**Assignment 15: Monitor EC2 Instance State Changes Using AWS Lambda, Boto3, and SNS**

**Objective:** Automatically monitor changes in EC2 instance states and send notifications whenever an instance is started or stopped.

**Task:** Set up a Lambda function that listens to EC2 state change events and sends SNS notifications detailing the state changes.

**Instructions:**

1. SNS Setup:

   - Navigate to the SNS dashboard and create a new topic.

   - Subscribe to this topic with your email.

2. Lambda IAM Role:

   - Create a role with permissions to read EC2 instance states and send SNS notifications.

3. Lambda Function:

   - Create a function and assign the above IAM role.

   - Use Boto3 to:

     1. Extract details from the event regarding the EC2 state change.  
     2. Send an SNS notification with details about which EC2 instance changed state and the new state (e.g., started, stopped).

4. EC2 Event Bridge (formerly CloudWatch Events):

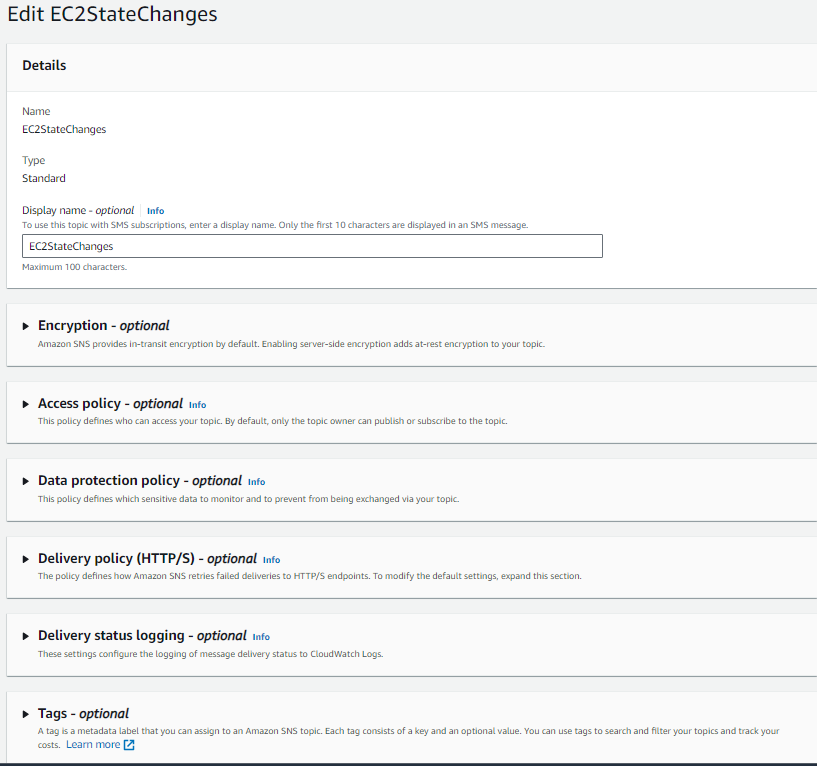
   - Set up an Event Bridge rule to trigger your Lambda function whenever an EC2 instance state changes.

5. Testing:

   - Start or stop one of your EC2 instances.

   - Confirm you receive an SNS notification about the state change.

Answer

SNS Setup:  


Navigate to the SNS dashboard in the AWS Management Console. 2 Click on "Topics" in the left-hand menu and then click the "Create topic" button.

Enter a name and display name for your topic (e.g. "EC2StateChanges").

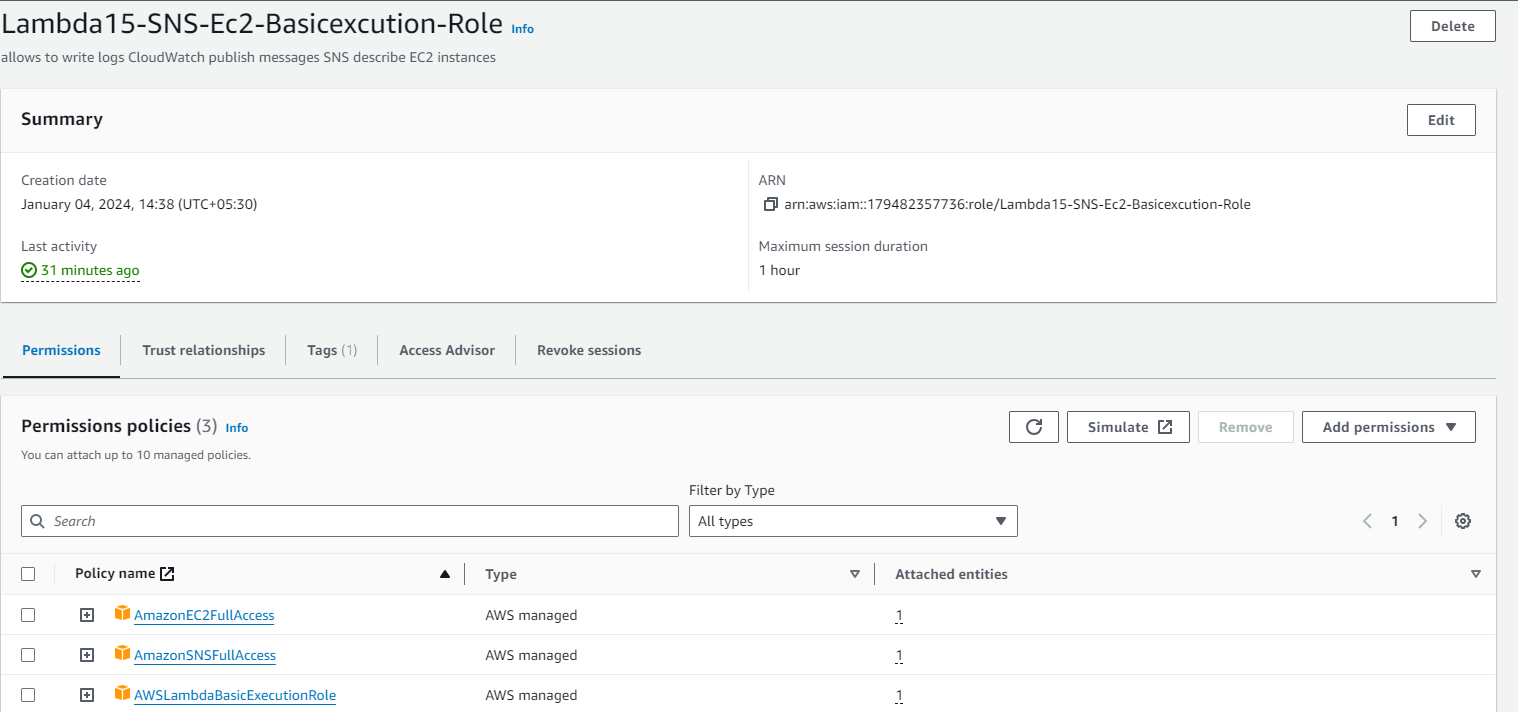
Click "Create topic".

Click on your new topic and then click the "Create subscription" button.

Enter your email address as the protocol and endpoint, and then click "Create subscription".

Check your email and click the confirmation link to confirm your subscription.

Lambda IAM Role:



Navigate to the IAM dashboard in the AWS Management Console.

Click on "Roles" in the left-hand menu and then click the "Create role" button.

Select "Lambda" as the service that will use this role and then click "Next: Permissions".

Attach the following policies to the role:

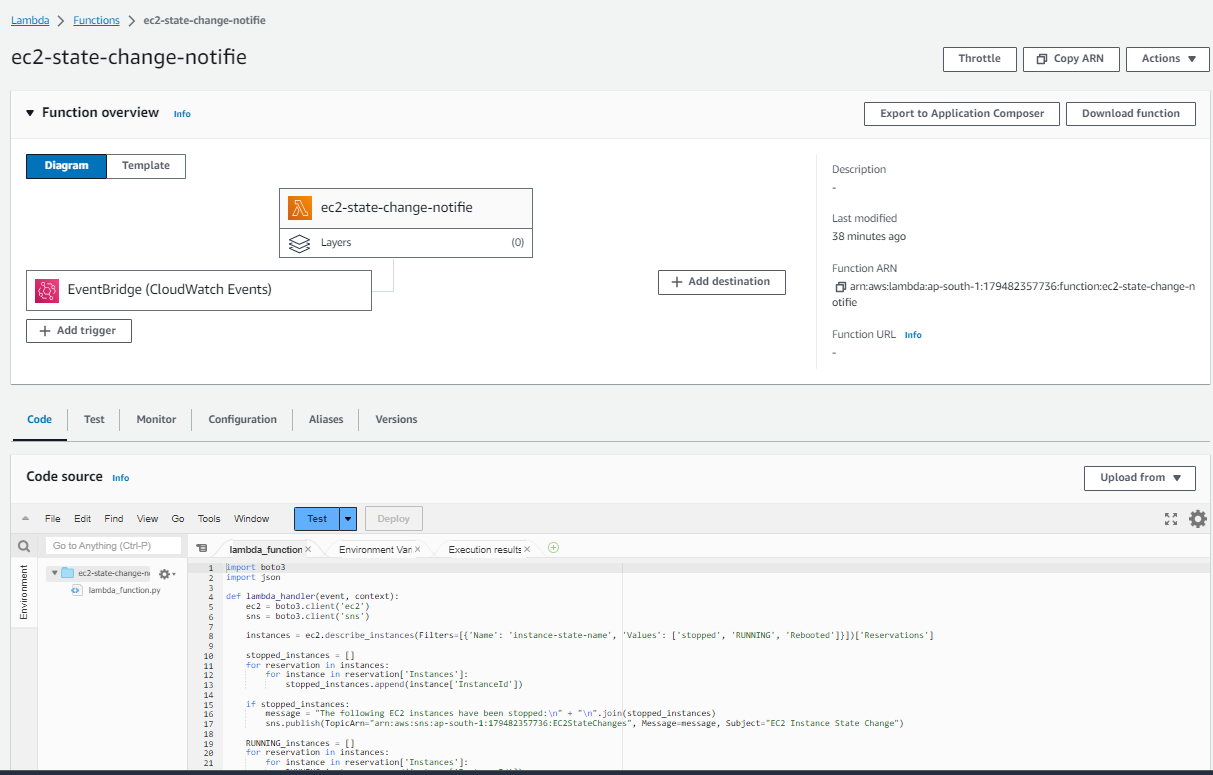
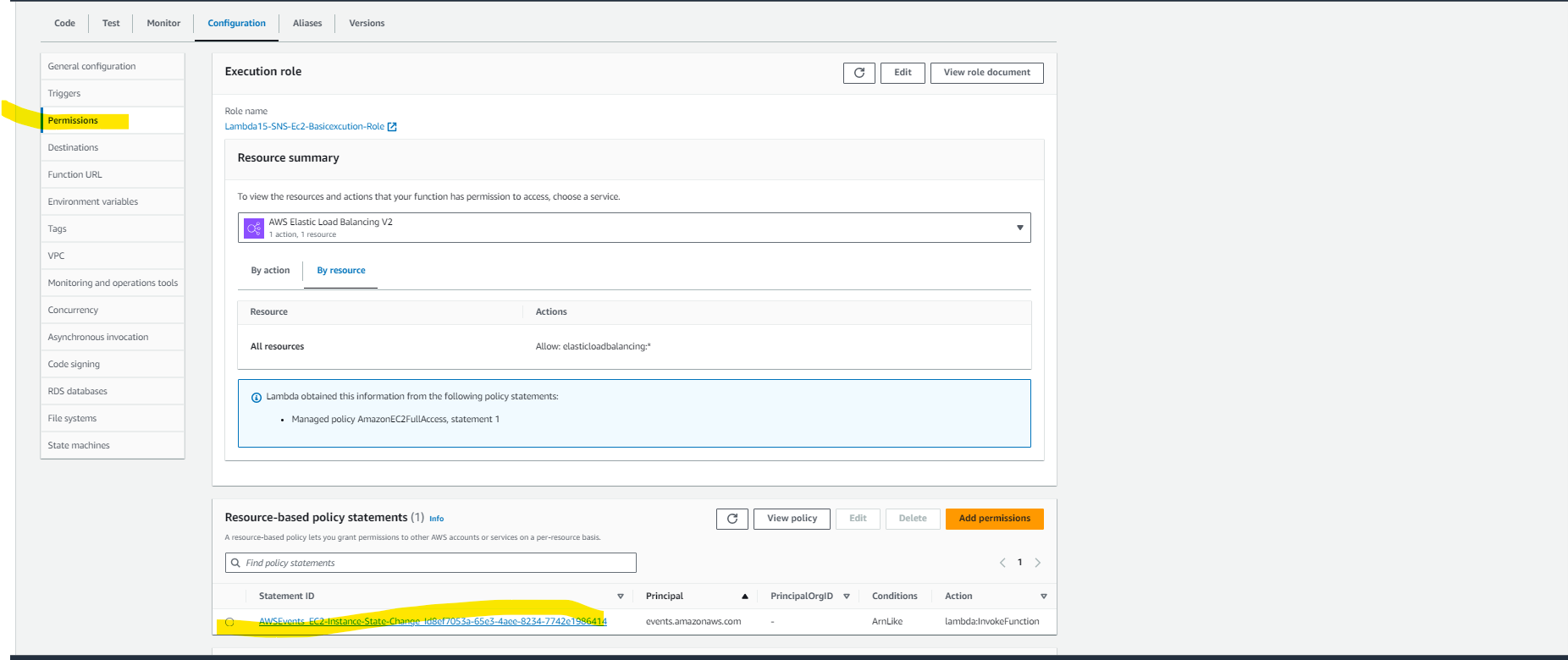
AWSLambdaBasicExecutionRole (this policy allows the Lambda function to write logs to CloudWatch)

AmazonSNSFullAccess (this policy allows the Lambda function to publish messages to an SNS topic)

AmazonEC2FullAccess (this policy allows the Lambda function to describe EC2 instances)

Click "Next: Tags", add any desired tags, and then click "Next: Review".

Enter a name and description for your role and then click "Create role".

Lambda Function:  
  


Navigate to the Lambda dashboard in the AWS Management Console.

Click the "Create function" button.

Select "Author from scratch" and then enter a name and runtime for your function (e.g. "ec2-state-change-notifier" and "python3.8").

Under "Choose or create an execution role", select "Use an existing role" and then choose the IAM role you created earlier.

Click "Create function".  
  
move to general configuration and extend the execution time from 3 seconds to 20 seconds to avoid timeout as 3s is by default

In the function code editor, replace the existing code with the following Python code  
  
import boto3

import json

def lambda\_handler(event, context):

ec2 = boto3.client('ec2')

sns = boto3.client('sns')

instances = ec2.describe\_instances(Filters=[{'Name': 'instance-state-name', 'Values': ['stopped', 'RUNNING', 'Rebooted']}])['Reservations']

stopped\_instances = []

for reservation in instances:

for instance in reservation['Instances']:

stopped\_instances.append(instance['InstanceId'])

if stopped\_instances:

message = "The following EC2 instances have been stopped:\n" + "\n".join(stopped\_instances)

sns.publish(TopicArn="arn:aws:sns:ap-south-1:179482357736:EC2StateChanges", Message=message, Subject="EC2 Instance State Change")

RUNNING\_instances = []

for reservation in instances:

for instance in reservation['Instances']:

RUNNING\_instances.append(instance['InstanceId'])

if RUNNING\_instances:

message = "The following EC2 instances have been running:\n" + "\n".join(RUNNING\_instances)

sns.publish(TopicArn="arn:aws:sns:ap-south-1:179482357736:EC2StateChanges", Message=message, Subject="EC2 Instance State Change")

Rebooted\_instances = []

for reservation in instances:

for instance in reservation['Instances']:

Rebooted\_instances.append(instance['InstanceId'])

if Rebooted\_instances:

message = "The following EC2 instances have been rebooted:\n" + "\n".join(Rebooted\_instances)

sns.publish(TopicArn="arn:aws:sns:ap-south-1:179482357736:EC2StateChanges", Message=message, Subject="EC2 Instance State Change")

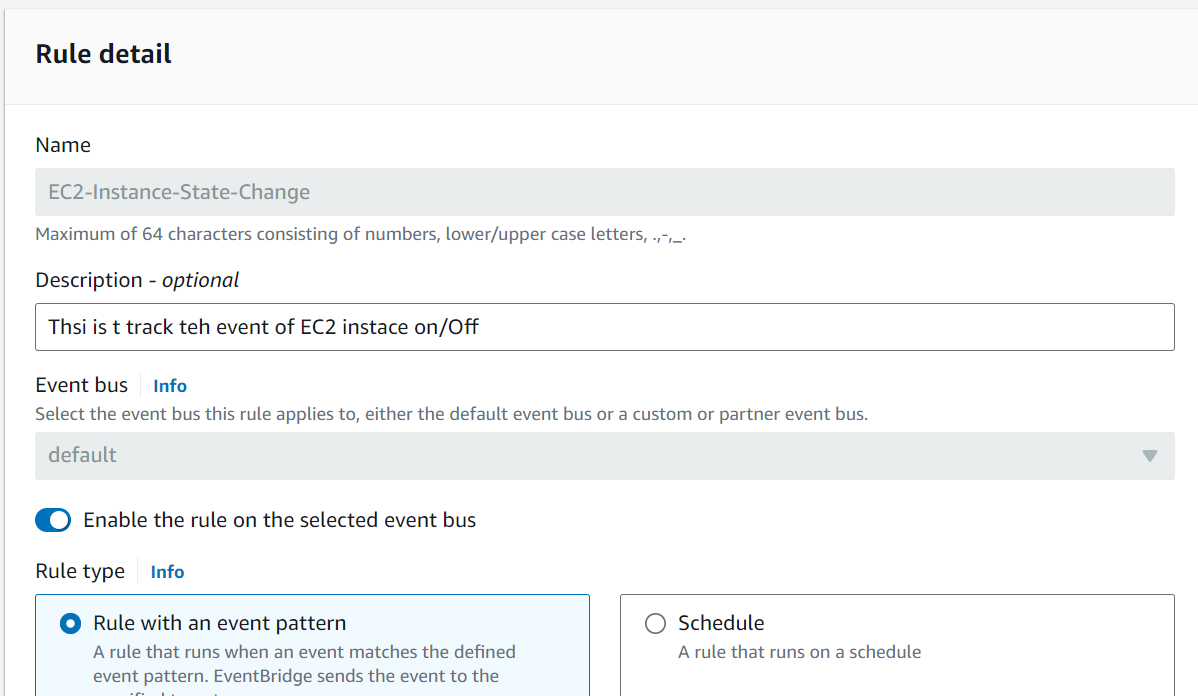
Replace the TopicArn value in the sns.publish() call with the ARN of your SNS topic.

Click "Save".

EC2 Event Bridge:

Navigate to the Event Bridge dashboard in the AWS Management Console.

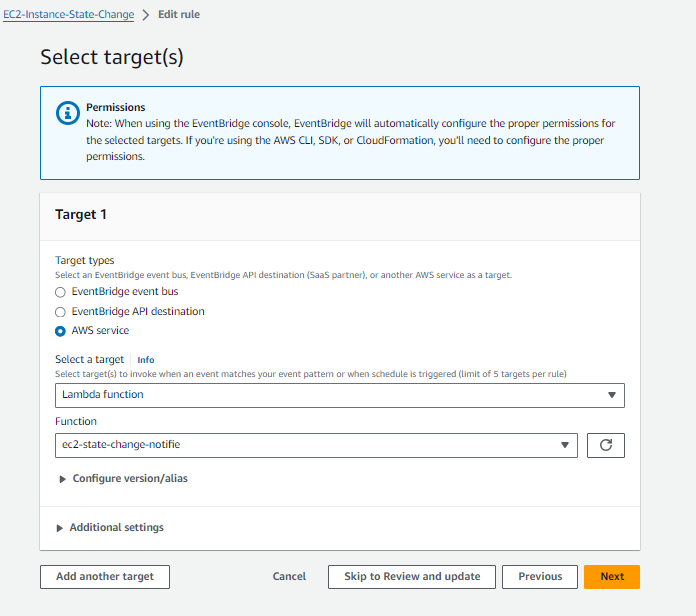
Click on "Rules" in the left-hand menu and then click the "Create rule" button.

**Step 1.**  In Rule detail Tab  


# Step 2. Build event pattern

# 

# Step 3. Select Target



Under "Select targets", choose "Lambda function" and then select the Lambda function you created earlier.

Click "Create".

Testing:

Start or stop one of your EC2 instances.

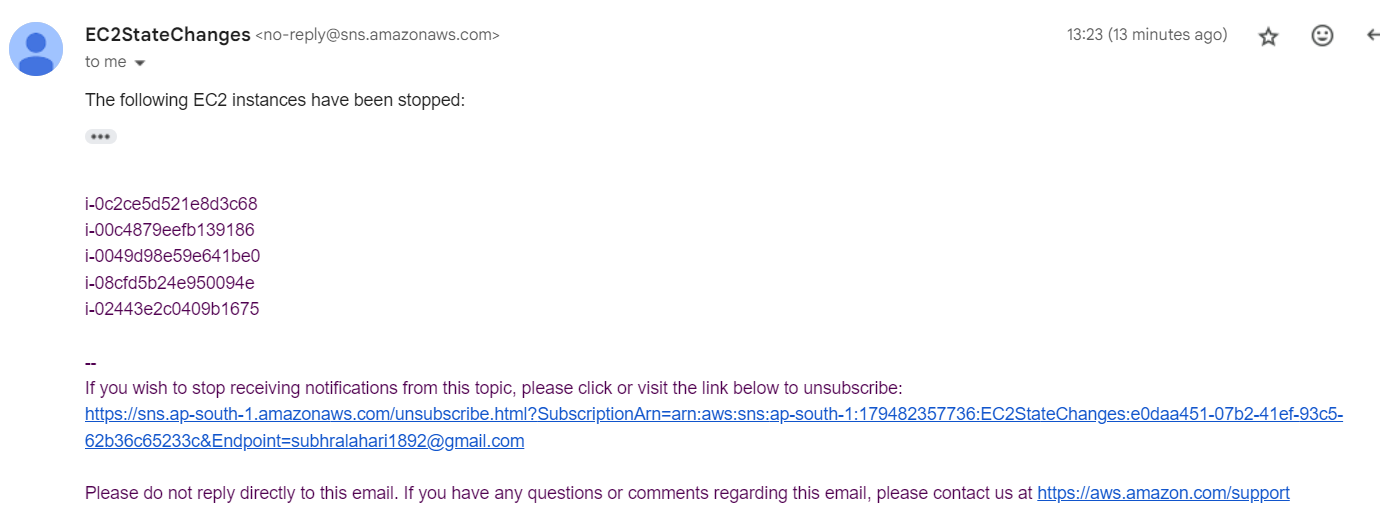
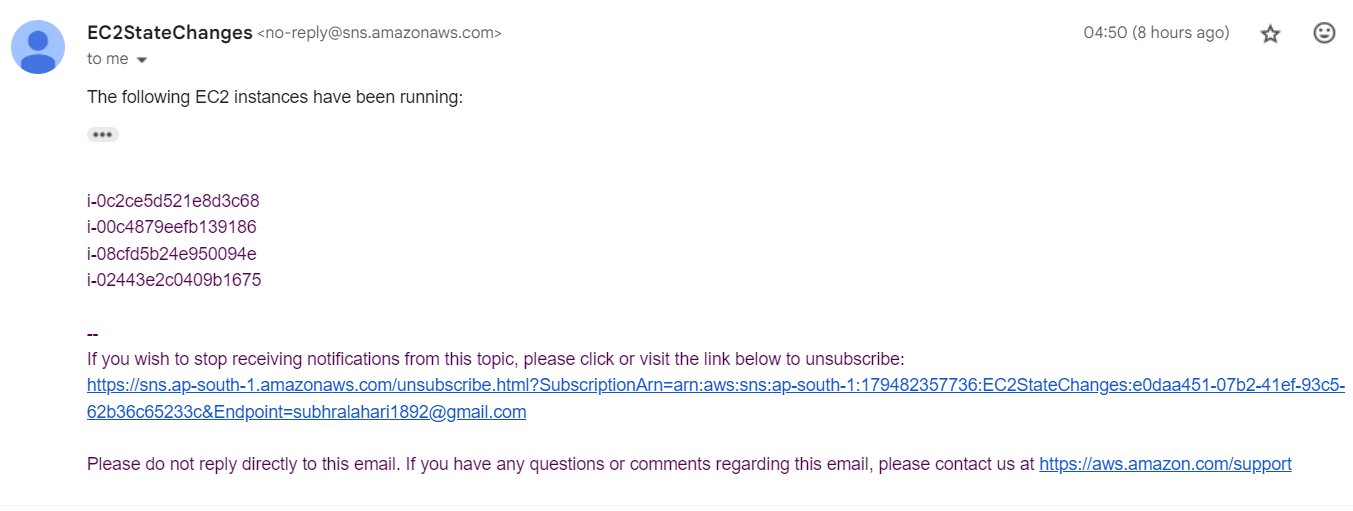
Check your email for a notification about the state change.

Here's how the different services are connected:

The Lambda function is triggered by an Event Bridge rule that listens for EC2 instance state change events.

The Lambda function extracts details about the state change from the event and sends an SNS notification with those details.

The SNS topic is subscribed to by your email address, so you receive a notification whenever an EC2 instance state changes.

Output  
  
  


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*