

EXPERIMENT NO. 12

NAME : PRANAV POL

CLASS : D15A

ROLL NO. : 42

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:

Creating a system to log activities when an image is added to an S3 bucket involves integrating Amazon S3 with AWS Lambda.

Amazon S3 (Simple Storage Service) is a service offered by AWS that provides object storage through a web service interface. It's designed to store and retrieve any amount of data from anywhere. You can use S3 to store images, videos, backups, data logs, and more.

AWS Lambda is a serverless compute service that allows you to run code in response to events without provisioning or managing servers. You write your code and set up a trigger, and Lambda takes care of the rest. This means that when a specified event occurs, such as an object being added to an S3 bucket, the Lambda function is automatically invoked. By setting up a Lambda function to trigger on new uploads to a specific S3 bucket, you can automate logging activities. This function will capture the event, process it, and log the message "An Image has been added." It ensures that every new upload is tracked efficiently and that you have a record of these actions.

This kind of setup is highly scalable, reliable, and costeffective, leveraging AWS's robust infrastructure. It's particularly useful for applications that require automated monitoring and logging of uploads for auditing or notification purposes.

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.

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](#)

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

Format: s3://bucket/prefix

Successfully created bucket "lambdabucketpranav"
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](#)

[General purpose buckets](#) [Directory buckets](#)

General purpose buckets (2) [Info](#) [All AWS Regions](#)

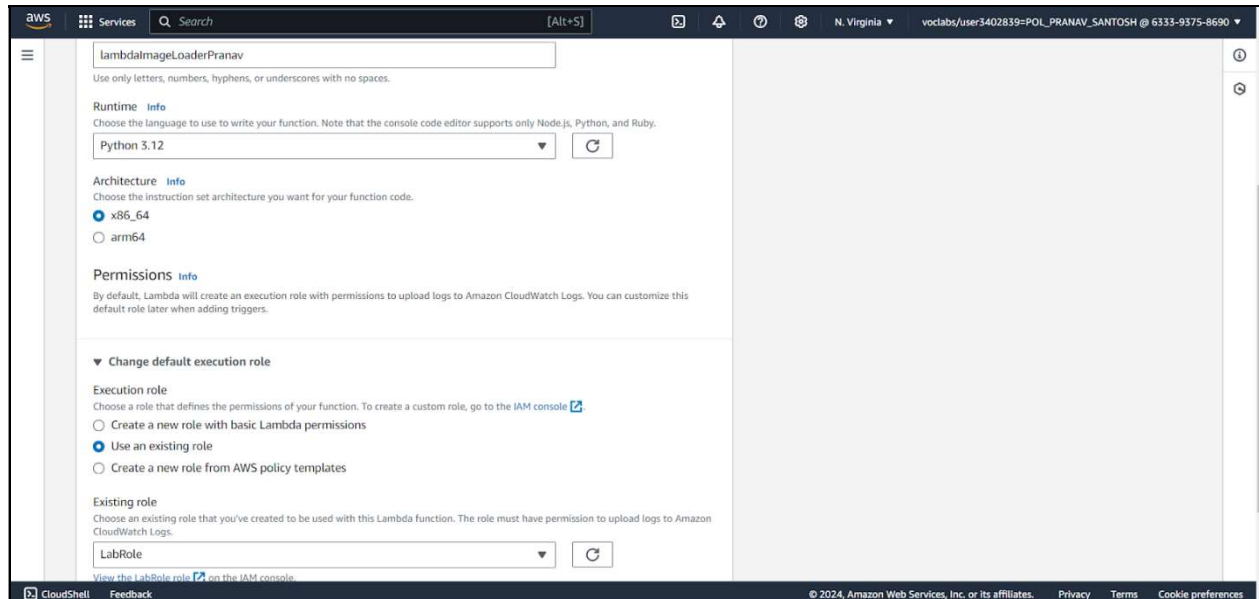
Buckets are containers for data stored in S3.

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

	Name	AWS Region	IAM Access Analyzer	Creation date
<input type="radio"/>	elasticbeanstalk-us-east-1-633393758690	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 18, 2024, 01:28:08 (UTC+05:30)
<input type="radio"/>	lambdabucketpranav	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 3, 2024, 15:06:02 (UTC+05:30)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

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



```
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']

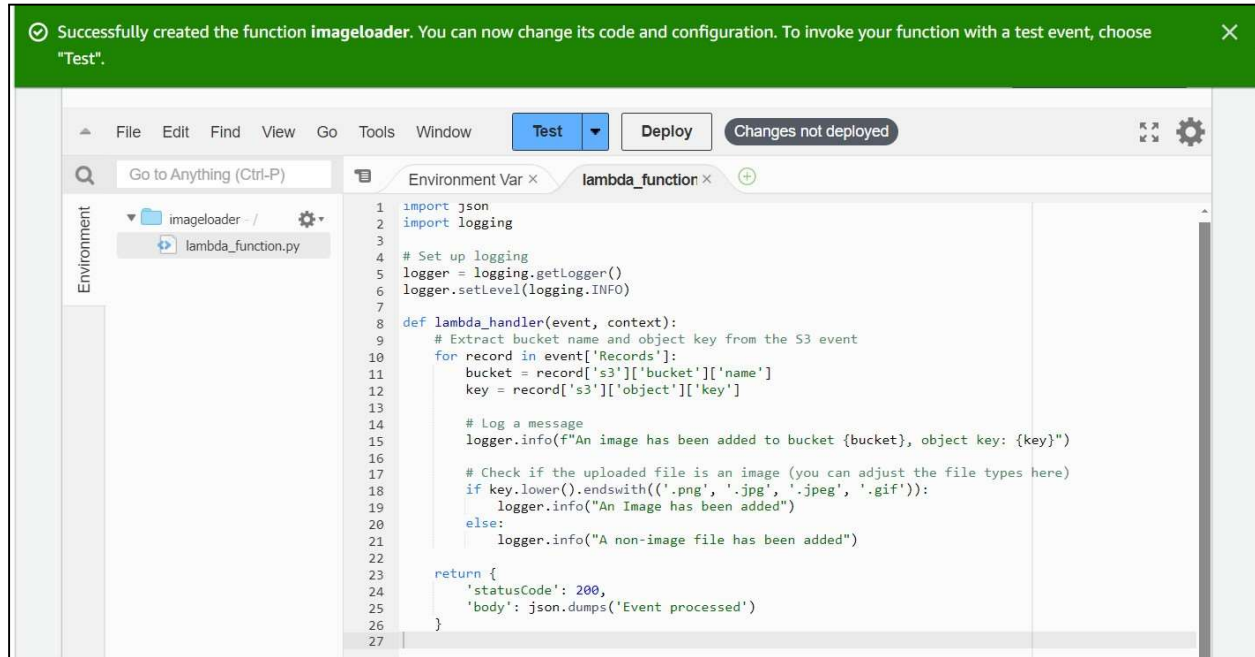
        # Log a message
        logger.info(f"An image has been added to bucket {bucket}, object key: {key}")

        # 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("Image Uploaded
            successfully") else:
                logger.info("A non-image file has been added")
```

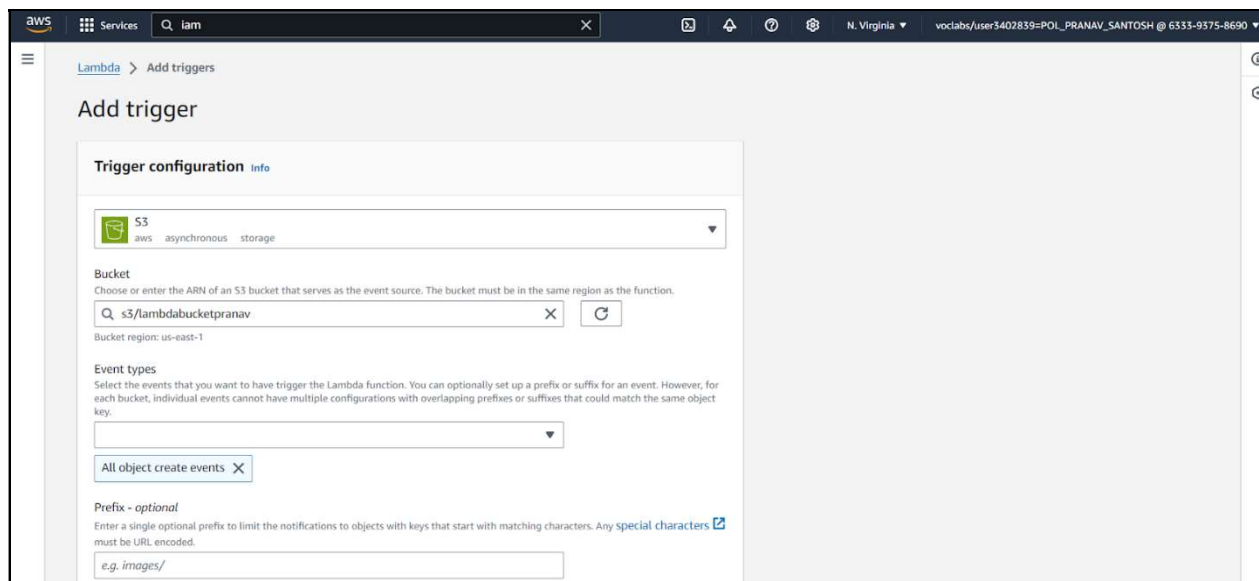
```

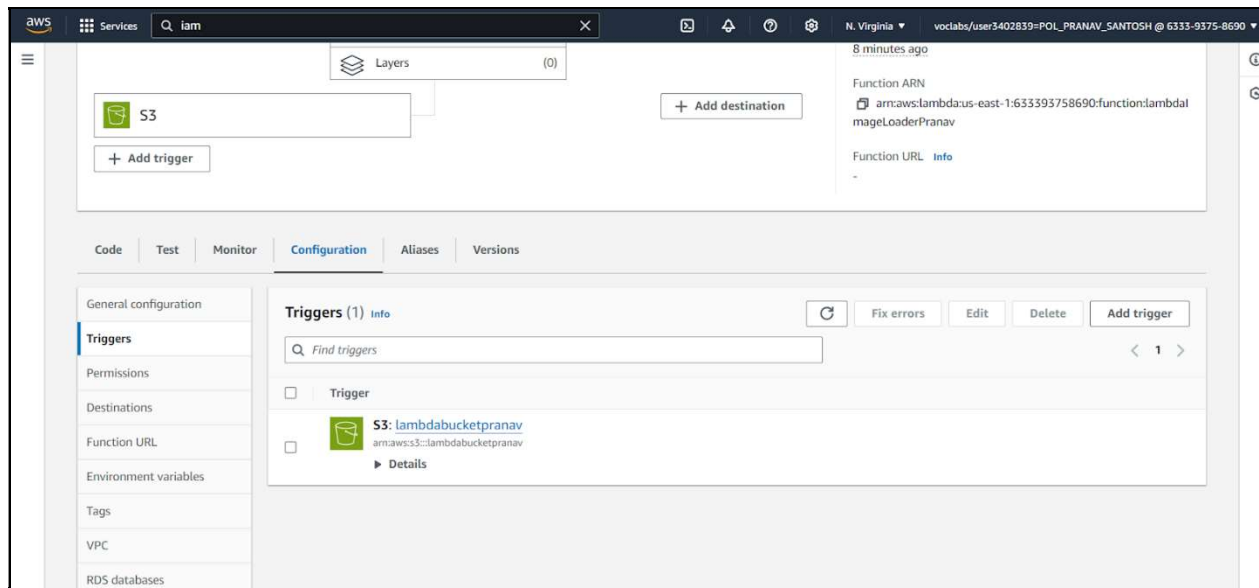
return {
    'statusCode': 200,
    'body': json.dumps('Event processed')
}

```

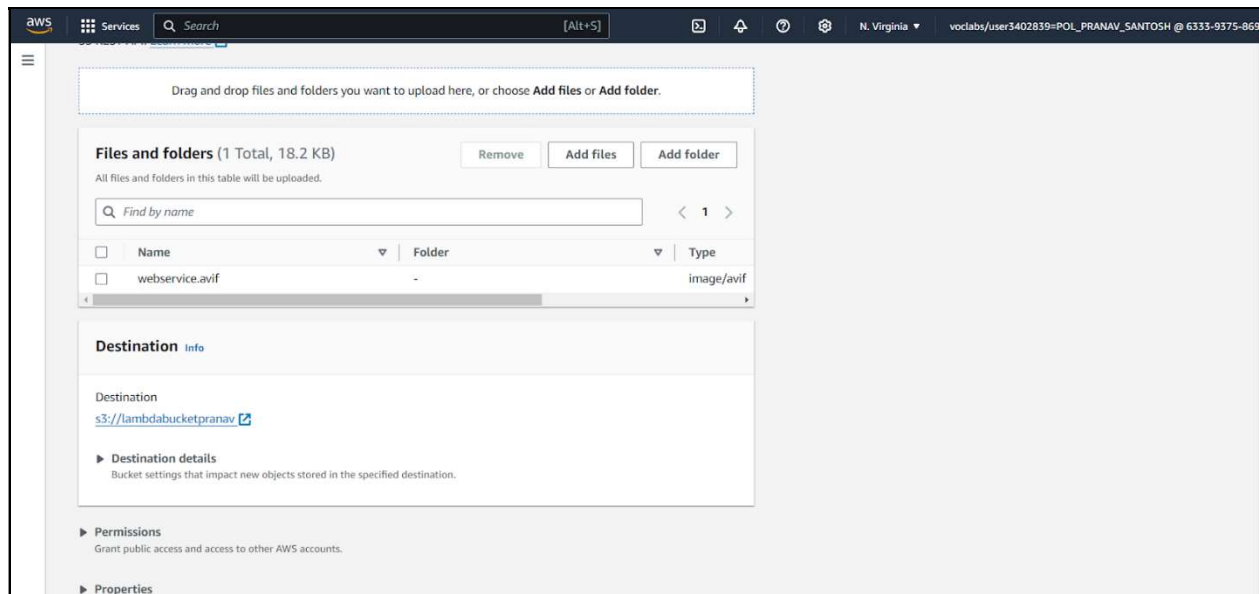


Configure S3 Trigger: Link the S3 bucket to the Lambda function by setting up a trigger. Specify that the function should be triggered when an object is created in the bucket (e.g., when an image is uploaded).

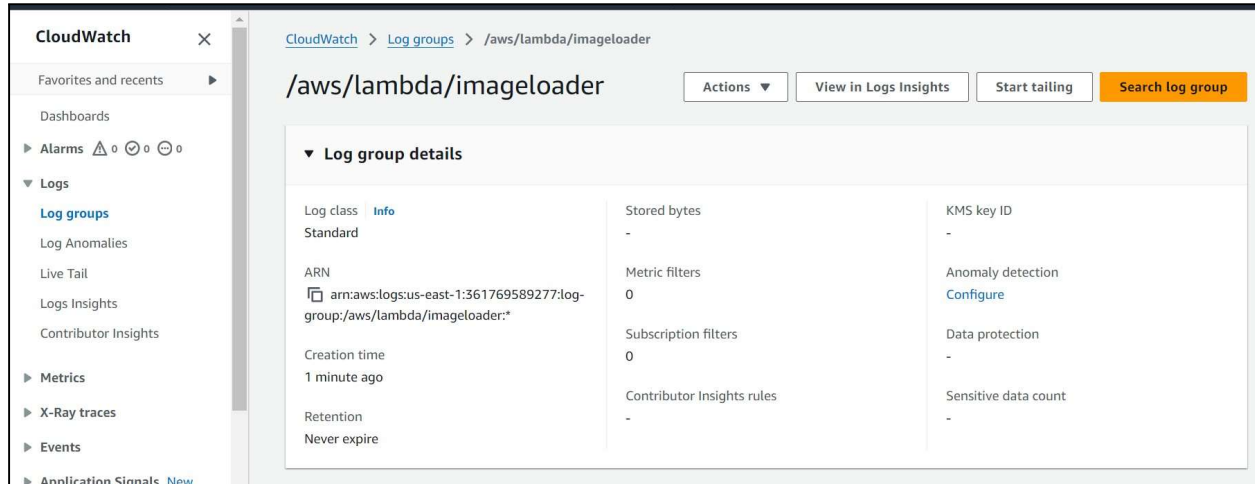




Upload an object (e.g., an image) to the S3 bucket to test the trigger

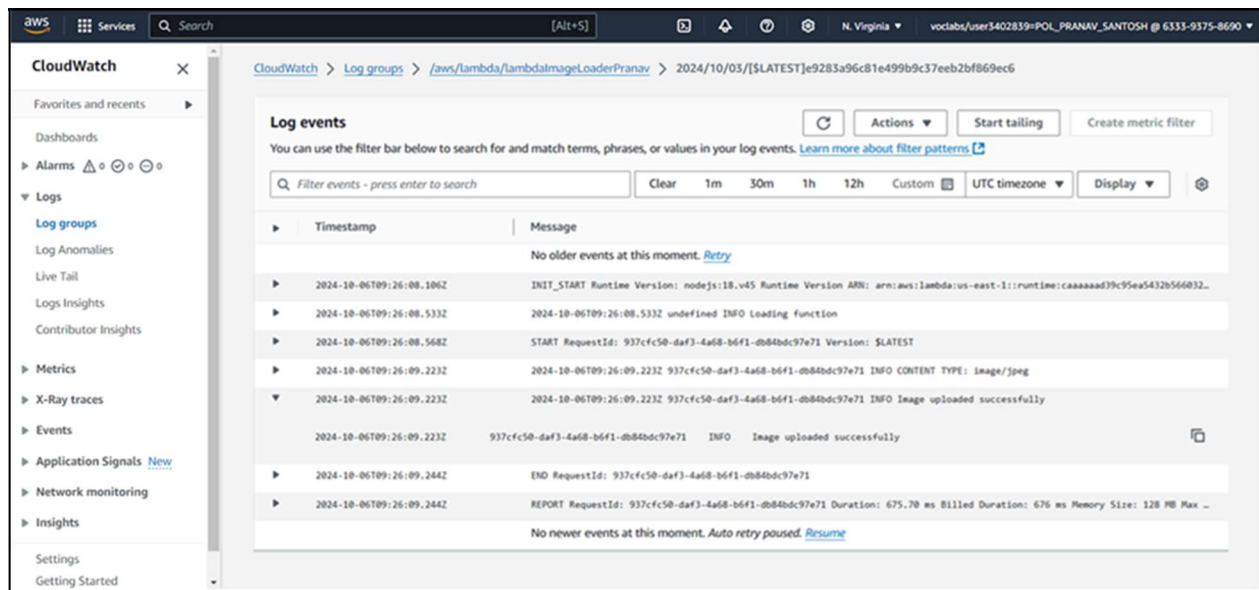


Test the Setup: Upload an object (e.g., an image) to the S3 bucket to test the trigger. The Lambda function should execute and log the message “An Image has been added” in AWS CloudWatch Logs



The screenshot shows the AWS CloudWatch console interface. On the left is a navigation sidebar with options like 'Dashboards', 'Alarms', 'Logs', 'Metrics', 'X-Ray traces', and 'Events'. The 'Logs' section is expanded, showing 'Log groups'. The main panel displays the details for the log group `/aws/lambda/imageloader`. At the top, there are buttons for 'Actions', 'View in Logs Insights', 'Start tailing', and 'Search log group'. Below this, the 'Log group details' section is visible, containing a table with the following information:

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



The screenshot shows the AWS CloudWatch console interface, specifically the 'Log events' view for the `/aws/lambda/imageloader` log group. The top navigation bar shows the path: `CloudWatch > Log groups > /aws/lambda/imageloader`. Below the navigation bar, there are buttons for 'Actions', 'Start tailing', and 'Create metric filter'. A search bar is present with the text 'Filter events - press enter to search'. The log events are displayed in a table with columns for 'Timestamp' and 'Message'. The events are as follows:

Timestamp	Message
2024-10-06T09:26:08.106Z	INIT_START Runtime Version: nodejs18.v45 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:caaaaad39c95ea5432b566032...
2024-10-06T09:26:08.533Z	2024-10-06T09:26:08.533Z undefined INFO Loading function
2024-10-06T09:26:08.568Z	START RequestId: 937cf50-daf3-4a68-b6f1-d084bdc97e71 Version: \$LATEST
2024-10-06T09:26:09.223Z	2024-10-06T09:26:09.223Z 937cf50-daf3-4a68-b6f1-d084bdc97e71 INFO CONTENT TYPE: image/jpeg
2024-10-06T09:26:09.223Z	2024-10-06T09:26:09.223Z 937cf50-daf3-4a68-b6f1-d084bdc97e71 INFO Image uploaded successfully
2024-10-06T09:26:09.223Z	937cf50-daf3-4a68-b6f1-d084bdc97e71 INFO Image uploaded successfully
2024-10-06T09:26:09.244Z	END RequestId: 937cf50-daf3-4a68-b6f1-d084bdc97e71
2024-10-06T09:26:09.244Z	REPORT RequestId: 937cf50-daf3-4a68-b6f1-d084bdc97e71 Duration: 675.70 ms Billed Duration: 676 ms Memory Size: 128 MB Max ...

