\*\*\* AWS Lambda Functions\*\*\*

Question: Trigger Lambda Function on Object Upload to S3

Task: Set up an AWS Lambda function to automatically execute whenever a new object is uploaded to an S3 bucket.

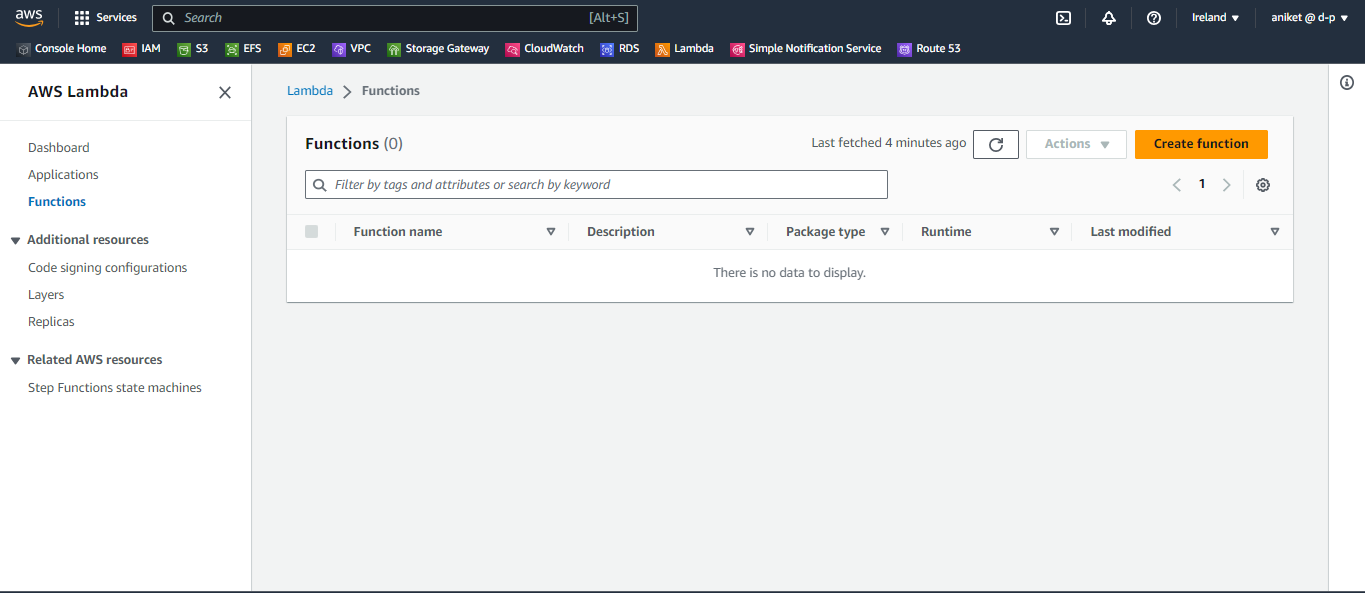
Create a Lambda function that logs a message when triggered.

Configure an S3 bucket to send an event notification to the Lambda function upon object creation and delation.

Upload a sample object to the S3 bucket and verify that the Lambda function is triggered and the message is logged.

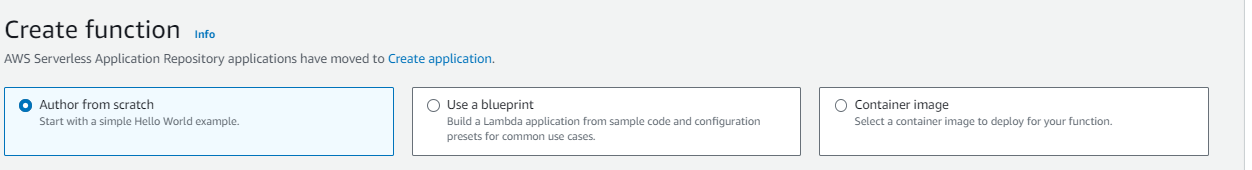
Step 1:-

First of all go to lambda service and then click on create function.



Step 2:-

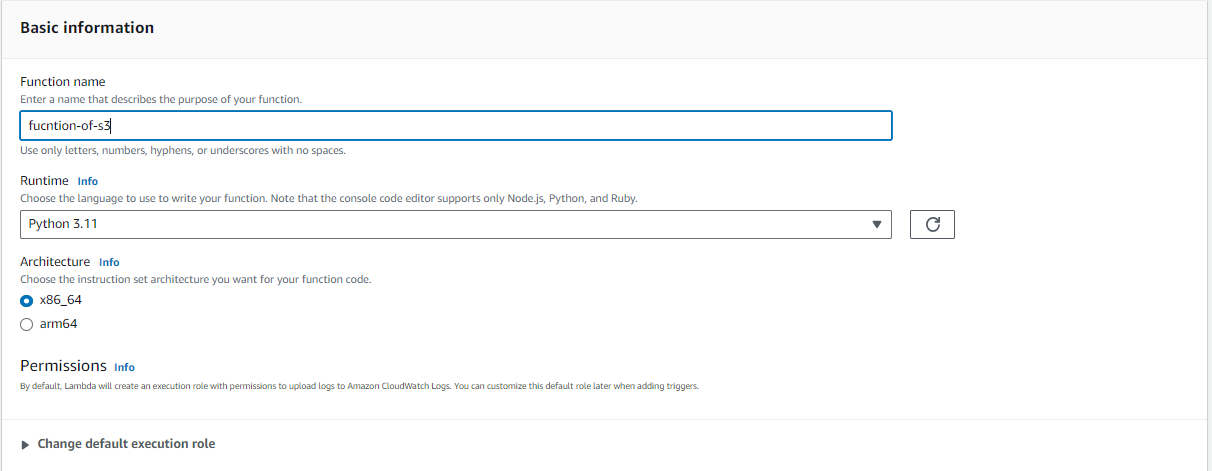
Now you have see the create function setting so first select author from scratch in create function.



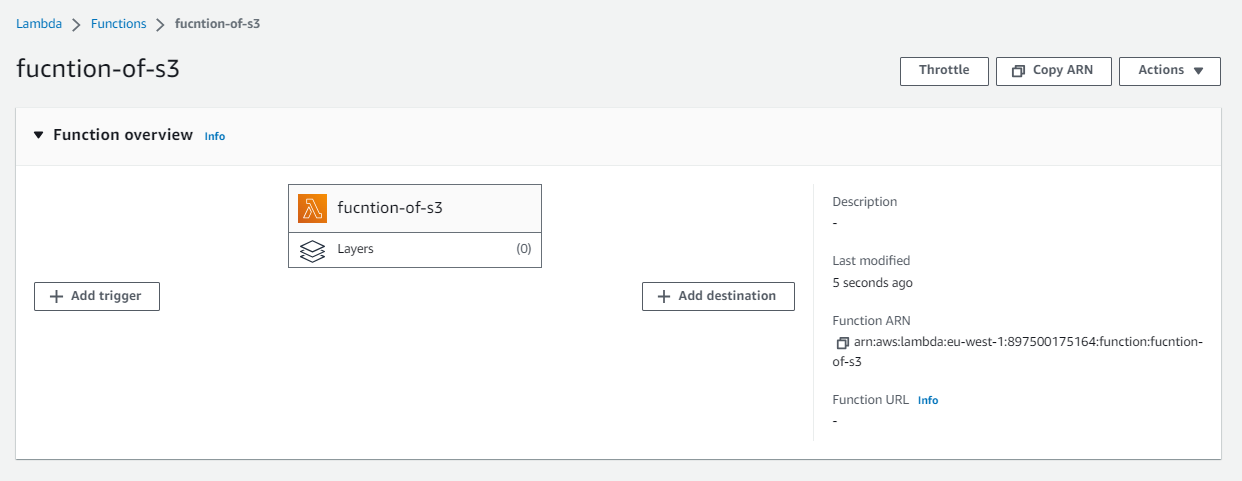
Step 3:-

Now gave name to your function and then select runtime language and the select architecture

And then click on create function.

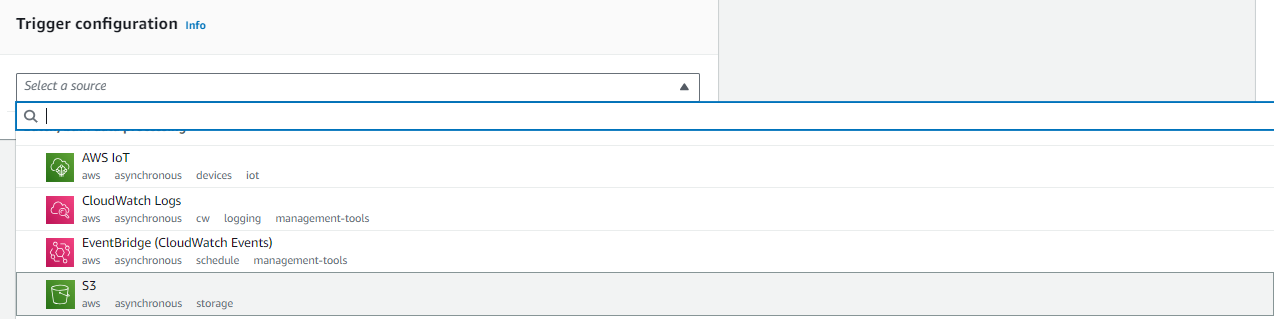


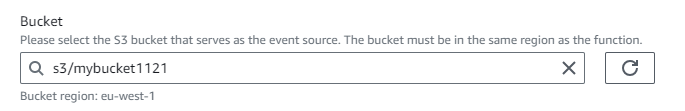
Step 3:-

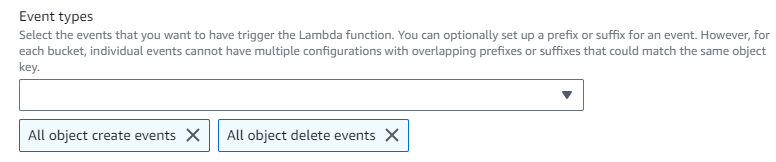
Now you have see your function then click on your function and then after you see the add trigger option click on that.

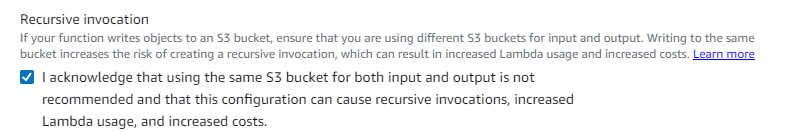
Step 4:-

After that select source as s3 and then select your bucket and then select event type when trigged the lambda function then this event logs generate and after that select I acknowledge and click on save.





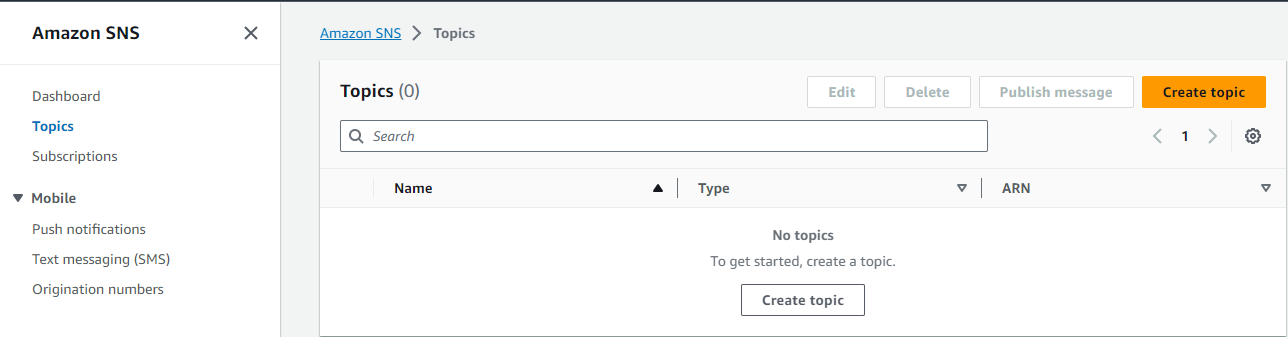






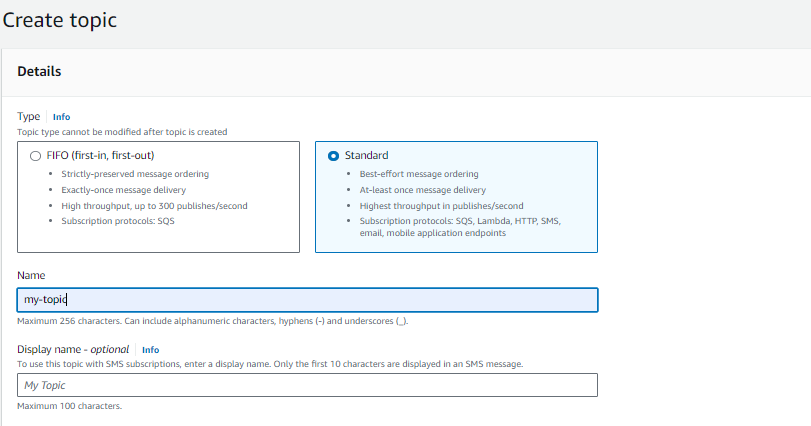
Step 5:-

Now add destination where to triggered function was occurred or messaged. So click on add destination so have select sns topic so first of all go to sns service. And click on create topic.



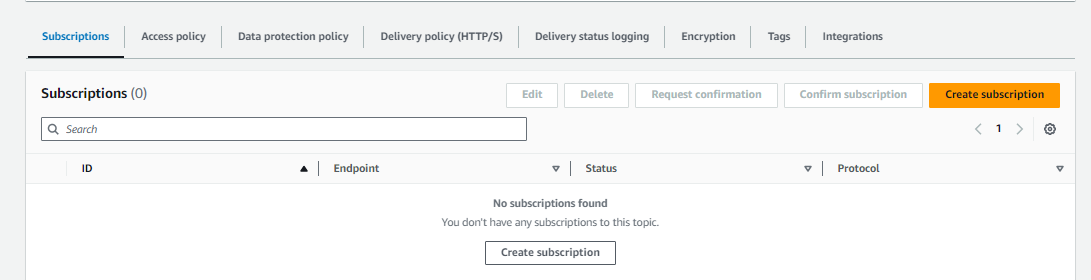
Step 5:-

Now next you have to gave name to topic and select type as standard and then click create topic.



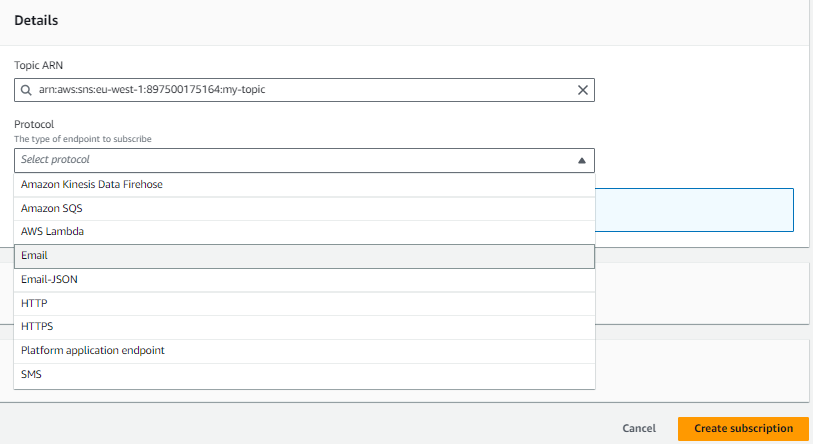
Step 6:-

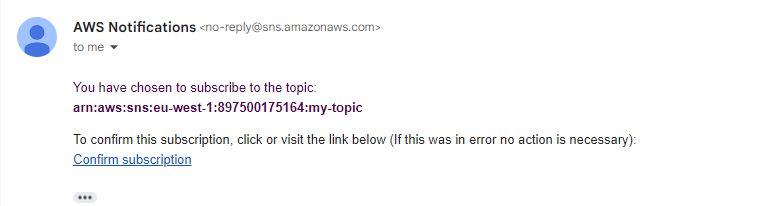
After that click on your topic name and in that you have see the option subscription click on that and after that click on create subscription.

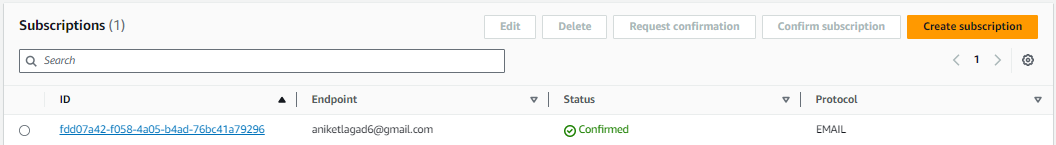


Step 7:-

Then by default selected your arn of topic leave as it is and then click on protocol and in that select type where to your massage want to send I am selected email and then gave email name and click on create subscription. After that go to your gmail account to conform it and then your sns topic was generated.

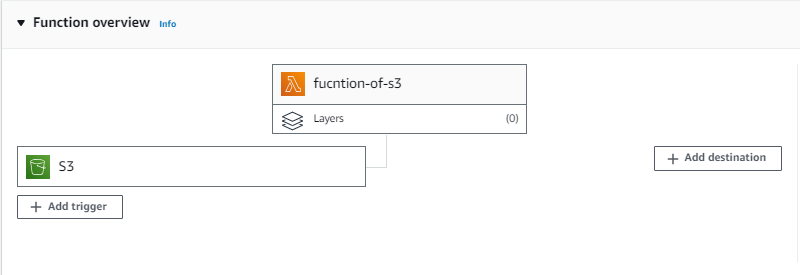






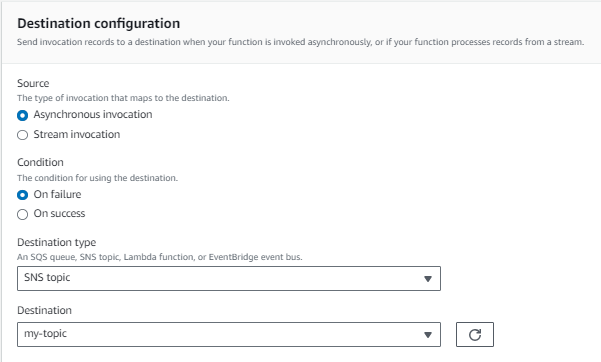
Step 8:-

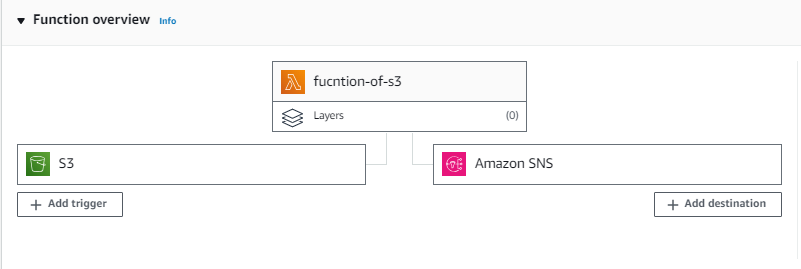
Now go to lambda service and click on your function and then now click on add destination option.



Step 9:-

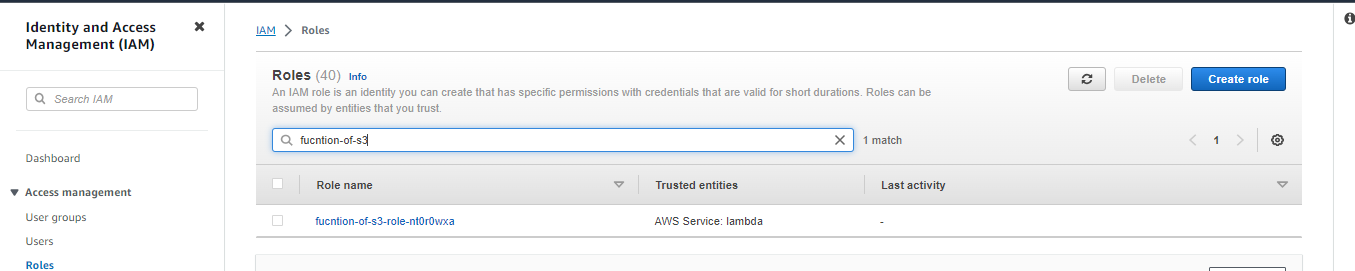
Now leave as its only select destination type as sns topic. And then destination select we created now onlt my-topic so select it. And see following image your destination was selected.

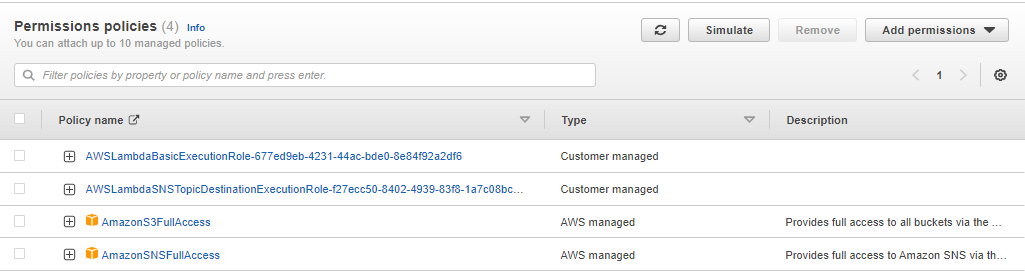




Step 10:-

Now we have send logs and massages whenever object added and deleted then lambda function was trigged and send logs to cloudwatch and email. But we have to gave permission to lambda function of s3 and sns. Go to I am service and then click on roles. And then paste function name in search box and then click on role name was searched. Add permission of s3 full access and full access on sns.





Step 11:-

Now get code paste it and then deploy the code. Following code to deploy

import json

import boto3

sns\_client = boto3.client('sns')

sns\_topic\_arn = ‘NAME\_YOUR\_SNS\_ARN’

def lambda\_handler(event, context):

# Log the event details

print("Received S3 event:", json.dumps(event))

# Check if the event has 'Records' key and it's a list

if 'Records' in event and isinstance(event['Records'], list):

for record in event['Records']:

event\_name = record['eventName']

# Check if it's an object creation event

if 'ObjectCreated' in event\_name:

message = f"New object '{record['s3']['object']['key']}' was uploaded to S3 bucket."

# Check if it's an object deletion event

elif 'ObjectRemoved' in event\_name:

message = f"Object '{record['s3']['object']['key']}' was deleted from S3 bucket."

else:

message = f"Unknown event type: {event\_name}"

# Publish the appropriate message to the SNS topic

sns\_client.publish(

TopicArn=sns\_topic\_arn,

Message=message

)

else:

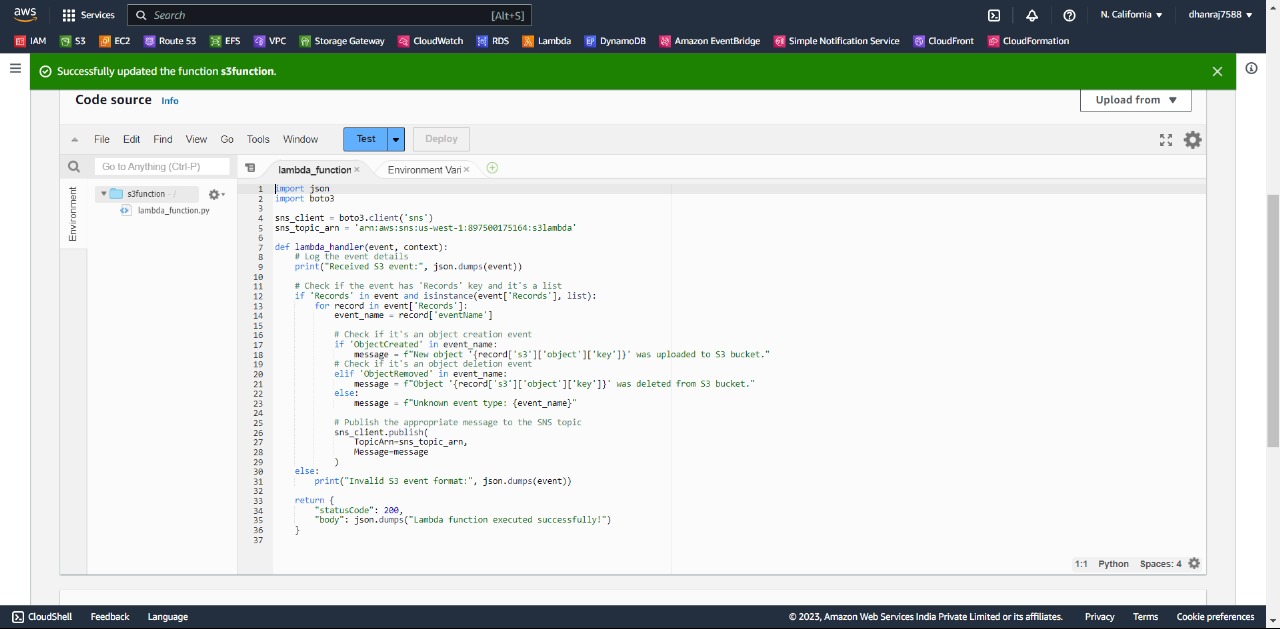
print("Invalid S3 event format:", json.dumps(event))

return {

"statusCode": 200,

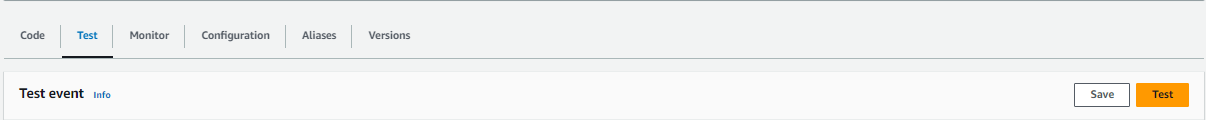
"body": json.dumps("Lambda function executed successfully!")

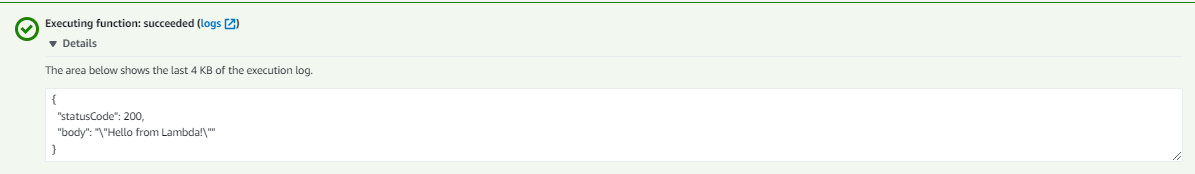
    }

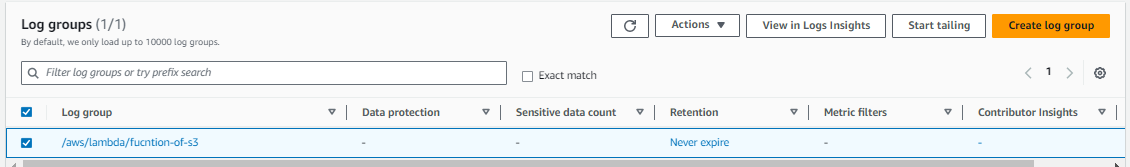


Step 12:-

Now click on text option and then you see the test option click on that and your log was send to cloudwatch.







\*\*\*THE END\*\*\*