

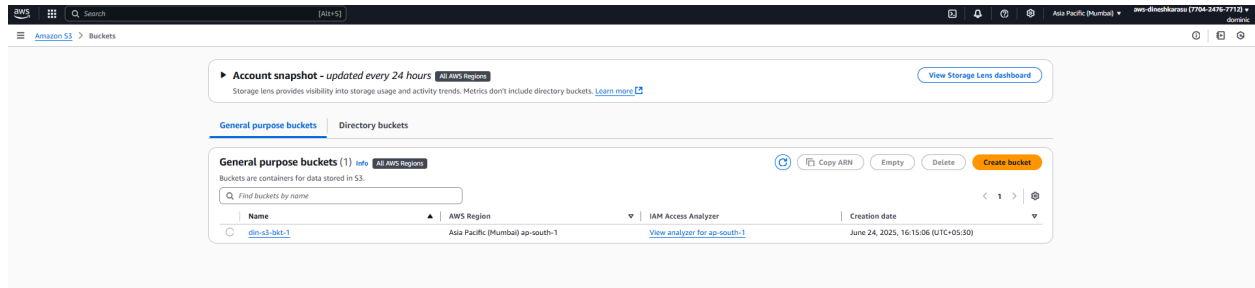
Use case

Event-Driven Architecture with Lambda, SNS/SQS

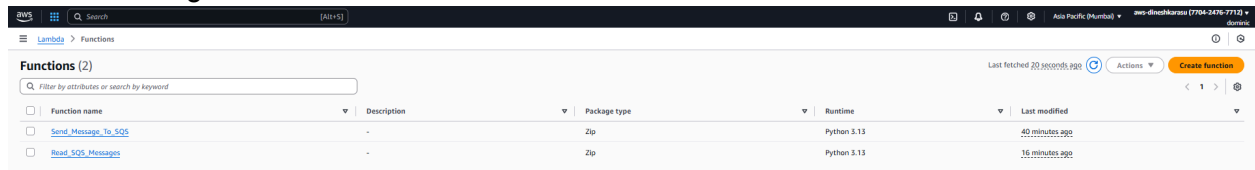
Use case Description

Upload to S3 triggers a Lambda function. Lambda sends a message to SQS. Another Lambda reads and processes SQS messages. Implement Dead Letter Queue (DLQ) for failure handling.

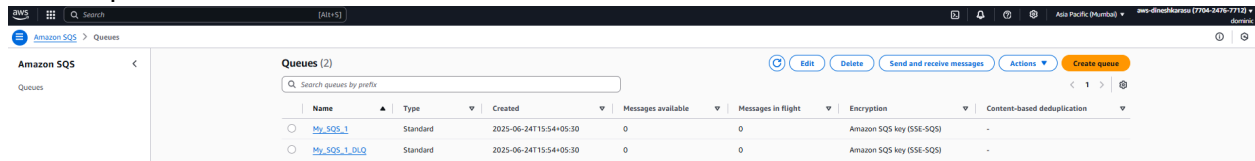
1. Create S3 Bucket



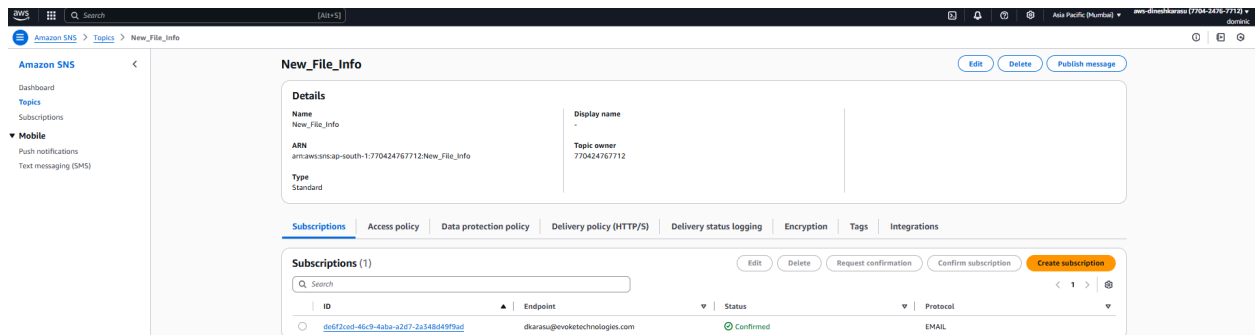
2. Create two Lambda functions one for Sending messages and the other for reading messages



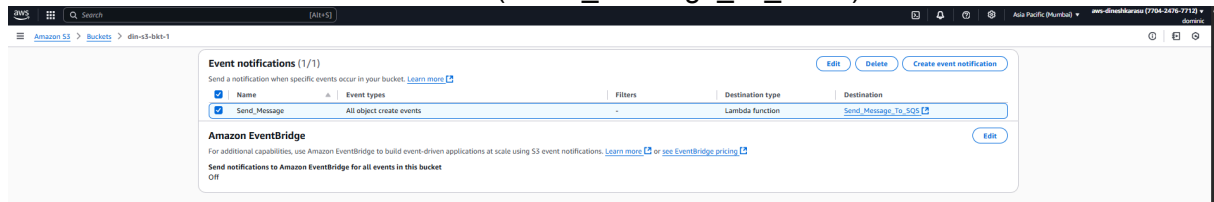
3. Create two SNS Queues one for storing the messages and other will be Dead letter queue for the first one



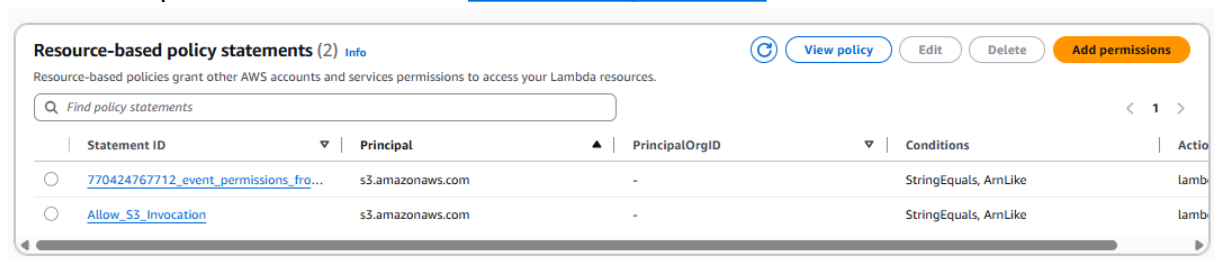
4. Create SNS topic to deliver the message over mail and confirm subscription with your email



5. Create Notification to Lambda (Send_Message_To_SQS) from the S3 Bucket



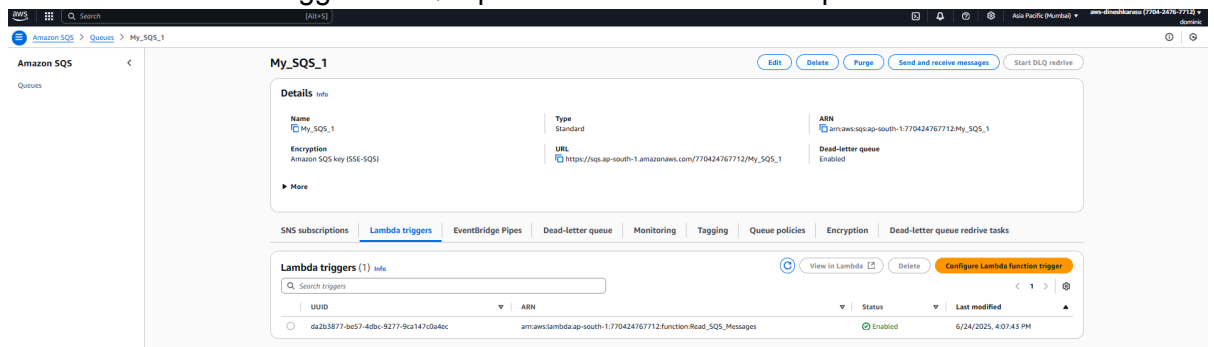
6. Add permissions to function [Send_Message_To_SQS](#) for S3 to invoke the function

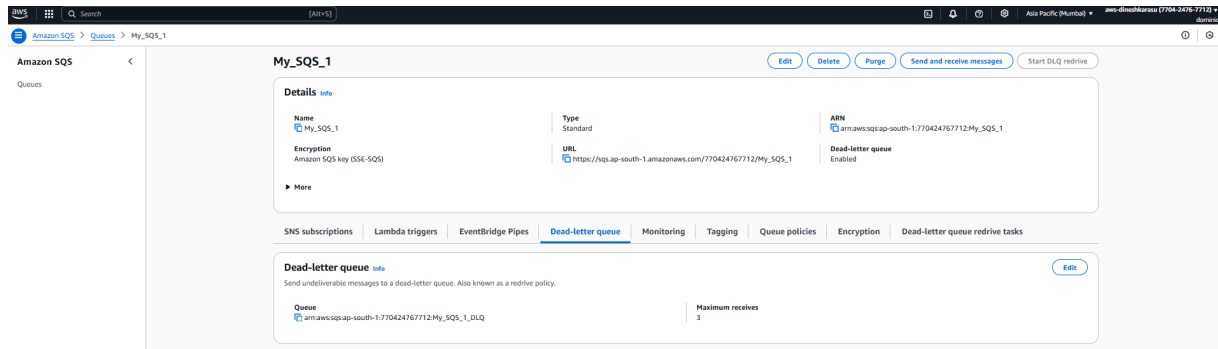


7. Write the following code to Lambda function

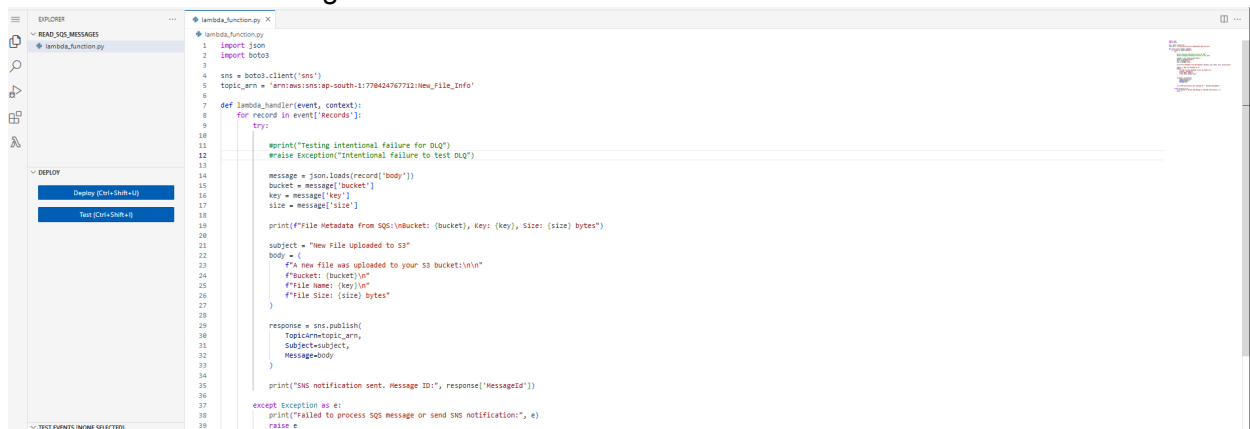


8. Add Lambda trigger to SQS queue and set dead-letter-queue

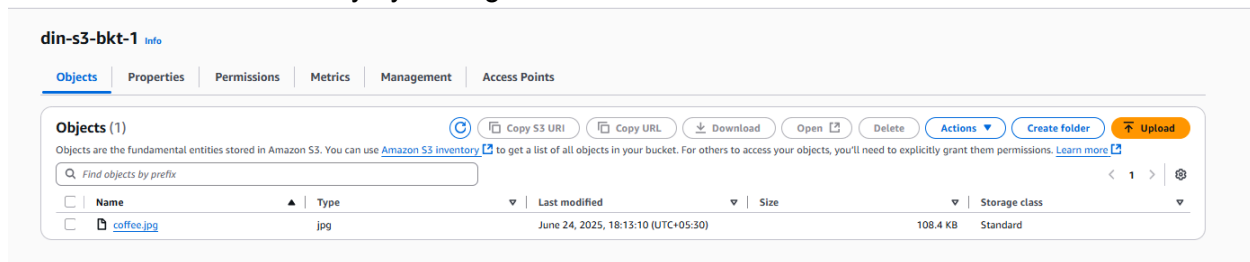




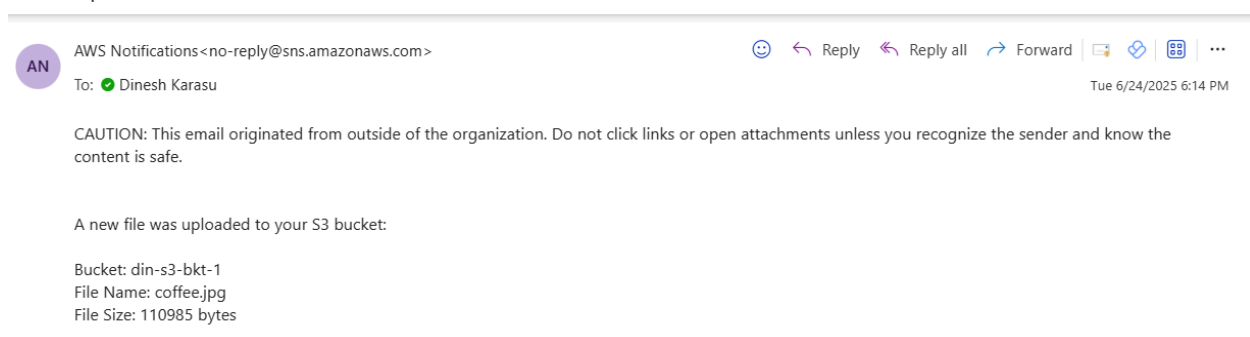
9. Write the following code to another lambda function



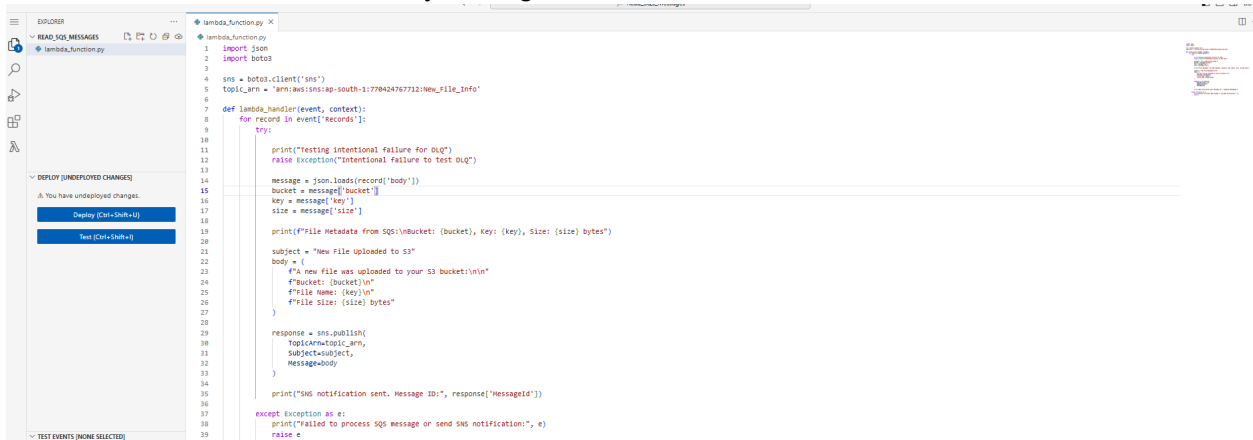
10. Test the functionality by adding a file to S3 bucket



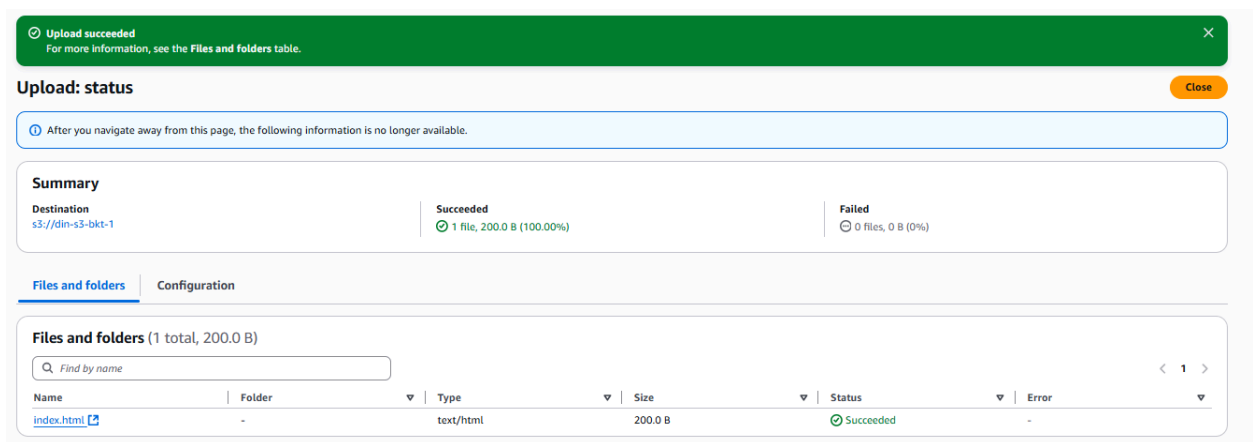
New File Uploaded to S3



11. Test Dead letter Queue by forcing the lambda to fail execution



```
1 import json
2 import boto3
3
4 sns = boto3.client('sns')
5 topic_arn = 'arn:aws:sns:ap-south-1:779424767712:New_File_Info'
6
7 def lambda_handler(event, context):
8     for record in event['records']:
9         try:
10             print("Testing intentional failure for DLQ")
11             raise Exception("Intentional failure to test DLQ")
12         except Exception as e:
13             message = json.loads(record['body'])
14             bucket = message['bucket']
15             key = message['key']
16             size = message['size']
17
18             print(f"File Metadata from S3:\nBucket: {bucket}, Key: {key}, Size: {size} bytes")
19
20             subject = "New File uploaded to S3"
21             body = {
22                 f"A new file was uploaded to your S3 bucket:\n\n"
23                 f"Bucket: {bucket}\n"
24                 f"File Name: {key}\n"
25                 f"File Size: {size} bytes"
26             }
27
28             response = sns.publish(
29                 TopicArn=topic_arn,
30                 Subject=subject,
31                 Message=body
32             )
33
34             print(f"SNS notification sent. Message ID: {response['MessageId']}")
35
36 except Exception as e:
37     print(f"Failed to process SQS message or send SNS notification:", e)
38     raise e
```



Upload succeeded
For more information, see the [Files and folders](#) table.

Upload: status Close

After you navigate away from this page, the following information is no longer available.

Summary

Destination s3://dlm-s3-bkt-1	Succeeded 1 file, 200.0 B (100.00%)	Failed 0 files, 0 B (0%)
---	---	------------------------------------

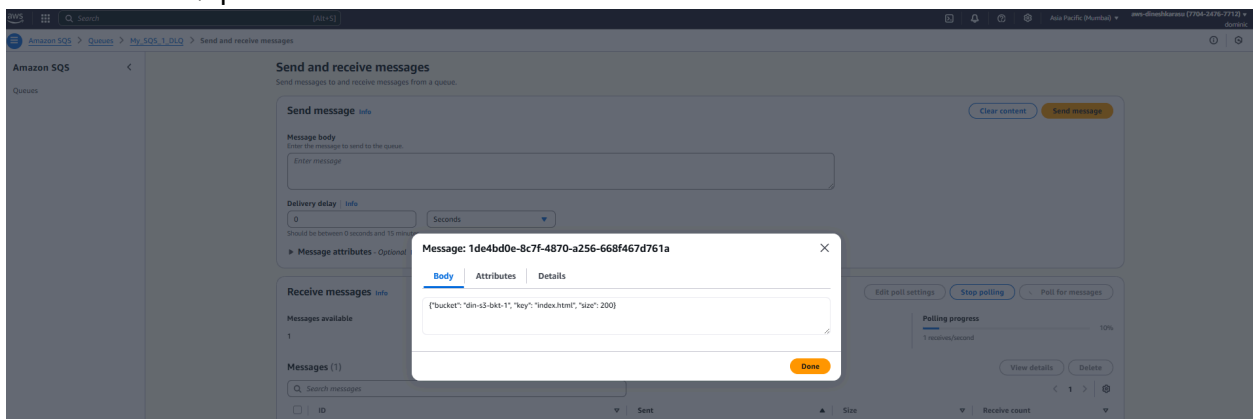
Files and folders | Configuration

Files and folders (1 total, 200.0 B)

Find by name

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	200.0 B	Succeeded	-

Check the DLQ queue if there's a new file



Amazon SQS > Queues > My SQS_1_DLQ > Send and receive messages

Send message

Message body
Enter the message to send to the queue.

Delivery delay
0 Seconds

Receive messages

Messages available: 1

Messages (1)

Message: 1de4bd0e-8c7f-4870-a256-668f467d761a

Body: {"bucket": "dlm-s3-bkt-1", "key": "index.html", "size": 200}