

MAJOR PROJECT

PROJECT NAME: Amazon Web Services August Major Project

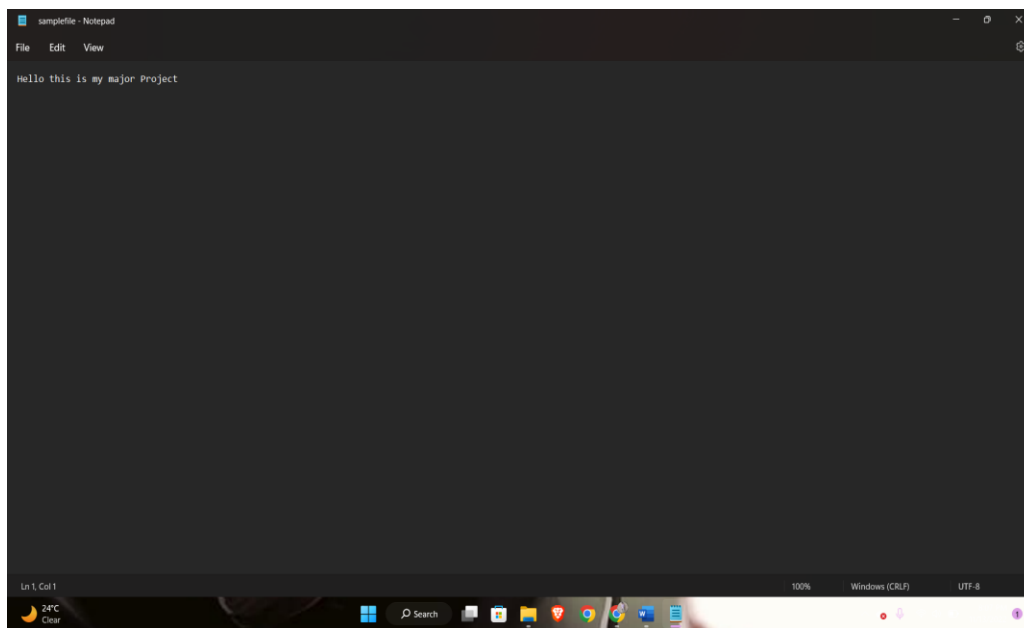
TASKS/CHALLENGES:

Create an AWS Lambda function to count the number of words in a text file. The general requirements are as follows:

TASK-1:

Use any text file.

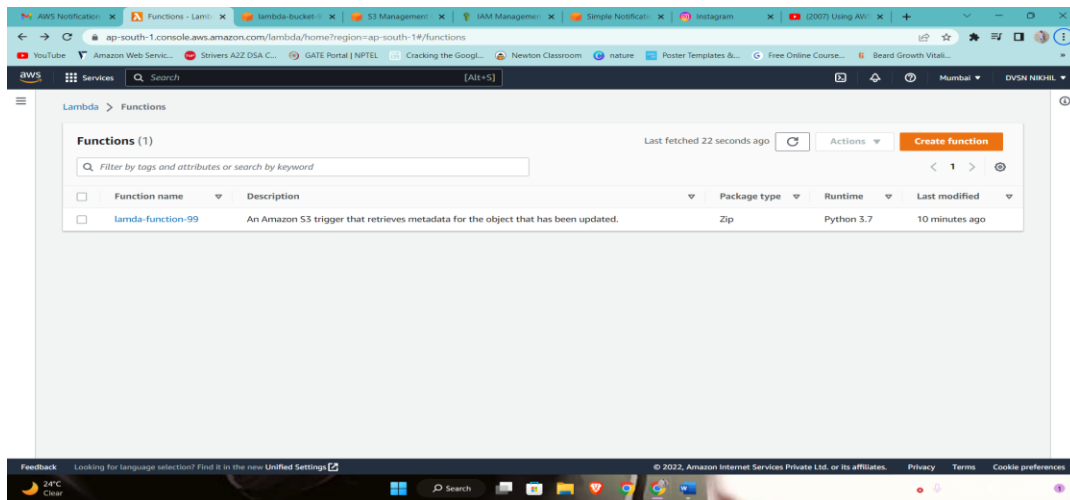
Ex: samplefile.txt



TASK-2:

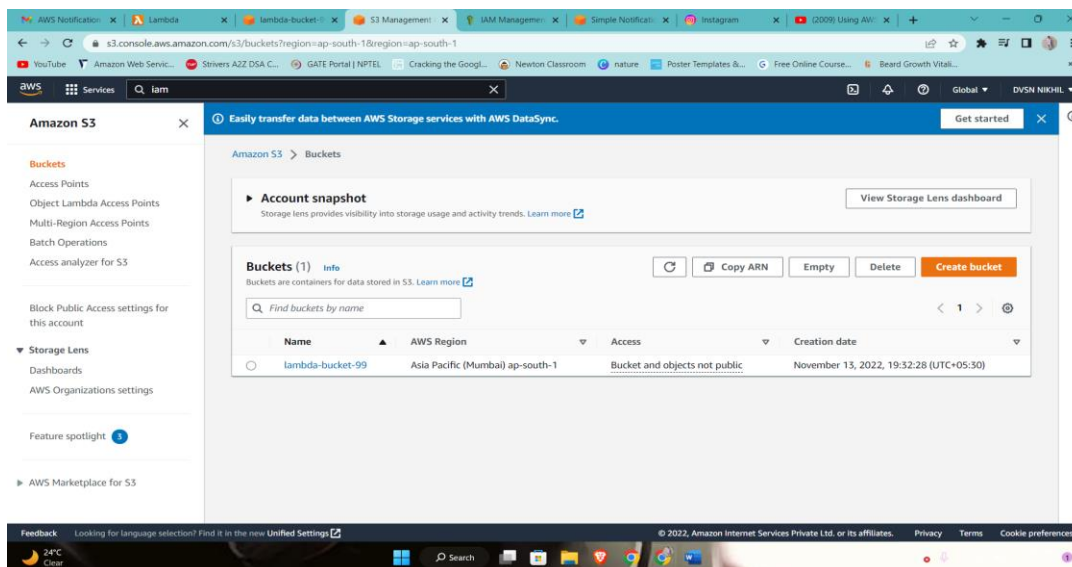
Use the AWS Management Console to develop a Lambda function in Python and to create its required resources.

1.Create a Lambda Function Eg: **lamda-function-99**.



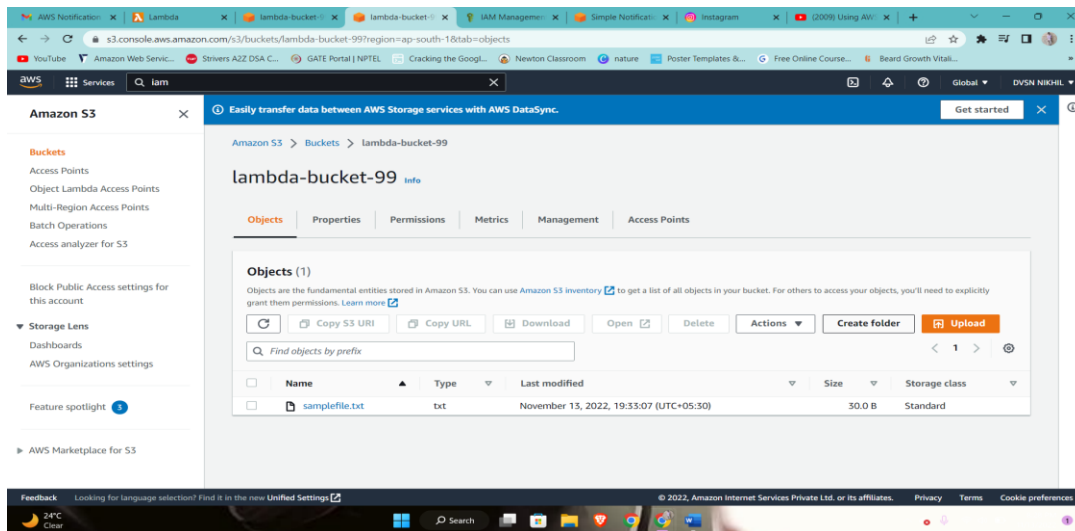
2.Go S3 and create an new bucket by clicking on **create bucket**.

Eg:**lambda-bucket-99**



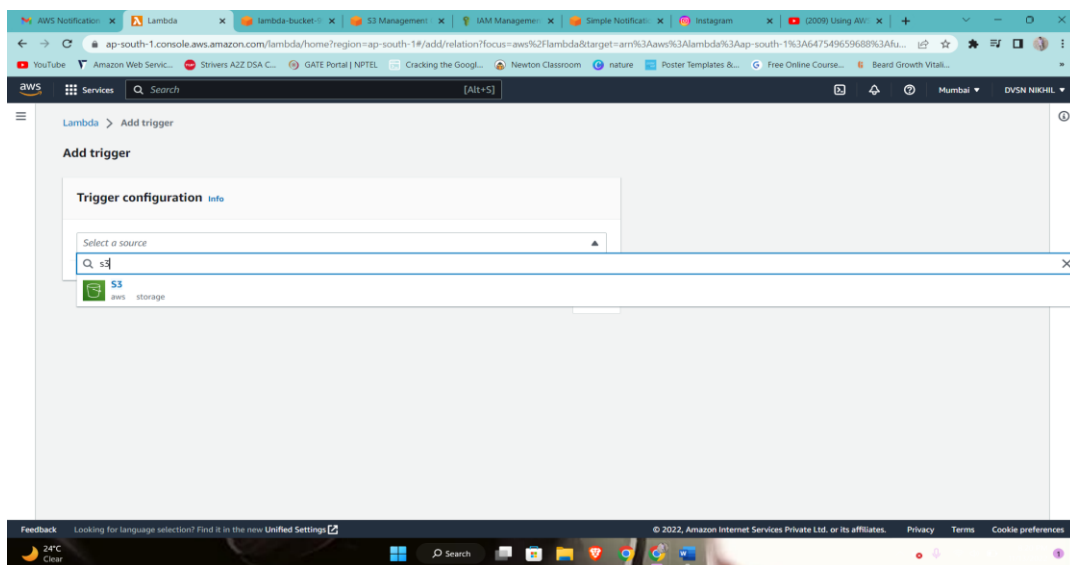
3. Click on the created bucket and go to the **objects** section and click on **Upload**.

4. Select the **samplefile.txt** file from the PC and click on upload.



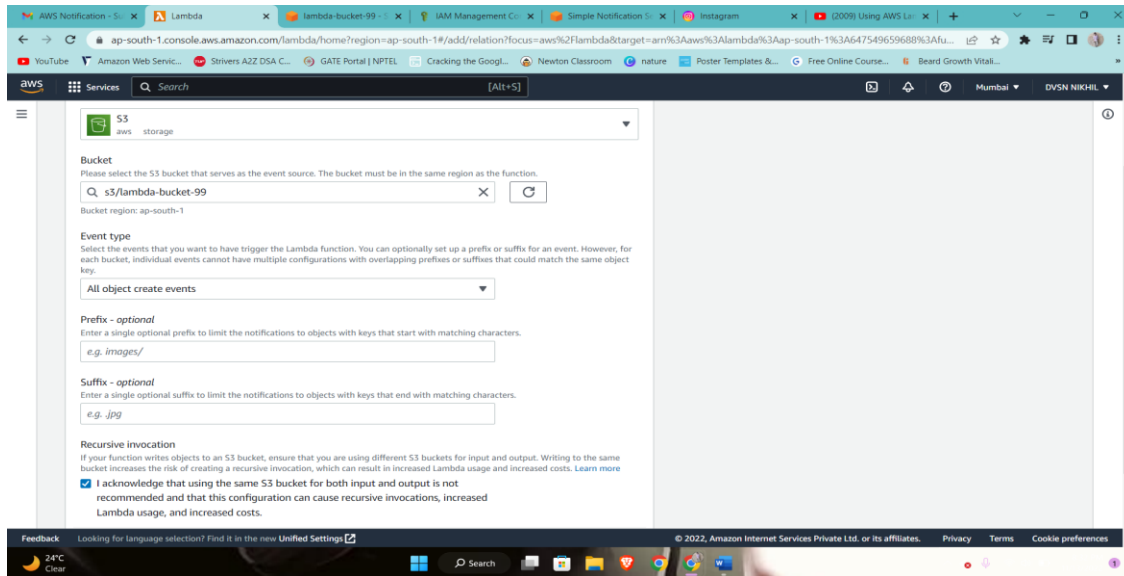
4. Go to the lambda Function<lambda-function-99> and click on **add trigger**.

5. In **Trigger configuration** select a source **S3**.



6. Specify the bucket name which we have created earlier **lambda-bucket-99**.

7. Check the checkbox and click on **Add**.



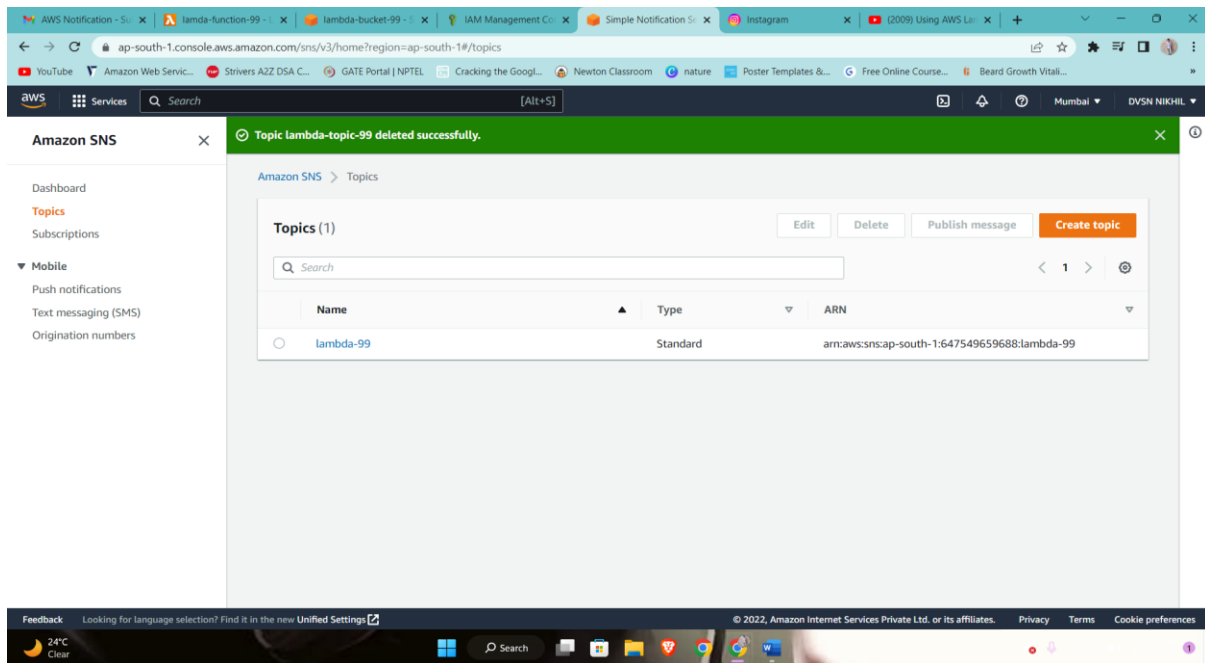
8. S3 is triggered to the lambda function

Task-3:

Report the word count in an email using an Amazon Simple Notification Service (SNS) topic.

Optionally, also send the result in an SMS (text) message.

1. Create the **SNS Topic** by giving a name of Eg: **lambda-99**



2. After creating the Topic then create a subscription by clicking on **Create Subscription**.

3. Give the required details for creating the subscription as shown in the screenshot, and click on **create subscription**.

The screenshot shows the 'Create subscription' page in the AWS console. The 'Details' section contains the following fields:

- Topic ARN:** A text box containing 'arn:aws:sns:ap-south-1:647549659688:lambda-99'.
- Protocol:** A dropdown menu with 'Email' selected.
- Endpoint:** A text box containing 'dvsnikhil@gmail.com'.

Below the 'Endpoint' field, there is a blue box with the text: 'After your subscription is created, you must confirm it. info'.

At the bottom of the 'Details' section, there is a link for 'Subscription filter policy - optional info'.

4. Subscription will be created as shown in screenshot.

The screenshot shows the 'Amazon SNS' page in the AWS console. The 'Details' section displays the following information:

- Name:** lambda-99
- ARN:** arn:aws:sns:ap-south-1:647549659688:lambda-99
- Type:** Standard
- Display name:** -
- Topic owner:** 647549659688

Below the 'Details' section, there are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery retry policy (HTTP/S)', 'Delivery status logging', 'Encryption', and 'Tags'. The 'Subscriptions' tab is selected.

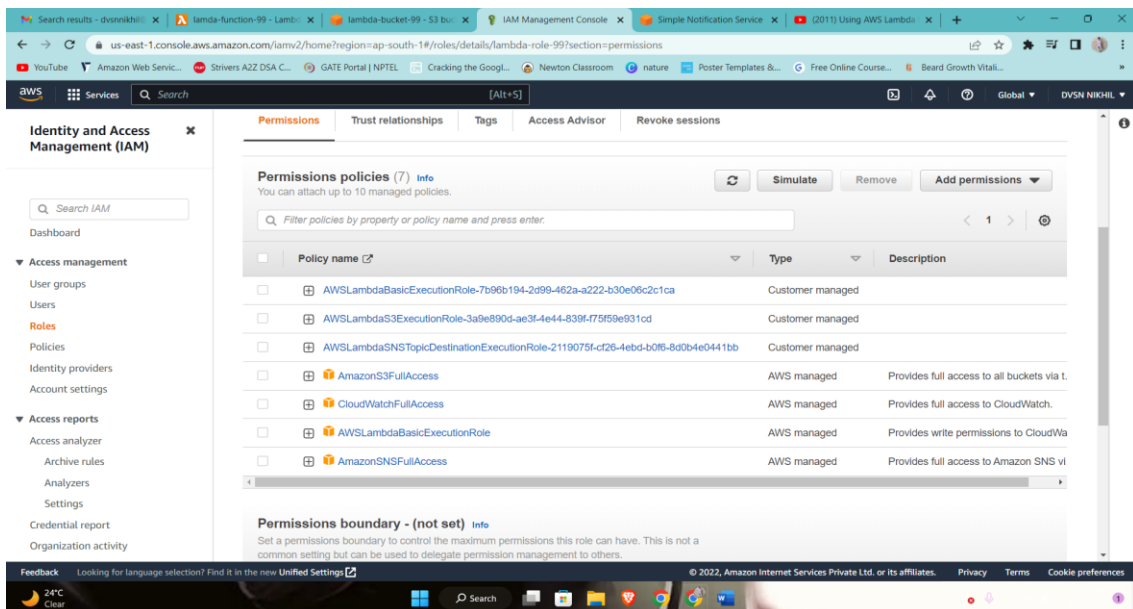
The 'Subscriptions (1)' section shows a table with one subscription:

ID	Endpoint	Status	Protocol
fee33cfe-9f85-4bb8-978d-a7f5d2cd8329	dvsnikhil@gmail.com	Confirmed	EMAIL

Task-4:

You will need an AWS Identity and Access Management (IAM) role for the Lambda function to access other AWS services.

1. Select the **lambda-role-99** in IAM Roles
2. And Click on **Add permissions**
3. Select following lambda functions:
 - AWSLambdaBasicExecutionRole
 - AmazonSNSFullAccess
 - AmazonS3FullAccess
 - CloudWatchFullAccess



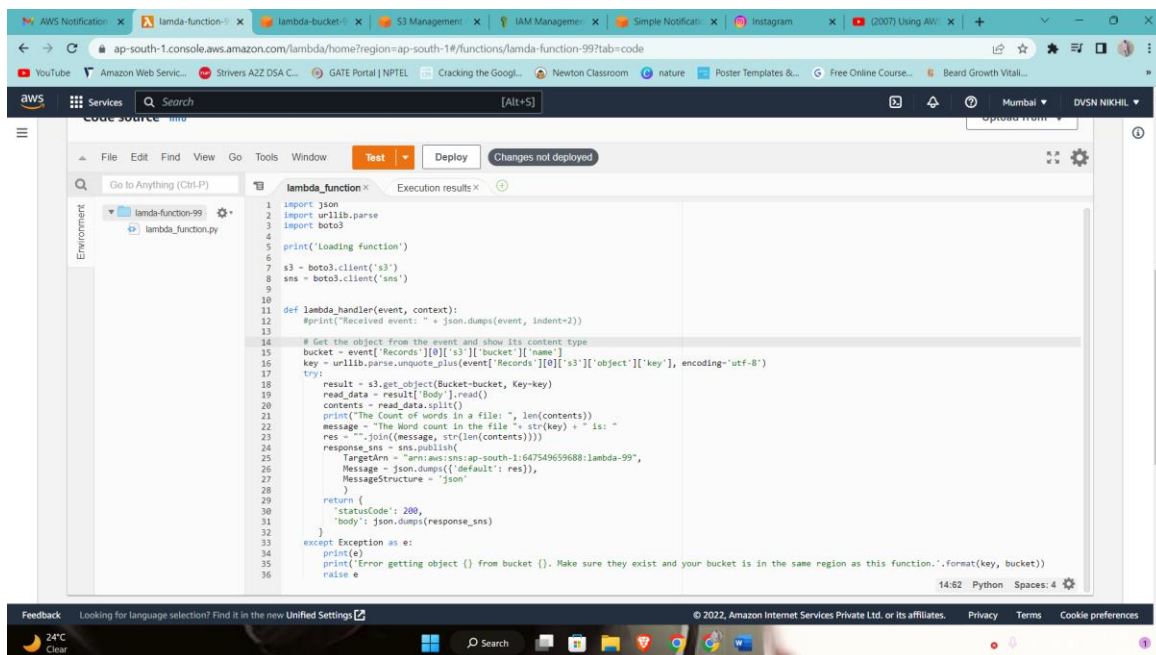
Task-5:

Format the response message as follows:

The Word count in the file **sampletext.txt** is n.

Specify the email subject line as: Word Count Result

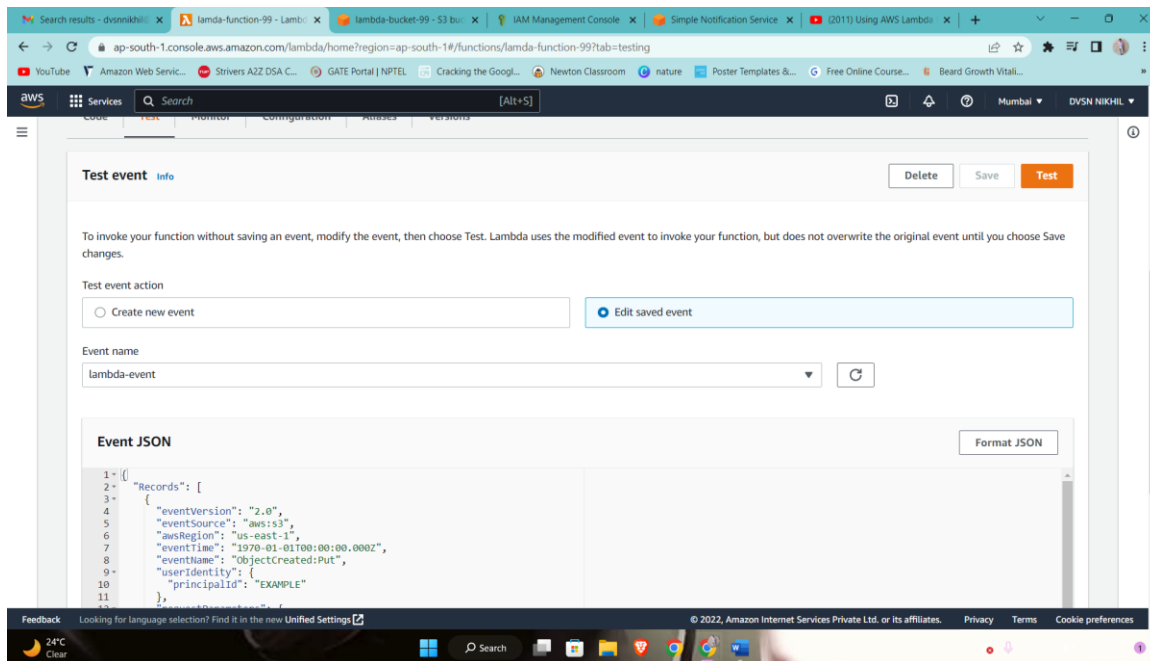
1. Click on the created lambda function and go to the code section and type the following code as shown below in screenshot.



The screenshot shows the AWS Lambda console interface. The browser address bar indicates the URL: `ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions/lambda-function-99?tab=code`. The console displays the code for a lambda function named 'lambda_function-99'. The code is written in Python and uses the `boto3` library to interact with S3 and SNS. It reads a file from an S3 bucket and publishes a message to an SNS topic. The code is as follows:

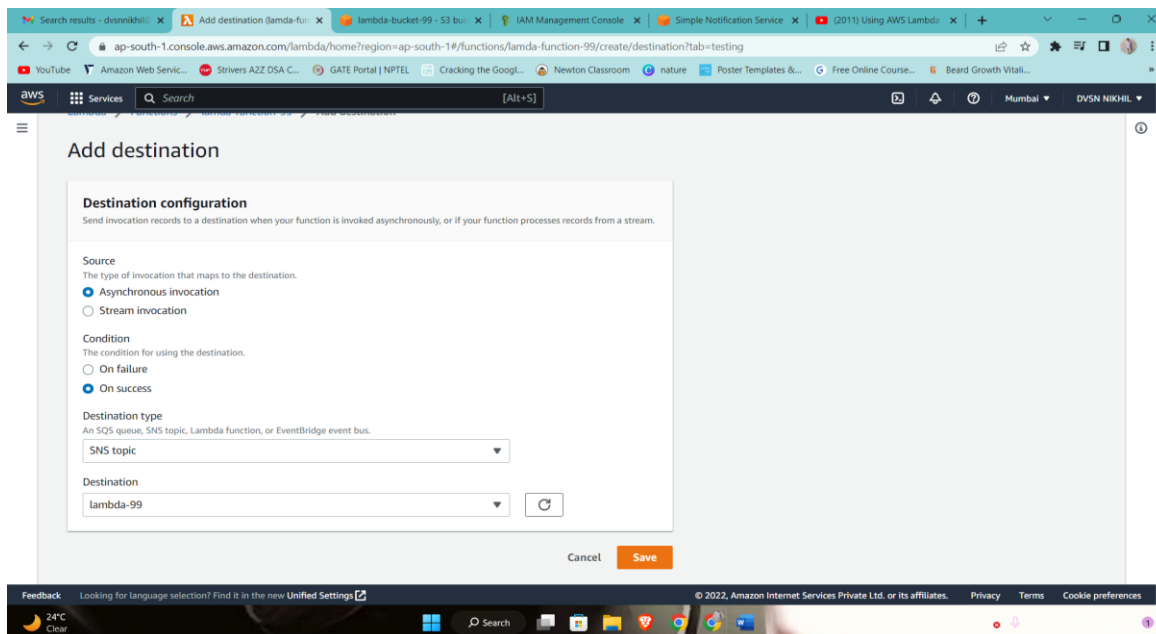
```
1 import json
2 import urllib.parse
3 import boto3
4
5 print('Loading function')
6
7 s3 = boto3.client('s3')
8 sns = boto3.client('sns')
9
10 def lambda_handler(event, context):
11     # Print the received event
12     print('Received event: ' + json.dumps(event, indent=2))
13
14     # Get the object from the event and show its content type
15     bucket = event['Records'][0]['s3']['bucket']['name']
16     key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')
17     try:
18         result = s3.get_object(Bucket=bucket, Key=key)
19         read_data = result['Body'].read()
20         contents = read_data.split()
21         print('The count of words in a file: ', len(contents))
22         message = "The Word count in the file " + str(key) + " is: "
23         res = "".join(message, str(len(contents)))
24         response_sns = sns.publish(
25             TargetArn = "arn:aws:sns:ap-south-1:647549659688:lambda-99",
26             Message = json.dumps({'default': res}),
27             MessageStructure = 'json'
28         )
29         return {
30             'statusCode': 200,
31             'body': json.dumps(response_sns)
32         }
33     except Exception as e:
34         print(e)
35         print('Error getting object {} from bucket {}. Make sure they exist and your bucket is in the same region as this function.'.format(key, bucket))
36         raise e
```

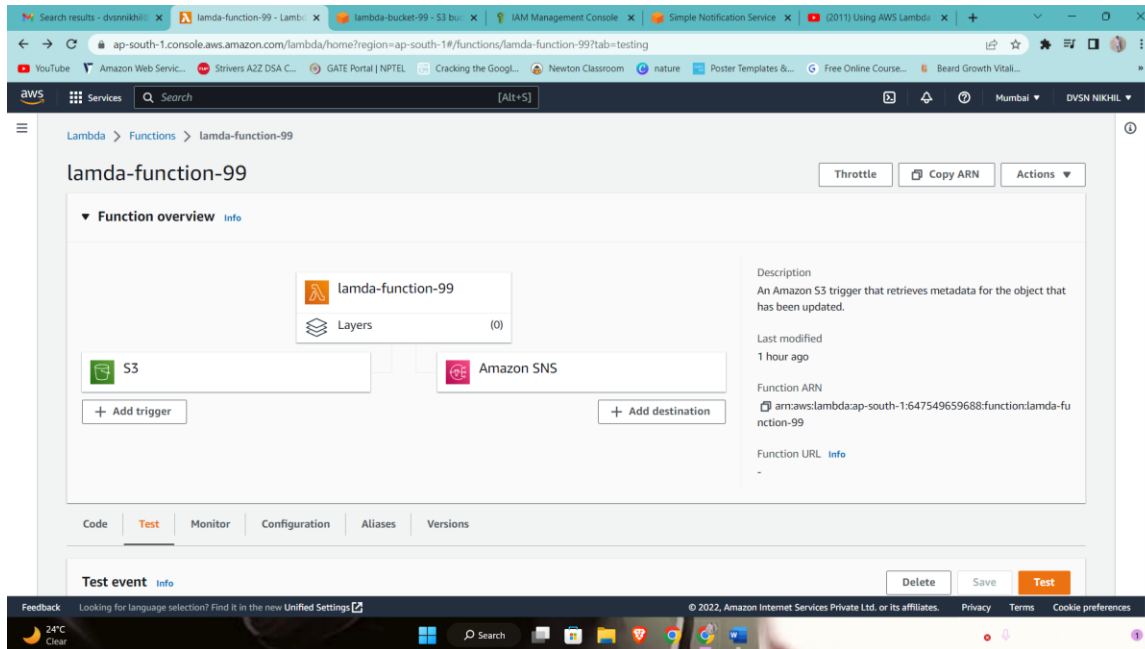

2. Test the code and Deploy it.



Task-6:

1. Now click on **add destination** in lambda function
2. Fill the details as like in screenshot, and click on **Save**





3.As we can see from the above screenshot the trigger and the destination is added to the lambda function.

4. Now Run the code the for the final testing and you can see the result.

5. You will get a notification to the specified email

dvsnnikhil@gmail.com as:

“ The Word Count of the samplefile.txt is 25.”

Thankyou.