**Assignment 8: Analyze Sentiment of User Reviews Using AWS Lambda, Boto3, and Amazon Comprehend**

**Objective**: Automatically analyze and categorize the sentiment of user reviews using Amazon Comprehend.

**Task:** Set up a Lambda function to receive user reviews, analyze their sentiment using Amazon Comprehend, and log the results.

**Instructions:**

1. Lambda IAM Role:

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

   - Attach policies that allow Lambda to use Amazon Comprehend.

2. Lambda Function:

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

   - Choose Python 3.x as the runtime.

   - Assign the IAM role created previously.

   - Write the Boto3 Python script to:

     1. Extract the user review from an event.

     2. Use Amazon Comprehend to analyze the sentiment of the review.

     3. Log the sentiment result.

3. Testing:

   - Manually trigger the Lambda function with sample reviews.

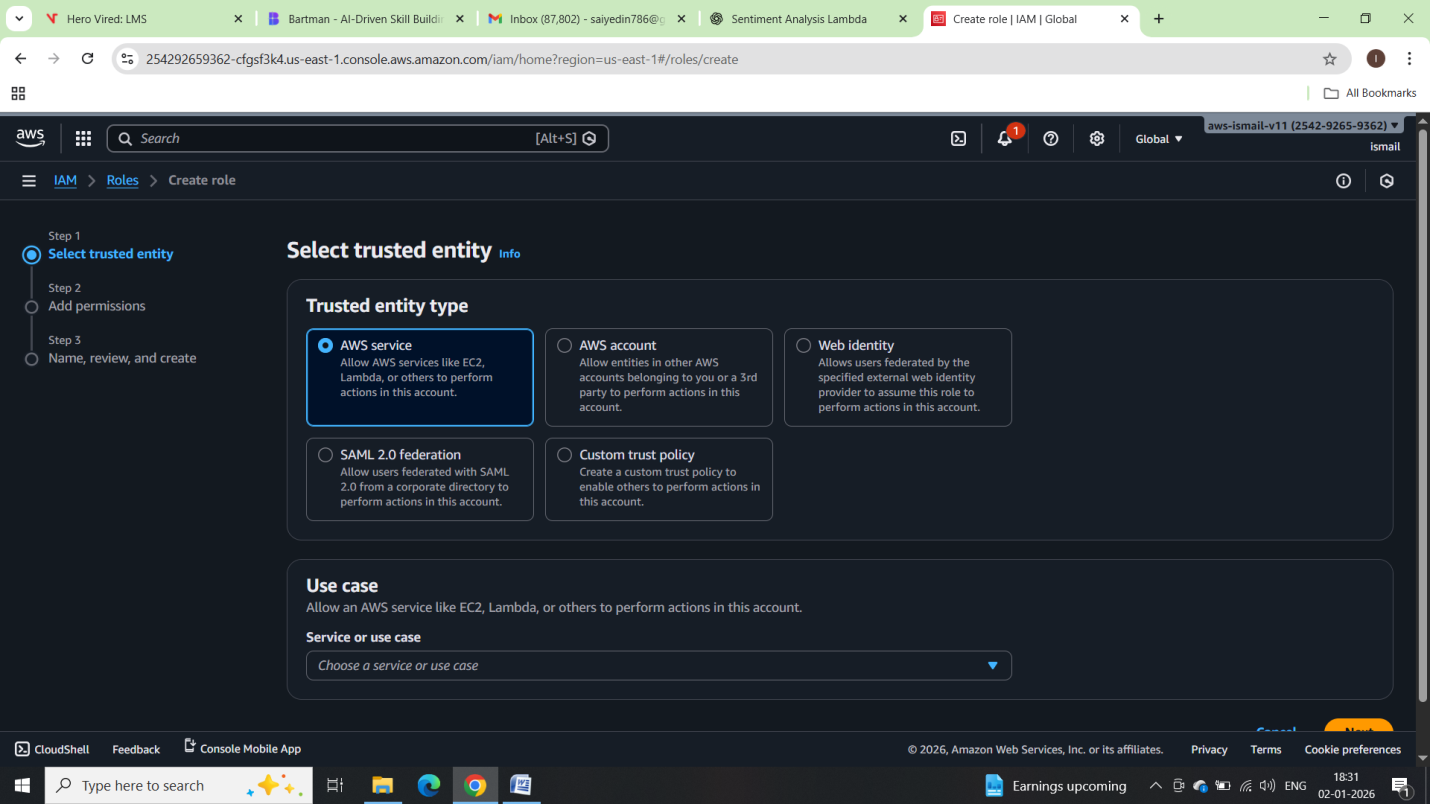
   - Confirm the sentiment analysis results in the Lambda logs.

Project Solution:

Create IAM Role for Lambda

**Step 1: Open IAM**

* Go to **IAM → Roles → Create role**
* Trusted entity: **AWS service**
* Use case: **Lambda**
* Click **Next**

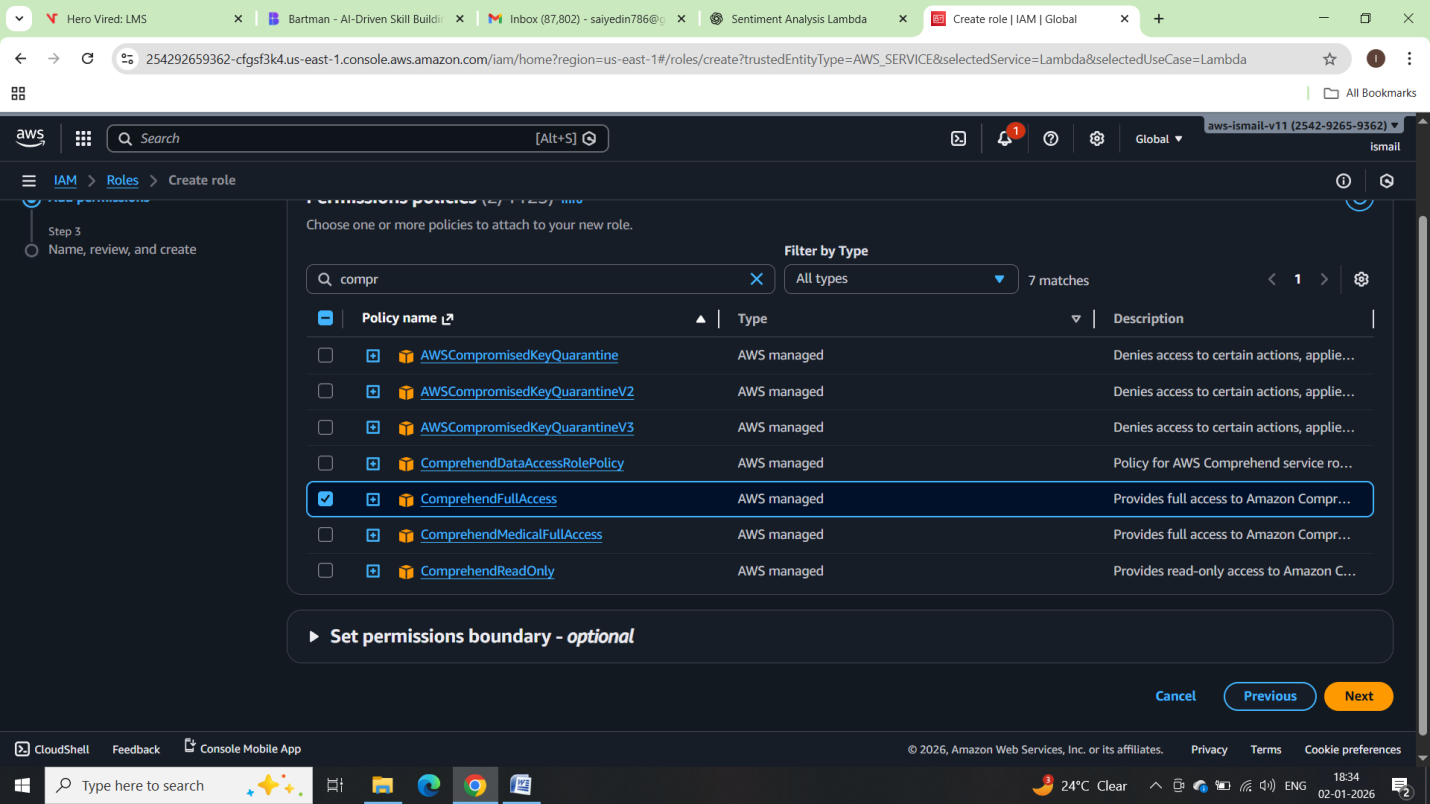


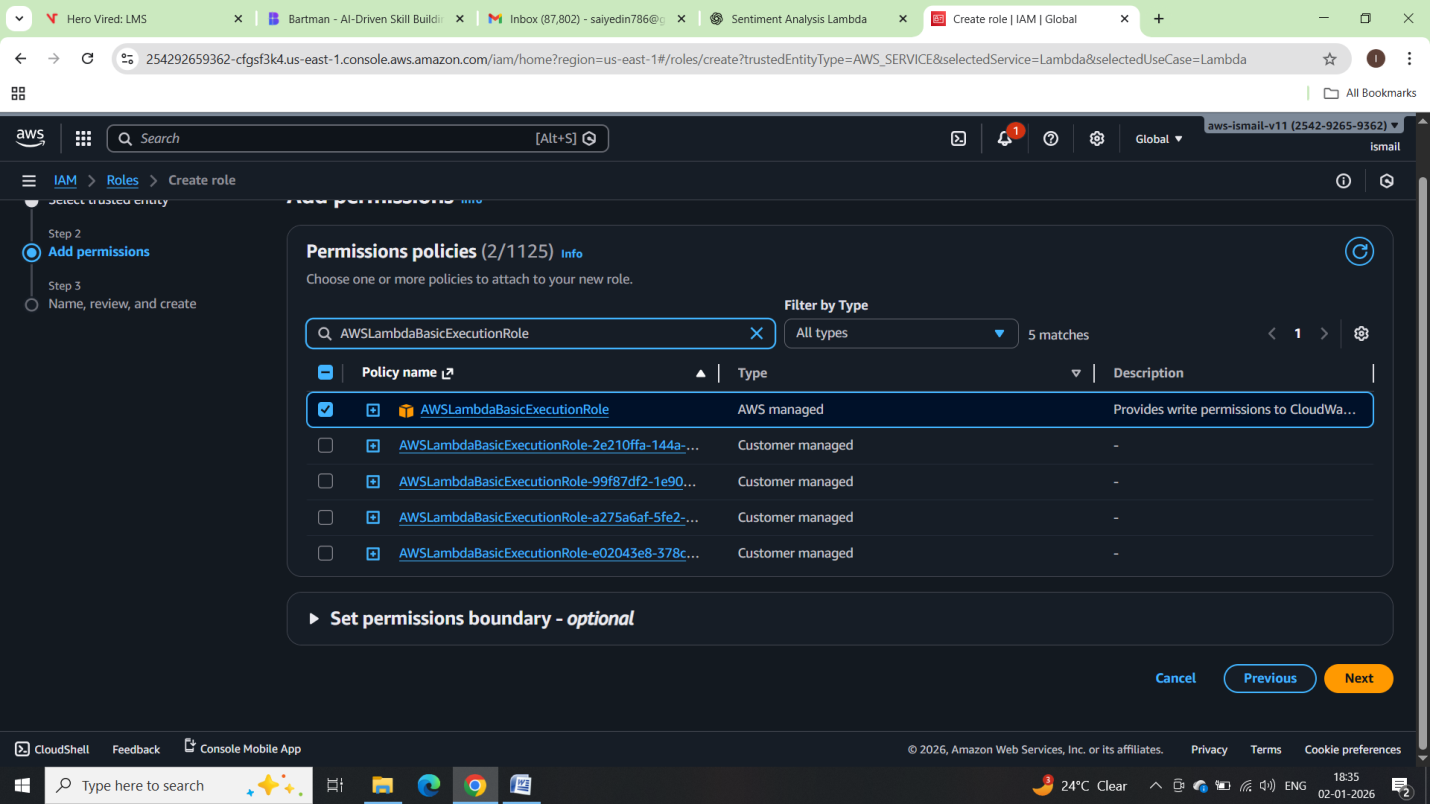
### Attach Policies

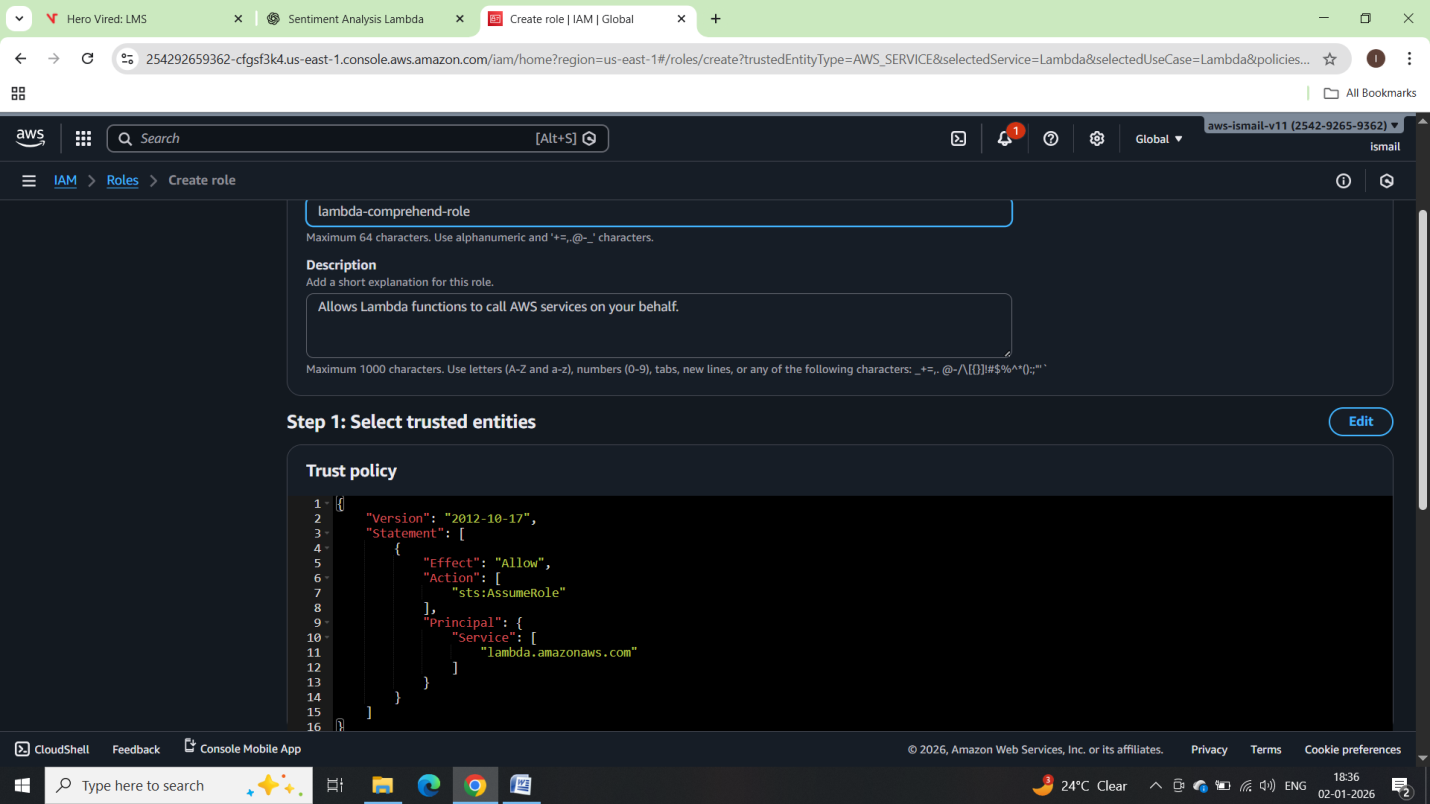
Attach the following policies:

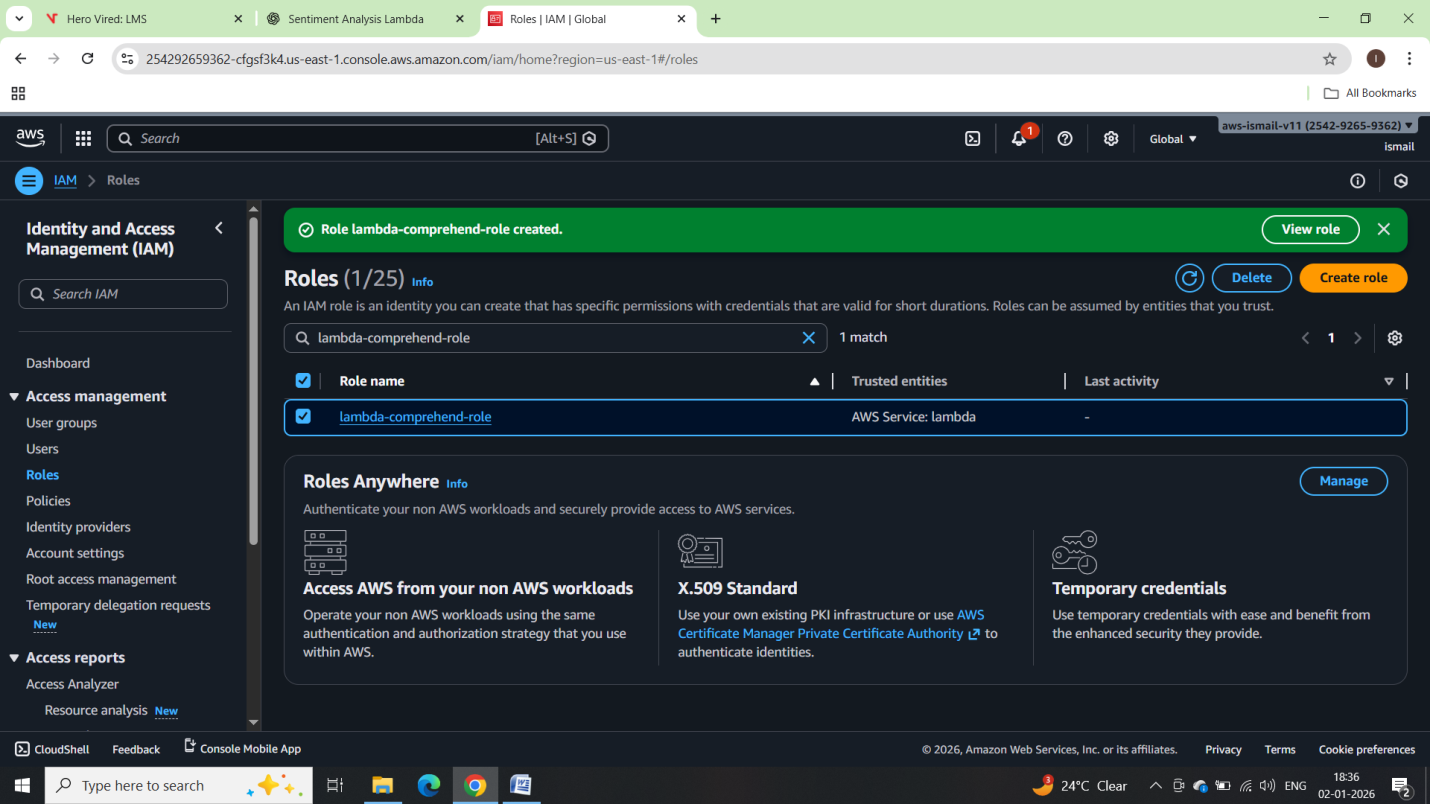
**ComprehendFullAccess**  
(or use custom policy below for best practice)

**AWSLambdaBasicExecutionRole**  
(required for CloudWatch logs)





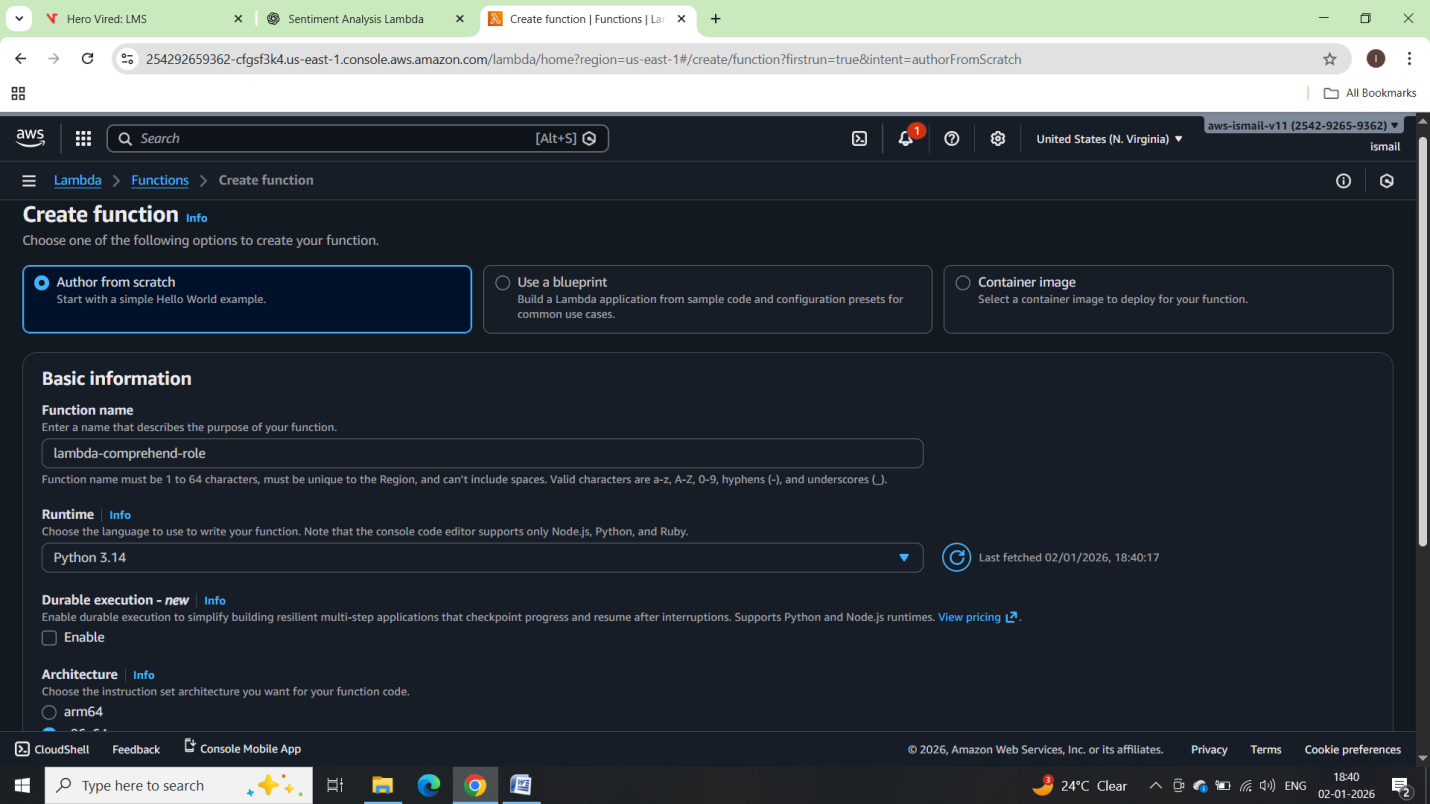


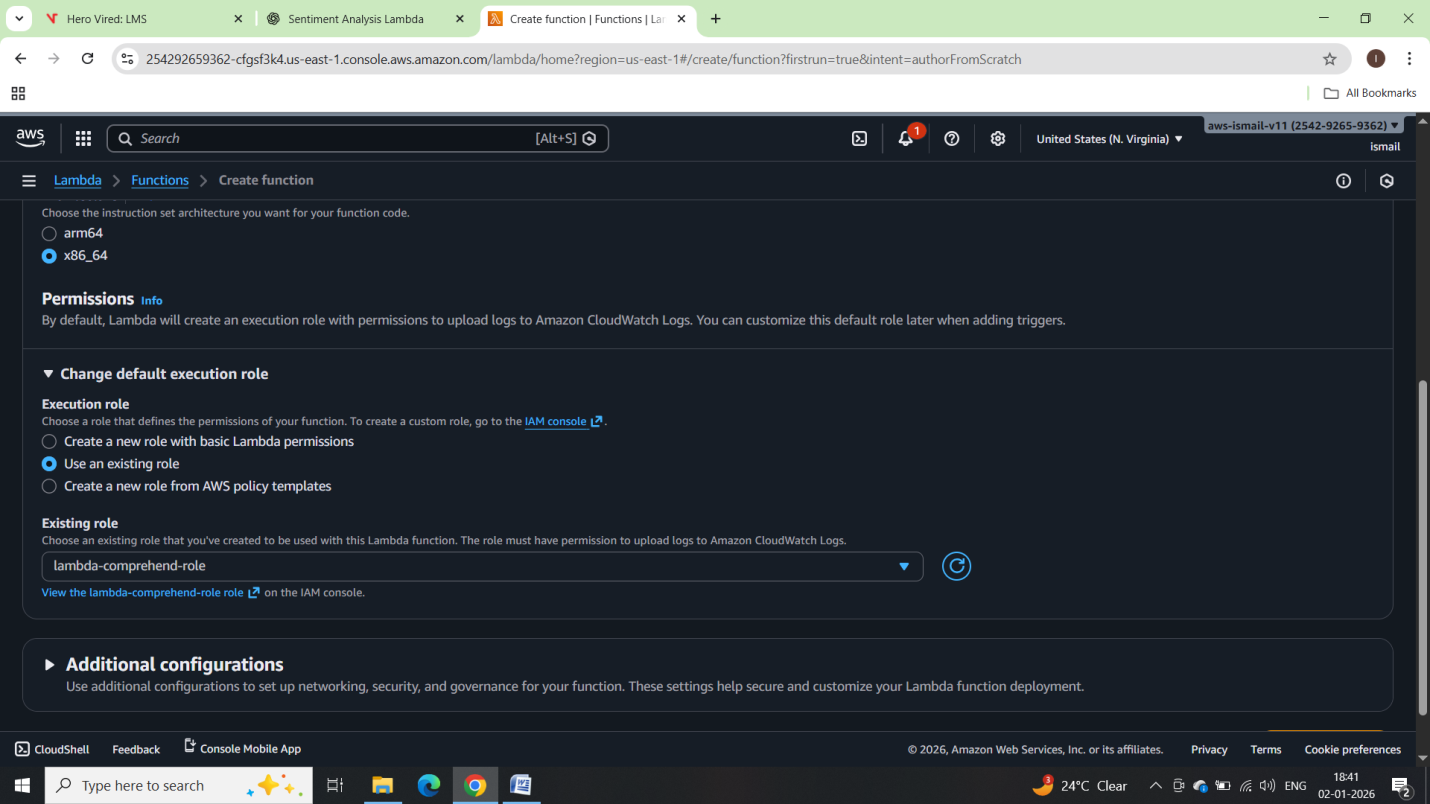


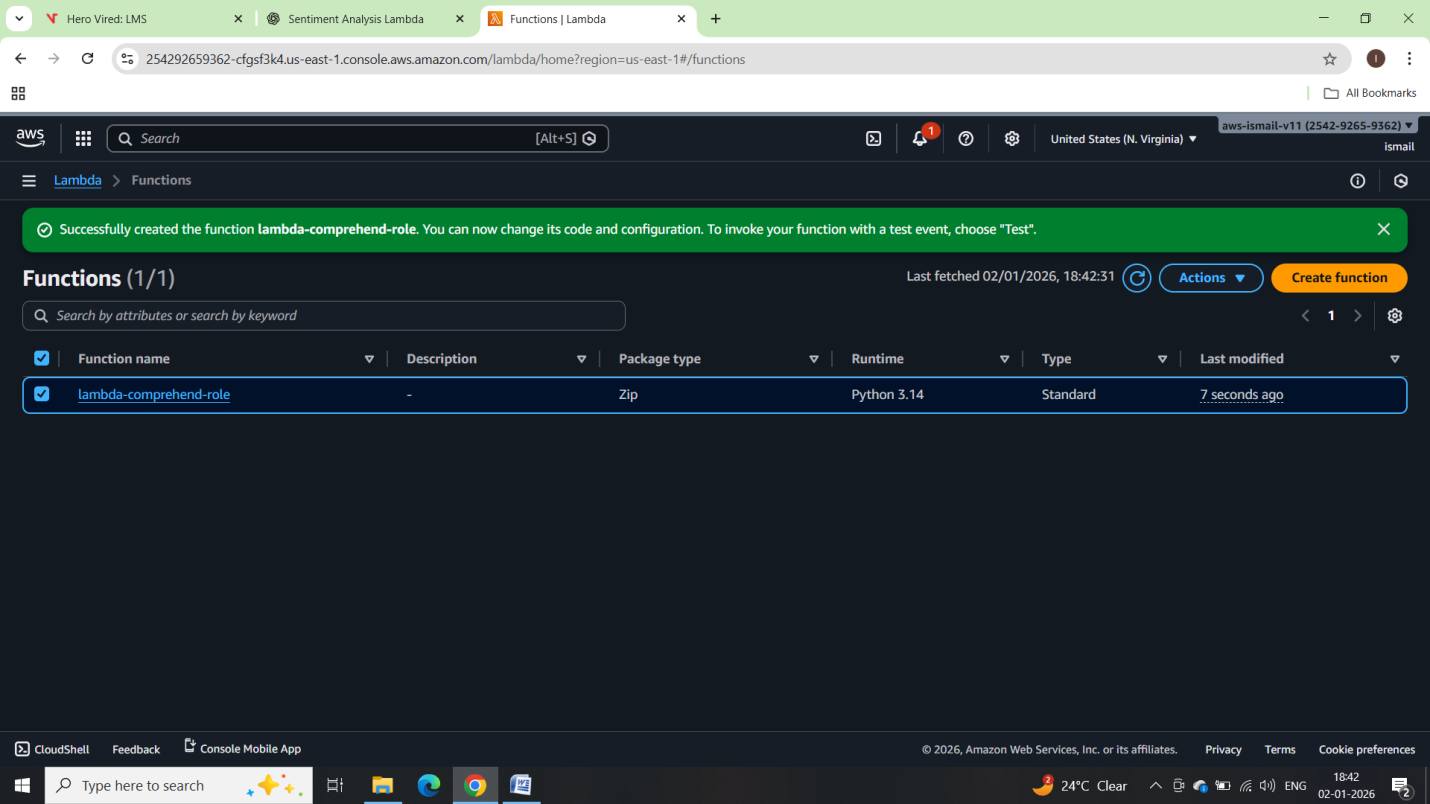
## Create Lambda Function

### Step 1: Open Lambda

* Go to **AWS Lambda → Create function**
* Choose **Author from scratch**
* Function name: sentiment-analysis-lambda
* Runtime: **Python 3.10**
* Execution role: **Use existing role**
* Select: lambda-comprehend-role
* Click **Create function**







Python boto3 code:

import json

import boto3

import logging

# Initialize logger

logger = logging.getLogger()

logger.setLevel(logging.INFO)

# Initialize Comprehend client

comprehend = boto3.client('comprehend')

def lambda\_handler(event, context):

try:

# 1. Extract review text from event

review\_text = event.get("review")

if not review\_text:

raise ValueError("No review text provided")

logger.info(f"User Review: {review\_text}")

# 2. Analyze sentiment

response = comprehend.detect\_sentiment(

Text=review\_text,

LanguageCode='en'

)

sentiment = response['Sentiment']

sentiment\_scores = response['SentimentScore']

# 3. Log results

logger.info(f"Detected Sentiment: {sentiment}")

logger.info(f"Sentiment Scores: {sentiment\_scores}")

return {

"statusCode": 200,

"review": review\_text,

"sentiment": sentiment,

"scores": sentiment\_scores

}

except Exception as e:

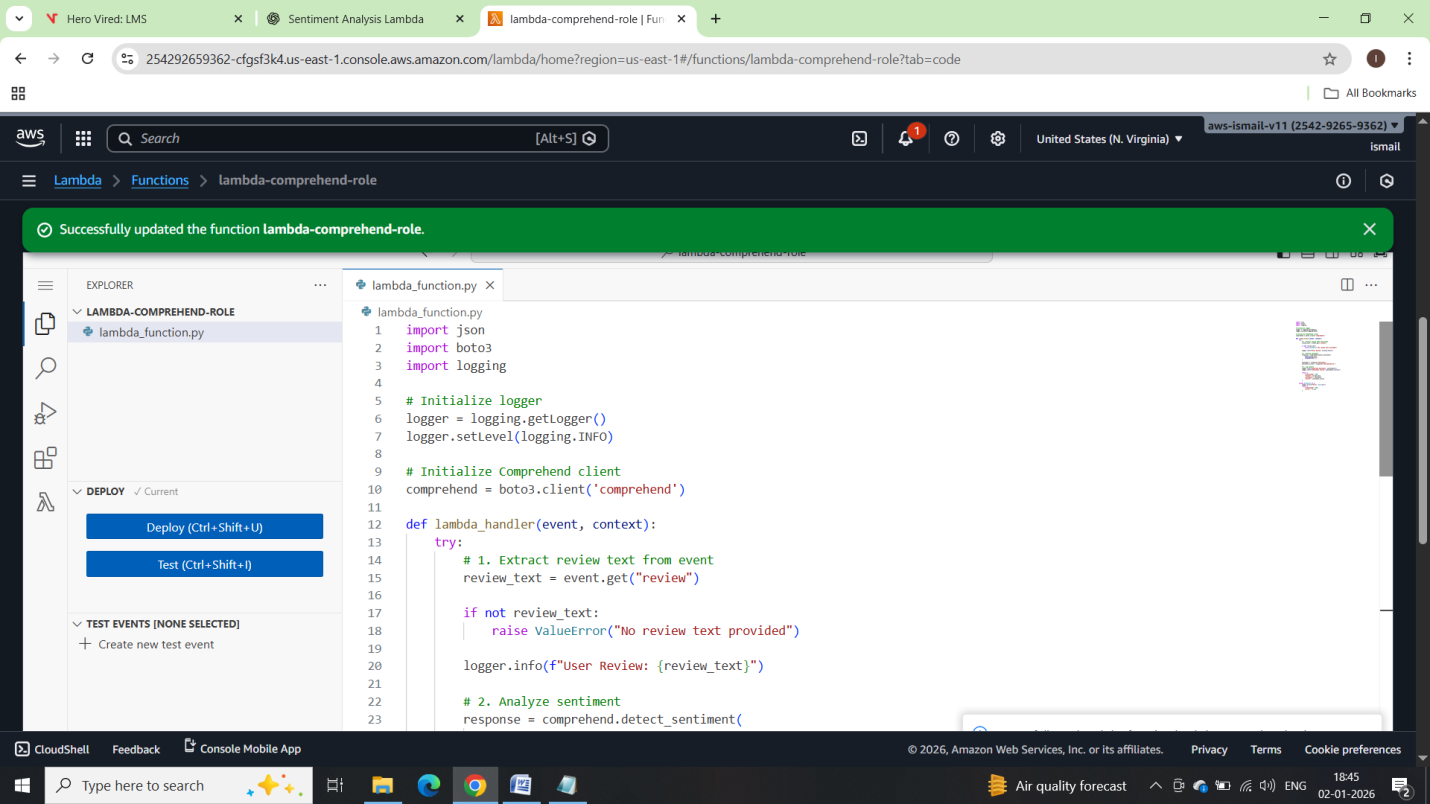
logger.error(f"Error: {str(e)}")

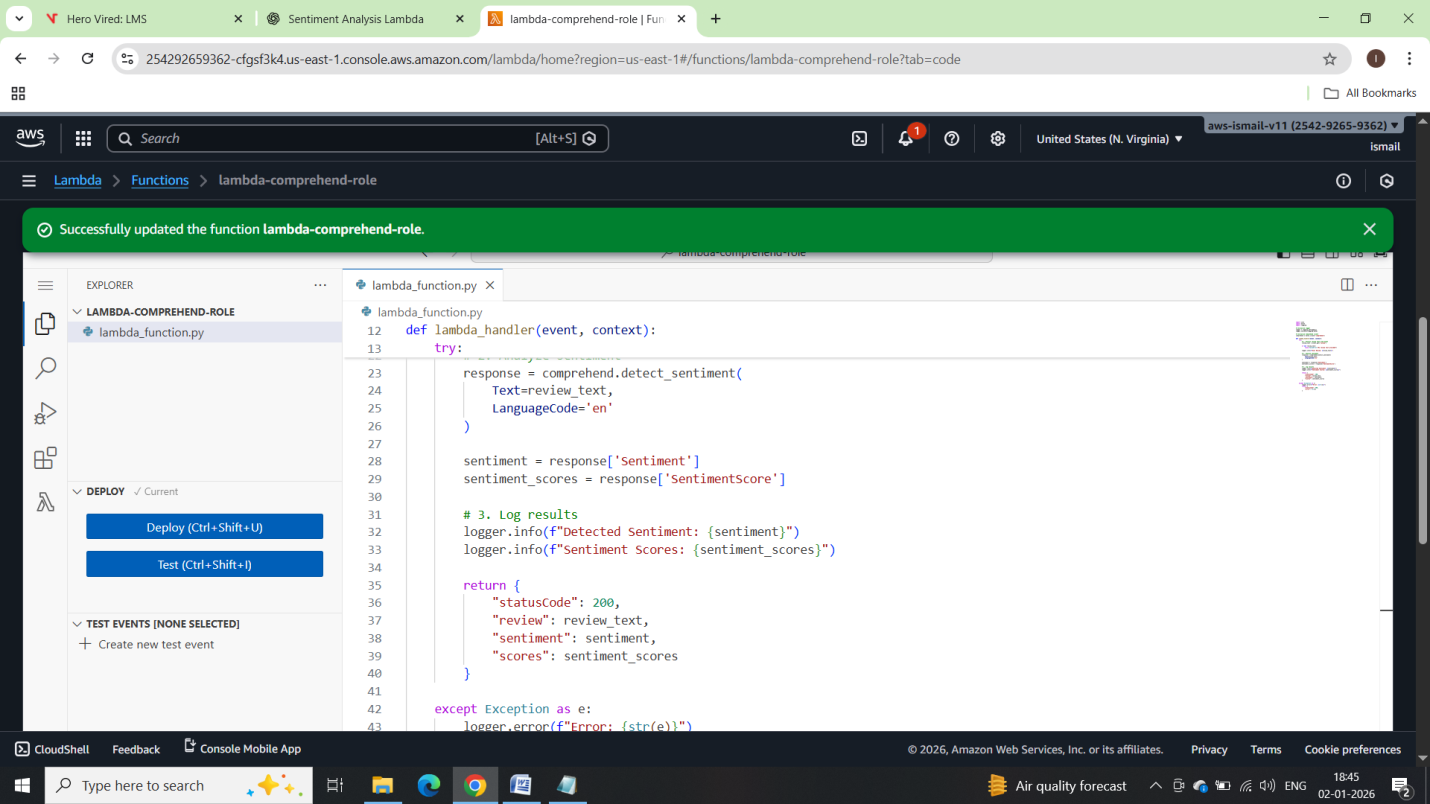
return {

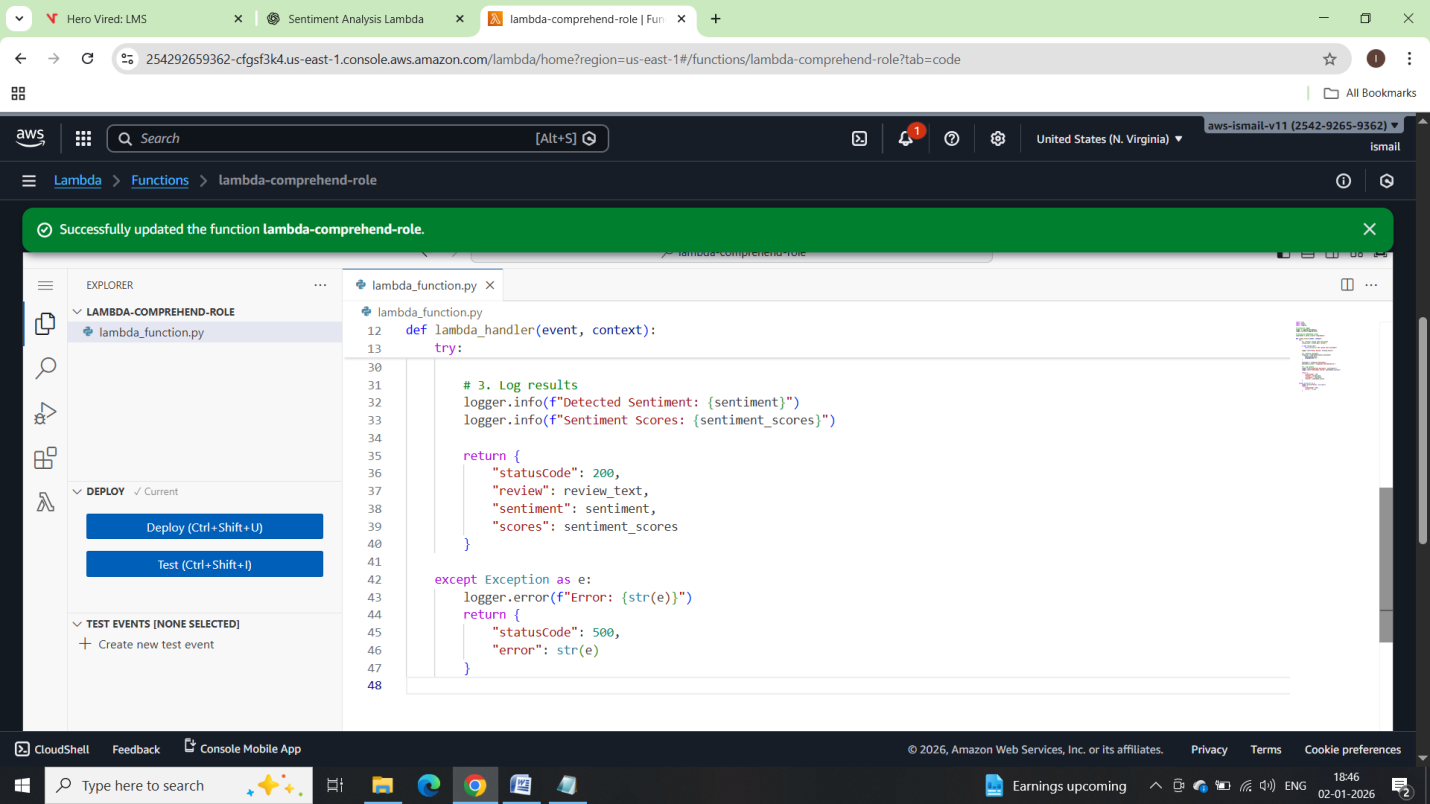
"statusCode": 500,

"error": str(e)

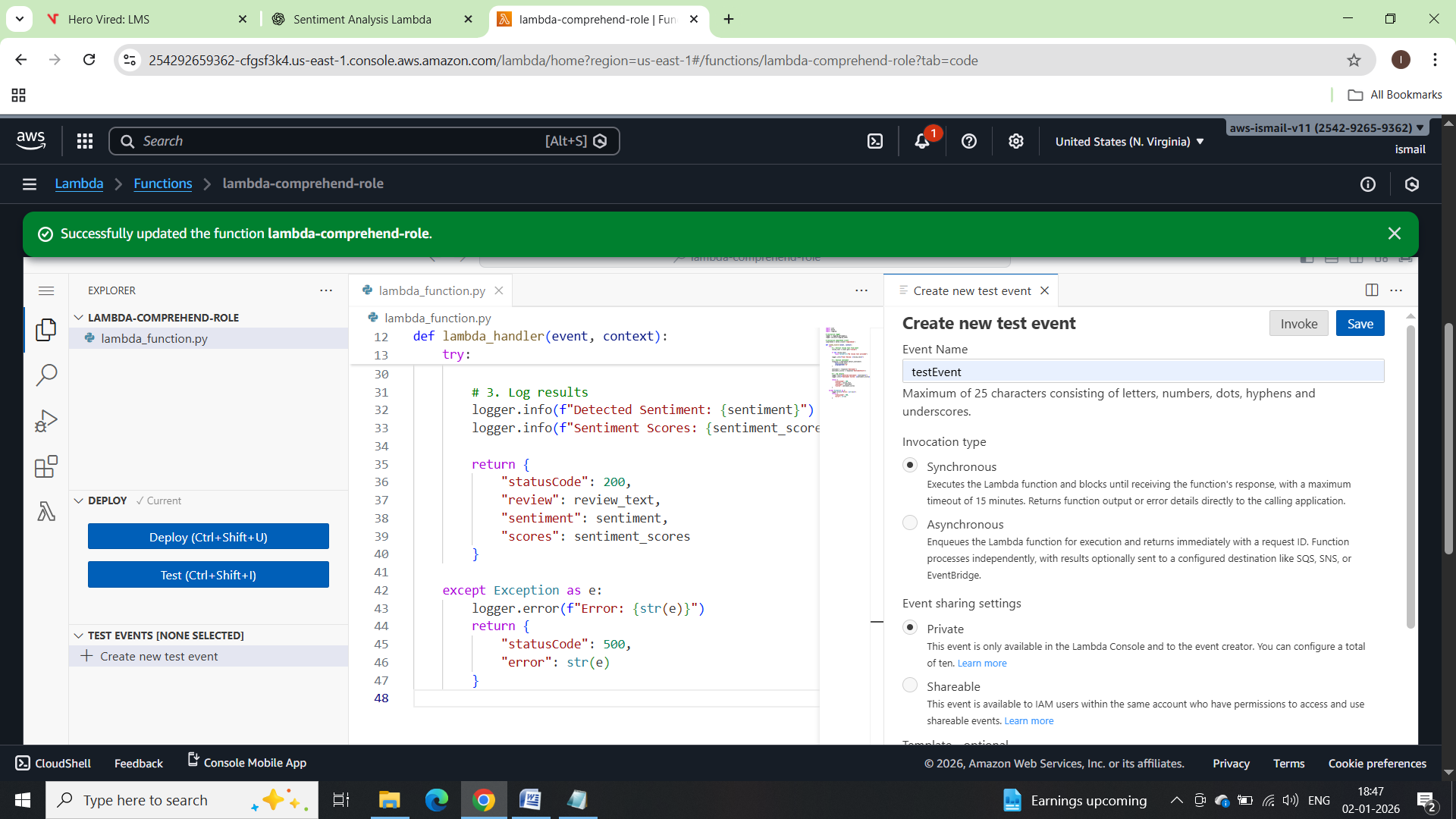
}

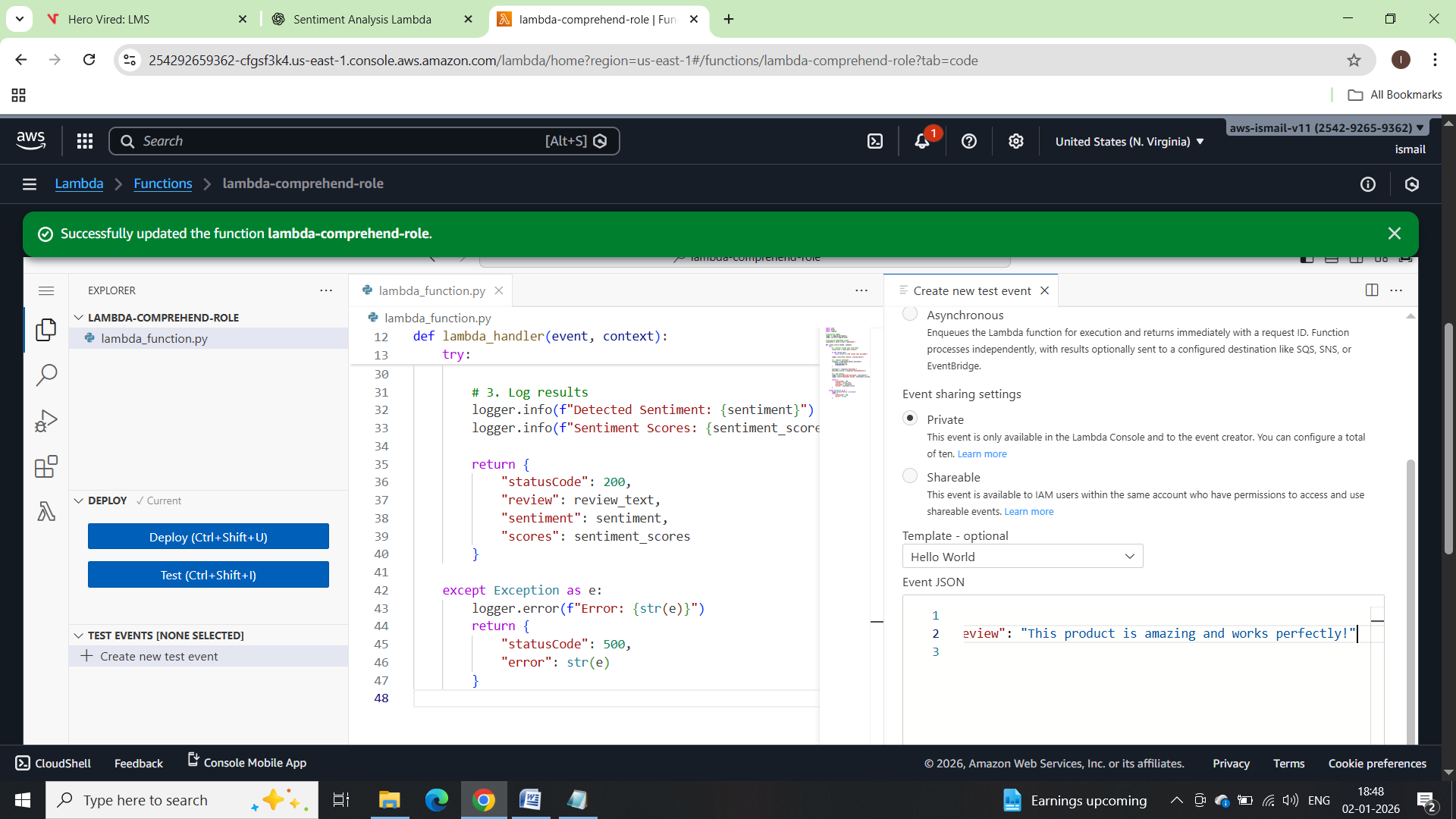




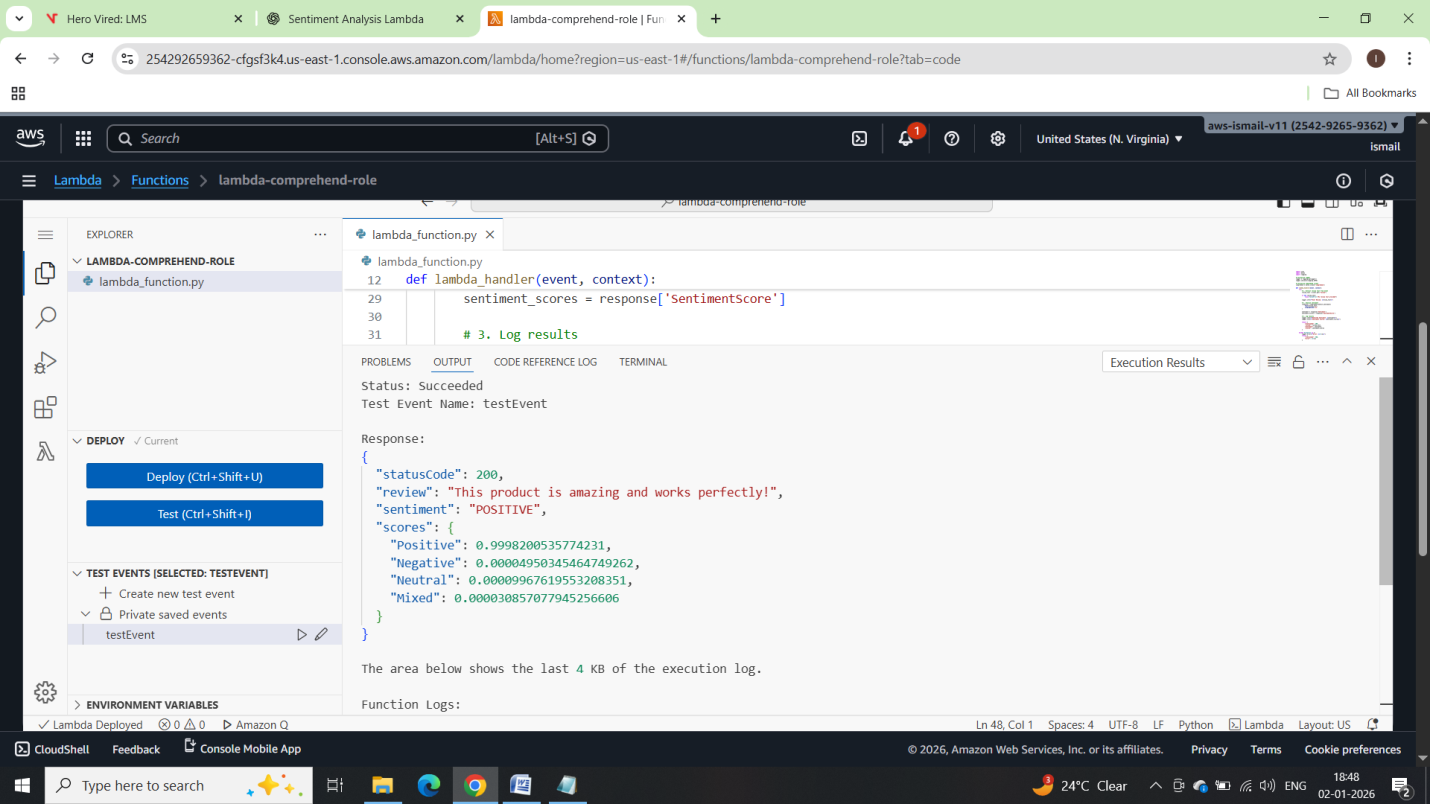


Create Test Event:





Now testing with test event:



Verifying cloudwatch Logs:

