

Advance Devops-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.

Outcomes:

Step 1:create a s3 bucket

aws Services Search [Alt+S] N. Virginia voclabs/user3404128=THADANI_NIKITA_MANISH @ 2798-2026-7065

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](#)
exp12

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*

Successfully created bucket "exp-no-12" [View details](#)

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

Amazon S3 > Buckets

► **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](#)

Step 2:

Create a function

And select Python 3 as Runtime

aws Services Search [Alt+S] N. Virginia voclabs/user3404128=THADANI_NIKITA_MANISH @ 2798-2026-7065

Amazon S3 > Buckets

► **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](#)

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.

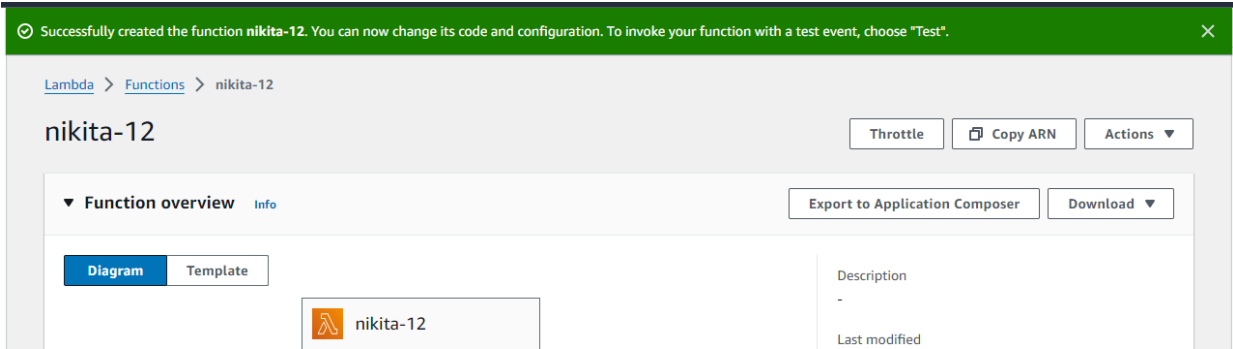
☐ **Container image**
Select a container image to deploy for your function.

Basic information

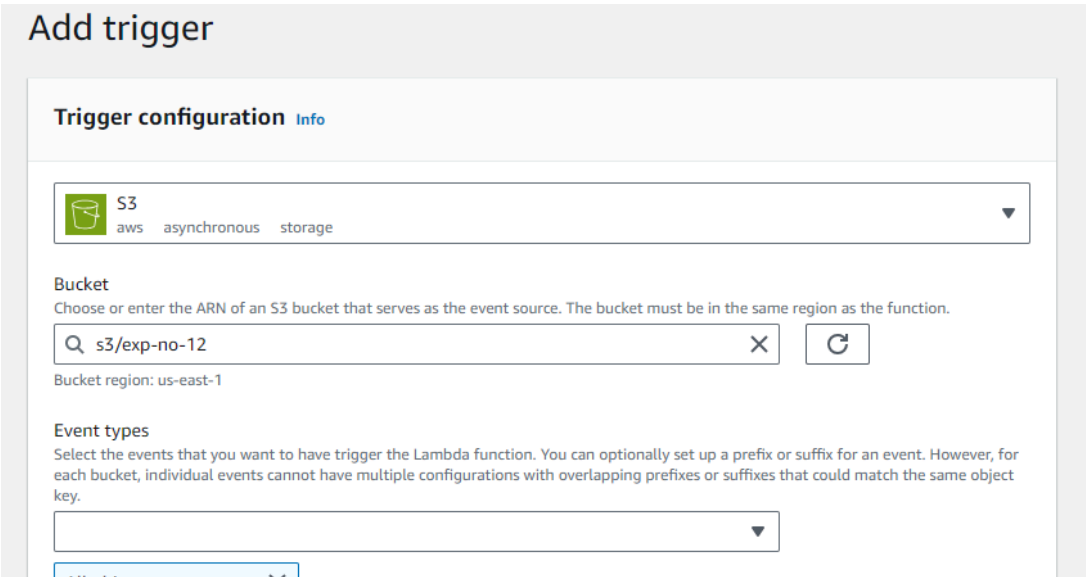
Function name
Enter a name that describes the purpose of your function.
nikita-12

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 (_).

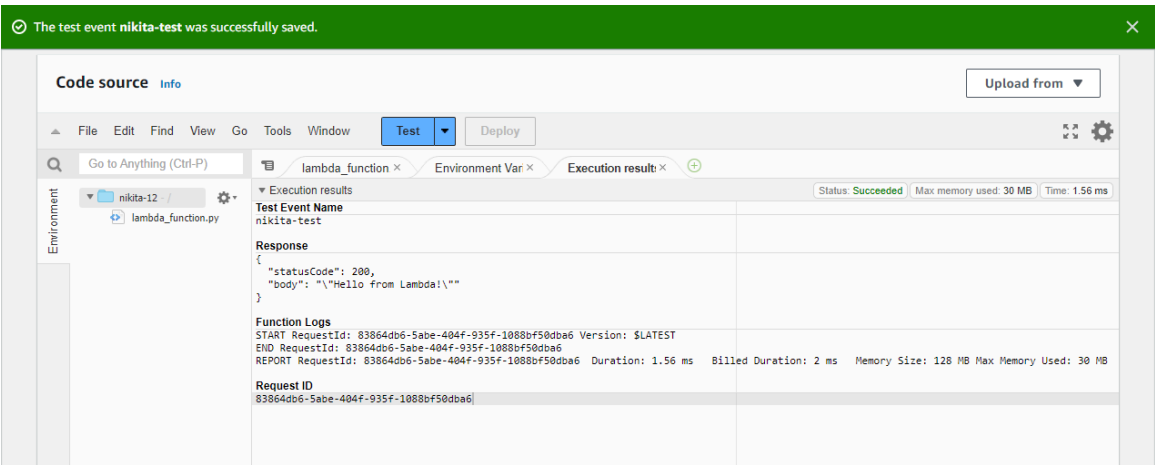
Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.9



Step 3:
Add a trigger



step 4:
Test it



Upload succeeded
View details below.

Summary

Destination
s3://nikita-vesit

Succeeded
1 file, 1.6 MB (100.00%)

Failed
0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (1 Total, 1.6 MB)

Find by name

< 1 >

Name	Folder	Type	Size	Status	Error
demo.jpg	-	image/jpeg	1.6 MB	Succeeded	-

Step 5:
Upload a image
It should be in jpg format

CloudWatch > Log groups > /aws/lambda/nikita-12 > 2024/10/11/[\$LATEST]8148efef2713489a960d35b23bad25b1

Log events

Actions Start tailing Create metric filter

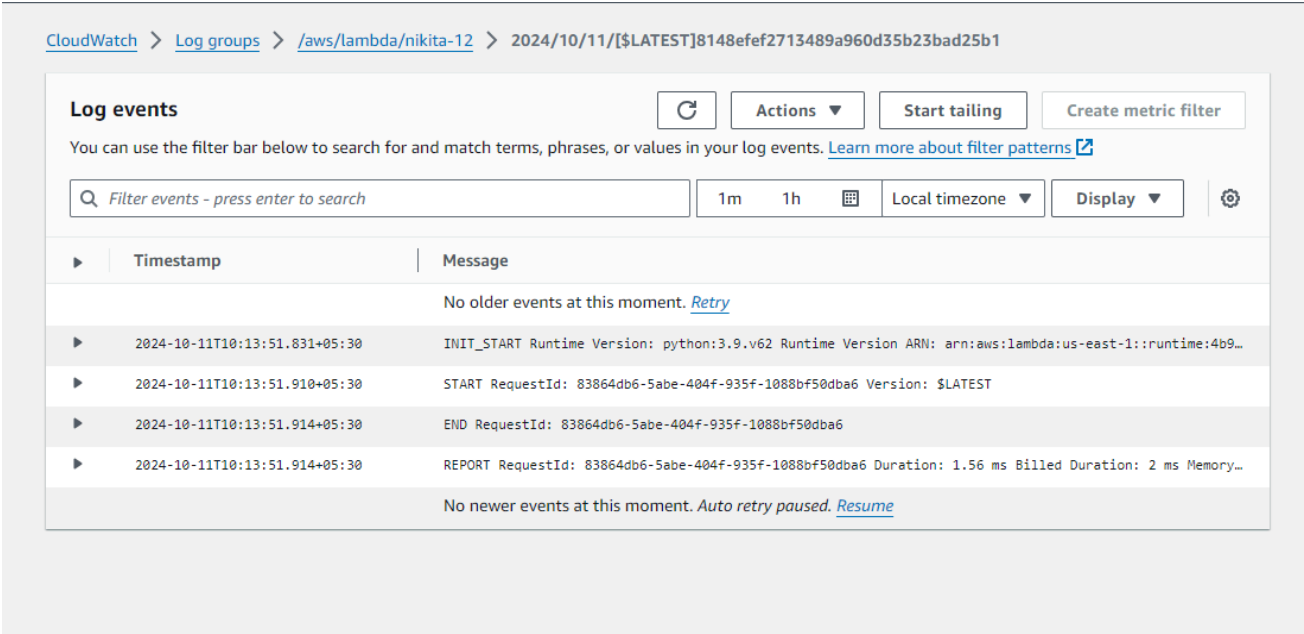
You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events - press enter to search

1m 1h UTC timezone Display

Timestamp	Message
No older events at this moment. Retry	
2024-10-11T04:43:51.831Z	INIT_START Runtime Version: python:3.9.v62 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:4b9...
2024-10-11T04:43:51.910Z	START RequestId: 83864db6-5abe-404f-935f-1088bf50dba6 Version: \$LATEST
2024-10-11T04:43:51.914Z	END RequestId: 83864db6-5abe-404f-935f-1088bf50dba6
2024-10-11T04:43:51.914Z	REPORT RequestId: 83864db6-5abe-404f-935f-1088bf50dba6 Duration: 1.56 ms Billed Duration: 2 ms Memory...
No newer events at this moment. Auto retry paused. Resume	

Step 6:
You can see the image you uploaded with the time



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.