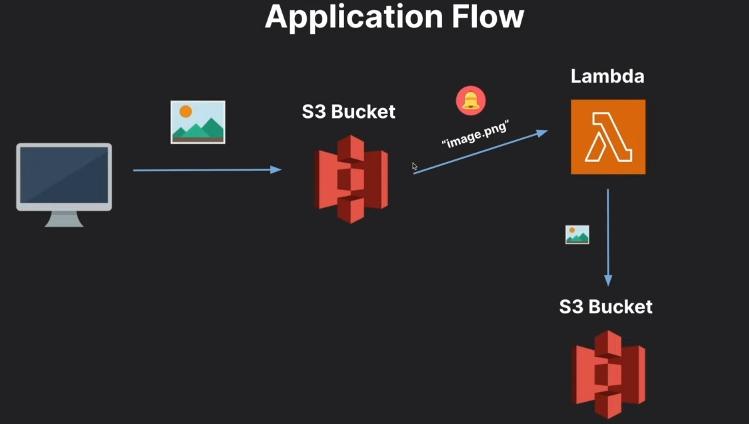
**Serverless Image Processing**

To create a serverless image processing application that automatically resizes and optimizes images uploaded to an Amazon S3 bucket, you can use AWS Lambda, Amazon S3, and Amazon CloudFront. Here's a high-level overview of the steps:

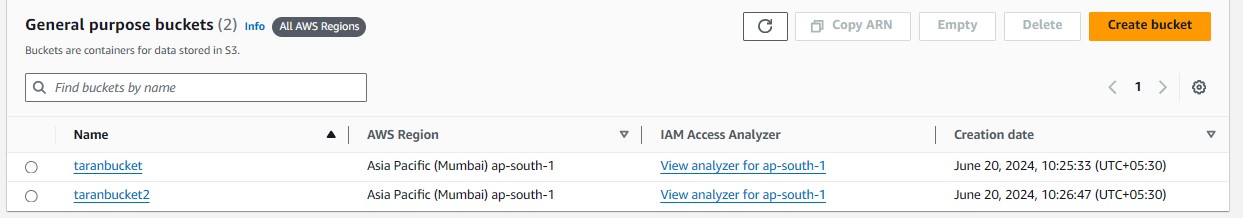


## Step 1 – Creating S3 buckets

### We will use two S3 buckets:

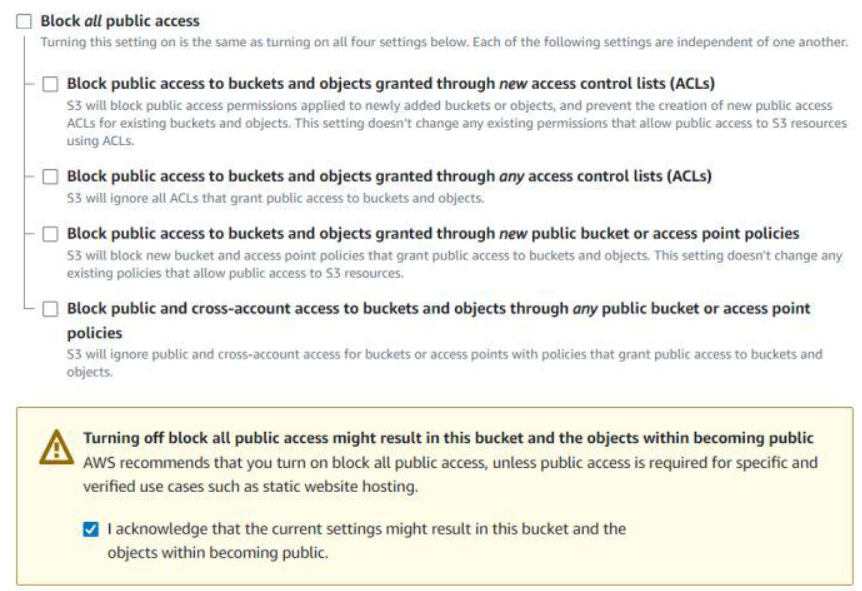
1. source Bucket: For storing uploaded images.

2. destination Bucket: For storing processed images.



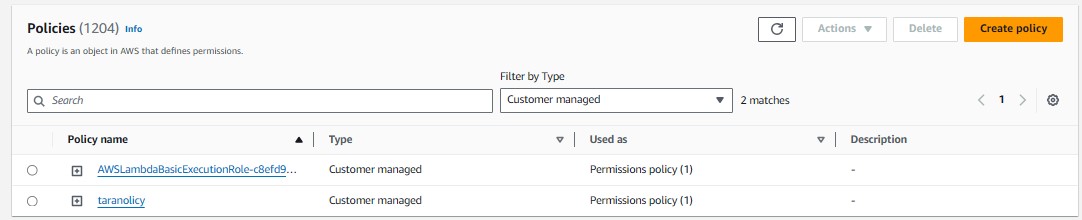
**Step 2 – Configuring S3 bucket policy**

**In ‘Block Public Access settings for this bucket’ section disable “block all public access”.**

****

**Step 3 – Creating police in Iam**

**Go to AWS I am console. Navigate to policies section. Click Create policies in (JSON) and name it “taranpolicy ”. Leave all other settings as default. Create the policy.**

****

**Policy:**

{

"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:::BUCKET\_NAME/\*"

},

{

"Effect": "Allow",

"Action": ["s3:PutObject"],

"Resource": "arn:aws:s3:::DEST\_BUCKET/\*"

}

]

}

**\*BUCKET\_NAME = SOURSE BUCKET NAME(“taranbucket”)**

**\*DEST\_BUCKET = DESTINATION BUCKET NAME(“taranbucket2”)**

**Step 4 –creating role in I am**

**Following Steps are Follows**

\*Go to aws I am console

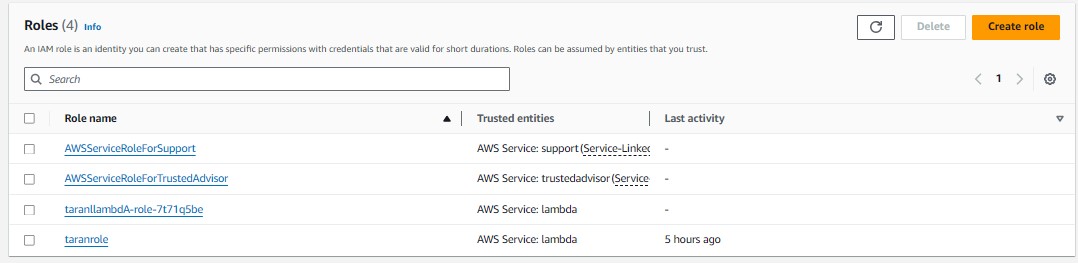
\*Create role

\*name taranrole

\*Use case – Lambda

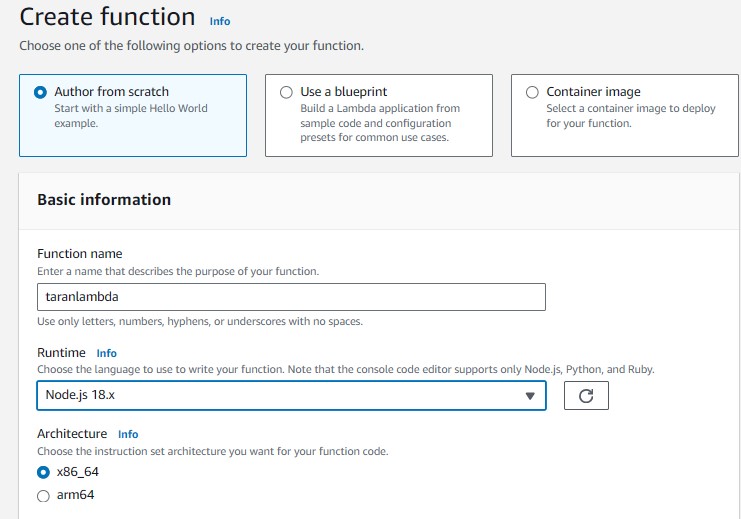
\*Select--taranpolicy

\*Then create role

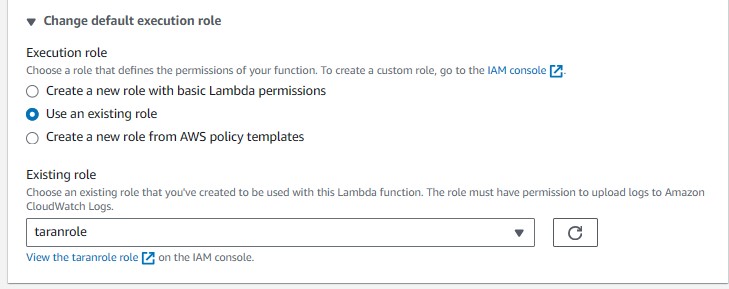
****

**Step 3 – Creating Lambda function**

Go to AWS Lambda console. Navigate to Functions section. Click Create Function and name it “ImageProcessing”. Select runtime as “NodeJS 18.x” and architecture as “x86\_64”. Leave all other settings as default. Create the function.

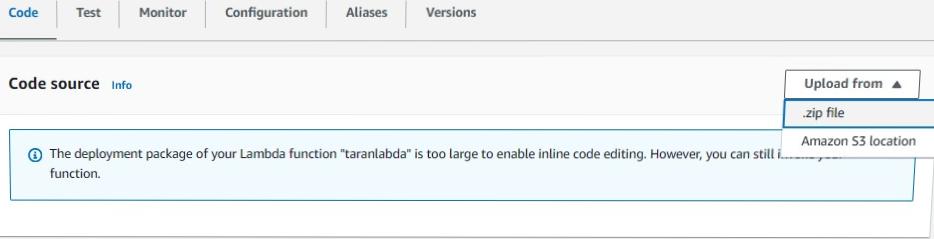


**\*Change default execution role**



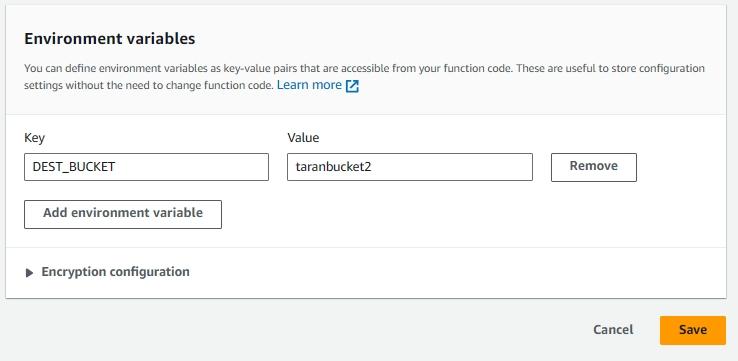
**\*Create function**

**Step 6 -upload zip file in Lambda function**

****

\*Zip file link-https://github.com/OneLightWebDev/image-resizer-lambda

**Step 7** - Edit environment variables



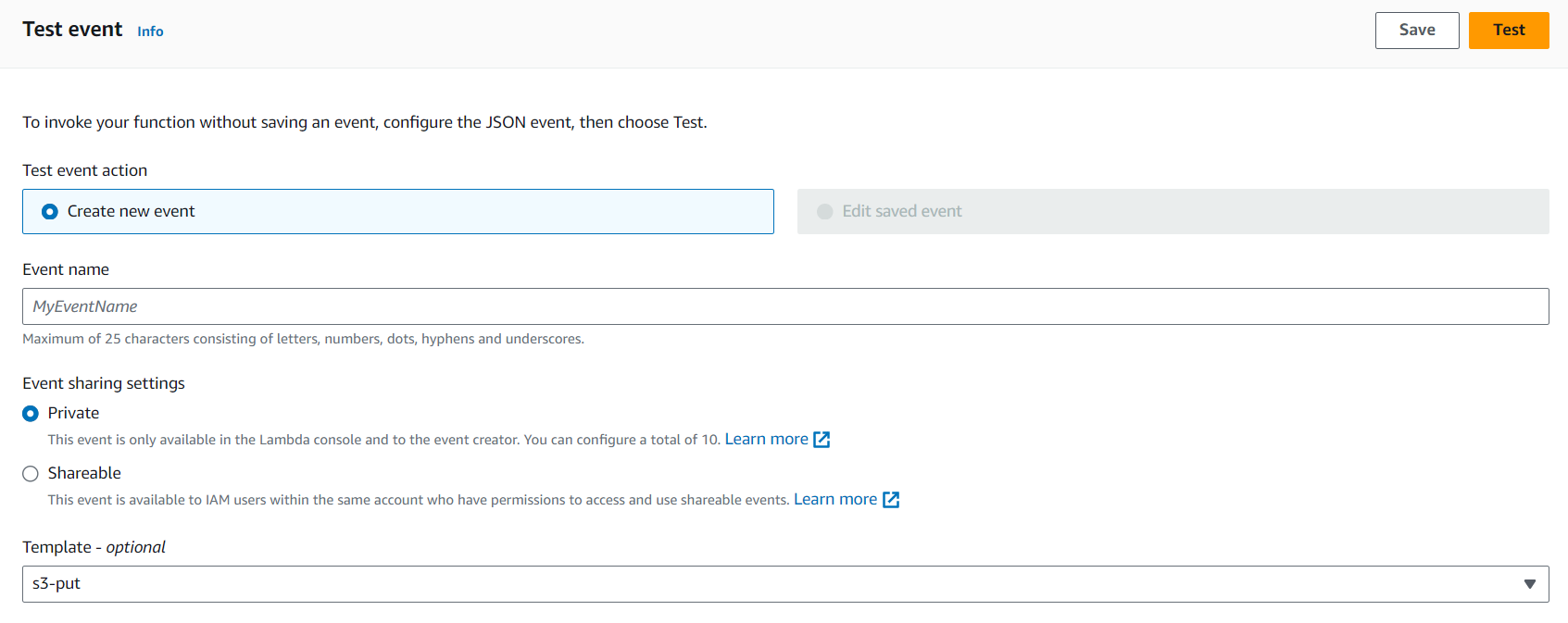
**Step 8 – Test Lambda Function**

\*Go to AWS Lambda console. Navigate to Functions section.

\*open function then will be created

\*open test console

\*template=s3-put

**EVENT JSON**

In event json we can change only 3 value

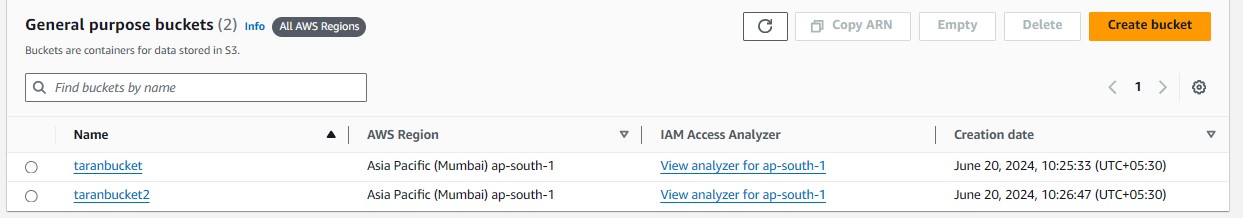
\*name, arn, key

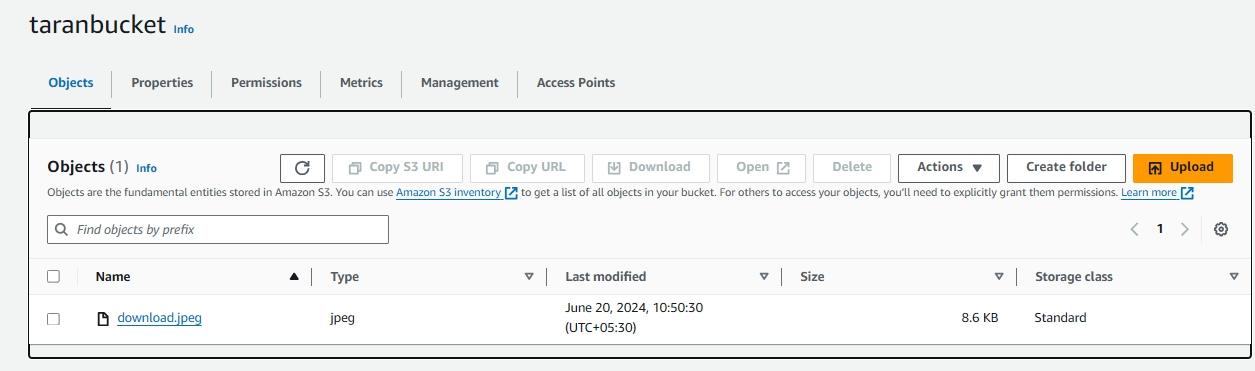


"name":"taranbucket",->

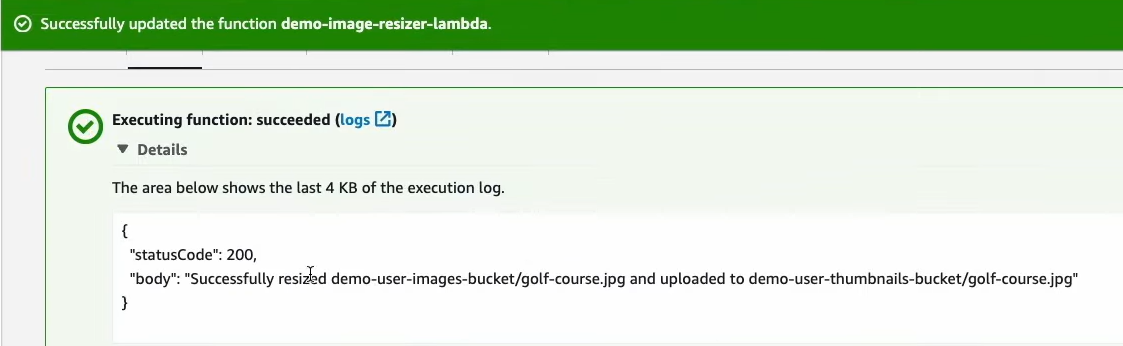
"arn": "arn:aws:s3::: taranbucket "->

"key": "download. jpeg",->





**Now we can test**



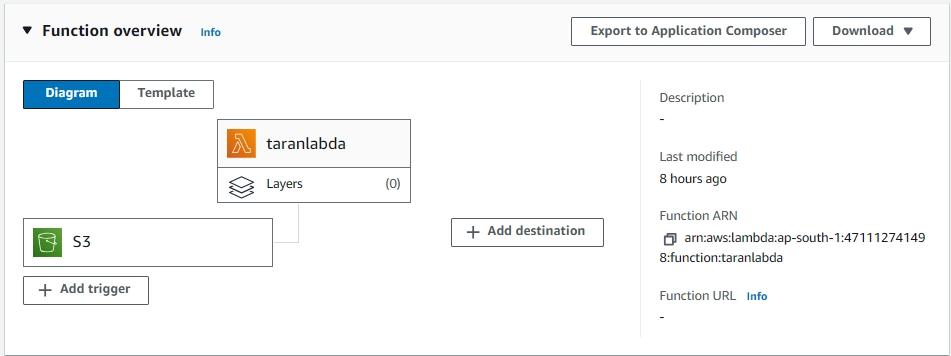
**Step 5 – Creating S3 trigger**

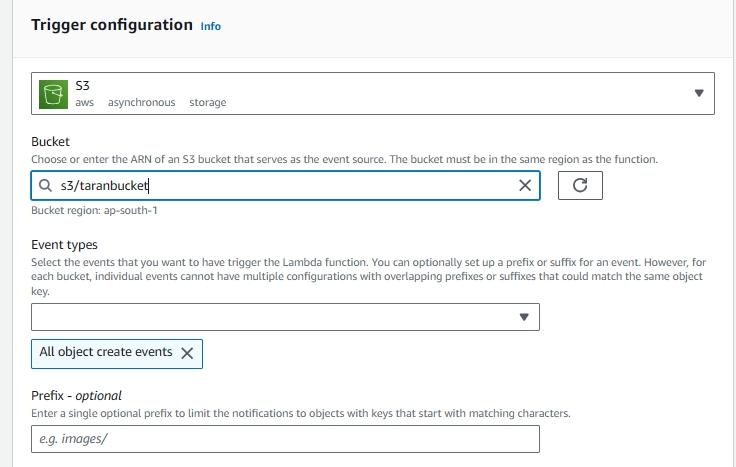
\*Add trigger

\*Select s3

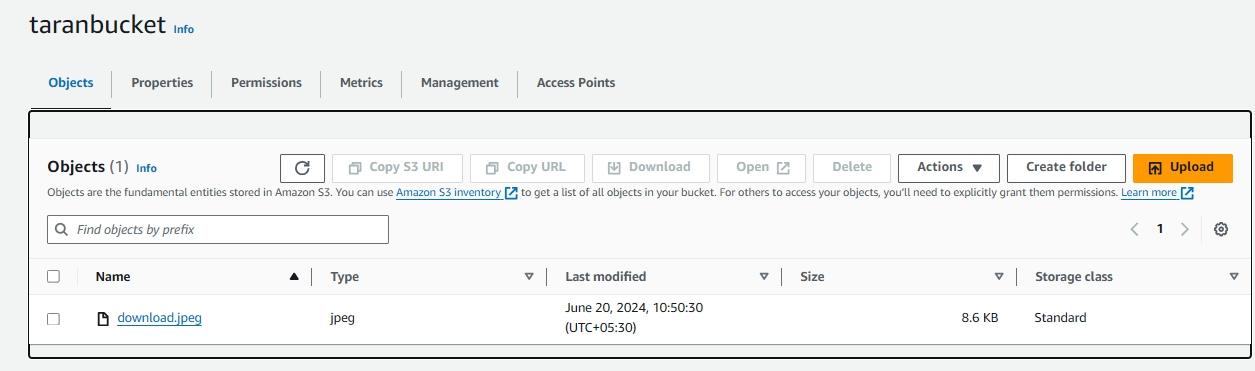
\*choose source Bucket name

\*Now Add





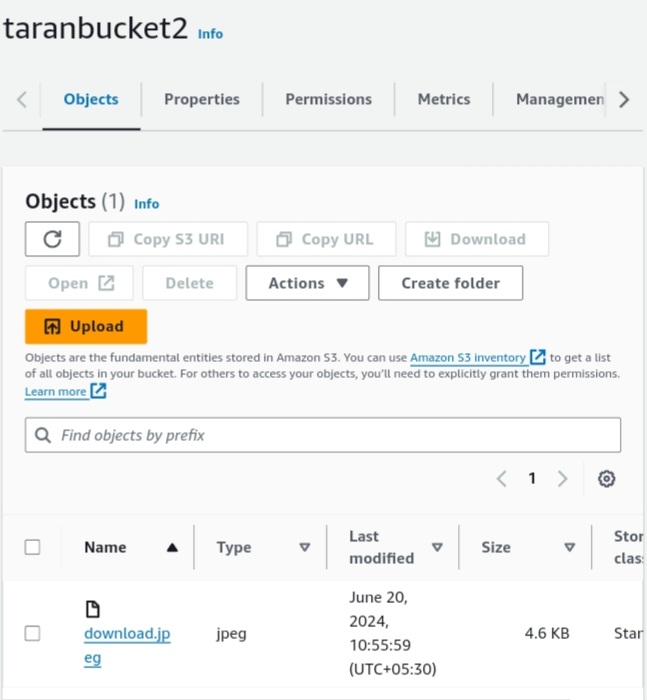
**Upload image in source Bucket**



**Original image**



**Destination Bucket**



**Resize Image**

