**Amazon Simple Queue Service (SQS)**

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. A mini project using SQS can be quite illustrative. Here's a guide to creating a basic project that demonstrates sending and receiving messages using SQS.

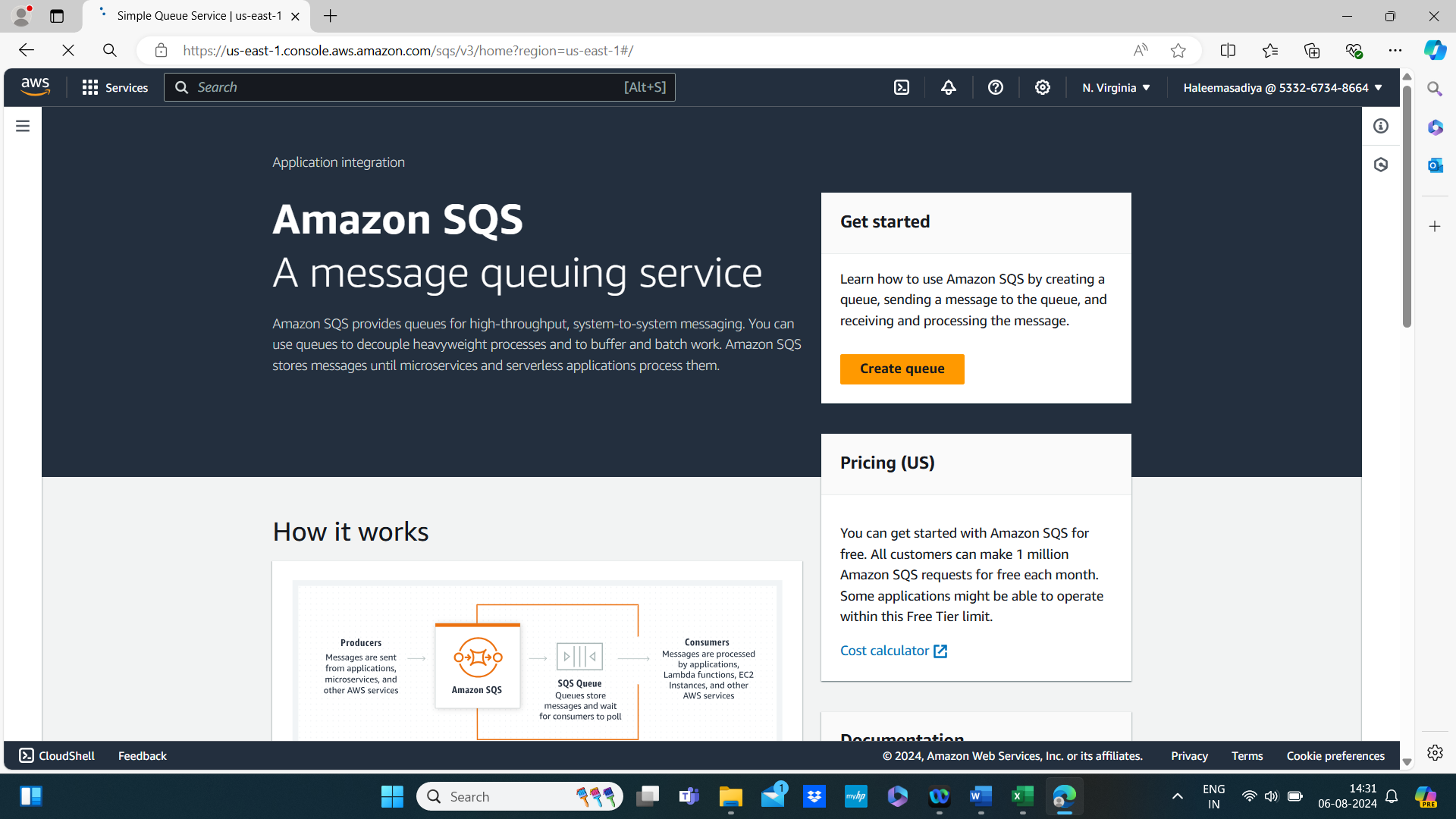
**Project Overview**

In this project, we will:

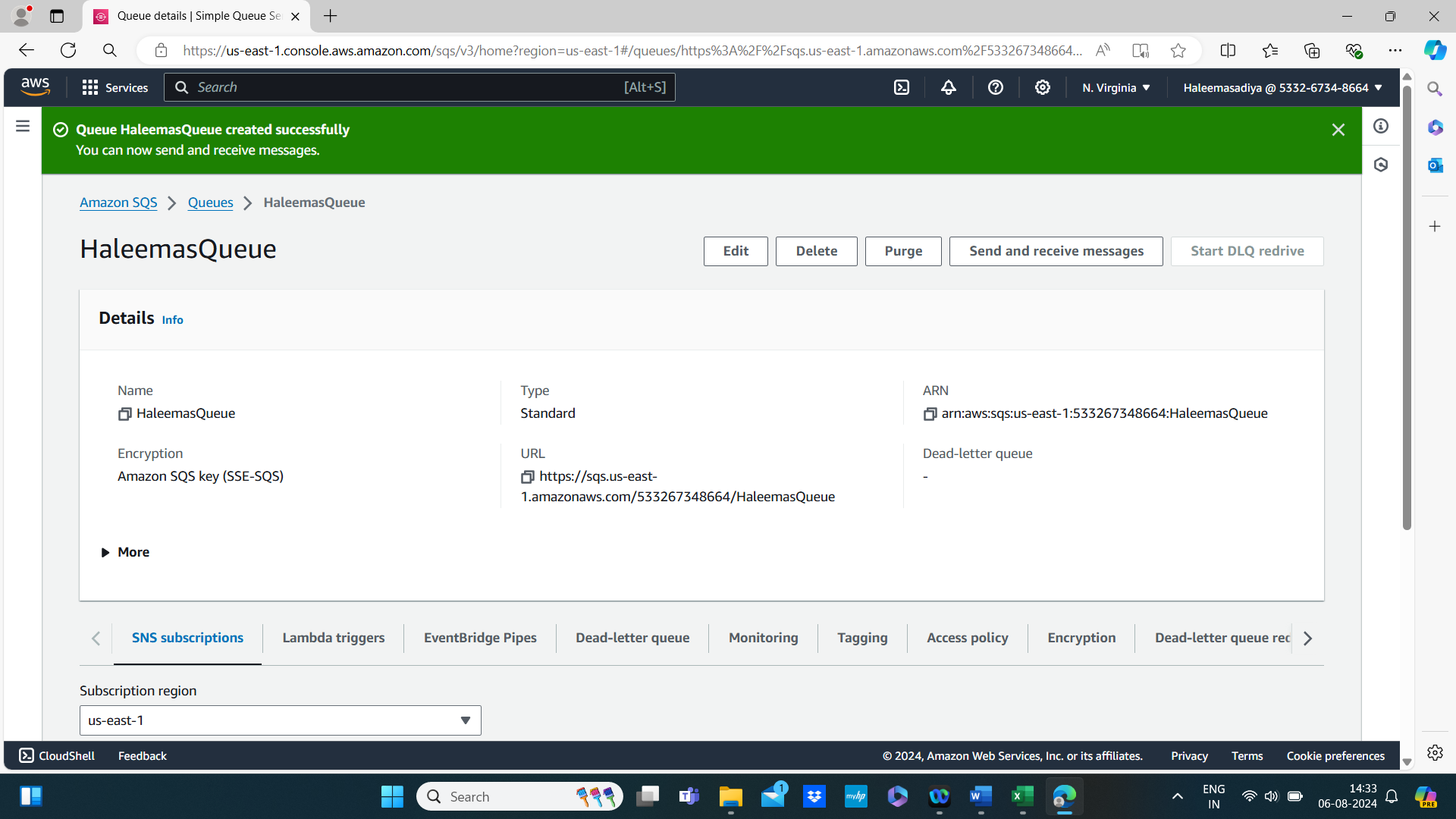
1. Create an SQS queue.
2. Set up an AWS Lambda function to send messages to the queue.
3. Create another Lambda function to receive messages from the queue.
4. Test the integration by sending and receiving messages.

**Step 1: Create an SQS Queue**

1. **Navigate to SQS:**
   * Go to the AWS Management Console.
   * Search for "SQS" in the services search bar and select it.

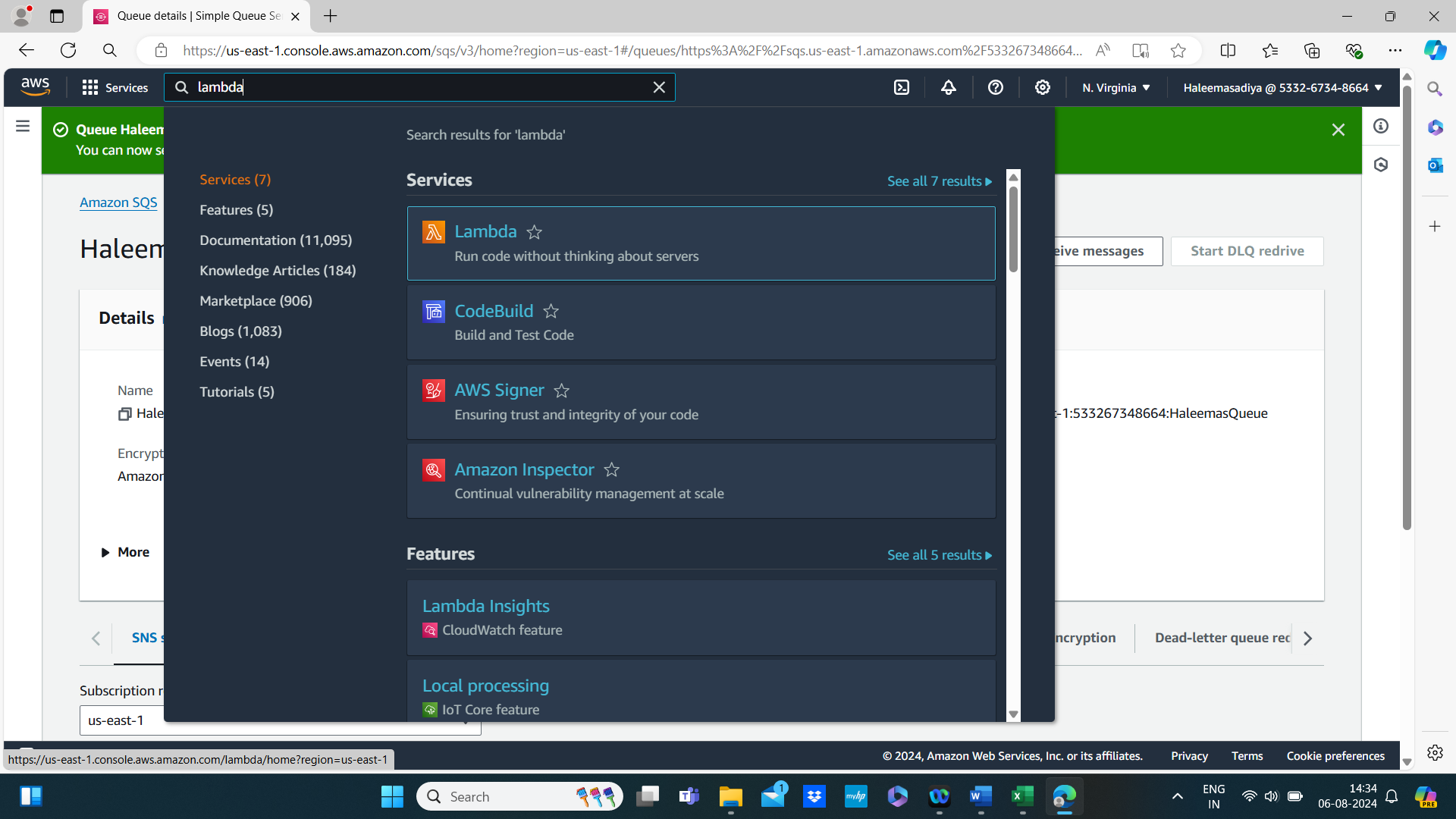


1. **Create a New Queue:**
   * Click on “Create queue.”
   * Choose the type of queue: Standard or FIFO. For simplicity, we’ll use Standard.
   * Enter a name for your queue (e.g., HaleemasQueue).
   * Configure the settings as needed, but the default settings should be fine for this example.
   * Click “Create queue.”

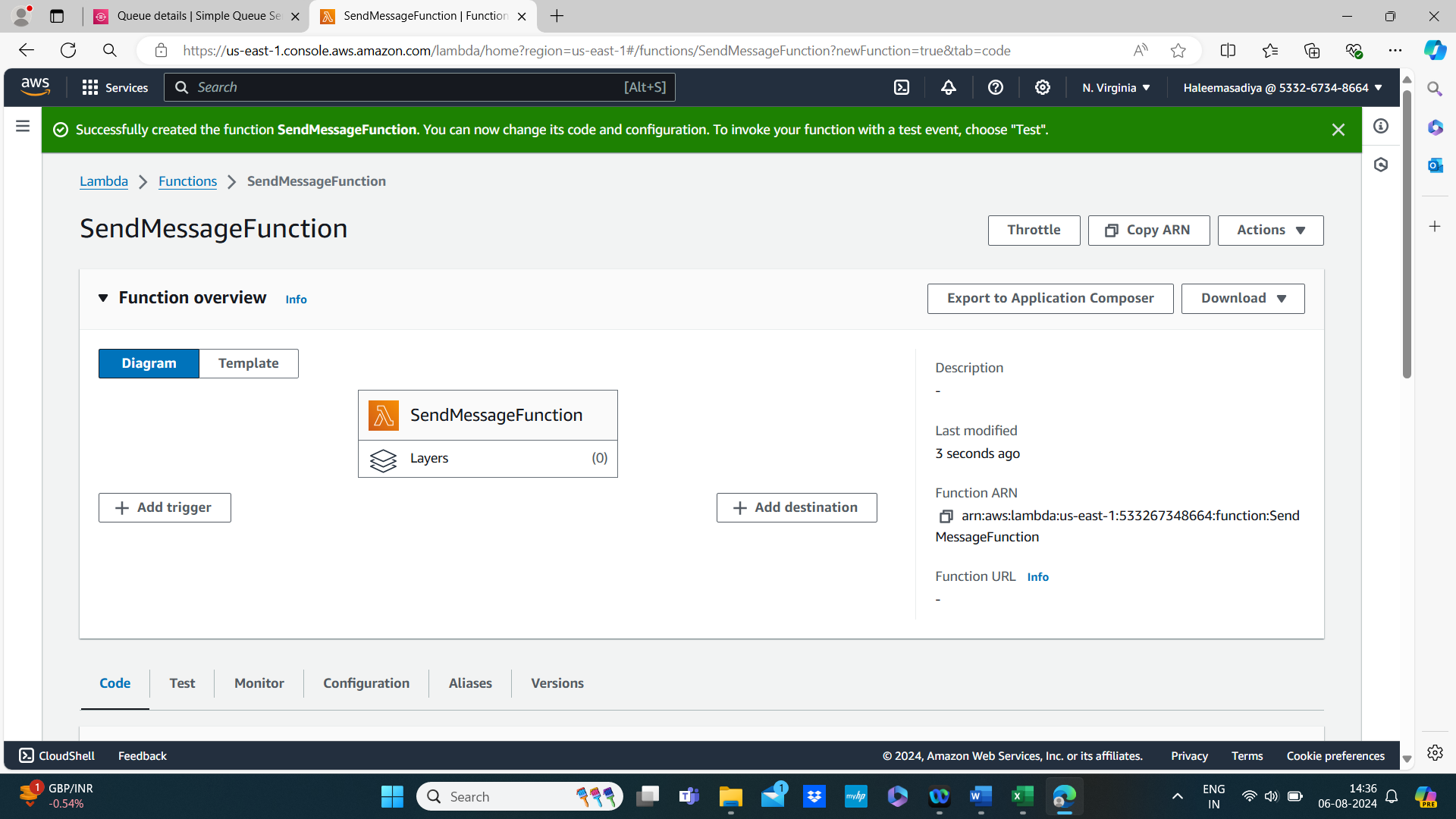


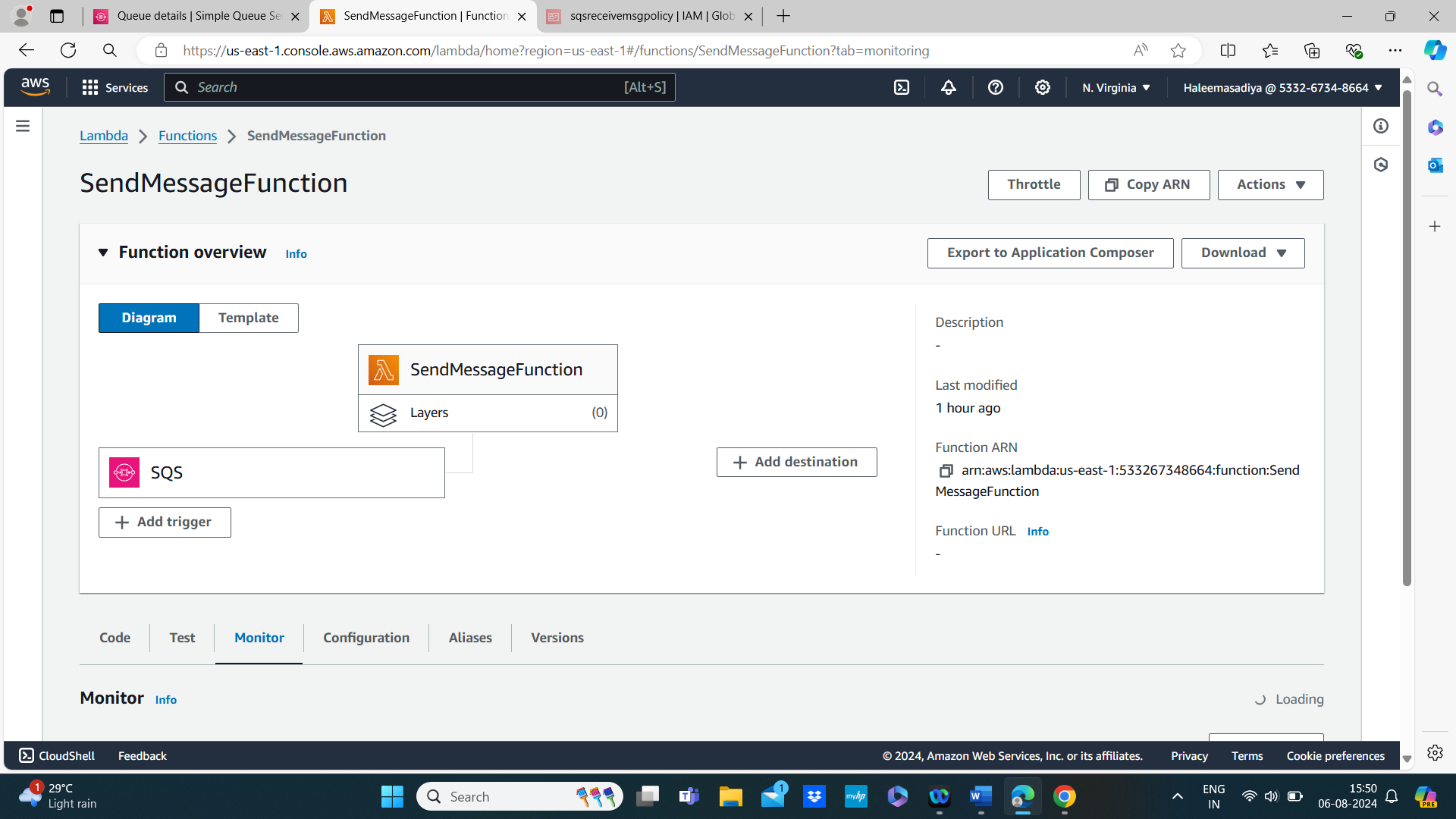
**Step 2: Create Lambda Function to Send Messages**

1. **Navigate to Lambda:**
   * Go to the AWS Management Console.
   * Search for "Lambda" in the services search bar and select it.

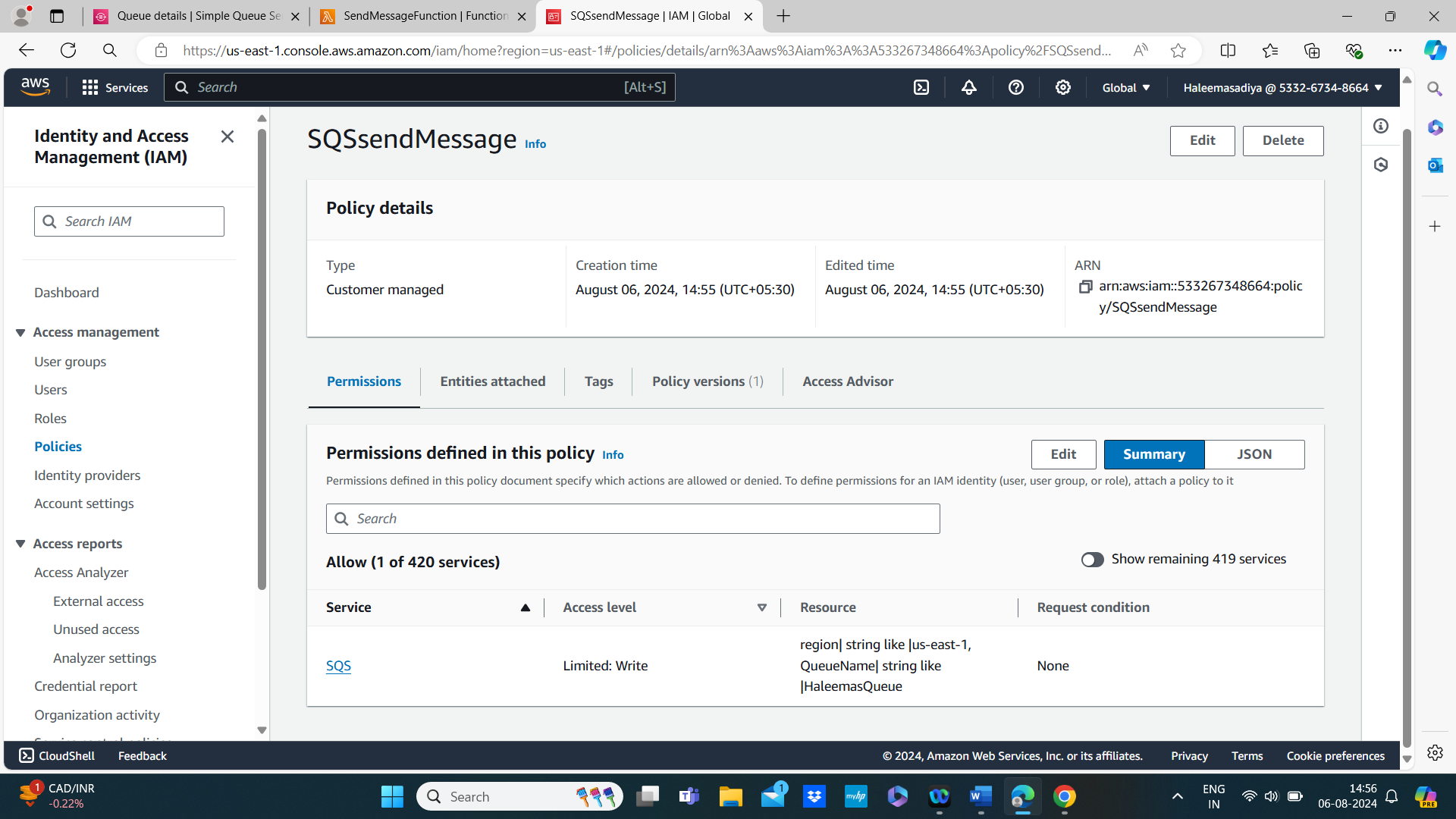


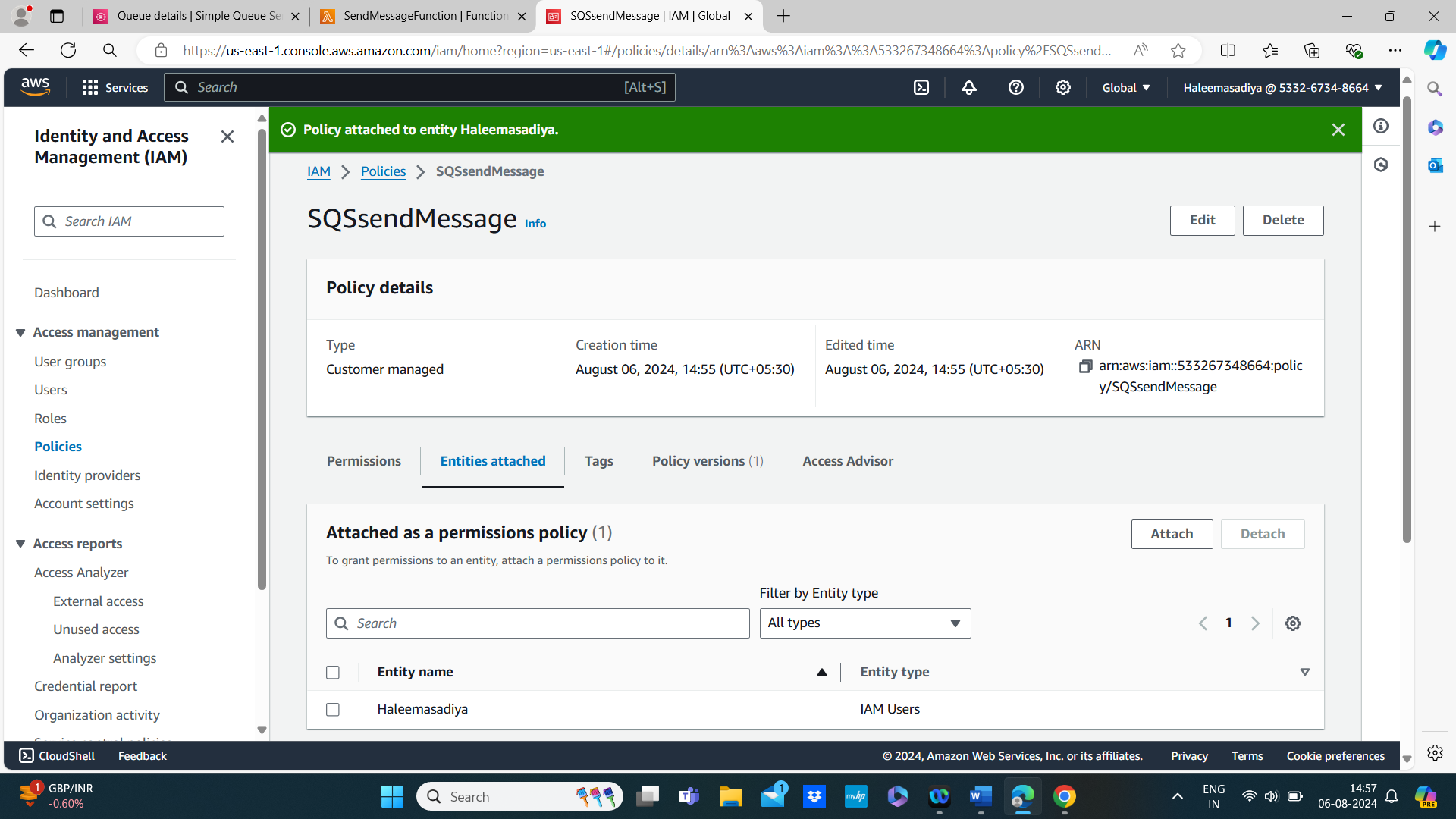
1. **Create a New Lambda Function:**
   * Click “Create function.”
   * Choose “Author from scratch.”
   * Name your function (e.g., SendMessageFunction).
   * Choose the runtime (e.g., Node.js, Python).
   * Click “Create function.”





1. **Add IAM Role Permissions:**
   * Ensure the Lambda function's execution role has permissions to send messages to SQS.
   * Attach the AmazonSQSFullAccess policy or a custom policy with sqs:SendMessage permission.





1. **Write the Code:**
   * Use the inline editor to add the following code (example in Python):

import json

import boto3

# Initialize the SQS client

sqs = boto3.client('sqs')

def lambda\_handler(event, context):

# Queue URL

queue\_url = 'https://sqs.YOUR\_REGION.amazonaws.com/YOUR\_ACCOUNT\_ID/MyQueue'

# Send a message to the SQS queue

response = sqs.send\_message(

QueueUrl=queue\_url,

MessageBody='Hello from Lambda!'

)

return {

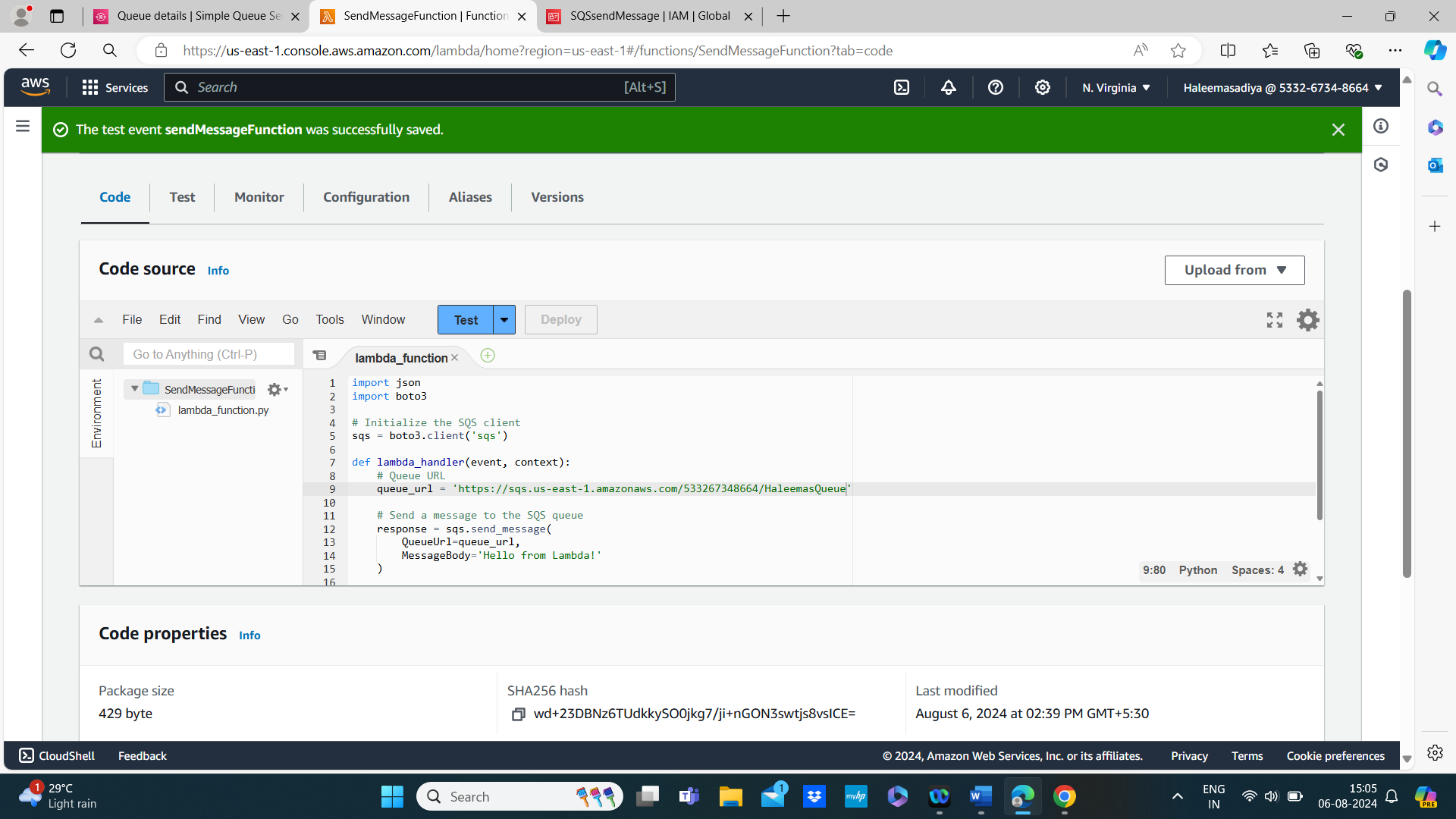
'statusCode': 200,

'body': json.dumps('Message sent!')

}

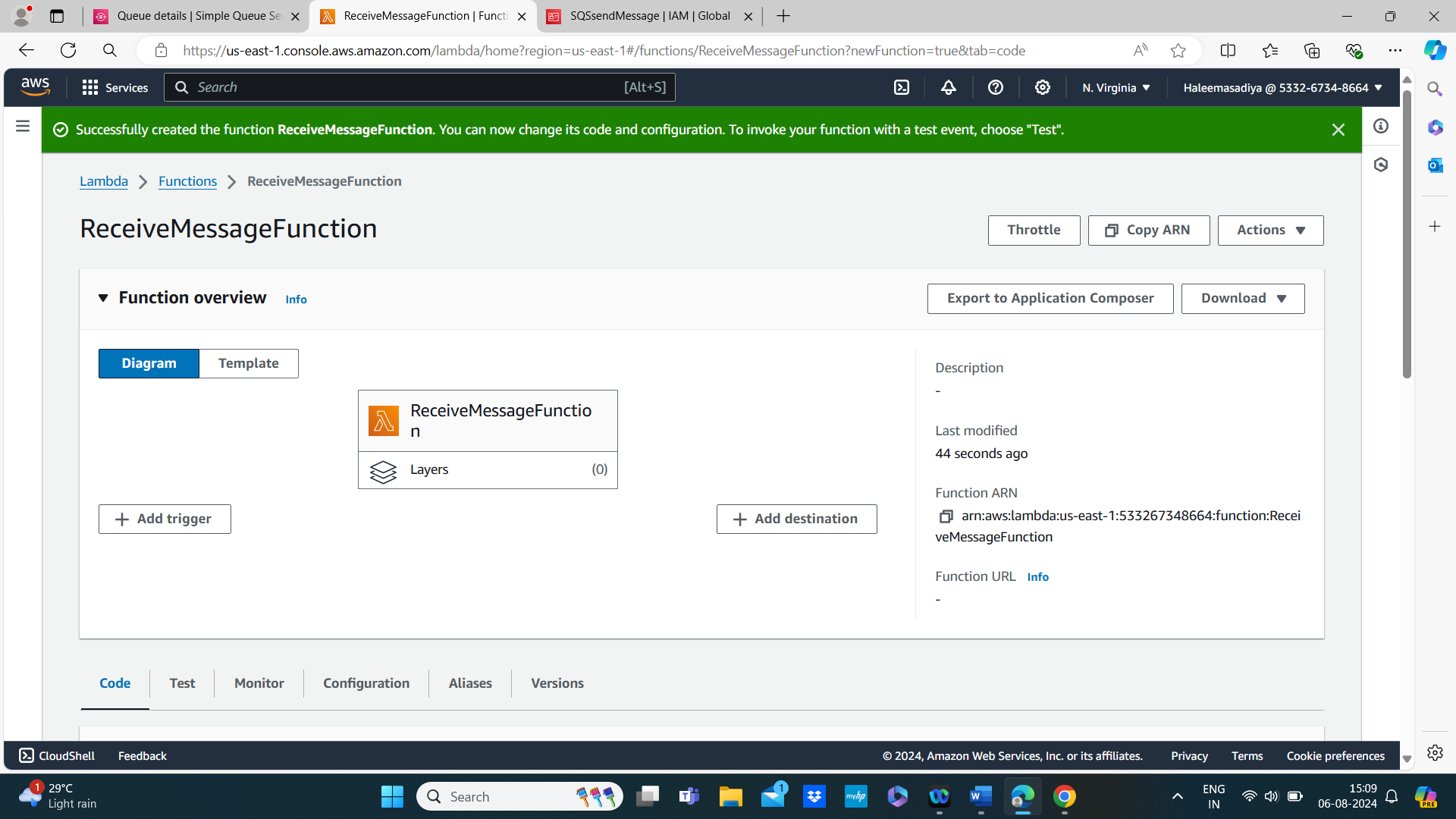
* + Replace YOUR\_REGION and YOUR\_ACCOUNT\_ID with your AWS region and account ID.

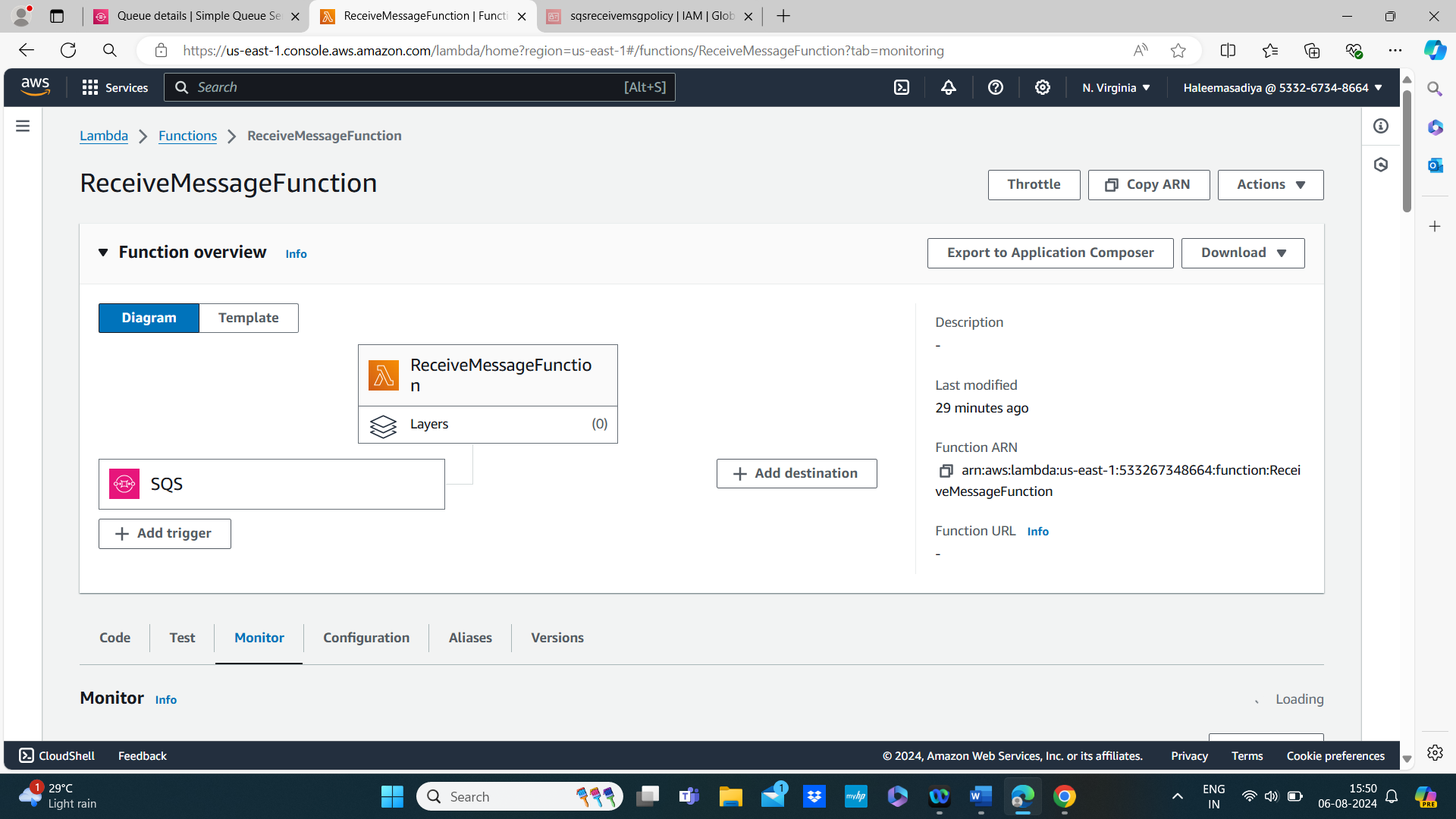
1. **Deploy and Test:**
   * Deploy the function and use the test feature to invoke it.



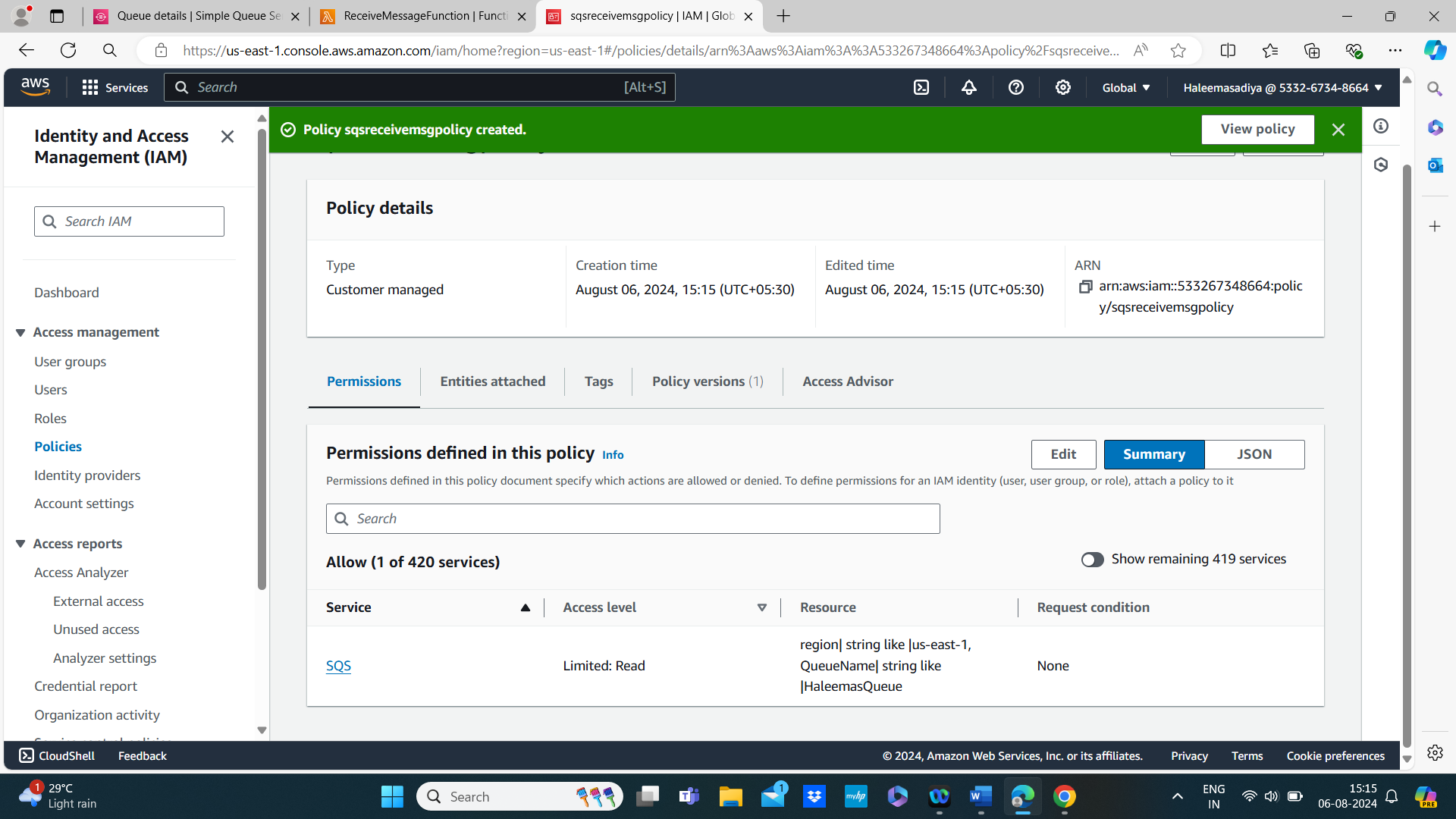
**Step 3: Create Lambda Function to Receive Messages**

1. **Create Another Lambda Function:**
   * Follow the same steps as before to create a new Lambda function (e.g., ReceiveMessageFunction).





1. **Add IAM Role Permissions:**
   * Ensure this function’s role has permissions to receive messages from SQS.
   * Attach the AmazonSQSFullAccess policy or a custom policy with sqs:ReceiveMessage permission.





1. **Write the Code:**
   * Use the inline editor to add the following code (example in Python):

import json

import boto3

# Initialize the SQS client

sqs = boto3.client('sqs')

def lambda\_handler(event, context):

# Queue URL

queue\_url = 'https://sqs.YOUR\_REGION.amazonaws.com/YOUR\_ACCOUNT\_ID/MyQueue'

# Receive a message from the SQS queue

response = sqs.receive\_message(

QueueUrl=queue\_url,

MaxNumberOfMessages=1,

WaitTimeSeconds=10

)

messages = response.get('Messages', [])

if messages:

for message in messages:

receipt\_handle = message['ReceiptHandle']

# Print the message body

print('Message:', message['Body'])

# Delete received message from queue

sqs.delete\_message(

QueueUrl=queue\_url,

ReceiptHandle=receipt\_handle

)

else:

print('No messages received.')

return {

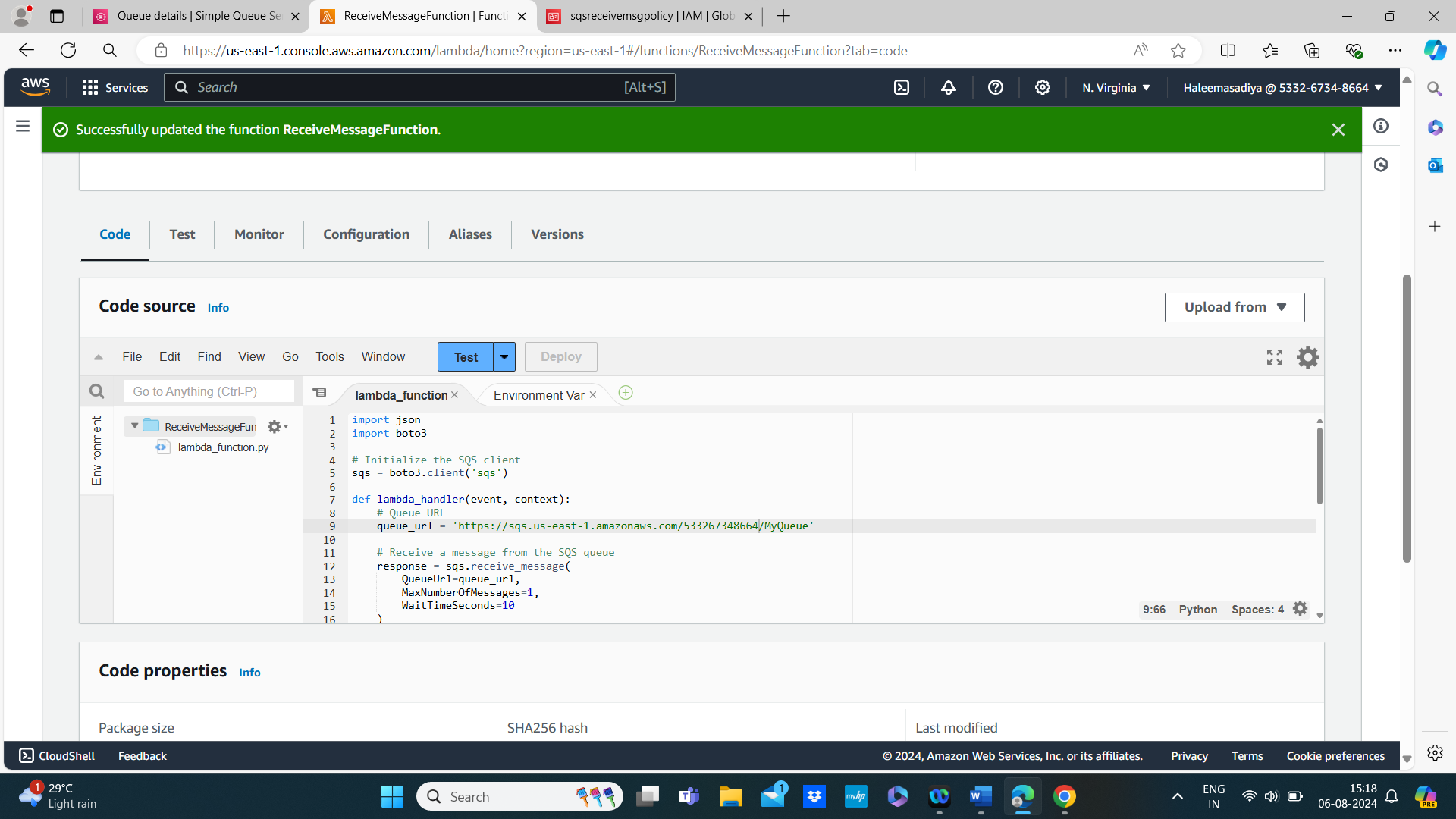
'statusCode': 200,

'body': json.dumps('Message received!')

}

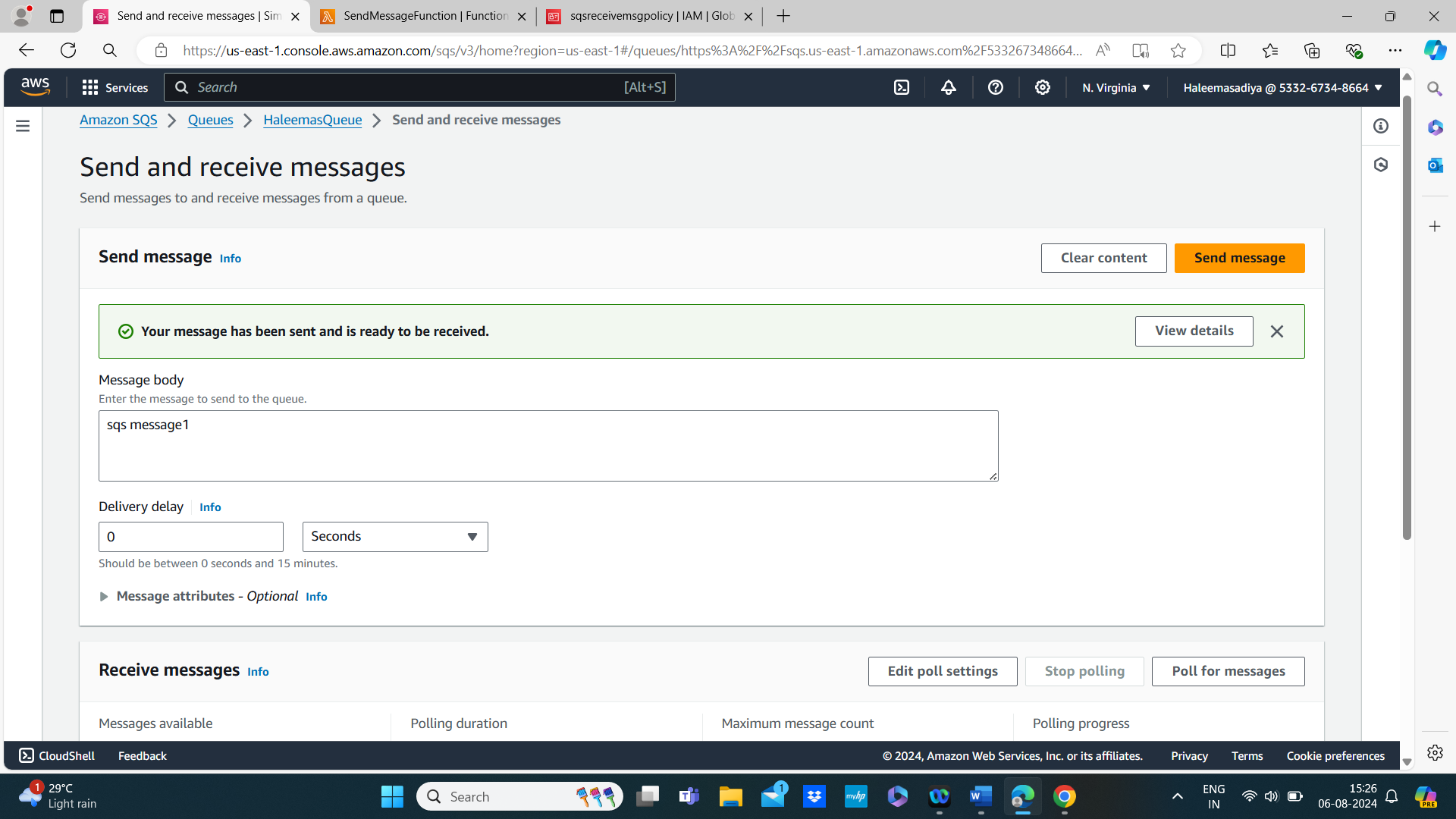
* + Replace YOUR\_REGION and YOUR\_ACCOUNT\_ID with your AWS region and account ID.

1. **Deploy and Test:**
   * Deploy the function and test it. Ensure that it is able to read from the queue and process messages.

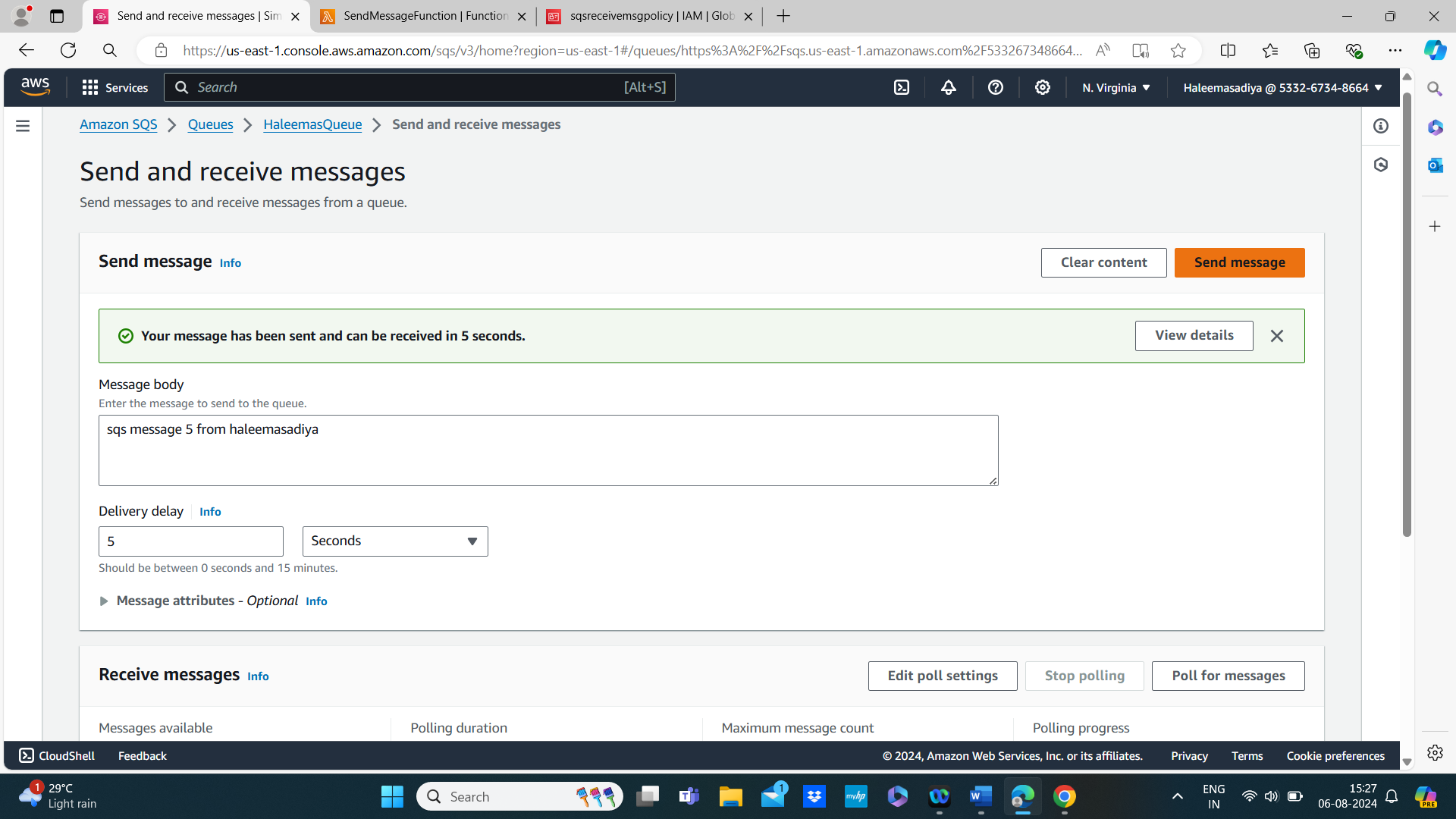


**Step 4: Test the Integration**

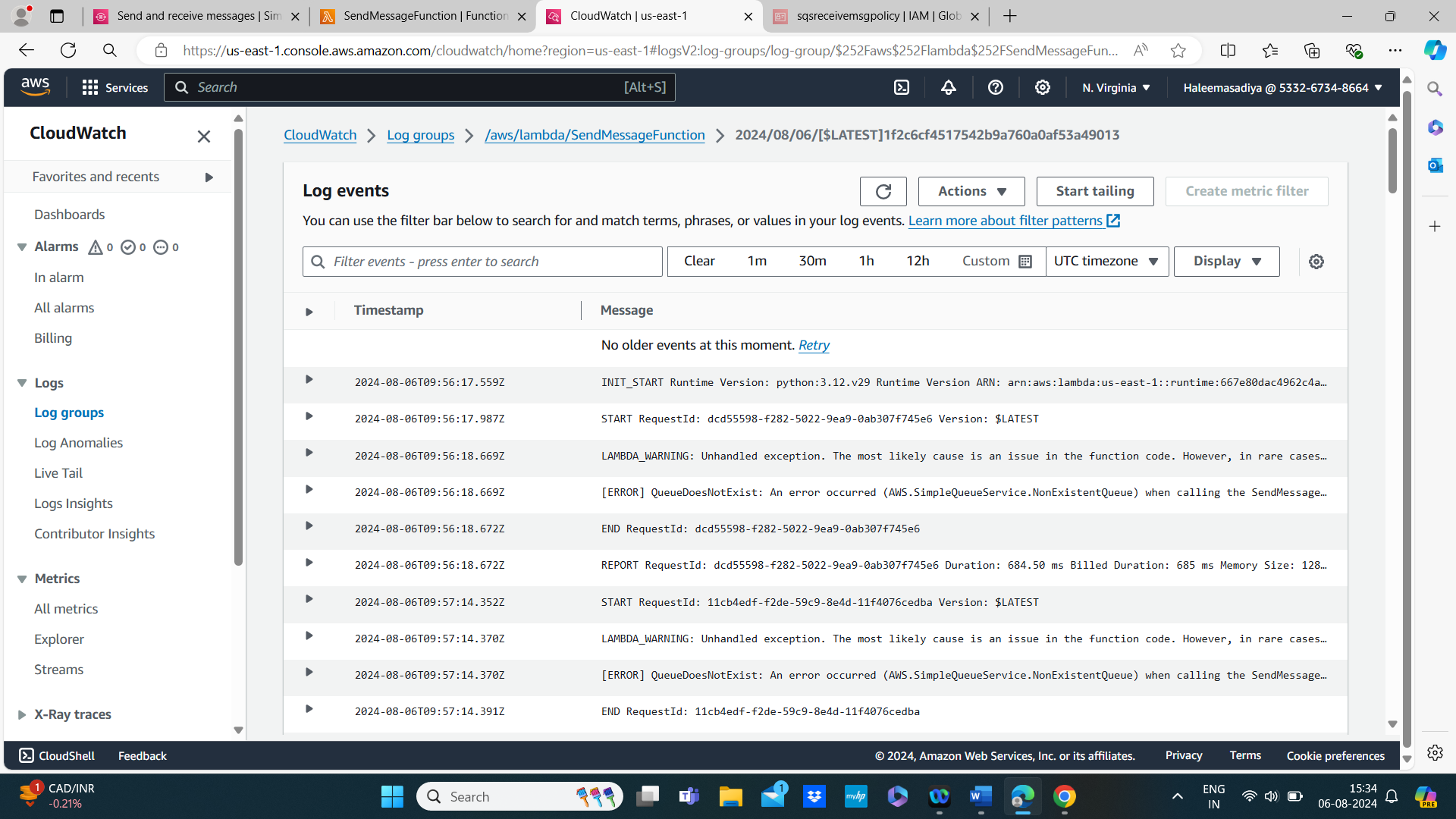
1. **Send Messages:**
   * Invoke the SendMessageFunction Lambda function to send a message to the SQS queue.

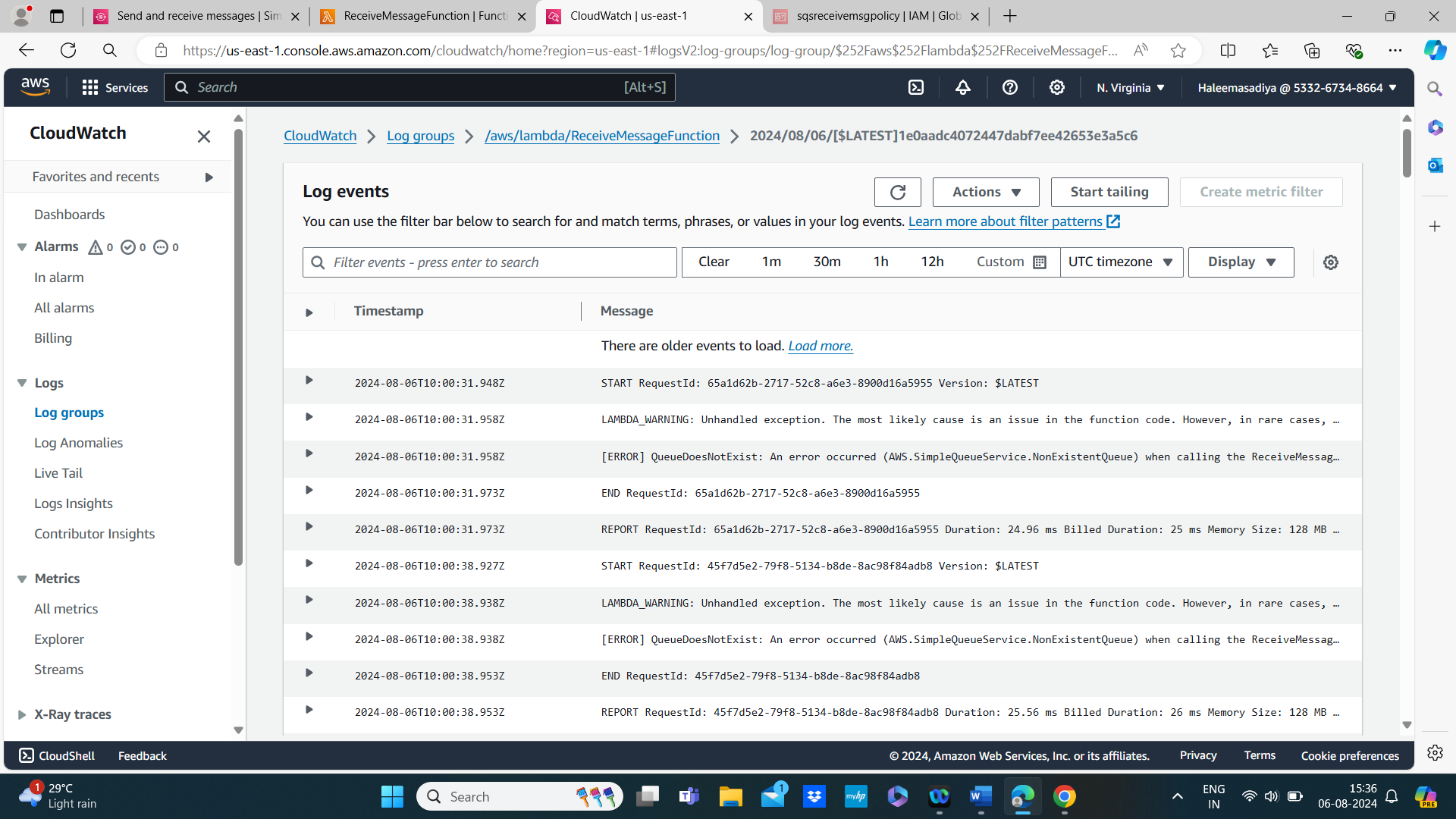


1. **Receive Messages:**
   * Invoke the ReceiveMessageFunction Lambda function to receive and process messages from the SQS queue.

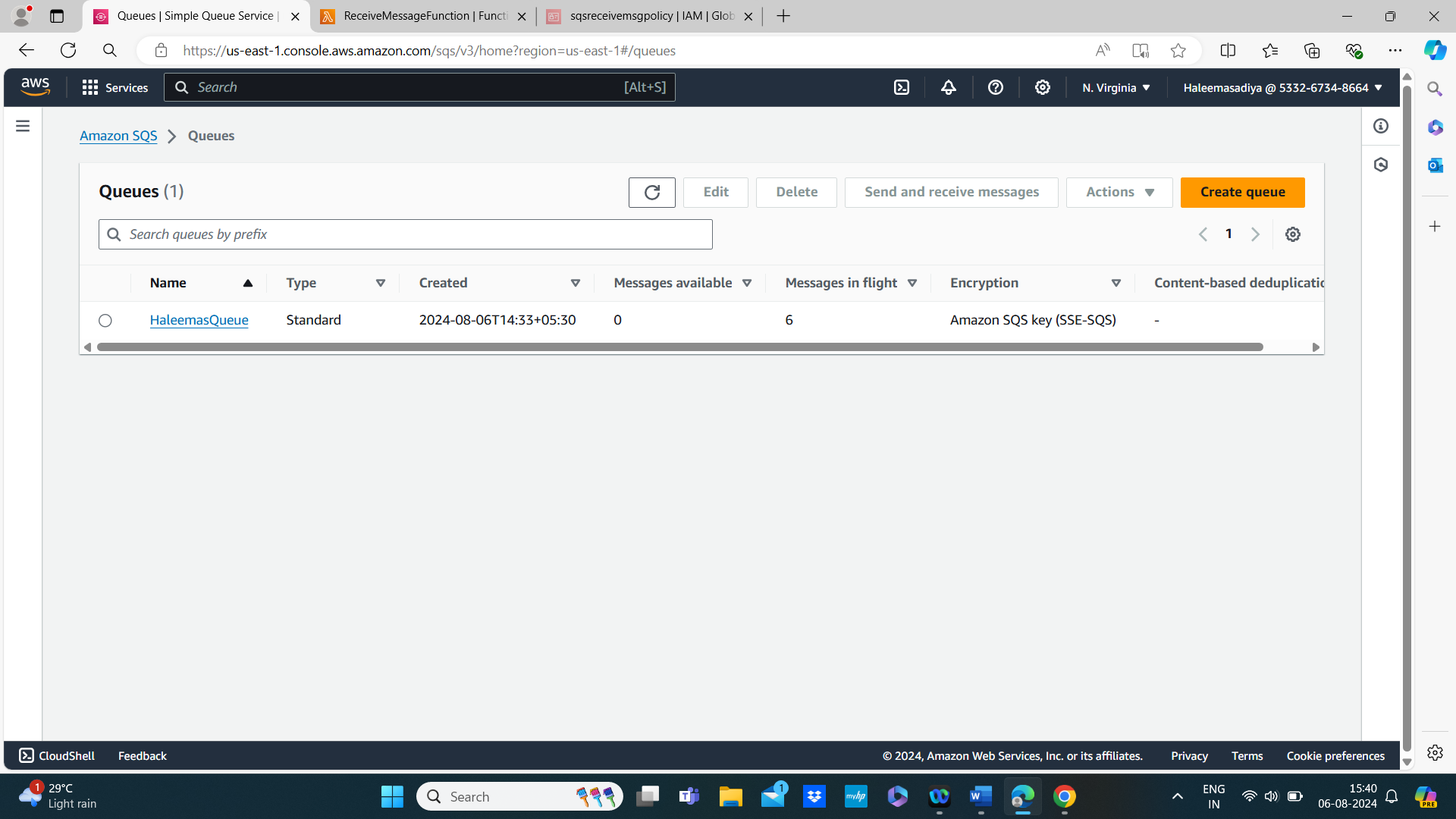


1. **Verify Results:**
   * Check the CloudWatch Logs for both Lambda functions to ensure that messages are being sent and received properly.





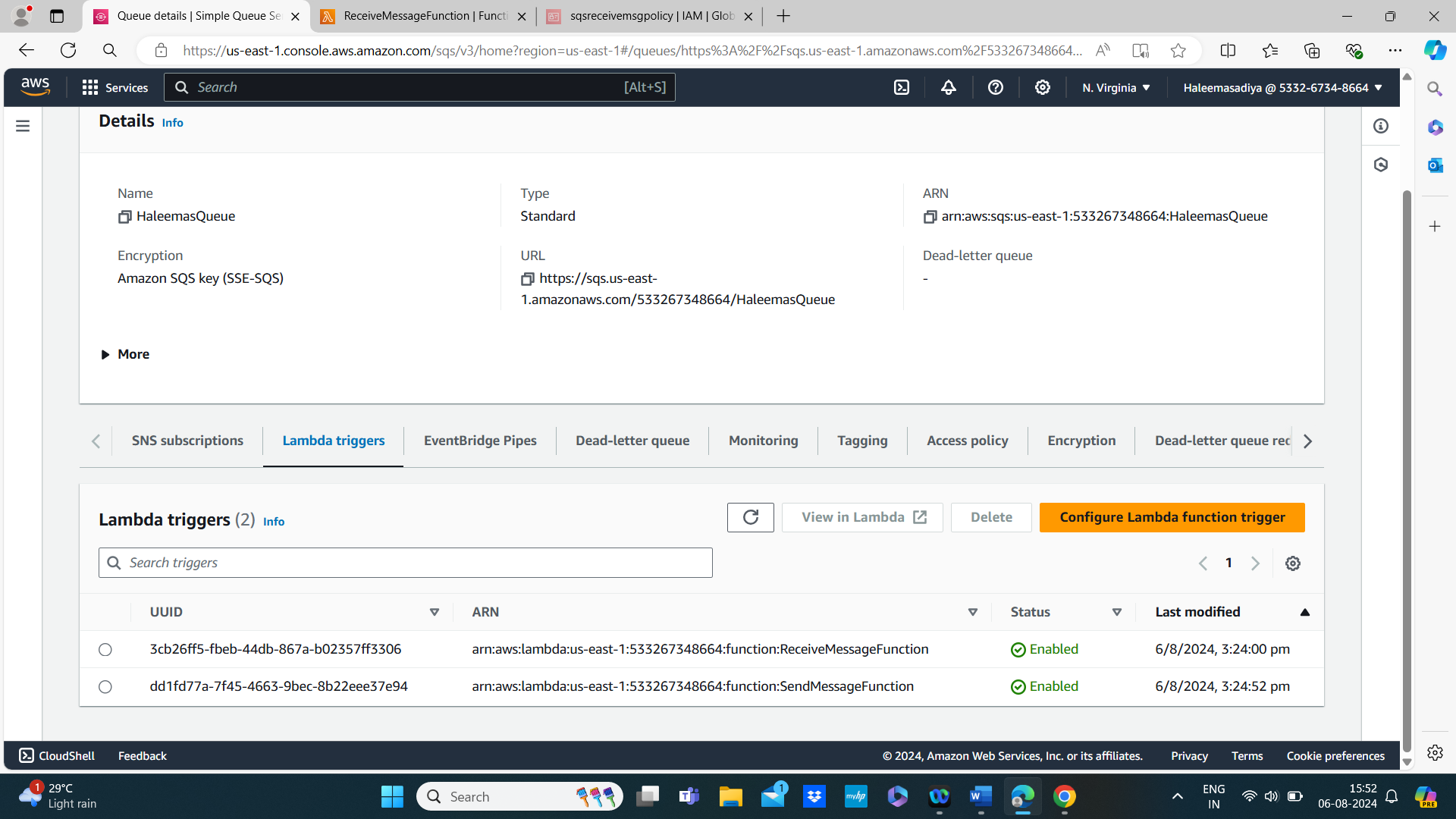
* + Verify that messages are correctly processed and deleted from the queue.



In Amazon SQS (Simple Queue Service), "messages in flight" refers to messages that have been received by a consumer but not yet deleted from the queue.

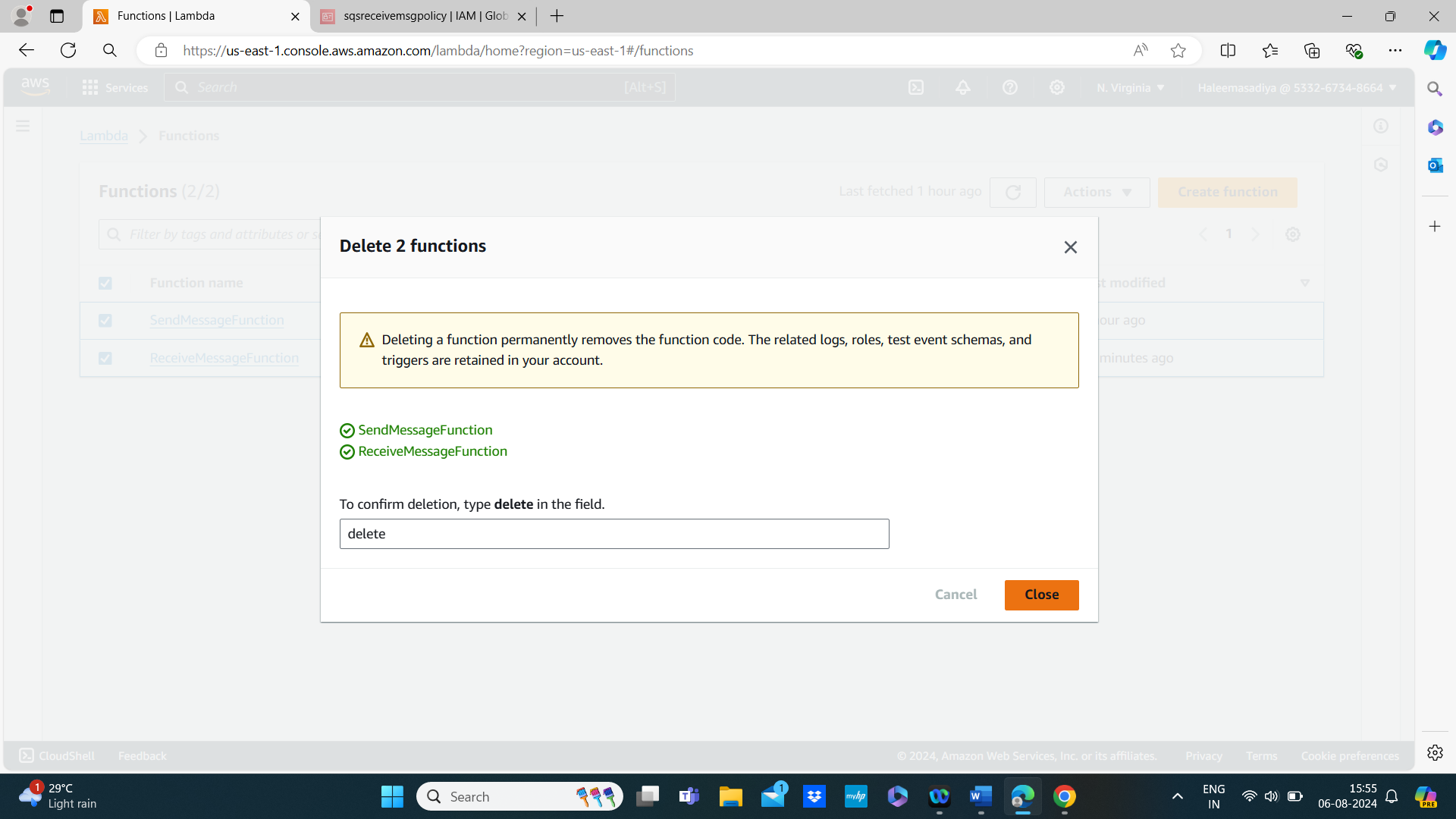
**Conclusion**

This mini project demonstrates the basic capabilities of Amazon SQS in conjunction with AWS Lambda. By sending and receiving messages between Lambda functions and SQS, you’ve set up a basic message queue system. You can expand on this by adding more functionality, such as error handling, message batching, or integrating with other AWS services.



Finally delete all the aws resources to avoid charges





Top of Form

Bottom of Form