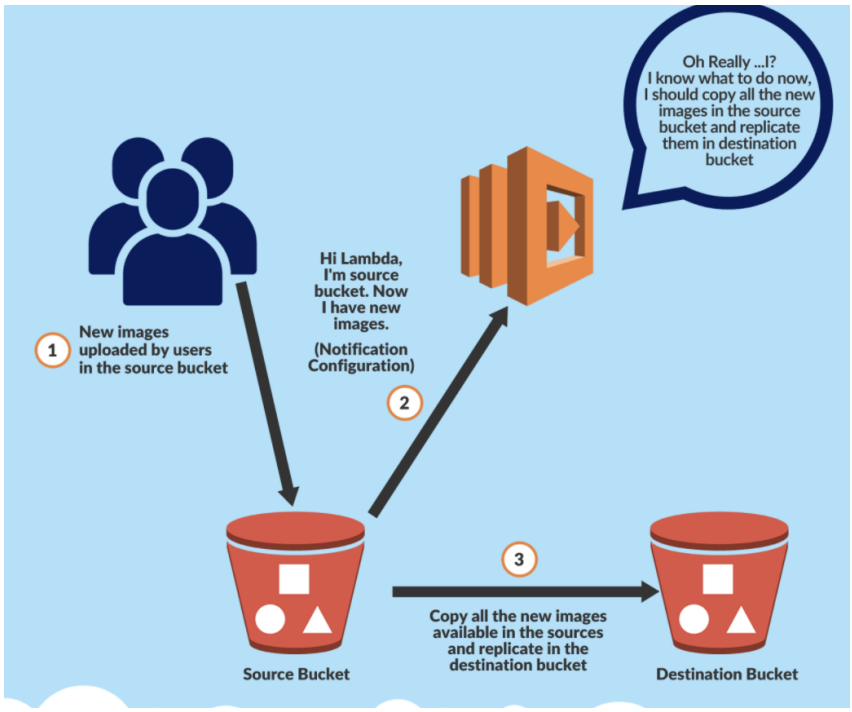
Project 4: Creating an AWS Lambda Function

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:

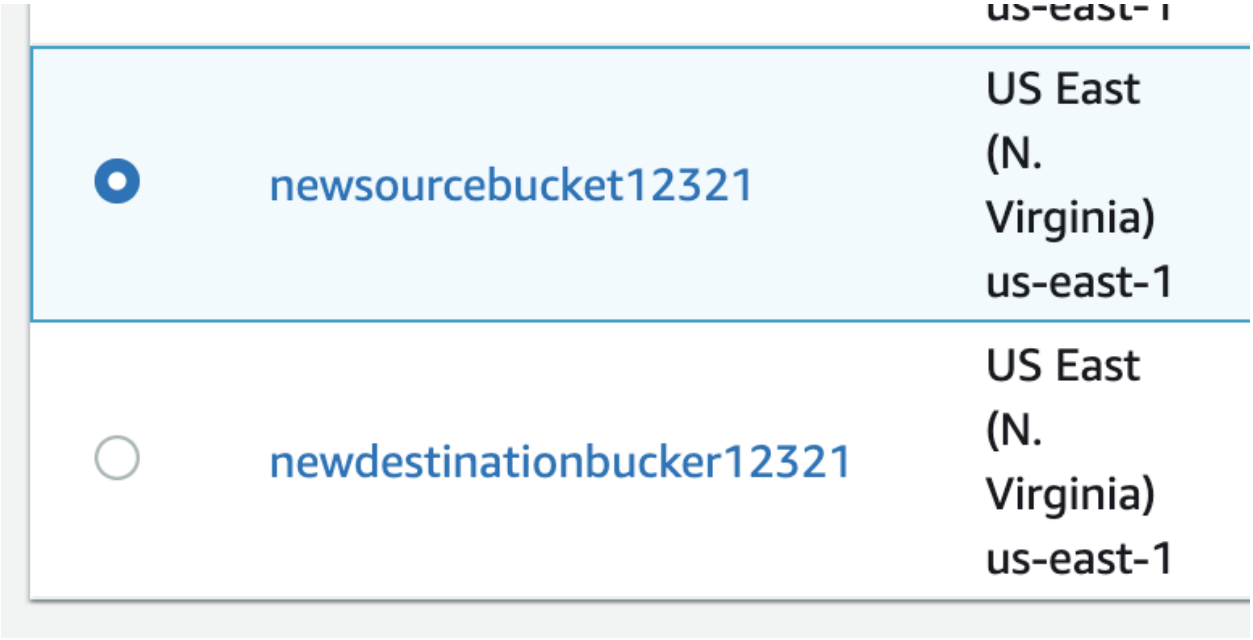
- We will be creating a sample Lambda function to be triggered on an S3 Object Upload Event. The lambda function will make a copy of that object and place it in a different S3 bucket.



Step 1: Create two new S3 buckets

- We start with creating two new S3 buckets: A Source Bucket and A Destinatioln Bucket leaving all permissions at default:
    - ARNS:
        - Arn:aws:s3:::newsourcebucket12321
        - Arn:aws:s3:::newdestinationbucker12321

-



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/*"
         ]
      }
   ]
}
```

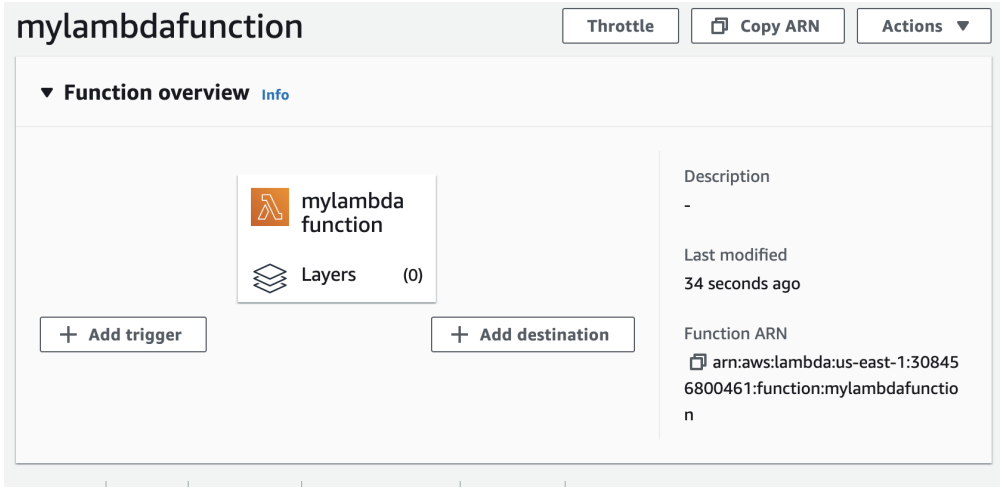
As a result, an IAM policy with the name mypolicy is created.

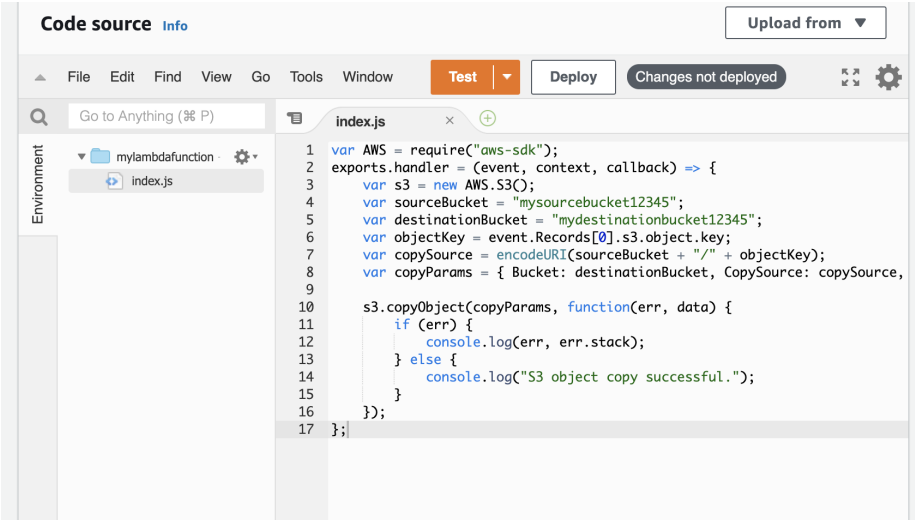
Step 3: Create an IAM Role

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.

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.

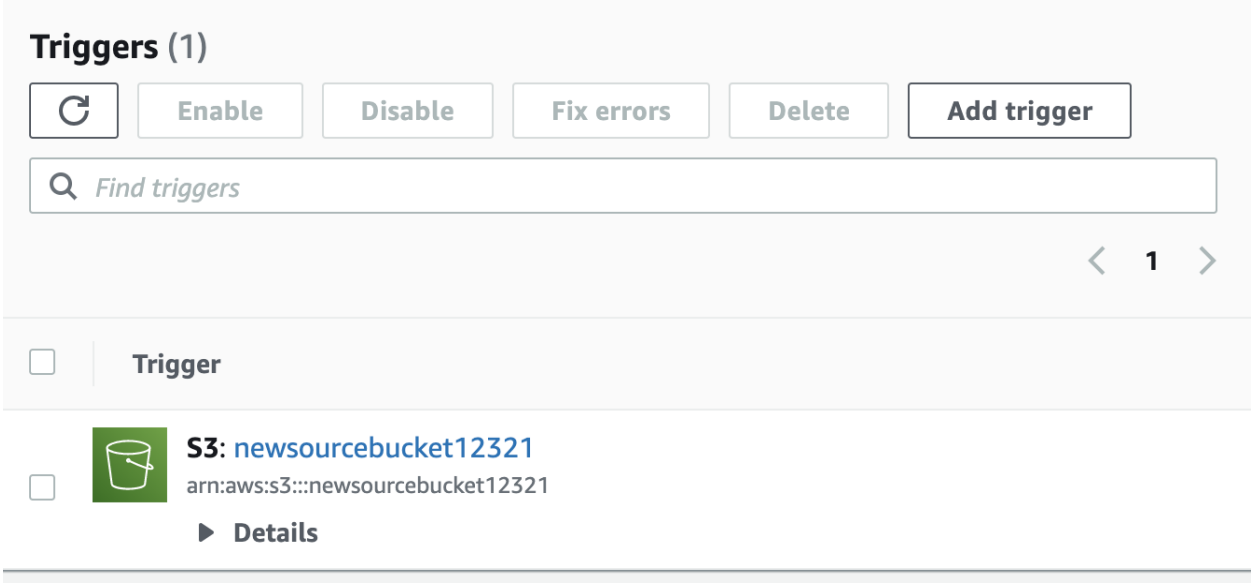




With the code all written on the source section, we then deploy the code.

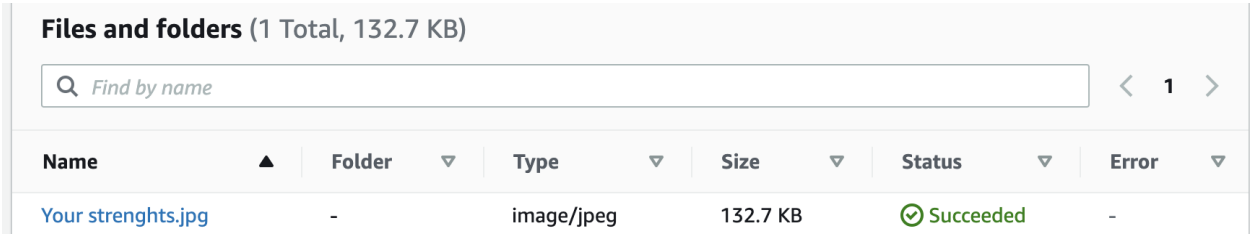
Step 5: Adding Triggers to the Lambda Function

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.



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.



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