**Assignment 6: Monitor and Alert High AWS Billing Using AWS Lambda, Boto3, and SNS**

**Objective:** Create an automated alerting mechanism for when your AWS billing exceeds a certain threshold.

**Task:** Set up a Lambda function to check your AWS billing amount daily, and if it exceeds a specified threshold, send an alert via SNS.

**Instructions:**

1. SNS Setup:

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

   - Subscribe your email to this topic.

2. Lambda IAM Role:

   - In the IAM dashboard, create a new role for Lambda.

   - Attach policies that allow reading CloudWatch metrics and sending SNS notifications.

3. Lambda Function:

   - Navigate to the Lambda dashboard and create a new function.

   - Choose Python 3.x as the runtime.

   - Assign the IAM role created in the previous step.

   - Write the Boto3 Python script to:

     1. Initialize boto3 clients for CloudWatch and SNS.

     2. Retrieve the AWS billing metric from CloudWatch.

     3. Compare the billing amount with a threshold (e.g., $50).

     4. If the billing exceeds the threshold, send an SNS notification.

     5. Print messages for logging purposes.

4. Event Source (Bonus):

   - Attach an event source, like Amazon CloudWatch Events, to trigger the Lambda function daily.

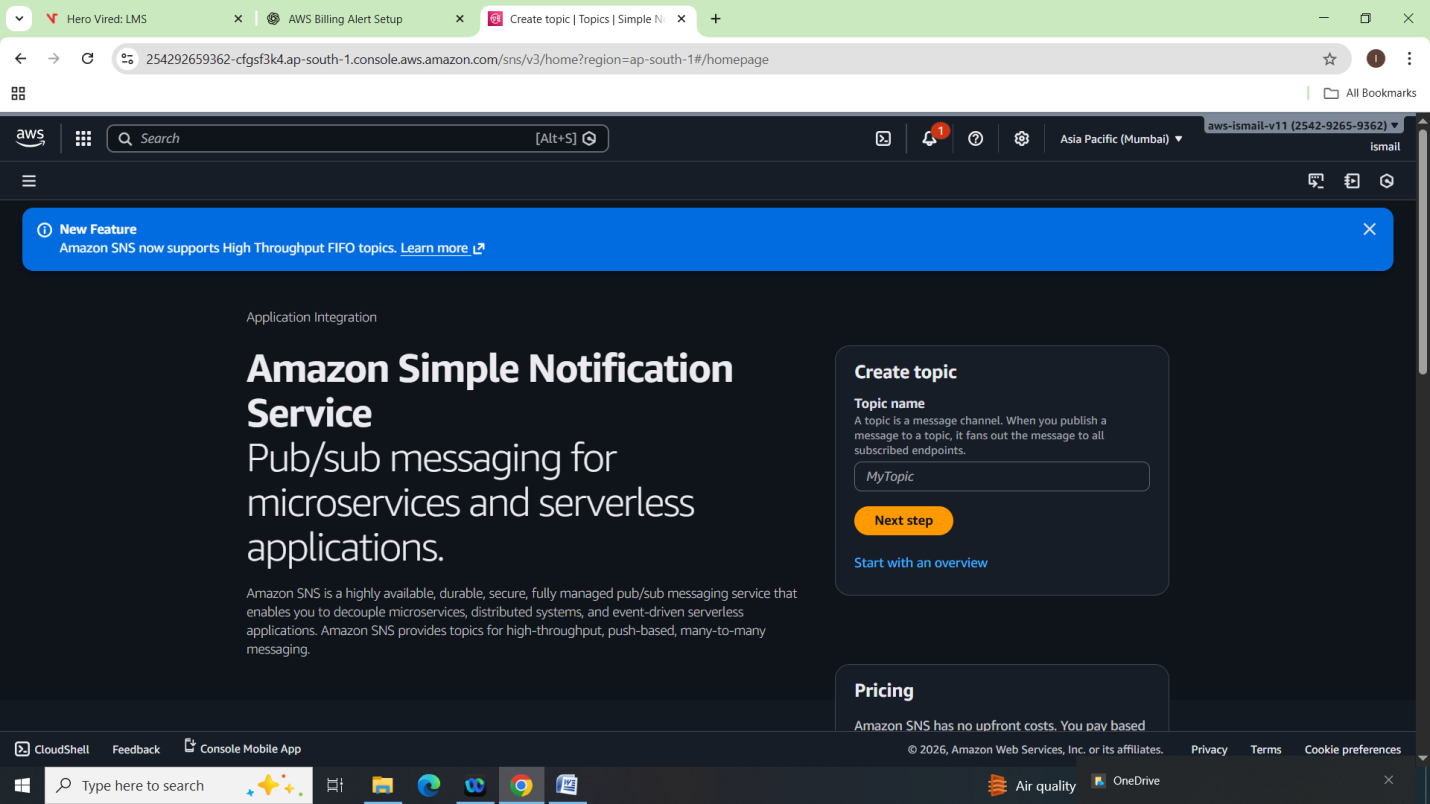
5. Testing:

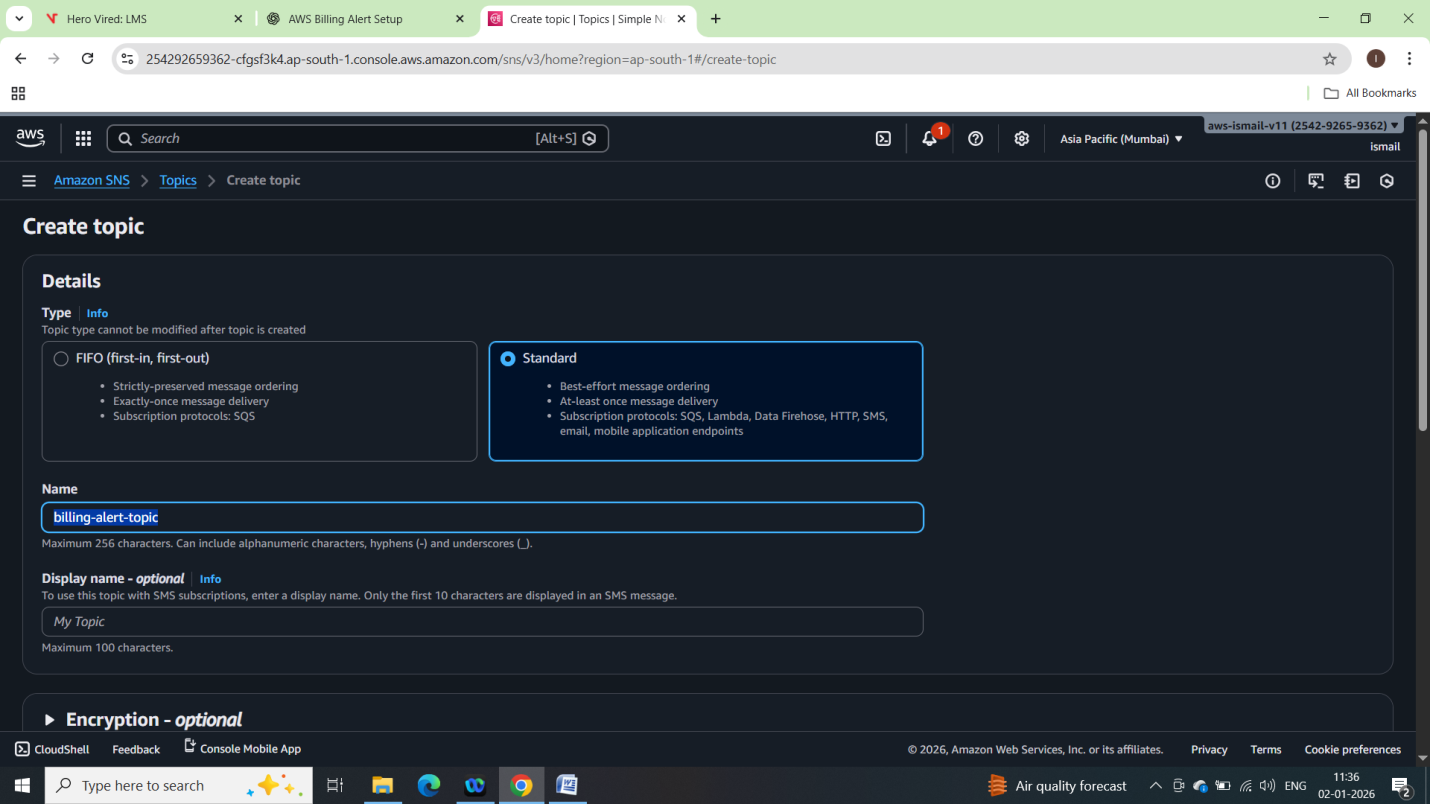
   - Manually trigger the Lambda function or wait for the scheduled event.

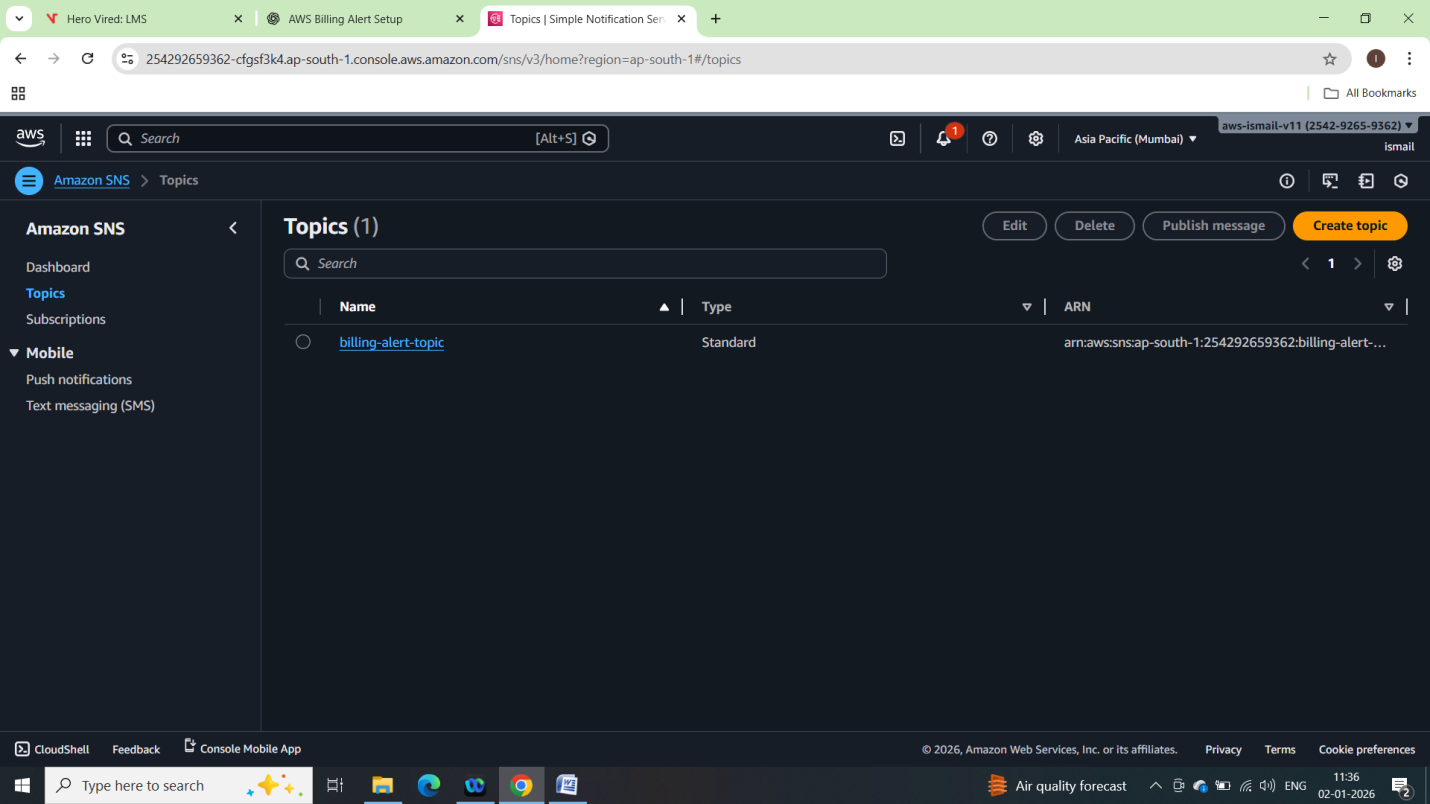
   - If your billing is over the threshold, you should receive an email alert.

Project solution:

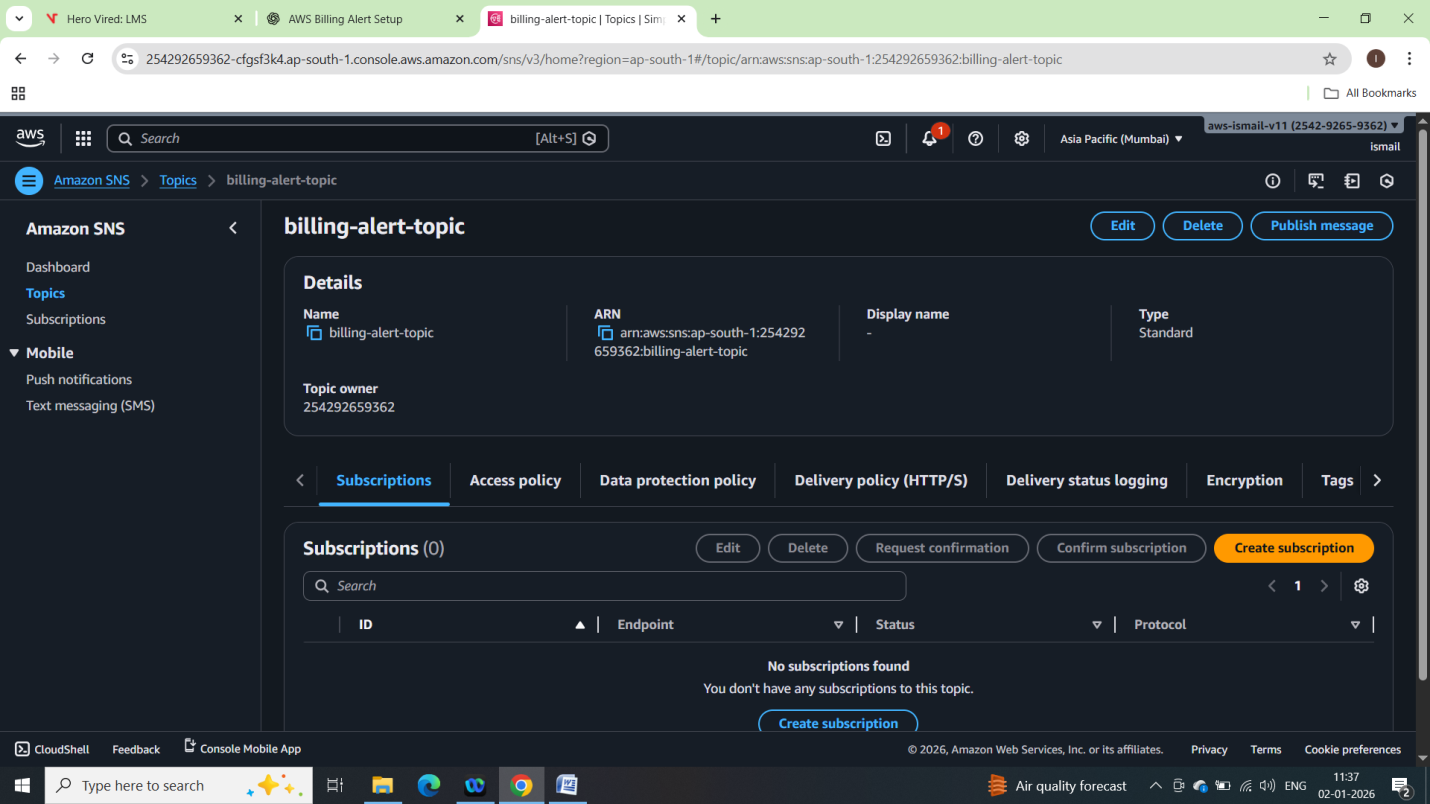
SNS Setup (Alert Mechanism)

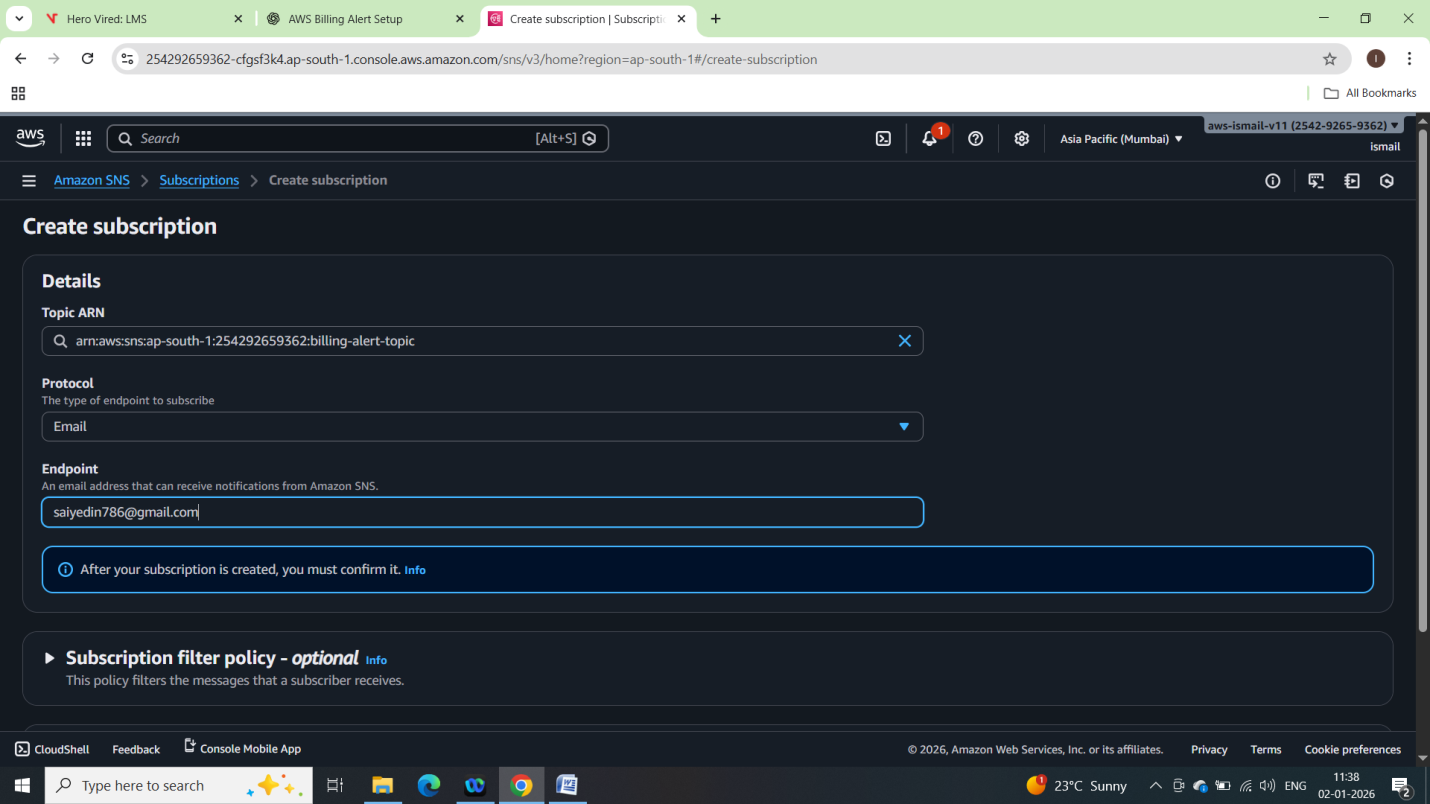


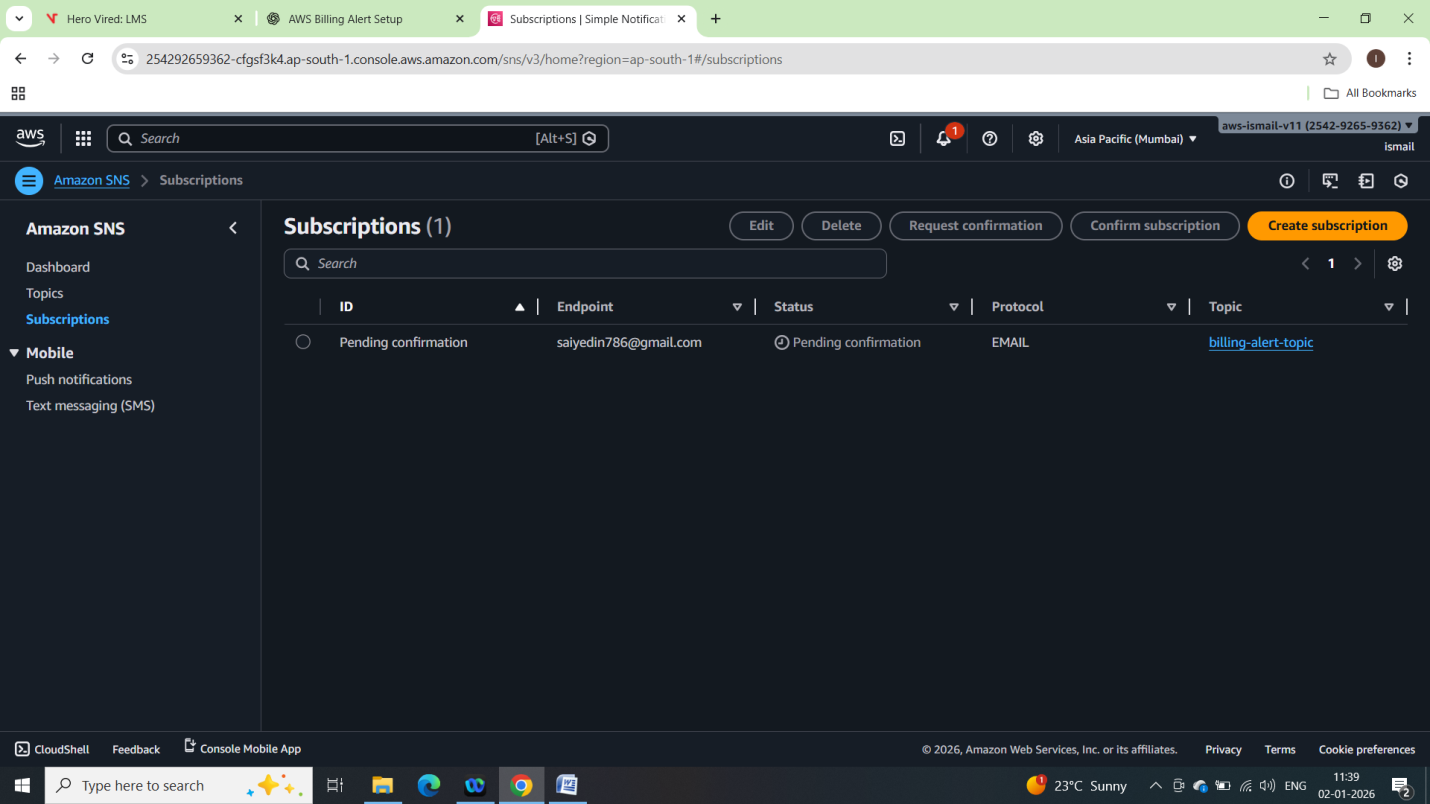


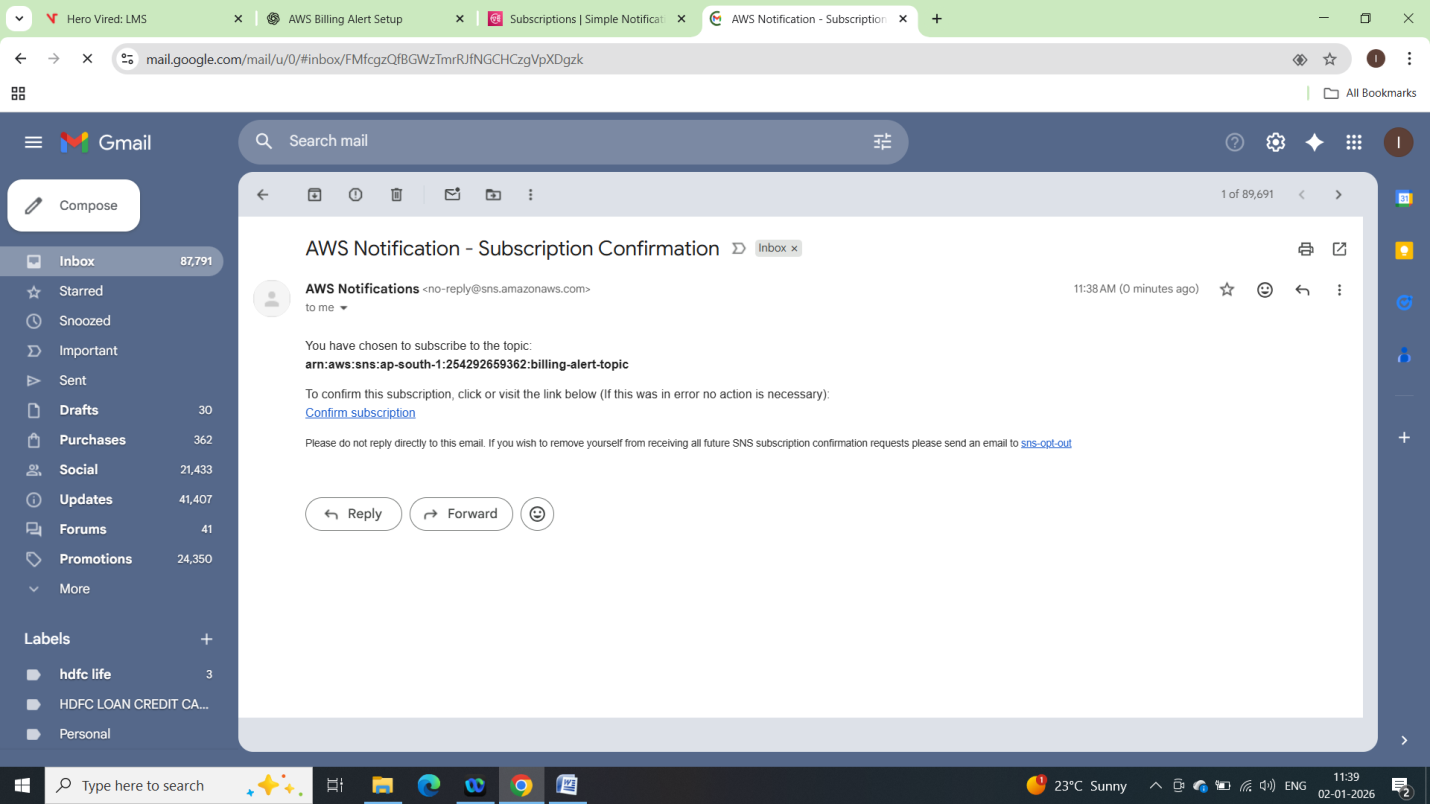


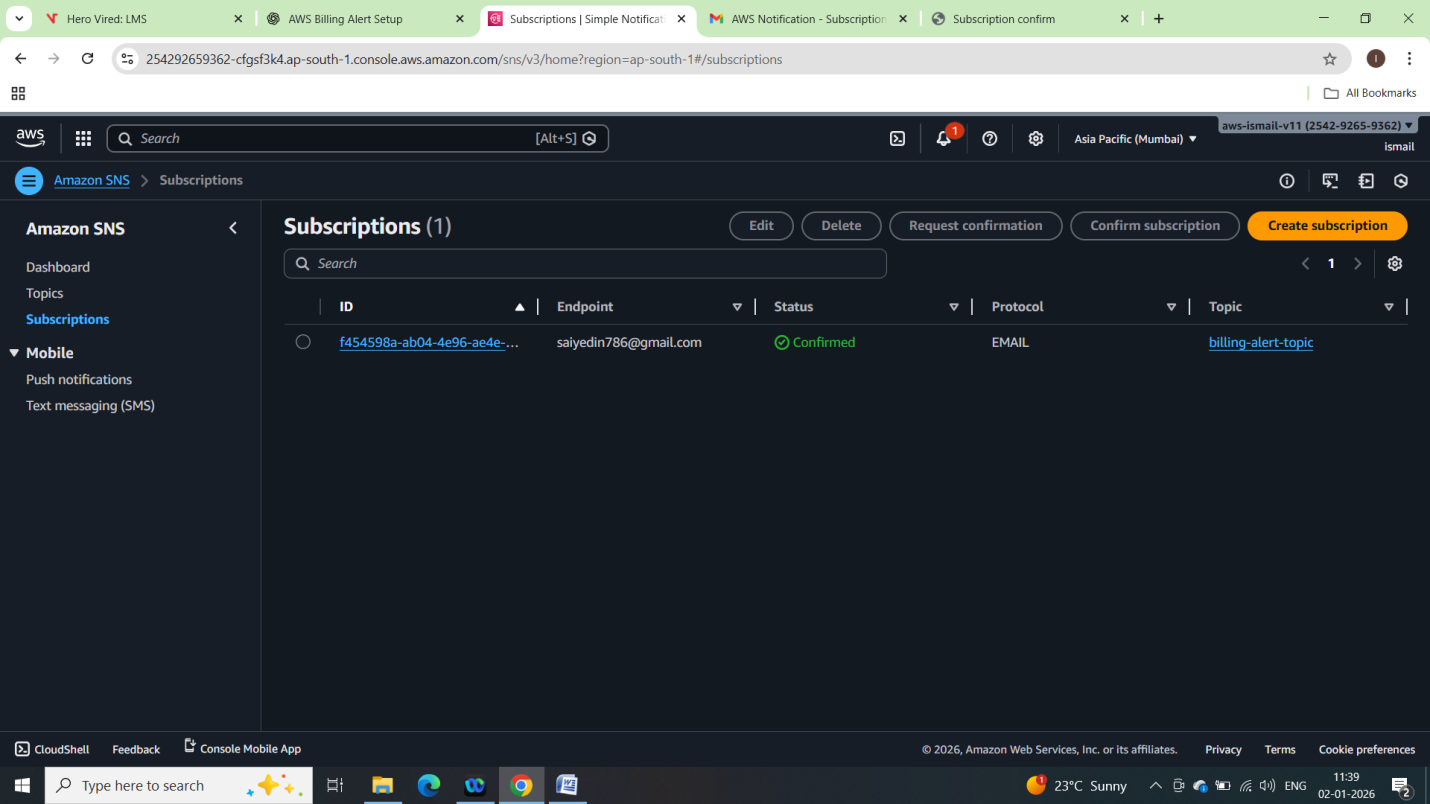
Subscribing Email:







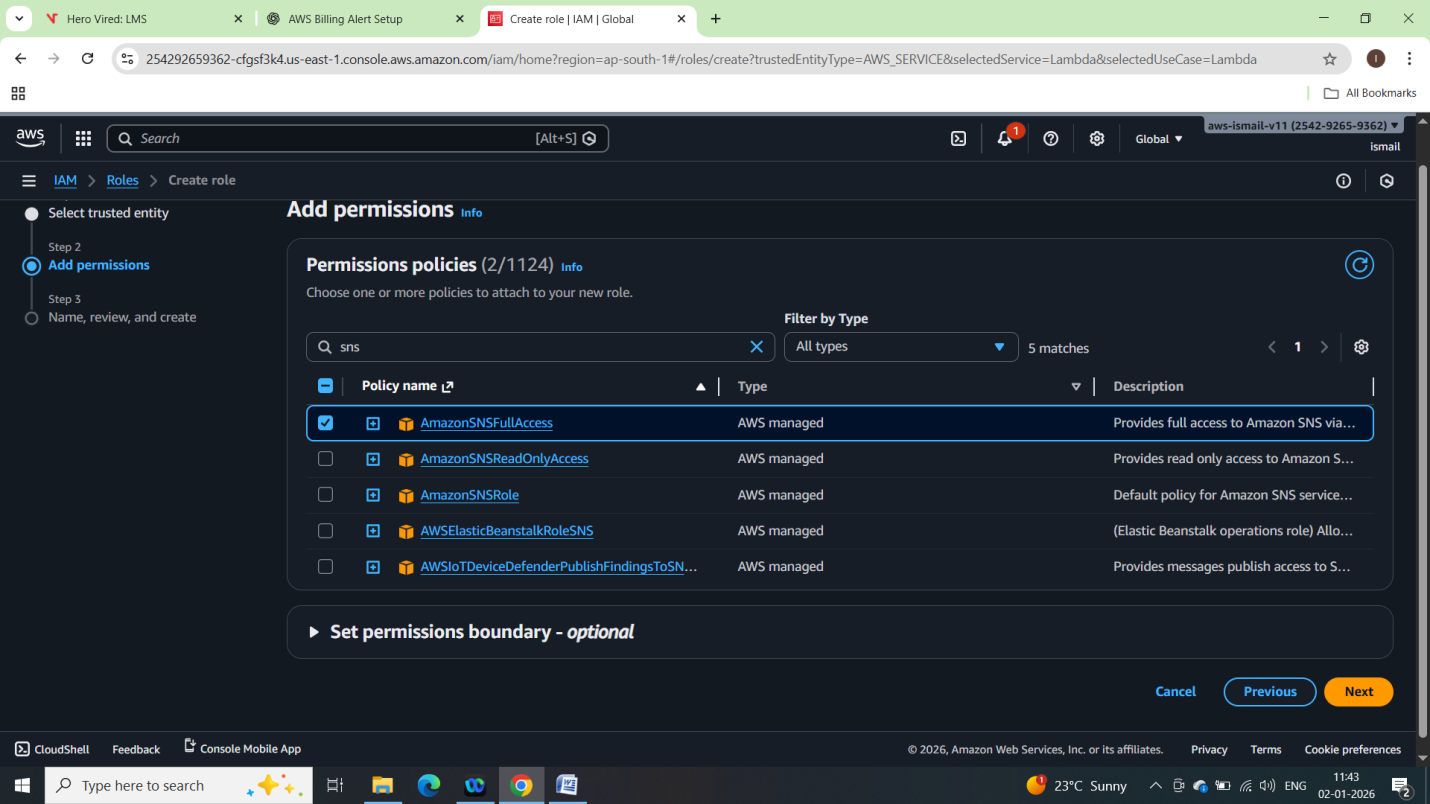


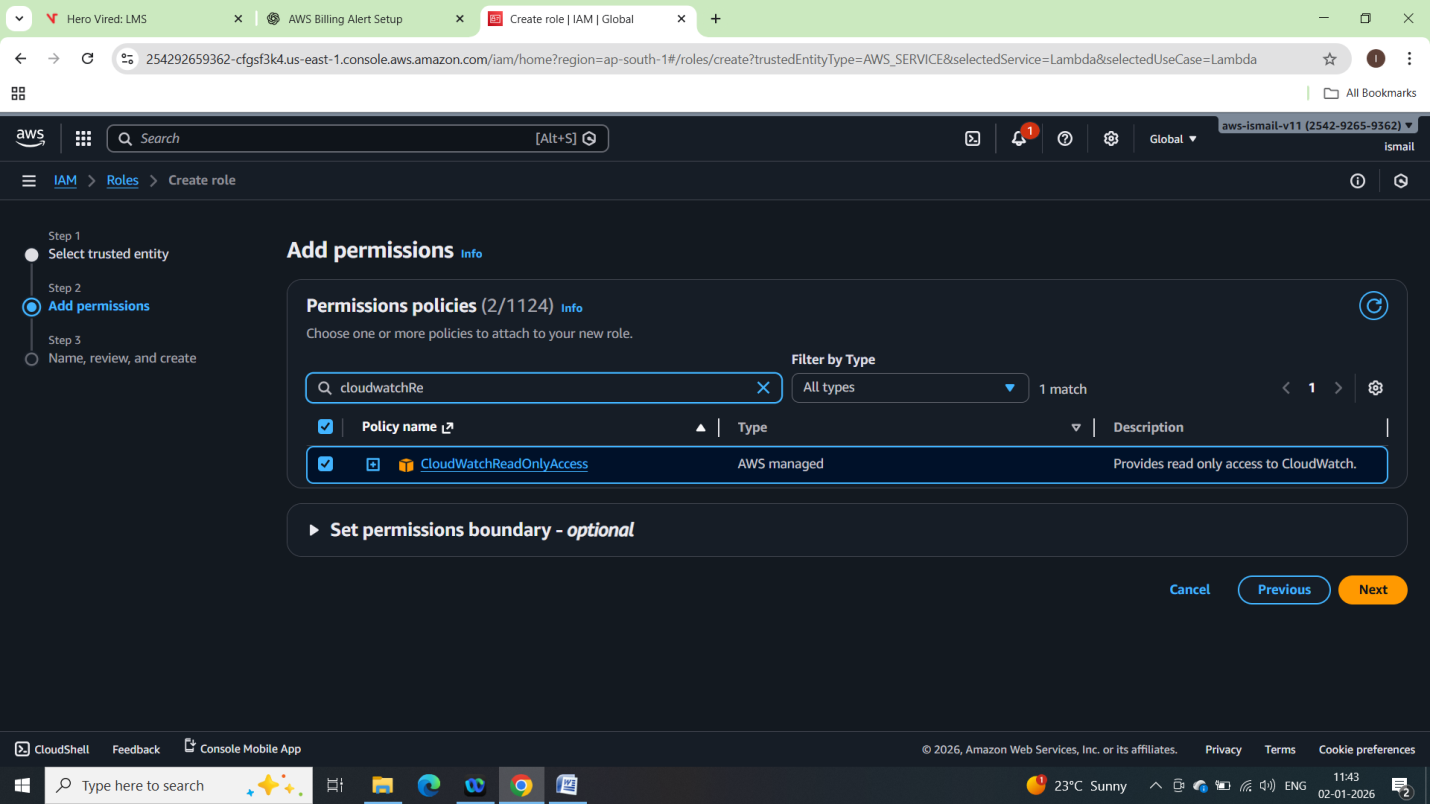


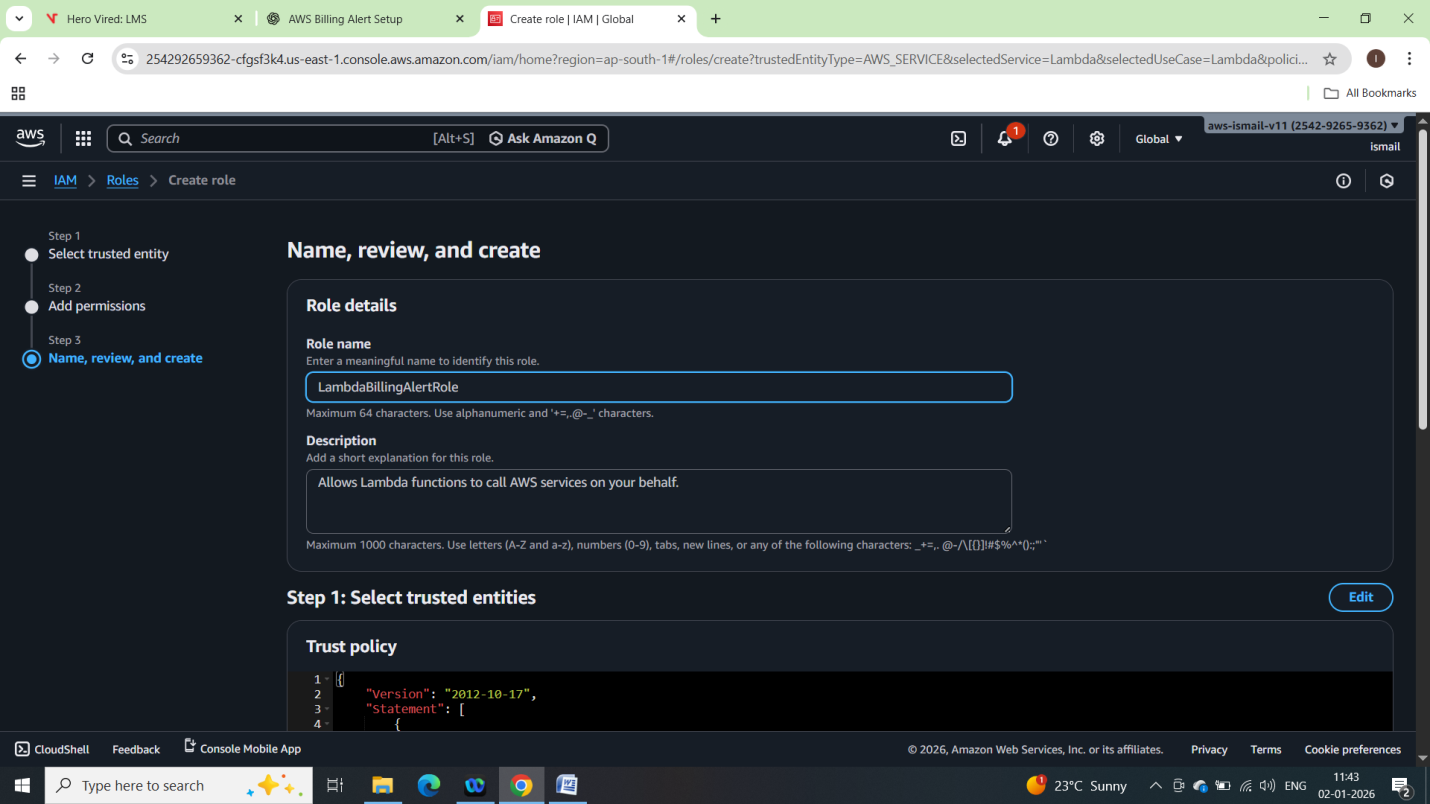
Lambda IAM Role Setup:

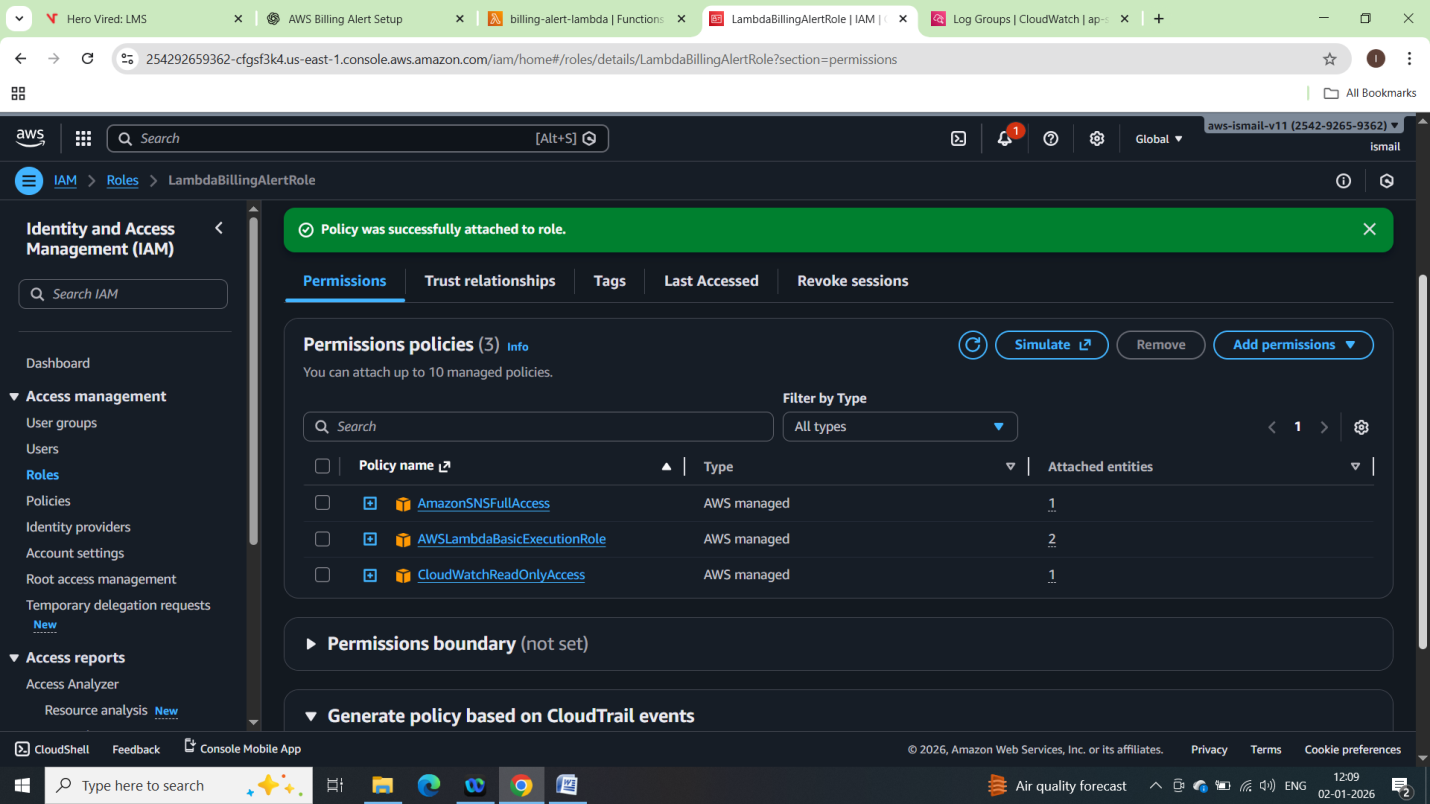
Creating IAM Role

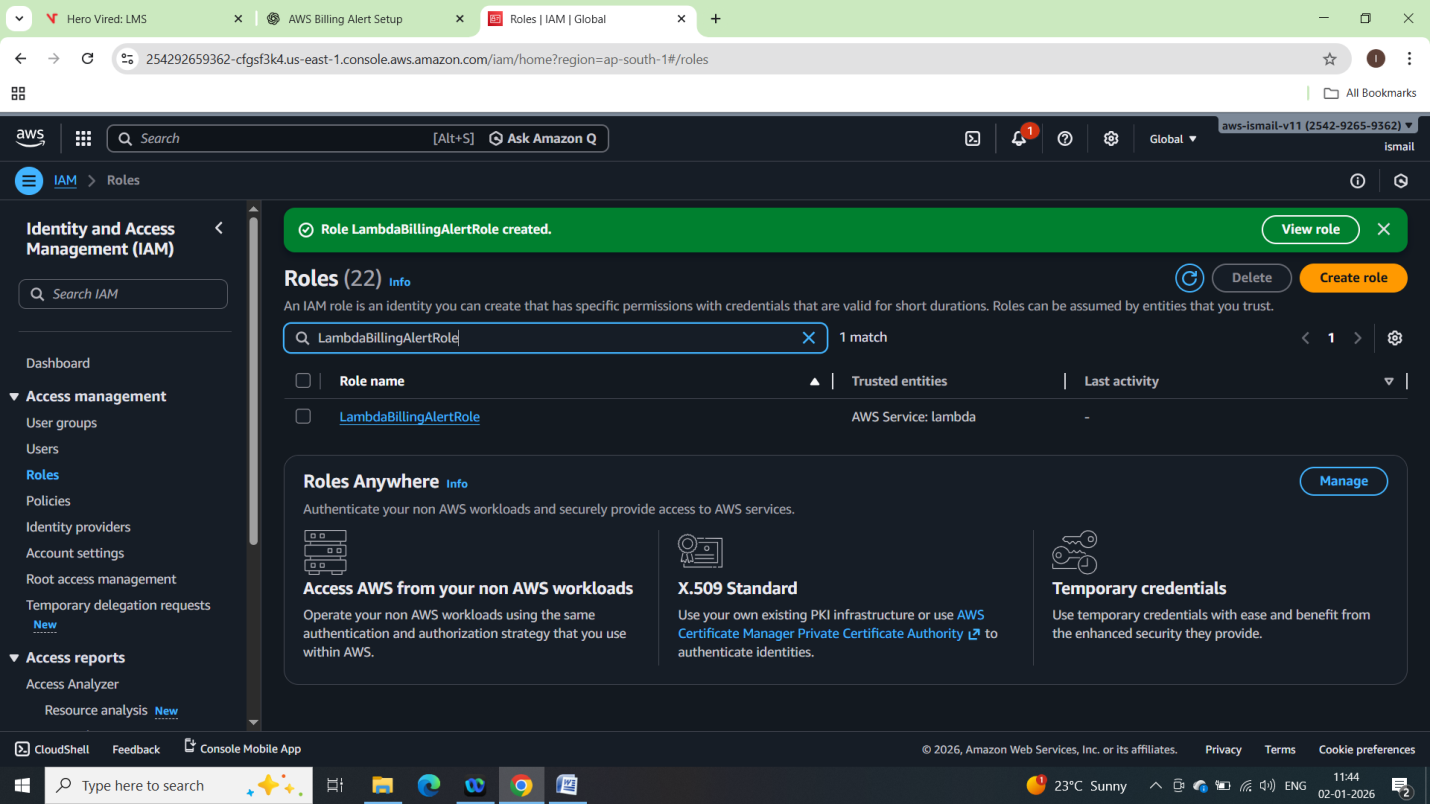




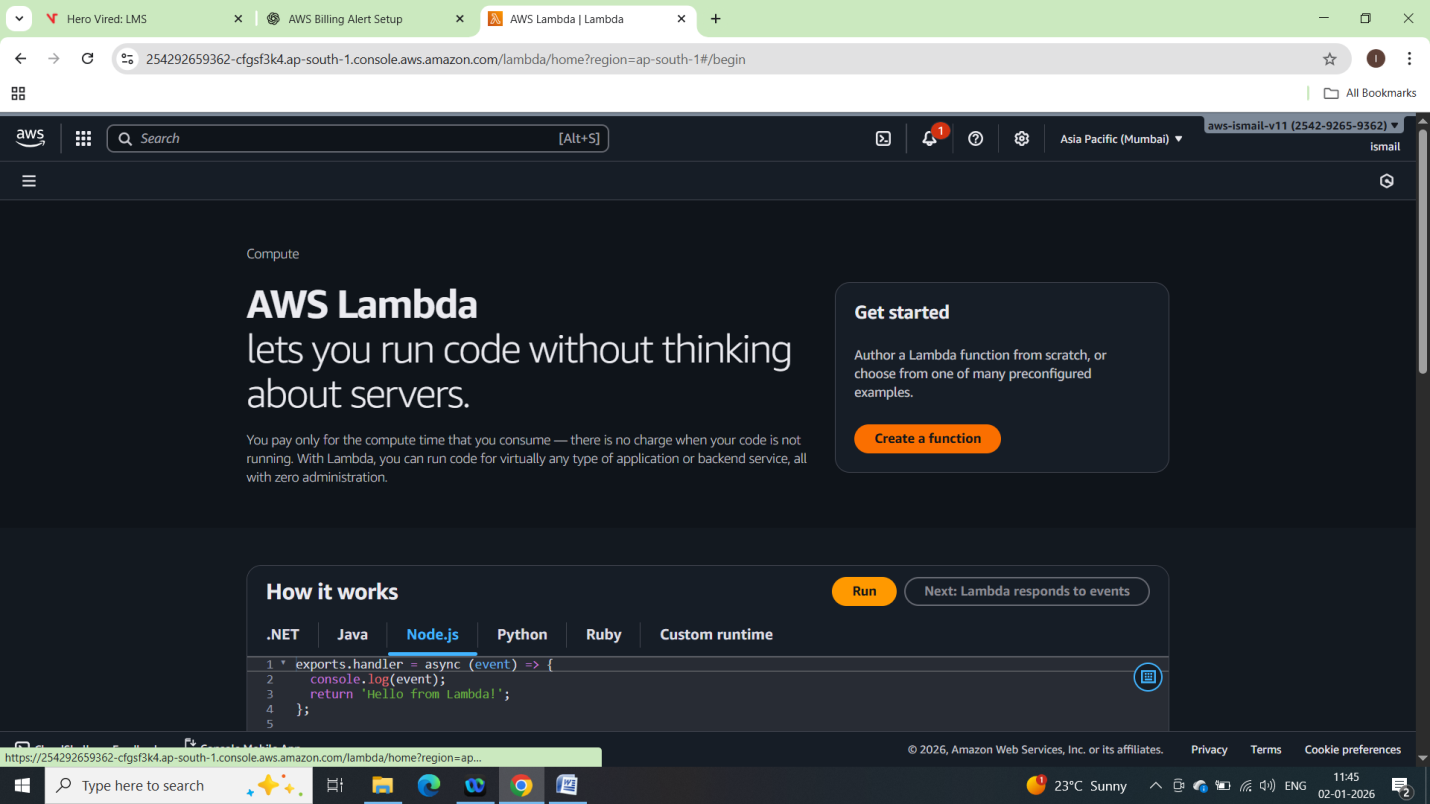


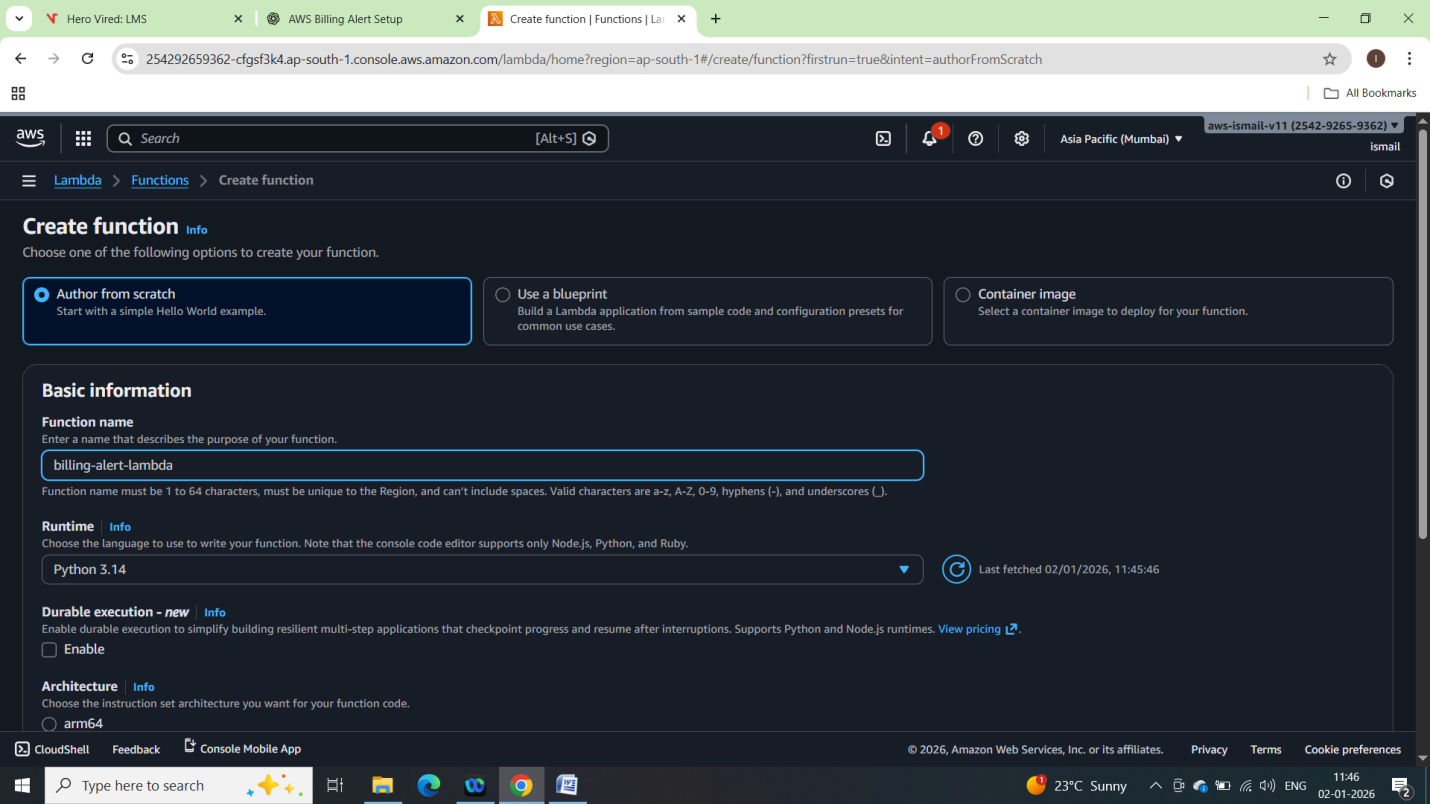


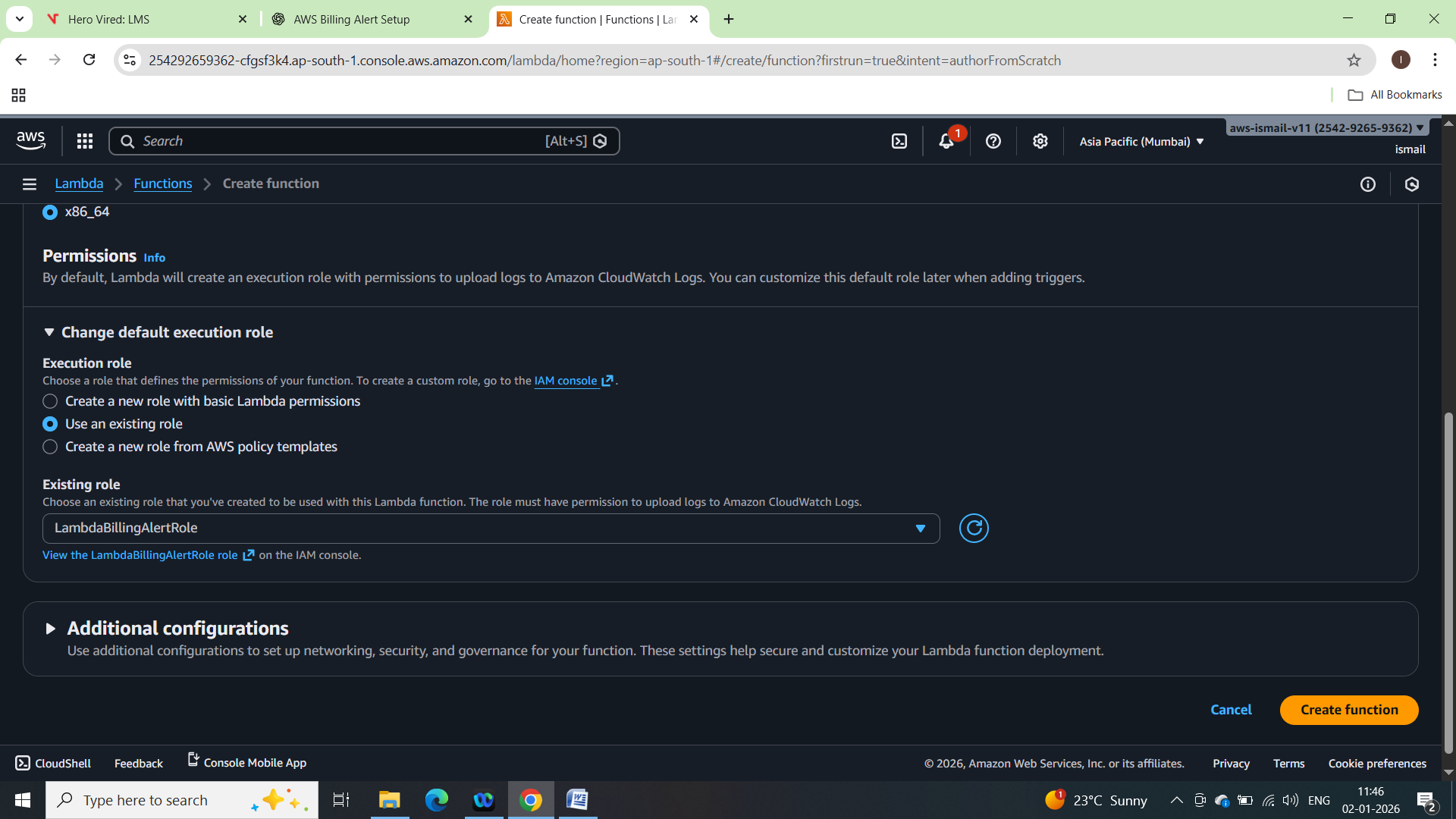


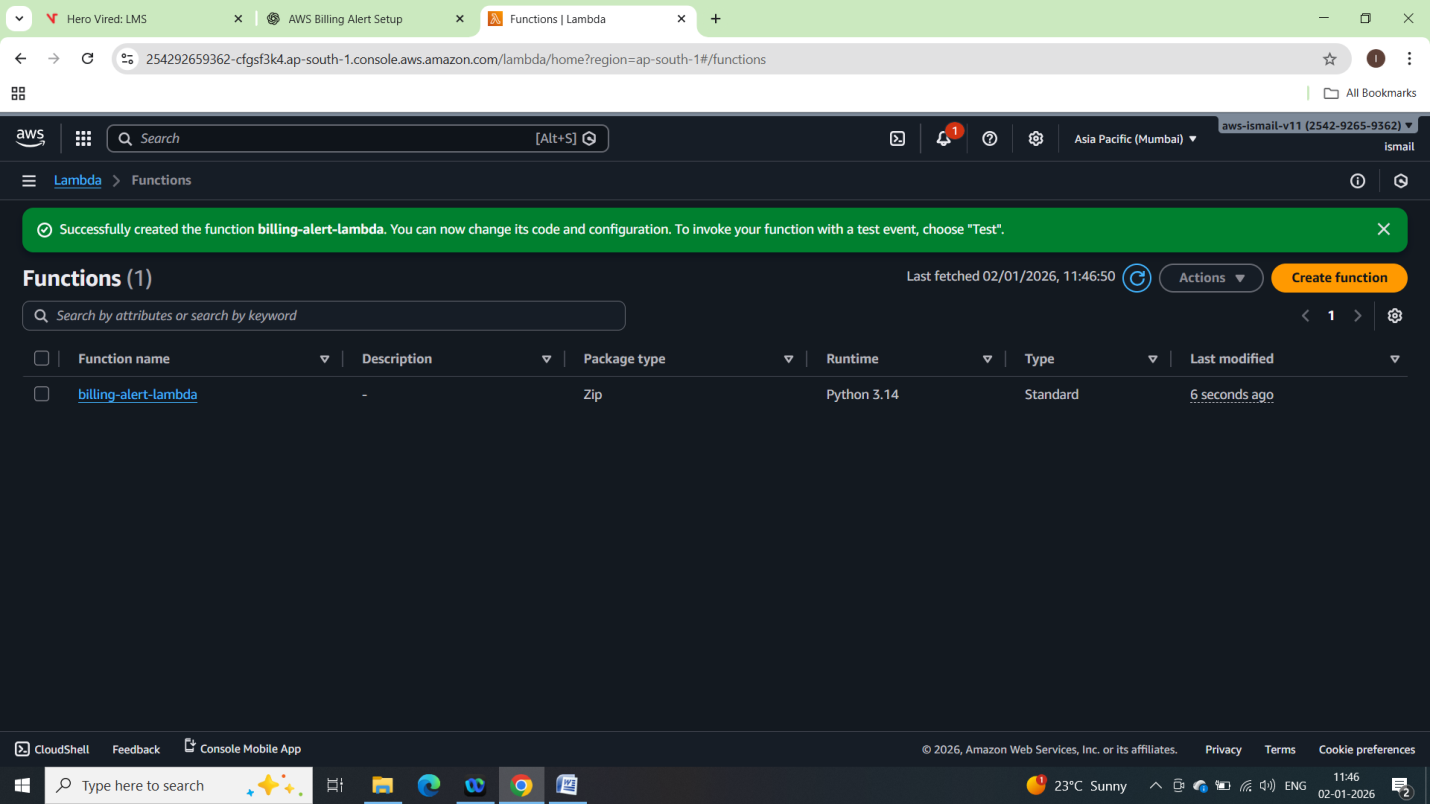


Lambda Function Setup:









Boto3-python Code for

import boto3

from datetime import datetime, timedelta

# ---------------- CONFIG ----------------

THRESHOLD = 50.0 # USD

SNS\_TOPIC\_ARN = "arn:aws:sns:ap-south-1:254292659362:billing-alert-topic"

# ----------------------------------------

def lambda\_handler(event, context):

# CloudWatch Billing metrics are only in us-east-1

cloudwatch = boto3.client("cloudwatch", region\_name="us-east-1")

sns = boto3.client("sns")

# Time range (last 1 day)

end\_time = datetime.utcnow()

start\_time = end\_time - timedelta(days=1)

# Get billing metric

response = cloudwatch.get\_metric\_statistics(

Namespace="AWS/Billing",

MetricName="EstimatedCharges",

Dimensions=[

{

"Name": "Currency",

"Value": "USD"

}

],

StartTime=start\_time,

EndTime=end\_time,

Period=86400,

Statistics=["Maximum"]

)

if not response["Datapoints"]:

print("No billing data available")

return

billing\_amount = response["Datapoints"][0]["Maximum"]

print(f"Current estimated billing: ${billing\_amount}")

# Compare with threshold

if billing\_amount > THRESHOLD:

message = (

f"AWS Billing Alert!\n\n"

f"Current estimated charges: ${billing\_amount}\n"

f"Threshold: ${THRESHOLD}\n"

)

sns.publish(

TopicArn=SNS\_TOPIC\_ARN,

Subject="AWS Billing Alert",

Message=message

)

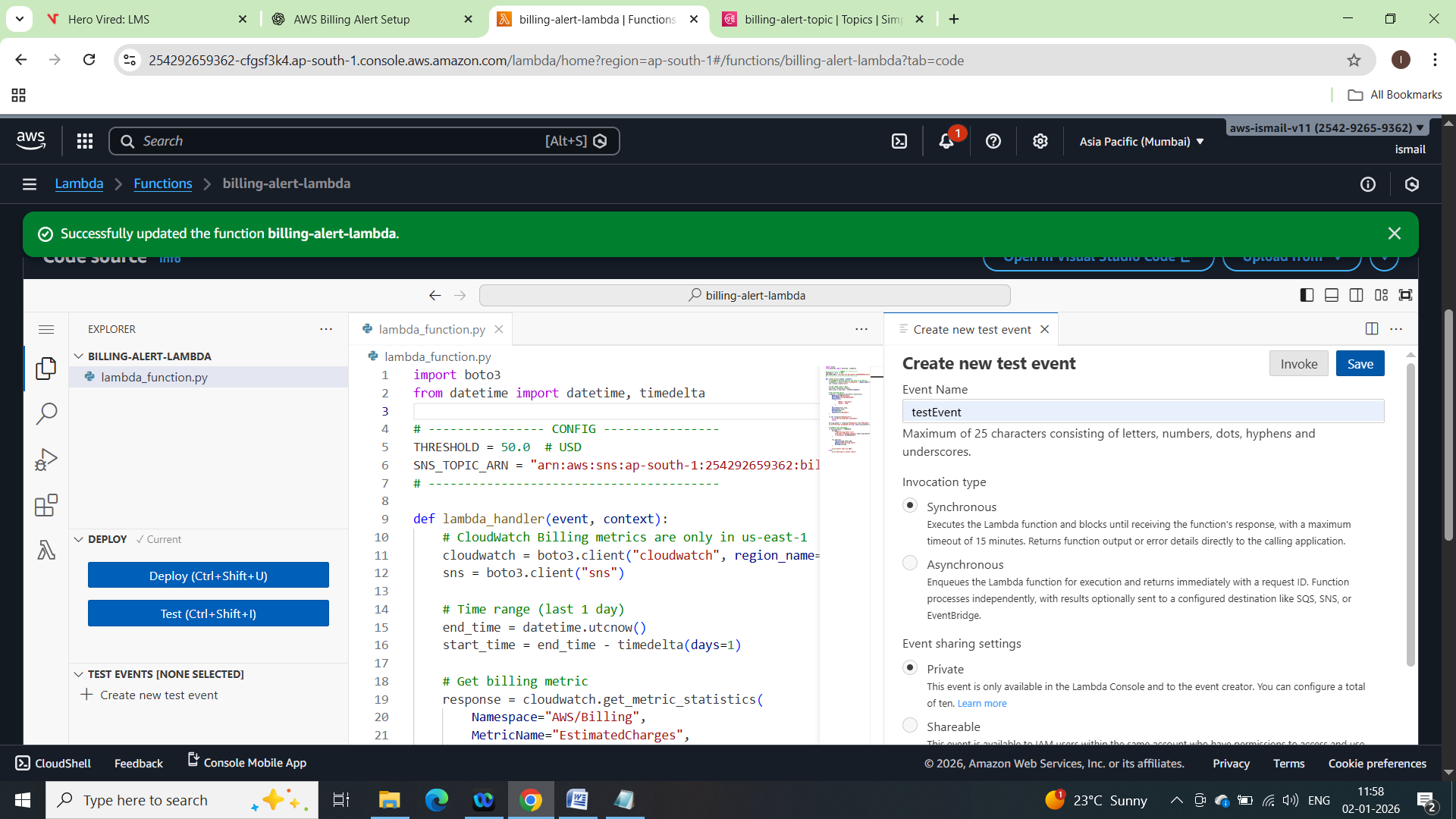
print("Alert sent via SNS")

else:

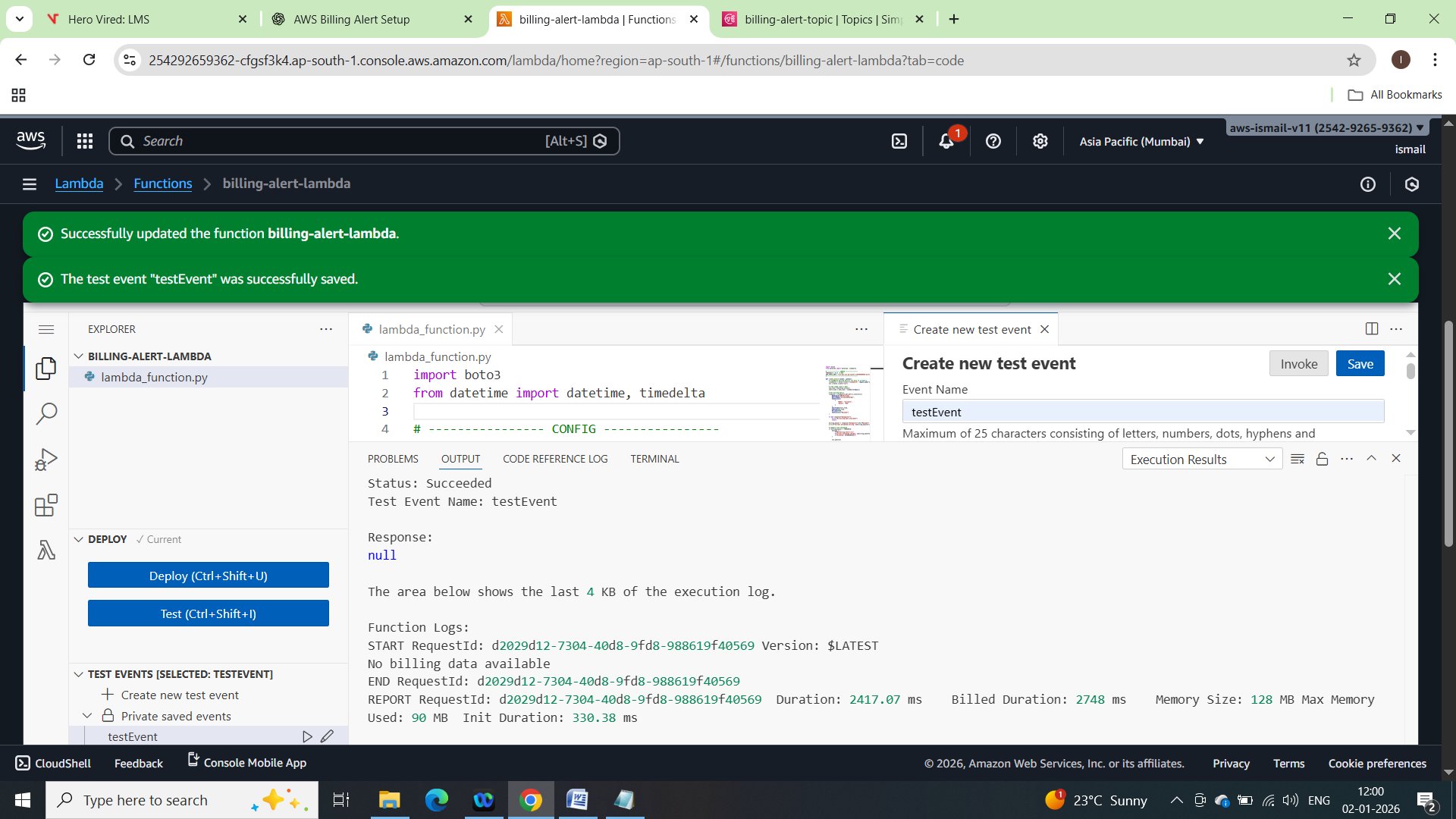
print("Billing is within limit")



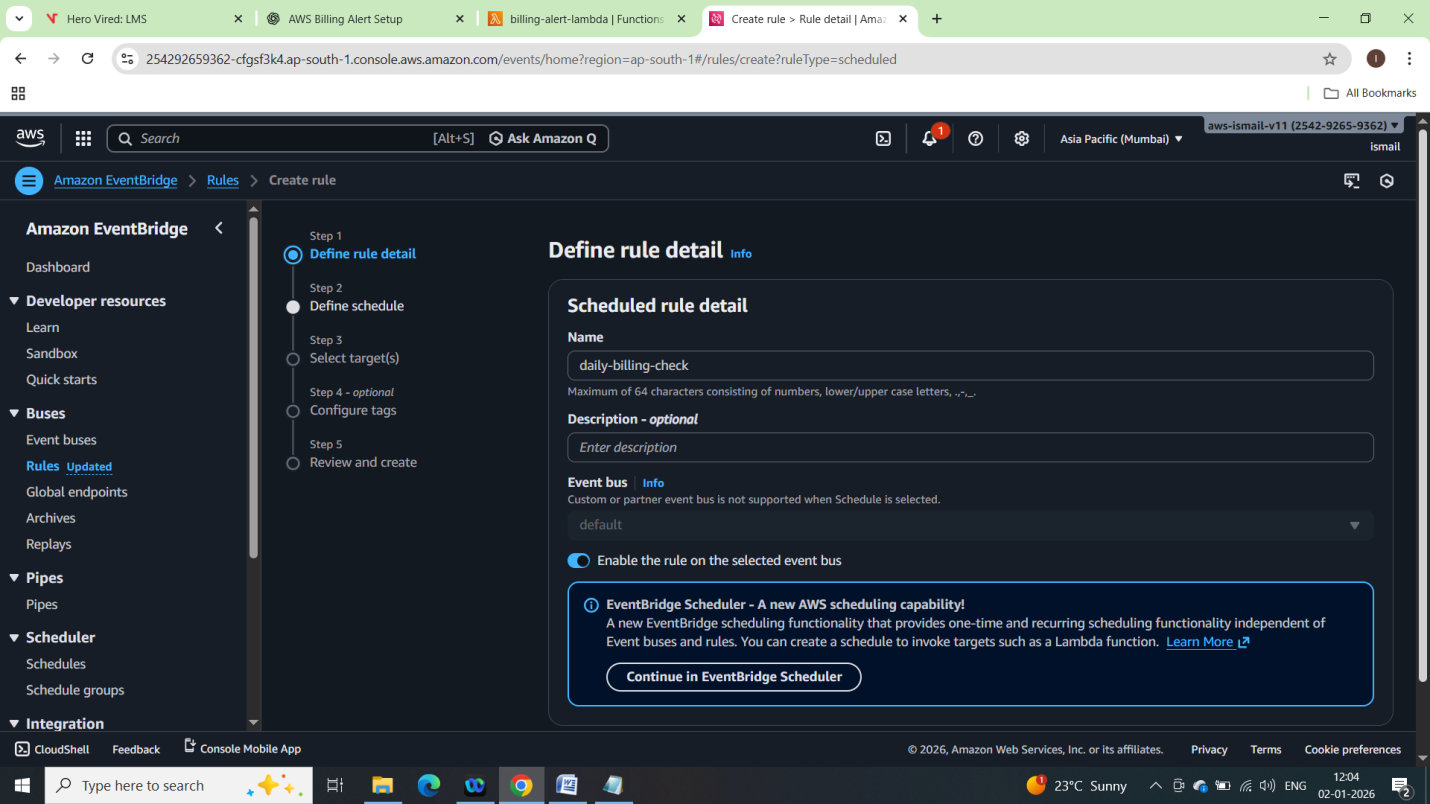
Creating a test event

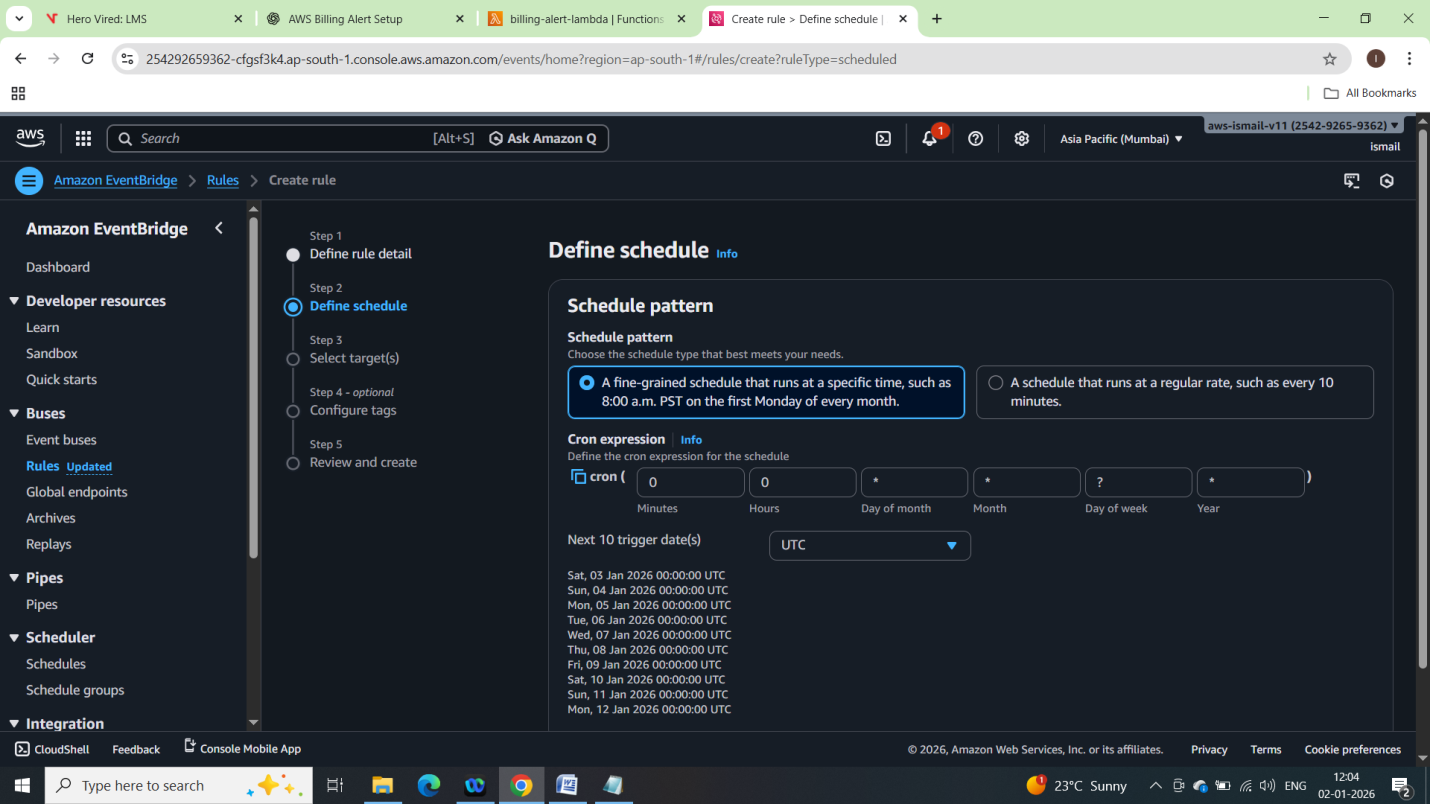


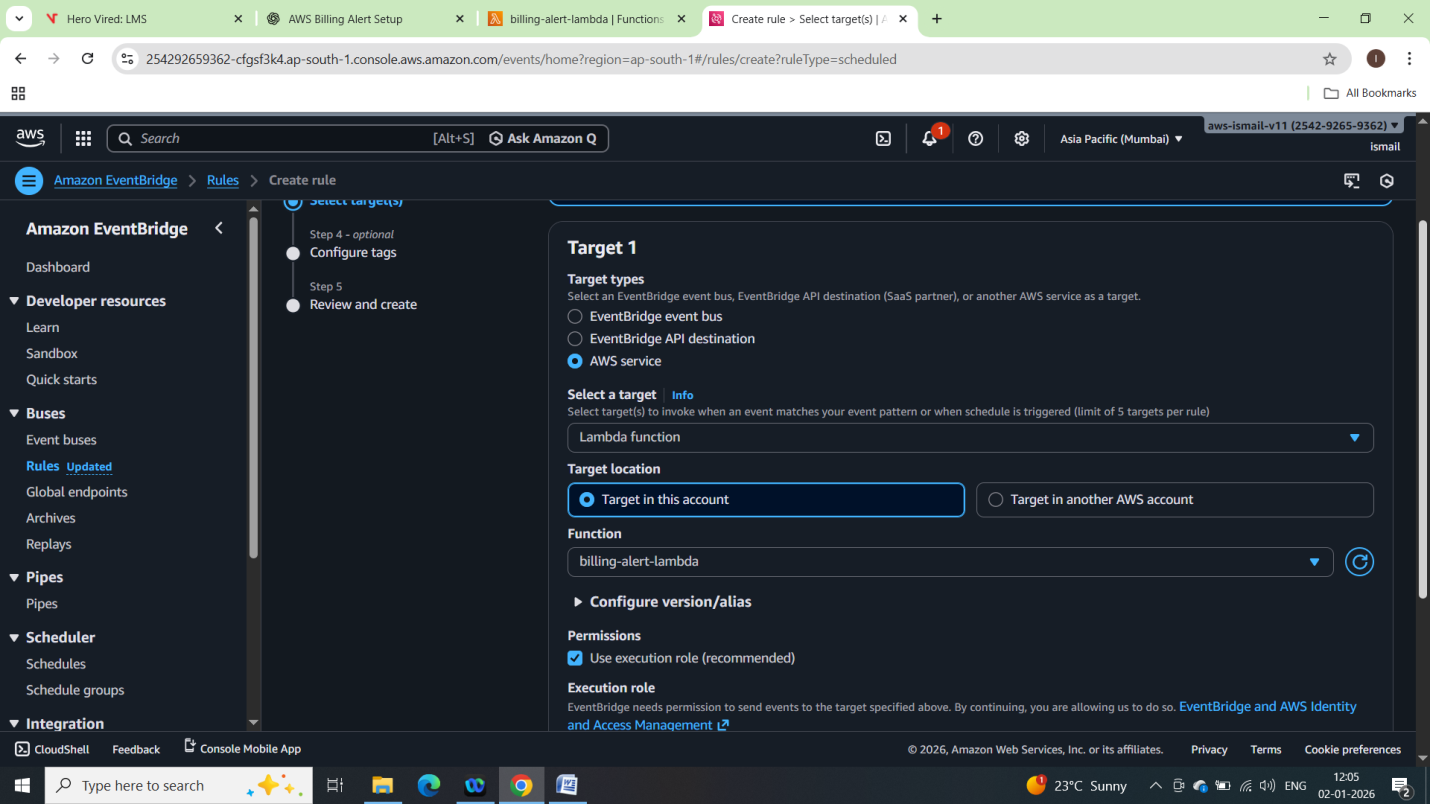
Testing with testEvent

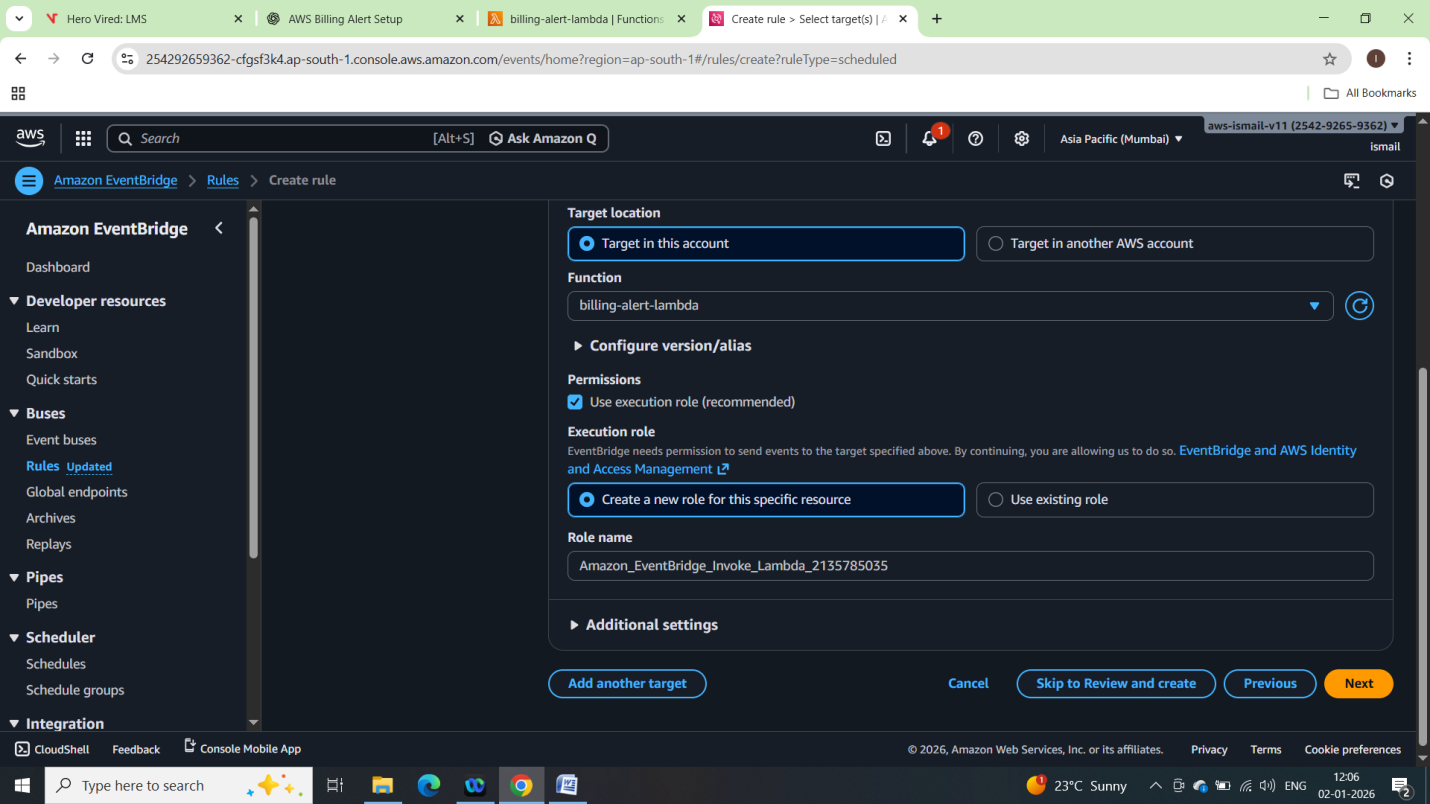


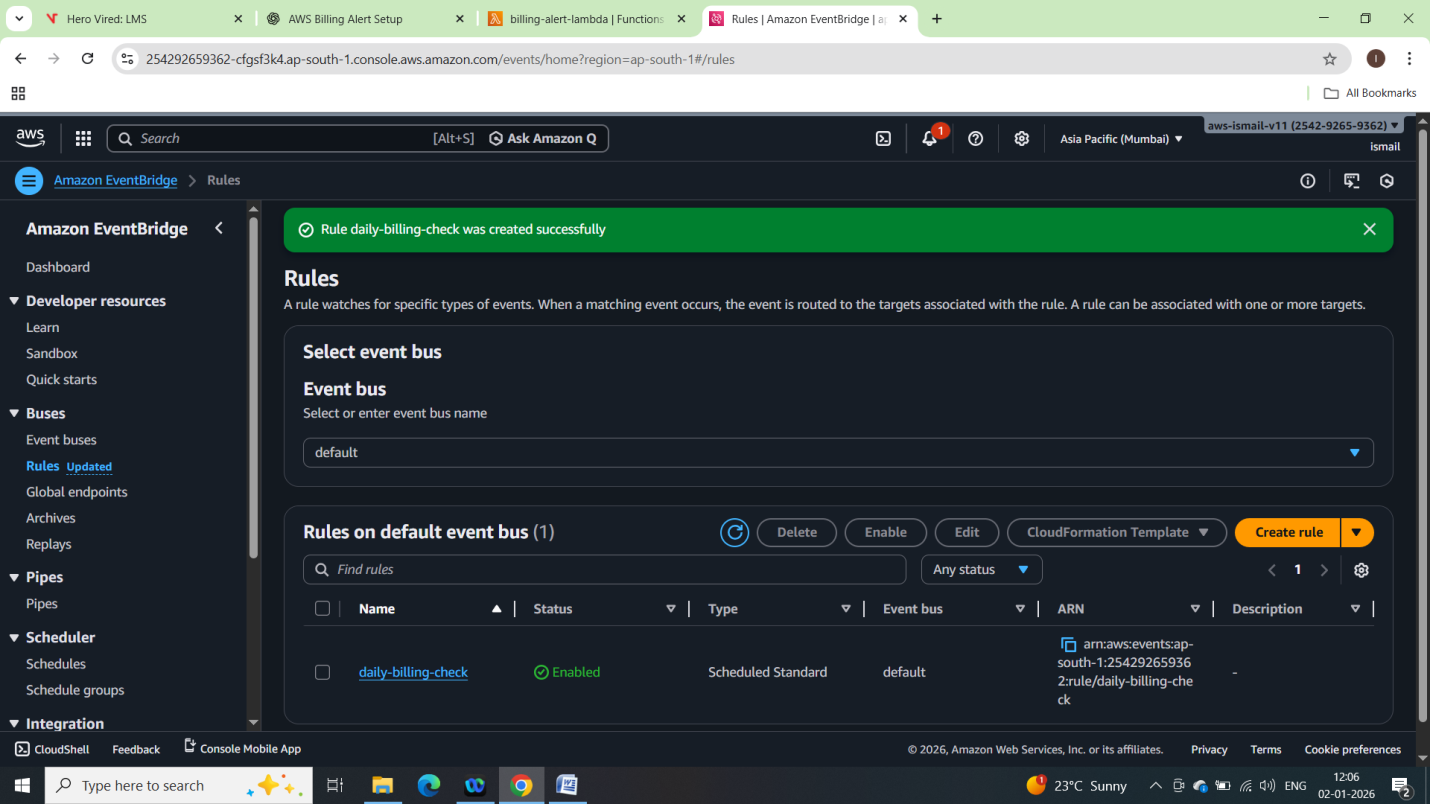
Create EventBridge Rule:











Triggering manually the lambda function:

