PROJECT -1

SERVERLESS IMAGE PROCESSING

PRESENTED BY AMOD PATHAK

INTRODUCTION :-

Serverless Image Handler on AWS creates a serverless architecture to initiate cost-effective image processing in the AWS Cloud. The architecture combines AWS services with sharp, an open-source image processing software, and is optimized for dynamic image manipulation. You can use this AWS Solution to help you maintain high-quality images on your websites and mobile applications to drive user engagement.

GOALS TO ACHIEVE:-

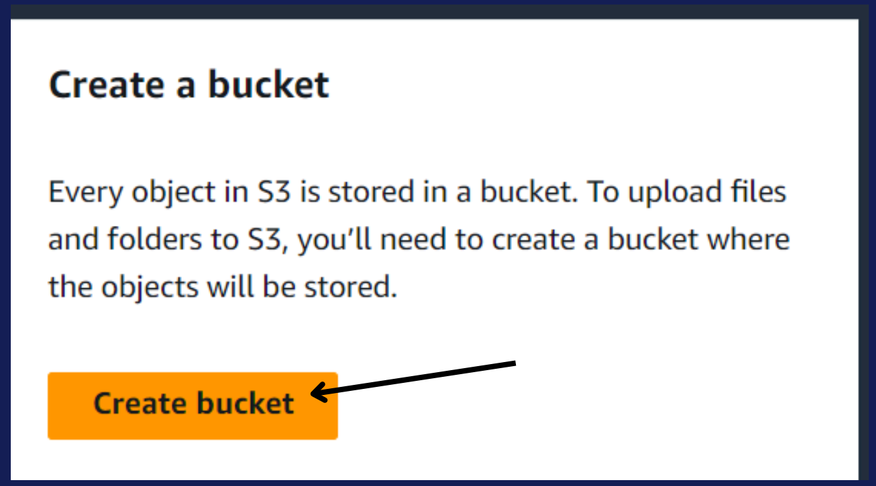
Create a serverless image prrocessing application that automatically resizes and optimizes images uploaded to an Amazon S3 bucket

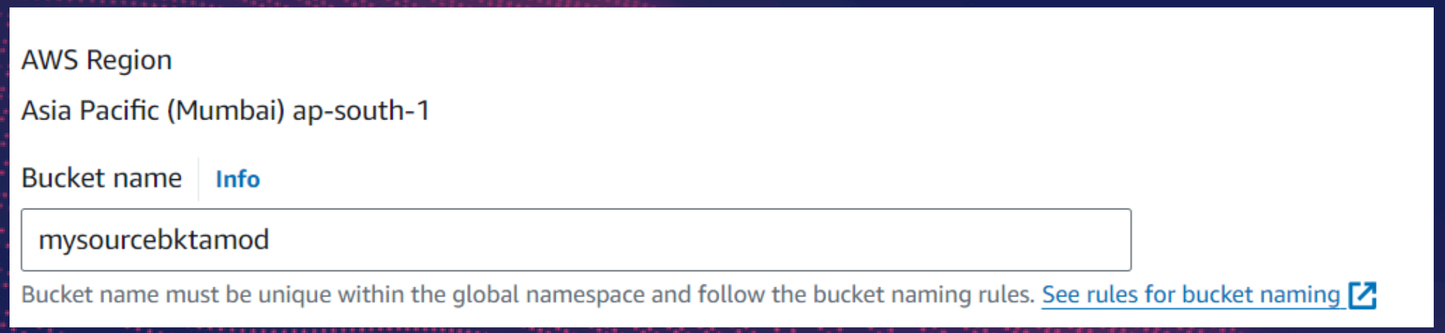
STEP 1 : Sign in to AWS console

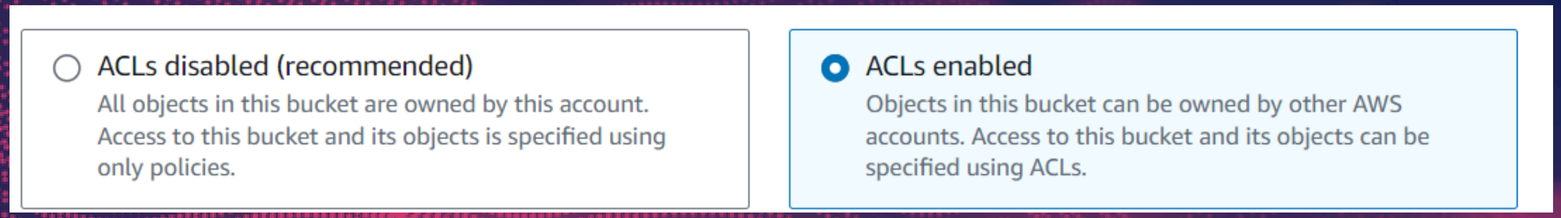
1. Go to AWS console and enter your details and log in to AWS cloud services

STEP 2 : CREATE A BUCKET

2. Click on SERVICES and open S3 bucket
3. Now click on create bucket and create a source bucket
4. Enter your bucket name “mysourcebktamod”
5. Now select ACL enabled option

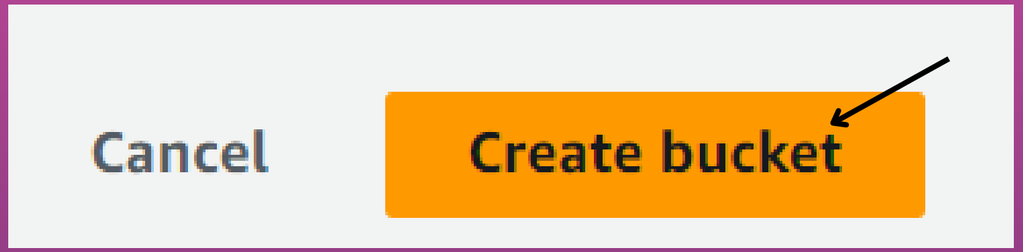






6. Remove “block all public access” and confirm the given request as mentioned

7. Leave all other options as default and create bucket

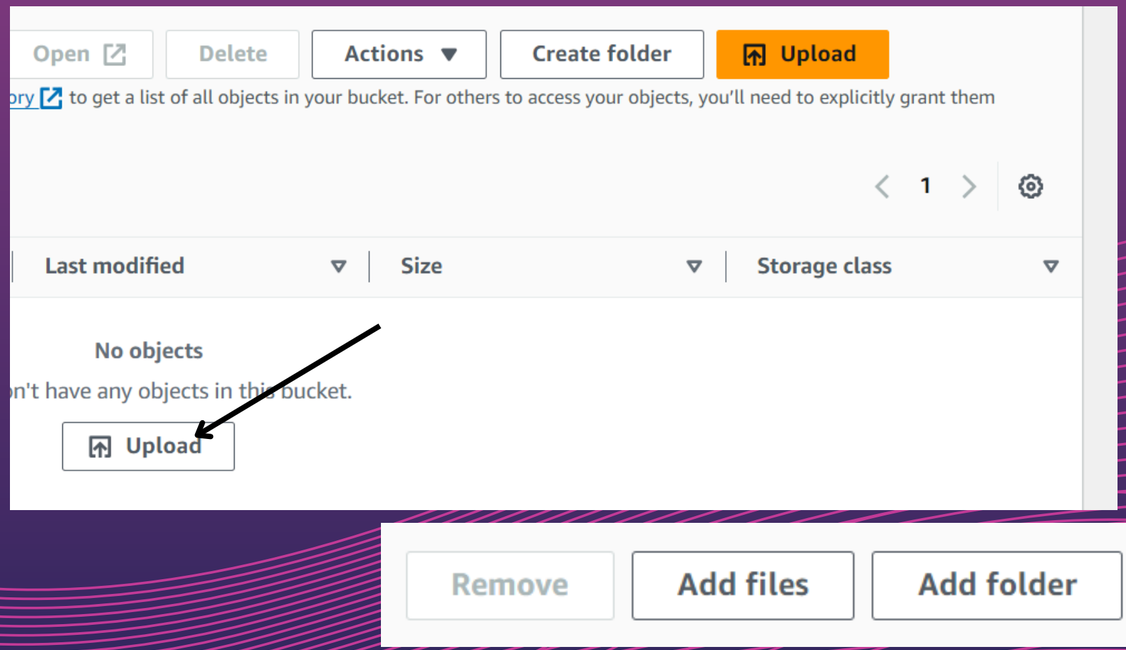


8. Just like source bucket ceate another bucket named as “mytargetbktamod”

Every bucket name is globally unique so select your unique bucket name

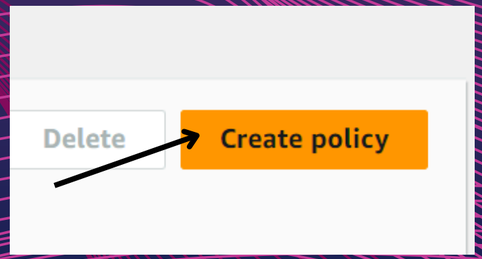
STEP 3 : UPLOAD YOUR IMAGE

9. Open your source bucket and click on upload and using add files option you can upload your image



STEP 4: CREATE ROLES AND POLICIES

10. In services menu go to “I AM” and click on policies and create a policy as mentioned

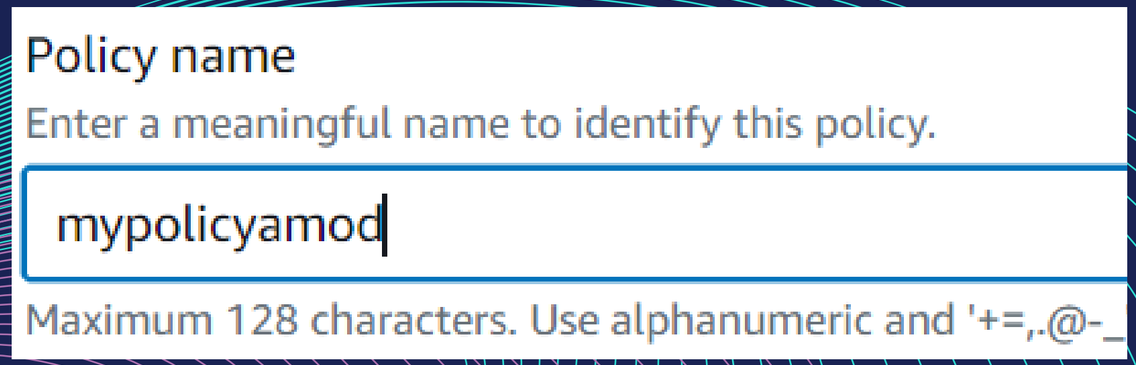


11. Click on Json and write this given code in policy generator and click on NEXT

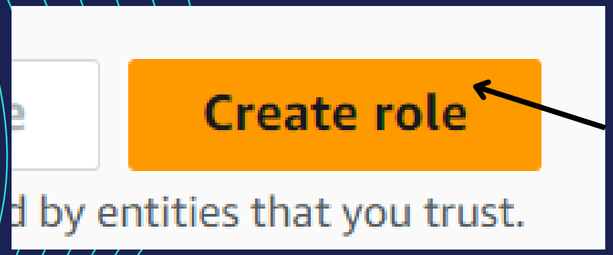
Policy editor

```
{
  "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:::mysourcebktkrati/*"
    },
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": "arn:aws:s3:::mydestinationbktkrati/*"
    }
  ]
}
```

12. Now enter your policy name and create policy

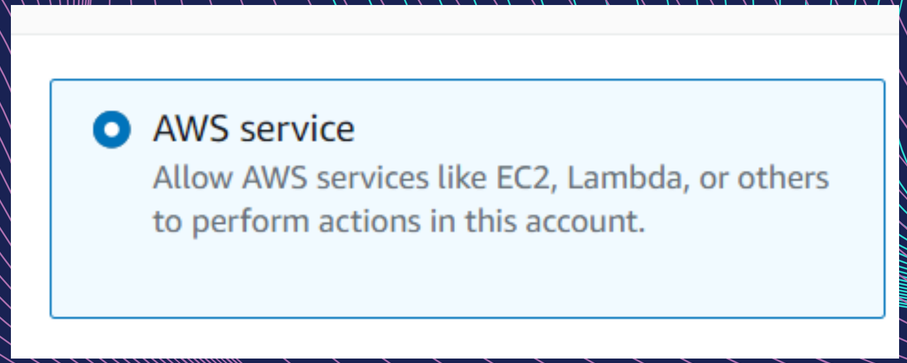


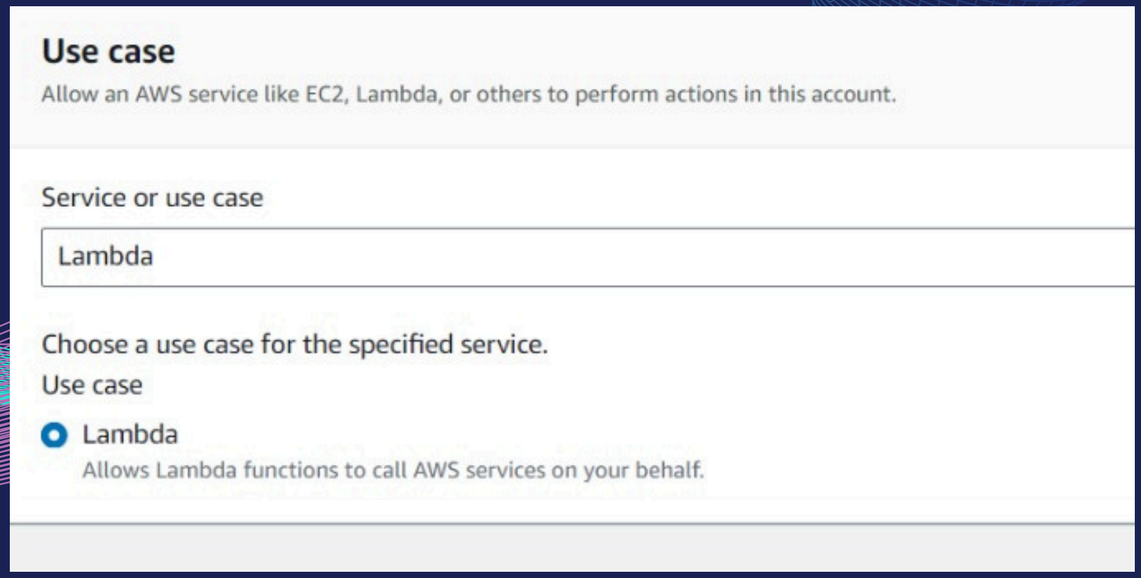
13. After this go to Roles click on " create roles"



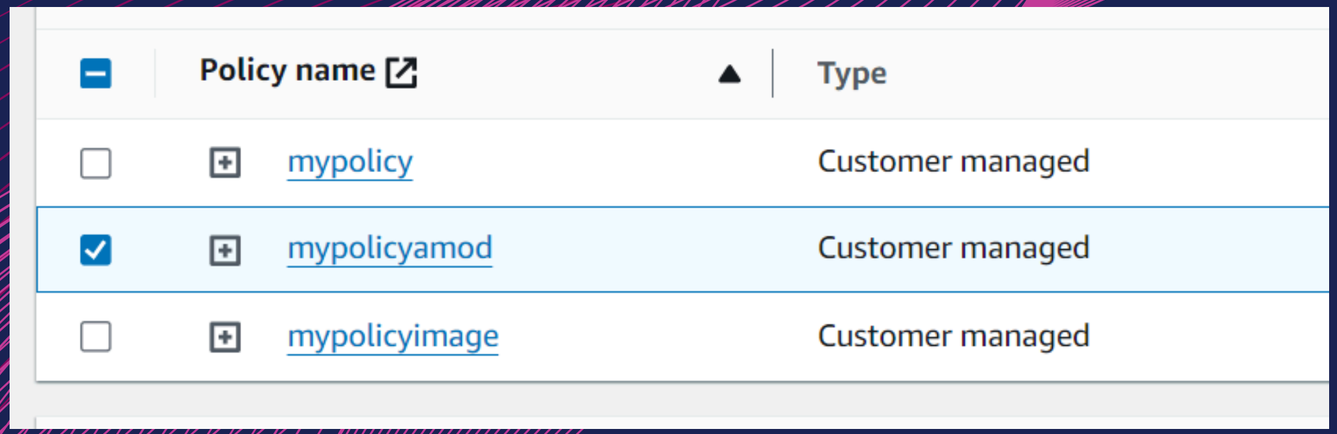
14.Now select AWS services

15. In use case option select Lambda and click on Next

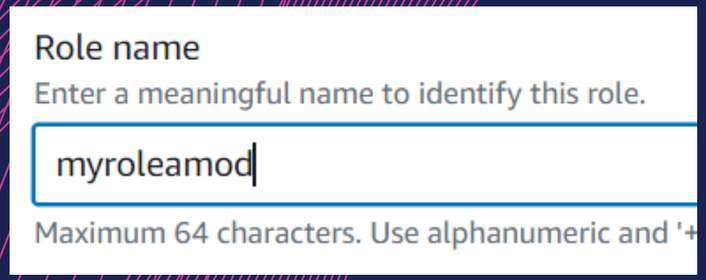




16. Now go to policy and select your created policy then click on Next



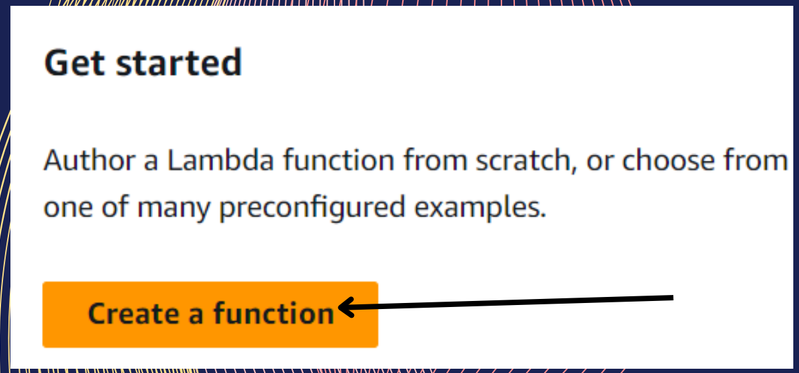
17. Enter your role name and create your role

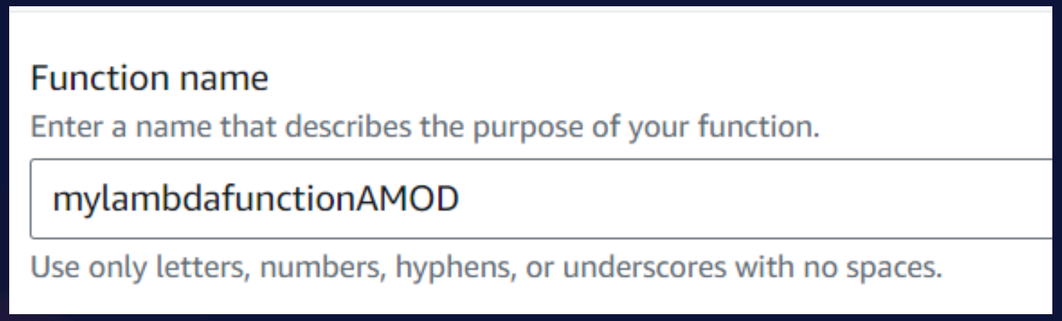


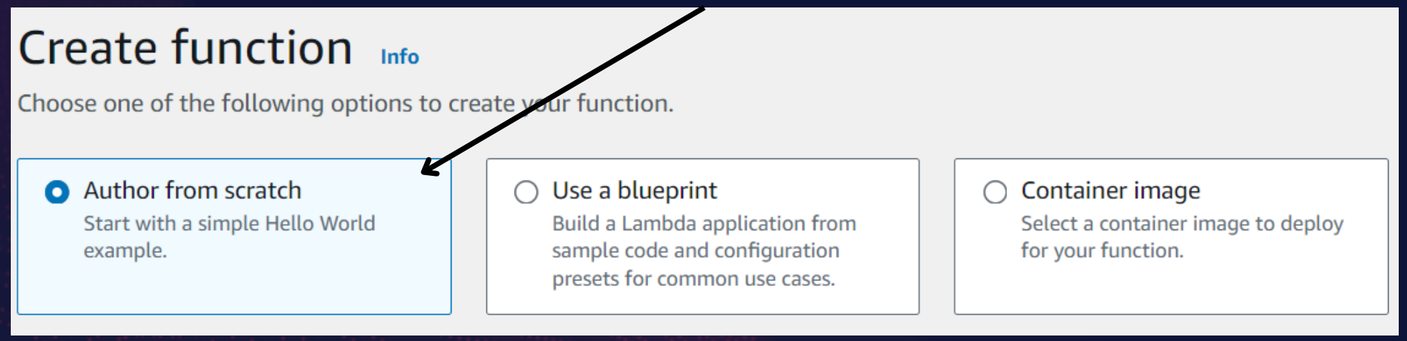
STEP 5 : CREATE A LAMBDA FUNCTION

18. Now go to services and open “lambda” and click on “create a function

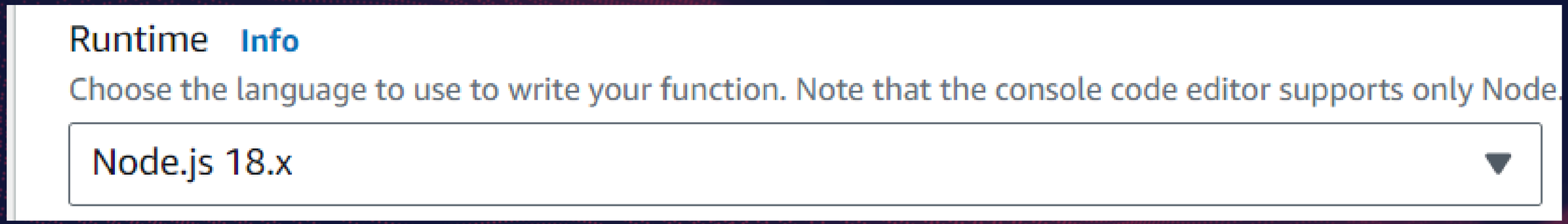
19. Select author from scratch option and name your function





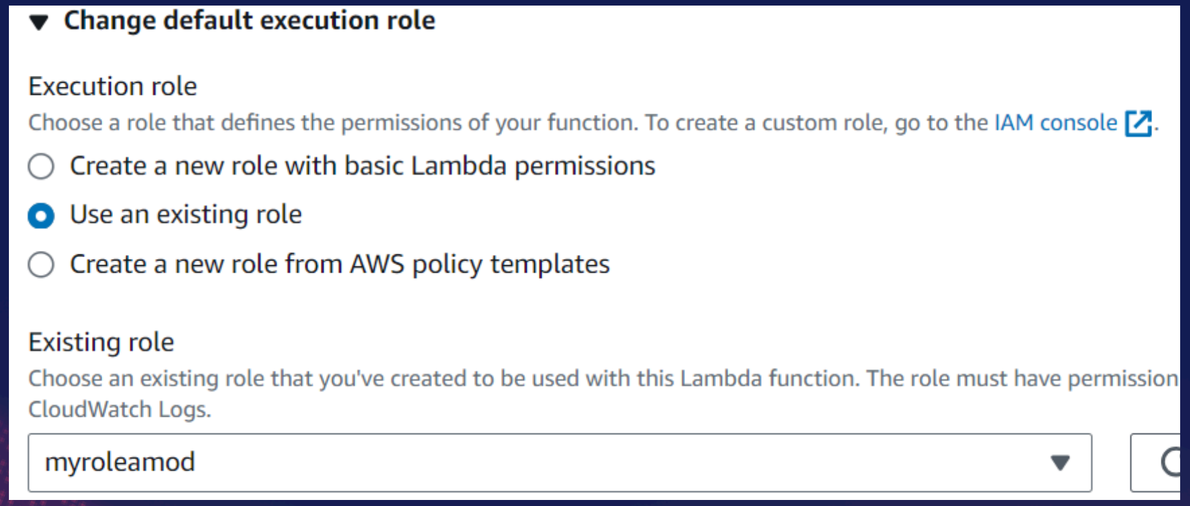


20. Now in Runtime option select”node.js 18.x”



21. In change default execution role select “use an existing role” and select your created role

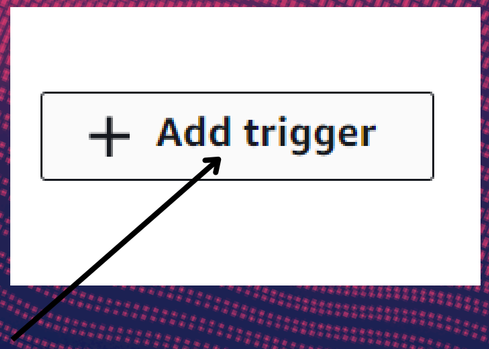
22. Now click on create function

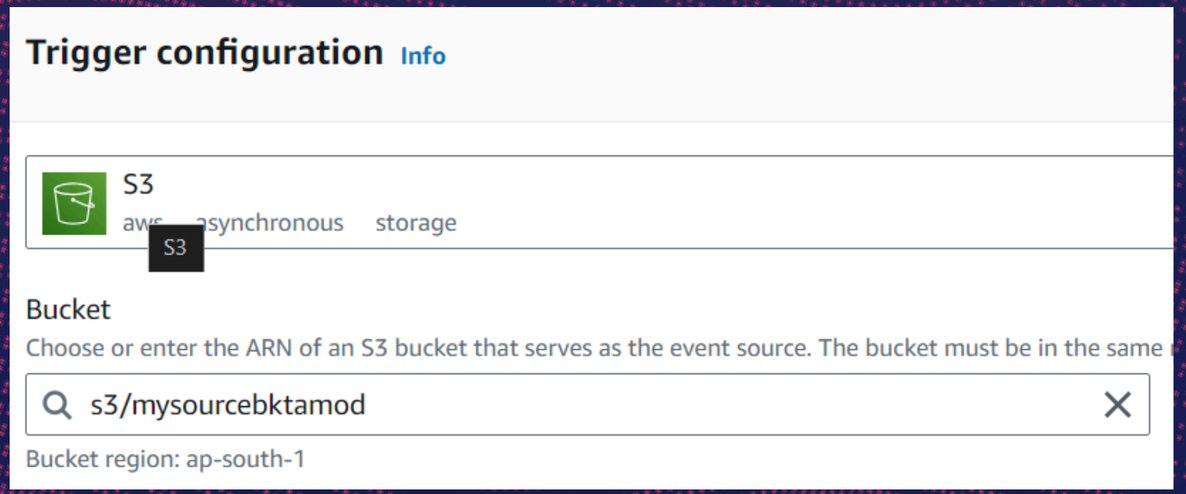


STEP 6 : ADDING TRIGGER AND CREATING ENVIRONMENT VARIABLES

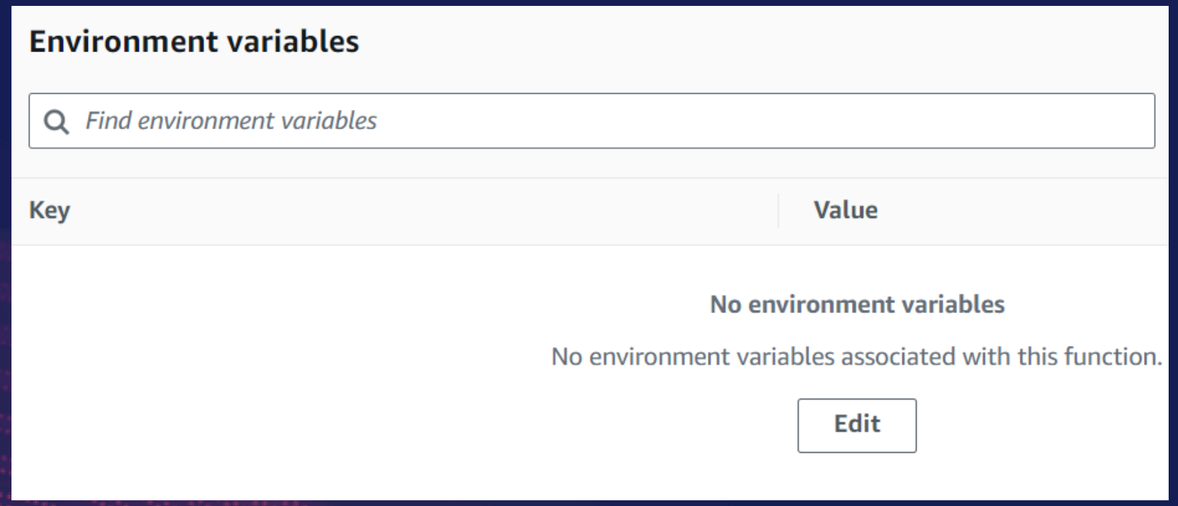
23.Now click on add trigger and select S3

24. Enter your source bucket name and click on add



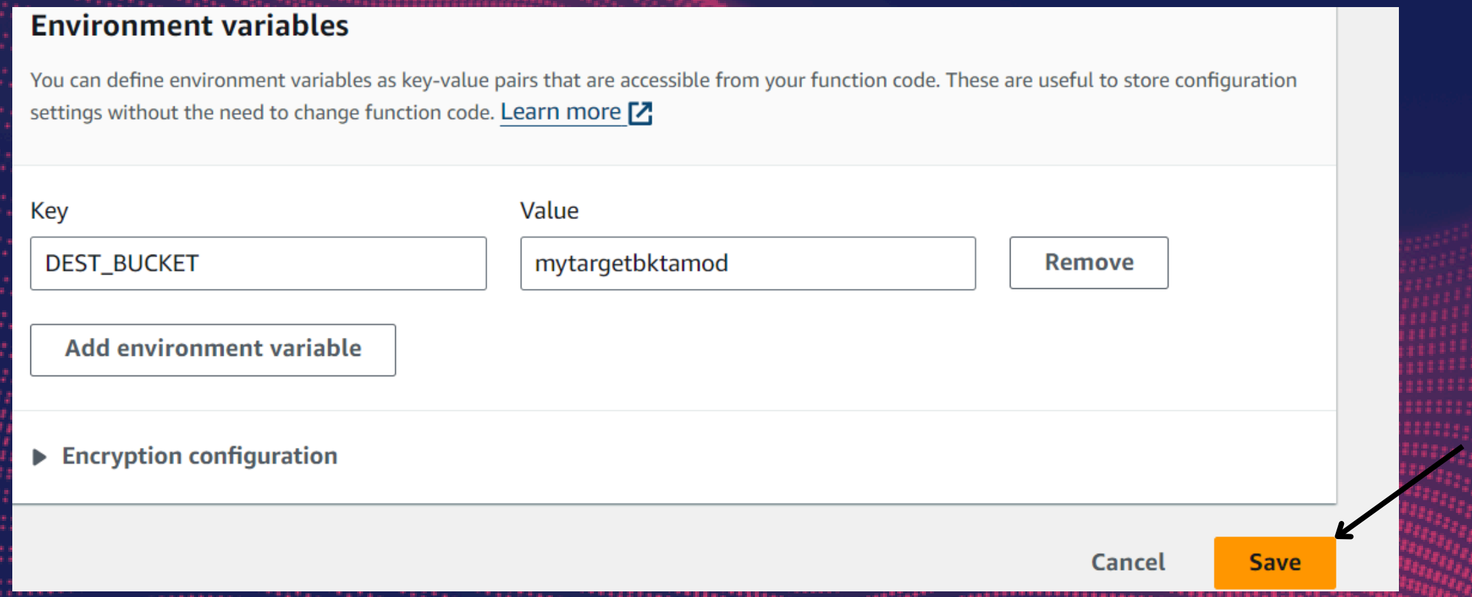


25. Now go to configuration and select environment variables



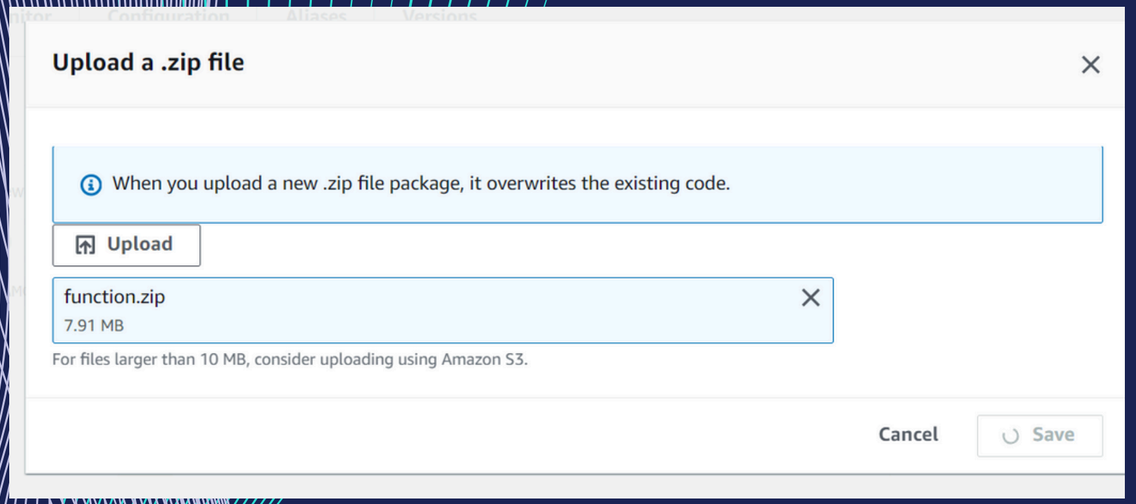
26. Select edit and enter key as" DEST_BUCKET and value as ": mytargetbktamod"(use your unique bkt

27. Click on Save



STEP 7 : UPLOADING ZIP FILE AND RUNNING TEST

28. Now go to Code , and under code select upload and upload your ZIP file



29. Then open "Test" and select s3-put in Template
30. Then enter name of your source bucket and name of your image manually at their respective columns

Event 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": "mysourcebktamod",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
          "arn": "arn:aws:s3:::mysourcebktamod"
        },
        "object": {
          "key": "wallpaper.jpg",
          "size": 1024,
          "eTag": "0123456789abcdef0123456789abcdef",
          "sequencer": "0A1B2C3D4E5F678901"
        }
      }
    }
```

31. Click on Test .
32. Now your execution is successfully completed and you can check your resized image in your target bucket

For Any Queries

Contact us at :

*919336453059

www.amodaws.com

amodpathak51@gmail.com

E-396 K.D.A colony daheli
sujanpur kanpur,208013

SUBMITTED TO Mr .Abhinav Dwivedi