**PROJECT**

**AIM:- Create a serverless image processing application that automatically resizes and optimizes images uploaded to an S3 Bucket**.

**PREREQUISITES:-**

1. An AWS Account.
2. Thorough Knowledge about AWS Services like S3, LAMBDA, IAM.

Knowledge about working with programming languages**.**

**WHAT IS AWS S3?**

Amazon S3 is a Simple Storage Service in[AWS](https://www.geeksforgeeks.org/aws-tutorial/)that stores files of different types like Photos, Audio, and Videos as Objects providing more scalability and security.  It is probably the most commonly used, go-to storage service for AWS users given the features like extremely high availability, security, and simple connection to other AWS Services.

**WHAT IS AWS LAMBDA?**

*AWS Lambda* is a serverless compute service that allows you to run code without provisioning or managing servers. It automatically handles the administration of the compute resources, including server and operating system maintenance, capacity provisioning, automatic scaling, and logging. This makes it an ideal solution for applications that need to scale up rapidly and scale down to zero when not in demand.

**WHAT IS AWS IAM?**

*IAM* stands for *Identity and Access Management* in AWS. It is a web service that helps you securely control access to AWS resources. With IAM, you can manage permissions that control which AWS resources users can access. IAM provides the infrastructure necessary to control authentication and authorization for your AWS accounts.

Let us now take a look at how we can create a serverless image processing application.

**STEPS TO CREATE A SERVERLESS IMAGE PROCESSING APPLICATION**

**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 S3 Buckets (source bucket &destination bucket).

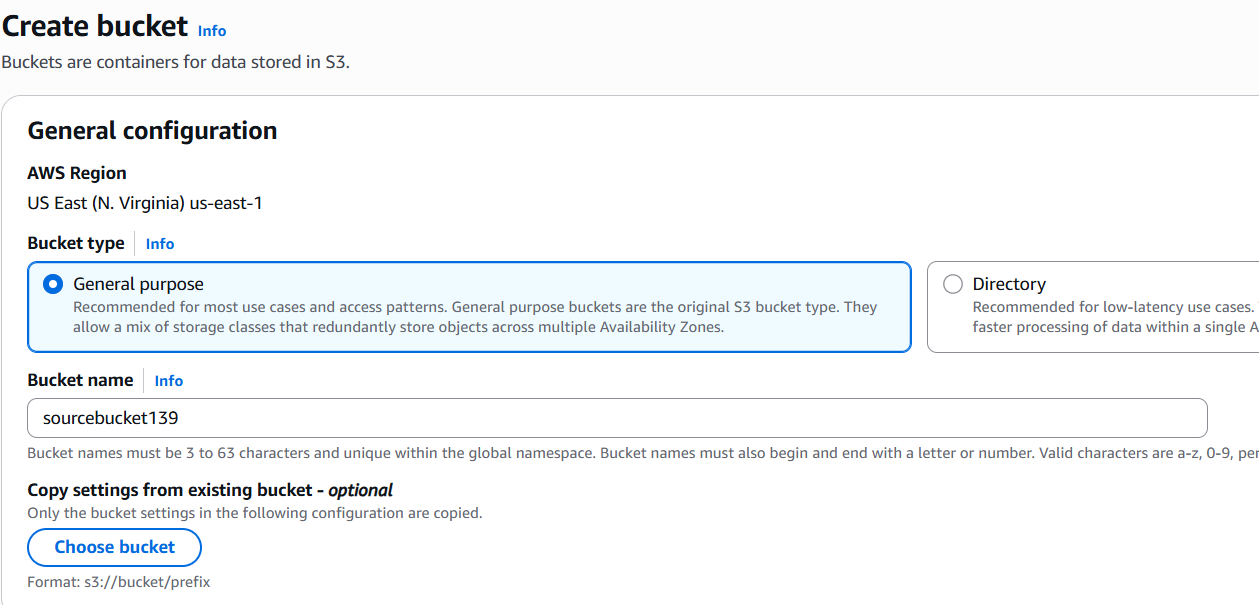
1. Navigate to S3 by clicking on the Services menu at the top, then click on S3 in the Storage section.

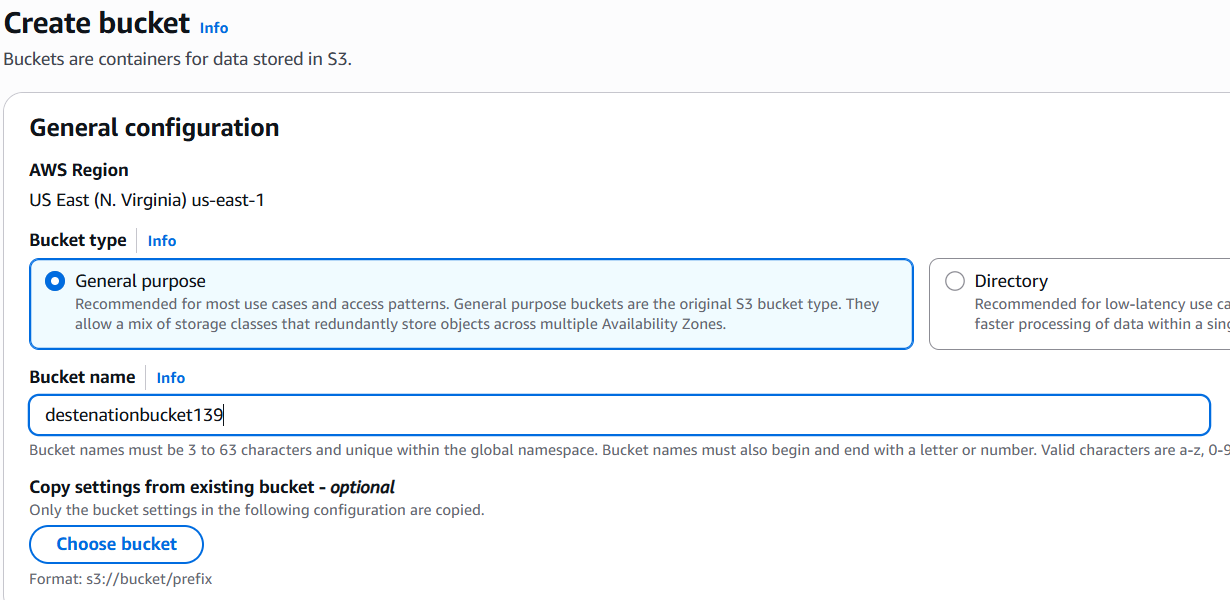
2. In the S3 dashboard, click on the Create Bucket button.

3. In the General Configuration, Bucket name: Enter abcxyz

· **Note**: S3 Bucket names are globally unique, choose a name that is available. Maybe you can enter your name and create one.

1. Leave all other credentials as default and click on create bucket. Your source bucket is now created.
2. Similarly, create another bucket which will be the destination bucket.





**NOTE:-** In our case we have given the names as sourcebucket139 & destinationbucket139

**TASK 3:-** Creating IAM Policy and attaching it with an IAM Role.

1. Navigate to IAM and then go to policy which is in the left side of IAM dashboard.
2. Now, in policy click on create policy and go to the JSON format in specific permissions section.
3. In the JSON format edit the default code with the code given below.

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"logs:PutLogEvents",

"logs:CreateLogGroup",

"logs:CreateLogStream"

],

"Resource": "arn:aws:logs:\*:\*:\*"

},

{

"Effect": "Allow",

"Action": [

"s3:GetObject"

],

"Resource": "arn:aws:s3:::sourcebucket139/\*"

},

{

"Effect": "Allow",

"Action": [

"s3:PutObject"

],

"Resource": "arn:aws:s3:::destinetionbucket139/\*"

}

]

}

**NOTE:-** In the following code change the arn of your source bucket and destination bucket.

4.Now, click on next and give a name to the policy and leave all other credentials as default and click on create.

5. Now go to IAM roles and click on create role

and in the use case section select LAMBDA and click on next.

6.Now, select the policy which you have created earlier click on next and then give a name to the role and click on create role.

**TASK 4:-** Create a LAMBDA function and adding the role which we have created.

1. Navigate to LAMBDA and click on create function choose author from scratch.

* Enter function name
* In the runtime section choose Node.js.18.x
* In architecture choose x86\_64
* Now, in change default execution role choose use an existing role and choose the role which you have created earlier.
* Leave all other credentials as default and click on create function.

1. Now go to the code section in the function you have created and upload the zip file attached below.



3.Now test the following code.

**TASK 5:-** Adding trigger to the lambda function.

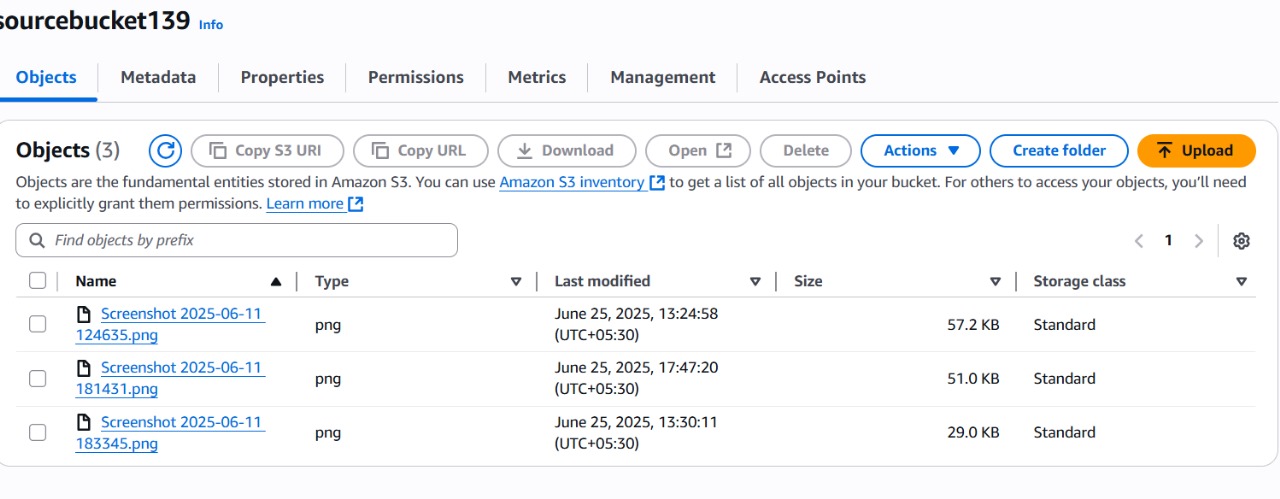
1. Go, to the add trigger option in the lambda function.
2. Choose S3 in the trigger configuration section .
3. Now, select the source bucket which you have created earlier.
4. Now, click on add trigger.

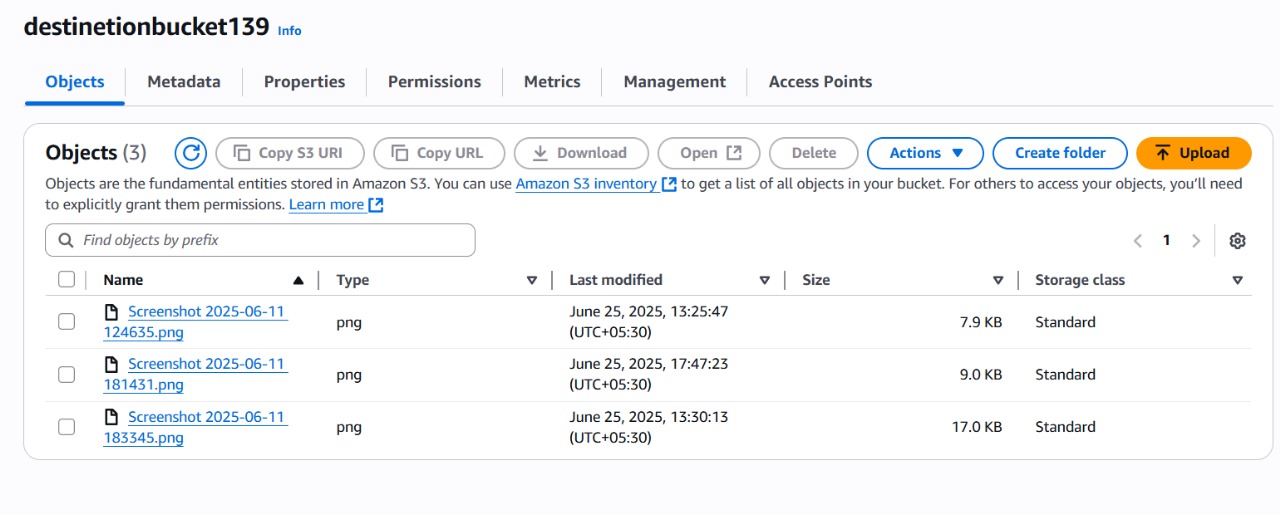
**TASK 6:-** Testing the image size resizing application.

1. Go to the source bucket and upload an image to the source bucket.
2. Now, go to the destination bucket and check whether the resized image is replicated in the destination bucket.

**RESULT**

**THE SERVERLESS IMAGE PROCESSING APPLICATION HAS BEEN CREATED.**

****

****