SERVERLESS VS PAAS: IS SERVERLESS THE NEW PAAS?

In the race to develop the fastest, most efficient software at the highest frequency, more organizations are looking to solutions like Platform-as-a-Service (PaaS). PaaS is a natural choice because it gives businesses more functionality with less management responsibility. This allows developers to focus on the act of developing without maintaining things like runtime environment.

However, even with their hands in a wealth of services offered through AWS and indeed pioneering IaaS, Amazon seems hesitant to enter the PaaS market. But could it be that Amazon's vision extends past PaaS?

Indeed, some experts believe that serverless architecture, like that found in AWS Lambda, will soon become the "new PaaS". And if that's the case, Amazon has been refining its offering since 2014.

Snapshot: Serverless vs PaaS

If you've been looking for a full-service framework solution for your enterprise needs, then it's likely that you've come across the concepts of serverless and PaaS before.

On the surface, these two solutions may seem very similar. But they have some unique differences that must be taken into consideration.

Serverless characteristics include the following:

- Cost provisioned

- Micromanaged
- Full of auto scaling capabilities
- Most efficient on the market
- Less dependent on code
- Operational with fewer glitches

PaaS characteristics include the following:

- A service that offers greater developer control
- Easy to use
- An offering with limited management responsibilities
- Scalable
- Efficient

What is PaaS?

In simple terms, Platform-as-a-Service refers to cloud platform services. But if you're reading this, you probably already knew that. You may also be aware PaaS is a framework for developers to create fully-customized applications with limited management responsibilities in the cloud. This is important for enterprise businesses seeking agility as they implement a DevOps approach to software development. The bottom line is that Paas requires fewer management responsibilities, allowing for greater focus to be placed on development.

PaaS differs from SaaS in that PaaS is equipped to handle both development and delivery of software over the Internet, while SaaS only allows for software delivery.

The advantages of PaaS for enterprise businesses include:

- Simple, cost-effective development and deployment
- Readily scalable
- Highly available
- Lots of features for a small amount of management and monitoring resources
- More automation, less need for coding
- Easy migration

For more information on PaaS and whether or not it's right for your enterprise, take a look at this article.

What is Serverless?

Serverless architecture is the implementation of serverless code to develop in the cloud. Like PaaS, serverless architecture represents a framework.

Frameworks commonly developed using serverless code and consumed by enterprise business users include:

- FaaS: Function or framework-as-a-Service: In the space of pre-packaged services, FaaS, falls somewhere in between Software-as-a-Service and Platform-as-a-Service. Think of FaaS as a ready-to-implement framework that can be easily tailored to the needs of an enterprise company.

- BaaS: Backend-as-a-service: Some will argue that BaaS takes things a step further as a so-called NoOps offering. NoOps refers to infrastructure that has been automated to the point that in-house developers have no part in its operation.
- Database: Database serverless frameworks access and automate your database functions. These are functions that both write and read from a database, as well as provide a response. Serverless database frameworks offer companies room to expand their global footprint, as multiple applications can be developed by region. However, they all run from a single location (powered by FaaS technology).

The serverless architecture that most readily competes with PaaS in its native form, is FaaS. FaaS is similar to PaaS in that it offers development features not found in simple SaaS offerings. But it does so while requiring fewer resources than PaaS.

Organizations who moved to PaaS because of its reduction of management resources are able to further reduce costs using serverless instead. However, the feedback from some is that serverless takes management control completely out of the developer's hands.

Still, FaaS applications scale better than PaaS applications and have a number of benefits that help enterprises reduce costs. These include:

- Reduced need for coding capabilities, fewer glitches and bugs
- Reduced cloud vendor lock-in
- Highly available
- More features for fewer management resources

There are some obvious similarities between serverless and PaaS. These include reliability, availability, scalability, providing a framework for development and reducing management.

What's the Difference Between PaaS and Serverless?

With all of that said, what's the difference between the two?

The major differences lie within the scope of the major advantages of serverless technology over PaaS. Serverless takes all of the PaaS advantages a step further with fewer drawbacks. For example:

- The serverless cost structure is event driven so you don't pay fixed monthly fees for services that go unused, this makes it highly efficient and reduces overall cost
- Serverless is micromanaged so internal administrative resources can be used for other activities
- Serverless offers true auto-scaling capabilities to customers

At the end of the day, serverless is more effective at reducing cost and speeding up time to deployment and software releases.

Implications for Enterprise Customers

The adoption of serverless as the new PaaS has implications for the world of development in a number of ways:

- Serverless apps are a far departure from a time where middleware was required to build applications, as vendors adopting serverless offerings will need to consider changing their

models of development
- Piggybacking off the last point, by offering serverless in a variety of services across AWS, Amazon is reducing the ability of as-a-service vendors to compete with them in this domain
- Developers will experience problematic learning curves as the tides shift from PaaS to serverless
- With PaaS, creating single environment applications was sometimes worth vendor lock-in, but with serverless code being useable in an open-source environment, container applications using serverless can now break the cycle of vendor lock.

These are all considerations for an academic discussion on the evolution of PaaS. However, considering the advantages of serverless platforms, it does seem like Amazon may have been onto something when they skipped PaaS and went straight to severless with AWS Lambda.

Amazon and the Rise of Serverless Solutions

That said, many industry experts questioned Amazon's rationale beyond the limited features of its automatic deployment tool known as Elastic Beanstalk. Indeed, Amazon's reluctance to acknowledge PaaS by creating a robust native product, could be that the company has already adopted serverless in the AWS platform, and therefore sees no use for PaaS.

While some businesses are hesitant to jump head first into serverless due to the micromanaging nature of the product, others have already adopted serverless features of AWS and use them daily in their enterprise.

Still, some are interested in the customization and control that comes with a uniquely PaaS product and it remains to be seen if Amazon will sway those potential customers. Or if they are even trying to.

Serverless Hosting Providers

The following providers are the big names offering serverless hosting environments:

- Lambda (AWS) - This is the most well-known, quintessential serverless framework on the market offered by Amazon.
- Azure Functions - Allows you to build apps faster with the Microsoft serverless platform.
- Cloud Functions - Google Cloud Functions offers serverless computing on Google's open infrastructure. The platform supports Javascript and executes in Node.js, offering familiar environments for developers to code.
- OpenWhisk - IBM Cloud Functions is based on Apache OpenWhisk. According to their website, "OpenWhisk manages the infrastructure, servers and scaling using Docker containers so you can focus on building amazing and efficient applications."
- Fission.io - This solution offers a framework for serverless functions on Kubernetes.

As evidenced above, most of the tech giants today have serverless product offerings.

PaaS Vendors

Here are some PaaS vendors to keep an eye on if you are considering this product line:

- AWS Elastic Beanstalk - Amazon

- Azure - Microsoft
- App Engine - Google
- BlueMix - IBM
- BitNami

By maintaining this small offering, Amazon still has a toe in the water so to speak when it comes to PaaS, although the organization is not competing anywhere close to the scale of its severless investments.

Final Thoughts

So is serverless really the new PaaS? It's hard to say.

As outlined above, serverless comes with a number of benefits not least of all the ability to customize and scale, providing a framework-as-a-service. But what you don't get when you use a serverless host for your application architecture is control.

The very nature of the hosting environment removes much of the control from developers. Developers who are looking for something more than SaaS, but also appreciate a greater degree of control over changes continue to choose PaaS as their solution.

But as more and more competitors follow trailblazers like Amazon, it remains to be seen if PaaS has long term viability.