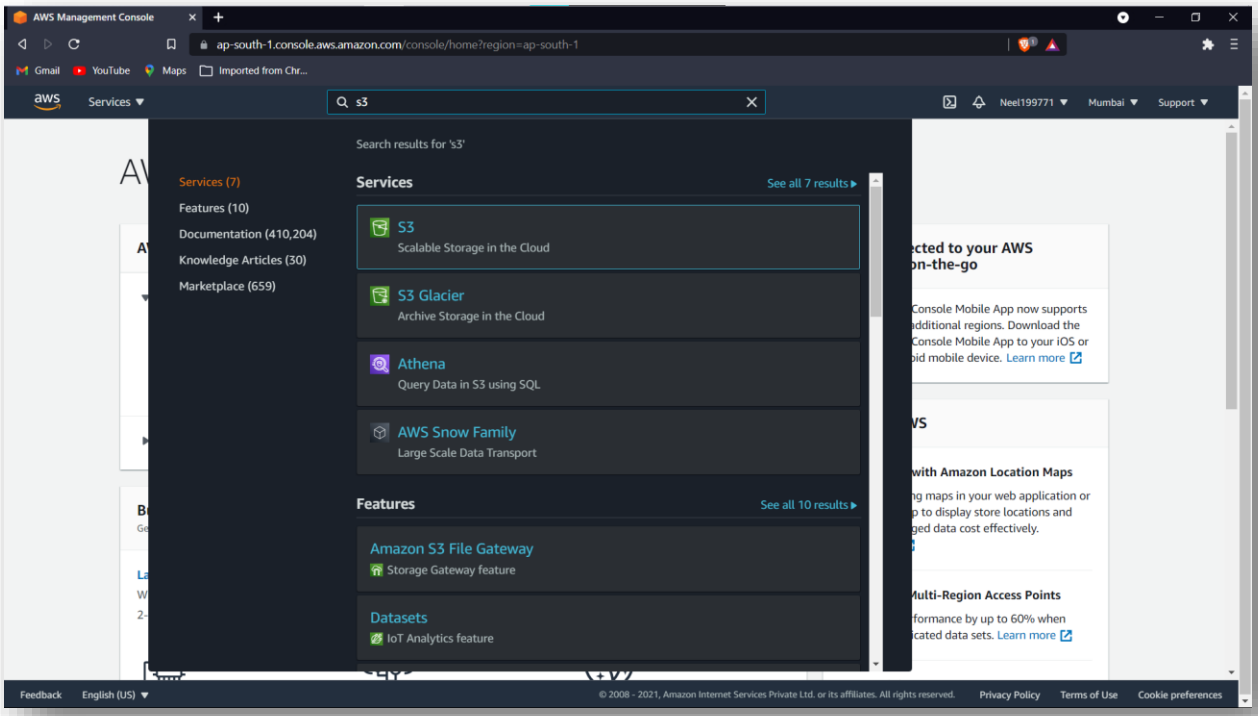
AWS Lambda Function
Lab-5

<u>Aim- To implement lamda Function.</u>

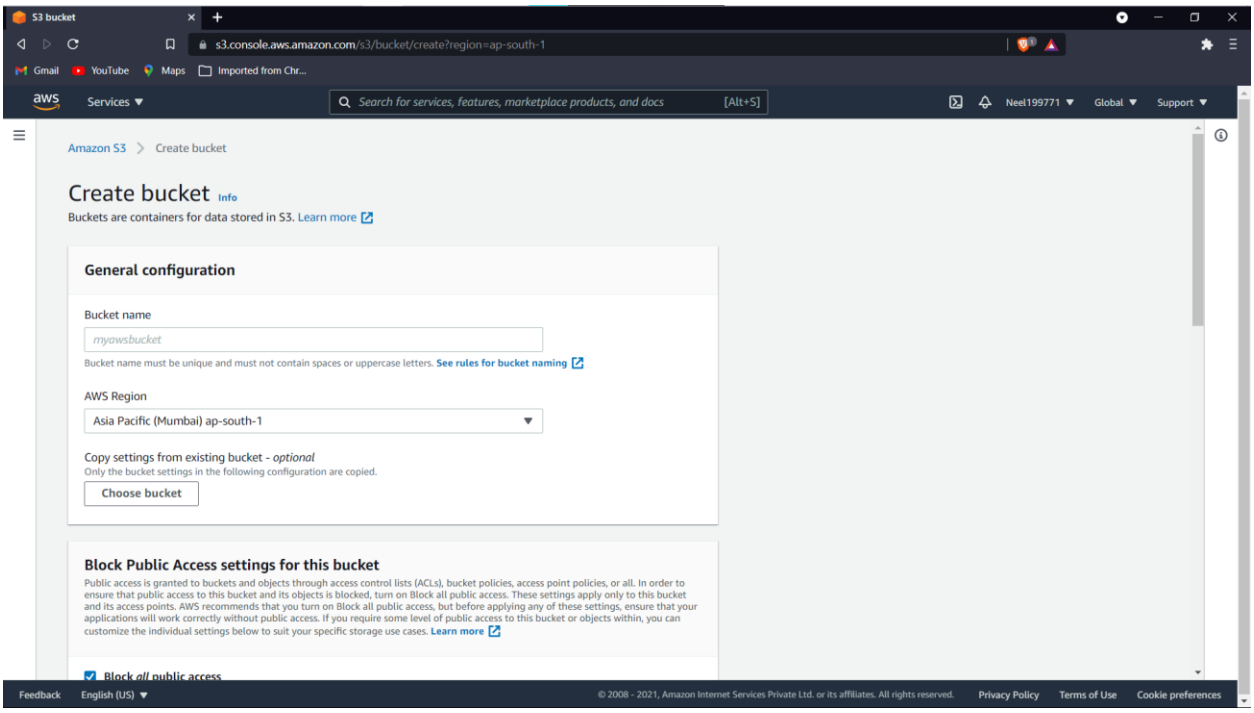
Login to Aws account-

Search S3 ,click on the option below shown-

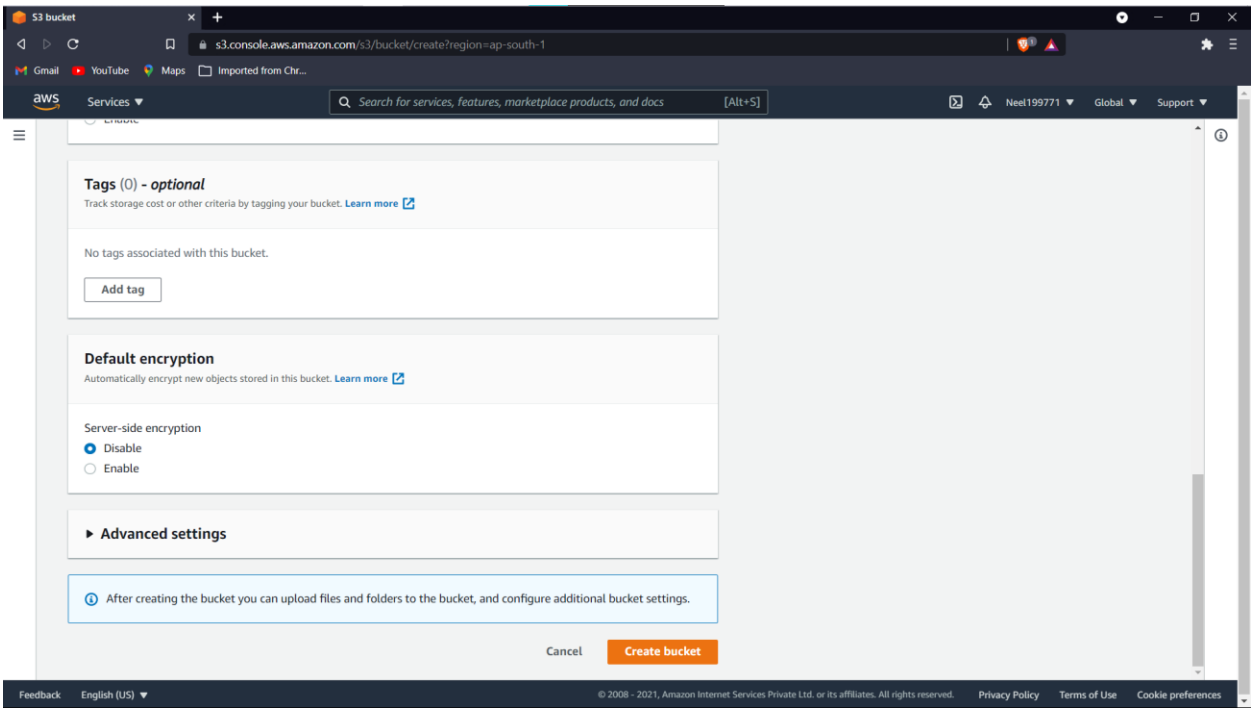


AWS Lambda Function
Lab-5

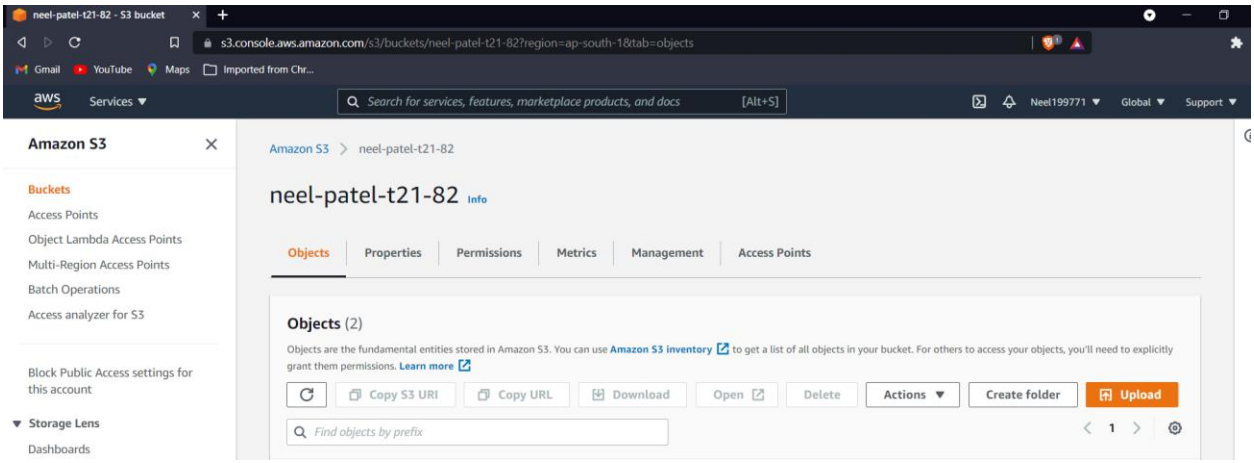
Create an S3 bucket by giving it a name



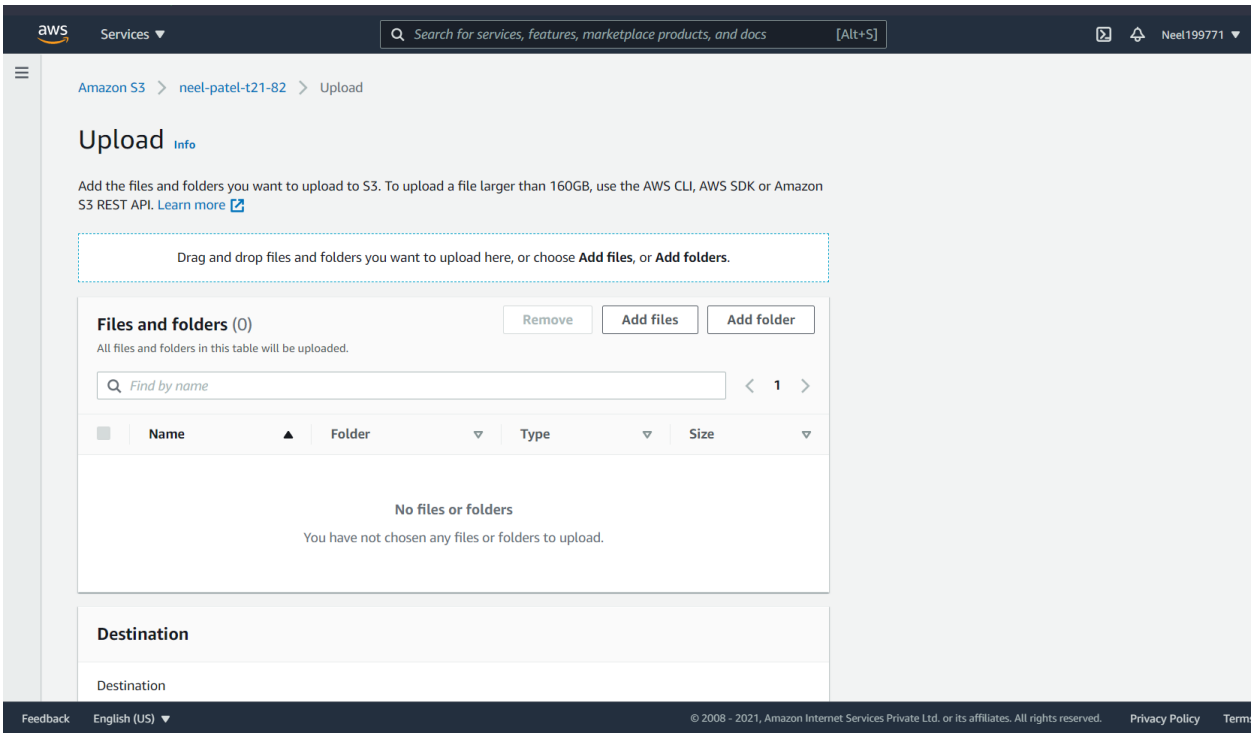
AWS Lambda Function
Lab-5



Click on upload button after the s3 bucket is created in the object section

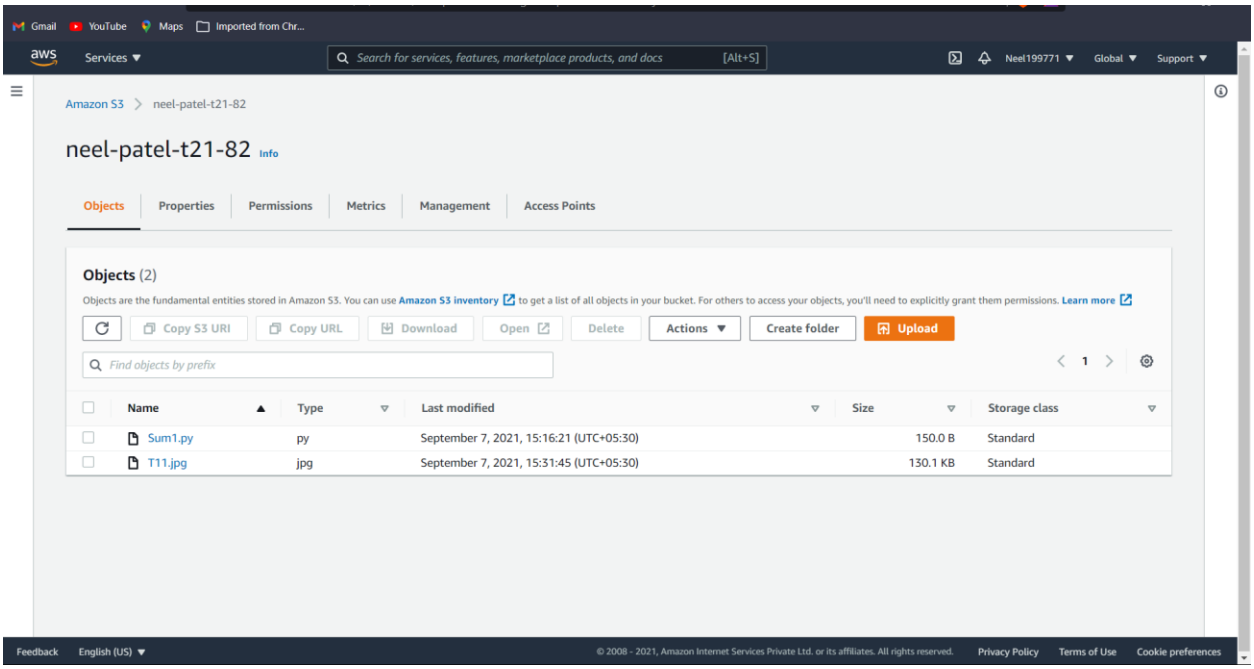


AWS Lambda Function
Lab-5

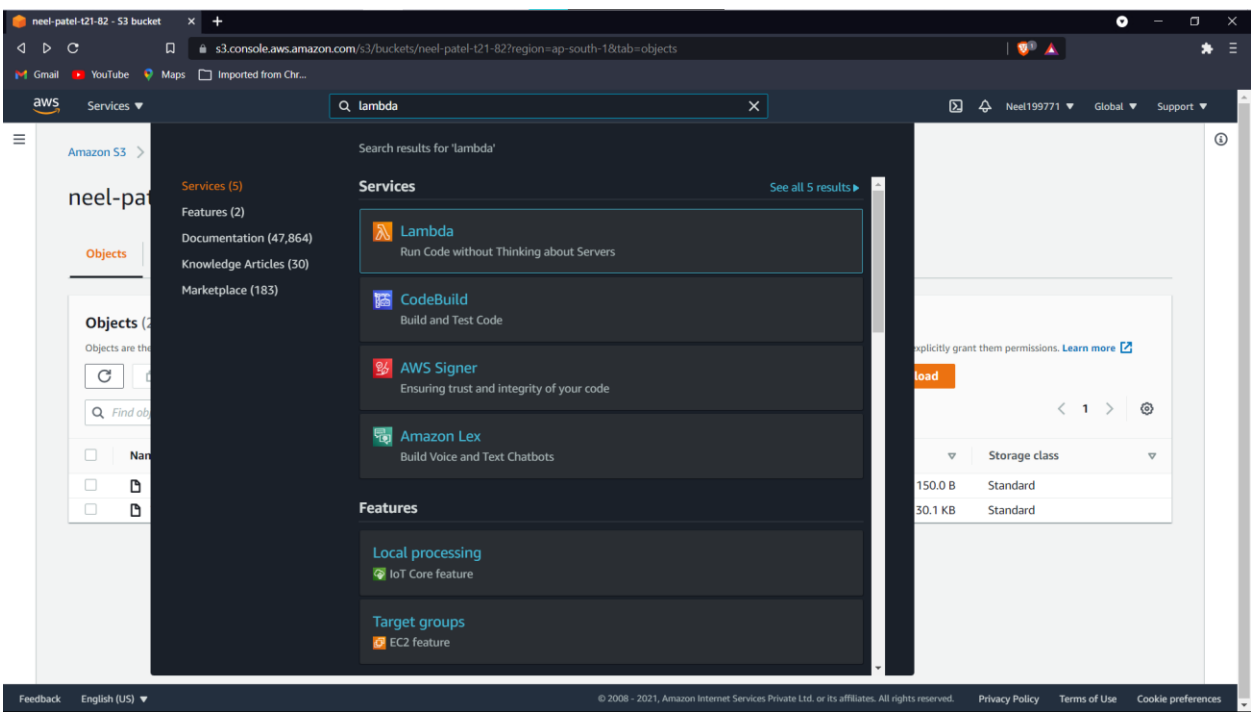


Add any .py or .java extenstion file and click on upload

AWS Lambda Function
Lab-5

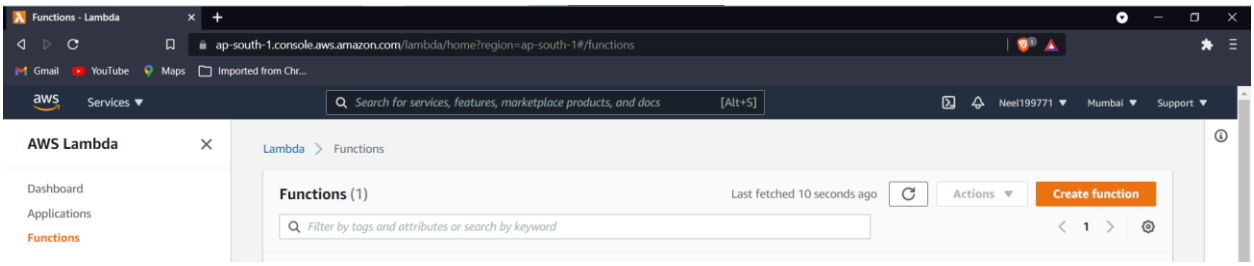


Now search Lamda

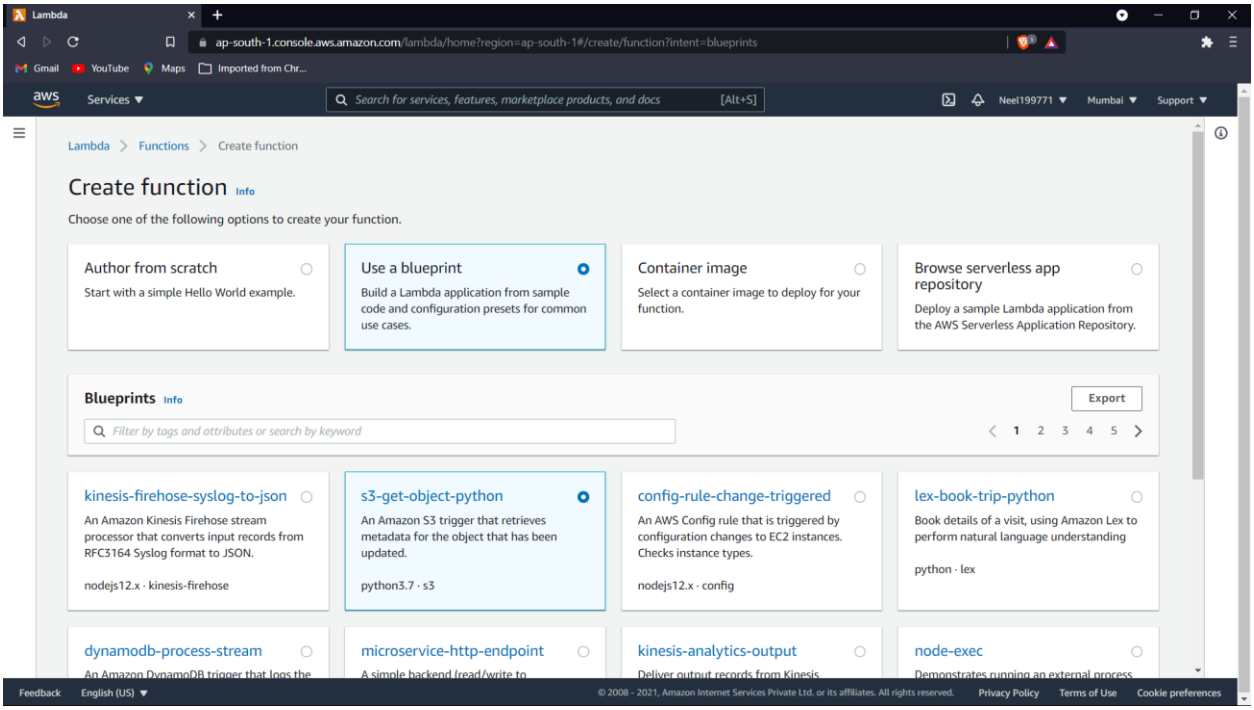


AWS Lambda Function
Lab-5

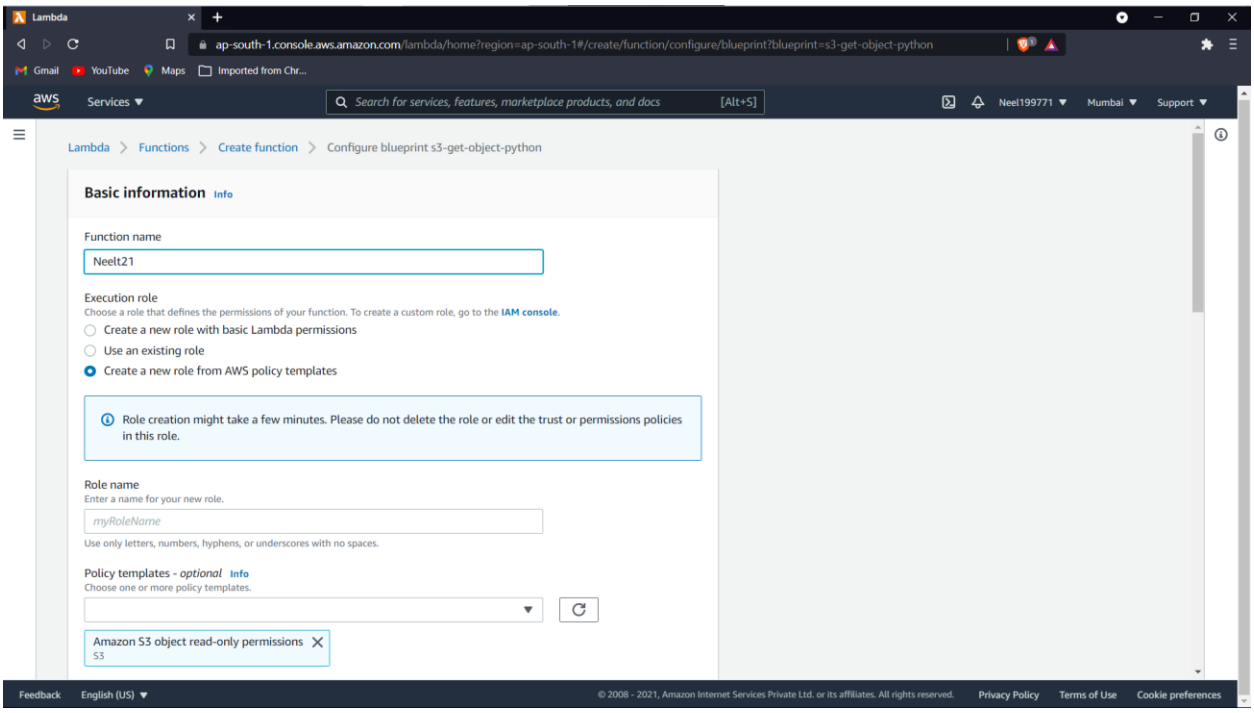
Click create function



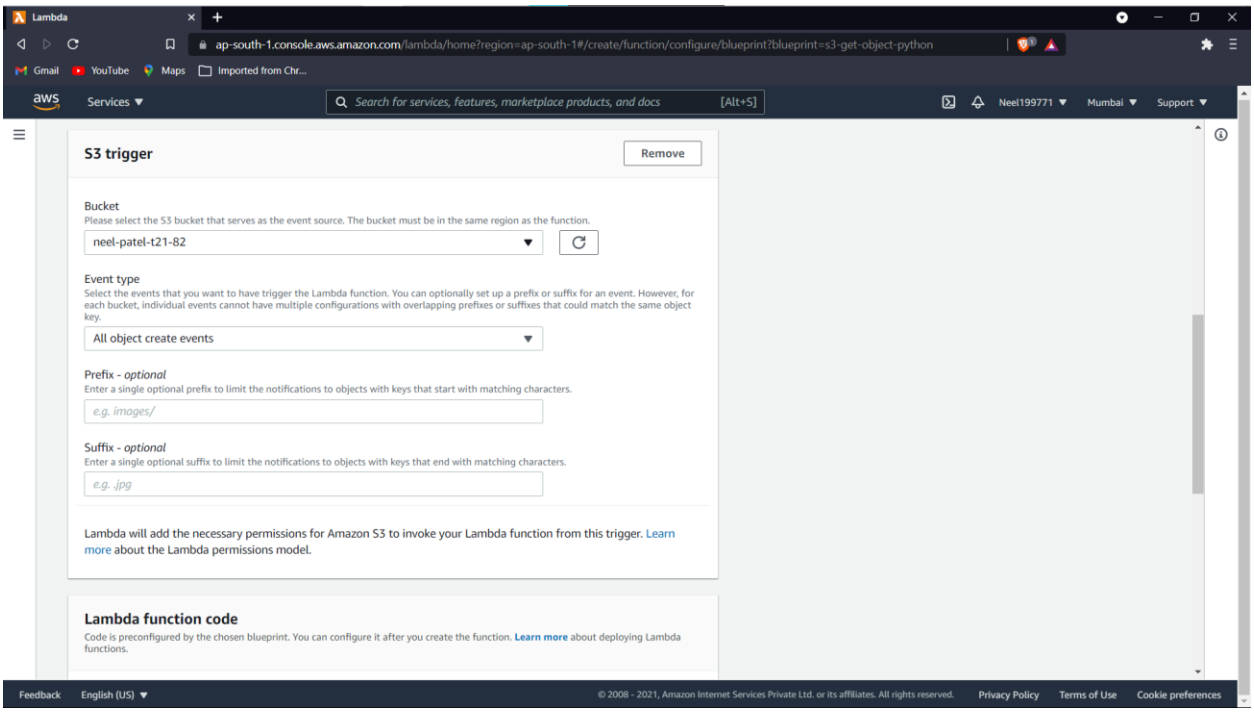
Click on below options and click on configure



AWS Lambda Function
Lab-5

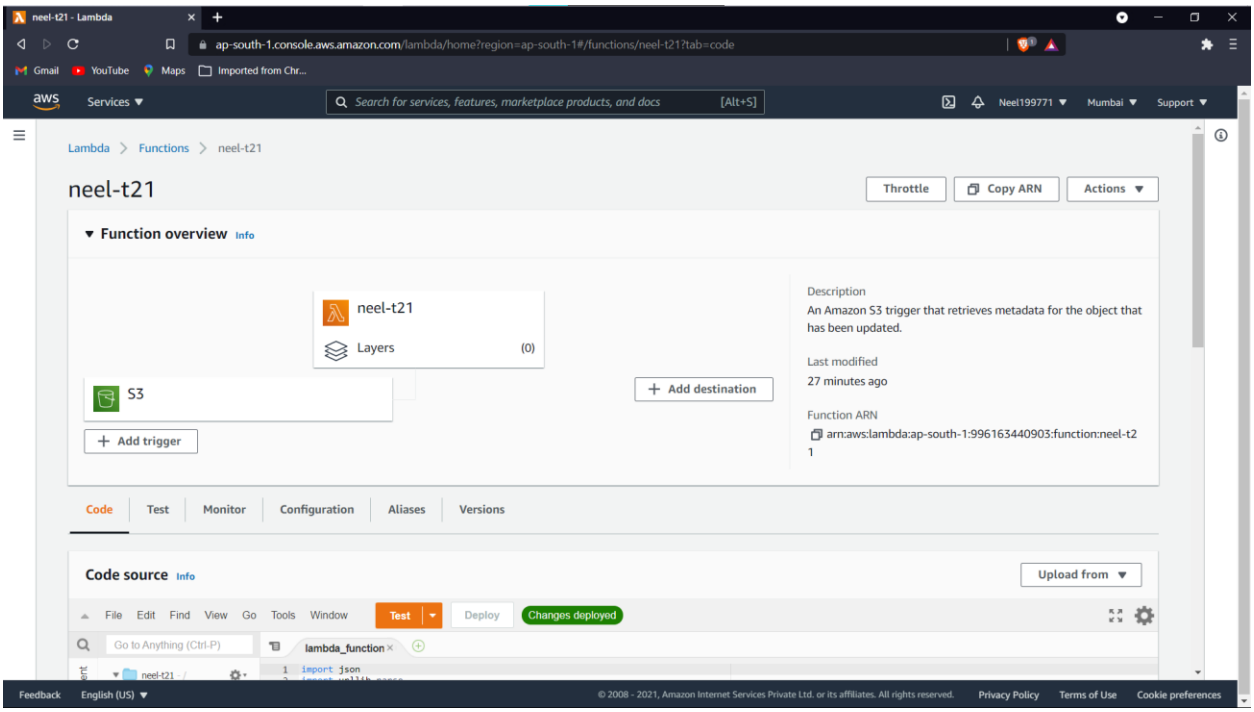


Select the bucket created and create trigger ,click on create function-

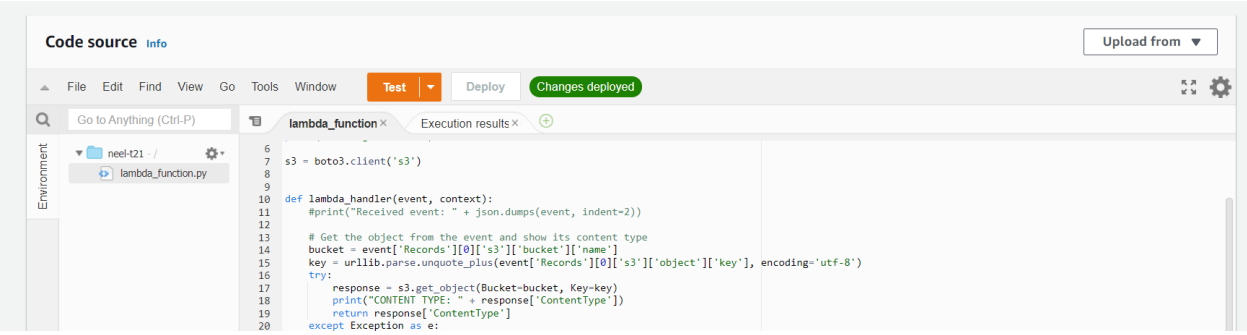


AWS Lambda Function
Lab-5

Check the given trigger is created

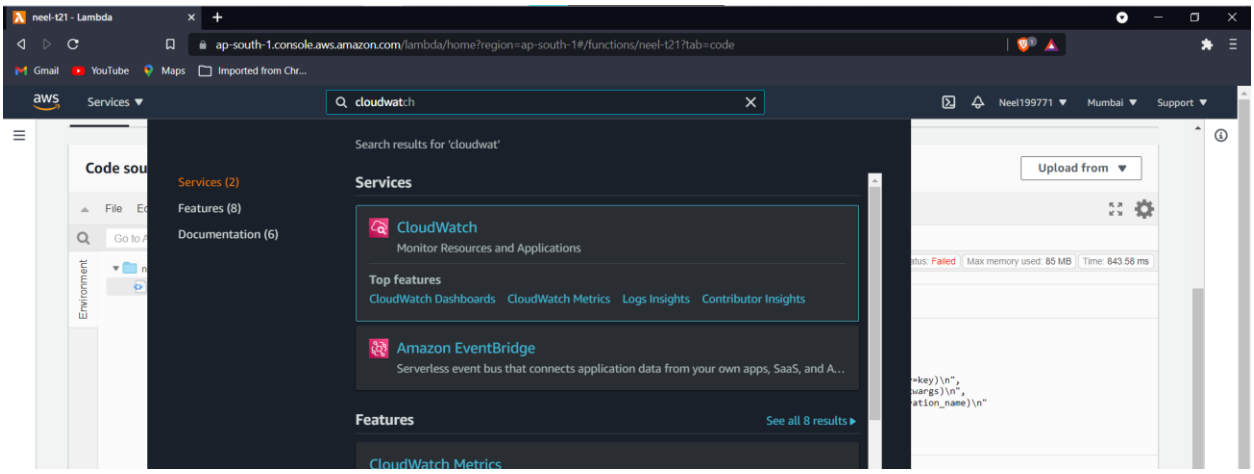


Click on the orange test button-

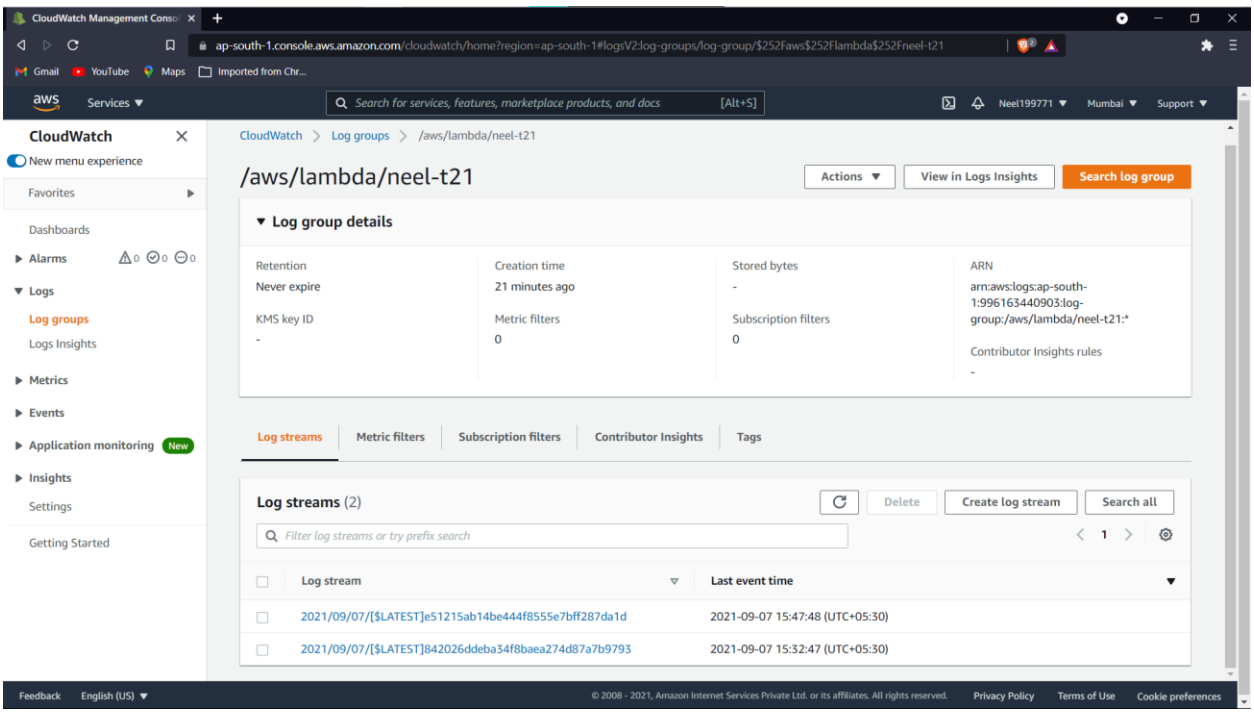


Now,

AWS Lambda Function
Lab-5



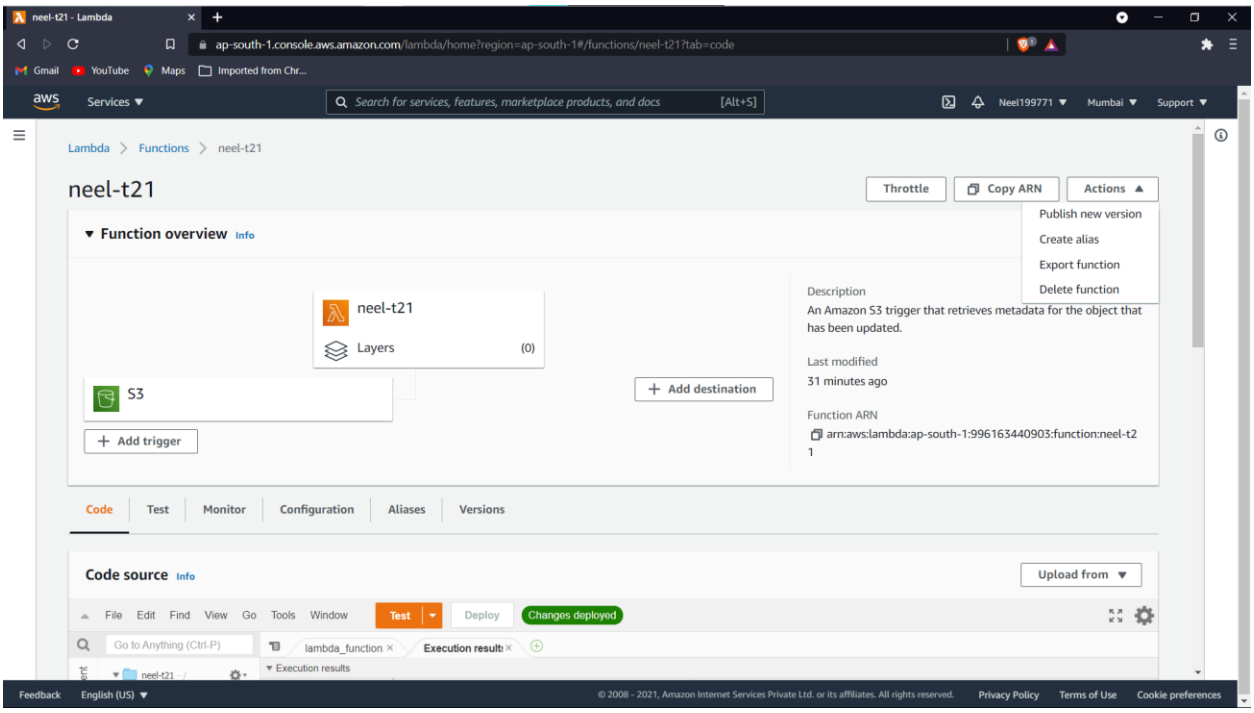
Check the logs of the test-

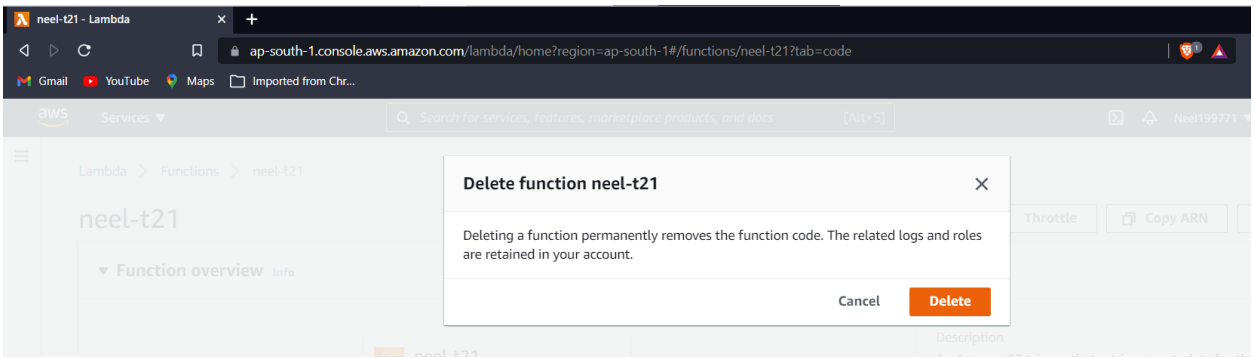


Now terminate-

Click on delete function.

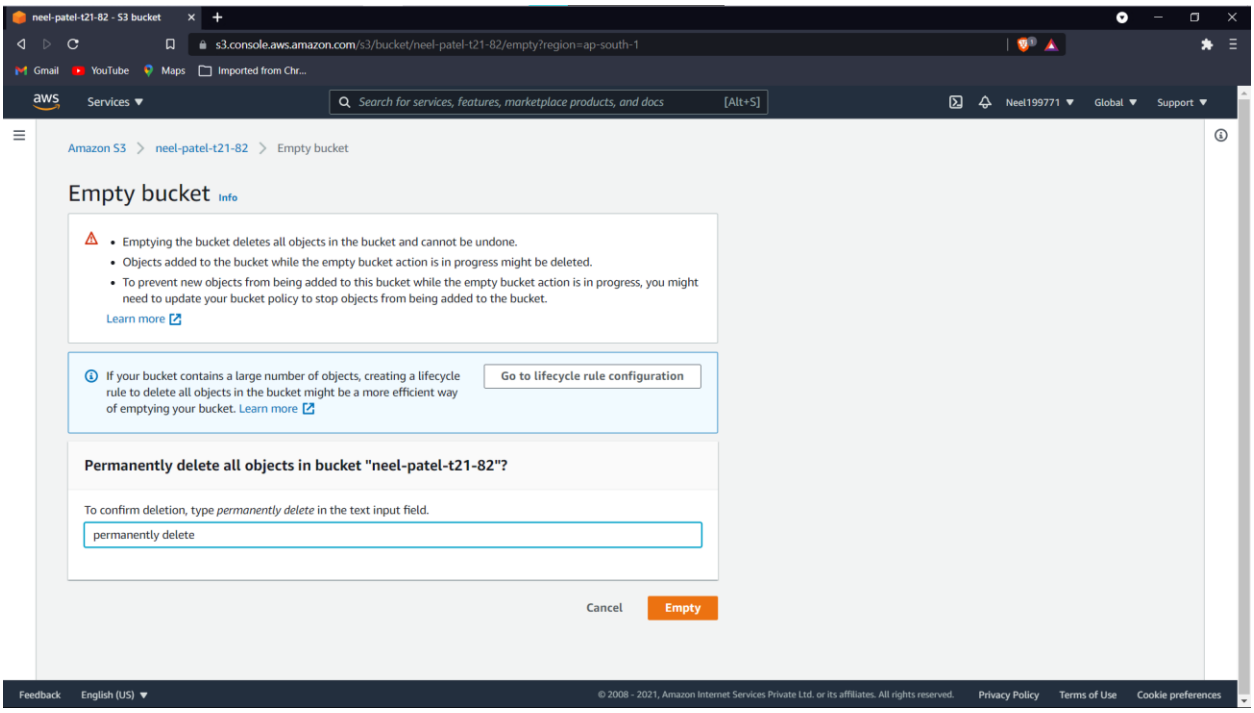
AWS Lambda Function
Lab-5



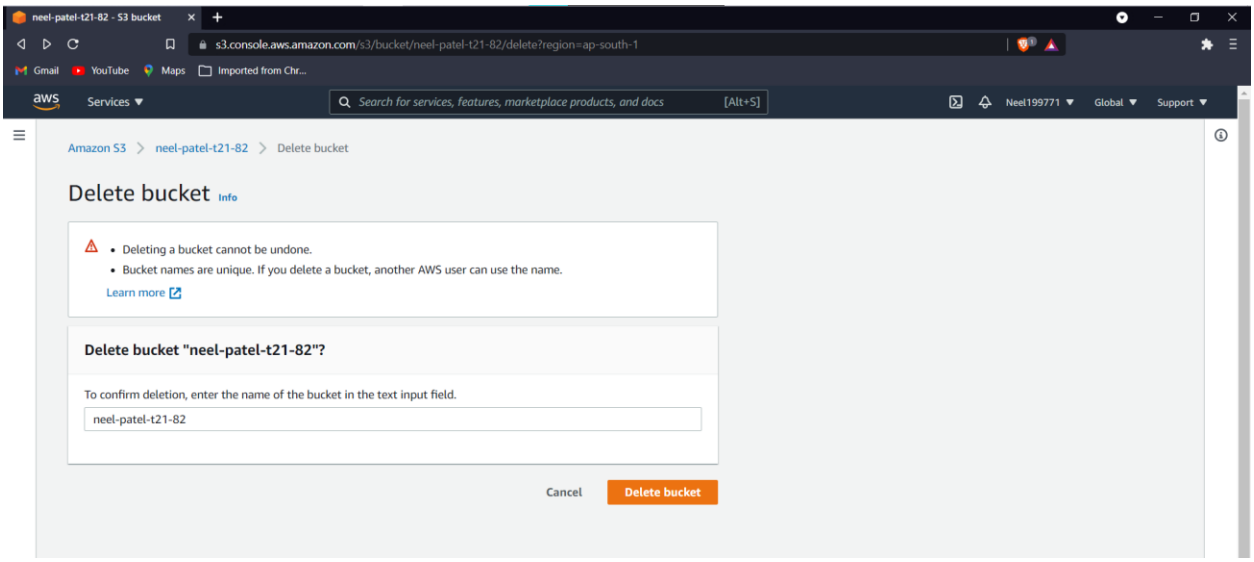


Empty bucket

AWS Lambda Function
Lab-5



Delete bucket-



Step 1: Create a Bucket

- In AWS S3, create a new bucket where you will upload objects.

Step 2: Create a New Policy

AWS Lambda Function
Lab-5

- Go to the IAM (Identity and Access Management) dashboard in AWS.
- Create a new policy.
- When creating the policy, select the JSON tab and paste the provided JSON code. This policy grants necessary permissions for your Lambda function to interact with S3 and CloudWatch Logs.

Step 3: Create a Policy Name

- Name your newly created policy.

Step 4: Create a New Role

- Go to the "Roles" page in AWS IAM.
- Select "Create a new role."
- Under "Policies for Role," select the policy you created earlier.
- Name the role.

Step 5: Upload an Image File

- Upload an image file to the S3 bucket you created in step 1.

Step 6: Create a Lambda Function

- Go to the Lambda dashboard in AWS.
- Create a new function and name it "s3-trigger tutorial."
- Choose the option to "Use an existing role" and select the Lambda execution role you created earlier.
- In the code panel, paste the provided Python code. This code is the Lambda function that will check the content type of the object uploaded to the S3 bucket.

Step 7: Test the Lambda Function

- Create a custom test event named "MyTestEvent."
- For the template, choose "S3 Put."
- In the event JSON, replace values as mentioned in the instructions (region, bucket name, and object key).
- Save the test event and press "Test." The Lambda function will execute, and you'll see the results in the Execution tab.

AWS Lambda Function
Lab-5

Step 8: Check Execution Tab

- In the Execution tab, you can check the "CONTENT TYPE" to see the content type of the uploaded object.
- The function logs should also contain a message like "An object has been added to the S3 Bucket."

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "logs:PutLogEvents",
    "logs:CreateLogGroup",
    "logs:CreateLogStream"
   ],
   "Resource": "arn:aws:logs:*:*:*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "s3:GetObject"
   ],
   "Resource": "arn:aws:s3:::*/*"
  }
 ]
}
```

```
import json
import urllib.parse
import boto3
```

AWS Lambda Function
Lab-5

```
print('Loading function')

s3 = boto3.client('s3')


def lambda_handler(event, context):

    # Get the object from the event and show its content type

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

    key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')


    try:

        response = s3.get_object(Bucket=bucket, Key=key)

        print("An object: " + response['ContentType'] + " has been added to the S3 Bucket")

        print("CONTENT TYPE: " + response['ContentType'])

        return response['ContentType']

    except Exception as e:

        print(e)

        print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))

        raise e
```