

# ***Project-2***

## ***Deploy a Static Website on AWS***

# ***Table of contents***

---

- 1. Introduction**
- 2. Setup Checklist for Mini Project**
- 3. Instructions**
- 4. Objective**
- 5. Architecture diagram**
- 6. Technology Used**
- 7. Guidelines on the Functionality to be built**
- 8. Advantages of deploy a static Website**

# 1 Introduction

---

This documentation describes how to build a serverless image processing application using AWS services. The application resizes and optimizes images automatically when they are uploaded to an S3 bucket. It is designed to be scalable, cost-effective, and maintenance-free.

## 2 Setup Checklist for Mini Project

---

Before getting started, ensure you have the following:

- An active AWS account
- AWS CLI installed and configured
- Node.js (v18 or later)
- AWS SAM CLI for building and deploying serverless applications
- Basic knowledge of JavaScript and AWS services

## 3 Instructions

---

Follow these steps to deploy and test the application:

1. Create two S3 buckets (`my-images-source` and `my-images-processed`)
2. Write the image processing logic using Node.js and Sharp
3. Define the infrastructure using AWS SAM
4. Build and deploy using SAM CLI
5. Upload images to the source bucket and observe the processed output

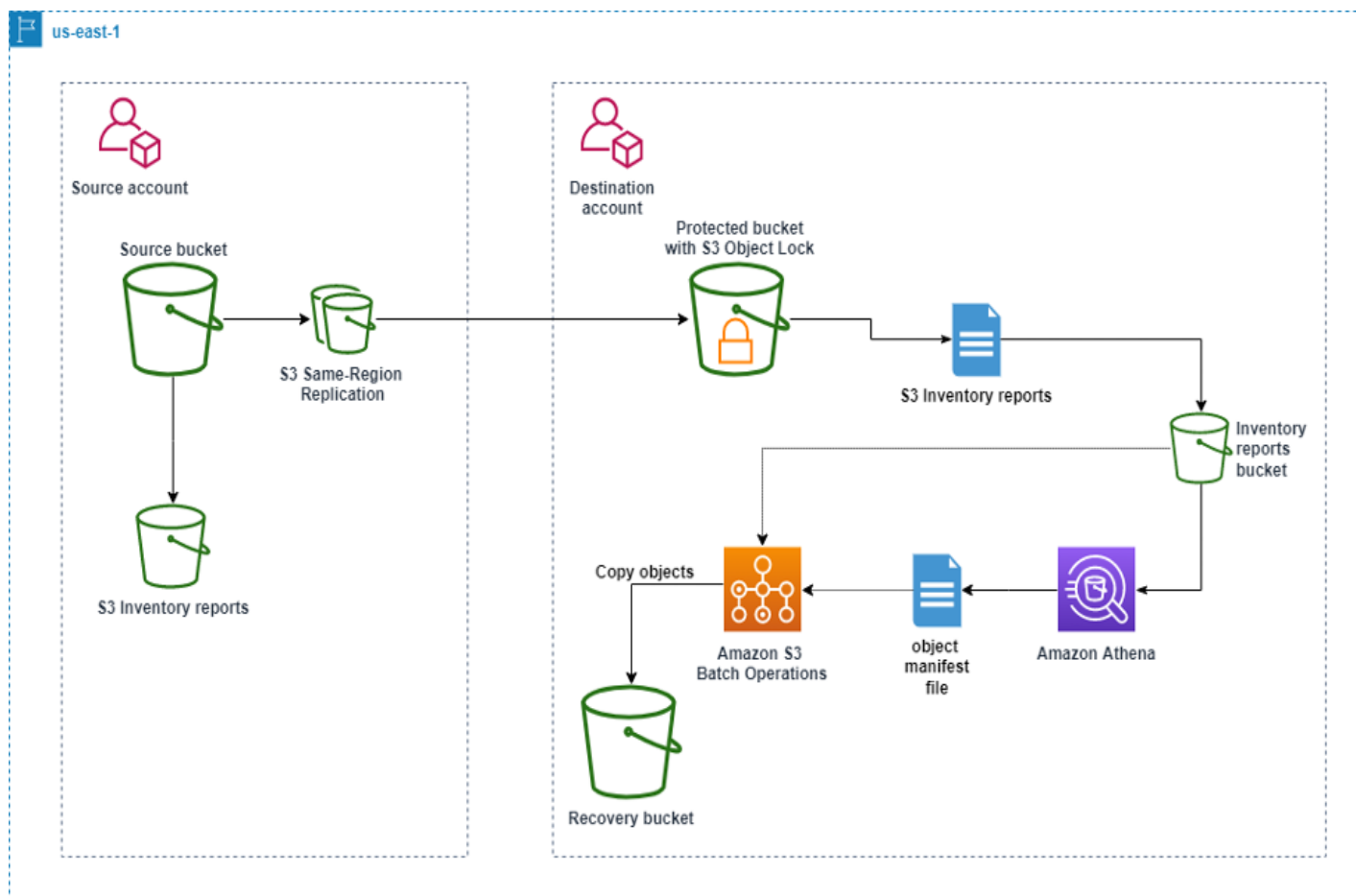
## 4 Objective

---

The goal is to automate image optimization by:

- Triggering a Lambda function on S3 upload events
- Resizing and compressing images using the Sharp library
- Storing optimized images in a target S3 bucket

## 5 Architecture diagram



## 6 Technology Used

---

- **Amazon S3:** For storing original and optimized images
- **AWS Lambda:** To process images without managing servers
- **AWS SAM:** For defining and deploying serverless resources
- **Sharp Library:** For efficient image transformation
- **Node.js:** JavaScript runtime for Lambda function code



# 7 Guidelines on the Functionality to be built

---

## Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
  - Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
  - Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1**.

## Task 2: Creating a S3 Bucket

1. In this task, we are going to create a new S3 bucket in the US East (N. Virginia) region with a unique name disabling ACLs, and allowing public access for hosting the static website.
2. Navigate to S3 by clicking on the Services menu at the top, then click on S3 in the Storage section.
3. In the S3 dashboard, click on the Create Bucket button.
4. In the General Configuration, Bucket name: Enter abcxzyz
  - Note: S3 Bucket names are globally unique, choose a name that is available. Maybe you can enter your name and create one.
5. AWS Region: Select US East (N. Virginia) us-east-1
6. Object ownership: Select ACLs disabled (recommended) option
7. In the option of Block Public Access settings for this bucket, Uncheck the option of Block all public access.
  - Check the I acknowledge that the current settings might result in this bucket and the objects within becoming public checkbox.
8. 7. Keep everything default and click on Create Bucket button.

## Task 3: Enable Static Website Hosting settings

1. In this task, we will enable static website hosting for our S3 bucket, configure it to use index.html and error.html, copy the provided endpoint, upload two files, and configure the bucket policy by copying its ARN and pasting the provided policy code.
2. To proceed, go to the **S3 bucket name** that you created and click on it. After that, navigate to the **Properties** tab which can be found at the top of the screen.

3. Scroll down to the **Static website hosting** section and click on **Edit** button.

### Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

Disabled

Edit

1. In the **Static website hosting** dialog box
  - Static website hosting: Select **Enable**
  - Hosting type: Choose Host a static website
  - Index document: Type index.html
  - Error document: Type error.html
  - Click on Save Changes

### Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

☐ Disable

☒ Enable

Hosting type

☒ Host a static website  
Use the bucket endpoint as the web address. [Learn more](#)

☐ Redirect requests for an object  
Redirect requests to another bucket or domain. [Learn more](#)

**i** For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Index document

Specify the home or default page of the website.



index.html

Error document - *optional*

This is returned when an error occurs.

error.html

- The next step is to download the zip file by clicking on the link, extract it, and upload two files named **index.html** and **error.html** to the S3 bucket you created earlier.
- Go to the **Properties** tab of your S3 bucket, and find the **Static website hosting** section. **Copy** the Endpoint provided in this section to your clipboard and **save** it for future reference

Objects (2)				
Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permission				
<input type="text" value="Find objects by prefix"/>				
<input type="checkbox"/>	Name	Type	Last modified	
<input type="checkbox"/>	 <a href="#">error.html</a>	html	December 15, 2020, 21:44 (UTC+05:30)	
<input type="checkbox"/>	 <a href="#">index.html</a>	html	December 15, 2020, 21:44 (UTC+05:30)	

6. To configure your S3 bucket, access the **Permissions** tab and make the necessary configurations.

- In the **Permissions** tab, Click on the **Edit** button beside the **Bucket Policy**.

**Bucket policy**  
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

EditDelete

No policy to display.

Copy

- You will be able to see a Blank policy editor.
- Before creating the policy, you will need to copy the **ARN** (Amazon Resource Name) of your bucket.
- Copy the **ARN** of your bucket to the clipboard. It is displayed at the top of the policy editor. it will look like **ARN:"arn:aws:s3:::your-bucket-name"**.
- In the policy below, **Update** the bucket ARN on the Resource key value and **paste** the below policy code in the editor.

```
{  
  "Id": "Policy1",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Stmt1",  
      "Action": [  
        "s3:GetObject"  
      ],  
      "Effect": "Allow",  
      "Resource": "replace-this-string-with-your-bucket-arn/*",  
      "Principal": "*"   
    }  
  ]  
}
```

Click on **Save changes** button.

## Task 4: Test the website

1. Now copy the **static website URL** (that we saved earlier) and run it in your browser. You will be able to see the index.html file's text. A sample screenshot is attached below:

## Task 5: Test the website's error page

- Copy the static website URL (which we saved earlier) , but this time, add some random characters to the end of the url to break it. When satisfied, hit **[Enter]**. You will be redirected to the **error.html** page automatically.



# 8 Advantages to deploy a Static Website

## Advantages of Deploying a Static Website

- **Scalability:** Automatically handles increasing traffic without additional setup
- **Low Cost:** No servers to manage; you pay only for the resources you use
- **Reliability:** Built on AWS's globally distributed infrastructure
- **Security:** Can be combined with IAM, CloudFront, and S3 bucket policies for secure delivery
- **Speed:** Optimized media enhances website loading times significantly

## Static Website Deployment

