T22-82 AdvDevops Lab8
Krishna Ramchandani

Aim:

To create a lambda function which will log "An image has been added" once you add a specific object to a specific bucket in S3.

LO Mapping: LO1, LO5

Theory:

AWS Lambda is a serverless compute service that allows you to run code without provisioning or managing servers. Lambda functions can be triggered by various AWS services, including S3. By setting up an S3 event trigger for a specific bucket, you can automatically invoke a Lambda function when objects are added, modified, or deleted from that bucket.

In this scenario, the goal is to trigger a Lambda function when a specific object is added to a specific S3 bucket. The Lambda function will log the message, "An image has been added."

Key components:

1. S3 Bucket Event Trigger: You can configure an S3 bucket to trigger a Lambda function on events like object creation. In this case, the event will be when an object is uploaded to the bucket.
2. AWS Lambda Function: A Lambda function will be written to handle the incoming event from the S3 bucket. The function will check if the event corresponds to the specific object (or object type, such as an image) and then log a message.
3. Permissions: The Lambda function needs appropriate permissions to access the S3 bucket and write to logs (using AWS CloudWatch Logs).

Steps:

Step 1: Create an S3 Bucket
1. Login to AWS Console: Start by logging into your AWS Management Console.
2. Navigate to S3: Use the search bar to find and select the S3 service.
3. Create a New Bucket:

- Click on "Create Bucket."
- Enter a unique name for your bucket, such as `image-upload-bucket`.
- - Select a region close to you.
- Configure any additional settings you need, then click "Create Bucket."

Step 2: Set Up an IAM Role for Lambda
1. Open IAM: In the AWS Console, navigate to the Identity and Access Management
(IAM) service.
2. Create a New Role:
- Go to "Roles" and click "Create role."
- Choose "AWS Service" and then select "Lambda" as the trusted entity.
- Attach the following policies:
- `AmazonS3ReadOnlyAccess` (to allow the Lambda function to read objects from S3).
- `CloudWatchLogsFullAccess` (to allow Lambda to write logs to CloudWatch).
- Name the role (e.g., `LambdaS3LoggingRole`) and create it.

Step 3: Create the Lambda Function
1. Access Lambda: Use the AWS Console to navigate to the Lambda service.

2. Create a Function:
- Click on "Create function" and choose "Author from scratch." - Provide a function name like `S3ImageUploadLogger`.
- Select Python as the runtime.
- Use the IAM role you created earlier (`LambdaS3LoggingRole`). - Click "Create function."

Step 4: Add S3 as a Trigger for the Lambda Function
1. Configure the Trigger:
- In the Lambda console, scroll to the "Function Overview" section.
- Click "Add Trigger" and choose "S3" from the list.

- Select the bucket you created earlier (`image-upload-bucket`).
- Under "Event type," select "All object create events."
- Leave the prefix and suffix fields empty unless you want the function to trigger only for specific objects.
- Confirm the trigger by clicking "Add."

Step 5: Write and Deploy the Lambda Code

1. Edit the Lambda Function:

- Replace the default code with the following Python code:

```
import json
import boto3
import logging

logger = logging.getLogger()
logger.setLevel(logging.INFO) def
lambda_handler(event, context):
bucket_name = event['Records'][0]['s3']['bucket']['name']
object_key = event['Records'][0]['s3']['object']['key']

logger.info(f"An Image has been added to {bucket_name} with key:
{object_key}")

return {
'statusCode': 200,
'body': json.dumps(f"Logged: An Image has been added to
{bucket_name} with
key: {object_key}")
}
```

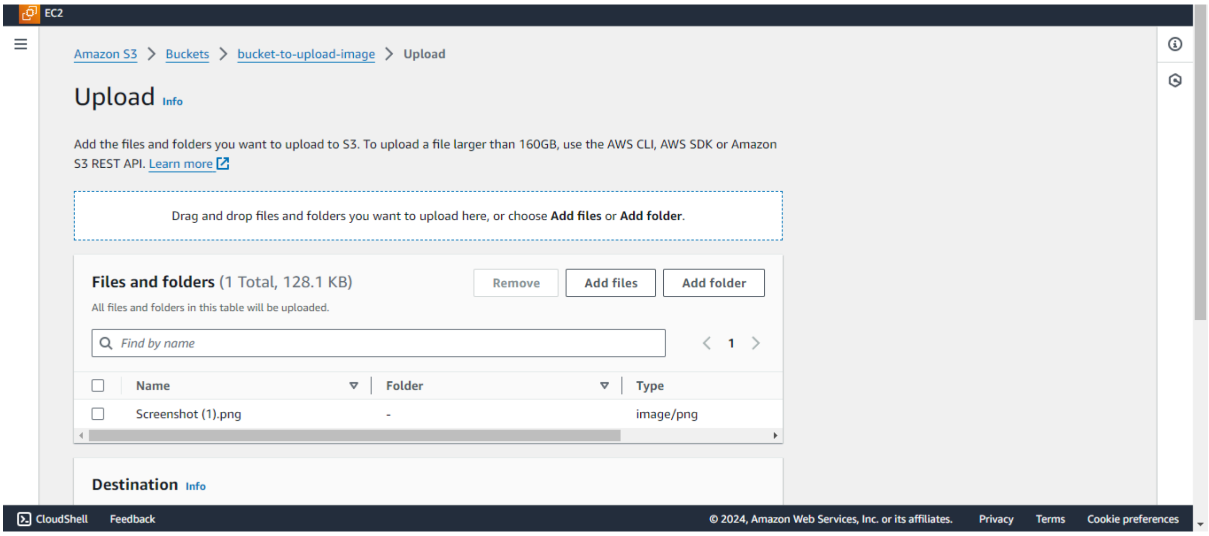
2. Deploy the Code: After entering the code, click "Deploy" to save and apply the changes.

Step 6: Test the Setup

1. Upload an Image to S3: Return to your S3 bucket and upload an image or any file.

2. Check CloudWatch Logs:
- Open the CloudWatch service from the AWS Console.
- Go to "Logs" and find the logs for your Lambda function.
- You should see a log entry confirming that an image has been added, including details
of the bucket and object key.

Output:



Upload

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**.

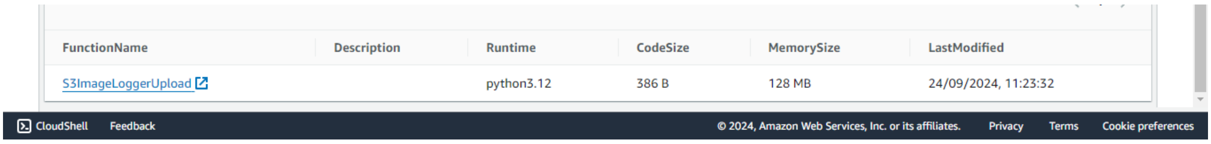
Files and folders (1 Total, 128.1 KB)

All files and folders in this table will be uploaded.

□	Name	Folder	Type
□	Screenshot (1).png	-	image/png

Destination Info

© 2024, Amazon Web Services, Inc. or its affiliates.



FunctionName	Description	Runtime	CodeSize	MemorySize	LastModified
S3ImageLoggerUpload		python3.12	386 B	128 MB	24/09/2024, 11:23:32

© 2024, Amazon Web Services, Inc. or its affiliates.

Conclusion:

By integrating AWS S3 with AWS Lambda, you can automate the process of monitoring and reacting to changes in your S3 bucket. In this case, we used an S3 event trigger to invoke a Lambda function whenever an image file is added to a specific bucket. The Lambda function logs the addition of the image to CloudWatch, which helps in tracking events in a serverless and scalable way. This approach can be extended to other file types or automated workflows in response to file uploads.