

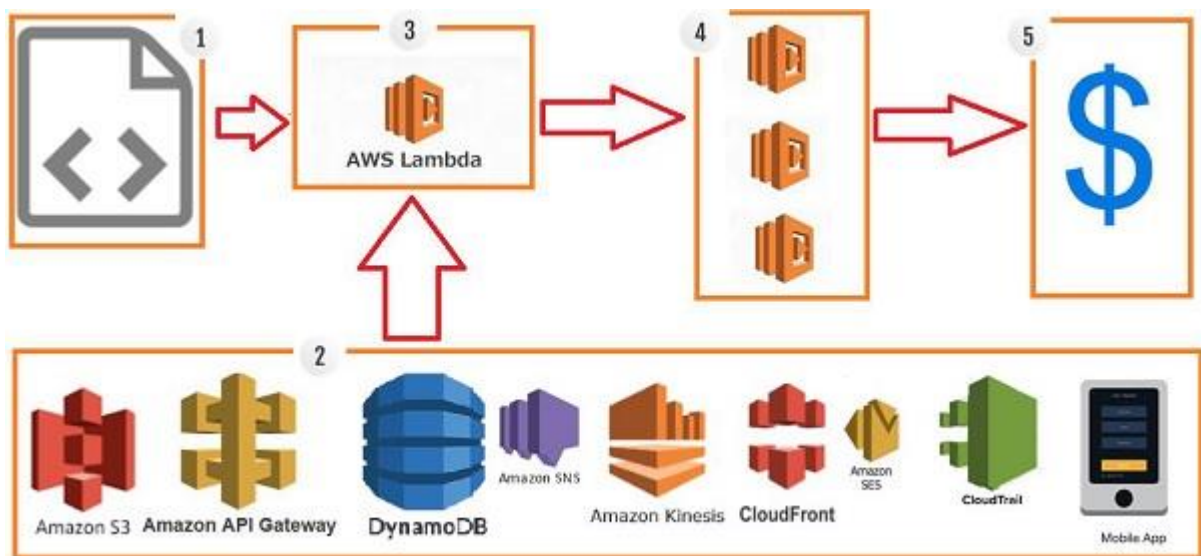
# What is AWS Lambda?

Definition of AWS Lambda as given by its official documentation is as follows –

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically, from a few requests per day to thousands per second. You pay only for the compute time you consume - there is no charge when your code is not running.

## How AWS Lambda Works?

The block diagram that explains the working of AWS Lambda in five easy steps is shown below –



**Step 1** – Upload AWS lambda code in any of languages AWS lambda supports, that is NodeJS, Java, Python, C# and Go.

**Step 2** – These are few AWS services on which AWS lambda can be triggered.

**Step 3** – AWS Lambda which has the upload code and the event details on which the trigger has occurred. For example, event from Amazon S3, Amazon API Gateway, Dynamo dB, Amazon SNS, Amazon Kinesis, CloudFront, Amazon SES, CloudTrail, mobile app etc.

**Step 4** – Executes AWS Lambda Code only when triggered by AWS services under the scenarios such as –

- User uploads files in S3 bucket
- http get/post endpoint URL is hit
- data is added/updated/deleted in dynamo dB tables
- push notification
- data streams collection
- hosting of website
- email sending

- mobile app, etc.

**Step 5** – Remember that AWS charges only when the AWS lambda code executes, and not otherwise.

### Advantages of using AWS Lambda

AWS Lambda offers multiple benefits when you are working on it. This section discusses them in detail –

#### Ease of working with code

AWS Lambda gives you the infrastructure to upload your code. It takes care of maintaining the code and triggers the code whenever the required event happens. It allows you to choose the memory and the timeout required for the code.

AWS Lambda can also execute parallel requests as per the event triggers.

#### Log Provision

AWS Lambda gives the details of number of times a code was executed and time taken for execution, the memory consumed etc. AWS CloudWatch collects all the logs, which helps in understanding the execution flow and in the debugging of the code.

#### Billing based on Usage

AWS Lambda billing is done on memory usage, request made and the execution, which is billed in increments of minimum 100ms. So for a 500ms execution, the billing will be after every 100ms. If you specify your AWS lambda code to be executed in 500ms and the time taken to execute is just 200ms, AWS will bill you only for the time taken, that is 200ms of execution instead of 500ms. AWS always charges for the execution time used. You need not pay if the function is not executed.

#### Multi Language Support

AWS Lambda supports popular languages such as Node.js, Python, Java, C# and Go. These are widely used languages and any developer will find it easy to write code for AWS Lambda.

#### Ease of code authoring and deploying

There are many options available for Lambda for authoring and deploying code. For writing your code, you can use AWS online editor, Visual Studio IDE, or Eclipse IDE. It also has support for serverless framework which makes writing and deploying of AWS Lambda code easy. Besides AWS console, we have AWS-cli to create and deploy code.

#### Other features

You can use AWS Lambda for free by getting a login to AWS free tier. It gives you service for free for 1 year. Take a look at the free services offered by AWS free tier.

## Disadvantages of using AWS Lambda

In spite of many advantages, AWS Lambda possesses the following disadvantages –

- It is not suitable for small projects.
- You need to carefully analyze your code and decide the memory and timeout. In case if your function needs more time than what is allocated, it will get terminated as per the timeout specified on it and the code will not be fully executed.
- Since AWS Lambda relies completely on AWS for the infrastructure, you cannot install anything additional software if your code demands it.

## Events that Trigger AWS Lambda

The events can trigger AWS Lambda are as follows –

- Entry into a S3 object
- Insertion, updation and deletion of data in Dynamo DB table
- Push notifications from SNS
- GET/POST calls to API Gateway
- Header's modification at viewer or origin request/response in CloudFront
- Log entries in AWS Kinesis data stream
- Log history in CloudTrail

## Use Cases of AWS Lambda

AWS Lambda is a compute service mainly used to run background processes. It can trigger when used with other AWS services. The list of AWS services where we can use AWS Lambda is given below –

### S3 Object and AWS Lambda

Amazon S3 passes the event details to AWS Lambda when there is any file upload in S3. The details of the file upload or deletion of file or moving of file is passed to the AWS Lambda. The code in AWS Lambda can take the necessary step for when it receives the event details. For Example creating thumbnail of the image inserted into S3.

### DynamoDB and AWS Lambda

DynamoDB can trigger AWS Lambda when there is data added, updated and deleted in the table. AWS Lambda event has all the details of the AWS DynamoDB table about the insert /update or delete.

### API Gateway and AWS Lambda

API Gateway can trigger AWS Lambda on GET/POST methods. We can create a form and share details with API Gateway endpoint and use it with AWS Lambda for further processing, for Example, making an entry of the data in DynamoDB table.

### SNS and AWS Lambda

SNS is used for push notification, sending SMS etc. We can trigger AWS lambda when there is any push notification happening in SNS. We can also send SMS to the phone number from AWS Lambda when it receives the trigger.

#### Scheduled Events and AWS Lambda

Scheduled Events can be used for cron jobs. It can trigger AWS Lambda to carry out the task at regular time pattern.

#### CloudTrail and AWS Lambda

CloudTrail can be helpful in monitoring the logs on the account. We can use AWS Lambda to further process the CloudTrail logs.

#### Kinesis and AWS Lambda

Kinesis is used to capture/store real time tracking data coming from website clicks, logs, social media feeds and a trigger to AWS Lambda can do additional processing on this logs.

#### CloudFront and Lambda@Edge

CloudFront is a content delivery network where you can host your website and Lambda@Edge can be used to process the headers coming from viewer request, origin request, origin response and viewer response. The headers modification includes tasks such as modifying cookie data, URL rewrite, used for AB testing to change the response send to the user back, adding extra headers info for security purpose etc.