

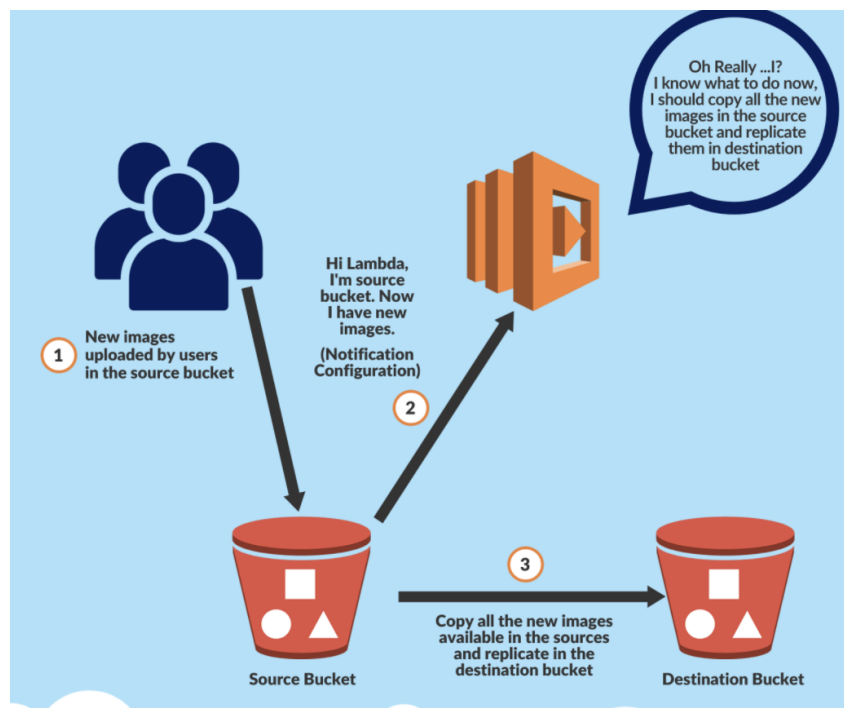
Project 4: Creating an AWS Lambda Function

What is AWS Lambda?

- Lambda is a compute that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance.

Objective for this project:

- We will be creating a sample Lambda function to be triggered on an S3 Object Upload Event. The lambda function will make a copy of that object and place it in a different S3 bucket.



Step 1: Create two new S3 buckets

- We start with creating two new S3 buckets: A Source Bucket and A Destination Bucket leaving all permissions at default:
 - ARNS:
 - Arn:aws:s3:::newsourcedbucket12321
 - Arn:aws:s3:::newdestinationbucker12321
-

<input checked="" type="radio"/>	newsourcebucket12321	US East (N. Virginia) us-east-1
<input type="radio"/>	newdestinationbucker12321	US East (N. Virginia) us-east-1

Step 2: Create an IAM Policy

- As a prerequisite to creating the Lambda function, we need to create a user role with a custom IAM policy written in the JSON format.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::newsourcebucket12321/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::newdestinationbucker12321/*"
      ]
    }
  ]
}
```

As a result, an IAM policy with the name mypolicy is created.

Filter policies ▾		Q myp	
	Policy name ▾	Type	Used as
<input type="radio"/>	mypolicy	Customer managed	None

Step 3: Create an IAM Role

Along with an IAM policy, it is required that we create an IAM role as well. An IAM role is intended to be assumable by anyone who needs it.

	Role name ▾	Trusted entities	Last activity ▾
<input type="checkbox"/>	myrole1	AWS service: lambda	None


Step 4: Creating a Lambda function


- Once we have our buckets, and our IAM policies and roles, we then create a Lambda Function and write the necessary code in it to trigger a Lambda Function. Lambda runs your code in response to events and automatically manages the underlying compute resources for you.

mylambdafunction

ThrottleCopy ARNActions ▾

▼ Function overview Info

mylambdafunction

Layers (0)

+ Add trigger

+ Add destination


Description

-

Last modified

34 seconds ago

Function ARN

arn:aws:lambda:us-east-1:308456800461:function:mylambdafunction

```
1 var AWS = require("aws-sdk");
2 exports.handler = (event, context, callback) => {
3   var s3 = new AWS.S3();
4   var sourceBucket = "mysourcebucket12345";
5   var destinationBucket = "mydestinationbucket12345";
6   var objectKey = event.Records[0].s3.object.key;
7   var copySource = encodeURIComponent(sourceBucket + "/" + objectKey);
8   var copyParams = { Bucket: destinationBucket, CopySource: copySource,
9
10    s3.copyObject(copyParams, function(err, data) {
11      if (err) {
12        console.log(err, err.stack);
13      } else {
14        console.log("S3 object copy successful.");
15      }
16    });
17 };
```

With the code all written on the source section, we then deploy the code.

Step 5: Adding Triggers to the Lambda Function

One last step before testing. We add in a trigger to the Lambda function to allow the Lambda to execute when uploading an object to an S3 bucket.

Triggers (1)

Enable **Disable** **Fix errors** **Delete** **Add trigger**

Find triggers

< 1 >


<input type="checkbox"/>	Trigger
<input type="checkbox"/>	S3: newsourcebucket12321 arn:aws:s3:::newsourcebucket12321 Details

Step 6: Test Lambda Function

We then test to see if the Lambda function is by uploading an image to the S3 Bucket. We place it in the source bucket while the Lambda is supposed to allow it to upload in the destination bucket.

Files and folders (1 Total, 132.7 KB)						
<input type="text" value="Find by name"/>						
< 1 >						
Name ▲	Folder ▼	Type ▼	Size ▼	Status ▼	Error ▼	
Your strenghts.jpg	-	image/jpeg	132.7 KB	✔ Succeeded	-	

We then check and see in the destination bucket that the file was uploaded. And voila. The Lambda function had been triggered.

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	 Your strenghts.jpg	jpg	June 25, 2021, 07:52:20 (UTC-07:00)	132.7 KB	Standard