

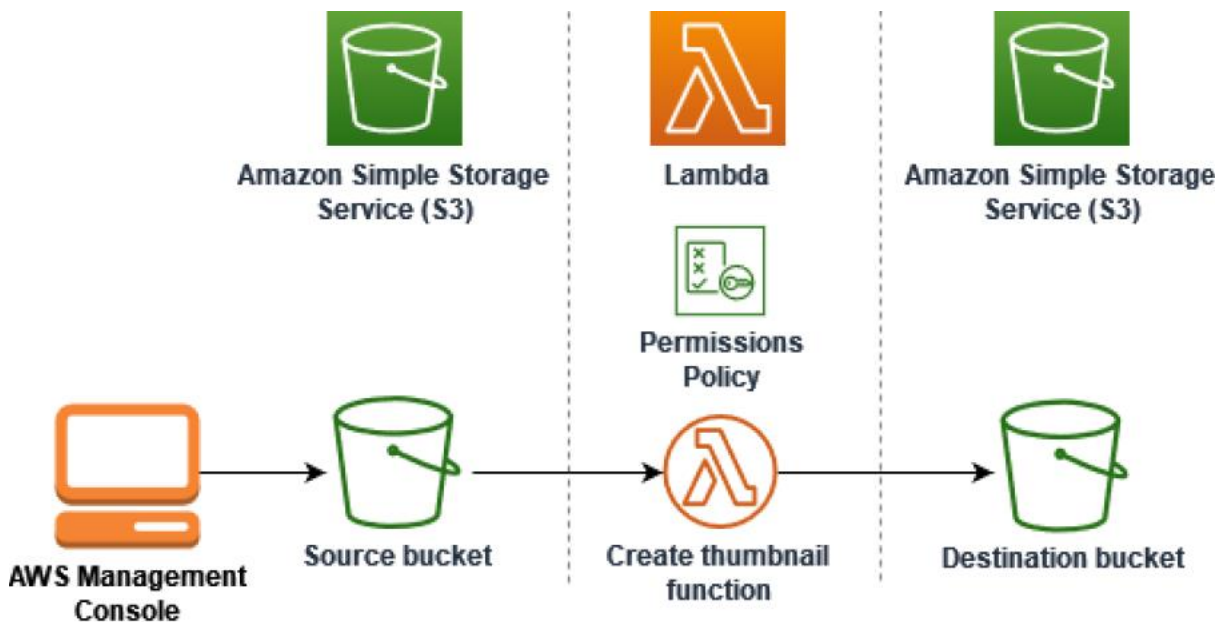
School: CLOUDBOOSTA	Course: Cloud Computing & DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Module: Serverless Computing & Database
	Class 6: Assignment 6 of 6_ Serverless_3
Student No: CBALC0168	Issue Date: April 1, 2026

Tasks:

1. Create and configure Amazon S3 buckets (source and destination)
2. Create and configure IAM role for Lambda permissions
3. Build an AWS Lambda function for image processing
4. Configure S3 trigger to invoke Lambda automatically
5. Upload and deploy Lambda function code
6. Test and validate automated thumbnail generation

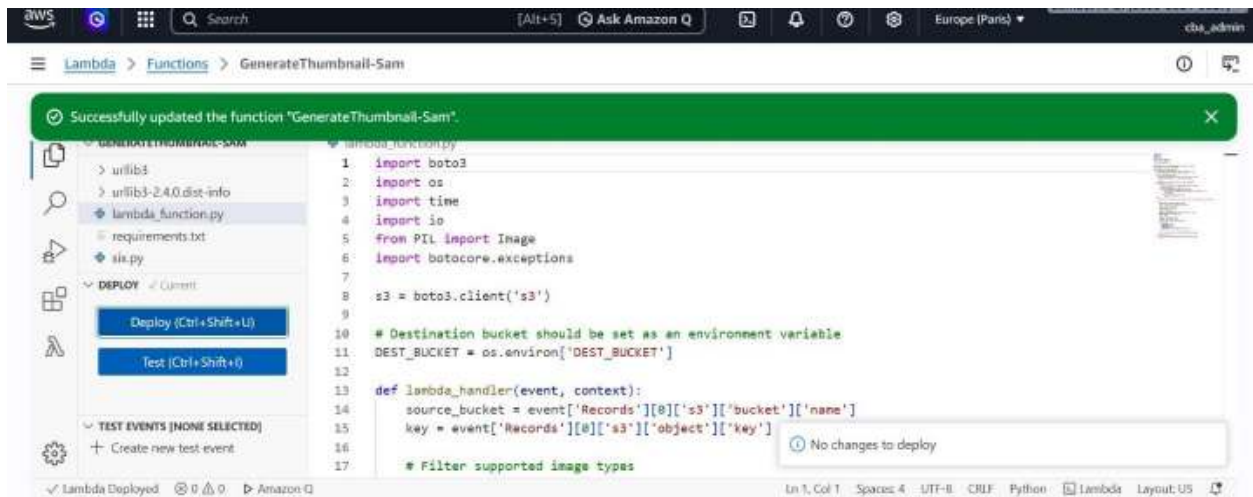
Goal:

To build a fully automated serverless system that generates thumbnail images whenever a file is uploaded to an Amazon S3 bucket.



Result:

- Two S3 buckets were successfully created (source and destination).
- AWS Lambda function was deployed and triggered automatically.
- Image uploaded to source bucket generated a resized thumbnail.
- Thumbnail stored in “**thumbnails/**” folder in destination bucket.



As seen in the screenshots:

- Page 21 shows the **thumbnails folder created automatically**
- Page 22 confirms the resized image stored successfully
- CloudWatch logs (page 23) confirm successful Lambda execution

Action:

• Step 1: Created S3 Buckets

- Created **source bucket** for image uploads
- Created **destination bucket** for storing thumbnails
- Verified successful creation via AWS console

Evidence:

- Page 7 shows source bucket created successfully
- Page 9 shows destination bucket created successfully

• Step 2: Configured IAM Role

- Created IAM Role for Lambda

- Assigned permissions:
 - AmazonS3FullAccess
 - AWSLambdaBasicExecutionRole

Evidence:

- Page 11 shows role creation with attached policies
- Page 12 confirms role creation success

• Step 3: Created Lambda Function

- Function Name: **GenerateThumbnail-Sam**
- Runtime: Python 3.12
- Linked IAM role for permissions

Evidence:

- Page 13 shows Lambda function configuration

• Step 4: Configured S3 Trigger

- Linked S3 source bucket to Lambda
- Event type: **PUT (object upload)**
- Enabled automatic execution

Evidence:

- Page 14 shows trigger configuration (PUT event)
- Page 15 confirms trigger successfully attached

• Step 5: Uploaded Lambda Code

- Uploaded Python code package (.zip)
- Configured environment variable:
 - **DEST_BUCKET = destination bucket name**

Evidence:

- Page 16 shows environment variable configuration
- Page 18 shows deployed Lambda code

• Step 6: Tested System

- Uploaded test image to source bucket
- Lambda triggered automatically

- Thumbnail generated and stored

Evidence:

- Page 19 shows successful upload
- Page 21 shows thumbnails folder created

• Step 7: Verification & Debugging

- Used **CloudWatch Logs** to monitor execution
- Verified successful Lambda invocation
- No critical errors observed

Evidence:

- Page 23 shows CloudWatch logs confirming execution

Screenshots follow.

Summary

This project successfully demonstrates a **serverless event-driven architecture** using AWS services:

- **Amazon S3 → Event Trigger**
- **AWS Lambda → Compute Layer**
- **IAM → Secure Access Control**

As explained in the lab guide, the workflow is:

Upload → Trigger → Process → Store

This architecture eliminates the need for servers and enables:

- Automatic scaling
- Cost efficiency (pay-per-use)
- High availability

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region

Europe (Paris) eu-west-3

Bucket namespace

Choose the namespace where you want to create your bucket. [Learn more](#)

☒ Global namespace

By default, S3 creates general purpose buckets in the global namespace.

☐ Account Regional namespace (recommended)

General purpose buckets created in your account. Regional namespaces are unique to your account. These buckets can never be created by another AWS account.

Bucket name [Info](#)

my-thumbnail-source-bucket-samuel

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn more](#)

Copy settings from existing bucket - optional

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Object Ownership

☒ ACLs disabled (recommended)

All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

☐ ACLs enabled

Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☒ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☒ Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☒ Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☒ Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☒ Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

☒ Disable

☐ Enable

Tags - optional

You can use bucket tags to analyze, manage and specify permissions for a bucket. [Learn more](#)

[Info](#) You can use s3:ListTagsForResource, s3:TagResource, and s3:UntagResource APIs to manage tags on S3 general purpose buckets for access control in addition to cost allocation and resource organization. To ensure a seamless transition, please provide permissions to s3:ListTagsForResource, s3:TagResource, and s3:UntagResource actions. [Learn](#)

[more](#)

No tags associated with this bucket.

Add new tag

You can add up to 50 tags.

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Secure your objects with two separate layers of encryption. For details on pricing, see [DSSE-KMS pricing](#) on the [Storage](#) tab of the [Amazon S3 pricing page](#).

- ☒ Server-side encryption with Amazon S3 managed keys (SSE-S3)
- ☐ Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ☐ Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- ☐ Disable
- ☒ Enable

► Advanced settings

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#)

Create bucket

✔ Successfully created bucket "my-thumbnail-source-bucket-samuel"

✕

my-thumbnail-source-bucket-samuel

Info

- Objects
- Metadata
- Properties
- Permissions
- Metrics
- Management
- File systems - new
- Access Points

Objects (0)



 Copy S3 URI

 Copy URL

 Download

 Open 

Delete

Actions 

Create folder

 Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

 Find objects by prefix

< 1 > 

 Name	 Type	 Last modified	 Size	 Storage class
No objects				
You don't have any objects in this bucket.				

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Paris) eu-west-3

Bucket namespace
Choose the namespace where you want to create your bucket. [Learn more](#)

☒ **Global namespace**
By default, S3 creates general purpose buckets in the global namespace.

☐ **Account Regional namespace (recommended)**
General purpose buckets created in your account Regional namespace are unique to your account. These buckets can never be created by another AWS account.

Bucket name [Info](#)

my-thumbnail-destination-bucket-samuel

✔ Successfully created bucket "my-thumbnail-destination-bucket-samuel"

my-thumbnailed-destination-bucket-samuel Info

[Objects](#) [Metadata](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [File systems - new](#) [Access Points](#)

Objects (0)



Copy S3 URI

Copy URL

Download

Open ↗

Delete

Actions ▼

Create folder

⬆️ Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 >

	Name	Type	Last modified	Size	Storage class
--	------	------	---------------	------	---------------

No objects

You don't have any objects in this bucket.

$\angle$

Dashboard

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

Temporary delegation requests

2

Role name

▲

Trusted entities

Last activity

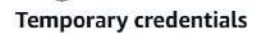
Info

Authenticate your non AWS workloads and securely provide access to AWS services.



X.509 Standard

Use your own existing PKI infrastructure or use [AWS Certificate Manager Private Certificate Authority](#)  to authenticate identities.



Use temporary credentials with ease and benefit from the enhanced security they provide.

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.

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.

Architecture [Info](#)

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

☐ arm64☒ x86_64

Permissions [Info](#)

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

▼ Change default execution role

Execution role

To access other AWS services and resources your function needs an IAM role. Choose a role with the right permissions for your function.

☐ Create default role☒ Use another role[Create new role](#)

[View role details in IAM](#)

► Additional configurations

Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

[Cancel](#)[Create function](#)

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: eu-west-3

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.



PUT



Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters. Any [special characters](#) must be URL encoded.

Suffix - optional

Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any [special characters](#) must be URL encoded.

Recursive invocation

If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)

- ☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger.

[Info](#)

Cancel

Add

GenerateThumbnail-Sam

[Throttle](#)[Copy ARN](#)[Actions](#)

✓ The trigger my-thumbnail-source-bucket-samuel was successfully added to function GenerateThumbnail-Sam.

Function overview Info

[Diagram](#)[Template](#)

 **GenerateThumbnail-Sam**

 Layers (0)

[Export to Infrastructure Composer](#)[Download](#)

Function ARN

 `arn:aws:lambda:eu-west-3:009850210027:function:GenerateThumbnail-Sam`

Description

-

Last modified

6 minutes ago



S3

[+ Add trigger](#)[+ Add destination](#)[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Configuration

[General configuration](#)[Triggers](#)[Permissions](#)[Destinations](#)[Function URL](#)[Environment variables](#)[Tags](#)[VPC](#)[RDS databases](#)[Monitoring and operations tools](#)[Concurrency and recursion detection](#)[Asynchronous invocation](#)[Code signing](#)[File systems](#)[State machines](#)

Triggers (1) Info

[Fix errors](#)[Edit](#)[Delete](#)[Add trigger](#)

< 1 >

☐ Trigger

☐  **S3: my-thumbnail-source-bucket-samuel**
`arn:aws:s3:::my-thumbnail-source-bucket-samuel`
[Details](#)

Edit environment variables

Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

Key	Value	
DEST_BUCKET	my-thumbnail-destination-bucket-samuel	<button>Remove</button>

Add environment variable

► Encryption configuration

Cancel

Save

✔ Successfully updated the function "GenerateThumbnail-Sam".

GenerateThumbnail-Sam

Throttle

Copy ARN

Actions ▾

▼ Function overview Info

Diagram | Template

GenerateThumbnail-Sam

Layers (0)

S3

+ Add trigger

+ Add destination

Export to Infrastructure Composer

Download ▾

Function ARN

`arn:aws:lambda:eu-west-3:009850210027:function:GenerateThumbnail-Sam`

Description

-

Last modified

22 seconds ago

- Configuration
- General configuration
- Triggers
- Permissions
- Destinations
- Function URL
- Environment variables
- Tags
- VPC
- RDS databases
- Monitoring and operations tools
- Concurrency and recursion detection
- Asynchronous invocation
- Code signing
- File systems
- State machines

Environment variables (1) Edit

The environment variables below are encrypted at rest with the default Lambda service key.

Key	Value
DEST_BUCKET	my-thumbnail-destination-bucket-samuel

Successfully updated the function "GenerateThumbnail-Sam".

GENERATE THUMBAIL-SAM

- > urllib3
- > urllib3-2.4.0.dist-info
- lambda_function.py
- requirements.txt
- six.py

DEPLOY ✓ Current

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

+ Create new test event

```
lambda_function.py
1 import boto3
2 import os
3 import time
4 import io
5 from PIL import Image
6 import botocore.exceptions
7
8 s3 = boto3.client('s3')
9
10 # Destination bucket should be set as an environment variable
11 DEST_BUCKET = os.environ['DEST_BUCKET']
12
13 def lambda_handler(event, context):
14     source_bucket = event['Records'][0]['s3']['bucket']['name']
15     key = event['Records'][0]['s3']['object']['key']
16
17     # Filter supported image types
```

No changes to deploy

Upload succeeded
For more information, see the [Files and folders](#) table.

Upload: status

Close

After you navigate away from this page, the following information is no longer available.

Summary

Destination
s3://my-thumbnail-source-bucket-samuel

Succeeded
 1 file, 6.6 KB (100.00%)

Failed
 0 files, 0 B (0%)

Files and folders

Configuration

Files and folders (1 total, 6.6 KB)

< 1 >

Name	Folder	Type	Size	Status	Error
guy-ritchie.jpg	-	image/jpeg	6.6 KB	Succeeded	-

Objects (1)

Copy S3 URI

Copy URL

Download

Open ↗

Delete

Actions ▼

Create folderUpload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 > ⚙

☐	Name ▲	Type	Last modified ▼	Size ▼	Storage class ▼
☐	guy-ritchie.jpg	jpg	April 9, 2026, 21:51:23 (UTC+03:00)	6.6 KB	Standard

Objects (1)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 >

☐	Name	Type	Last modified	Size	Storage class
☐	thumbnails/	Folder	-	-	-

Objects (1)

  Copy S3 URI  Copy URL  Download  Open  Delete

Actions 

Create folder  Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

 Find objects by prefix

< 1 > 

☐	Name	Type	Last modified	Size	Storage class
☐	 guy-ritchie.jpg	jpg	April 9, 2026, 21:51:26 (UTC+03:00)	2.9 KB	Standard

CloudWatch

Favorites and recents

Ingestion

Dashboards

Alarms 0 0 0

AI Operations

GenAI Observability

Application Signals (APM) New

Infrastructure Monitoring

Logs

Log Management

Log Analytics Preview

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Metrics New

Network Monitoring

Setup

Log events

Actions

Start tailing

Create metric filter

Filter events - press enter to search

1m 1h 5 UTC timezone

Display

Timestamp	Message
No older events at this moment. Retry	
2026-04-09T18:51:24.164Z	INIT_START Runtime version: python:3.12.mainlinev2.v0 Runtime Version ARN: arn:aws:lambda:eu...
2026-04-09T18:51:24.771Z	START RequestId: bd2eab2f-4582-4fbc-ab01-28956d5948f1 Version: \$LATEST
2026-04-09T18:51:25.325Z	END RequestId: bd2eab2f-4582-4fbc-ab01-28956d5948f1
2026-04-09T18:51:25.126Z	REPORT RequestId: bd2eab2f-4582-4fbc-ab01-28956d5948f1 Duration: 553.02 ms Billed Duration: ...
No newer events at this moment. Auto retry paused . Resume	