Siddhant Sathe **D15A/51**

EXPERIMENT 12

Aim: To create a Lambda function which will log “An Image has been added” once you add an object to a specific bucket in S3

Theory:

AWS Lambda and S3 Integration: AWS Lambda allows you to execute code in response to various events, including those triggered by Amazon S3. When an object is added to an S3 bucket, it can trigger a Lambda function to execute, allowing for event-driven processing without managing servers.

Workflow:

1. Create an S3 Bucket:

○ First, create an S3 bucket that will store the objects. This bucket will act as the trigger source for the Lambda function.

2. Create the Lambda Function:

○ Set up a new Lambda function using AWS Lambda’s console. You can choose a runtime environment like Python, Node.js, or Java.

○ Write code that logs a message like “An Image has been added” when triggered.

3. Set Up Permissions:

○ Ensure that the Lambda function has the necessary permissions to access S3. You can do this by attaching an IAM role with policies that allow reading from the bucket and writing logs to CloudWatch.

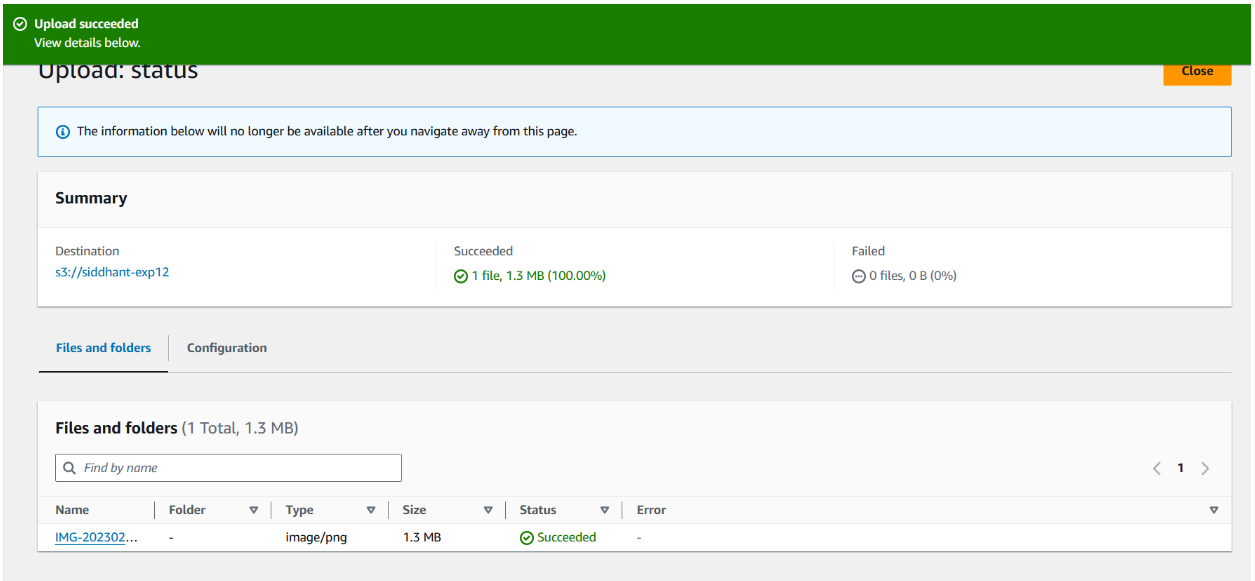
4. Configure S3 Trigger:

○ Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

5. Test the Setup:

○ Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs.

Step 1: Create a S3 bucket and deselect the block public access. Upload an image in the bucket



Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

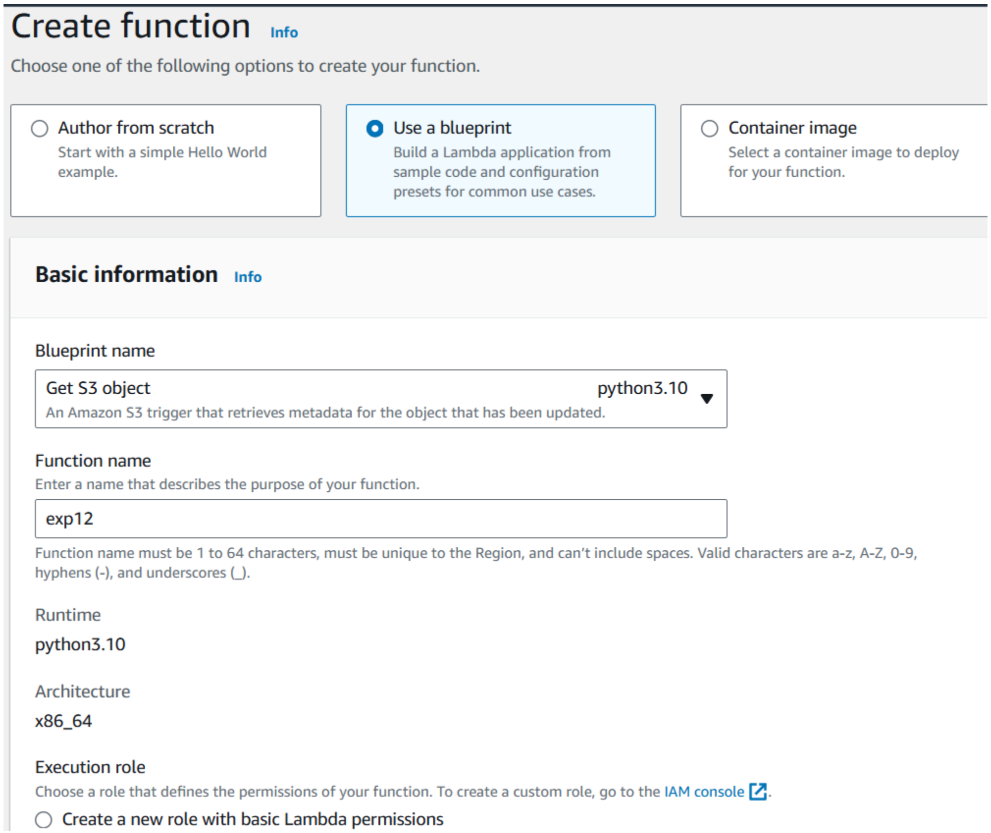
Summary

Destination	Succeeded	Failed
s3://siddhant-exp12	1 file, 1.3 MB (100.00%)	0 files, 0 B (0%)

Files and folders (1 Total, 1.3 MB)

Name	Folder	Type	Size	Status	Error
IMG-202302...	-	image/png	1.3 MB	Succeeded	-

Step 2: Open lambda console and click on create function button.Give a name to your Lambda function, Select the language to use to write your function.



Create function

Choose one of the following options to create your function.

○ Author from scratch
Start with a simple Hello World example.

● Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

Basic information

Blueprint name

Get S3 object python3.10
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Function name
Enter a name that describes the purpose of your function.

exp12

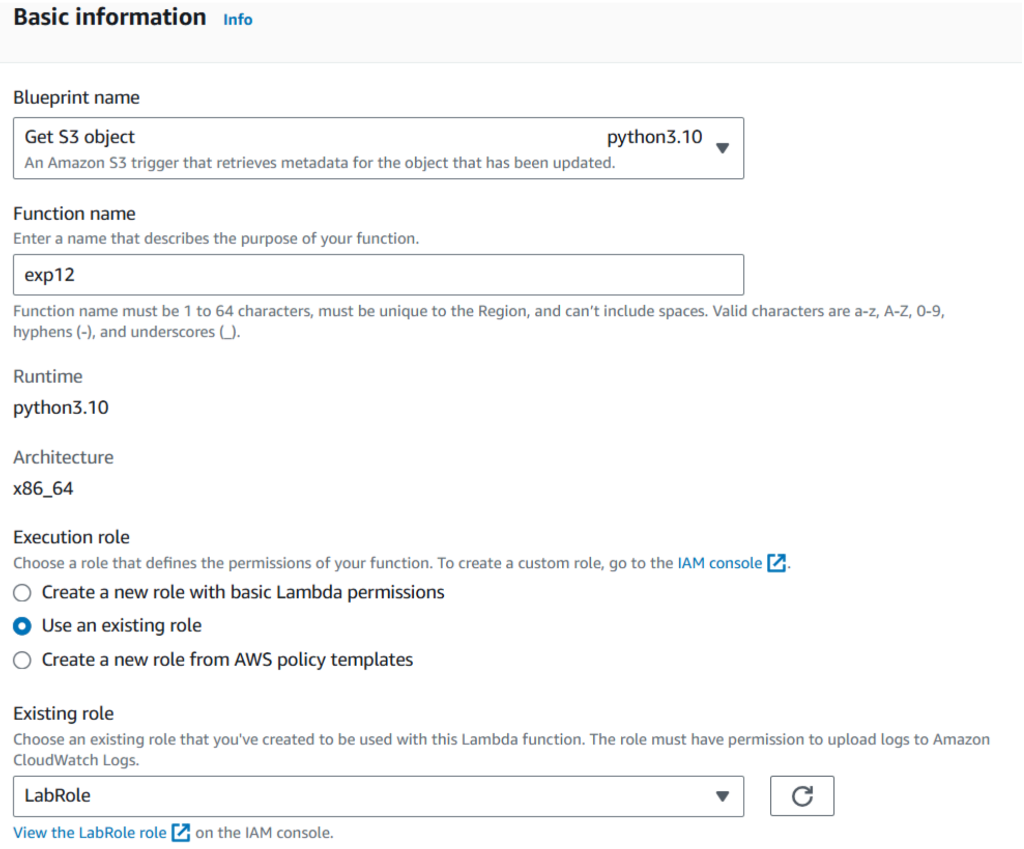
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime
python3.10

Architecture
x86_64

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

○ Create a new role with basic Lambda permissions



Basic information Info

Blueprint name

Get S3 object python3.10
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Function name

Enter a name that describes the purpose of your function.

exp12

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime

python3.10

Architecture

x86_64

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

○ Create a new role with basic Lambda permissions

● Use an existing role

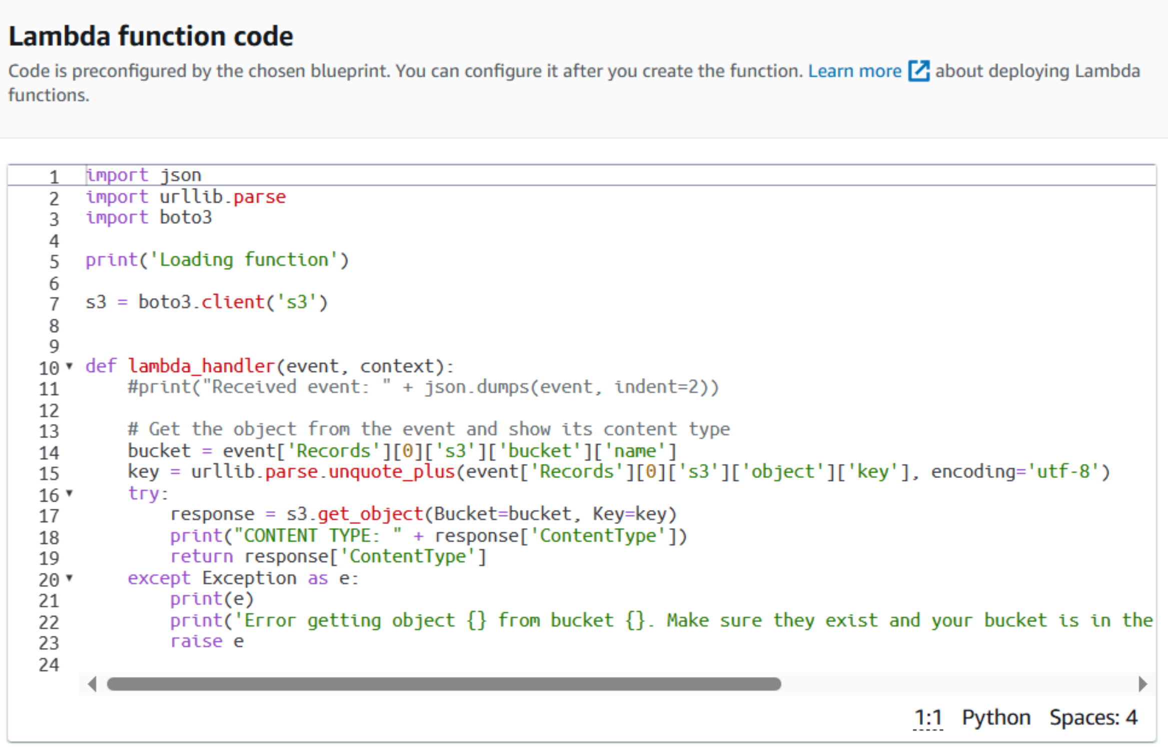
○ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LabRole

View the LabRole role on the IAM console.



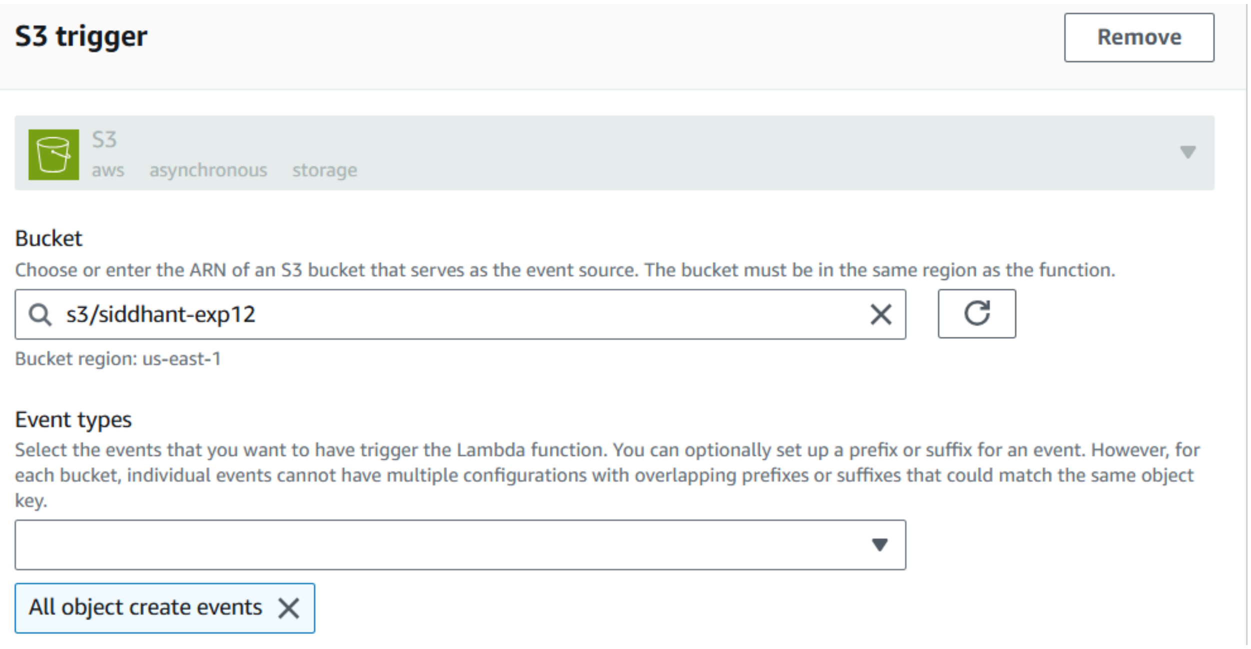
Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function. Learn more about deploying Lambda functions.

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

print('Loading function')

s3 = boto3.client('s3')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))

    # 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("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
        raise e

```

1:1 Python Spaces: 4

Step 3: Create a trigger for that lambda function. Select the S3 trigger. Select the S3 cucket in which you have uploaded the image



S3 trigger

S3
aws asynchronous storage

Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

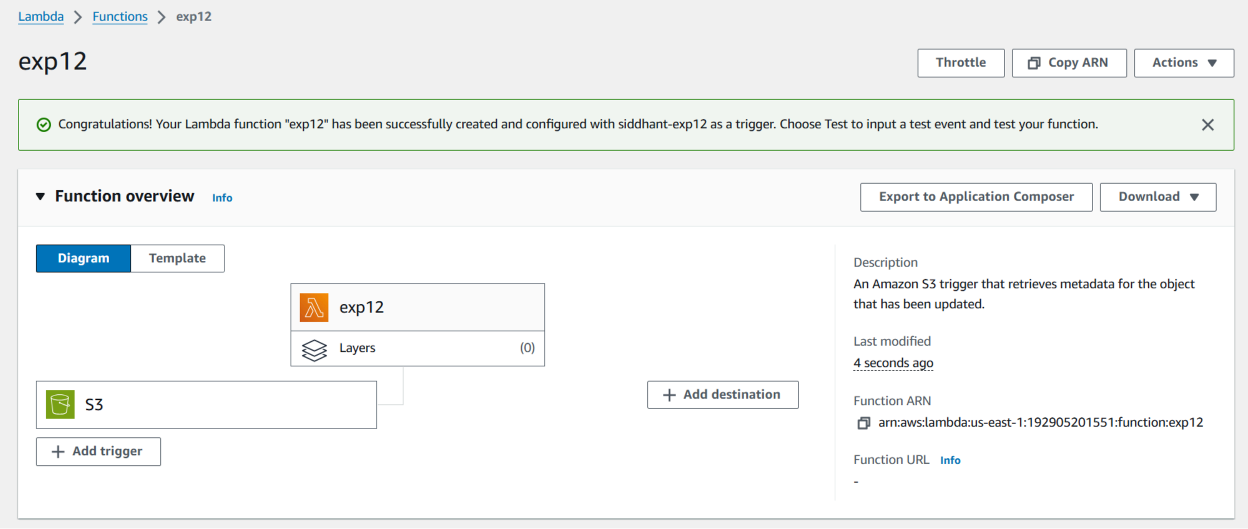
s3/siddhant-exp12

Bucket region: us-east-1

Event types

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events



Lambda > Functions > exp12

exp12

Congratulations! Your Lambda function "exp12" has been successfully created and configured with siddhant-exp12 as a trigger. Choose Test to input a test event and test your function.

Function overview Info

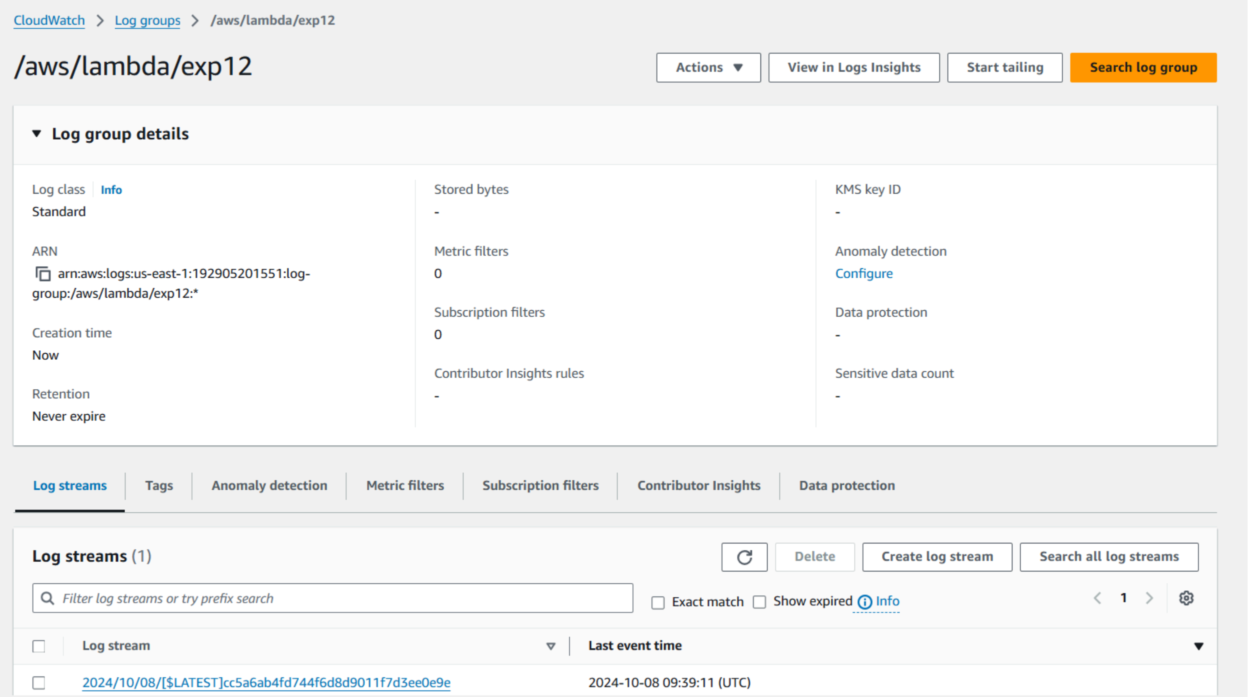
Description
An Amazon S3 trigger that retrieves metadata for the object that has been updated.

Last modified
4 seconds ago

Function ARN
arn:aws:lambda:us-east-1:192905201551:function:exp12

Function URL Info
-

Step 4 : Search for CloudWatch in services. After Selecting CloudWatch select log groups and then select your respecting lambda function.



/aws/lambda/exp12

▼ Log group details

Log class | Info
Standard

ARN
arn:aws:logs:us-east-1:192905201551:log-group:/aws/lambda/exp12:*

Creation time
Now

Retention
Never expire

Stored bytes
-

Metric filters
0

Subscription filters
0

Contributor Insights rules
-

KMS key ID
-

Anomaly detection
Configure

Data protection
-

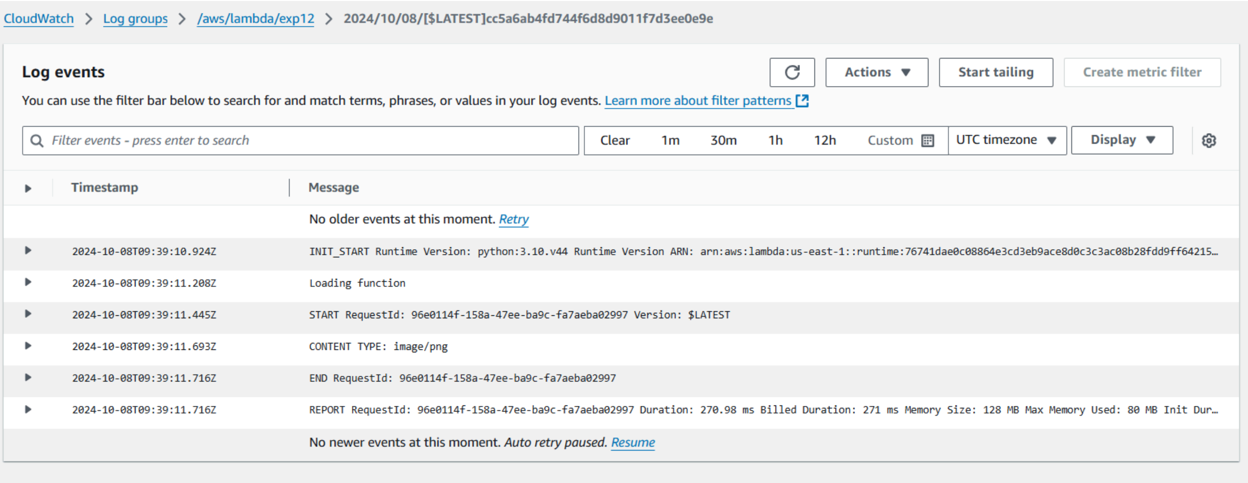
Sensitive data count
-

Log streams (1)

□ Exact match □ Show expired

□	Log stream	Last event time
□	2024/10/08/[$LATEST]cc5a6ab4fd744f6d8d9011f7d3ee0e9e	2024-10-08 09:39:11 (UTC)

Step 5: Now lets see the log on Cloud watch.To see it go to monitor section and then click on view cloudwatch logs


CloudWatch > Log groups > /aws/lambda/exp12 > 2024/10/08/[$LATEST]cc5a6ab4fd744f6d8d9011f7d3ee0e9e
Actions ▼
Start tailing
Create metric filter
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone ▼
Display ▼

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                  Message
                           No older events at this moment. Retry
2024-10-08T09:39:10.924Z   INIT_START Runtime Version: python:3.10.v44 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:76741dae0c08864e3cd3eb9ace8d0c3c3ac08b28fdd9ff64215…
2024-10-08T09:39:11.208Z   Loading function
2024-10-08T09:39:11.445Z   START RequestId: 96e0114f-158a-47ee-ba9c-fa7aeba02997 Version: $LATEST
2024-10-08T09:39:11.693Z   CONTENT TYPE: image/png
2024-10-08T09:39:11.716Z   END RequestId: 96e0114f-158a-47ee-ba9c-fa7aeba02997
2024-10-08T09:39:11.716Z   REPORT RequestId: 96e0114f-158a-47ee-ba9c-fa7aeba02997 Duration: 270.98 ms Billed Duration: 271 ms Memory Size: 128 MB Max Memory Used: 80 MB Init Dur…
                           No newer events at this moment. Auto retry paused. Resume
```