

Name: Abhishek Chaurasiya

Class: D15B / 09

ADV DEVOPS 12

Here are the steps to create a Lambda function that logs "An Image has been added" once an object is added to a specific S3 bucket in AWS Learner Lab:

1. Create an S3 Bucket

- Go to the AWS Management Console.
- Navigate to the S3 service.
- Click on "Create bucket."
- Enter a unique bucket name and choose a region.
- Configure other settings as needed and click "Create bucket."

Amazon S3 > Buckets > Create bucket

Create bucket [Info](#)

Buckets are containers for data stored in S3.

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](#)

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

Successfully created bucket "abhibucket09"

View details

To upload files and folders, or to configure additional bucket settings, choose **View details**.

Account snapshot - updated every 24 hours

All AWS Regions

View Storage Lens dashboard

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

General purpose buckets

Directory buckets

General purpose buckets (2)

Info

All AWS Regions

Refresh

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

< 1 > ⚙

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	fdp-mistakenly-vast-mayfly-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 20, 2024, 23:46:52 (UTC+05:30)
<input type="radio"/>	abhibucket09	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 29, 2024, 14:24:22 (UTC+05:30)

2. Create a Lambda Function

- Go to the AWS Management Console.
- Navigate to the Lambda service.
- Click on "Create function."
- Choose "Author from scratch."
- Enter a name for your function, e.g., **S3ImageLogger**.
- Select a runtime (e.g., Python 3.x or Node.js).
- Click "Create function."

Basic information

Function name

Enter a name that describes the purpose of your function.

niS3ImageLogger

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime

Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.12

Refresh

Architecture

Info

Choose the instruction set architecture you want for your function code.

☒ x86_64
 ☐ arm64

Permissions

Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this

▼ Change default execution role

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 CloudWatch Logs.

LabRole

[View the LabRole role](#) on the IAM console.

Successfully created the function **niS3ImageLogger**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

[Lambda](#) > [Functions](#) > niS3ImageLogger

niS3ImageLogger

Throttle

Copy ARN

Actions ▼

► Function overview Info

Export to Application Composer

Download ▼

Code

Test

Monitor

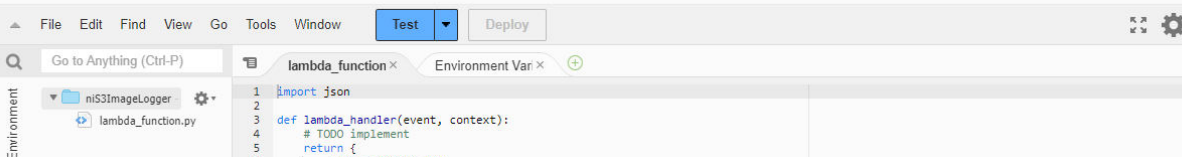
Configuration

Aliases

Versions

Code source Info

Upload from ▼



3. Write the Lambda Function Code

- In the Lambda function console, scroll down to the code editor.

Replace the default code with the following code snippet (assuming you're using Python):

python

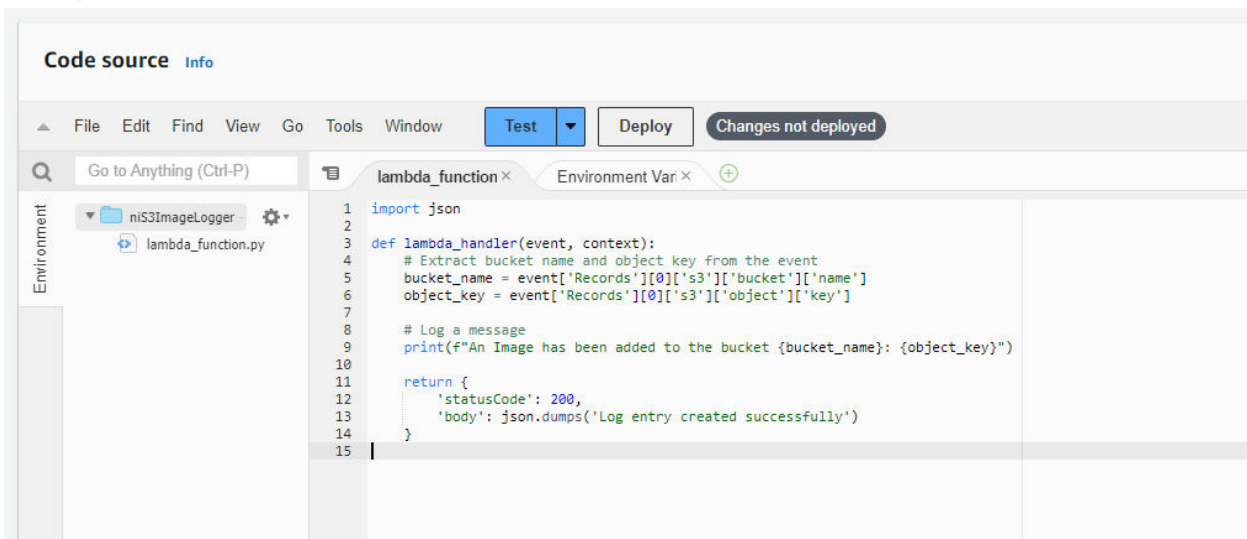
Copy code

```
import json
```

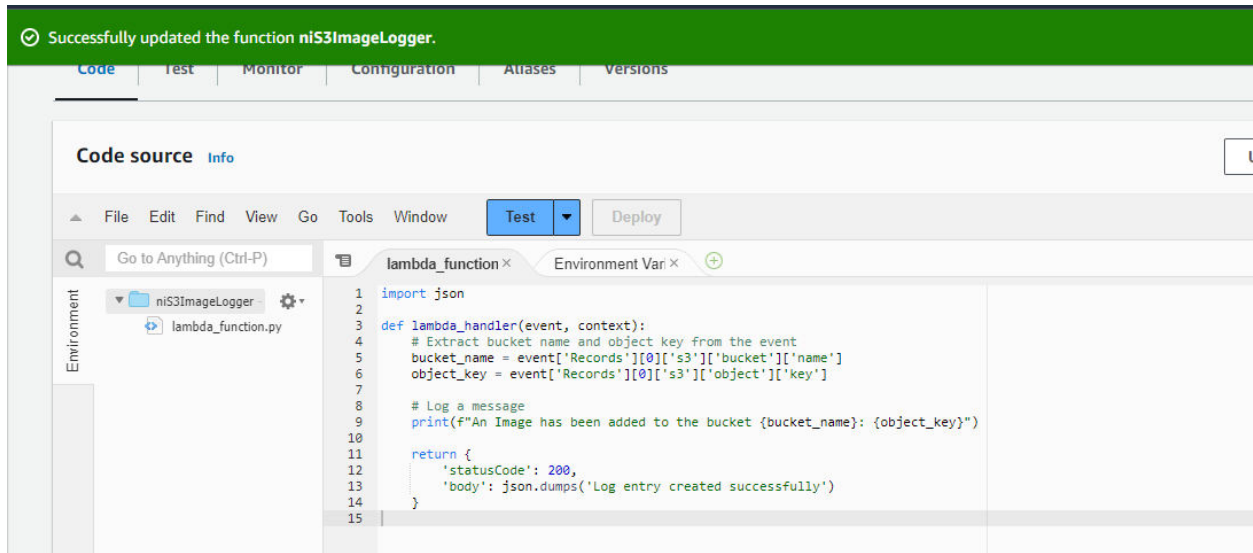
```
def lambda_handler(event, context):
    # Extract bucket name and object key from the event
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = event['Records'][0]['s3']['object']['key']

    # Log a message
    print(f"An Image has been added to the bucket {bucket_name}: {object_key}")

    return {
        'statusCode': 200,
        'body': json.dumps('Log entry created successfully')
    }
```



- Click "Deploy" to save your changes.



4. Set Up S3 Trigger for the Lambda Function

- Scroll down to the "Function overview" section in the Lambda console.
- Click on "Add trigger."
- Select "S3" from the list of triggers.
- Choose the S3 bucket you created earlier.
- In the "Event type" dropdown, select "All object create events."
- Optionally, specify a prefix or suffix to filter the events (e.g., for images only, you can use suffix `.jpg`, `.png`).
- Click "Add."

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 ✕

✓ The trigger abhibucket09 was successfully added to function niS3ImageLogger. The function is now receiving events from the trigger. ✕

Function overview [Info](#)

Export to Application Composer

Download ▼

Code Test Monitor **Configuration** Aliases Versions

General configuration

Triggers

Permissions

Destinations

Function URL

Environment variables

Triggers (1) [Info](#)



Fix errors

Edit

Delete

Add trigger

< 1 >



Trigger



S3: [abhibucket09](#)

arn:aws:s3:::abhibucket09

► Details

5. Grant Permissions to Lambda

- Navigate to the "Permissions" tab of your Lambda function.
- Ensure the Lambda function's execution role has the necessary permissions to access the S3 bucket.
- If needed, attach the [AmazonS3ReadOnlyAccess](#) policy or create a custom policy with the necessary permissions.

Tags
VPC
RDS databases
Monitoring and operations tools
Concurrency and recursion detection
Asynchronous invocation
Code signing
File systems
State machines

User: arn:aws:sts::659038782739:assumed-role/voclab2/user-3462843-2022-mdm:peonekar@ves.ac.in is not authorized to perform: iam:GetPolicy on resource: policy arn:aws:iam::659038782739:policy/c127008a318464917224321t1w659038782739-VocLabPolicy2-XkffSrLjyAhm with an explicit deny in an identity-based policy

Resource-based policy statements (1) [Info](#)

Resource-based policies grant other AWS accounts and services permissions to access your Lambda resources.

Statement ID
Principal
PrincipalOrgID
Conditions
Action

☐
[lambda-8b6dc529-6...](#)
s3.amazonaws.com
-
StringEquals, ArnLike
lambda:InvokeFunci...

Auditing and compliance

AWS CloudTrail can log this function's invocations for operational and risk auditing, governance, and compliance. [Get started](#) on the CloudTrail console.

6. Test the Setup

- Upload an image file to your S3 bucket.

Upload succeeded
View details below.

Summary

Destination
s3://nidhi47bucket


Succeeded
☒ 1 file, 4.8 KB (100.00%)

Failed
☐ 0 files, 0 B (0%)

Files and folders
Configuration

Files and folders (1 Total, 4.8 KB)

Name
Folder
Type
Size
Status
Error


-
image/jpeg
4.8 KB
☒ Succeeded
-

- Go to the "Monitoring" tab in your Lambda function to check the logs.
- Alternatively, use CloudWatch Logs to view the output and confirm that the message "An Image has been added" has been logged.

CloudWatch

Favorites and recents

Dashboards

Alarms 0 0 0

Logs

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Metrics

X-Ray traces

Events

Application Signals New

CloudWatch > Log groups > /aws/lambda/nis3ImageLogger > 2024/08/29/[\$LATEST]b75e0037c7df43d7b5e54db7ec697ec4

Log events

Filter events - press enter to search

1m 1h UTC timezone Display

Timestamp

Message

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

2024-08-29T09:10:23.091Z

INIT_START Runtime Version: python:3.12.v30 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:ac...

2024-08-29T09:10:23.207Z

START RequestId: 3241d961-2593-41f5-a6f9-b7257d5b3966 Version: \$LATEST

2024-08-29T09:10:23.208Z

An Image has been added to the bucket nidhi47bucket: img1.jpg

2024-08-29T09:10:23.210Z

END RequestId: 3241d961-2593-41f5-a6f9-b7257d5b3966

2024-08-29T09:10:23.210Z

REPORT RequestId: 3241d961-2593-41f5-a6f9-b7257d5b3966 Duration: 2.18 ms Billed Duration: 3 ms Memory...

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

This setup should ensure that each time an image is uploaded to the specified S3 bucket, the Lambda function will log the appropriate message.

