

Serverless Computing

By Garvit Singh

Serverless Computing

Serverless computing is a cloud computing model that allows developers to build and run applications without managing the underlying server infrastructure.

In a serverless architecture, the cloud provider takes care of server provisioning, scaling, maintenance, and server management, allowing developers to focus solely on writing code to handle specific functions or tasks.

Serverless computing is well-suited for various use cases, including web applications, data processing, real-time event processing, and IoT (Internet of Things) applications.

It offers the benefits of cost-effectiveness, scalability, and reduced operational overhead. Developers can write code in their preferred programming languages and focus on building features and applications while leaving infrastructure management to the cloud provider.

1. **AWS Lambda**

- AWS Lambda is Amazon Web Services' serverless computing service.
- It enables you to run code in response to events or triggers without having to provision or manage servers.
- Developers create functions, which are individual units of code that are triggered by various AWS services or custom events.
- Common triggers include API Gateway, S3 bucket changes, database updates, and more.
- AWS Lambda automatically scales the necessary resources to handle incoming requests, and you pay only for the compute

time consumed during the execution of your functions.

2. Azure Functions

- Azure Functions is Microsoft's serverless computing platform within the Azure cloud ecosystem.
- It provides a similar service to AWS Lambda.
- With Azure Functions, you write code in response to events generated by various Azure services, external HTTP requests, or other triggers.
- Like AWS Lambda, it automatically manages infrastructure, scaling, and billing based on usage.

3. Google Cloud Functions

- Google Cloud Functions is Google Cloud Platform's serverless computing offering.
- It allows developers to write event-driven functions that can respond to changes in Google Cloud services, HTTP requests, or custom triggers.

- Google Cloud Functions automatically handles the deployment, scaling, and resource allocation, and you are billed for the actual compute resources used during function execution.

4. Event-Driven Architecture

- Serverless computing is closely associated with event-driven architecture, which is a design pattern that focuses on handling events and triggers to initiate specific actions or functions.
- In an event-driven architecture, events (such as user actions, data changes, or system alerts) trigger the execution of functions in a serverless environment.
- Event-driven systems are highly responsive and can scale automatically to meet demand, making them suitable for applications with variable workloads and real-time processing requirements.

Serverless Architectures

Serverless computing is a cloud computing paradigm that allows developers to build and run applications without managing the underlying server infrastructure. In a serverless architecture, cloud providers automatically manage server provisioning, scaling, and maintenance, allowing developers to focus exclusively on writing code to handle specific functions or tasks. Key features of serverless architectures include:

Function as a Service (FaaS)

- Serverless computing typically revolves around the concept of functions.
- These are individual units of code that are triggered by events or requests.
- When an event occurs, such as an HTTP request or data upload, the corresponding function is executed, and resources are

allocated dynamically for the duration of that function's execution.

Scalability

- Serverless platforms automatically scale resources up or down based on the number of incoming requests or events.
- This means your application can handle high traffic without manual intervention.

Pay-as-You-Go Billing

- With serverless, you are billed only for the actual compute resources used during the execution of functions.
- This is in contrast to traditional server-based models, where you pay for reserved or provisioned resources.

Event-Driven

- Serverless applications are inherently event-driven, responding to triggers and events that initiate the execution of specific functions.

Event-Driven Architectures

Event-driven architectures are a design approach in which software components communicate and interact through events. Events are typically generated by various system components, such as user actions, data changes, or external integrations. These events trigger actions or functions that respond to the event. Key characteristics of event-driven architectures include:

Loose Coupling

- Components in an event-driven architecture are loosely coupled.
- They don't directly call each other's functions but instead emit and consume events.
- This decoupling makes it easier to change, update, or extend components independently.

Asynchronous Communication

- Events are typically processed asynchronously.
- When an event is generated, the system doesn't wait for an immediate response.
- Instead, it continues processing other tasks and allows event handlers to respond when they are ready.

Scalability

- Event-driven architectures naturally support scalability.
- New instances of event handlers can be added as the volume of incoming events increases, allowing the system to handle greater loads.

Flexibility

- Event-driven systems are highly flexible and adaptable.
- They can respond to a wide range of events and can be easily extended or modified to accommodate new event types or

changes in business logic.

Thanks For Reading! ❤️



By GARVIT SINGH

Information Technology Undergraduate