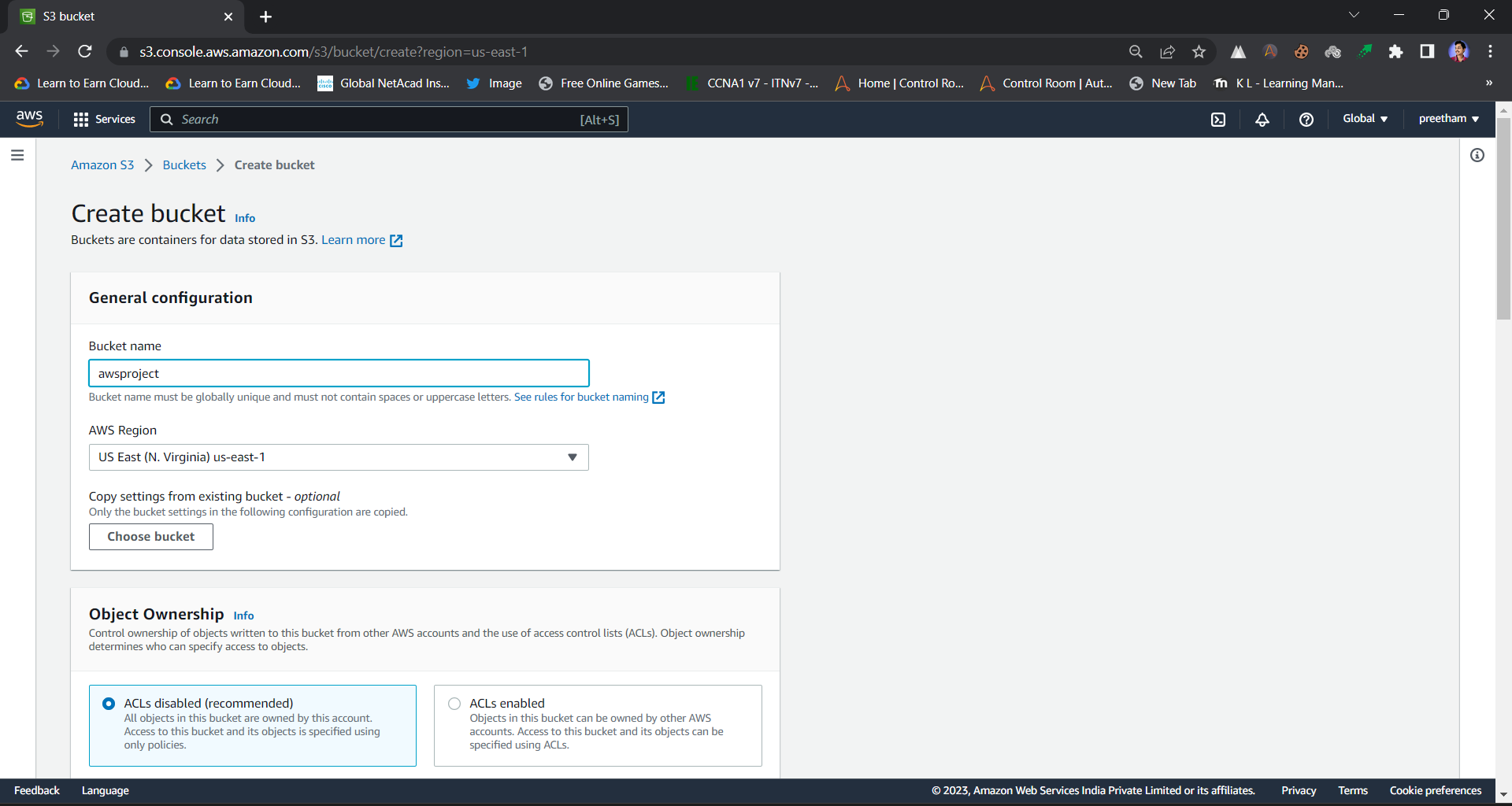
* Hello Everyone!!

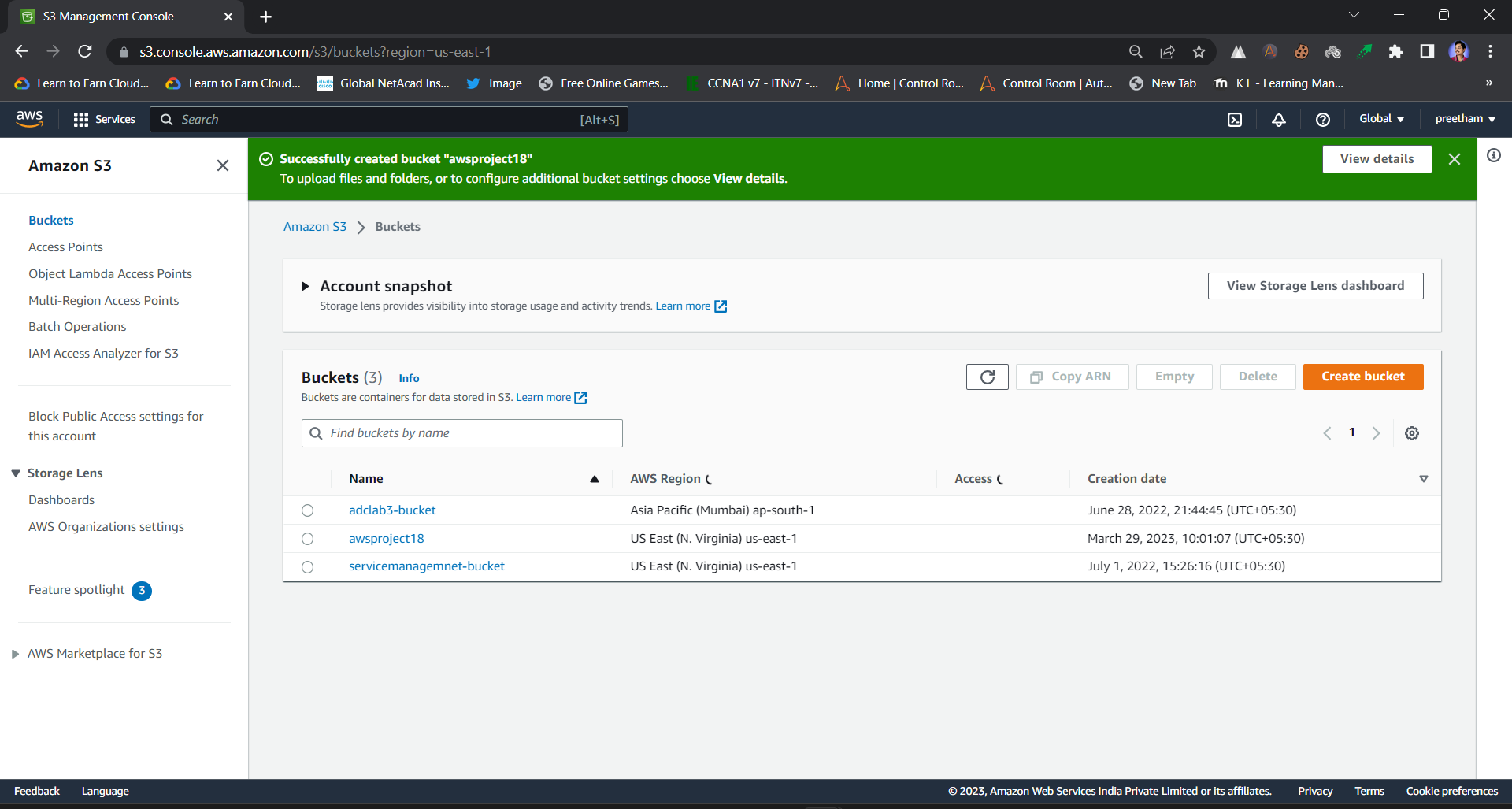
Today I’m here to talk about my project, which I’m showing through this article.It is regarding to the process of how to send email using AWS lambda and SES(simple emailing service)

**Process in Steps:**

**Step 1:**

* First Login to AWS console as a root user using your credential. In the search bar search s3 bucket and click on s3, shown in the suggestion.
* Create a s3 bucket using a globally unique name and leave other option as default.





**Step 2:**

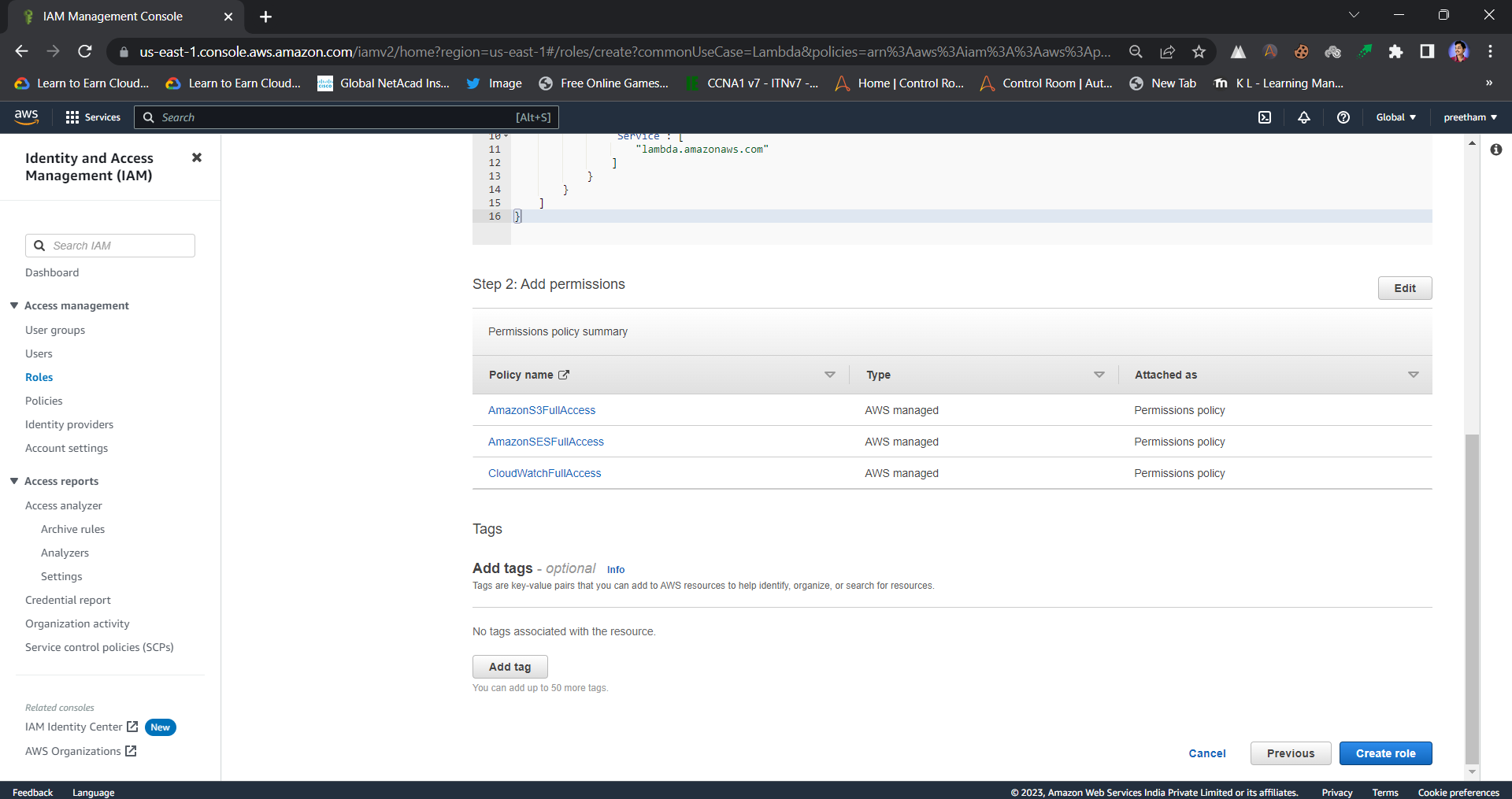
**->**Now Go to search bar and search IAM. Move to IAM console and in the left bar click on “Roles -> create role”.

->Attach 3 polices to the role

1. Cloud Watch Full Access

2. Amazon SES Full Access

3. Amazon S3 Full Access



Graphical user interface, text

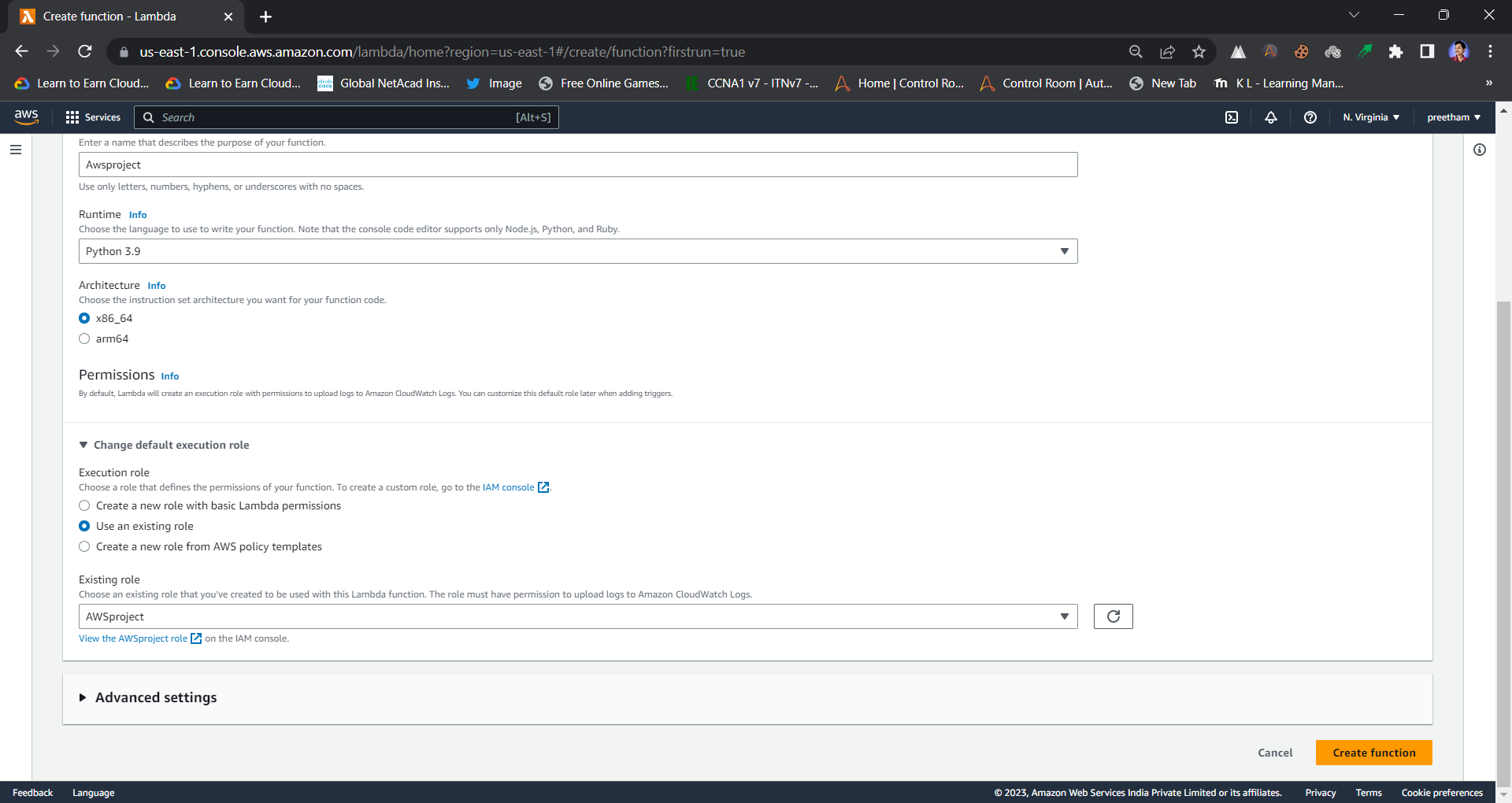
Description automatically generated

**Step 3:**

* Search Lambda in the search bar and create a Lambda function by giving a name to the function and choosing the runtime as “python3.9”.
* Under permission tag choose default execution role to “use an existing role” and choose the particular role that you have created in previous step.
* Now click on create function.

Graphical user interface, text, application

Description automatically generated



Graphical user interface, text

Description automatically generated

**Step 4:**

->Add a trigger to the function by clicking on “Add trigger” button. Choose S3 in trigger configuration and choose the particular bucket that you have created in “step 1”.

-> Now click on “add trigger”

Graphical user interface, application

Description automatically generated

**Step 5:**

* Verify the mail you will use for sending the mail using AWS SES.
* Search and click on “simple email service”. In the left bar click on “verify identity” and create and identity by giving a name to identity and mail address that you are going to use.
* A mail for the verification will be sent that particular mail having a link to verify the mail. Click on the link and then the verification process will be completed.

Graphical user interface, application, Teams

Description automatically generated

**Step 6:**

* Return to the lambda function window and click on “code”. Use the following python code

**Code:**

import json

import boto3

def lambda\_handler(event, context):

file\_name = event['Records'][0]['s3']['object']['key’]

bucketName=event['Records'][0]['s3']['bucket']['name’]

print("Event details : ",event)

print("File Name : ",file\_name)

print("Bucket Name : ",bucketName) subject = 'Event from ' + bucketName client = boto3.client("ses")

body = “””

              <br> This is a notification mail to inform you regarding s3 event. The file {} is inserted in the {} bucket . """.format(file\_name, bucketName)

 message = {"Subject": {"Data": subject}, "Body": {"Html": {"Data": body}}} r

esponse = client.send\_email(Source = "Put the sender's mail id", Destination = {"ToAddresses": ["Put Destination Mail id"]}, Message = message)

print("The mail is sent successfully")

Change the source mail and destination mail and deploy the code.

Graphical user interface, text, application

Description automatically generated

**Step 7:**

* Push any local file into the s3 bucket and after that test the code on clicking on “test” button.
* You can monitor the bucket and lambda function by opening CloudWatch window.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated