* **Server less image processing**
* (Simple Storage Service) is a cloud data storage service. It is one of the most popular services of AWS. It has high scalability, availability, security and is cost effective. S3 has different storage tiers depending on the use case. Some common use cases of AWS S3 are
* **Storage:** It can be used for storing large amounts of data.
* **Backup and Archive:** S3 has different storage tiers based on howfrequent the datais accessed which can be used to backup critical data at low costs.
* **Static website**: S3 offers static website hosting through HTML files stored in S3.
* **Data lakes and big data analytic:** Companies can useAWS S3 as a data lake and then run analytic on it for getting business insights and take critical decisions
* **AWS Lambda**

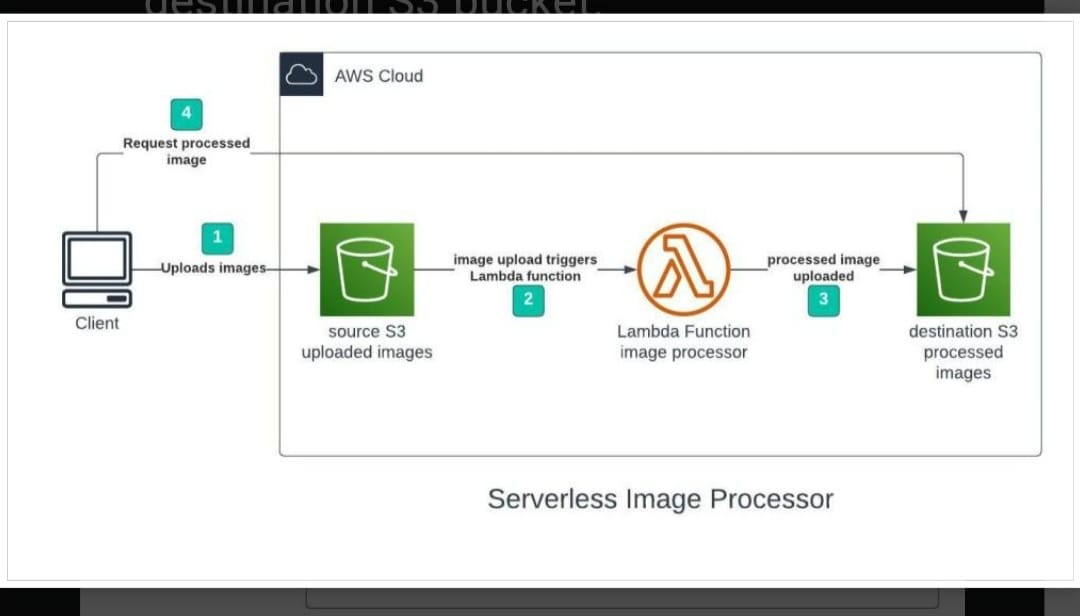
AWS Lambda is a server less, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. Lambda functions run on demand i.e. they execute only when needed and you pay only for what you compute. Lambda is well integrated with may other AWS services. It supports a wide variety of programming languages.

**File processing**: You can use Lambda for processing files as they are uploaded in an S3 bucket or whenever some event triggers the function.

**Data and analytics**: You can pass a data stream to your Lambda function and then create analysis from that.

**Website:** Lambda can also be used for creating websites. This is cost effective because you are charged only for the time when the servers are running

* **Server less Image Processing Flow**

1. User uploads a file to the source S3 bucket (which is used for storing uploaded images).
2. When the image is uploaded to a source S3 bucket, it triggers an event which invokes the Lambda function. The lambda function processes the image.
3. Processed image is stored in the destination S3 bucket.
4. The processed image is requested by the user

* **Step 1: Set up an S3 Bucket**
* Go to AWS Console → Search for S3
* In the AWS console, create two buckets (e.g., my-image-uploads and my-image-processed) in your region.
* Keep other options as default, but uncheck "Block all public access" (so we can access the image if needed)
* **Step 2: Create a Lambda Function**
* Go to AWS Console → Search for Lambda.
* Click Create Function.
* Choose Author from scratch:
* Name: resizeImageFunction
* Runtime: Python 3.12
* Role: Select Create a new role with basic Lambda permissions.
* Click Create Function.
* **Step 3: Add Required Python Code to Lambda**
* Click the function → scroll down to Code → paste the code below:

s3 = boto3.client('s3')

def lambda\_handler(event, context):

# Get the image info from the S3 event

bucket = event['Records'][0]['s3']['bucket']['name']

key = event['Records'][0]['s3']['object']['key']

# Download the image

response = s3.get\_object(Bucket=bucket, Key=key)

image\_content = response['Body'].read()

# Resize the image

with Image.open(io.BytesIO(image\_content)) as image:

image = image.resize((300, 300)) # Resize to 300x300

buffer = io.BytesIO()

image.save(buffer, format='JPEG')

buffer.seek(0)

# Upload the resized image to a new location

new\_key = "resized/" + key

s3.put\_object(Bucket=bucket, Key=new\_key, Body=buffer, ContentType='image/jpeg')

return {

'statusCode': 200,

'body': 'Image resized and uploaded!'

}

* **Step 4: Add Permissions to Lambda Role**
* This step allows your function to access S3.
* Go to IAM → Roles.
* Find the role starting with lambda-role-resizeImageFunction.
* Click Attach Policies.
* Search for and attach: AmazonS3FullAccess.
* **Step 5: Set up an S3 Trigger for Lambda**
* Go back to S3 → Open your bucket.
* Go to Properties → Scroll to Event notifications → Click Create event notification.
* Name it ResizeTrigger.
* Event type: PUT (i.e., when someone uploads a file).
* Prefix: uploads/ (this means only images uploaded in uploads/ folder will trigger it).
* Destination: Lambda function → Choose your function (resizeImageFunction).
* **Step 6: Test It?**
* Go to your S3 bucket.
* Create a folder called uploads.
* upload a .jpg image inside the uploads/ folder.
* After a few seconds, check the resized/ folder — the resized image should appear.