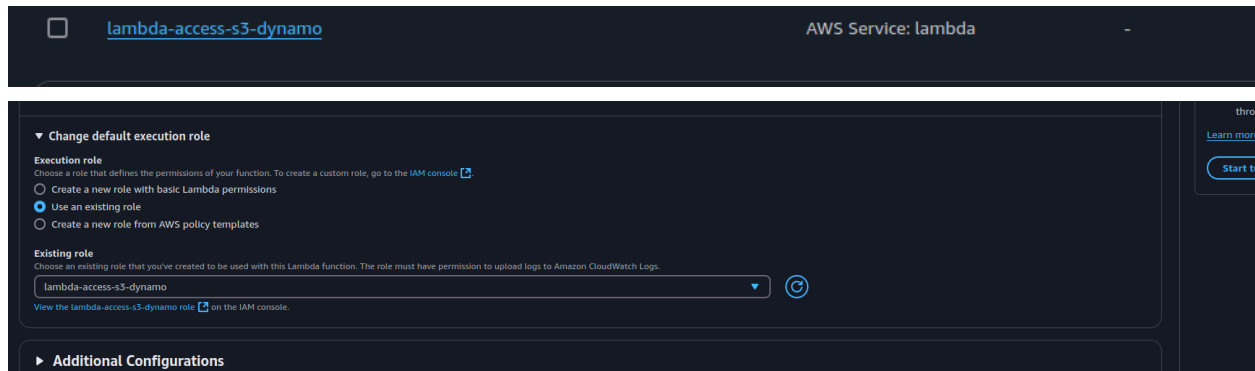
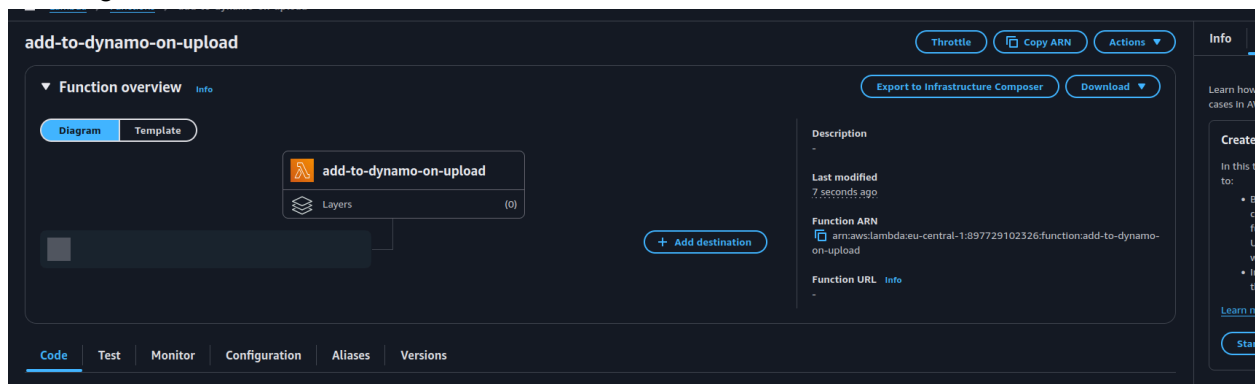


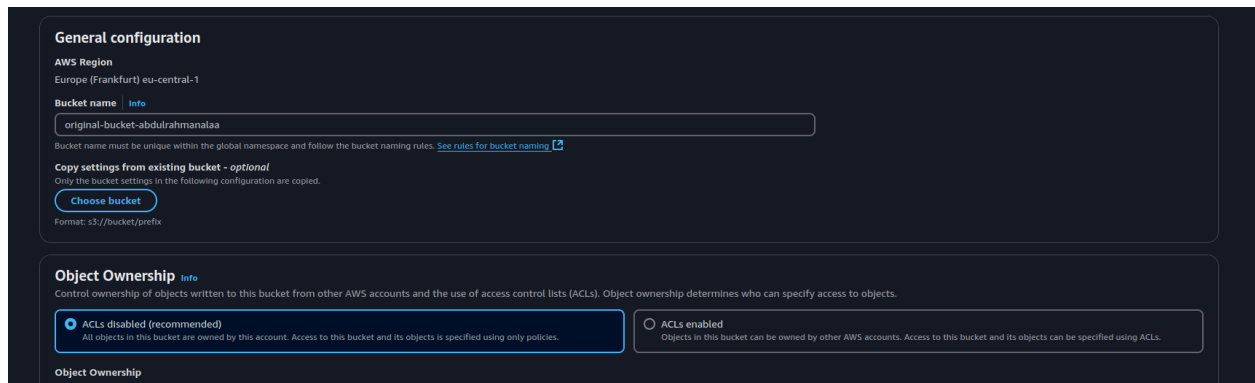
Q1: create a lambda trigger on s3 upload to add file to dynamodb
First create the proper lambda role with access to dynamodb and s3



Creating the lambda function



Creating the S3 bucket



Adding the lambda trigger

Trigger configuration

Info

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.

Q

s3/original-bucket-abdulrahmanalaa

X

🔄

Bucket region: eu-central-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 X

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any special characters must be URL encoded.

e.g. images/

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.

e.g. .jpg

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Info

Learn how to use S3 as an event source for Lambda functions.

Create a new S3 bucket

In this tutorial, you will create a new S3 bucket and configure it to trigger a Lambda function.

Learn more

Get started

add-to-dynamo-on-upload

Throttle

Copy ARN

Actions

🟢 The trigger original-bucket-abdulrahmanalaa was successfully added to function add-to-dynamo-on-upload. The function is now receiving events from the trigger.

X

Function overview

Info

Export to Infrastructure Composer

Download

Diagram

Template

add-to-dynamo-on-upload

Layers (0)

+ Add destination

Description

-

Last modified

2 minutes ago

Function ARN

arn:aws:lambda:eu-central-1:897729102326:function:add-to-dynamo-on-upload

Function URL

Info

Code

Test

Monitor

Configuration

Aliases

Versions

General configuration

Info

Edit

Triggers

Permissions

Destinations

Description

-

Timeout

0 min - 3 sec

Memory

128 MB

SnapshotStart

Info

Ephemeral storage

512 MB

Creating a dynamodb table with the file-name which is the as its partition key

Your feedback is an important part of helping us provide a better customer experience. Take this short survey to let us know how we're doing.

Creating the s3-filenames table. It will be available for use shortly.

X

Tables (1)

Info

Find tables

Any tag key

Any tag value

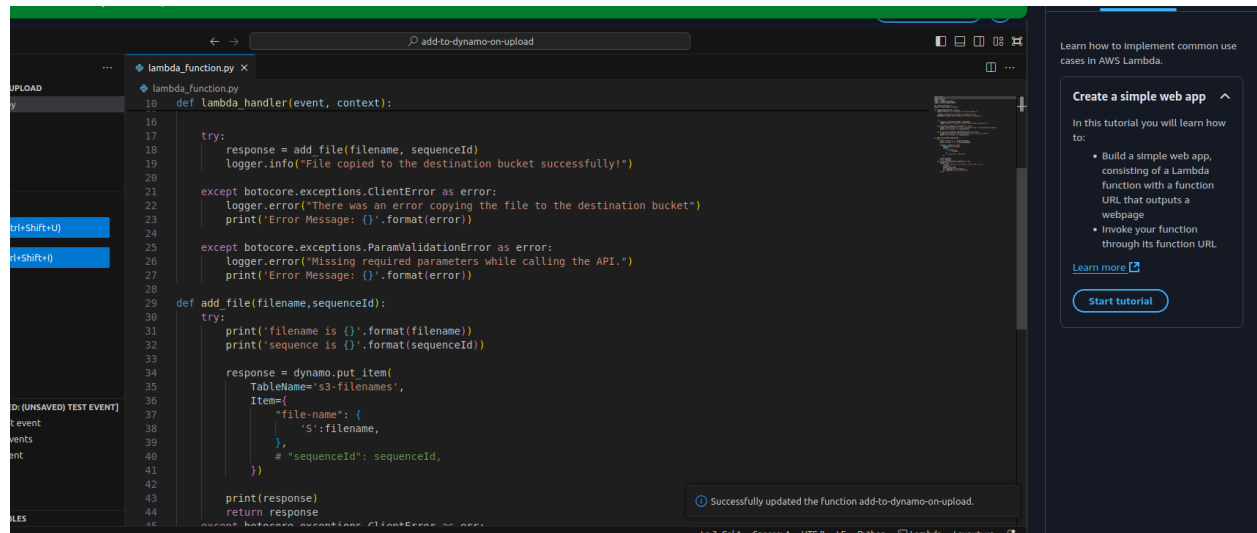
Actions

Delete

Create table

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mode	Write capacity mode	Total size	Table class
s3-filenames	Creating	file-name (S)	-	0	0	Off	☆	On-demand	On-demand	0 bytes	Standard

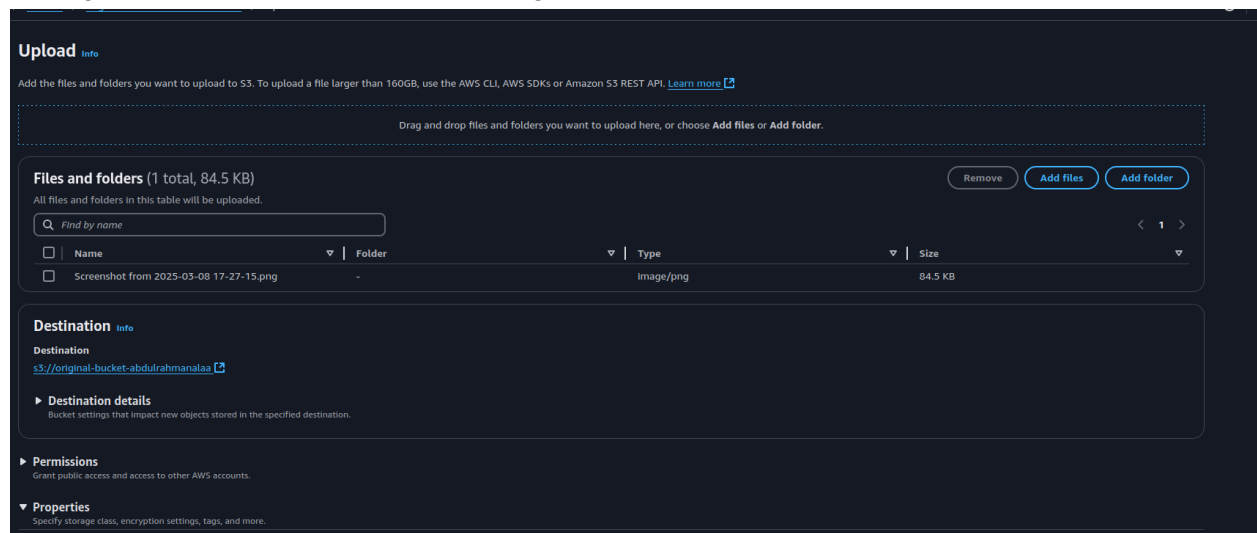
Writing the function's code and now to test we must upload an item to the original-bucket-abdulrahmanalaa



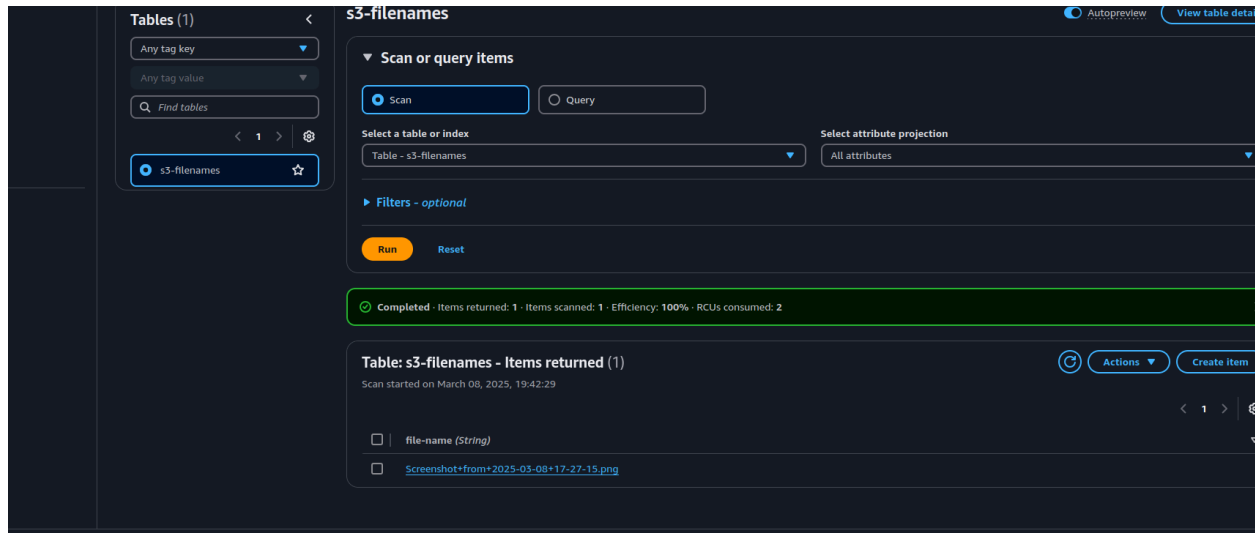
The screenshot shows the AWS Lambda console with the code for a function named 'add-to-dynamo-on-upload'. The code is written in Python and defines a lambda handler and an add_file function. The handler calls add_file with the filename and sequenceId from the event. The add_file function prints the filename and sequenceId, then uses boto3 to put an item into a DynamoDB table named 's3-filenames'. The item contains the filename and sequenceId. A status message at the bottom indicates 'Successfully updated the function add-to-dynamo-on-upload.'

```
10 def lambda_handler(event, context):
11
12     try:
13         response = add_file(filename, sequenceId)
14         logger.info("File copied to the destination bucket successfully!")
15     except botocore.exceptions.ClientError as error:
16         logger.error("There was an error copying the file to the destination bucket")
17         print("Error Message: {}".format(error))
18
19     except botocore.exceptions.ParamValidationError as error:
20         logger.error("Missing required parameters while calling the API.")
21         print("Error Message: {}".format(error))
22
23 def add_file(filename, sequenceId):
24     try:
25         print('filename is {}'.format(filename))
26         print('sequence is {}'.format(sequenceId))
27
28         response = dynamo.put_item(
29             TableName='s3-filenames',
30             Item={
31                 "file-name": {
32                     'S': filename,
33                 },
34                 # "sequenceId": sequenceId,
35             })
36
37         print(response)
38         return response
39     except botocore.exceptions.ClientError as error:
```

Adding the screenshot from 17-27-15.png



We can see that the filename is added into the dynamodb



And to review the code for adding the dynamodb entry

It can be found at

<https://github.com/abdurahmanalaa123/ITI-sessions/tree/master/AWS/Lab4>