ACL Digital
an ALTEN group company

Serverless Architecture

Development and adoption of serverless solutions help business organizations of all sizes to soar past the digital transformation journey

WHITEPAPER

Table of Contents

Serverless architecture (a.k.a. Serverless Computing) is a software design principle that enables developers to build, deploy, and run the application as a service without the dependency on managing the infrastructure. In most cases, the applications are modeled into individual functions that can be deployed and scaled separately.

The cloud has ensured that companies worry less about their IT infrastructure and instead focus on their core offerings. In the present scenario, most business organizations are moving to the cloud, freeing themselves of infrastructure problems. With serverless architecture, enterprises can quickly re-imagine their offerings, from ideation to production. From maximizing resources to agility to innovate, there are numerous benefits that serverless computing offers.

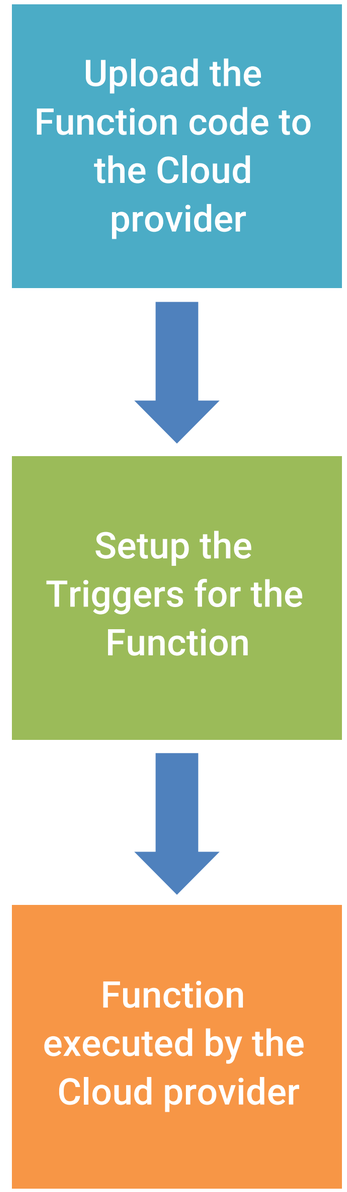
This whitepaper exemplifies serverless architecture's usage to transform how we design multi-tier architectures and execute patterns, services, and other applications. Easy adoption and development of serverless solutions enable enterprises of all sizes to advance in the digital transformation journey by eliminating obstacles such as infrastructure, operations, development, and deployment costs.

Industry Solutions and Use Cases

In the conventional method, the user communicates with servers, and servers, in turn, communicate with the applications and process the request and provide a response to the user. In this approach, the corresponding team must manage the servers, which requires considerable time and resources. The team must control the server software, underlying hardware, security patches, backup, restore, etc. Still, in the case of serverless architecture, the team can offload the work to manage the servers to the third-party provider and concentrate only on writing the application code.

Serverless architecture is also known as Function as a Service (FaaS). In this approach, developers will develop the application as a set of functions where each Function will perform a unique functionality, and the events will trigger these functions. These trigger events can be either a mail or an HTTP request. Generally, once the developers develop these functions, they will deploy their functions, which trigger the service provided by their cloud provider. Whenever any Function is requested, the cloud provider process the corresponding Function by either running on the existing server or the cloud provider will start the new server and executing the Function. In this approach, the developers do not need to know how this Function will be implemented in the server and can concentrate only on writing the application code.

The FaaS offerings from the cloud provider include AWS Lambda, Azure Functions, Google Functions, IBM Cloud Functions, Alibaba Functions, Oracle Functions, etc.



Serverless vs Server Architecture

In a server-based approach, the applications will be hosted in the servers, and users could be able to communicate to the deployed application through the servers. The team should manage the server hardware, server software, security updates, backup, recovery, etc.

In the case of the Serverless architecture approach, the team can offload all the server managing work to the third-party provider and concentrate only on development activities.

In the server-based approach, the complete application will be hosted in one or more servers, allowing it to scale the application as a whole. Still, in the case of a serverless approach, it is possible to scale every individual functional module based on the needs of the business.

Serverless (Faas) vs Platform as a service (Paas)

Platform as a Service (PaaS) is a deployment approach similar to the serverless architecture. Still, the whole application will be deployed as a single unit, so there will be no scope to scale individual functionality inside the application. The popular products offered as PaaS are Azure Web Apps, AWS Elastic Beanstalk, Heroku, etc.

Serverless architecture is the Function as a Service (FaaS) model in which applications will be developed as the individual, autonomous functions, deployed to the FaaS service offered by the third-party cloud provider, and scaled automatically based on the usage. FaaS is the most cost-effective way for the needed computing resource.

Serverless vs Container Architecture

Like the serverless approach, container architecture allows the developer to deploy their application without knowing or managing the underlying infrastructures, but there are some differences between these architectures. In the case of container architecture, the developer must update and maintain the container they deploy and its system settings and dependencies. Still, in the case of serverless architecture, the maintenance of the server is completely taken care of by the cloud provider who provides the service to host the functions developed by the developers.

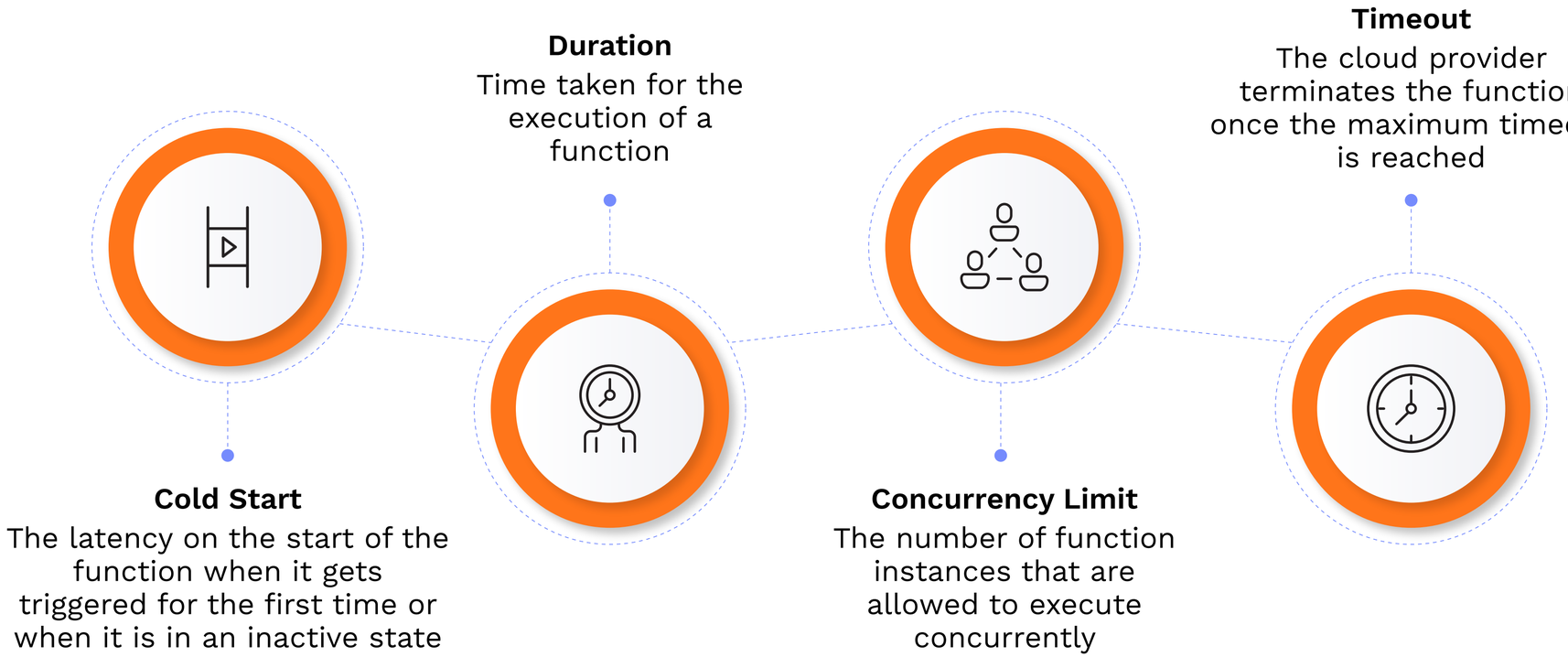
Also, in the case of serverless application, it will scale automatically, whereas container orchestrator like Kubernetes, Docker Swarm, etc., is needed for scaling the container applications.

Container architecture is suitable for high traffic applications, whereas serverless architecture is more suitable for trigger-based applications.

Concepts of Serverless Architecture

Although serverless architecture allows the developer to deploy the application without knowing the underlying infrastructure, there will still be a learning curve needed to implement the function that takes the complex workflow. The following are some of the parameters of the serverless architecture.

Invocation – Triggering or execution of a function



Advantages and Disadvantages

In recent times, a lot of enterprises started adopting the serverless architecture because of the various advantages such as:

Scalability

Function instances will be automatically scaled up and scale down based on the traffic within the concurrency limit

Productivity

The developer need not have to know or manage the underlying infrastructure and concentrate entirely on the development activity to rapidly implement the new things

Cost

Cloud provider will change only based on the per-invocation basis, so we need not be paying for any unused virtual machines

Still, there are some pain points while adopting the serverless architecture. They are:

Performance

When the function is inactive for a period, the process will automatically shut down. In this case, invoking the function creates the latency as it should get loaded to the server again by the cloud provider

Security

Cloud providers will share the same server with the other customers, & if the server configuration is not done correctly, then there will be a possibility our application data will be exposed to other cloud provider's customer

Provider Lock-in

Mostly, we are locked to the cloud provider as we don't have much liberty to mix and match services from other providers while using the serverless architecture

Testing

The developer should be able to perform only the unit testing on the functional code, but it is not possible to perform integration testing, which helps to test the integration of the frontend with the backend

Loss Of Control

The developer will not have control over the technical stack. In case of any failure, the developer must wait for the cloud provider to fix the issue

Tools and Frameworks

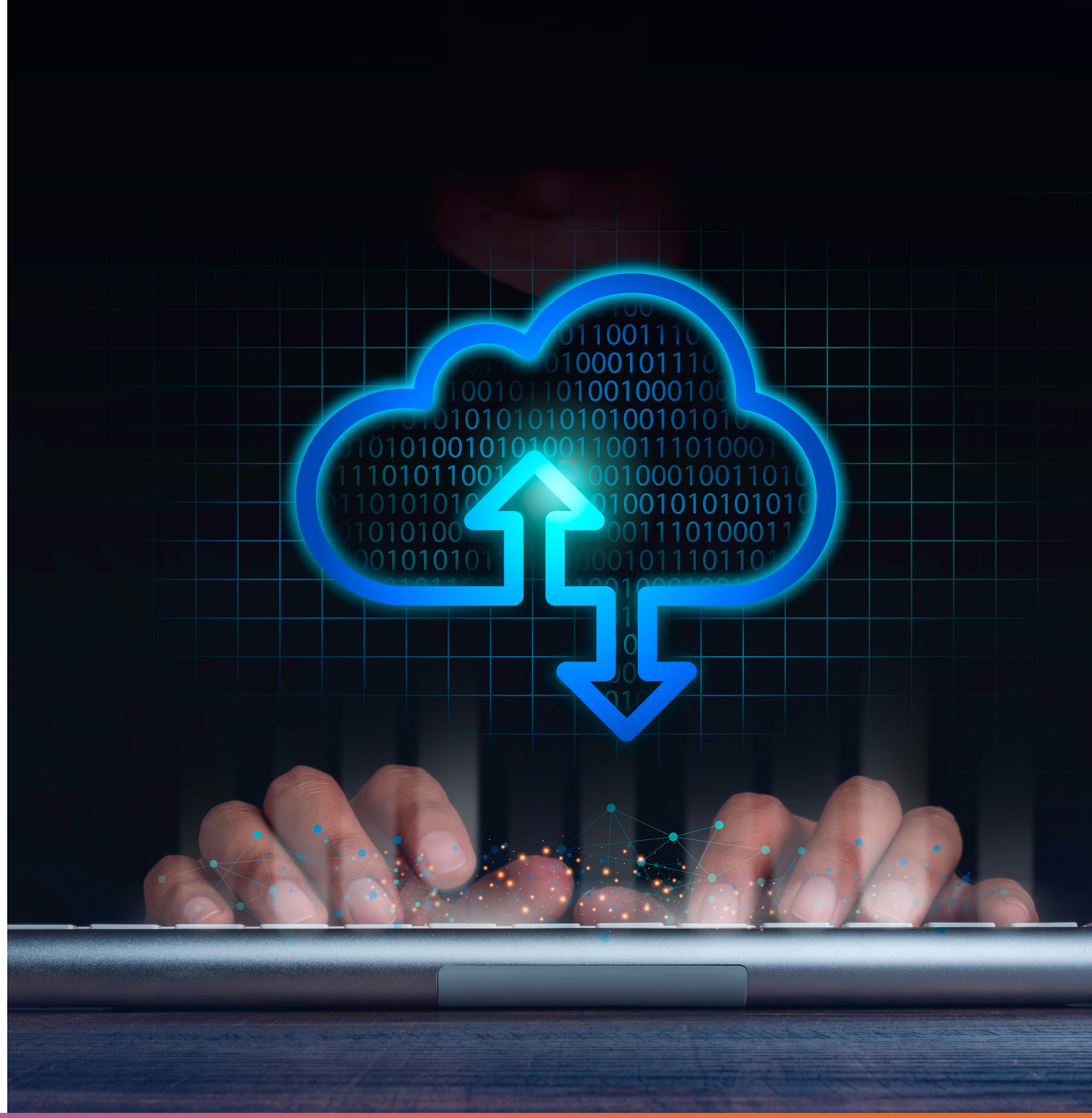
Tools are inevitable for developing and maintaining the functions, which helps the organization transform their application into serverless architecture.

The tools like Amazon’s Serverless Application Model (SAM) interact with the AWS platform through API and allow developers to define their functions, triggers, and permissions. AWS provides tools to test the functions locally before deploying to the cloud. Also, security tools scan our function for vulnerabilities and block unauthorized access and code injections.

Who Should Use Serverless?

Serverless architecture is more suitable for minor applications with a smaller number of functions. It may not be ideal for large applications but is still suitable for more complex ones.

Typically, serverless is more suitable for developing the rules-based use cases, scheduling tasks, sending emails, etc.



Conclusion

In recent times, serverless architecture is emerging as the leading architecture pattern as it allows the enterprise to make rapid development on the functionality.

In ACL Digital, we have successfully implemented many serverless functions on the AWS and Azure Cloud. We have the experience to suggest the right solution around the serverless architecture.

ACL Digital
an ALTEN group company

Digital
Product
Engineering

ACL Digital, an ALTEN Group Company, is a digital product innovation and engineering leader. We help our clients design and build innovative products (AI, Cloud, and Mobile ready), content and commerce-driven platforms, and connected, converged digital experiences for the modern world through a design-led Digital Transformation framework. By integrating our strategic design, engineering, and industry capabilities, we help our clients decode the digital world and accelerate their growth journey.

EMAIL US
business@acldigital.com

TALK TO US
+1 (408) 755 3000

www.acldigital.com