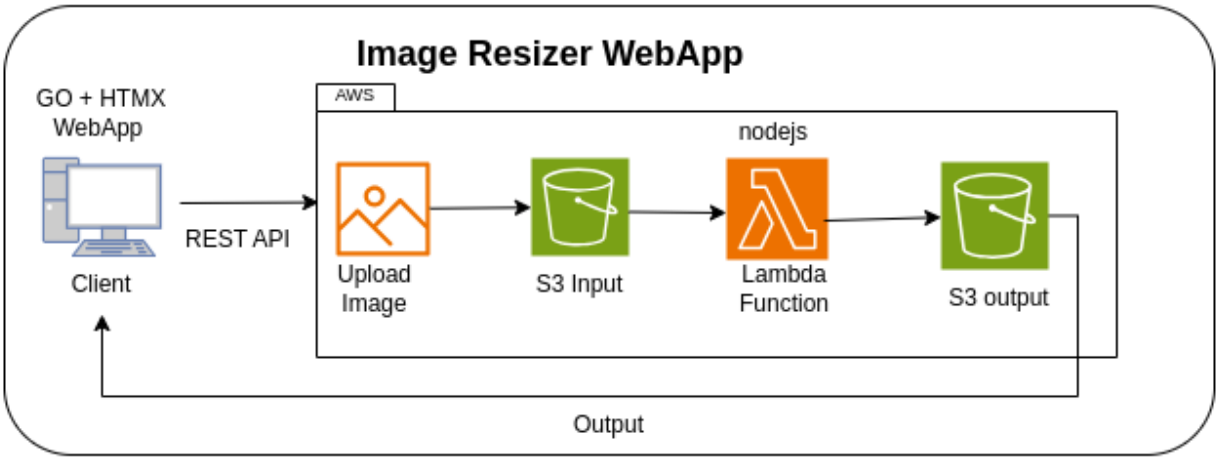
TITLE: Image Resize SAAS Lambda Project.

Use Case: when filling forms many times the user needs photos of certain width and height, for that use case this project can be used.
Working: when user uploads image it gets stored in input S3 bucket then AWS lambda function triggers and convert image to appropriate width and size and output stored into another S3 bucket and can be download as required.
Beneficiaries: Students, Working Professionals etc.

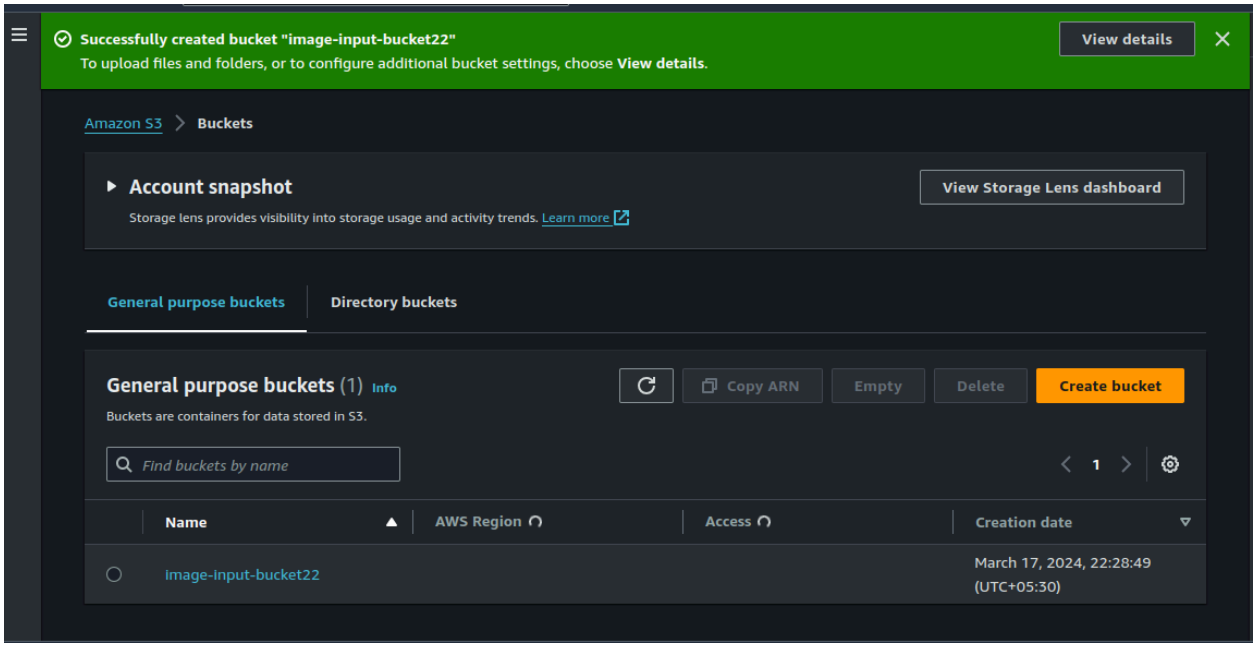
Technology Used:
Client Side: Go lang , Gin framework, HTMX.
Backend : AWS S3 , AWS Lambda, AWS IAM, nodejs, sharp library.

Architecture Diagram:

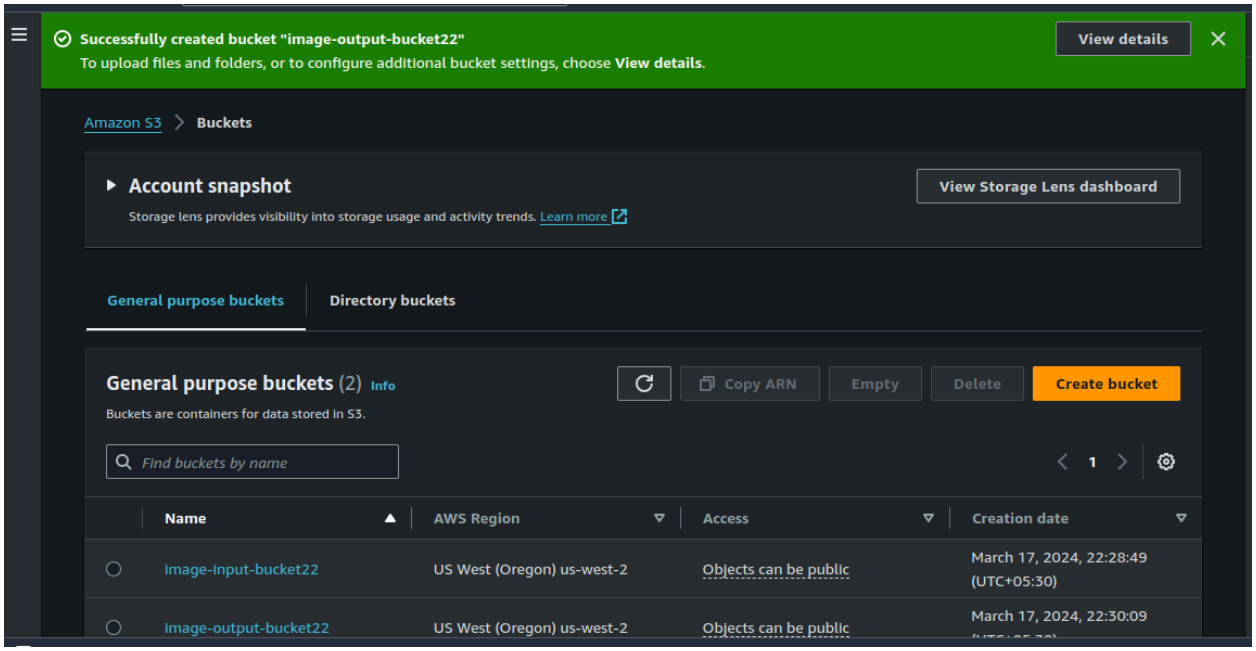


In the above diagram client send the image to the input S3 Bucket,then the lambda function evoked and then it process the image and then save it to another bucket where user can take and utilized the resized image.
Steps to perform : 1. Create a 2 s3 bucket with the appropriate names as shown in the images, i have given *image-input-bucket22* and *image-output-bucket22*.

Input Bucket:



Output S3 Bucket:



Upload image into input Bucket for Testing

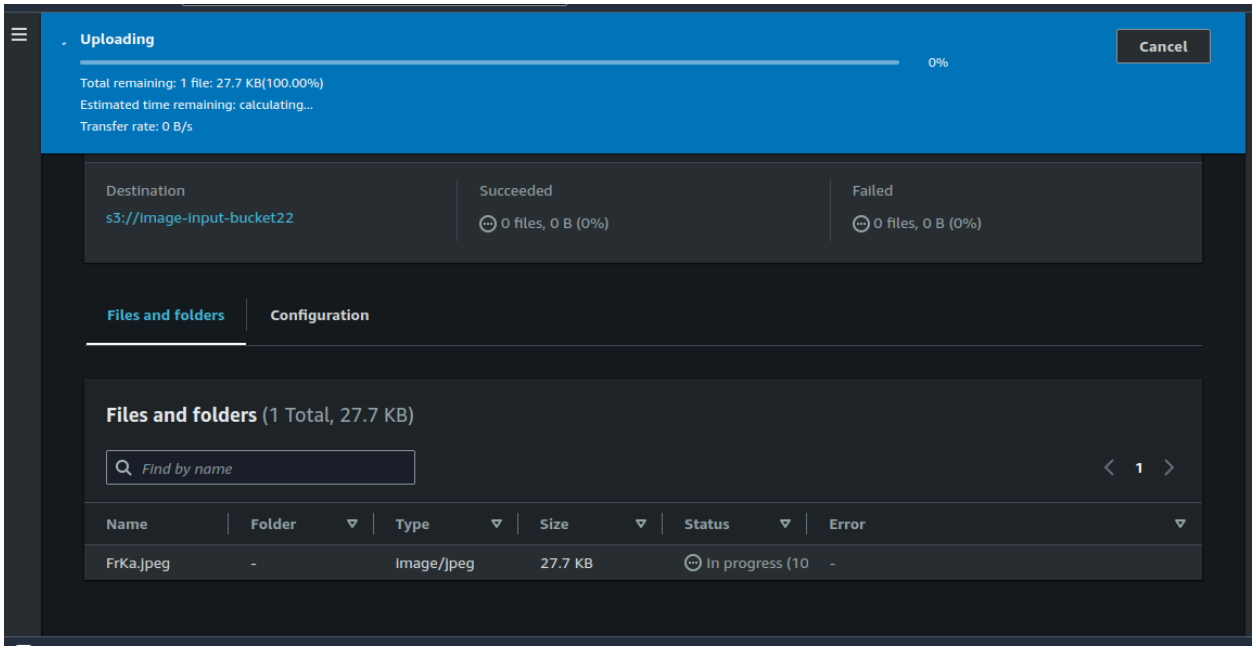
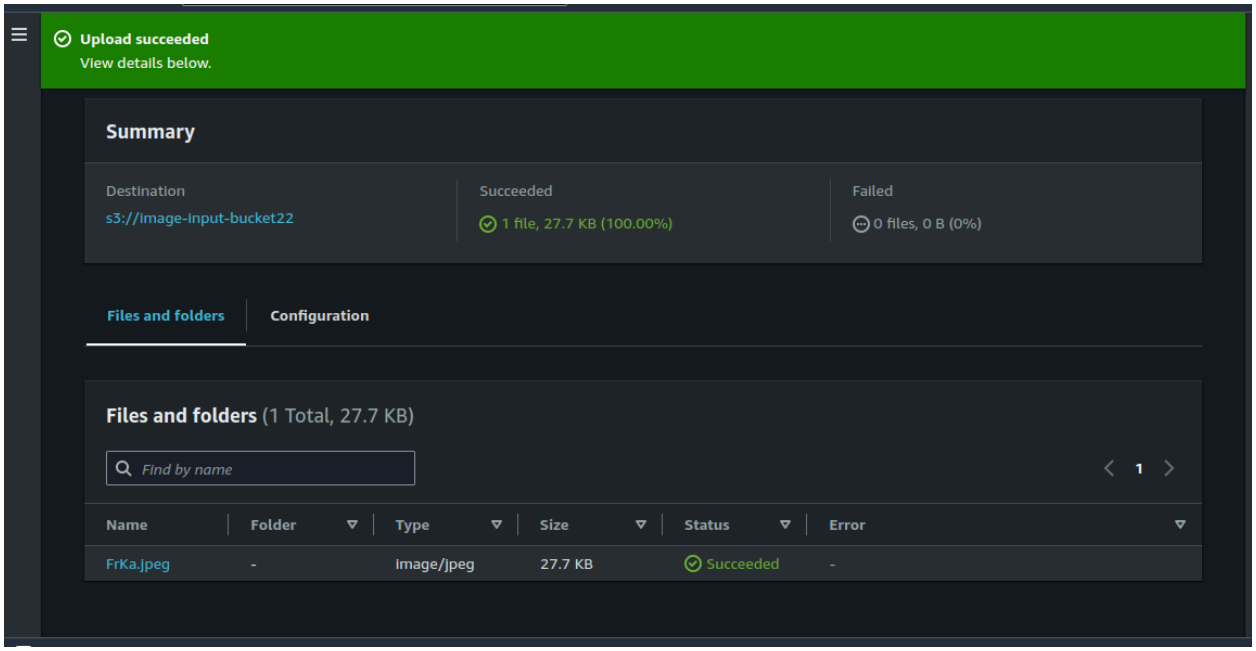
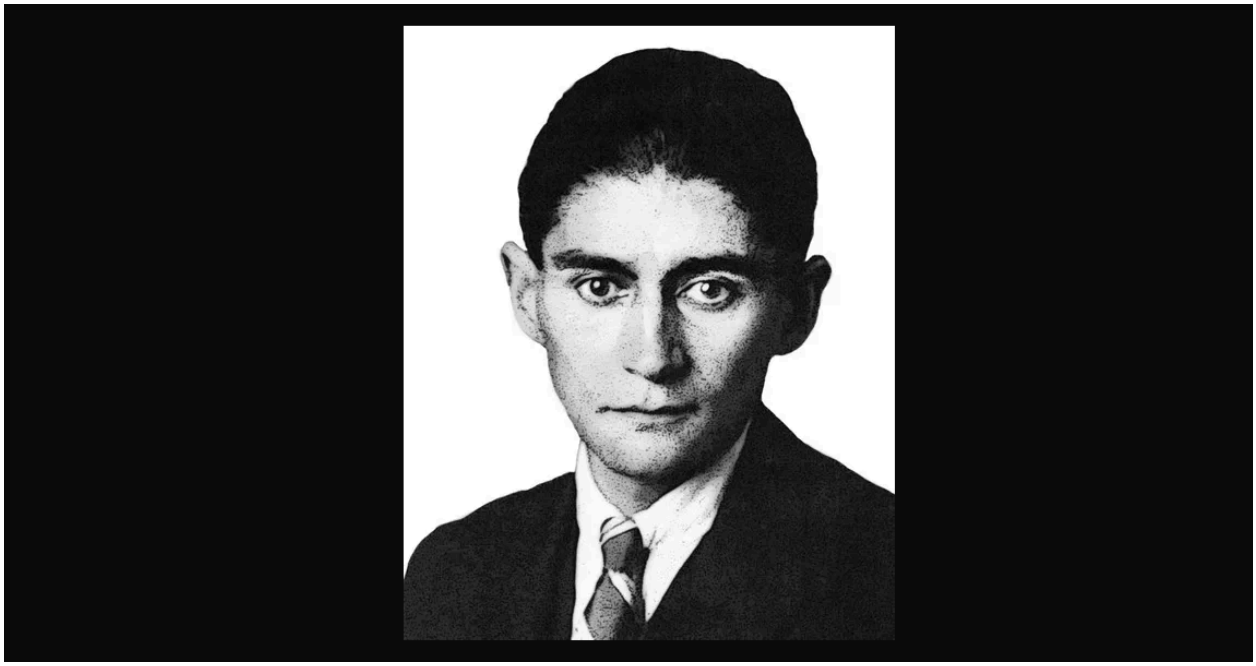


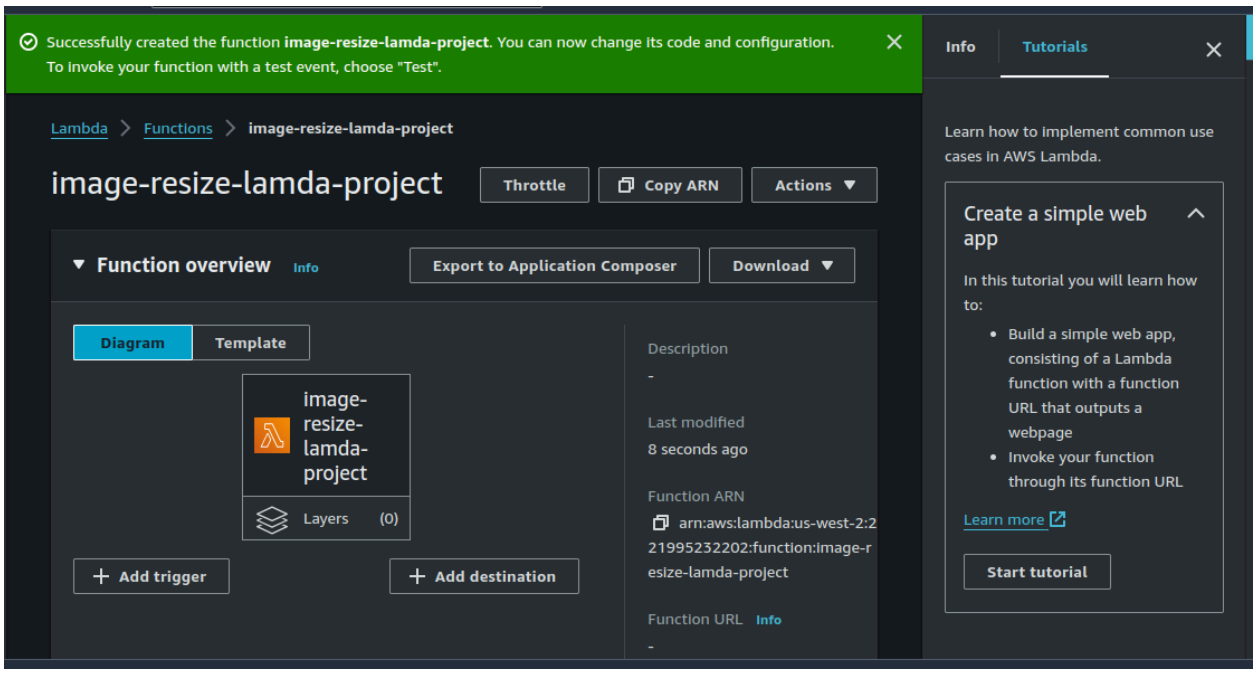
Image upload successfully:



Open original image to see the dimensions of the image:



Create a AWS Lambda Function



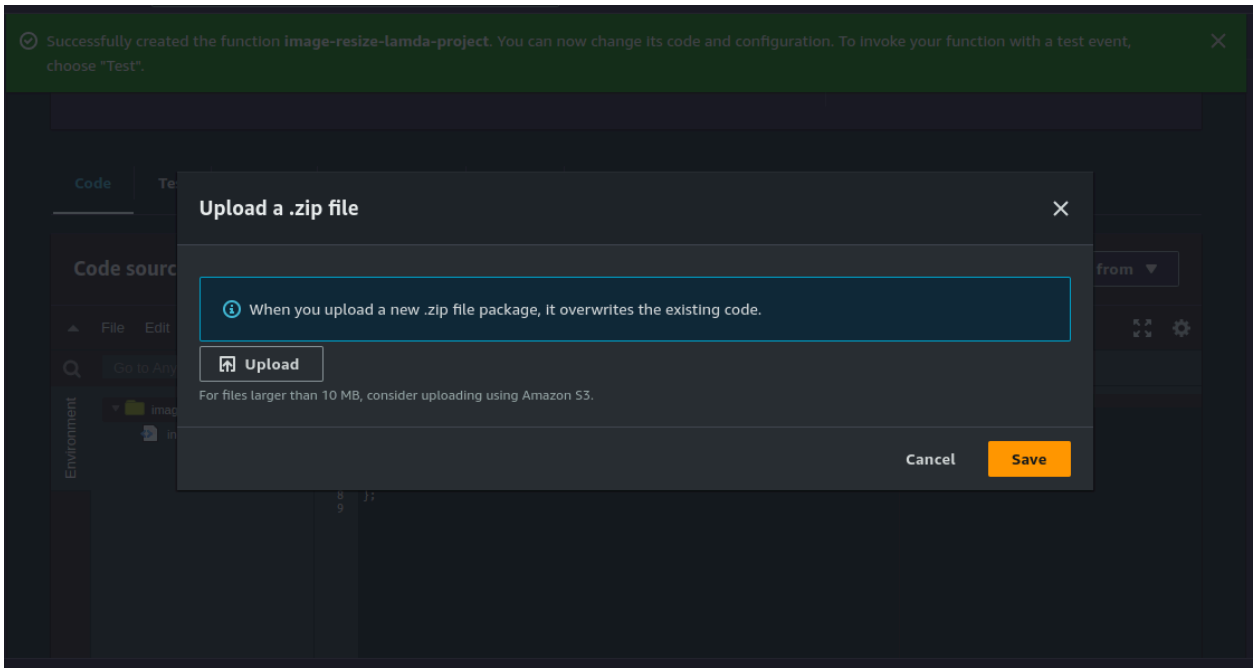
Note while creating Lambda Function it requires permission to access S3 bucket, you can use IAM to create a new S3 Policy.

Sample S3 policy:

```
{
  "Version": "1.0.0",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogGroup",
        "logs:CreateLogStream"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": "arn:aws:s3:::image-input-bucket22/*"
    },
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": "arn:aws:s3:::image-output-bucket22/*"
    }
  ]
}
```

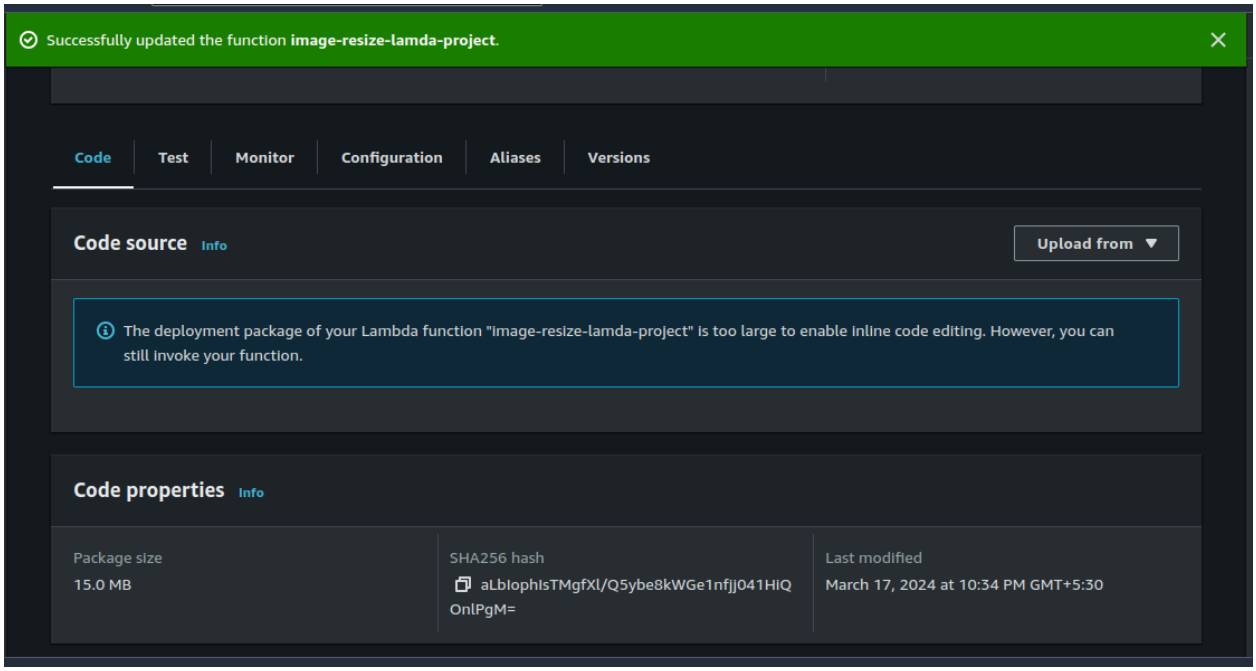
Upload a zip package to AWS Lambda code section.



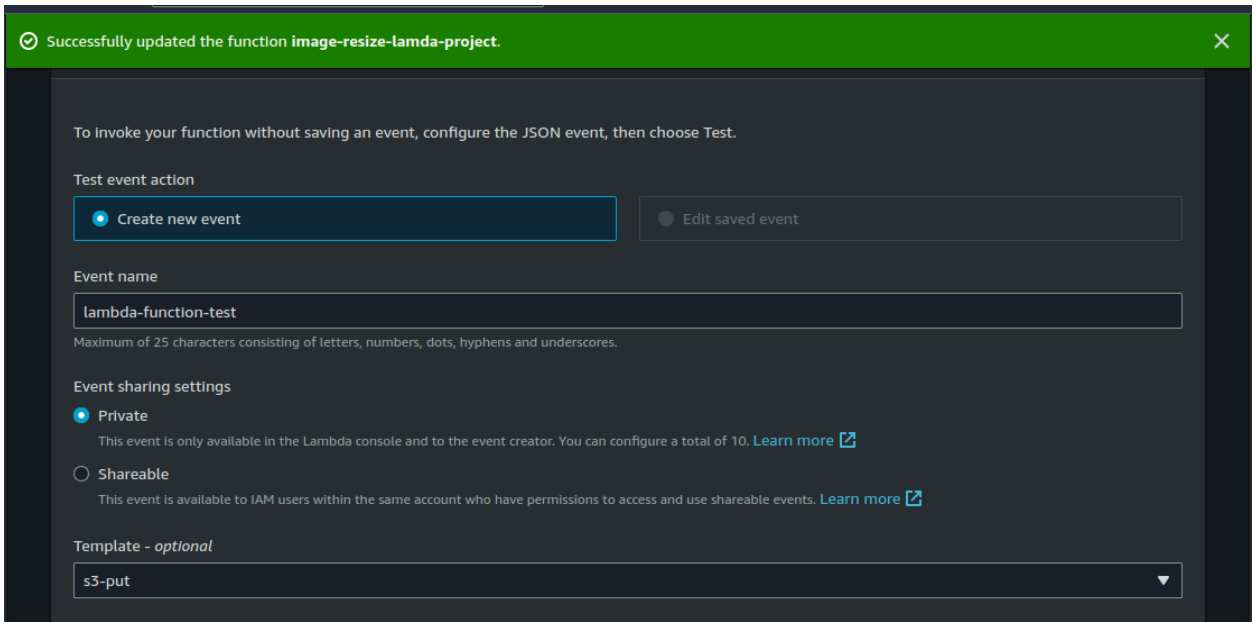
Code done uploading now setup environment variables for the project.

To compile package for production: *npm install --arch=x64 --platform=linux --target=16x sharp*

To create a package for upload : *npm run package*



Generate a Test case to Test the newly created Lambda Function.



Create an Event JSON to check the Function: create a S3 put Event Template. And change the input AWS S3 key,ARN and input parameters.

Event JSON Format JSON


```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH"
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "image-input-bucket22",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::image-input-bucket22"
        },
        "object": {
          "key": "FrKa.jpeg",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
```


Create Env variables to store the image to desired s3 bucket.
Edit > enter desired parameter and values

Code | Test | Monitor | Configuration | Aliases | Versions

- General configuration
- Triggers
- Permissions
- Destinations
- Function URL
- **Environment variables**
- Tags
- VPC
- RDS databases
- Monitoring and

Environment variables Edit

Find environment variables 1

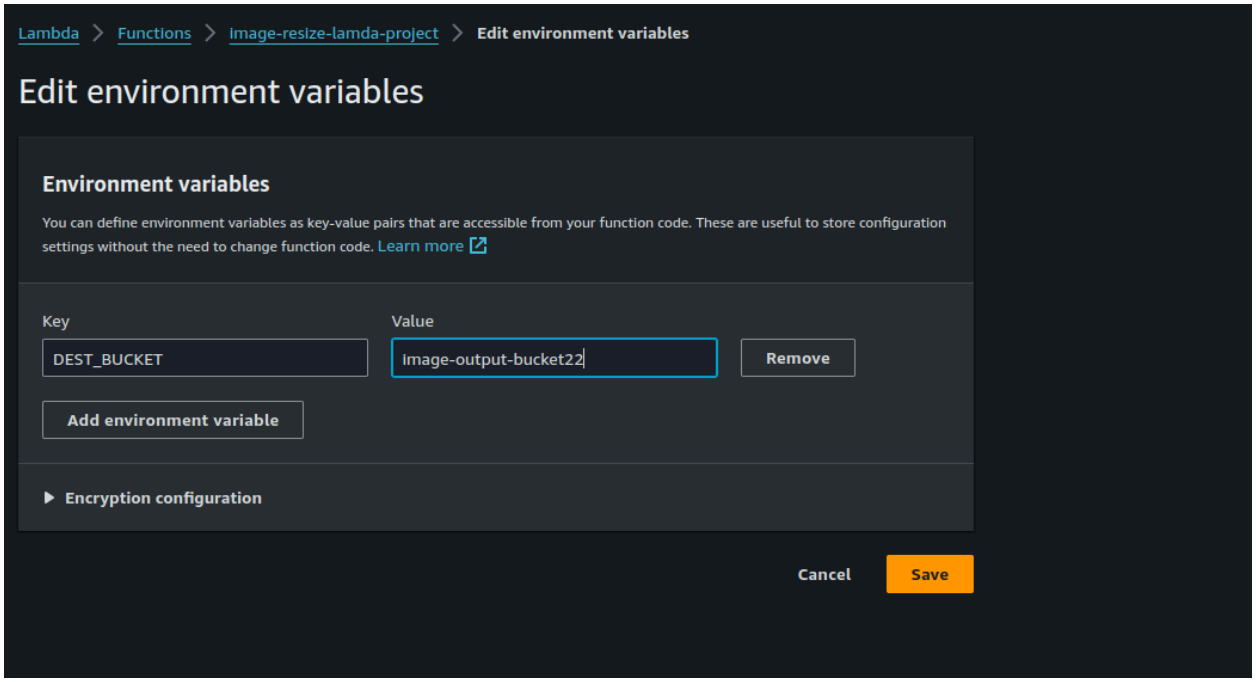
Key	Value

No environment variables

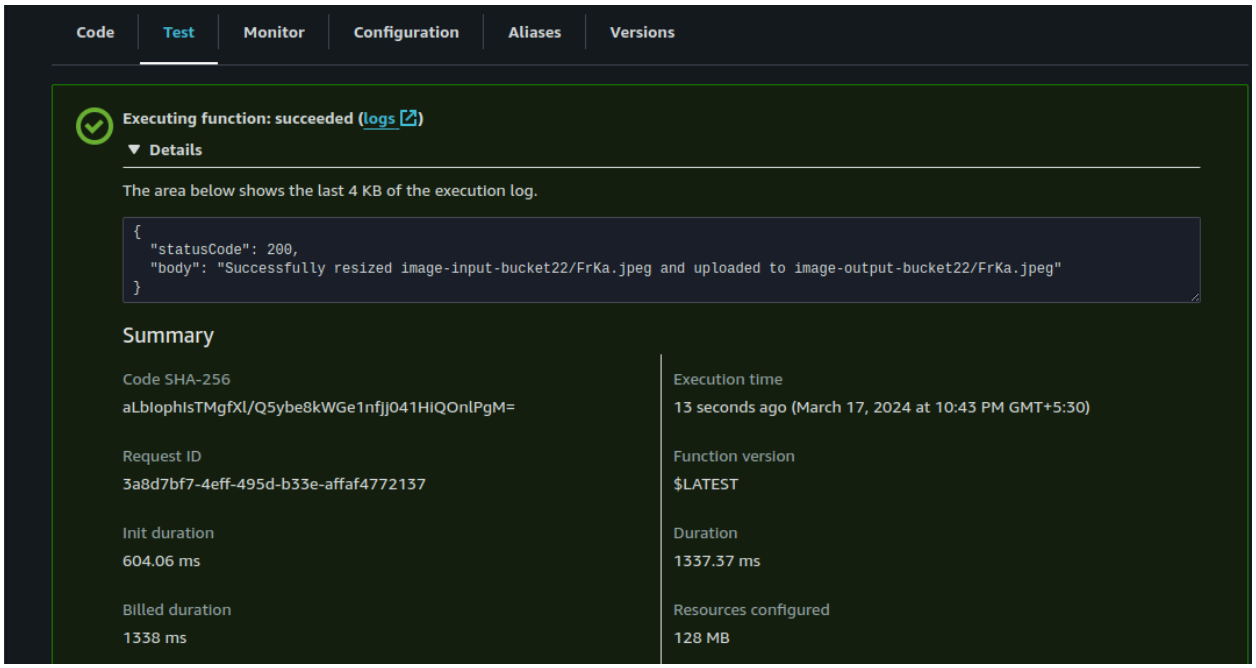
No environment variables associated with this function.

Edit

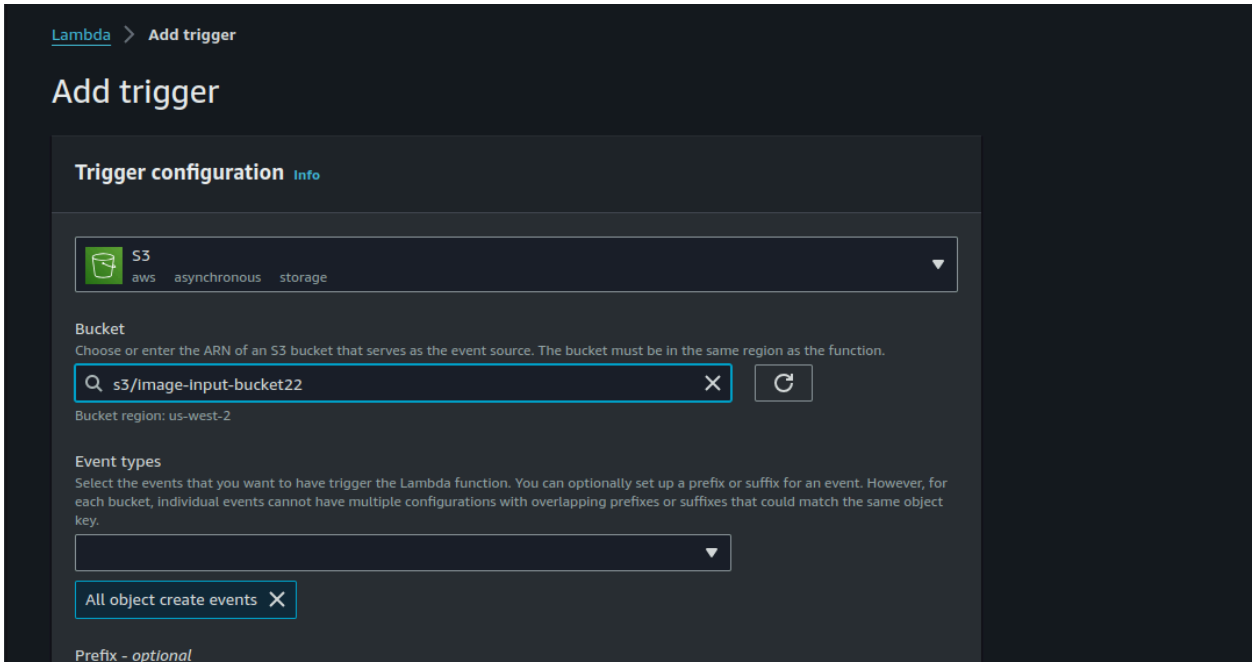
Key: DEST_BUCKET , **Value**:image-output-bucket22



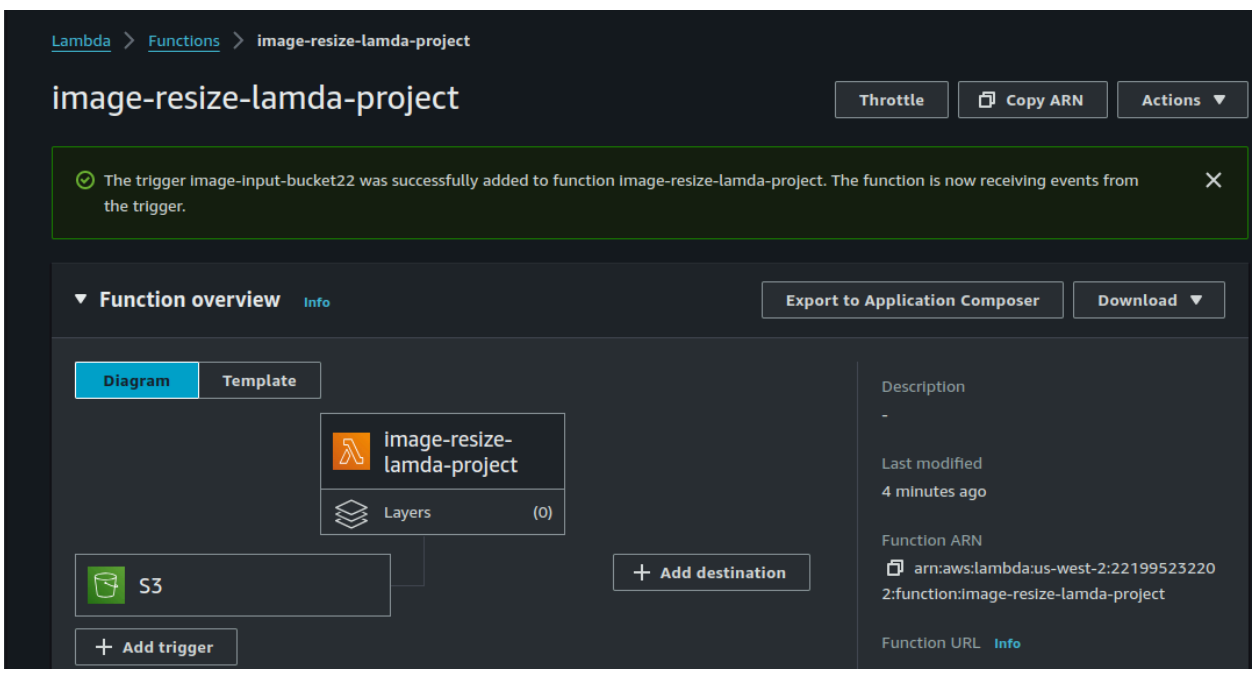
Run the rest: you will get Http status code 200 for your Test API

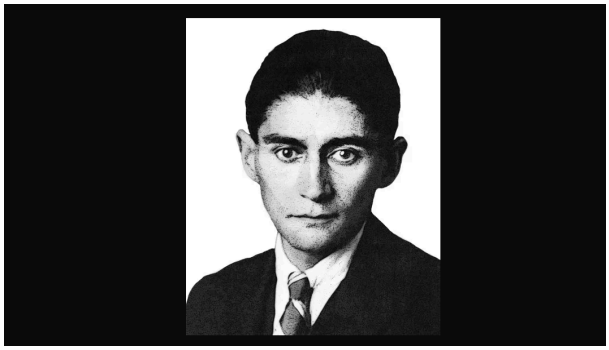


Check Your Output S3 bucket you can see your resized image in your bucket, now add the trigger to invoke your Lambda.

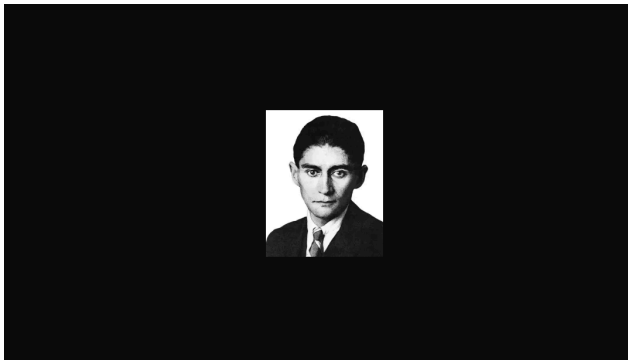


Now whenever you drop image to your input S3 bucket ,it will automatically convert and resize the image and you can see it in output S3 Bucket.





A. Before Image resize



B. After Image Resize

GoLang + HTMX Client