

WHAT IS SERVERLESS COMPUTING? SERVERLESS COMPUTING EXPLAINED



If you are looking for new ways to deploy applications faster and more frequently, then it's time to consider serverless computing.

Serverless computing allows developers to build apps without the headache of managing infrastructure. More specially, it allows them to write in *serverless code* without having to:

- Provision a server
- Ensure its functionality
- Create test environments on a server
- Maintain server uptime

This frees up resources to focus their attention on accelerating innovation in today's competitive digital economy.

The Importance of the Cloud

Serverless computing is a cloud-based service which means that technically speaking servers *are* involved. However, what makes serverless computing different from architectures we traditionally see in enterprise environments is not a lack of server, but rather *how* those servers are implemented and managed. With serverless computing, the service provider takes care of all the infrastructure

(server-side IT), which means all your team needs to do is write code.

Remember that traditional servers have fixed resources that need to be provisioned for use. On the other hand, serverless computing does not have fixed resources and as such, can run multiple applications from a single server.

It is also important to note that serverless computing is stateless. States are created as I/O requests are received and then destroyed in compute instances. The process is 100% automated and does not require the kind of human interaction and maintenance than a traditional server does.

This makes serverless computing an efficient, cost and resource effective way to build and use applications.

Serverless Architecture

Beyond the definition of serverless computing itself, explained in the paragraphs above, there are other considerations and terms you need to understand about serverless architecture to truly grasp serverless computing.

Software Containers or FaaS

Software containers function as a virtual repository for serverless code. Containers make it easy for developers to write serverless code for more than one platform, like an iOS app and a desktop. Each application has its own container with functionality specific to the platform.

The idea of using individual containers on a third-party platform to power application development is referred to as Function-as-a-Service or *FaaS*.

FaaS allows developers to run serverless code without storing data.

BaaS

Backend-as-a-Service or *BaaS* and FaaS are similar in that they both require a dependency on a third-party service provider. When we think of BaaS, the most well-known example is Amazon's AWS Lambda service.

With Lambda, developers are still using serverless code in software containers, but Amazon works as an intermediary between the user and the implementation of code in a software container. Amazon provides guidelines for developers to follow when submitting code. Amazon then enters the code into a container, via automated processes, providing the entire backend development process as a service to their clients.

Other Types of Serverless Architecture

The foundation for serverless computing is its architecture. It is powered by code designed to work with FaaS or BaaS being offered by third-party vendors. As far as charges are concerned, these platforms offer a *pay-as-you-go* structure. Based on runtimes, they charge consumers only for what they use-not for pre-provisioned sets of units.

In addition to the above mentioned popular FaaS and BaaS platforms, there are a few other serverless architectures of note. These include:

- **Serverless Databases** - Serverless databases are intended to store data, unlike FaaS and BaaS.
- **Serverless Frameworks** - These are testing frameworks for developers to code in.

Drawbacks of Making the Switch to Serverless Computing

There are a few potential drawbacks of serverless computing to bear in mind:

Response Latency

Serverless computing is an unparalleled framework when it comes to rapid development of applications. But, how well do they perform?

One drawback of switching to serverless computing may be response latency. Response latency is the term that defines the time between when a request is stimulated and when a program reacts. In the cloud, serverless computing is not continually running -- it gets powered down between requests. As a result, having to start from dormant serverless code can cause response latency to occur.

Limitations

Third-party vendors restrict resource limits, which means serverless computing is not ideal for high-computing operations. Even if they were no such limits, certain applications would not be cost-effective from a third-party vendor.

Security

When it comes to serverless computing here are some security considerations:

- Responsibility of security falls on the servicing company, not the consumer
- There is a larger attack plain with multiple entry points into the server

Customers sometimes feel reassured by the security demands being shifted to the company the owns the server, but large servers with multiple entry points may be more vulnerable. And if an attack does happen, consumers are powerless against it.

Debugging

It is generally more challenging to debug serverless code. In addition, developers also have a harder time spotting problems as the code doesn't lend itself to digging in for a detailed review of issues or smells.

Understanding, anticipating and preparing for these challenges will be critical in meeting the needs of the end user and helping your organization gain a competitive edge with serverless computing.

Serverless Computing and Enterprise DevOps

Including [DevOps](#) departments in enterprise organizations makes a bold statement about the status of a company's [digital transformation](#) and capabilities. DevOps is an innovative and elite group that serves as the bridge between development and operations. DevOps teams are well-versed in Agile

principles and a variety of coding languages. DevOps developers can build software, but spend much of their time churning out patches, instead, to customize and integrate a number of enterprise applications.

That said, it is only fitting that serverless computing is entered into the DevOps mix, so that more enterprise companies can enjoy its benefits.

Benefits of Serverless Computing

There are several key benefits of serverless computing. Those who make the switch to building applications on serverless platforms experience the following major enhancements to their business models:

Cost

Switching to serverless computing can have a positive impact on the overall IT budget. Without the hardware of a server, maintenance costs are relatively low. Labor cost is possibly also reduced since round-the-clock monitoring and maintenance of servers is not required. In addition, since serverless computing services charge based on runtimes, you are never paying for more of a service than you use.

Operations

Since the third-party vendor ensures that server performance meets enterprise demands, operations are simplified. Companies benefit from allocating resources to other projects and eliminating the need for scalability planning and other strategies that come along with traditional computing models.

Productivity

Due to the nature of cloud computing, the requests that are available for end-users are limited to ones that are relatively simple to code. This gives developers more time to work on other projects that help enhance a company's business model or service portfolio.

Efficiency

With fewer people, processes and technologies required to deploy applications, serverless computing offers an efficient way to launch fully scalable applications.

Serverless computing offers:

- Zero server management
- Scaling options
- Optimal availability
- Eliminates idle capacity

Focus

With no hardware maintenance requirements, software development teams are free to focus on

coding leading to better applications. Serverless computing allows for rapid building of applications so developers can make the most of their skill sets.

An Era of Serverless Computing

Every industry from finance to education can benefit from serverless computing. This means your clients will be asking for serverless solutions sooner rather than later.

More specifically, for those looking to build event-based apps quickly and efficiently, serverless computing is the way to go in order to conserve resources, increase efficiency and boost productivity.

BMC offers unparalleled [application deployment and development support](#) to enterprise organizations. Let us help your DevOps teams deliver applications quicker and with higher quality by providing self-service development capabilities that work within existing continuous delivery processes.

- [Control-M Automation API](#) - Simplify scheduling and enables developers to avoid rework, reduce pressure, and accelerate the speed of application delivery by up to 30%.