

Server Less Computing with AWS: Exploring the Power of Lambda and Step Functions

Punam Toke*, Aradhya Desai

Department of Computer Science, Dr. D. Y. Patil, Arts, Commerce & Science College Pimpri, Pune -18,
Maharashtra, India

DOI: <https://doi.org/10.51583/IJLTEMAS.2025.1413SP020>

Received: 26 June 2025; Accepted: 30 June 2025; Published: 23 October 2025

Abstract — A recent approach to creating and launching applications without managing servers is server less computing. This study examines how AWS Lambda and AWS Step Functions, two essential services, allow server less computing. While Step Functions assist in coordinating several processes into an easy workflow, AWS Lambda enables developers to execute code in response to events without worrying about infrastructure. When combined, they facilitate the development of applications that are scalable, effective, and economical. In this article, we describe the operation of these services, their advantages, and practical applications that demonstrate their value in resolving issues in a timely and dependable manner.

Keywords — Server less Computing, Auto Scaling, Cloud computing, High Availability, Pay-per-use pricing

I. Introduction

What is Server Less Computing?

A cloud-based execution model known as server less computing does away with the requirement to create or maintain infrastructure by allowing cloud providers to control and allocate resources for running applications. Server less functions operate on demand in response to events, scaling automatically based on workload, in contrast to traditional computing models where applications run on pre-allocated servers or containers.

Cloud computing has transformed the development and operation of apps in recent years. Server less computing, a paradigm that frees developers from managing underlying servers or infrastructure so they can concentrate on creating code, is one of the most significant changes in this field. Developers just deploy their code; the cloud provider takes care of the rest, including fault tolerance, scalability, and availability, instead of procuring or maintaining servers.

One of the biggest suppliers of cloud solutions, Amazon Web Services (AWS), provides a number of technologies that facilitate server less designs. AWS Lambda Functions and AWS Lambda are two of the most famous and effective products in this industry. Without having to worry about server management, AWS Lambda allows customers to execute code in response to events like HTTP requests, file uploads, or database updates. However, by arranging several AWS services and Lambda functions in a reliable order, AWS Step Functions offer a visual method of building and overseeing processes.

The basic ideas, characteristics, and real-world uses of AWS Lambda and Step Functions are examined in this paper. It emphasizes how these services allow scalable, event-driven architectures, lower operating costs, and streamline application development. The purpose of the article is to demonstrate the strength and adaptability of server less architecture on AWS by examining actual use cases and industry best practices.

AWS Lambda is a service that lets you run your code without managing servers. It automatically runs your code when something happens—like a user visiting a website, a database being updated, or a system changing state. You just upload your code (as a .zip file or a container), and AWS takes care of everything else, like setting up servers, scaling to handle traffic, applying security updates, and monitoring performance. Lambda is great for developers who want to focus on writing code and not worry about the technical setup.

In this article, we'll look at what AWS Lambda does, its main features, how it's priced, and some real-world examples of how it's used.

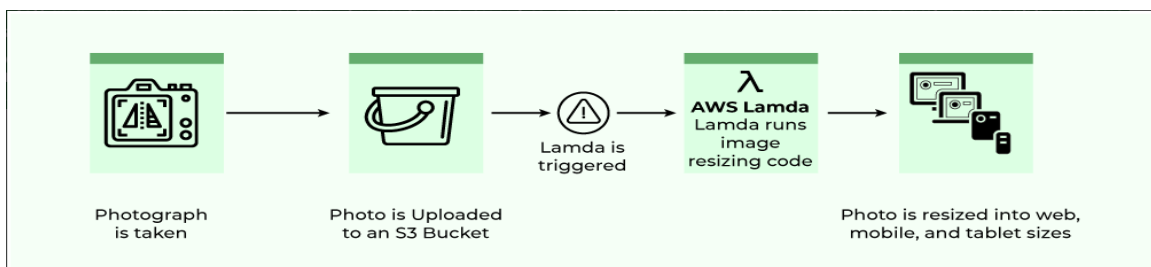


Figure 2 – AWS Lambda

What are Lambdas Functions?

AWS Lambda is a server less computing service managed by AWS. It allows developers to execute code without having to configure or maintain servers. You just upload your code as a ZIP file, and AWS Lambda runs it for you. According to the level of demand, Lambda will scale your functions up or down. It's mostly used for event-driven tasks, like processing data from Amazon S3 buckets or handling HTTP requests.

Example:

- Get data from Amazon S3 storage.
- Answer requests sent through the internet.
- Create apps that don't need a traditional server to run.

AWS Lambda Function Use Cases

There are many ways to activate the lambda, a few which are listed here.

File Processing: Simple storage services (S3) can be used to start AWS lambda. When a file is added to the S3 service, Lambda data processing begins.

Web Apps: AWS Lambda, which grows up and down in response to incoming traffic, may be combined with web apps.

Internet of Things (IoT) Applications: When processing data from devices connected to IOT applications, you can use the AWS lambda depending on specific criteria. It will examine the information obtained from the Internet of Things application.

Stream Processing: Real-time streaming data can be processed for log filtering, application tracking, and other reasons by combining Lambda functions with Amazon Kinesis.

AWS Lambda Function Features

Some of the functions that Amazon Web Services (AWS) offers are as follows:

Auto Scaling and High Availability: When unexpected traffic comes in, AWS Lambda will ensure that your application is highly accessible to end customers. The application can be scaled to achieve high availability.

Server less Execution: AWS does not require manual server provisioning. Based on the triggers you specified, AWS Lambda will provision the underlying infrastructure whenever a new file is uploaded to a certain location. It will then automatically take care of the infrastructure.

Pay-Per-Use Pricing: You will only be billed by AWS for the duration that the compute engine was functioning. AWS charges you according to how long it takes to run the code.

Programming Language Support: The AWS lambda function is compatible with a variety of programming languages. At your convenience, you can use the language to construct the function.

The languages that AWS Lambda supports are as follows:

Node.js

Python

C#

Java

PowerShell

Go

Connects with Additional AWS Services: AWS Lambda can be linked with several AWS services, such as the ones listed below:

Gateway API

The S3

Step Functions of Dynamo DB

SNS (Simple Notification Service)

SQL (structured query language)

Pricing of AWS Lambda Function

The number of requests and the amount of computation time your function requires are the two primary determinants of AWS Lambda price. Each month, you receive one million requests at no cost. You then have to pay \$0.20 for each million more requests. You get billed for compute time according to the amount of memory you devote to the function and the duration of your code execution (in milliseconds). For instance, you pay only if your function runs for 100 milliseconds and consumes 128MB of memory - Just a small single dollar for each run. Additionally, AWS provides you with 400,000 GB of compute time every month at no cost. This implies that Lambda may be entirely free or extremely cheap for minor or infrequent use. For many applications, it's a cost-effective choice because you only pay when your code executes.

II. Methodology

Working with AWS Lambda

Uploading the code to AWS Lambda is the first step. Next, configure the code to be triggered by HTTP endpoints, mobile apps, or other AWS services. AWS Lambda will only use the computational resources required to execute the code when it is activated. Only the computed time is charged to the user.

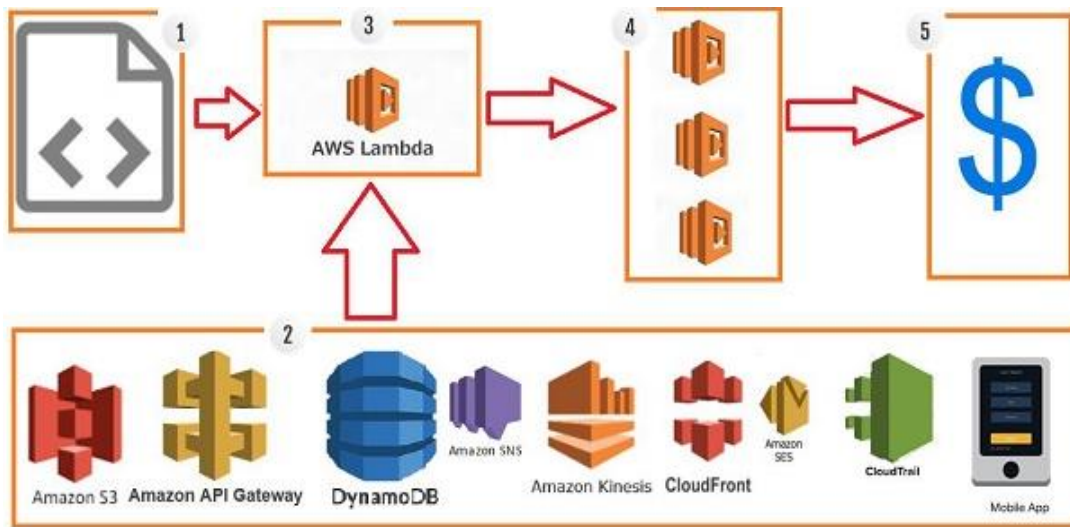
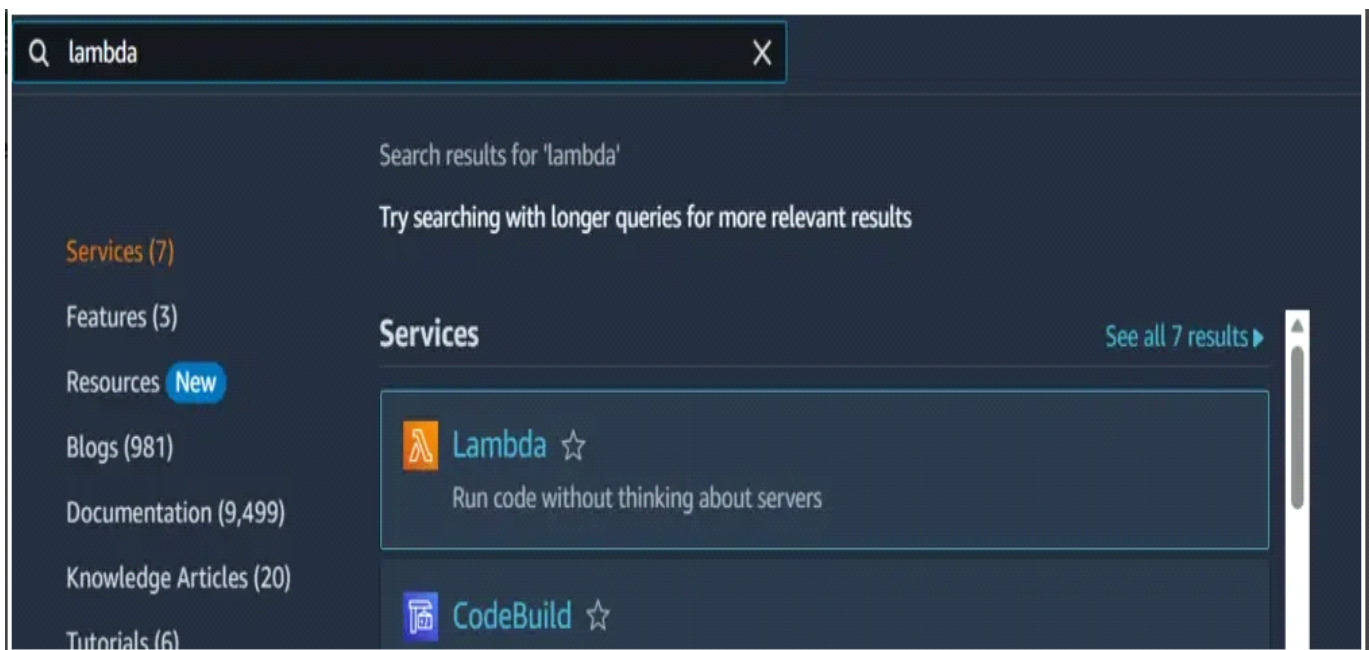


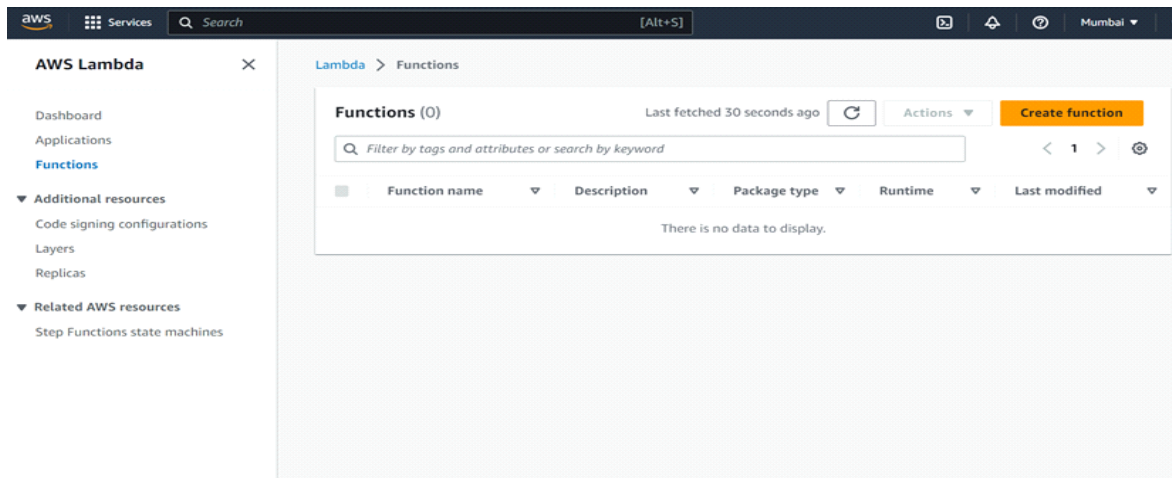
Figure 2 – How Lambda Works

Steps to Create AWS Lambda Functions Using AWS Console

Log in and Open Lambda Console

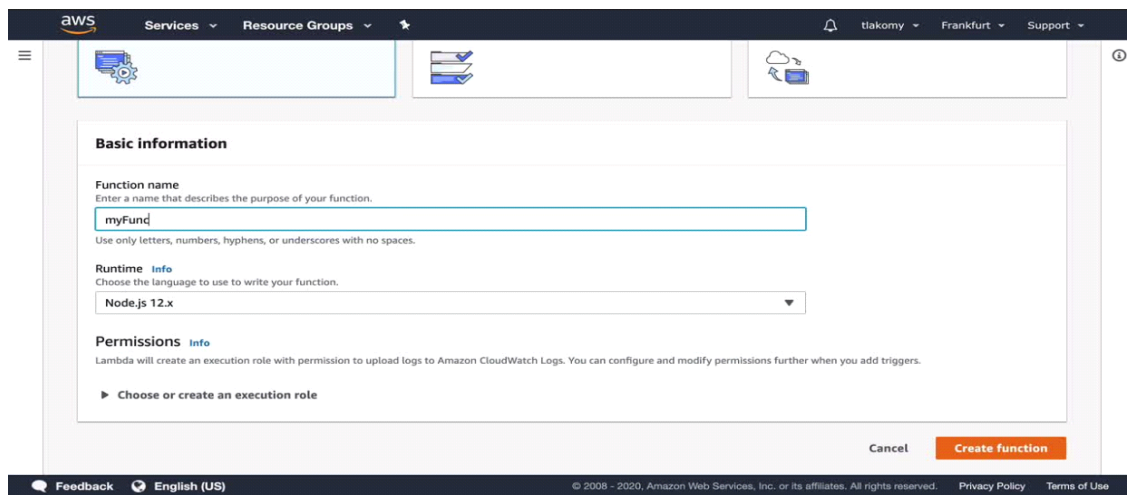


Creating a new Function.

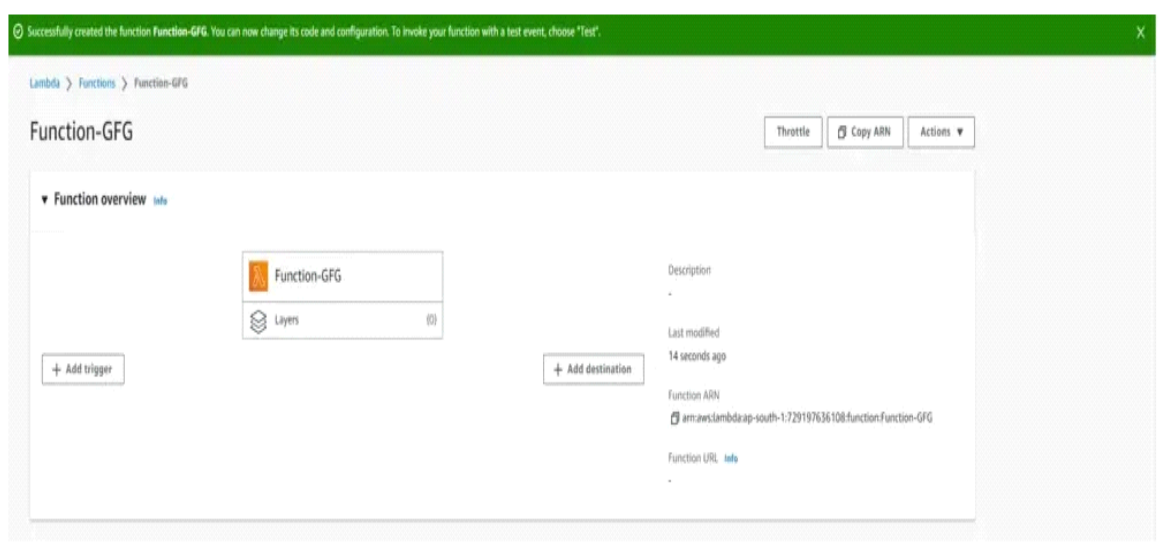


Here, we're going to apply a sample hello world program created from scratch using Author, with the specifics customized to your specifications.

Click on create function (Author from Scratch)



Our function is successfully created.



Based on your goals, the next steps after creating a function in AWS Lambda will vary.

For Example:

1. You can write code directly in the console.
2. Configure the Lambda Function.
3. Add a Trigger (Connect a Source)
4. Test the Lambda.
5. Monitor and debug.
6. Deploy and integrate.

III. Result

Strong results from the study of serverless computing and its application with Amazon Web Services (AWS) Lambda provide a thorough understanding of the revolutionary impact on cloud infrastructures. By carefully examining the features, advantages, and challenges of AWS Lambda, our inquiry has shed light on the shift in perspective from traditional server-based architectures to occasion-driven, auto-scaling, and pay-more-only as costs arise registering models.

Our research into the engineering, execution model, and combination capabilities of AWS Lambda reveals a flexible and adaptable setup that seamlessly adapts to many industry use cases. Real-world applications have shown how well AWS Lambda works to improve operational efficiency, shorten development times, and promote quick innovation across a range of fields.

IV. Conclusion

Server less computing with AWS, using Lambda and Step Functions, makes it easy to build and run applications without worrying about managing servers. AWS Lambda runs your code automatically when needed, and you only pay for the time it runs. Step Functions help you connect multiple Lambda functions and services in a clear, step-by-step way, making your workflows organized and reliable. Together, they let you build powerful, scalable, and cost-effective applications faster, with less effort and more focus on your actual logic instead of infrastructure.

References

1. <https://www.geeksforgeeks.org/devops/introduction-to-aws-lambda/>
2. https://www.tutorialspoint.com/aws_lambda/aws_lambda_introduction.htm
3. <https://docs.aws.amazon.com/>
4. <https://www.ijfans.org/uploads/paper/44c05864cf9a13e973b0ec1e3a610975.pdf>