PROJECT PRESENTATION

PRESENTED BY ANURAG VERMA

INTRODUCTION :-

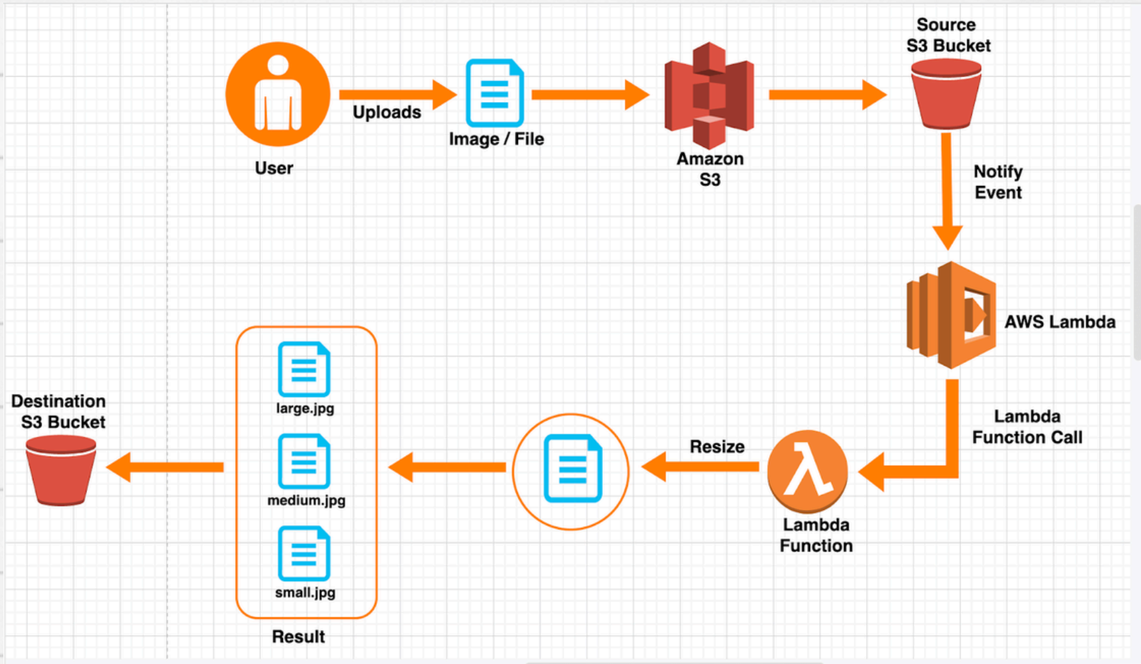
Amazon Web Services (AWS) is a comprehensive cloud computing platform provided by Amazon. It offers a wide range of services, including computing power, storage options, and networking capabilities, on a pay-as-you-go basis. AWS allows businesses to scale and grow efficiently by providing infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Key services include Amazon EC2 (virtual servers), Amazon S3 (storage), Amazon RDS (database management), and AWS Lambda (serverless computing).

Serverless Image Processing

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

Amazon S3: Used to store original images uploaded by users.

AWS Lambda: Serverless compute service used to execute the image processing tasks.

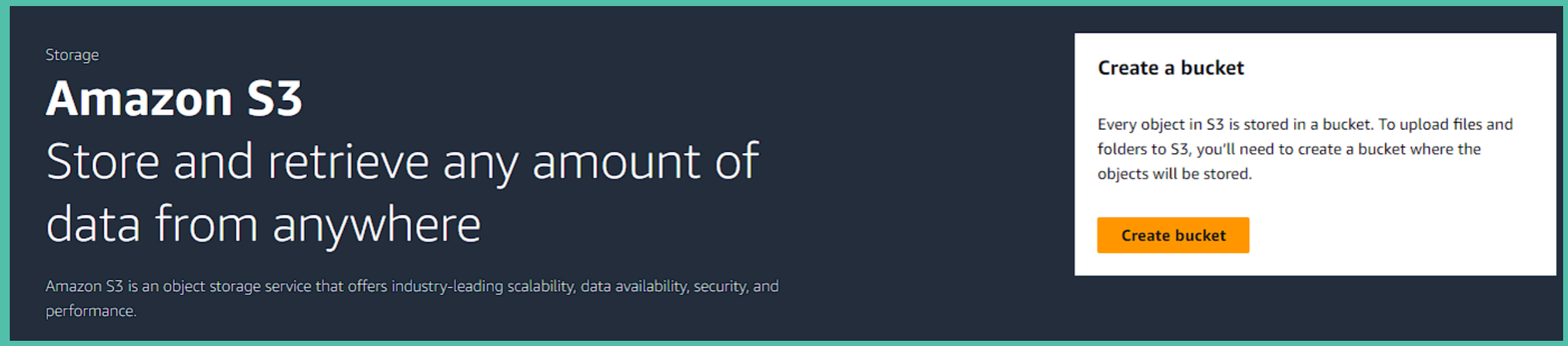


For Create a serverless image processing application that automatically resizes and optimizes images You need to follow the given steps:-

STEP 1:- Login in your AWS account.

STEP 2:- In AWS console home page,search S3.

STEP 3:- Click on S3.



STEP 4:- Bucket Name :- mysourcebucketanurag

AWS Region

Asia Pacific (Mumbai) ap-south-1

Bucket name Info

mysourcebucketanurag

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

STEP 5:- Now,click on ACLs enable

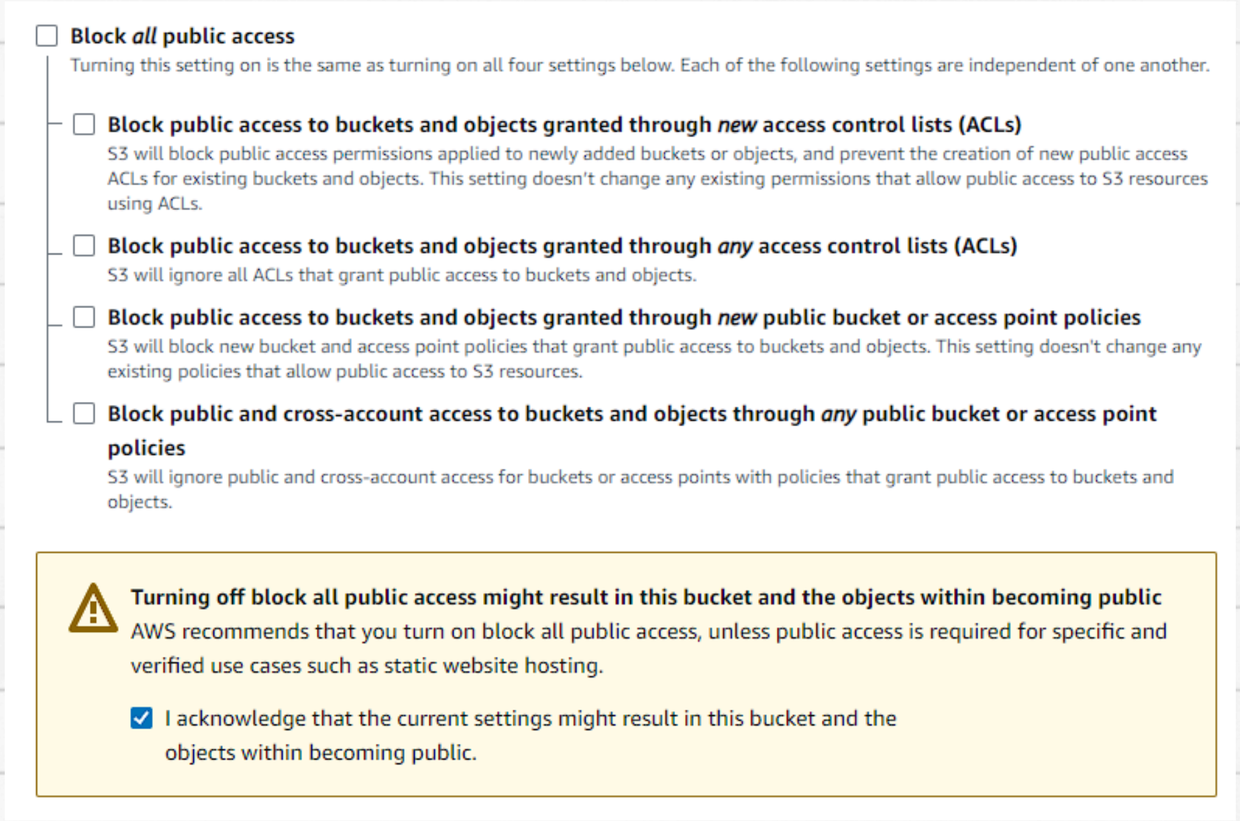
Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

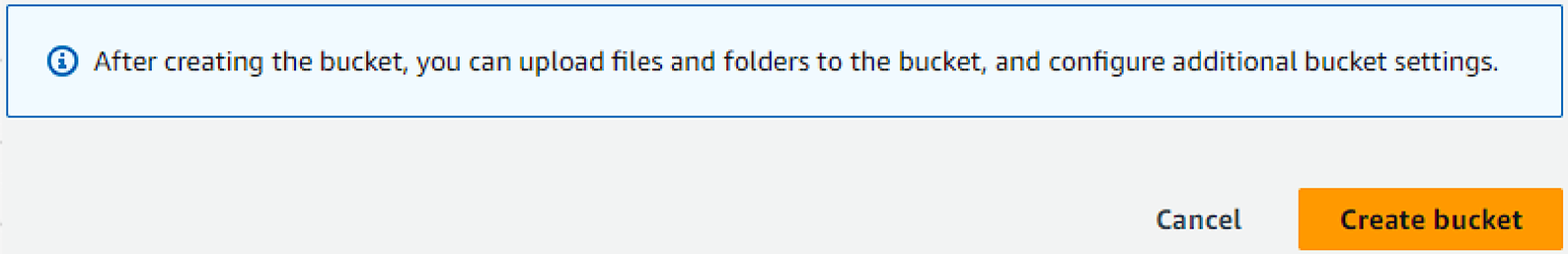
ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

STEP 6:- Now, turn off all block public access service



STEP 7:- Now leave all the setting default, and click on create the bucket



STEP 8:- Once the bucket is created successfull, select your S3 bucket
Click on the Copy ARN button to copy the ARN.
Save the source bucket ARN in a text file for later use
arn:aws:s3:::mysourcebucketanurag

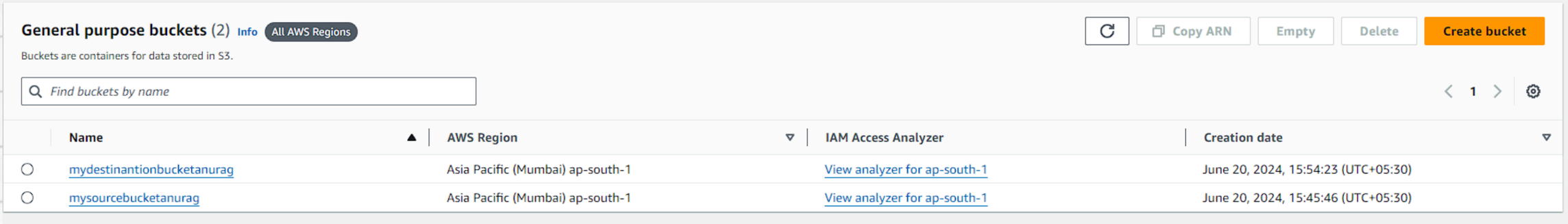
STEP 9:- Create Destination Bucket

STEP 10:- Click on Create bucket button

STEP 11:- Bucket Name: Enter mydestinationbucketanurag

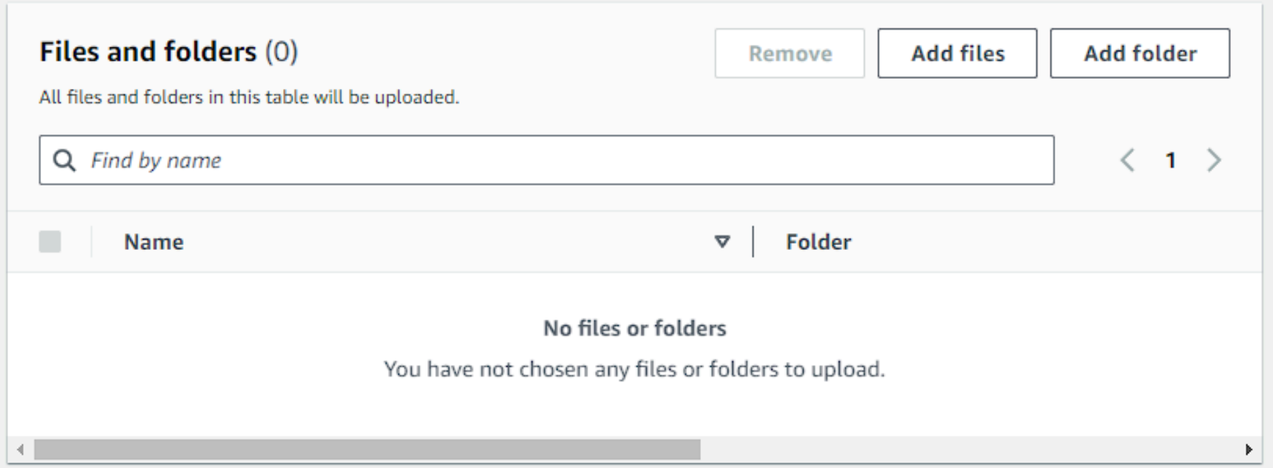
STEP 12:- Once the bucket is created successfully, Select your S3 bucket.

STEP 13:- Click on the Copy ARN button to copy the ARN.
Save the destination bucket ARN in a text file for later use.
arn:aws:s3:::mydestinantionbucketanurag

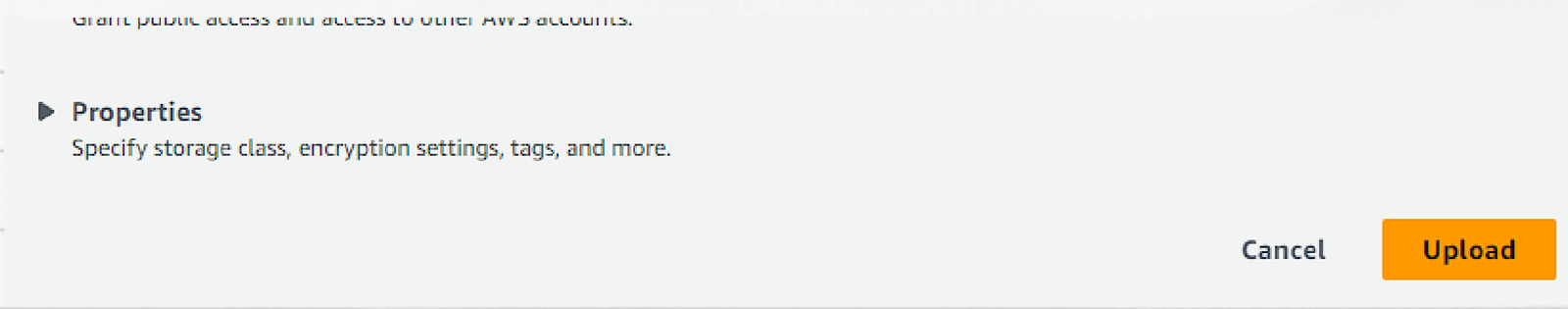


STEP 14:- Now we have two S3 buckets (Source and Destination). We will make use of our AWS Lambda function to copy the content from source bucket to destination bucket.

STEP 15:- Now we have to click the mysourcebucketanurag and click the add files option



STEP 16:- Now upload the file which you have been selected



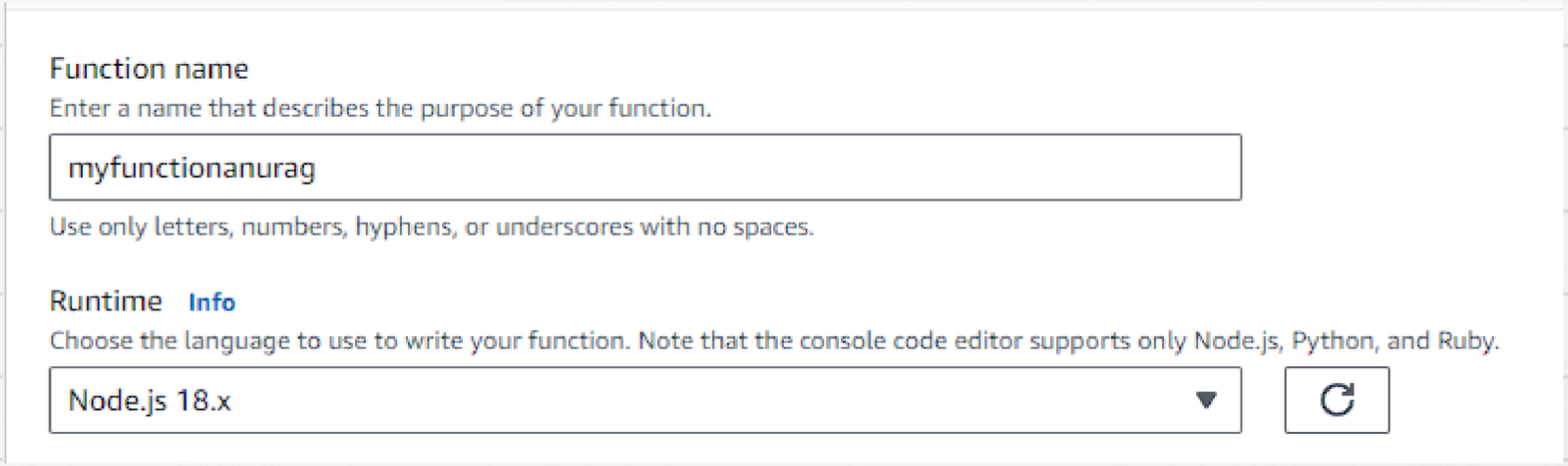
STEP 17:- Now ,Create a Lambda Function

Go to the Services menu and click on Lambda under Compute section.

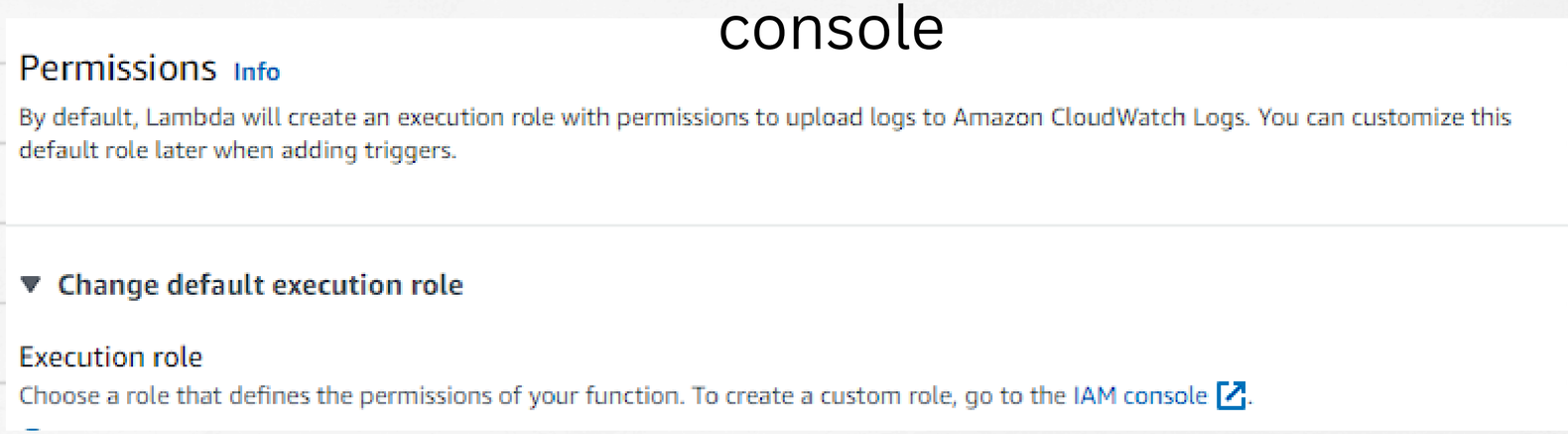
Click on the Create a function button.

Function name : Enter myfunctionanurag

Runtime : Select Node.js 18x



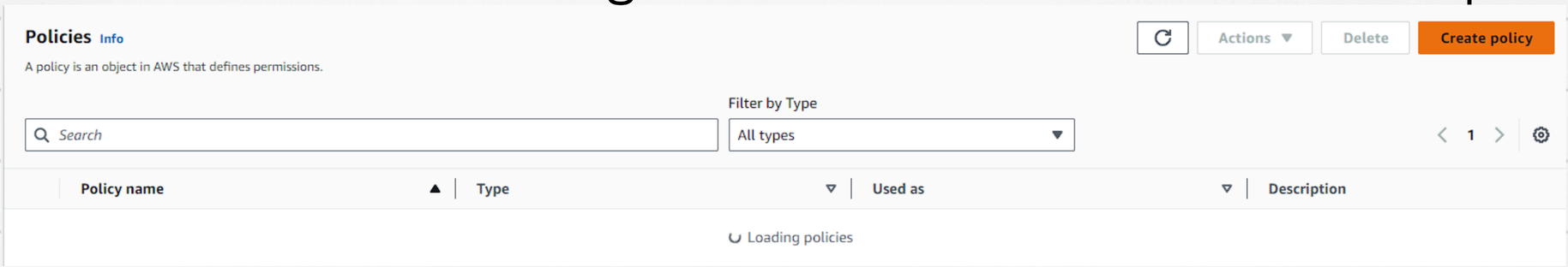
now, we have to click on the change default execution role and click on the IAM console



STEP 18:- Now , click on the IAM services and create a policy

Go to Services and Select IAM under Security, Identity and Compliance.

Click on Policies in the left navigation bar and click on the Create policy button.



Click on the JSON tab, Remove the existing code and copy-paste the below policy statement into the editor:

Policy JSON:

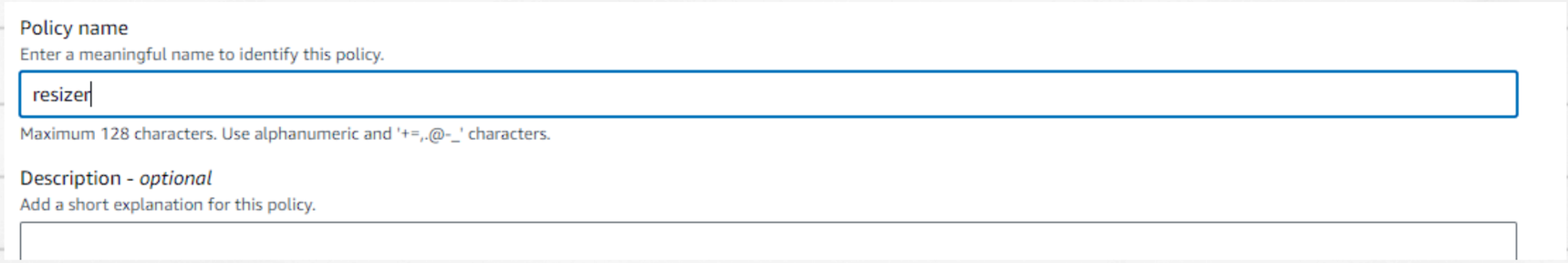
```
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": [
"logs:PutLogEvents",
"logs:CreateLogGroup",
"logs:CreateLogStream"
],
"Resource": "arn:aws:logs:*:*:*"
},
{
"Effect": "Allow",
"Action": ["s3:GetObject"],
"Resource": "arn:aws:s3:::mysourcebucketanurag/*"
},
{
"Effect": "Allow",
"Action": ["s3:PutObject"],
"Resource": "arn:aws:s3:::mydestinantionbucketanurag/*"
}
]
}
```

Replace the Source and destination ARN name of the bucket (which you have saved before) in the option Resource.

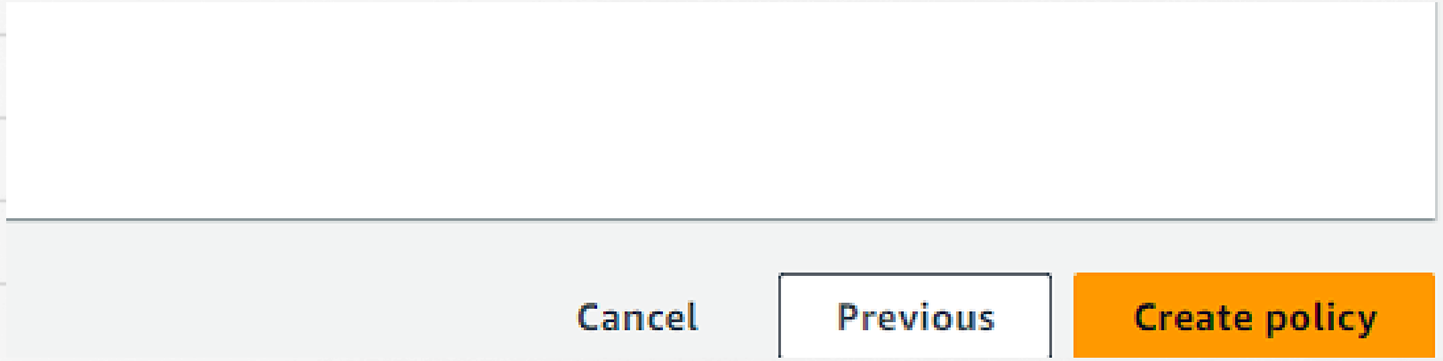
Leave everything as default and click on Next button

STEP 19:- On the Review Policy Page

Click on the Create policy button.



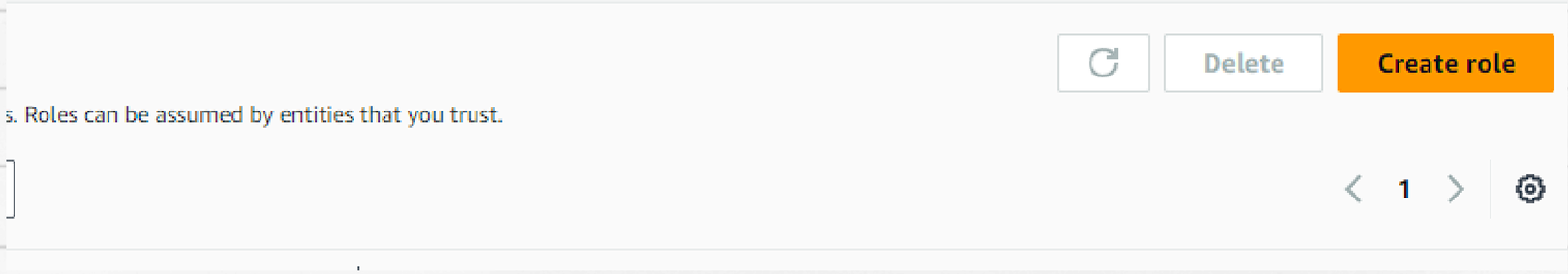
Now, leave all setting as default and click on the create policy



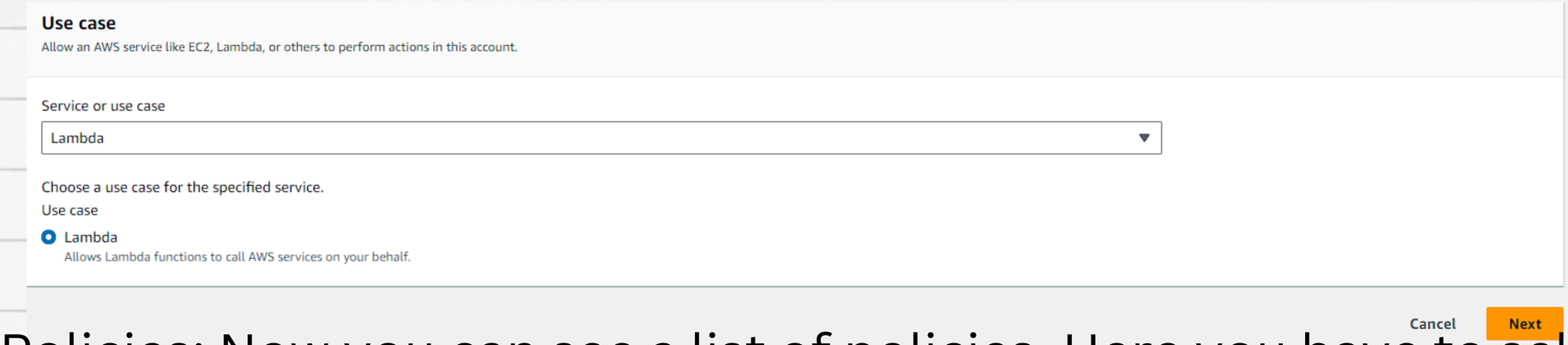
STEP 20:- Create an IAM Role

In the left menu, click on Roles. Click on the Create role button.

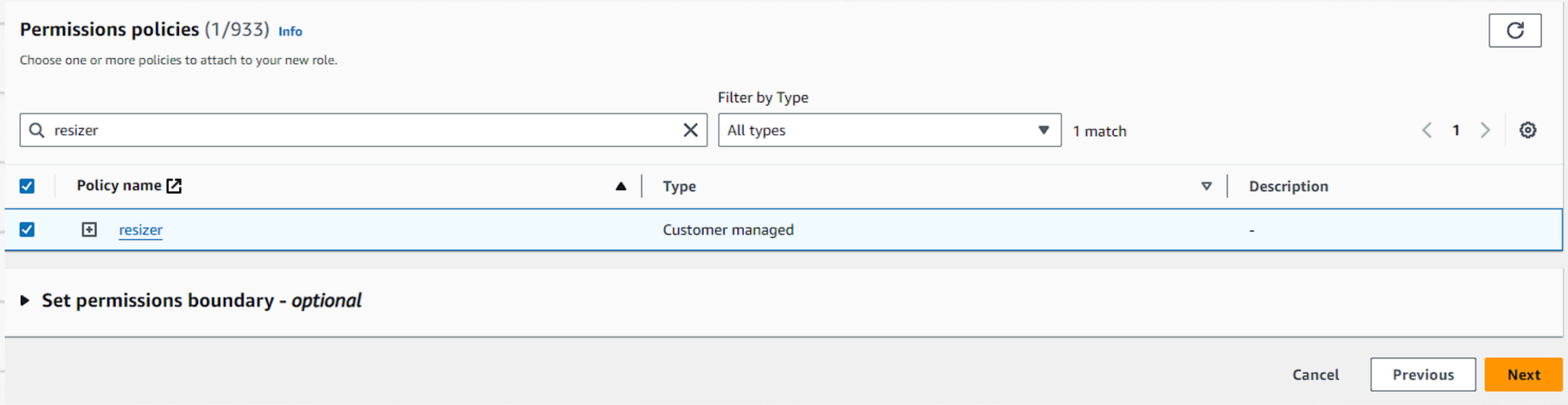
Select Lambda from AWS Services list.



From Use case: Select Lambda



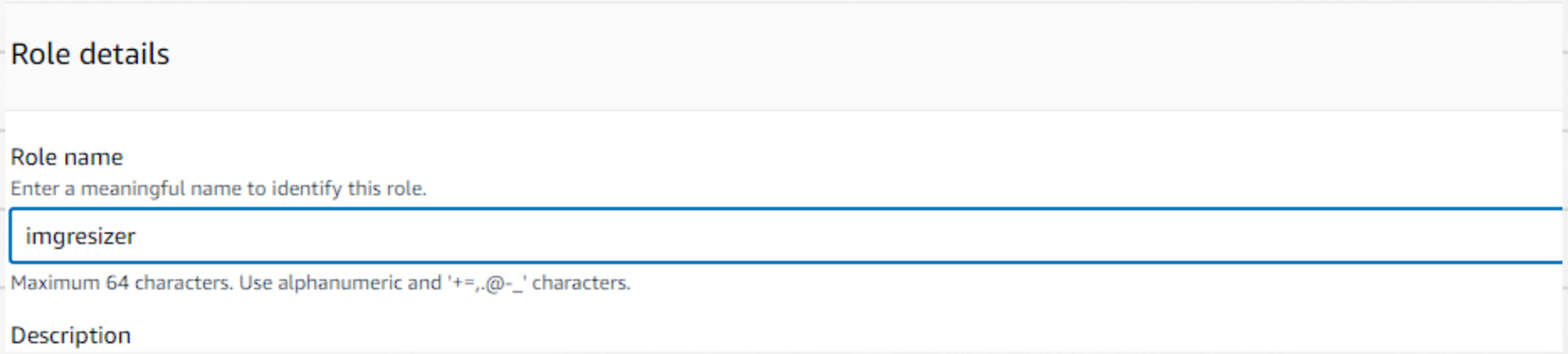
Filter Policies: Now you can see a list of policies. Here you have to select the resizer policy



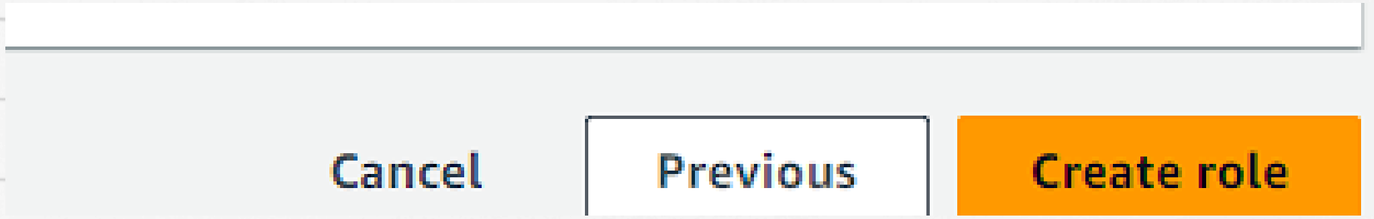
Select your policy and click on the Next button.

STEP 21:- Click on the Create Role button

Role Name: Enter imgresizer



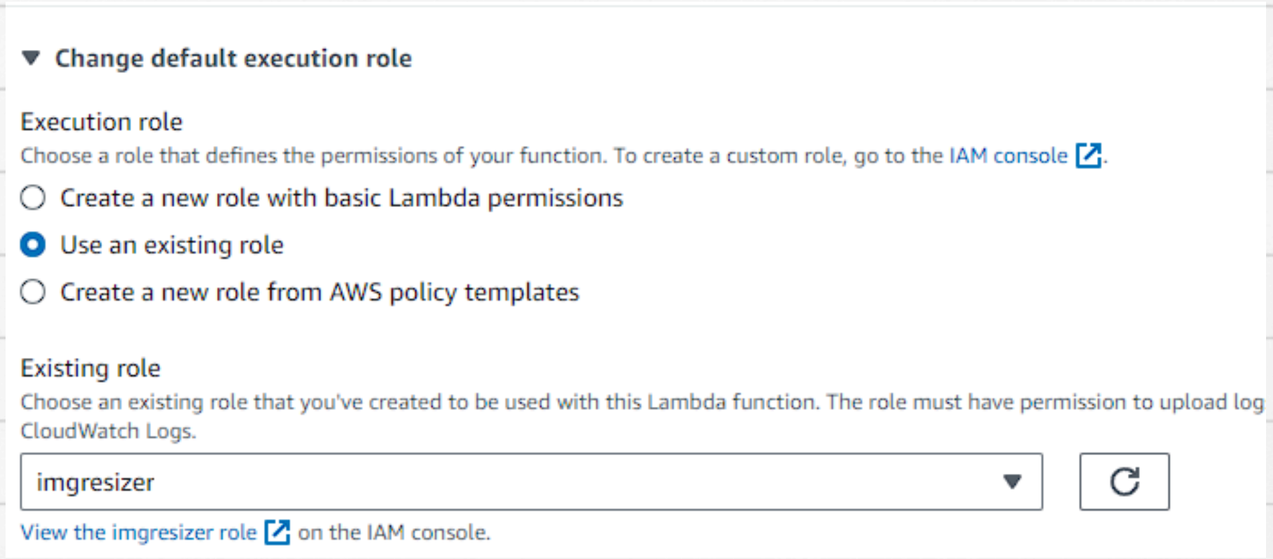
Now click on the create role



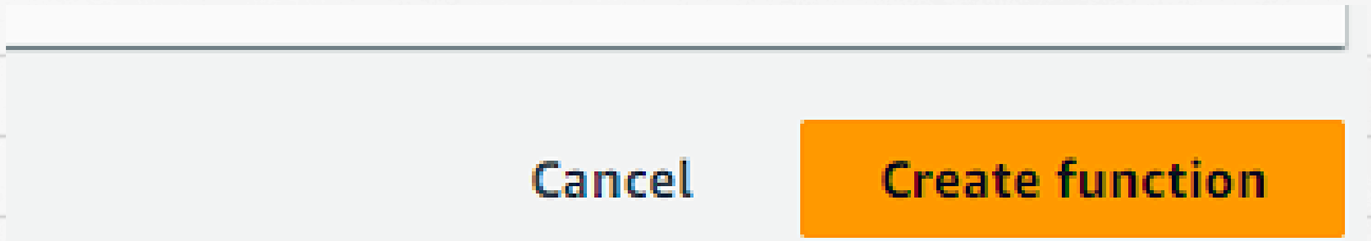
STEP 22:- Now, after creating the IAM policies and roles you have to get back in the lambda function setting

STEP 23:- Now you have to click change default execution role and select the use an existing role

Now, select the created imgresizer role



and continue to the create function

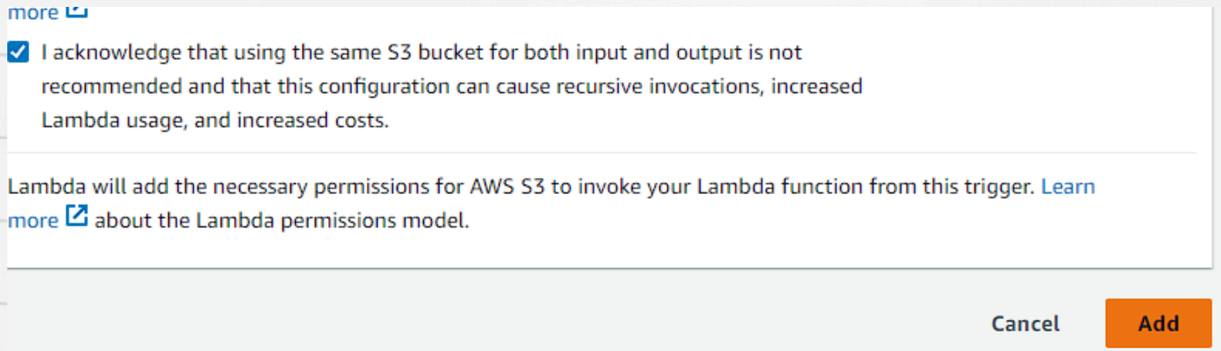


STEP 23:- Adding Triggers to Lambda Function

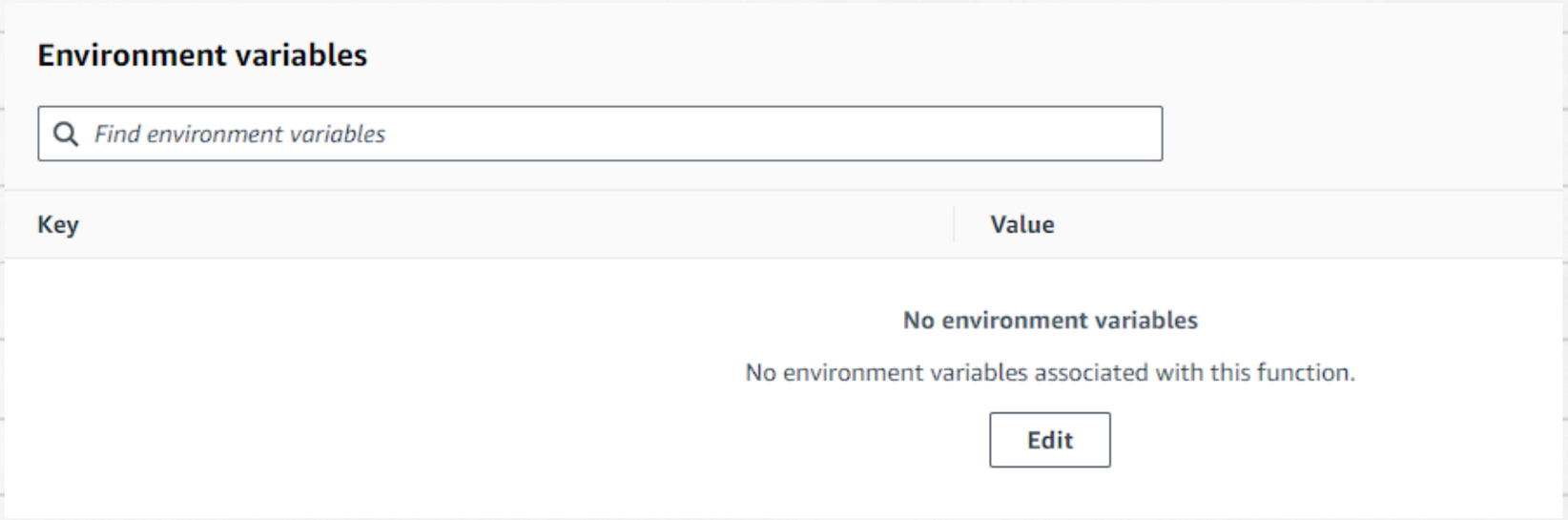
Scroll down the list and select S3 from the trigger list.

Bucket: Select your source bucket - mysourcebucketanurag

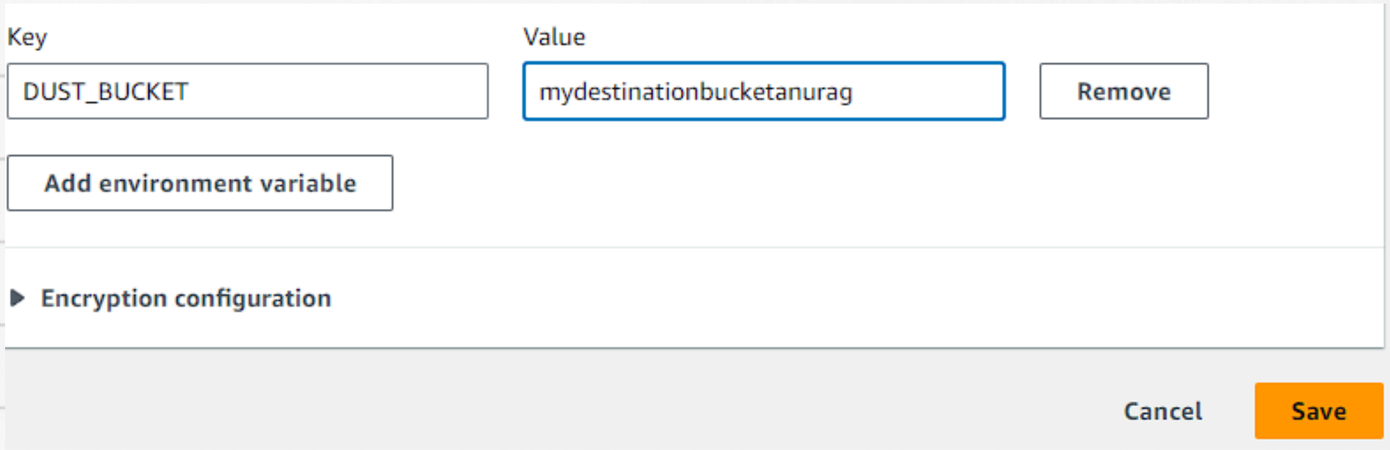
Leave other fields as default.



STEP 24:- Now, you have to create environment variables

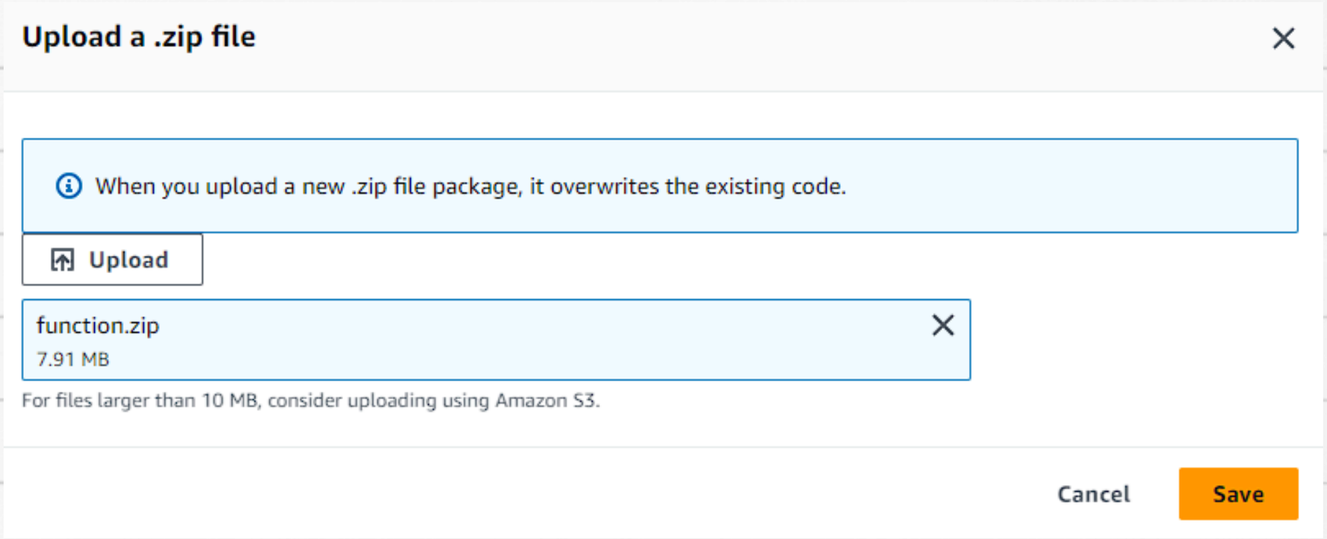


STEP 25:- click on the edit option and add the key: DEST_BUCKET and value: mydestinationbucketanurag and continue to the save option

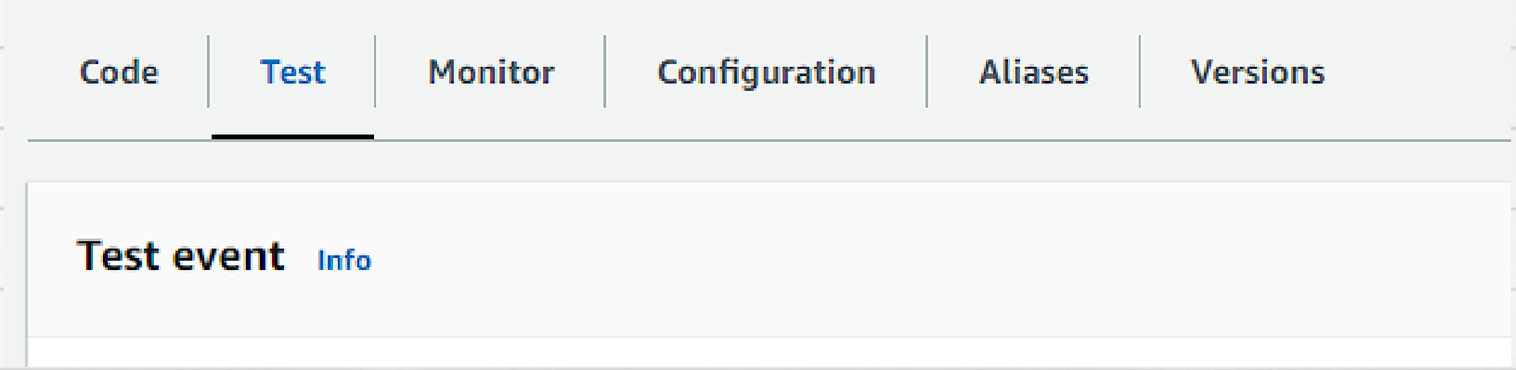


STEP 26:- Now click on the source code and click on the upload from and select .zip file

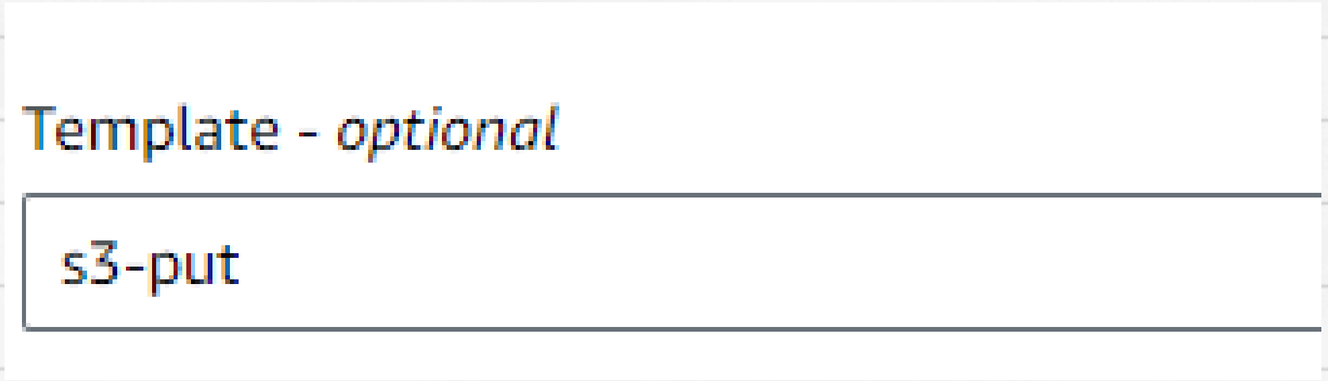
Now upload the given function.zip file



STEP 27:- After that click on the Test



STEP 28:- Now, click on the Template - optional and select the s3-put



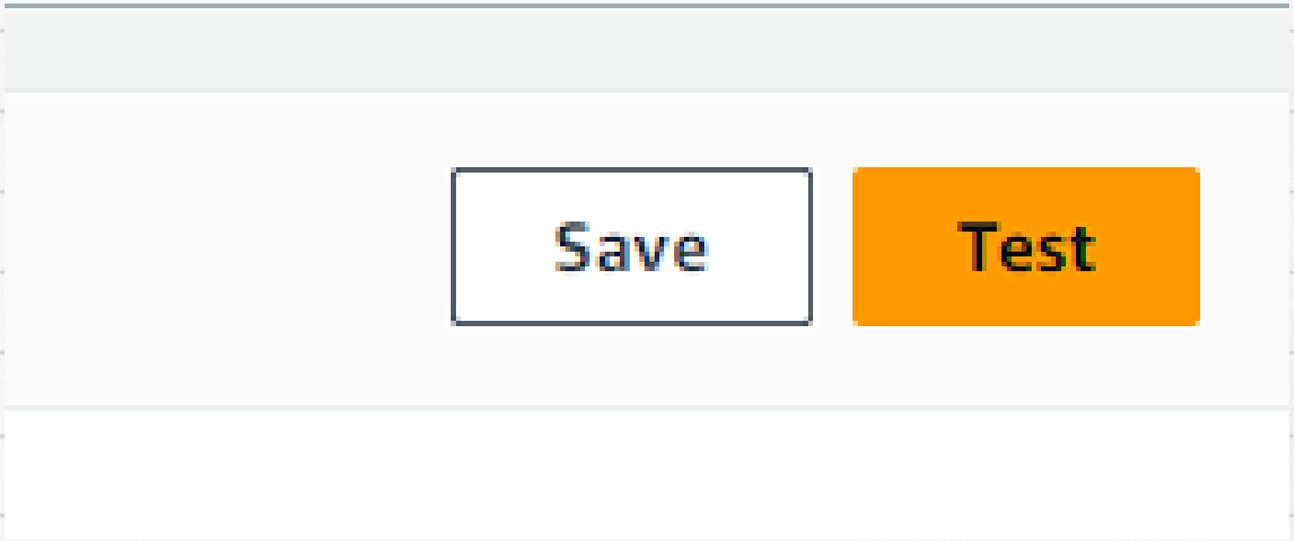
STEP 28:- Now, put the source bucket name : mysourcebucketanurag in the given code

Also put the uploaded file name in the code KEY:shinchan.jpg


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

STEP 29:- Now, click on the Test



STEP 30:- After the Test completed Now, go to the S3 bucket and click on the mydestinationbucketanurag

Where you can show the uploaded file in the destination bucket now select the file and click on the open

mydestinantionbucketanurag Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all obje

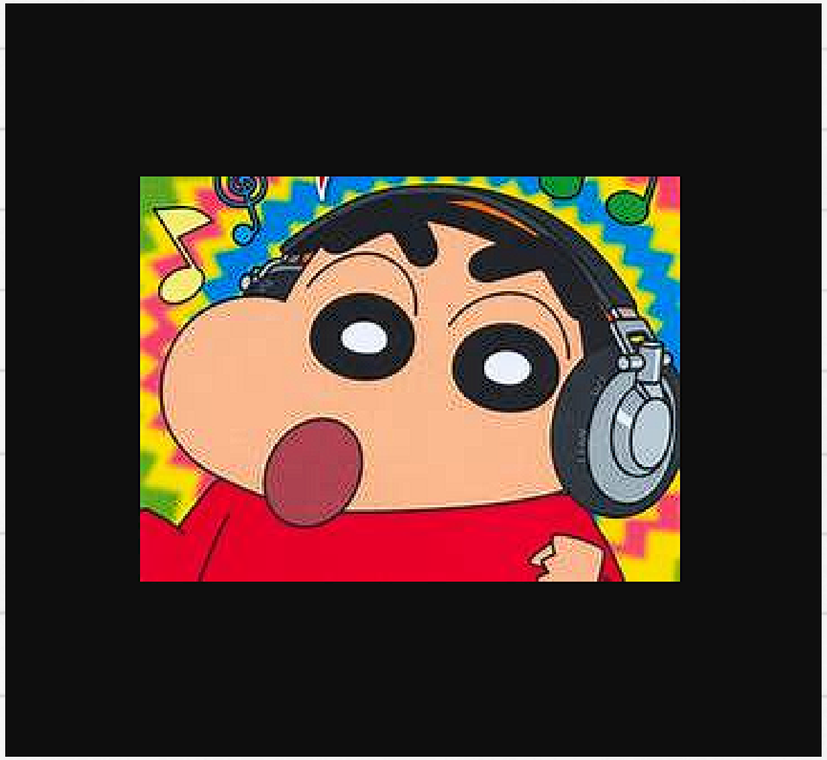
Find objects by prefix

Name	Type
shinchan.jpg	jpg

Select the file and click on the open

Download Open

plicitly grant them permissions. Learn more



THANK YOU VERY MUCH!

Anurag verma
BCA-4A