

Hosting: In the context of websites and the internet, hosting refers to the process of storing website files and data on a server that is connected to the internet, so that people can access the website through their web browser. There are Different types of hosting like

Shared Hosting: Shared hosting is a type of web hosting service in which multiple websites are hosted on a single physical server. Each website shares the same resources, such as CPU, RAM, and storage space, as well as the server's IP address.

In shared hosting, the hosting company divides the server resources among multiple customers, allowing them to host their websites at a lower cost than if they were to use a dedicated server. However, because multiple websites are using the same resources, the performance of each website can be affected if one website on the server is consuming too many resources.

Dedicated Hosting: Dedicated hosting is a type of web hosting service in which a single physical server is dedicated to hosting one customer's website or applications. This means that all the resources of the server, including CPU, RAM, storage space, and bandwidth, are exclusively available to that customer only.

Cloud Hosting: Cloud hosting is a type of web hosting service that uses a network of interconnected virtual servers to host websites or applications. In cloud hosting, multiple servers work together to provide a scalable and flexible hosting environment.

The cloud hosting provider typically uses a virtualization technology to create multiple virtual servers, which can be dynamically allocated to customers as needed. This allows for automatic resource allocation and load balancing, ensuring that websites or applications hosted on the cloud are always available and can handle traffic spikes without interruption.

Virtual Servers: A virtual server, also known as a virtual private server (VPS), is a type of server that runs on a virtualized platform within a physical server. Virtual servers share the resources of the physical server with other virtual servers but are isolated from each other as if they were running on separate physical servers.

Each virtual server runs its own operating system and software applications, and has its own resources, including CPU, RAM, storage, and bandwidth. The virtualization software allows for the creation of multiple virtual servers on a single physical server, and allows them to operate independently of each other. Virtual servers are different from normal servers in that they share the resources of a physical server, but operate independently as if they were separate physical servers.

Serverful(Traditional web hosting): Serverful architecture refers to the traditional approach to building and deploying applications, where developers provision and manage physical or virtual servers on which the applications are deployed and run. With serverful architecture, developers are responsible for managing the underlying infrastructure, networking, as well as scaling the infrastructure to meet changing demands.

Examples of serverful architecture include deploying applications on dedicated physical servers, virtual machines, or on-premises data centers. Serverful architecture is still widely used today, particularly for applications that require high performance, low latency, or fine-grained control over resources. [Example: Amazon EC2, Azure Virtual Machines \(While Amazon EC2 does support scalability and elasticity, it is not a serverless computing service. With EC2, customers can rent virtual servers, also known as instances\)](#)

Pros of serverful architecture:

- Full control over infrastructure: Complete control over the underlying infrastructure
- Good for complex applications: Better choice for complex applications that require high performance, low latency, and fine-grained control over resources.
- Cost-effective for long-running workloads

Cons of serverful architecture:

- It requires customers to manage and configure their own virtual servers

- Higher operational overhead: Serverful architecture requires more operational overhead, as you need to manage and maintain the infrastructure yourself.
- Not ideal for unpredictable workloads: Serverful architecture can be difficult to scale up and down for unpredictable workloads, as it requires manual intervention to provision resources.

When to use serverful architecture:

- When you need full control over the infrastructure.
- When you have complex applications that require fine-grained control over resources.
- When you have predictable workloads that require a high level of compute resources.

Serverless: Serverless architecture is a cloud computing model where the cloud provider manages the underlying infrastructure and automatically provisions and de-provisions resources as needed, based on the demand of the application or service. In a serverless architecture, the cloud provider takes care of all the server and infrastructure management, allowing developers to focus on writing and deploying code.

With serverless architecture, developers break down their applications into small, independent functions, also known as serverless functions. Each function performs a specific task or operation, and the cloud provider handles the allocation and scaling of resources to run those functions. Serverless architecture offers several benefits, including reduced operational overhead, automatic scaling, and cost savings (customers are only charged for the actual usage of computing resources).

Examples of serverless computing services are AWS Lambda, Azure Functions, and Google Cloud Functions.

Pros of serverless architecture:

- Low operational overhead: Cloud provider manages the underlying infrastructure
- .Highly scalable: Cloud provider automatically provisions and de-provisions resources on demand.
- Cost-effective for short-lived workloads that don't require a high level of compute resources.

Cons of serverless architecture:

- Less control over infrastructure, a disadvantage for applications requiring fine-grained control.
- Higher latency: Cloud provider needs to provision, warm up resources before running your code.
- Not ideal for long-running workloads: Serverless architecture can be more expensive than serverful architecture for long-running workloads that require a high level of compute resources.

When to use serverless architecture:

- When you want to reduce operational overhead and focus on writing and deploying code.
- When you have unpredictable workloads that require automatic scaling.
- When you have short-lived workloads that don't require a high level of compute resources.

Video

Arpit Bhayani: [Link](#)

[Link](#)

[Link](#)

what would happen if a serverless architecture comes under DDos attack?

Enable throttling on your API gateway, reduce the amount of calls per second to your API Gateway, enable API keys. Clients have to pass the key inside an HTTP header