

# Instructions for creating AWS lambda function for caching the images.

## 1. Creating IAM Role:

Before creating a lambda function we need to create an IAM Role for our lambda function to get full access to S3.

Roles:

- AWSLambdaBasicExecutionRole
- AmazonS3FullAccess

### Select trusted entity [Info](#)

**Trusted entity type**

☒ **AWS service**  
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

☐ **AWS account**  
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

Lambda

Choose a use case for the specified service.

Use case

☒ **Lambda**  
Allows Lambda functions to call AWS services on your behalf.

Cancel

Next

### Add permissions [Info](#)

**Permissions policies (1/883) [Info](#)**  
Choose one or more policies to attach to your new role.

Q S3

Filter by Type

All types

9 matches

<input type="checkbox"/>	Policy name <a href="#">✕</a>	Type	Description
<input type="checkbox"/>	<a href="#">AmazonDMSRedshiftS3Role</a>	AWS managed	Provides access to man
<input checked="" type="checkbox"/>	<a href="#">AmazonS3FullAccess</a>	AWS managed	Provides full access to .
<input type="checkbox"/>	<a href="#">AmazonS3ObjectLambdaExecutionRolePolicy</a>	AWS managed	Provides AWS Lambda
<input type="checkbox"/>	<a href="#">AmazonS3OutpostsFullAccess</a>	AWS managed	Provides full access to .

Q AwsLambda

Filter by Type

All types

13 matches

<input type="checkbox"/>	Policy name <a href="#">✕</a>	Type	Description
<input type="checkbox"/>	<a href="#">AWSLambda_FullAccess</a>	AWS managed	Grants full access to A
<input type="checkbox"/>	<a href="#">AWSLambda_ReadOnlyAccess</a>	AWS managed	Grants read-only acces
<input checked="" type="checkbox"/>	<a href="#">AWSLambdaBasicExecutionRole</a>	AWS managed	Provides write permiss
<input type="checkbox"/>	<a href="#">AWSLambdaBasicExecutionRole-f91af250-1dee-445e-...</a>	Customer managed	-
<input type="checkbox"/>	<a href="#">AWSLambdaDynamoDBExecutionRole</a>	AWS managed	Provides list and read .
<input type="checkbox"/>	<a href="#">AWSLambdaENIManagementAccess</a>	AWS managed	Provides minimum nei

After configuring the permission our role should look like this. Give it the name you want and hit create role to finish the process.

### Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

aws-lambda-s3-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 alphanumeric and '+=, @-\_' characters.

Step 1: Select trusted entities

Edit

Trust policy

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": {
10        "Service": [
11          "lambda.amazonaws.com"
12        ]
13      }
14    ]
15  }
16 }
```

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy
<a href="#">AWSLambdaBasicExecutionRole</a>	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create role

## 2. Creating lambda function

Step 1:

Here we are using the role that we created. Give it a name of your choice and make sure lambda has the same configuration as follows.

### Create function [Info](#)

AWS Serverless Application Repository applications have moved to [Create application](#).

☒ **Author from scratch**  
Start with a simple Hello World example.

☐ **Use a blueprint**  
Build a Lambda application from sample code and configuration presets for common use cases.

☐ **Container image**  
Select a container image to deploy for your function.

#### Basic information

**Function name**  
Enter a name that describes the purpose of your function.  
  
Use only letters, numbers, hyphens, or underscores with no spaces.

**Runtime** [Info](#)  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  

▼

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  

☒ x86\_64

☐ arm64

**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.  

▼

[View the aws-lambda-s3 role](#) on the IAM console.

## Function overview.


✓ Successfully created the function **cacheImageFroms3Lambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".


[Lambda](#) > [Functions](#) > cacheImageFroms3Lambda

### cacheImageFroms3Lambda

Throttle Copy ARN Actions

▼ **Function overview** Info

**cacheImageFroms3Lambda**

Layers (0)

+ Add trigger + Add destination

Description  
-

Last modified  
7 seconds ago

Function ARN  
arn:aws:lambda:us-east-1:612821522863:function:cacheImageFroms3Lambda

Function URL [Info](#)  
-

Code Test Monitor Configuration Aliases Versions

Add our zipped code from <https://github.com/Tsolmonx/resize-image-lambda-php> using command

```
zip -r9q index.zip .
```

Code source Info

Upload from ▲  
.zip file  
Amazon S3 location

File Edit Find View Go Tools Window Test Deploy

Go to Anything (% P)

Environment

cacheImageFroms3Lambda

index.mjs

```
1 export const handler = async (event) => {
2   // TODO implement
3   const response = {
4     statusCode: 200,
5     body: JSON.stringify('Hello from Lambda!'),
6   };
7   return response;
8 };
9
```

### 3. Runtime settings

Edit Runtime settings as follows. Index.php is our entry point.

[Lambda](#) > [Functions](#) > [cacheImageFroms3](#) > Edit runtime settings

## Edit runtime settings

**Runtime settings** [Info](#)

**Runtime**  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  

Custom runtime on Amazon Linux 2

↻

**Handler** [Info](#)  

index.php

**Architecture** [Info](#)  
Choose the instruction set architecture you want for your function code.  

☒ x86\_64

☐ arm64

### 4. Layers

We need two layers for our function to work properly.

1: arn:aws:lambda:us-east-1:534081306603:layer:php-80:62

2: arn:aws:lambda:us-east-1:403367587399:layer:gd-php-80:22

[Lambda](#) > [Layers](#) > Add layer

## Add layer

**Function runtime settings**

Runtime Custom runtime on Amazon Linux 2	Architecture x86_64
---------------------------------------------	------------------------

**Choose a layer**

**Layer source** [Info](#)  
Choose from layers with a compatible runtime and instruction set architecture or specify the Amazon Resource Name (ARN) of a layer version. You can also [create a new layer](#).

☐ **AWS layers**  
Choose a layer from a list of layers provided by AWS.

☐ **Custom layers**  
Choose a layer from a list of layers created by your AWS account or organization.

☒ **Specify an ARN**  
Specify a layer by providing the ARN.

**Specify an ARN**  
Specify a layer by providing the Amazon Resource Name (ARN).  

arn:aws:lambda:region:12345678901:layer:example:1

Verify

Cancel

Add

# Configuration overview

Code

Test

Monitor

Configuration

Aliases

Versions

Code source

Info

Upload from ▼

ⓘ

The deployment package of your Lambda function "cachelmaImageFroms3" is too large to enable inline code editing. However, you can still invoke your function.

Code properties

Info

Package size 5.0 MB	SHA256 hash 📄 zit37ZXtUZTHoXrs4BwvYfDLCphAtfdKh UblnU8oAvM=	Last modified October 11, 2023 at 12:41 PM GMT+8
------------------------	-------------------------------------------------------------------	-----------------------------------------------------

Runtime settings

Info

Edit

Edit runtime management configuration

Runtime Custom runtime on Amazon Linux 2	Handler <div>Info</div> index.php	Architecture <div>Info</div> x86_64
---------------------------------------------	--------------------------------------	----------------------------------------

▶ Runtime management configuration

Layers

Info

Edit

Add a layer


Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
1	php-80	62	provided.al2	-	arn:aws:lambda:us-east-1:
2	gd-php-80	22	provided	-	arn:aws:lambda:us-east-1:



## 5. Adding trigger to our function

Whenever images are uploaded to S3 we will resize the image to 3 different sizes and upload it back to S3. To prevent an infinite loop when resizing and uploading it back to S3 we need to be careful and add a prefix for our trigger(uploads/). Adding this prefix will ensure our function works it's intended way and our resized image will be added in cache/ directory. And provide the bucket and subscribe to All object create event.


### Add trigger

**Trigger configuration** [Info](#)

 **S3**  
aws asynchronous storage


**Bucket**  
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.  
  
  
 **Invalid input.**


**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.

**Suffix - optional**  
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

**Recursive invocation**  
If your function writes objects to an S3 bucket, ensure that you are using different S3 buckets for input and output. Writing to the same bucket increases the risk of creating a recursive invocation, which can result in increased Lambda usage and increased costs. [Learn more](#)   
☒ I acknowledge that using the same S3 bucket for both input and output is not recommended and that this configuration can cause recursive invocations, increased Lambda usage, and increased costs.

Lambda will add the necessary permissions for AWS S3 to invoke your Lambda function from this trigger. [Learn more](#)  about the Lambda permissions model.

[Cancel](#) [Add](#)







Here in cache/ directory we can see our lambda function created our desired resized images to its respective folders.

[Amazon S3](#) > [Buckets](#) > [dev-tsomo](#) > cache/

cache/

Objects

Properties

Objects (3)

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

Copy S3 URI

Copy URL

Download

Open

Delete

Actions ▼

Create folder

Find objects by prefix

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size
<input type="checkbox"/>	large/	Folder	-	
<input type="checkbox"/>	medium/	Folder	-	
<input type="checkbox"/>	thumbnail/	Folder	-	

uwp3797360.webp [Info](#)

Copy S3 URI

Download

Open

Object ac

Properties

Permissions

Versions

Object overview

Owner

tsolmondark

AWS Region

US East (N. Virginia) us-east-1

Last modified

October 11, 2023, 13:49:45 (UTC+08:00)

Size

107.7 KB

Type

webp

Key

cache/medium/uwp3797360.webp

S3 URI

s3://dev-tsomo/cache/medium/uwp3797360.webp

Amazon Resource Name (ARN)

arn:aws:s3:::dev-tsomo/cache/medium/uwp3797360.webp

Entity tag (Etag)

48979daf720475070e495443f5eb1e42

Object URL

<https://dev-tsomo.s3.amazonaws.com/cache/medium/uwp3797360.webp>