LAMBDA DOCUMENTATION

What is AWS Lambda?

AWS Lambda is a serverless compute service provided by AWS. It lets you run code without provisioning or managing servers. You write your code, upload it, and Lambda handles the rest.

Key Features

- Serverless: No need to manage infrastructure.
- Event-driven: Triggered by services like S3, API Gateway, DynamoDB.
- Scalable: Automatically scales with demand.
- Pay-as-you-go: Pay only for compute time used.

Multi-language support: Node.js, Python, Java, C#, Go, Ruby, etc.

LAMBDA DOCUMENTATION

Components of Lambda

1. Function - Your code packaged in a ZIP or container image.
2. Trigger - Event sources like S3, API Gateway.
3. Execution Role - IAM permissions.
4. Runtime - The language environmcent.
5. Environment Variables - Store configuration data.

How Lambda Works (Step-by-Step)

1. Create a Lambda function.
2. Write or upload your code.
3. Set a trigger (e.g., S3, API Gateway).
4. Lambda waits for the event.
5. Lambda runs the code when triggered.
6. Returns output and stops.

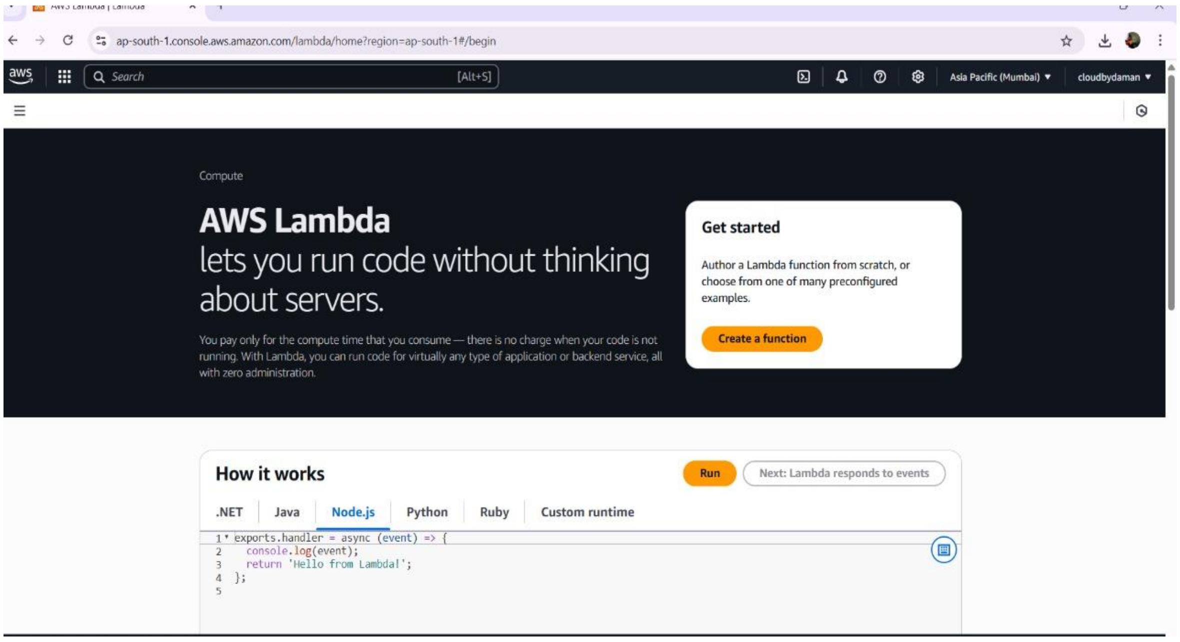
Example Use Cases

- Image processing on S3 uploads
- Real-time data transformation
- Web/mobile app backends
- Chatbots or Alexa skills

LAMBDA DOCUMENTATION

Creating a Lambda Function

1. Go to AWS Console > Lambda



AWS Lambda lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

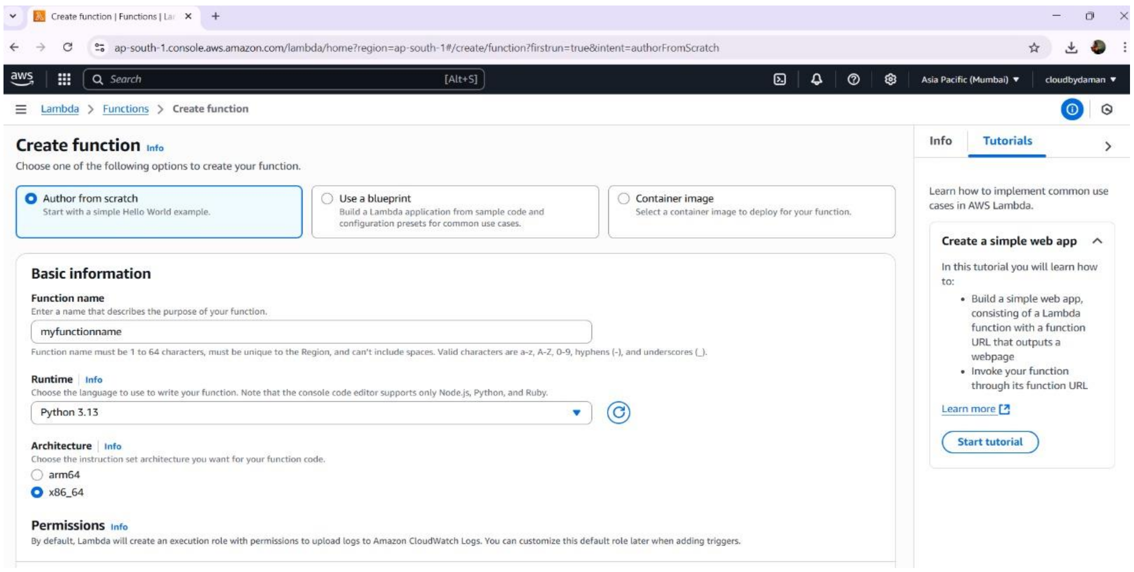
```
exports.handler = async (event) => {
  console.log(event);
  return 'Hello from Lambda!';
};
```

2. Click 'Create Function'

3. Choose 'Author from scratch'

LAMBDA DOCUMENTATION

Enter name, runtime, and role


ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/create/function?firstrun=true&intent=authorFromScratch

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

- Author from scratch — Start with a simple Hello World example. (selected)
- 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.
myfunctionname
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture Info
Choose the instruction set architecture you want for your function code.
arm64 [✓ x86_64]

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.

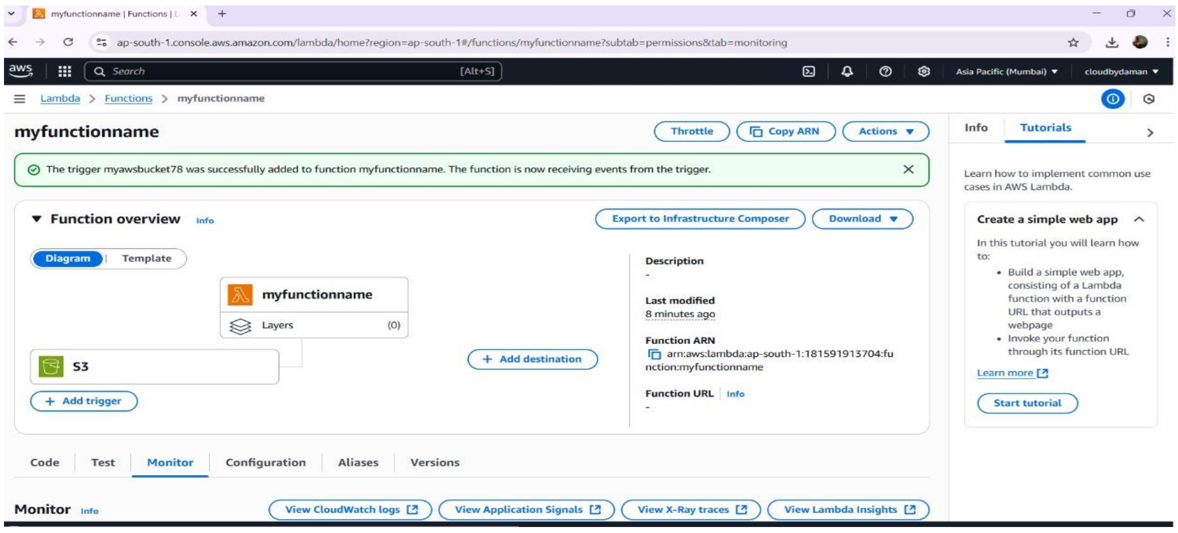
Info | Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app
In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more
Start tutorial

4. Write code in editor


ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions/myfunctionname?subtab=permissions&tab=monitoring

Lambda > Functions > myfunctionname

myfunctionname Throttle | Copy ARN | Actions

The trigger myawsbucket78 was successfully added to function myfunctionname. The function is now receiving events from the trigger.

Function overview Info — Export to Infrastructure Composer | Download

Diagram | Template

myfunctionname — Layers (0)
S3
+ Add trigger
+ Add destination

Description
-

Last modified
8 minutes ago

Function ARN
arn:aws:lambda:ap-south-1:181591913704:function:myfunctionname

Function URL Info
-

Code | Test | Monitor | Configuration | Aliases | Versions

Monitor Info — View CloudWatch logs | View Application Signals | View X-Ray traces | View Lambda Insights

Info | Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app
In this tutorial you will learn how to:
- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more
Start tutorial

5. Set trigger source

6. Click 'Deploy' and test

LAMBDA DOCUMENTATION

Sample Code (Python)

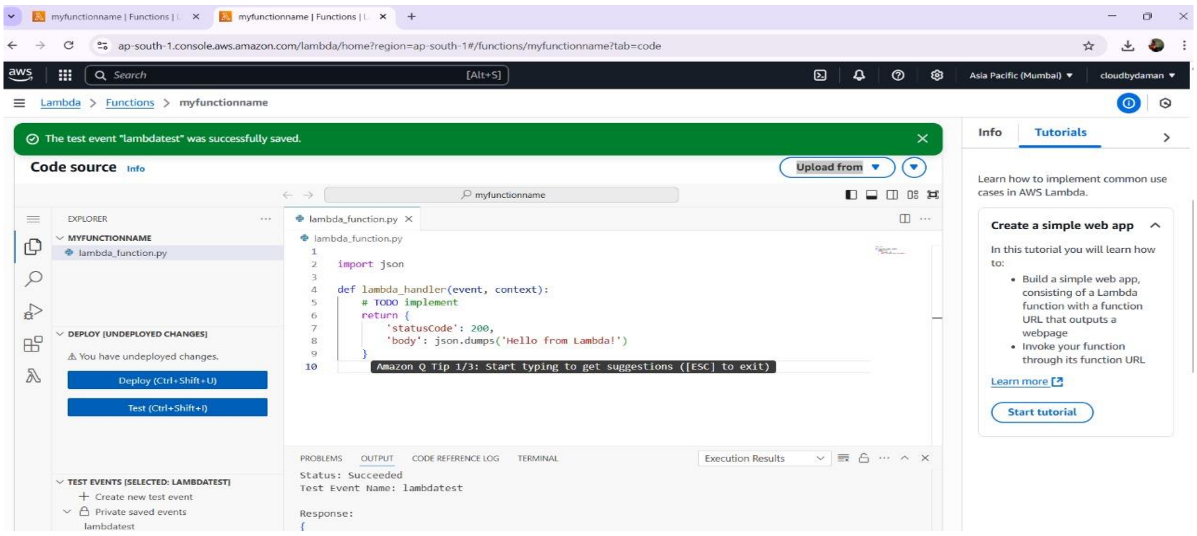
```
def lambda_handler(event, context):

    return {

        'statusCode': 200,

        'body': 'Hello from Lambda!

}
```



The test event "lambdatest" was successfully saved.

Code source

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)

DEPLOY [UNDEPLOYED CHANGES]

⚠ You have undeployed changes.

```
Status: Succeeded
Test Event Name: lambdatest

Response:
{
```

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

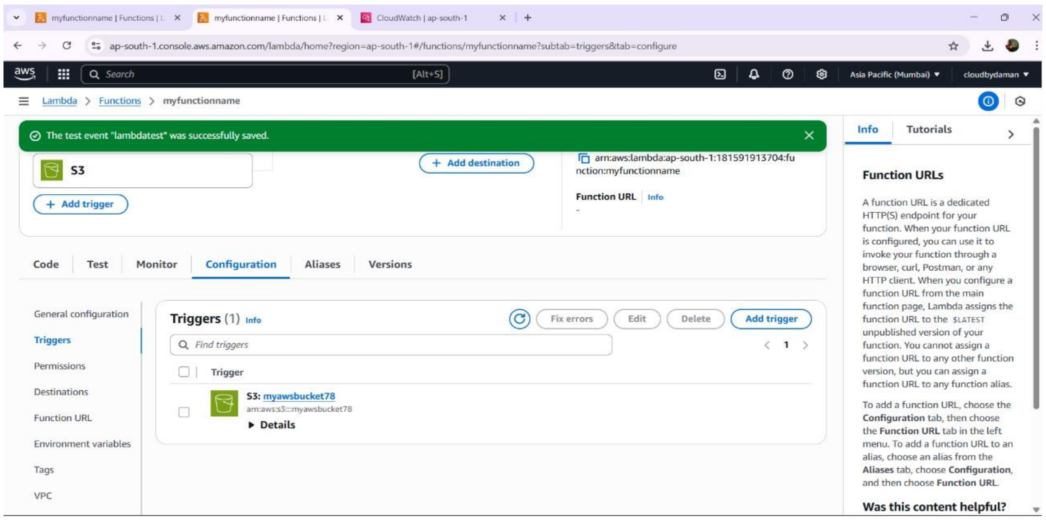
7. Go to the ‘configuration’ tab.

8. In the ‘Triggers’ section , click “Add trigger”.

9. Select any service like S3,API etc.

LAMBDA DOCUMENTATION

(Make sure if you add S3 in the trigger then, S3 bucket want to be created).


ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions/myfunctionname?subtab=triggers&tab=configure

Lambda > Functions > myfunctionname

The test event "lambdatest" was successfully saved.

S3

+ Add trigger

+ Add destination

arn:aws:lambda:ap-south-1:181591913704:function:myfunctionname

Function URL Info

-

Code | Test | Monitor | Configuration | Aliases | Versions

General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC

Triggers (1) Info — Fix errors | Edit | Delete | Add trigger

Find triggers

Trigger
S3: myawsbucket78 arn:aws:s3:::myawsbucket78 ▶ Details

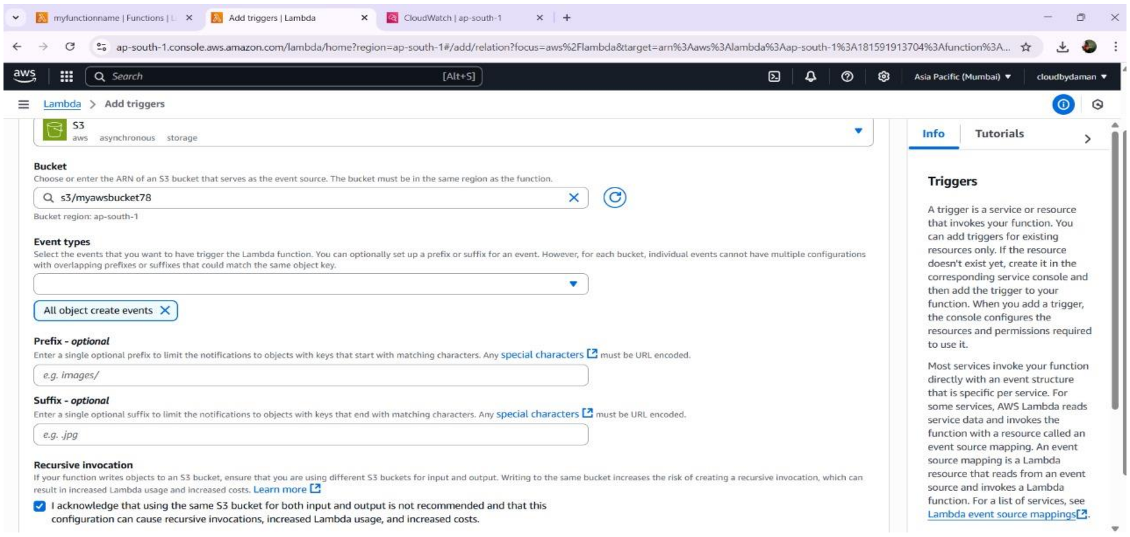
Function URLs

A function URL is a dedicated HTTP(S) endpoint for your function. When your function URL is configured, you can use it to invoke your function through a browser, curl, Postman, or any HTTP client. When you configure a function URL from the main function page, Lambda assigns the function URL to the $LATEST unpublished version of your function. You cannot assign a function URL to any other function version, but you can assign a function URL to any function alias.

To add a function URL, choose the **Configuration** tab, then choose the **Function URL** tab in the left menu. To add a function URL to an alias, choose an alias from the **Aliases** tab, choose **Configuration**, and then choose **Function URL**.

Was this content helpful?

10. Configuration the trigger and save.


ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/add/relation?focus=aws%2Flambda&target=arn%3Aaws%3Alambda%3Aap-south-1%3A181591913704%3Afunction%3A...

Lambda > Add triggers

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.
s3/myawsbucket78
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. Any special characters must be URL encoded.
e.g. images/

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters. Any special characters must be URL encoded.
e.g. .jpg

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.

Triggers

A trigger is a service or resource that invokes your function. You can add triggers for existing resources only. If the resource doesn't exist yet, create it in the corresponding service console and then add the trigger to your function. When you add a trigger, the console configures the resources and permissions required to use it.

Most services invoke your function directly with an event structure that is specific per service. For some services, AWS Lambda reads service data and invokes the function with a resource called an event source mapping. An event source mapping is a Lambda resource that reads from an event source and invokes a Lambda function. For a list of services, see Lambda event source mappings.

Related Services

S3 - Trigger Lambda on file upload

API Gateway - Create REST/HTTP APIs

DynamoDB - Trigger on DB changes

CloudWatch - Logging and monitoring

LAMBDA DOCUMENTATION

Best Practices

- Keep functions short and focused
- Use environment variables
- Monitor via CloudWatch
- Handle exceptions and log errors
- Set proper timeouts and memory