

# **Project 1**

## **Serverless Image Processing**

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

# AWS Lambda

AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any type of application or backend service without provisioning or managing servers. Lambda functions run on demand i.e. they execute only when needed and you pay only for what you compute. Lambda is well integrated with many other AWS services. It supports a wide variety of programming languages.

Some common use cases for AWS Lambda are:

1. File processing: You can use Lambda for processing files as they are uploaded in an S3 bucket or whenever some event triggers the function.
2. Data and analytics: You can pass a data stream to your Lambda function and then create analysis from that.
3. Website: Lambda can also be used for creating websites. This is cost effective because you are charged only for the time when the servers are running.

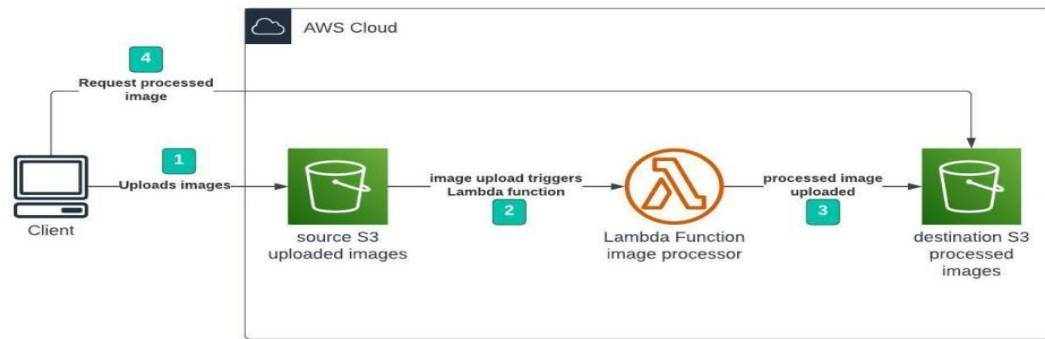
## # Why is AWS Lambda an essential part of the Serverless architecture?

When building Serverless applications, AWS Lambda is one of the main candidates for running the application code. Typically, to complete a Serverless stack you'll need:

- a computing service
- a database service
- an HTTP gateway service

### Serverless Image Processing Flow

- User uploads a file to the source S3 bucket (which is used for storing uploaded images).
- When the image is uploaded to a source S3 bucket, it triggers an event which invokes the Lambda function. The lambda function processes the image.
- Processed image is stored in the destination S3 bucket.



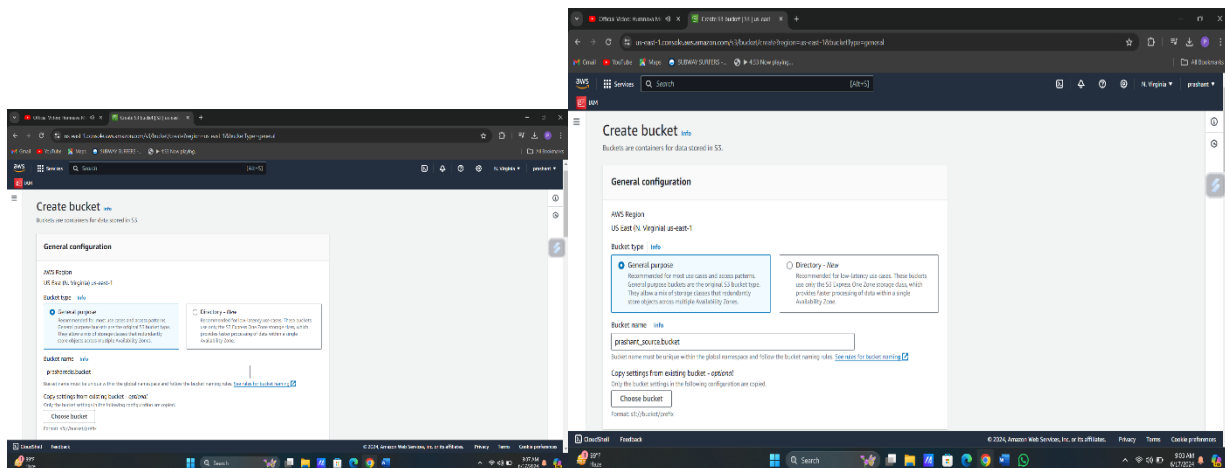
Serverless Image Processor

## Step 1 – Creating S3 buckets

We will use two S3 buckets:

- source Bucket: For storing uploaded images.
- destination Bucket: For storing processed images.

Go to S3 console and click Create bucket. Enter bucket name as ‘serverless-bucket-uploaded-images’. Choose any AWS region as ‘ap-south-1’.



## Step 2 – Configuring S3 bucket policy

In ‘Block Public Access settings for this bucket’ section disable “block all public access”. You will get a warning that the bucket and its objects might become public. Agree to the warning. (Note: we are making this bucket public only for this project, it is not recommended to make an S3 bucket public if not needed).

- ☐ **Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- ☐ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
  - ☐ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
  - ☐ **Block public access to buckets and objects granted through *new* public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
  - ☐ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



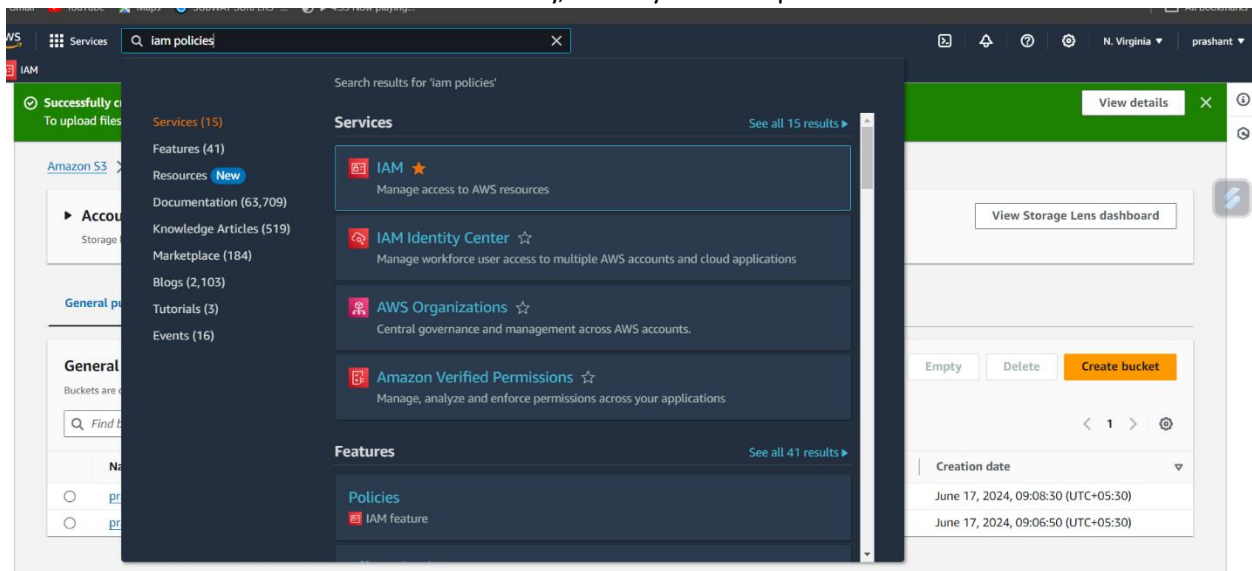
**Turning off block all public access might result in this bucket and the objects within becoming public**

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

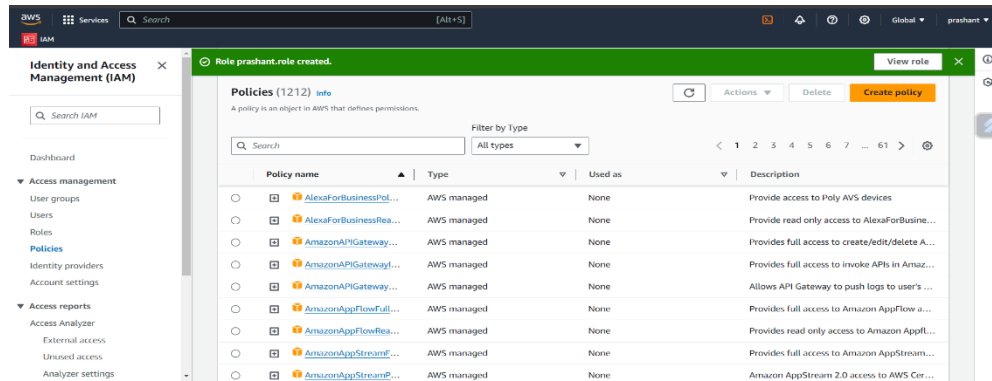
- ☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

## Task 3: Create an IAM Policy

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



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



3. Click on the JSON tab, Remove the existing code and copy-paste the below policy statement into the 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:::BUCKET_NAME/*"
    },
    {
      "Effect": "Allow",

```

```

    "Action": ["s3:PutObject"],
    "Resource": "arn:aws:s3:::DEST_BUCKET/*"
  }
]
}

```

**Specify permissions** [Info](#)

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

**Policy editor** Visual **JSON** Actions ▼

▼ **Select a service**  
Specify what actions can be performed on specific resources in a service.

4. Leave Everything as default and click on Next button.

5. On the Review Policy page:

6. Enter Policy Name and Click on the Create policy button.

**Review and create** [Info](#)

Review the permissions, specify details, and tags.

**Policy details**

**Policy name**  
Enter a meaningful name to identify this policy.  
  
Maximum 128 characters. Use alphanumeric and '+=, @-.' characters.

**Description - optional**  
Add a short explanation for this policy.  
  
Maximum 1,000 characters. Use alphanumeric and '+=, @-.' characters.

7. An IAM Policy with the name prashant.policy is created

**Policies (1207)** [Info](#) Refresh Actions ▼ Delete Create policy

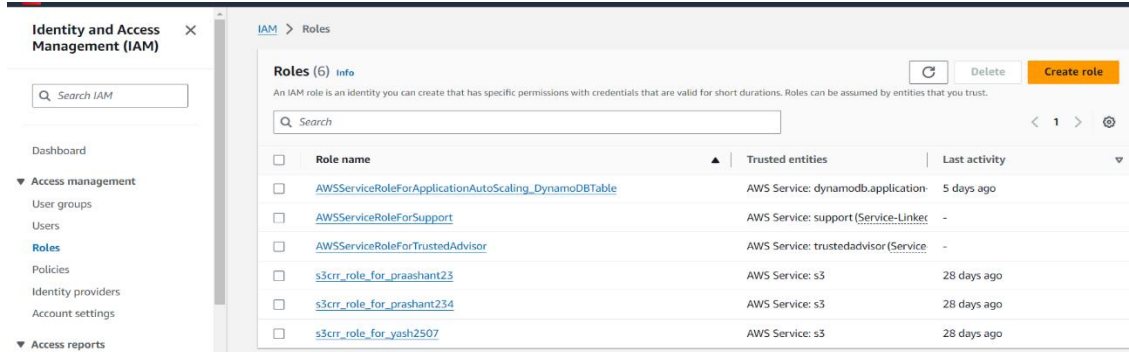
A policy is an object in AWS that defines permissions.

Filter by Type  
 × All types ▼ 1 match < 1 >

	Policy name ▲	Type ▼	Used as ▼	Description
<input type="radio"/>	<a href="#">prashant.policy</a>	Customer managed	None	-

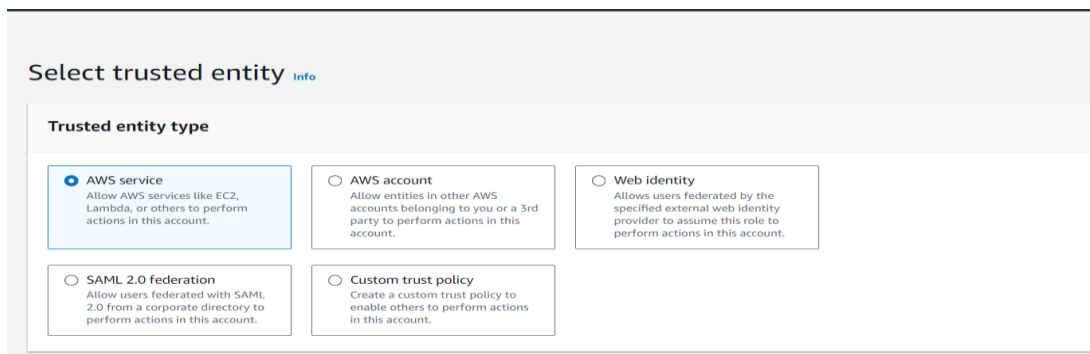
## Task 4: Create an IAM Role

1. In the left menu, click on Roles and click on the Create Role button.

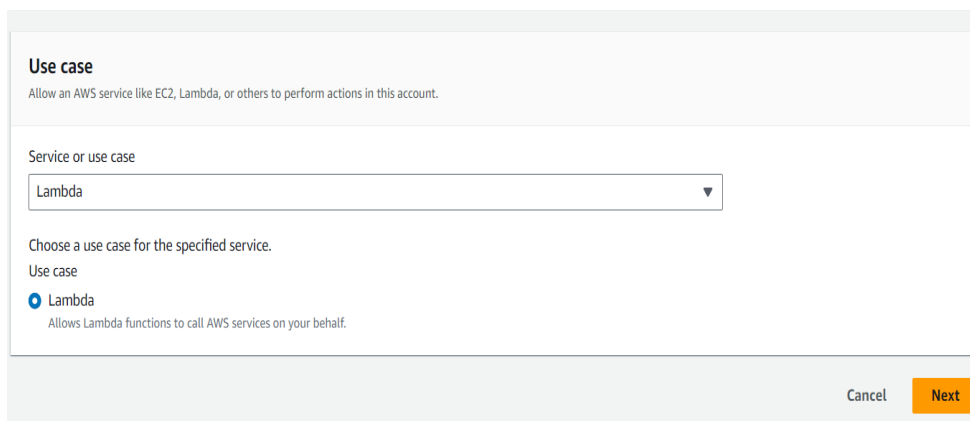


2. Select Lambda from AWS Services list.

- From Trusted Entity Type: Select AWS Service
- From Use case: Select Lambda



- Click on Next button.



4.Role Name: Enter cloudrole

5.Click on the Create Role button

## Name, review, and create

### Role details

#### Role name

Enter a meaningful name to identify this role.

prashant.role

Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters.

#### Description

Add a short explanation for this role.

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: '\_', '+', '@', '-', '/', '[', ']', '!', '\$', '%', '^', '&', '\*', '(', ')', '~', '=', '<', '>'.

## Add permissions [Info](#)

### Permissions policies (1/935) [Info](#)

Choose one or more policies to attach to your new role.

Filter by Type			
Q prashant X		All types ▼	2 matches < 1 > ⚙
<input type="checkbox"/>	Policy name <a href="#">Info</a>	Type	Description
<input checked="" type="checkbox"/>	<a href="#">prashant.policy</a>	Customer managed	-
<input type="checkbox"/>	<a href="#">s3crr_for_prashant234_9bd4c3</a>	Customer managed	-

► Set permissions boundary - optional

Cancel

Previous

Next

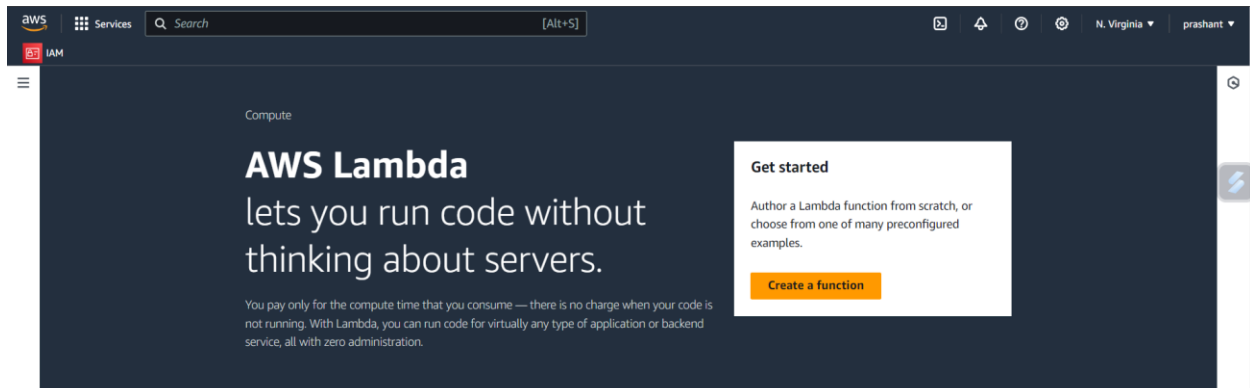
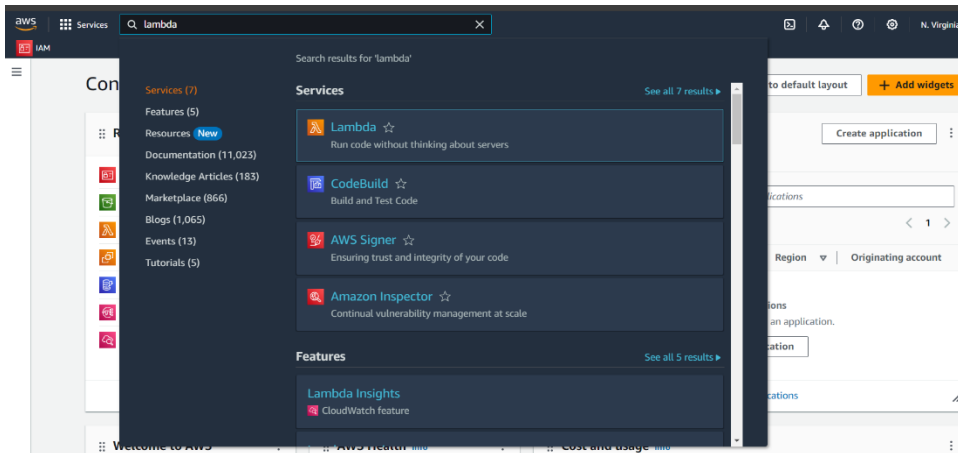
- You have successfully created an IAM role by name prashant.role.

Roles (7) <a href="#">Info</a>			
An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.			
Q Search < 1 > ⚙			
<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	<a href="#">prashant.role</a>	AWS Service: lambda	2 hours ago
<input type="checkbox"/>	<a href="#">prashant123-role-nmma3klm</a>	AWS Service: lambda	Yesterday



## Task 5: Creating Lambda function

1. Go to AWS Lambda Console, Navigate to functions section . Click Create function



2. Name it and select runtime and Leave all other settings as default

A screenshot of the AWS Lambda 'Create function' page. The breadcrumb navigation shows 'Lambda > Functions > Create function'. The main heading is 'Create function' with an 'Info' link. Below this, it says 'Choose one of the following options to create your function.' There are three radio button options: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section contains the following fields: 'Function name' (with the value 'prashantlambda231'), 'Runtime' (with the value 'Node.js 18.x'), and 'Architecture' (with the value 'x86\_64').

3. Change Default execution role and create function

**Permissions** [Info](#)


By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

**Execution role**  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions  
☒ Use an existing role  
☐ Create a new role from AWS policy templates

**Existing role**  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

prashant.role ▼ 

[View the prashant.role role](#) on the IAM console.

► **Advanced settings**

Cancel **Create function**


## Task 6 -upload zip file in Lambda function


[Lambda](#) > [Functions](#) > prashantlambda23

**prashantlambda23** Throttle Copy ARN Actions ▼

▼ **Function overview** [Info](#) Export to Application Composer Download ▼

**Diagram** **Template**


 prashantlambda23
 

 Layers (0)

+ Add trigger + Add destination

**Description**  
-

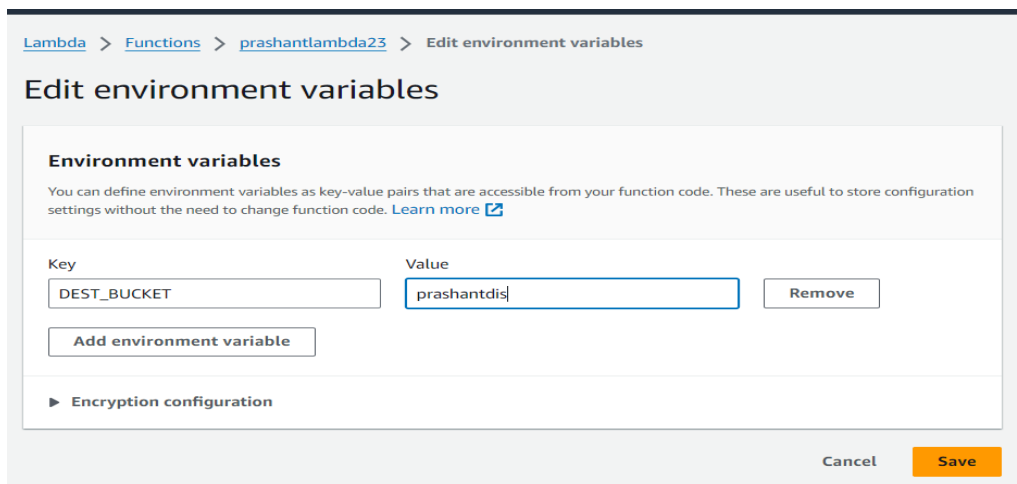
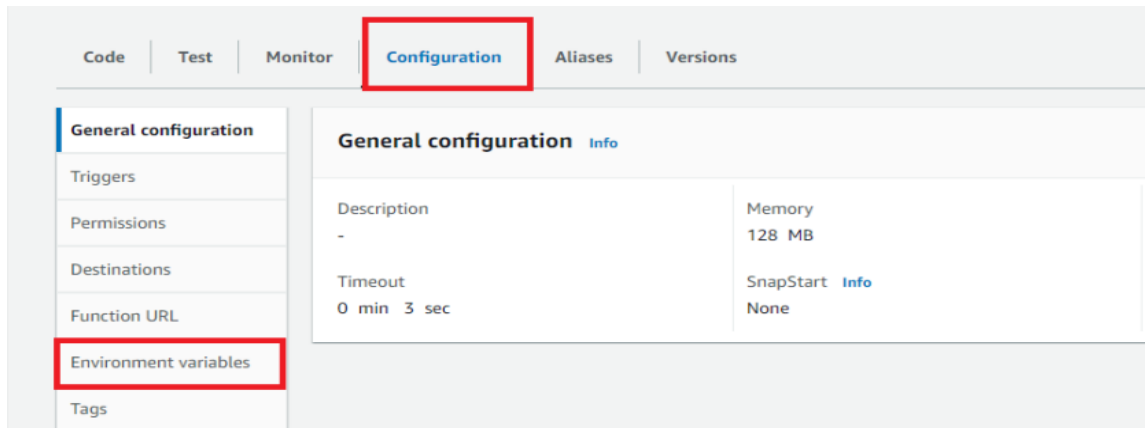
**Last modified**  
13 seconds ago

**Function ARN**  
 arn:aws:lambda:ap-south-1:905418447105:function:prashantlambda23

**Function URL** [Info](#)  
-

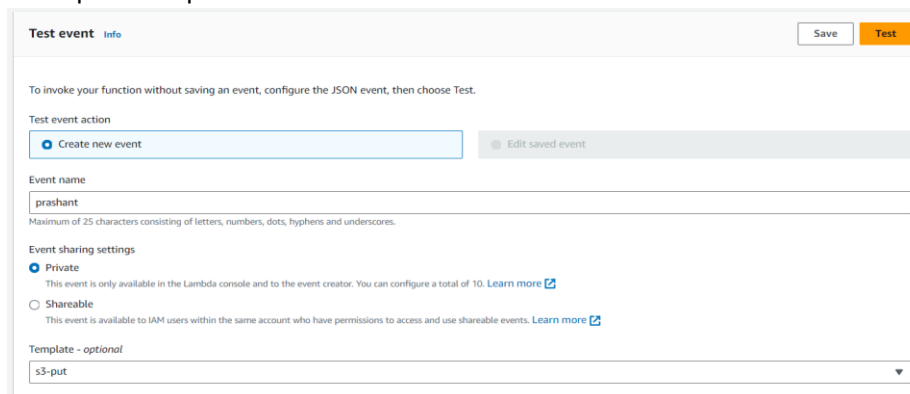
\*Zip file link-<https://github.com/OneLightWebDev/image-resizer-lambda>

4.Edit Environment Variable



## Task 7: Test Lambda Function

- \*Go to AWS Lambda console. Navigate to Functions section.
- \*open function then will be created
- \*open test console
- \*template=s3-put



```
{  
  "Records": [  
    {  
      "event": {  
        "name": "prashant"  
      },  
      "id": "1"  
    }  
  ]  
}
```

```

{
  "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": "prashantsource",
      "ownerIdentity": {
        "principalId": "EXAMPLE"
      },
      "arn": "arn:aws:s3:::prashantsource"
    },
    "object": {
      "key": "test%2Fkey",
      "size": 1024,
      "eTag": "0123456789abcdef0123456789abcdef",
      "sequencer": "0A1B2C3D4E5F678901"
    }
  }
}
]
}

```

**Now We can Test:**

Code

Test

Monitor

Configuration

Aliases

Versions

✓

Executing function: succeeded ([logs](#))

► Details

Test event

Info

Save

Test

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

Create new event

Edit saved event

## Task 8: Creating S3 Trigger

- \*Add trigger
- \*Select s3
- \*choose source Bucket name
- \*Now Add

Lambda

>

Add triggers

## Add trigger

Trigger configuration

Info

S3

aws

asynchronous

storage

▼

Bucket

Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

Q s3/prashantsource

×

↺

Bucket region: ap-south-1

Event types

Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

▼

All object create events

×

Prefix - optional

Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

## Task 9: Upload image in Source Bucket

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

Objects (2) Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 >

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">mountain.jpeg</a>	jpeg	June 19, 2024, 22:48:37 (UTC+05:30)	1.8 MB	Standard
<input checked="" type="checkbox"/>	<a href="#">tiger.jpg</a>	jpg	June 19, 2024, 22:48:38 (UTC+05:30)	86.8 KB	Standard

Original Image

A photograph of a young tiger cub with orange fur and black stripes, sitting on a large, grey, textured rock. The cub is looking directly at the camera with a curious expression. The background is a soft-focus natural setting with green foliage and some dry grass.

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

Objects (2) Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 >

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">mountain.jpeg</a>	jpeg	June 19, 2024, 22:50:49 (UTC+05:30)	1.8 MB	Standard
<input checked="" type="checkbox"/>	<a href="#">tiger.jpg</a>	jpg	June 19, 2024, 22:50:50 (UTC+05:30)	86.8 KB	Standard

Resize Image

