**Assignment 20: Load Balancer Health Checker**

**Objective:**Design a Lambda function that checks the health of registered instances behind an Elastic Load Balancer (ELB) and notifies via SNS if any instances are unhealthy.

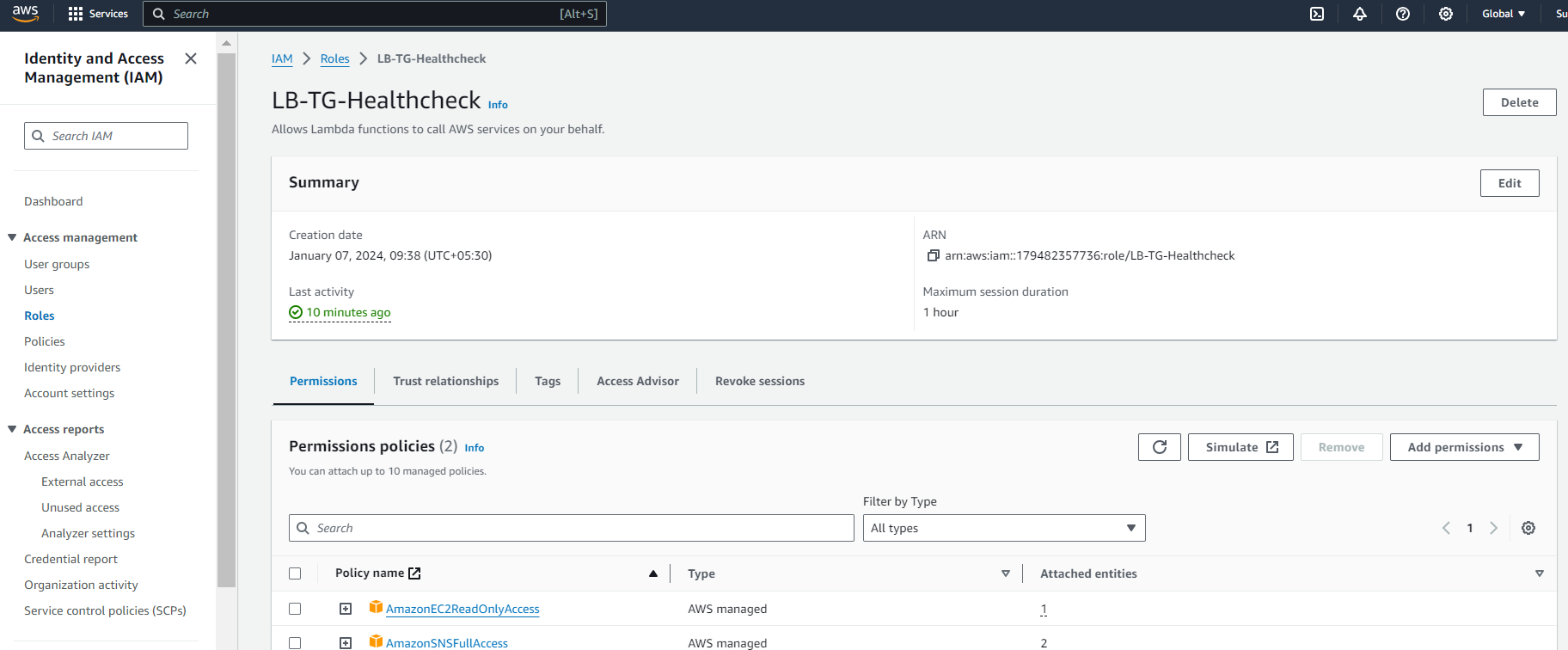
**Instructions:**

1. Create a Lambda function.

2. With Boto3, configure the function to:

1. Check the health of registered instances behind a given ELB.  
2. If any instances are found to be unhealthy, publish a detailed message to an SNS topic.

3. Set up a CloudWatch event to trigger this Lambda function every 10 minutes.

.  
  
**Step 1: Lambda IAM Role:**

Navigate to the IAM service.

Click on "Roles" in the left navigation pane.

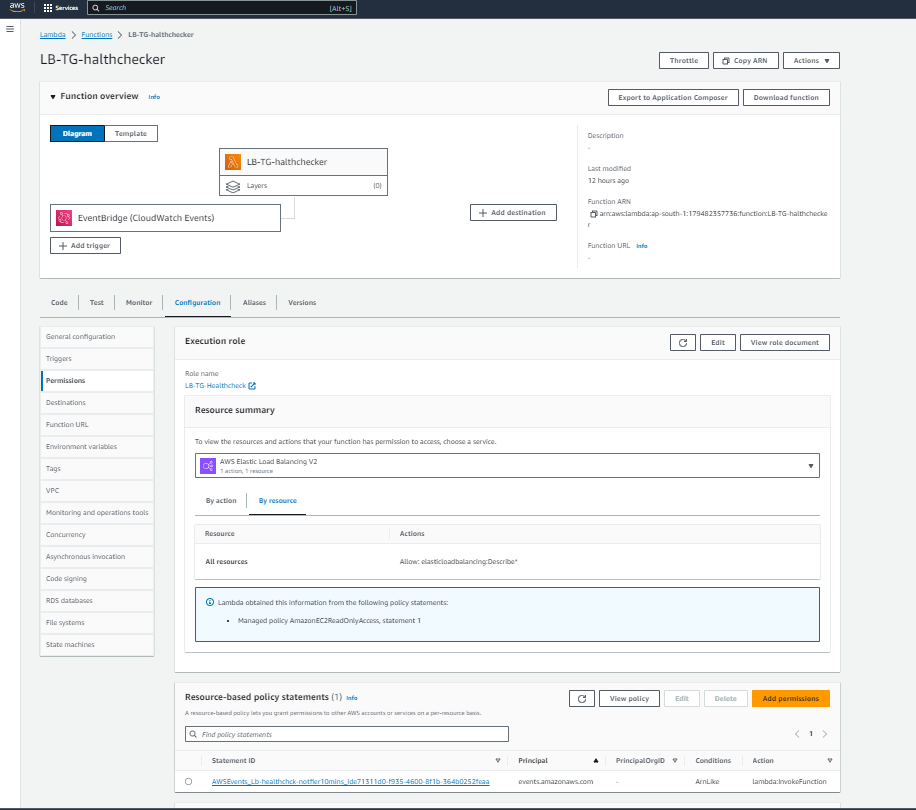
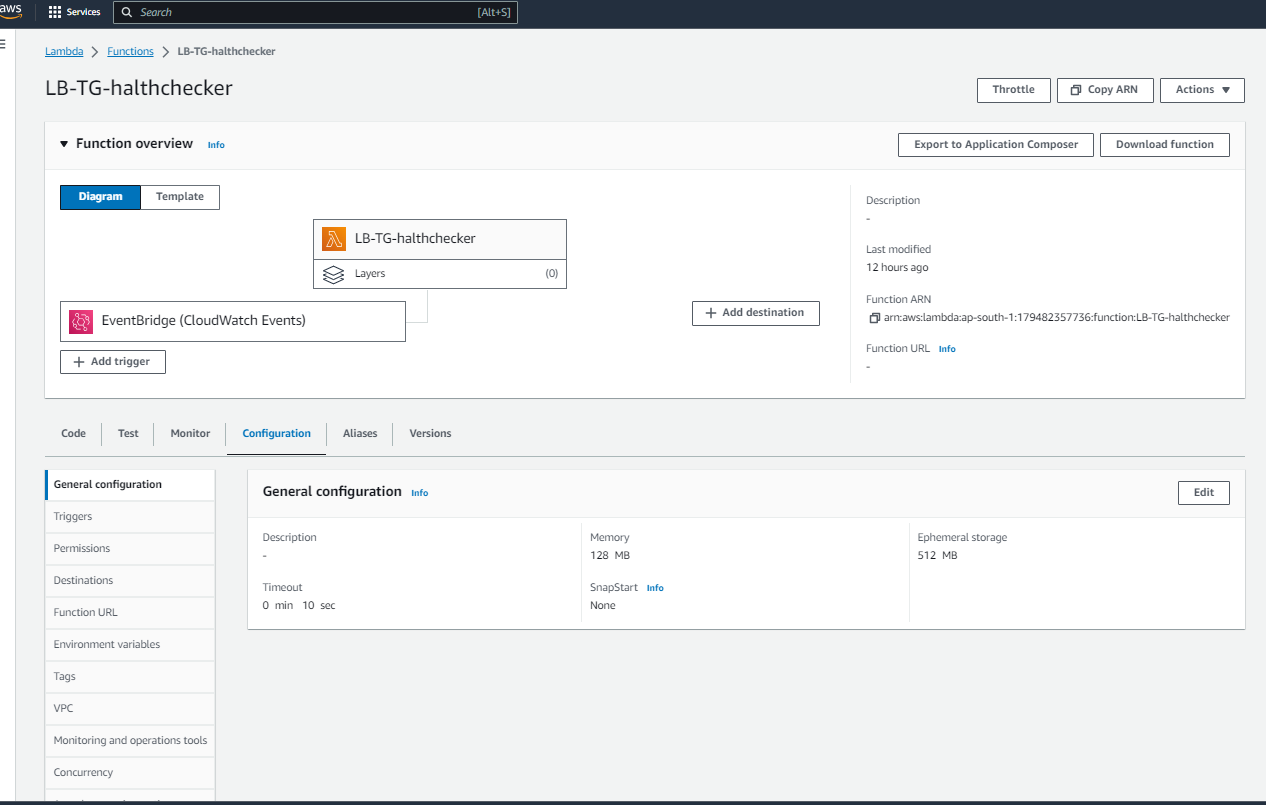
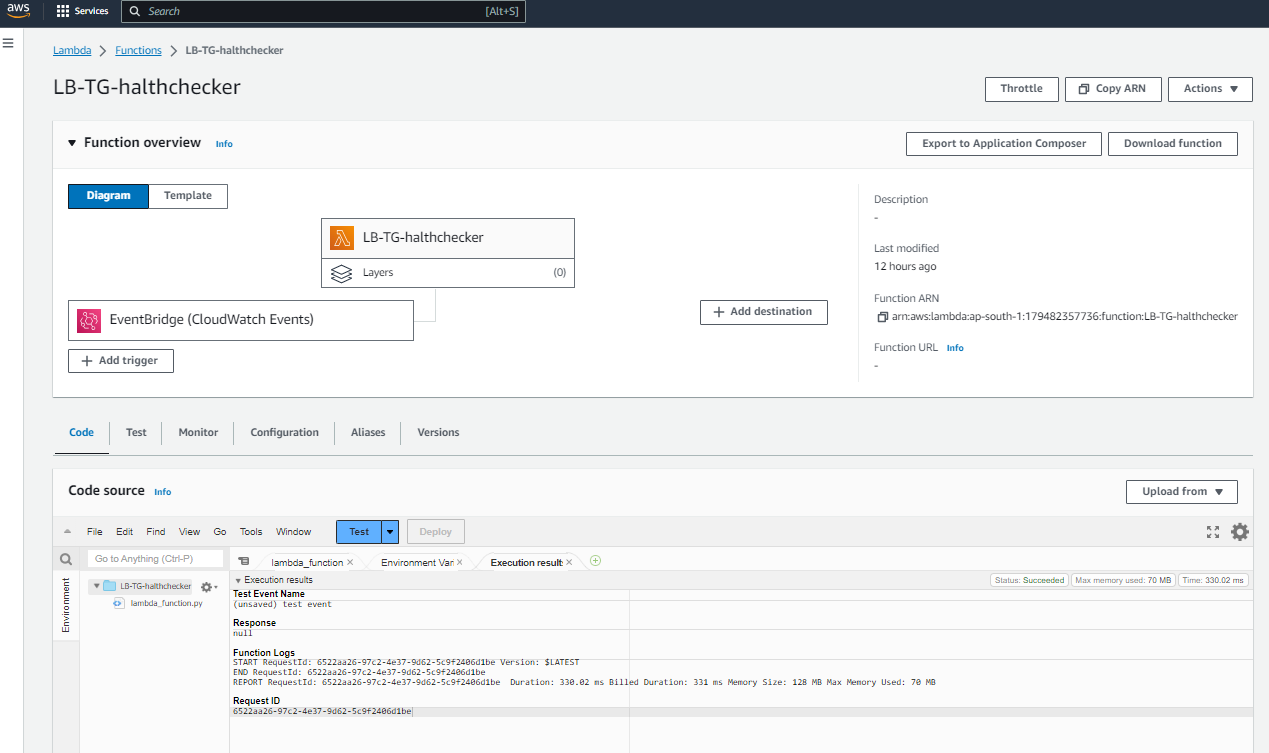
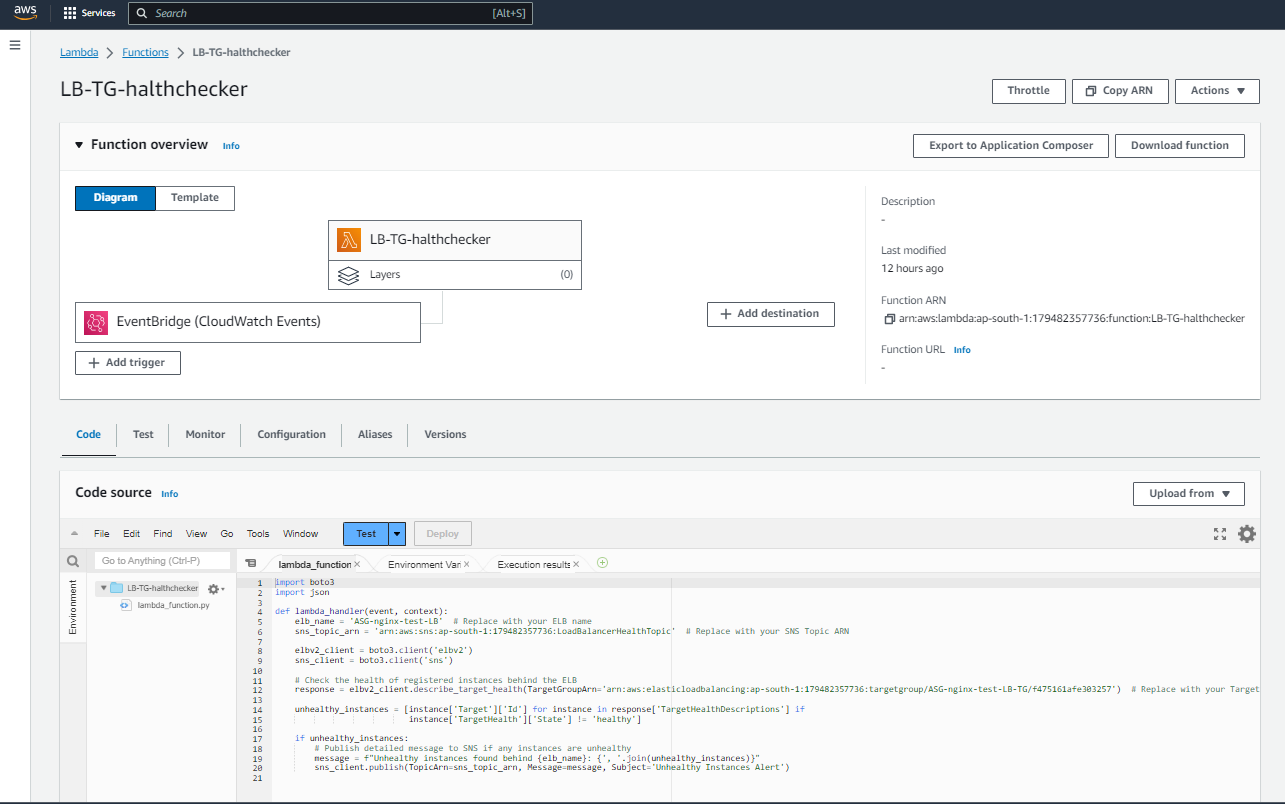
Click the "Create role" button.

Choose Lambda as the service that will use this role.

Attach policies that grant the necessary permissions, such as AmazonEC2ReadOnlyAccess and AmazonSNSFullAccess.

Name the role and create it.

Note the ARN of the created IAM role.

**Step 2: Deploy Lambda Function::  
  
  
  
**Navigate to the Lambda dashboard and create a new function by following these steps:

a. Click on the "Create function" button.

b. Choose "Author from scratch" and give your function a name

c. Choose Python 3.x as the runtime.

d. Under "Permissions", choose "Use an existing role" and select the IAM role you created in the previous step.

e. Write the **Boto3 Python script** to constantly health check the Instances in the Target group of the Load balancer and send message using configured SNS service if the instance health state is health or unhealthy .   
  
  
Here's an example script:import boto3

import json

def lambda\_handler(event, context):

elb\_name = 'ASG-nginx-test-LB' # Replace with your ELB name

sns\_topic\_arn = 'arn:aws:sns:ap-south-1:179482357736:LoadBalancerHealthTopic' # Replace with your SNS Topic ARN

elbv2\_client = boto3.client('elbv2')

sns\_client = boto3.client('sns')

# Check the health of registered instances behind the ELB

response = elbv2\_client.describe\_target\_health(TargetGroupArn='arn:aws:elasticloadbalancing:ap-south-1:179482357736:targetgroup/ASG-nginx-test-LB-TG/f475161afe303257') # Replace with your Target Group ARN

unhealthy\_instances = [instance['Target']['Id'] for instance in response['TargetHealthDescriptions'] if

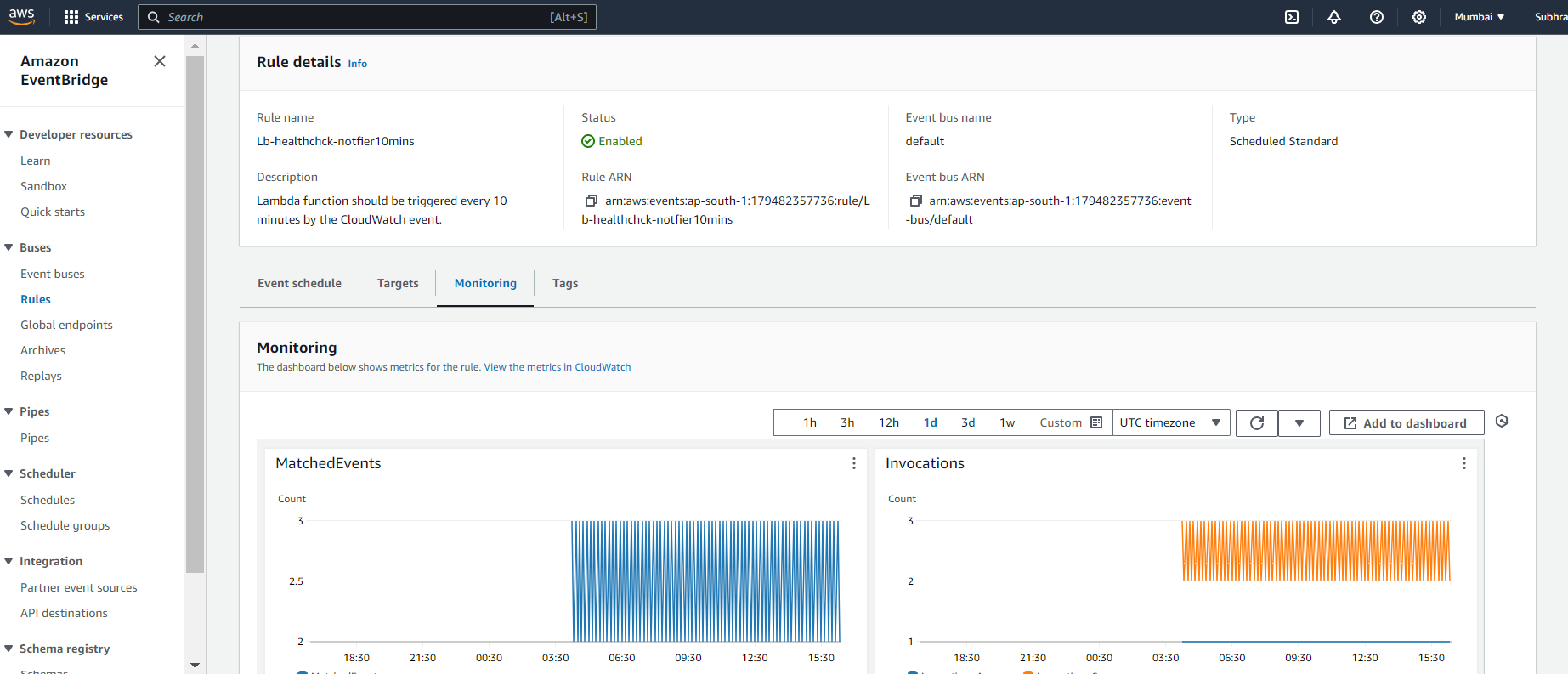
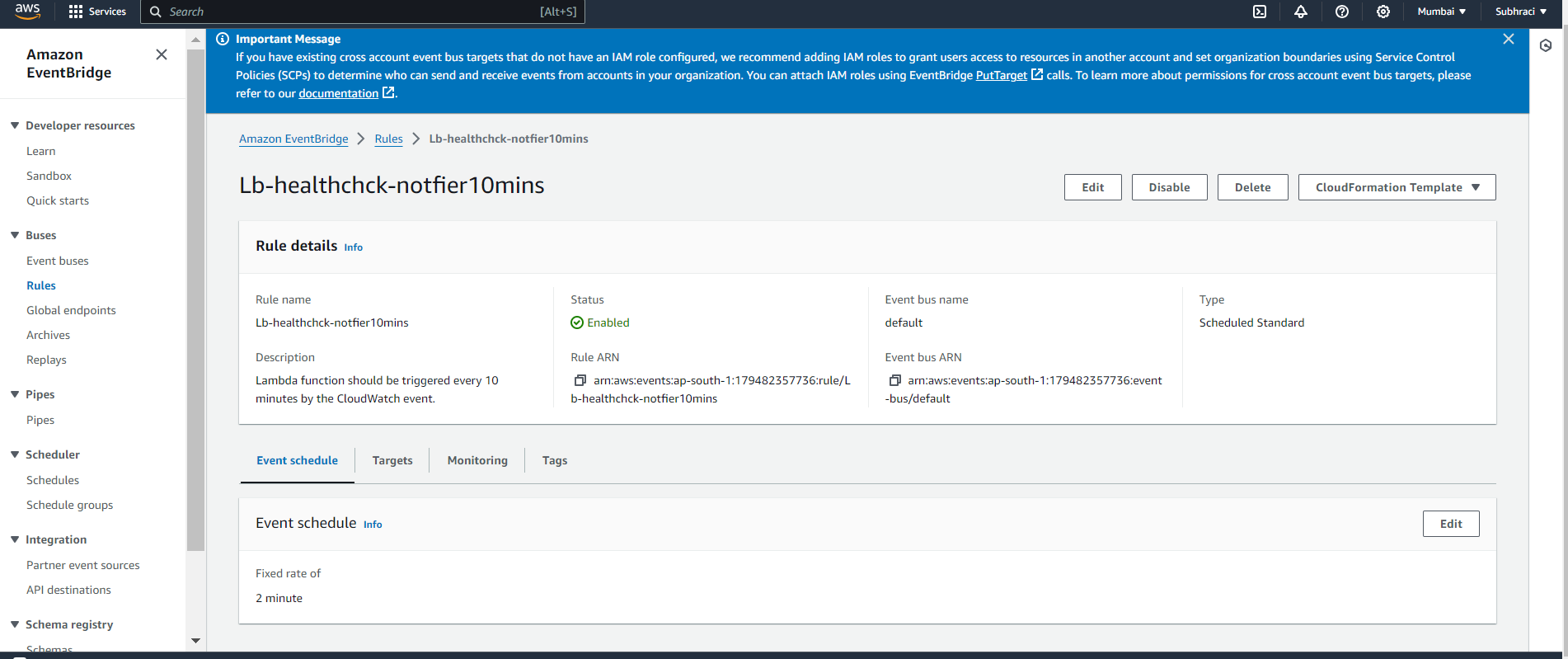
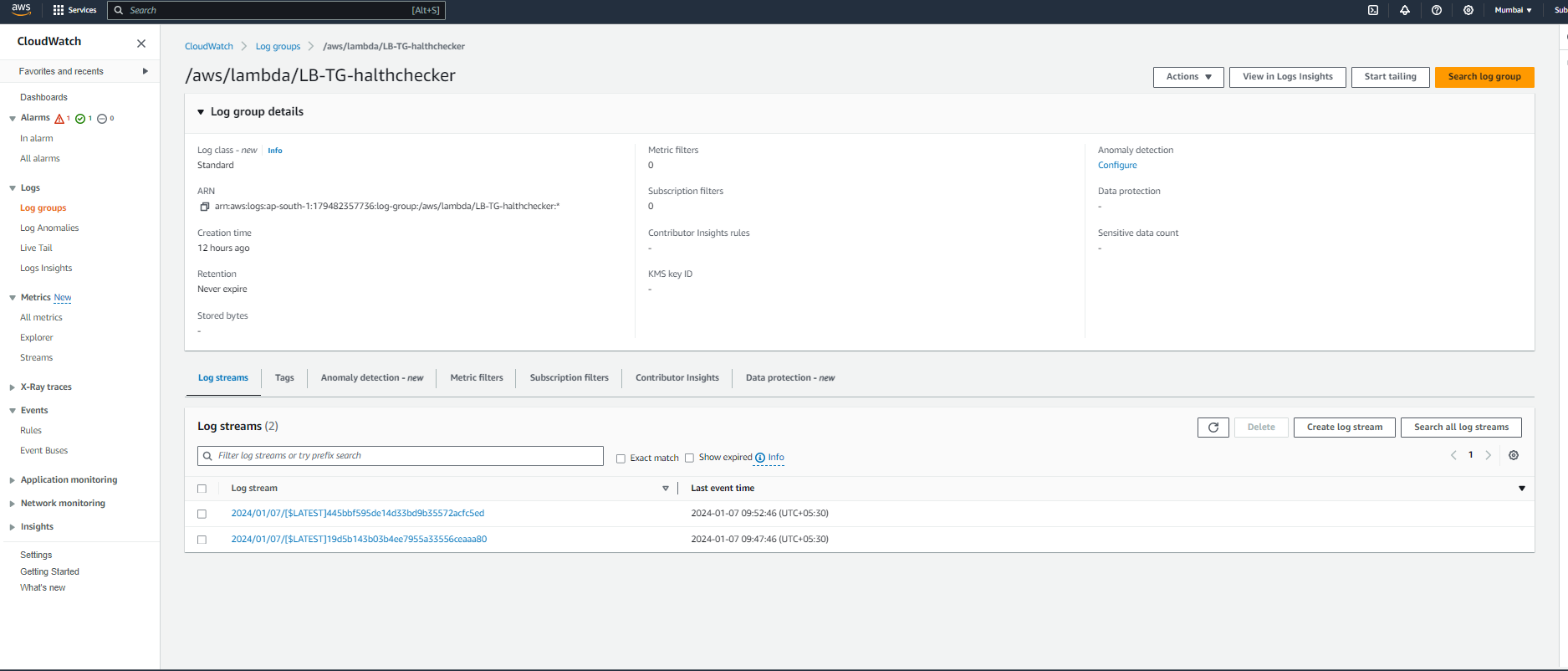
instance['TargetHealth']['State'] != 'healthy']

if unhealthy\_instances:

# Publish detailed message to SNS if any instances are unhealthy

message = f"Unhealthy instances found behind {elb\_name}: {', '.join(unhealthy\_instances)}"

sns\_client.publish(TopicArn=sns\_topic\_arn, Message=message, Subject='Unhealthy Instances Alert')

**Step 3: Set Up CloudWatch Event Manually  
  
  
**

Go to the AWS Management Console.

Navigate to the CloudWatch service.

Click on "Rules" in the left navigation pane.

Click the "Create rule" button.

In the "Event Source" section, choose "Event Source Type" as "Event Source".

In the "Event Source" section, choose "Event Type" as "Scheduled".

Set the "Fixed rate of" to "10 minutes" ( 2 mins to check and confirm quickly if the output desired is seen )

In the "Targets" section, click "Add target".

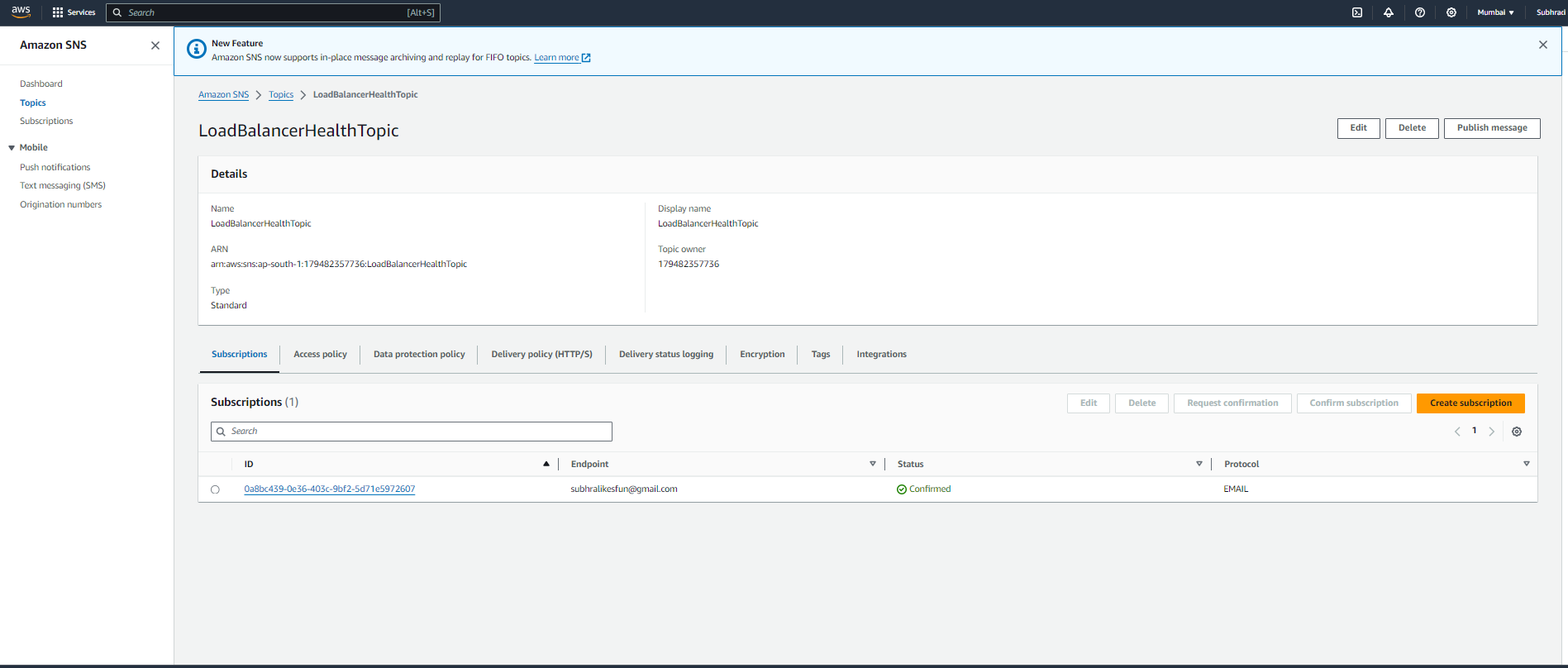
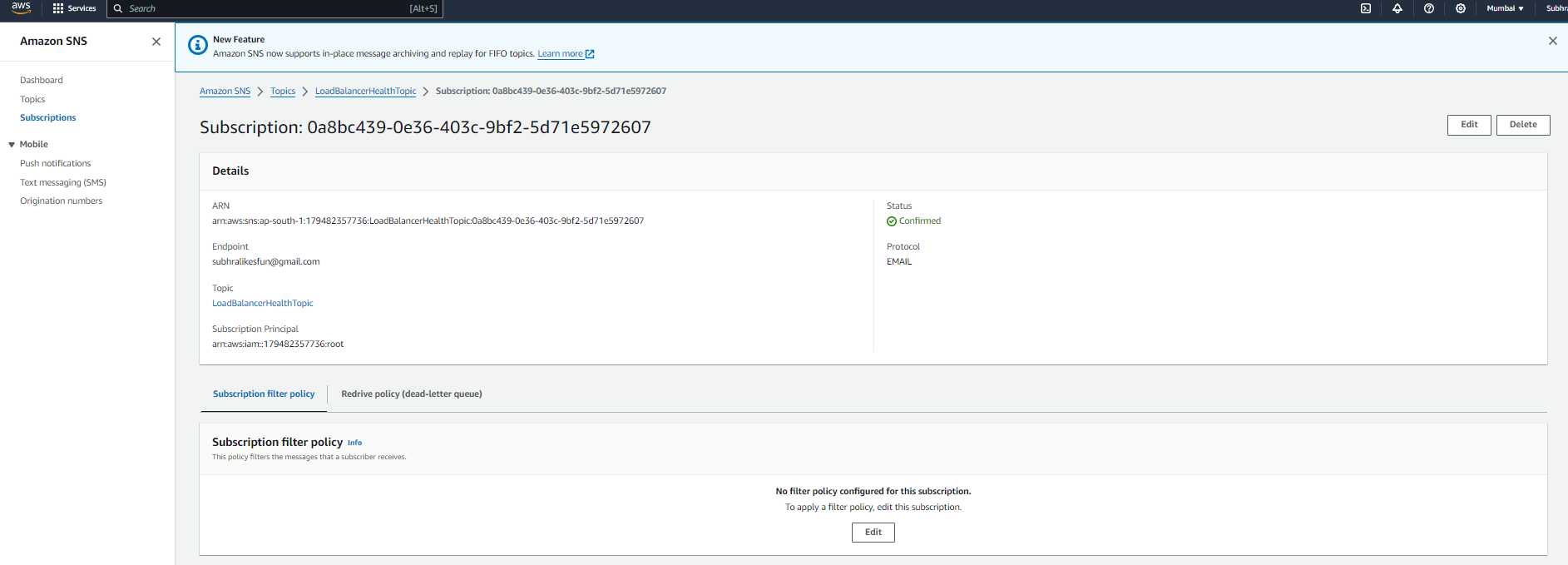
Choose "Lambda function" as the target type.

Choose the Lambda function you created for the Load Balancer Health Checker.

Click "Configure details".

Provide a name for your rule and a description.

Click "Create rule".

**Step 4: Set Up SNS Resources Manually  
  
  
**

Create an SNS Topic:

Go to the AWS Management Console.

Navigate to the Amazon SNS service.

Click on "Topics" in the left navigation pane.

Click the "Create topic" button.

Enter a name for your topic and click "Create topic".

Note the ARN of the created SNS topic.

Create an IAM Role for Lambda:

**Step 5. Test with Unhealthy Instances:**

Intentionally make an instance behind your ELB unhealthy (e.g., stop the instance).

Manually trigger your Lambda function or wait for the scheduled CloudWatch event.

Check the CloudWatch Logs and SNS notifications to ensure that the Lambda function correctly identifies and reports the unhealthy instance  
  
