

ADVANCED DEVOPS EXP 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

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.

General configuration

AWS Region
US East (N. Virginia) us-east-1


Bucket type [Info](#)

☒ **General purpose**
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

☐ **Directory**
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name [Info](#)


shravanibucket2

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#) 

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.



Choose bucket

Format: s3://bucket/prefix

 **Successfully created bucket "shravanibucket2"**
To upload files and folders, or to configure additional bucket settings, choose [View details](#).

General purpose buckets | Directory buckets

General purpose buckets (4) [Info](#) [All AWS Regions](#)

  Copy ARN

Buckets are containers for data stored in S3.

	Name ▲	AWS Region ▼	IAM Access Analyzer
<input type="radio"/>	codepipeline-us-east-1-474858304070	US East (N. Virginia) us-east-1	View analyzer for us-east-1
<input type="radio"/>	elasticbeanstalk-eu-west-2-361769589277	Europe (London) eu-west-2	View analyzer for eu-west-2
<input type="radio"/>	elasticbeanstalk-us-east-1-361769589277	US East (N. Virginia) us-east-1	View analyzer for us-east-1
<input type="radio"/>	shravanibucket2	US East (N. Virginia) us-east-1	View analyzer for us-east-1

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

Lambda > Functions > Create function

Create function [Info](#)

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

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

imageloader

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 [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 20.x


✓ Successfully created the function **imageloader**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".


imageloader

Throttle Copy ARN Actions

▼ **Function overview** [Info](#) Export to Application Composer Download

Diagram Template


 **imageloader**

 Layers (0)

+ Add trigger + Add destination

Description
-

Last modified
2 minutes ago

Function ARN
 arn:aws:lambda:us-east-1:361769589277:function:imageloader

Function URL [Info](#)
-

```

import json
import logging

# Set up logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)

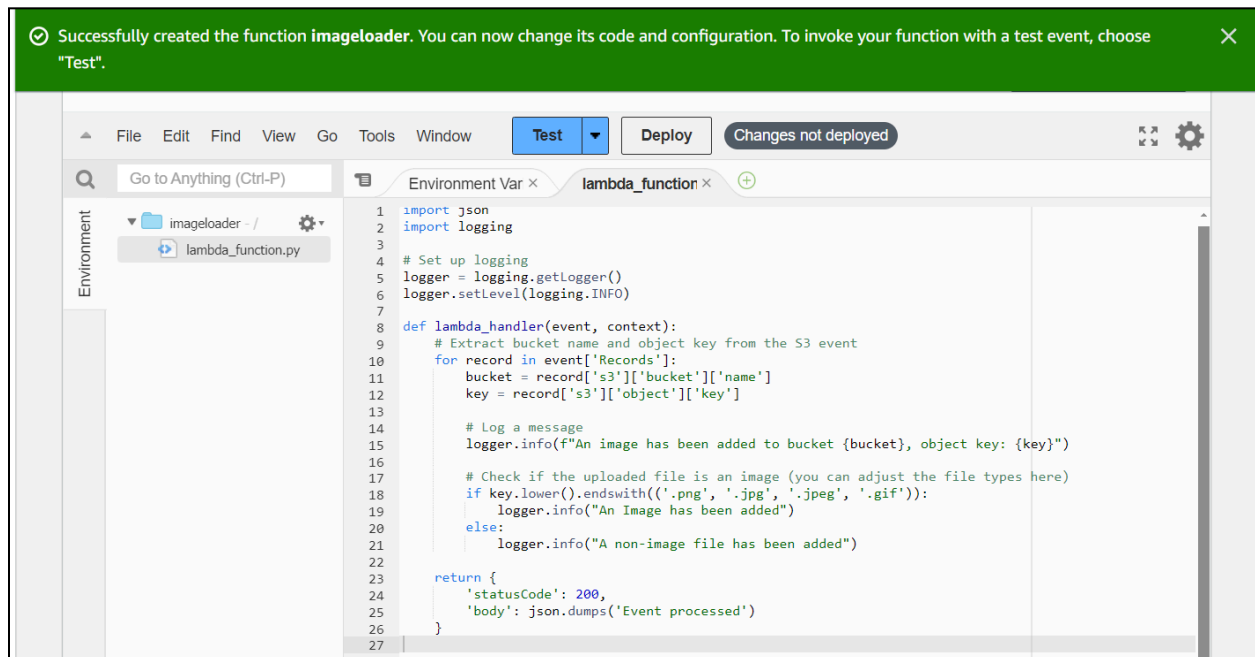
def lambda_handler(event, context):
    # Extract bucket name and object key from the S3 event
    for record in event['Records']:
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        # Log a message
        logger.info(f"An image has been added to bucket {bucket}, object key: {key}")

        # Check if the uploaded file is an image (you can adjust the file types here)
        if key.lower().endswith(('png', '.jpg', '.jpeg', '.gif')):
            logger.info("An Image has been added")
        else:
            logger.info("A non-image file has been added")

    return {
        'statusCode': 200,
        'body': json.dumps('Event processed')
    }

```




3. 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).

[Lambda](#) > Add triggers

Add 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.

× ↺

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 ×

imageloader

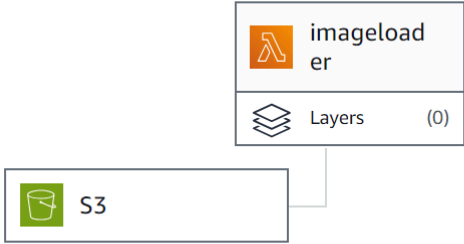
Throttle Copy ARN Actions ▼

✓ The trigger shravanibucket2 was successfully added to function imageloader. The function is now receiving events from the trigger. ×

▼ **Function overview** [Info](#)

Export to Application Composer Download ▼

Diagram Template



Diagram

The diagram shows an S3 bucket icon connected to the 'imageloader' function box. The function box has a 'Layers (0)' section below it. A '+ Add trigger' button is at the bottom left, and a '+ Add destination' button is to the right of the S3 bucket icon.

imageloader

Layers (0)

+ Add destination

+ Add trigger

Description

-

Last modified

9 minutes ago

Function ARN

arn:aws:lambda:us-east-1:361769589277:function:imageloader

Function URL [Info](#)

The screenshot shows the AWS Lambda console with the 'Configuration' tab selected. On the left sidebar, 'Triggers' is highlighted. The main area displays 'Triggers (1)' with an 'Info' link and buttons for 'Fix errors', 'Edit', 'Delete', and 'Add trigger'. A search bar labeled 'Find triggers' is present. Below it, a table lists the triggers. One trigger is shown: 'S3: shravanibucket2' with the ARN 'arn:aws:s3:::shravanibucket2' and a 'Details' link.

Trigger
<input type="checkbox"/> S3: shravanibucket2 arn:aws:s3:::shravanibucket2 Details

4. Upload an object (e.g., an image) to the S3 bucket to test the trigger

The screenshot shows the Amazon S3 console for the bucket 'shravanibucket2'. The 'Objects' tab is selected. The main area displays 'Objects (0)' with an 'Info' link and buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', and 'Delete'. There are also buttons for 'Actions', 'Create folder', and 'Upload'. A text box explains that objects are fundamental entities in Amazon S3 and provides links to 'Amazon S3 inventory' and 'Learn more'. A search bar labeled 'Find objects by prefix' is present. Below it, a table header is shown with columns: 'Name', 'Type', 'Last modified', and 'Size'. The table currently shows 'No objects'.

Name	Type	Last modified	Size
No objects			

[Amazon S3](#) > [Buckets](#) > [shravanibucket2](#) > Upload

Upload

[Info](#)

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 91.9 KB)

Remove

Add files

Add folder

All files and folders in this table will be uploaded.

Find by name

< 1 >

<input type="checkbox"/>	Name	Folder	Type	Size
<input type="checkbox"/>	upload.png	-	image/png	91.9 KB

Destination

[Info](#)

Destination

[s3://shravanibucket2](#)

► Destination details

Bucket settings that impact new objects stored in the specified destination.

► Permissions

Grant public access and access to other AWS accounts.

► Properties

Specify storage class, encryption settings, tags, and more.

Cancel

Upload

Upload succeeded
View details below.

Summary

Destination s3://shravanibucket2	Succeeded 1 file, 91.9 KB (100.00%)	Failed 0 files, 0 B (0%)
-------------------------------------	--	-----------------------------

Files and folders

Configuration

Files and folders (1 Total, 91.9 KB)

< 1 >

Name	Folder	Type	Size	Status	Error
upload.png	-	image/png	91.9 KB	Succeeded	-

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

CloudWatch

Favorites and recents

Dashboards

Alarms

▼ Logs

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Metrics

X-Ray traces

Events

Application Signals [New](#)

CloudWatch > Log groups > /aws/lambda/imageloader

/aws/lambda/imageloader

Actions View in Logs Insights Start tailing Search log group

▼ Log group details

Log class Info Standard	Stored bytes -	KMS key ID -
ARN arn:aws:logs:us-east-1:361769589277:log-group:/aws/lambda/imageloader:*	Metric filters 0	Anomaly detection Configure
Creation time 1 minute ago	Subscription filters 0	Data protection -
Retention Never expire	Contributor Insights rules -	Sensitive data count -

Log events

Actions

Start tailing

Create metric filter

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](#)

Q

Filter events - press enter to search

Clear

1m

30m

1h

12h

Custom

UTC timezone

Display

▶

Timestamp

▶

Message

No older events at this moment. [Retry](#)

▶

2024-10-08T05:00:37.474Z

INIT_START Runtime Version: python:3.11.v44 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:b1c790bce6ec3c3a14a715f557a25d2daffc580e2fa1439a9ee32ac12f1dd582

▶

2024-10-08T05:00:37.561Z

START RequestId: d230f778-3a97-493e-963f-29beec0922e Version: \$LATEST

▶

2024-10-08T05:00:37.562Z

[INFO] 2024-10-08T05:00:37.562Z d230f778-3a97-493e-963f-29beec0922e An image has been added to bucket shravanibucket2, object key: upload.png

▶

2024-10-08T05:00:37.562Z

[INFO] 2024-10-08T05:00:37.562Z d230f778-3a97-493e-963f-29beec0922e An Image has been added

▶

2024-10-08T05:00:37.563Z

END RequestId: d230f778-3a97-493e-963f-29beec0922e

▶

2024-10-08T05:00:37.563Z

REPORT RequestId: d230f778-3a97-493e-963f-29beec0922e Duration: 1.98 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 33 MB Init Duration: 86.42 ms

No newer events at this moment. [Auto retry paused. Resume](#)

An image has been added to bucket shravanibucket2, object key: upload.png

An Image has been added

Conclusion:

Integrating AWS Lambda with S3 allows for real-time, automated processing of events such as file uploads. In this example, a Lambda function is configured to log a message whenever an image is added to a specific S3 bucket. This setup demonstrates the power and flexibility of serverless computing by automating tasks without requiring manual intervention or server management. By leveraging AWS Lambda, developers can efficiently handle event-driven workflows, reduce operational overhead, and quickly deploy scalable solutions that respond to specific actions within cloud environments