

Serverless Computing: A Deep Dive

Serverless computing is a **cloud computing execution model** where cloud providers manage the infrastructure dynamically, allowing developers to write and deploy code without worrying about server provisioning, scaling, or maintenance. Despite its name, serverless computing **does not mean there are no servers**; rather, it means that developers do not have to manage the underlying infrastructure manually.

This guide will explore **what serverless computing is, how it works, its advantages and disadvantages, common use cases, and how it compares to other cloud models.**

What is Serverless Computing?

Serverless computing allows developers to run applications without having to manage the underlying hardware or software infrastructure. It is primarily event-driven, meaning that **code execution is triggered by specific events**, such as API requests, database changes, or file uploads.

Instead of running a constantly active server, **serverless applications execute in short bursts** only when triggered. Once the execution is complete, the environment is automatically scaled down, ensuring **cost efficiency**.

In a traditional cloud model, developers must allocate and manage **virtual machines (VMs), containers, or dedicated servers**. In contrast, serverless computing abstracts away this complexity, allowing developers to focus solely on writing application logic.

How Does Serverless Computing Work?

Function as a Service (FaaS)

The most common implementation of serverless computing is **Function as a Service (FaaS)**. In this model:

- Developers **write individual functions** that execute a specific task.
- These functions are **triggered by events** (e.g., an HTTP request, file upload, or a database update).
- The cloud provider automatically **allocates computing resources**, executes the function, and then releases the resources once execution is complete.

For example, **AWS Lambda** is a FaaS platform where a function can be triggered when a new file is uploaded to an **Amazon S3 bucket**.

Backend as a Service (BaaS)

Another approach to serverless computing is **Backend as a Service (BaaS)**. In this model, **third-party cloud services** handle backend functionalities like:

- **Authentication & Identity Management** (e.g., Firebase Authentication)
- **Databases & Storage** (e.g., Firebase Firestore, AWS DynamoDB)

- **Push Notifications** (e.g., Firebase Cloud Messaging)
- **File Storage** (e.g., AWS S3, Google Cloud Storage)

This allows developers to **focus on the frontend** while leveraging pre-built backend services.

Key Characteristics of Serverless Computing

Event-Driven Execution → Functions run only when triggered, making it efficient.

Automatic Scaling → The cloud provider automatically scales resources up or down based on demand.

No Server Management → Developers don't need to worry about maintaining, updating, or securing servers.

Stateless Execution → Functions are independent and do not persist data between executions.

Billing Based on Execution → Instead of paying for pre-allocated resources, users only pay for the actual execution time.