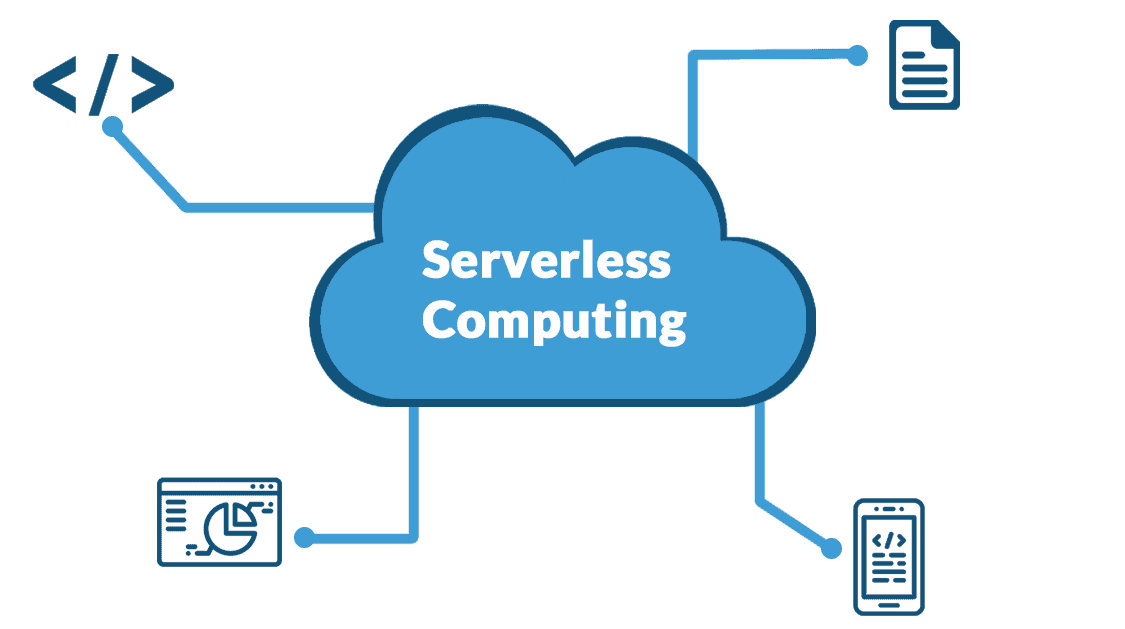
**What is serverless computing?**

* Serverless computing provides inbuilt infrastructures and a runtime environment to develop applications rapidly.
* It is the next-generation evolution of Platform as a Service.
* There is no need to worry about infrastructure, scaling, management, and provisioning at all.
* Serverless computing enables developers to build applications faster by eliminating the need for them to manage infrastructure.
* With serverless applications, the cloud service provider automatically provisions, scales and manages the infrastructure required to run the code.
* The serverless name comes from the fact that the tasks associated with infrastructure provisioning and management are invisible to the developer.
* This approach enables developers to increase their focus on the business logic and deliver more value to the core of the business.
* You can manage real-life applications also and pay for the resources you consume. Whenever the application is in an idle state, you are not charged for it.
* Serverless computing helps teams increase their productivity and bring products to market faster and it allows organizations to better optimize resources and stay focused on innovation.



## What does Serverless Computing do?

Serverless Computing helps users directly focus on the development part of the application rather than worrying about the infrastructure. Developers can innovatively code for the application, whereas serverless computing can take care of the server-related requirements with flexibility. There is no need to worry about any sort of provisioning, purchasing, and managing backend servers. It follows an architecture where vendors provide required backend services to developers.

**Advantages of Serverless Computing**

* You need to **pay for only those resources you use**.
* There is no **need to manage servers** and pay extra costs for their maintenance.
* It provides **dynamic scalability**.
* It provides **quick deployments** and **constant updates**.
* It promises very **low latency** as the code runs in the nearest region, closer to the end-users.

## Best Servers in serverless computing

There are a couple of best trending servers in serverless computing listed down below:

### **Microsoft Azure**:



[Azure](https://k21academy.com/microsoft-azure/microsoft-azure-core-services-for-beginners/) is a Microsoft cloud computing product to**build, test, deploy and manage applications**. There are many data centers available globally to provide the best services with minimum latency.

### **AWS Lambda:**



It is a serverless,**event-driven cloud computing platform** developed by Amazon. It runs codes in response to various events and automatically manages computing resources required by that code.

### **Google Cloud Function for Firebase:**



It is a serverless framework that enables users to automatically run the backend code in response to various events triggered by HTTPS requests and firebase features. Users do not need to manage and scale their own servers as their code is stored in Google’s cloud and run in a managed environment.

### **Oracle Functions:**



It helps developers to**deploy and manage applications** in a managed environment.

## Top benefits of serverless computing

### **No infrastructure management**

Using fully managed services enables developers to avoid administrative tasks and focus on core business logic. With a serverless platform, you simply deploy your code and it runs with high availability.

### **Faster time to market**

Serverless applications reduce the operations dependencies on each development cycle, increasing development teams’ agility to deliver more functionality in less time.

### **Dynamic scalability**

With serverless computing, the infrastructure dynamically scales up and down within seconds to match the demands of any workload.

### **More efficient use of resources**

Shifting to serverless technologies helps organizations reduce TCO and reallocate resources to accelerate the pace of innovation.

## Serverless application patterns

**Serverless functions**

Serverless functions accelerate development by using an event-driven model, with triggers that automatically execute code to respond to events and bindings to seamlessly integrate additional services. A pay-per-execution model with sub-second billing charges only for the time and resources it takes to execute the code.

**Serverless Kubernetes**

Developers bring their own containers to fully managed, Kubernetes-orchestrated clusters that can automatically scale up and down with sudden changes in traffic on spiky workloads.

**Serverless workflows**

Serverless workflows take a low-code/no-code approach to simplify orchestration of combined tasks. Developers can integrate different services (either cloud or on-premises) without coding those interactions, having to maintain glue code or learning new APIs or specifications.

**Serverless application environments**

With a serverless application environment, both the back end and front end are hosted on fully managed services that handle scaling, security and compliance requirements.

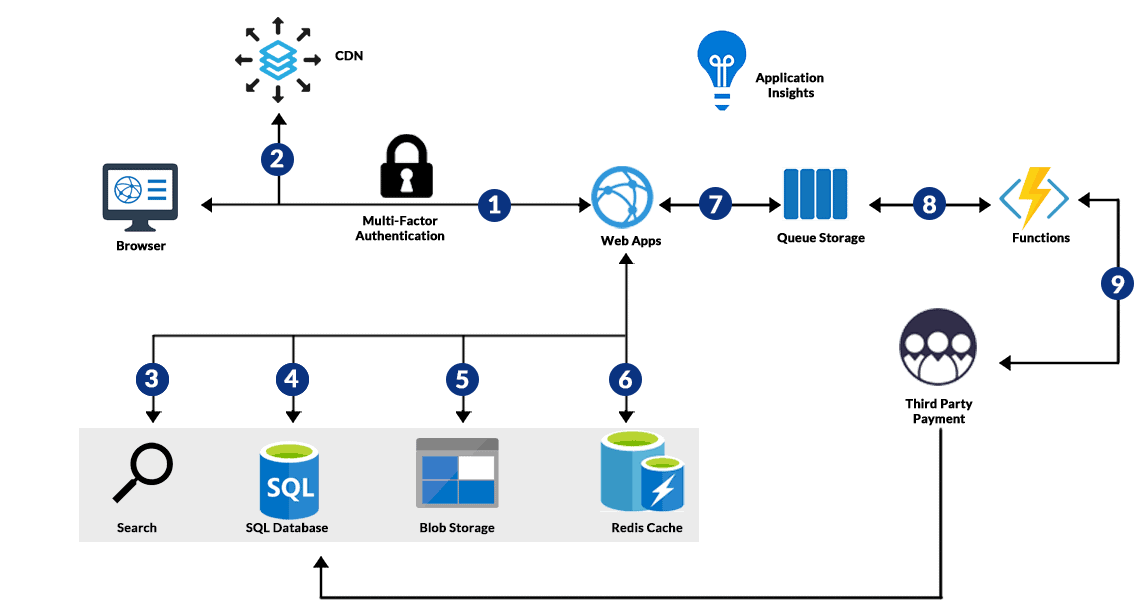
**Serverless API gateway**

A serverless API gateway is a centralised, fully managed entry point for serverless backend services. It enables developers to publish, manage, secure and analyse APIs at global scale.

## Microsoft Azure Serverless Architecture

## Let’s try to understand the Azure Serverless Architecture.

## Users go to their browser to access the application and sign in. Now, the browser starts pulling the static resources like images from Azure CDN (Content Delivery Network). Users make use of SQL queries to search various products from the database. This time, the website pulls the product catalog from the database and product images from blob storage.



To provide a better performance, Azure cache for Redis cached the page output. The Azure function makes payments to the third party and stores the record in the SQL database. Users submit their orders and order is placed in a queue. Now it’s the turn of the Azure function to process the payment.