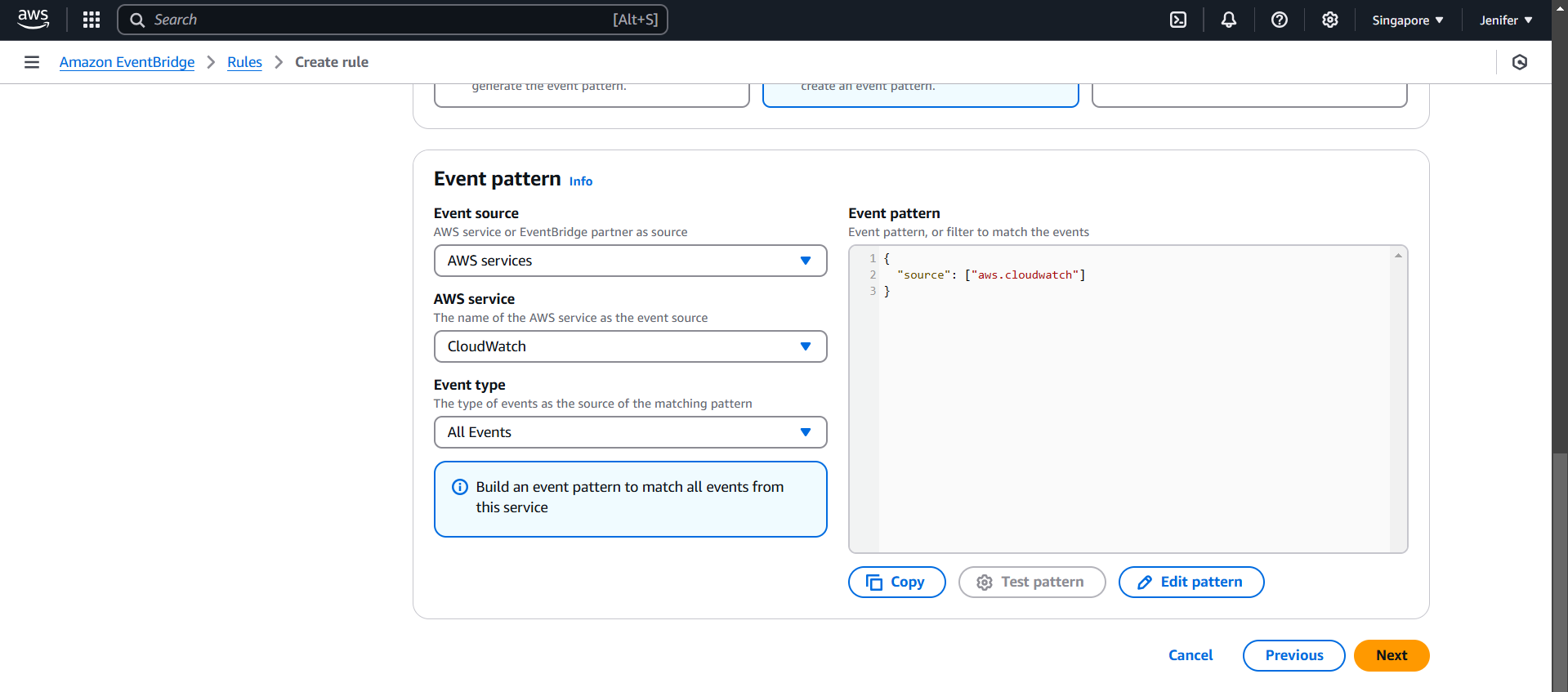
2. Code Deployment



**VI. Cloud Watch Rule Configurations**

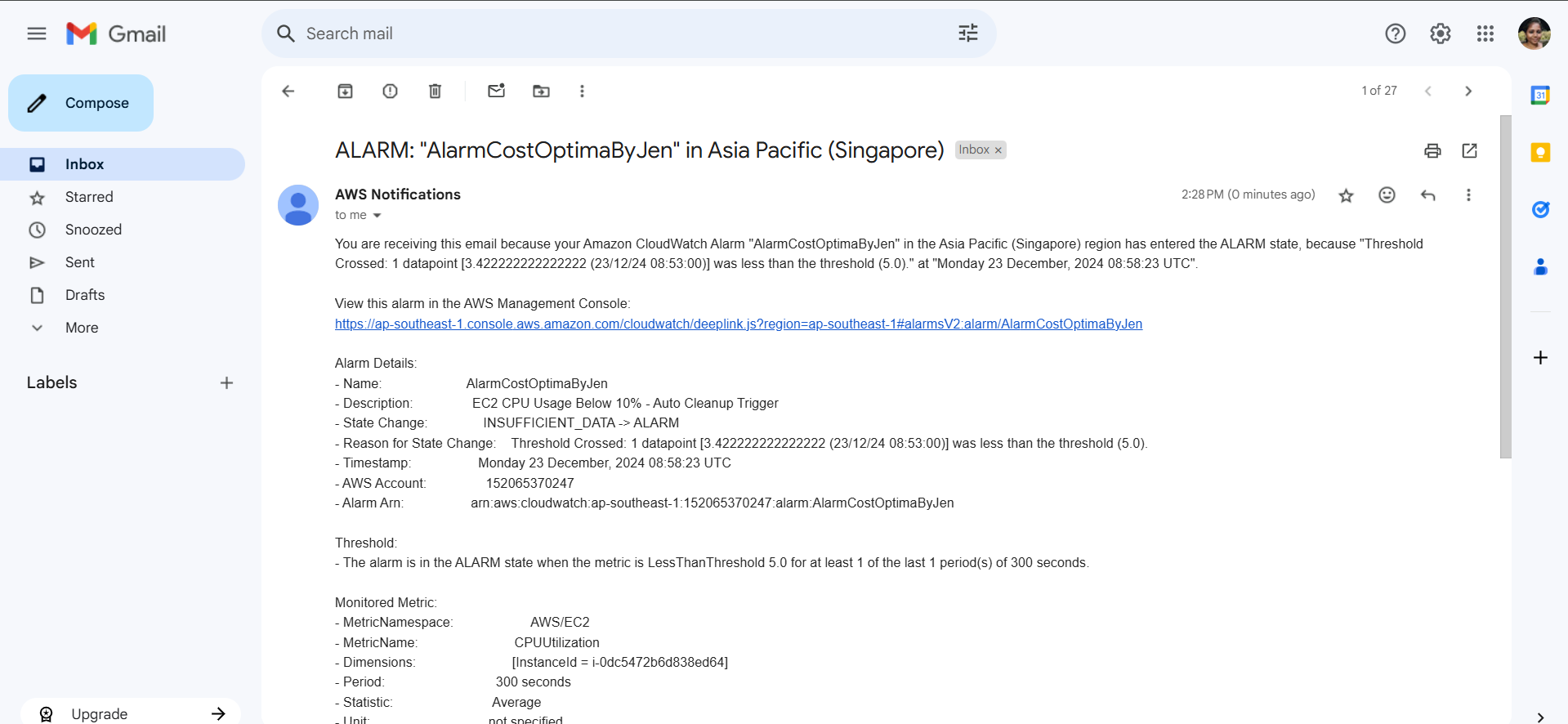
****

****

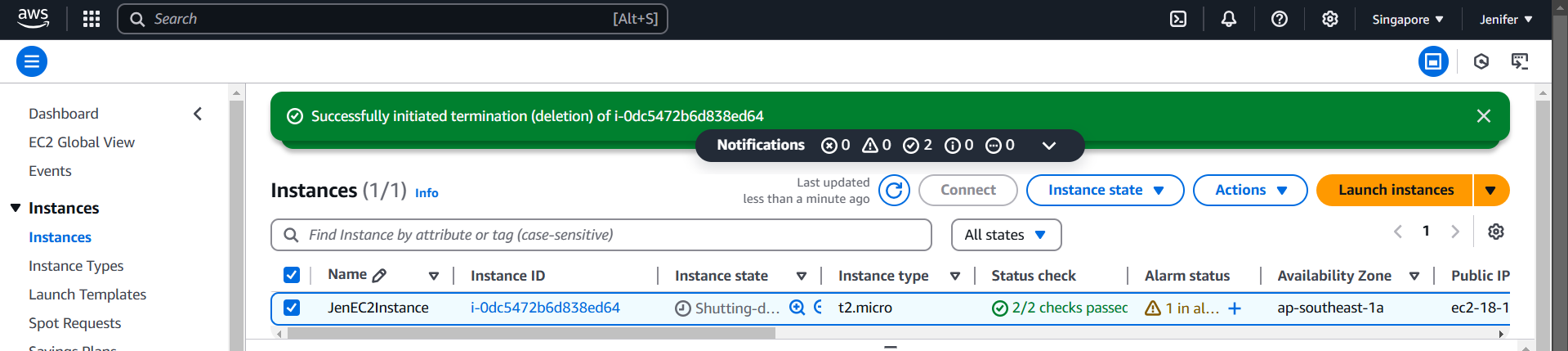
****

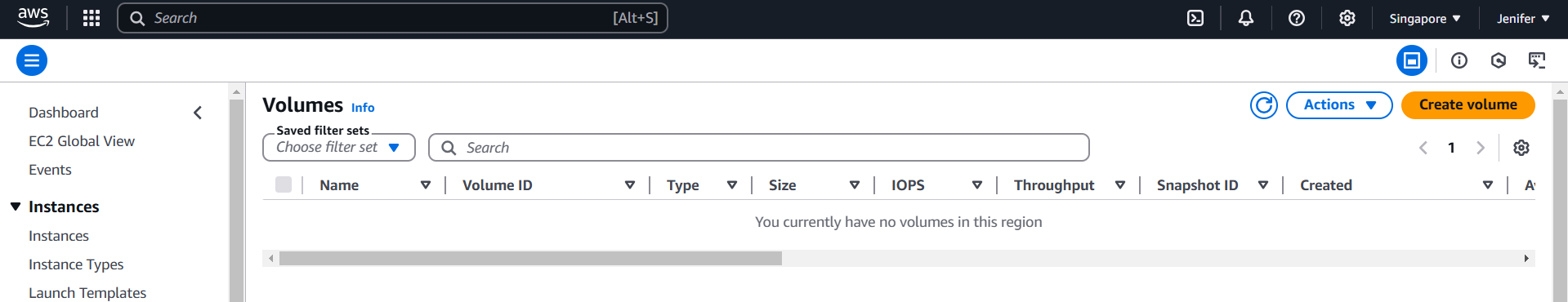
**V. The Outputs**

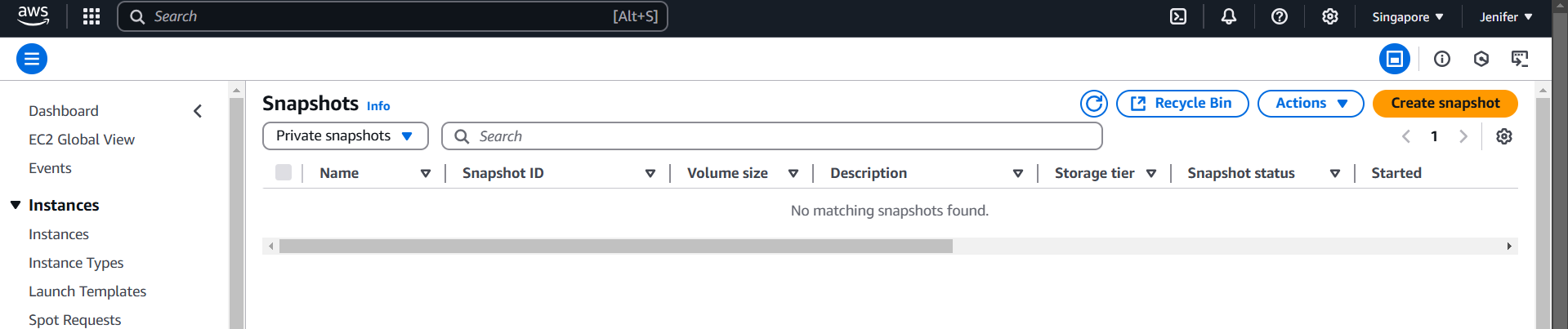
1. Alert Message got received

****

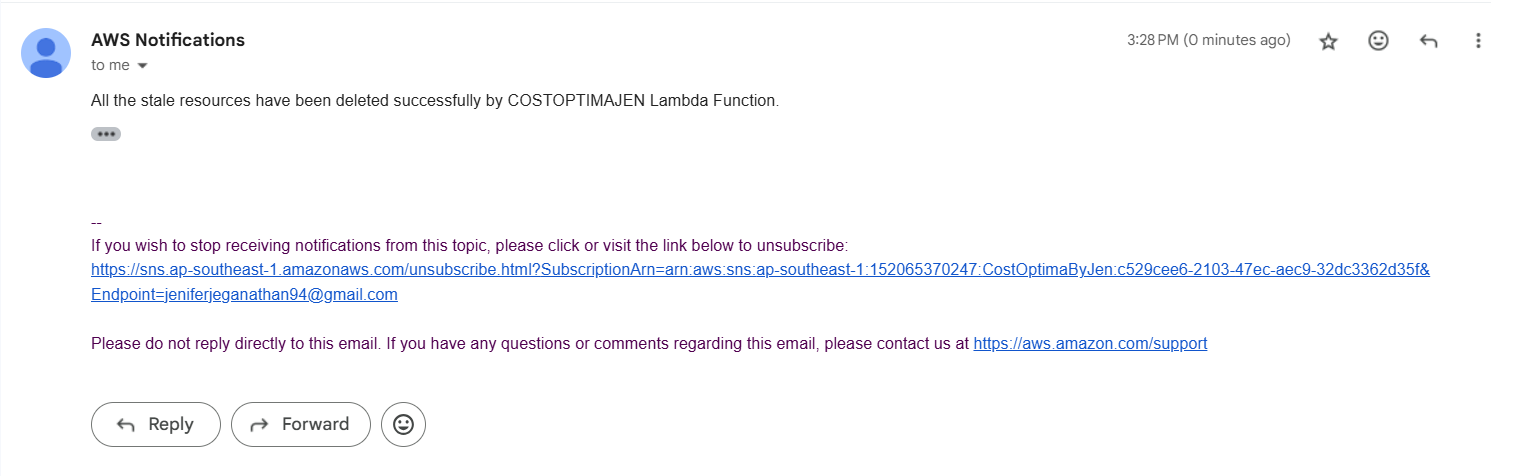
2. All the Resources got deleted by Invoking Lambda Function







3. Deletion Mail got Invoked



**Wrap-Up:**

The task was successfully completed by leveraging several AWS services to automate infrastructure management, monitor resources, and ensure data persistence and security. The key components, including EC2 instances, EBS volumes, snapshots, CloudFormation, CloudWatch, SNS, and Lambda, were efficiently configured to meet the requirements.