

# SERVERLESS IMAGE PROCESSING

Discover the power of AWS serverless image processing, an innovative solution that improves efficiency, scalability, and scalability, and user satisfaction. With serverless image processing, you'll be able to process images faster and more and more reliably than ever before.

### Table of Contents

1. Our Aim
2. Architecture Overview
3. Step-by-Step Guide
   1. Setting Up the S3 Buckets
   2. Creating an IAM Role
   3. Writing the Lambda Function
   4. Configuring S3 to Trigger Lambda
   5. Deploying the Lambda Function
   6. Testing the Setup
4. How it works
5. The Benefits of Serverless Image Process
6. Examples of image process use cases
7. ImprovedUserSatisfaction
8. Conclusion



# OUR AIM

Our aim was to create a serverless solution that automates image processing tasks, ensuring top-tier

top-tier performance. Our solution offers automatic resizing, compression, and format conversion for user- conversion for user-uploaded images. By adopting serverless architecture, we achieved scalability, cost- scalability, cost-effectiveness, and streamlined development.

### Architecture Overview

* The architecture involves the following key components:
* **Amazon S3**: Two buckets are used; one for uploading the original images and the other for storing the processed images.
* **AWS Lambda**: A function triggered by the S3 bucket to process the images.
* **AWS IAM**: A role with permissions for S3 and Lambda services.

+ + + + + +

| Upload Bucket |-----> | Lambda |-----> | Output Bucket |

+ + + + + +

**Step-by-Step Guide**

##### Setting Up the S3 Buckets Create the Input Bucket:

* Go to the S3 console.
* Click on "Create bucket".
* Name the bucket (e.g., image-upload-bucket).
* Select a region and create the bucket.

##### Create the Output Bucket:

* Repeat the above steps to create another bucket for processed images (e.g., image-output-bucket).

##### Creating an IAM Role Create the Role:

* Go to the IAM console.
* Click on "Roles" > "Create role".
* Select "Lambda" as the trusted entity.
* Attach the necessary policies: AmazonS3FullAccess and AWSLambdaBasicExecutionRole.
* Name the role (e.g., lambda-s3-image-processing-role) and create it.

##### Writing the Lambda Function

**Create the Lambda Function:**

* Go to the Lambda console.
* Click on "Create function".
* Choose "Author from scratch".
* Provide a name (e.g., imageProcessingFunction), runtime (e.g., Python 3.9), and choose the IAM role created earlier.

**Configuring S3 to Trigger Lambda**

##### Configure S3 Event Notification:

* + - Go to the input bucket.
    - Click on "Properties" > "Event notifications" > "Create event notification".
    - Configure to trigger on ObjectCreated events, and specify the Lambda function.

##### Deploying the Lambda Function

* + The Lambda function is now deployed and ready to process images. Ensure the input bucket has permissions to invoke the Lambda function.

##### Testing the Setup

* + **Upload an Image:**
    - Upload an image to the input bucket using the AWS S3 console or AWS CLI.
  + **Check the Output Bucket:**
    - Verify that the resized image appears in the output bucket.

##### Code Explanation

* + **Importing Libraries**: boto3 for interacting with AWS services, PIL from Pillow for image processing.
  + **Lambda Handler**: Handles S3 events, fetches the image, resizes it, and uploads the processed image to the output bucket.
  + **Image Processing**: Resizes the image to a fixed size of 300x300 pixels. This can be adjusted as needed.

## How it Works



#### User Uploads Image

When a user uploads an image, AWS

Lambda is triggered.

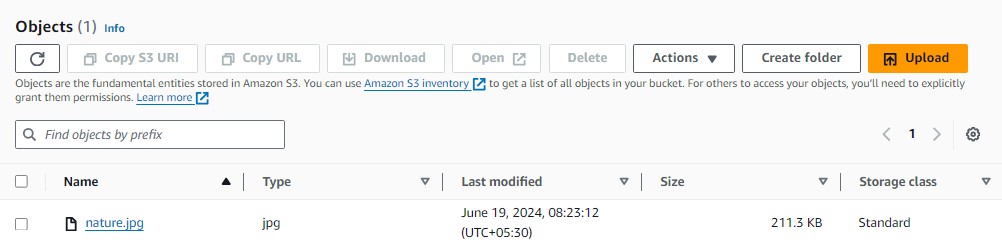
#### Lambda Function Processes Image

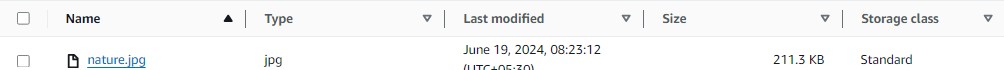
Our serverless function processes processes the image and stores the the result in an S3 bucket.

#### Processed Images Stored in Stored in S3

Once processed, the images are are stored in an S3 bucket for easy easy retrieval and distribution.

**Before Processing**



**After Processing**

## The Benefits of Serverless Image Processing

#### Scalability

**1**

Thanks to the auto-scaling features of AWS, businesses can handle any workload needed for image processing, no matter how big or small.

#### Reliability

Our solution is built on AWS Lambda, a fully managed and resilient compute service.

**2**

#### Flexibility

Being serverless means our solution can handle any workload without worrying about server settings.

**3**

## Examples of Image Processing Use Cases

#### eCommerce

Enhance product images by adjusting colors, brightness and contrast in order to highlight product details and improve marketing strategy.

#### Media and Entertainment

Use image processing to help recognize faces, objects, and scenes, which can enable dynamic features and enhanced user engagement.

#### Healthcare

Image processing can help in medical diagnoses by analyses of medical images such as X-rays and CT scans, detecting specific medical disorders, tumors or other abnormal changes.

## Improved User Satisfaction

#### Better User Experience

**1**

**2**

**3**



**Improved Customer Loyalty**

Our solution can help you build lasting relationships with customers by improving engagement with your brand and boosting

customer satisfaction.

Our serverless image processing solution helps reduce downtime and improve overall user satisfaction, keeping your customers happy and engaged.

#### Increased Productivity

With our solution, businesses can focus on core activities rather than worrying about image processing, ultimately increasing

productivity.



## Conclusion and Key Takeaways

* Serverless image processing is a cost-effective solution that can handle any workload, no matter how big or small.
* It can improve user satisfaction and drive customer loyalty while reducing downtime and improving overall efficiency.
* With image processing, businesses can create more engaging content and increase productivity, keeping pace with today's digital age.