TechTarget | Search Microservices

E-guide

PRO+ Content

IaaS vs. PaaS: An Enterprise Developer's Guide

In this e-guide

In this e-guide:

For organizations looking to leverage and roll out microservices, DevOps, or digital transformation, a major decision stands in their way: "To IaaS or to PaaS?"

Read on to uncover the benefits, challenges, and differentiating characteristics of each to help you make an informed decision for your architecture or methodology.

In this e-guide

Deciding between IaaS and PaaS for microservices

Darryl Taft, News writer | SearchMicroservices.com

As more organizations move to become digital enterprises, among the first moves they make is to step away from large monolithic applications to smaller, more flexible microservices-based applications.

This approach to digital transformation is driven by the need for greater agility and scalability. With software becoming a differentiating factor in just about every industry, every company must provide or maintain its own software factory of sorts to be able to update its software offerings as quickly and as often as possible to keep up with the competition.

The microservices architecture is emerging as a key method of providing development teams with flexibility and other benefits, such as the ability to deliver applications at warp speed using infrastructure as a service (IaaS) and platform as a service (PaaS) environments. With microservices, developers can break monoliths into a bunch of smaller, independent services, each with its own business logic.

The microservices can be scaled and deployed separately and written in different programming languages. But a key decision many organizations face when deploying their microservices architecture is choosing between IaaS and PaaS environments.

In this e-guide

Microservices explained

A microservice is a tightly scoped, strongly encapsulated, loosely coupled, independently deployable and independently scalable application component, according to Gartner. Indeed, microservices are based on the concept of service-oriented architecture (SOA), Gartner analyst Anne Thomas said. In fact, some people refer to microservices as "SOA done right," she said in a report.

And unlike monolithic architectures, microservices divide complex applications into smaller parts consisting of services that communicate with each other through a message broker, database and REST API, Forrester Research analyst Randy Heffner said.

Yet microservices architecture is still in the "innovators" stage of adoption, according to Gartner. And of the companies using it, most are born-on-the-internet, cloud-native entities such as Netflix, Google, Twitter, Uber and Airbnb. In a 2016 Gartner study, the research company noted that early adopters of microservices have been able to cut development lead times by as much as 75%.

Meanwhile, monolithic architectures typically require long-term commitments to a particular technology stack and can require significant resources to scale. Microservices help users avoid those issues as each service is deployed separately and has its own business logic. That reduces dependencies between teams and makes it easier to add new features on the fly.

In this e-guide

"Adopting a microservices architecture is a key component to cloud-native application development, which is, above all, about speeding up iteration in innovation and value delivery," said Rhett Dillingham, senior analyst for cloud services at Moor Insights & Strategy. "To be most effective in cloud-native development, using microservices requires adopting features supporting a continuous delivery and DevOps approach. Use of Docker containers to ease implementation of DevOps within a container orchestration tool like Kubernetes, which eases continuous delivery, is common practice. Most cloud platforms offer features [in software] or services [in IaaS] automating the building and running of microservices using these tools."

How microservices technology works on PaaS vs. IaaS

What each organization stands to gain from moving to a microservices architecture depends on their cloud infrastructure choice, said Sergey Sverchkov, a project manager and development lead at Altoros, a Silicon Valley system integrator focused on the cloud. Sverchkov said running an app on bare IaaS is more affordable, "but you will need a DevOps team to maintain it." However, using a PaaS is more expensive, but it can shrink release cycles from weeks to hours, he noted.

The differences between IaaS and PaaS are distinct. Platform as a service basically includes APIs for operational tasks such as provisioning a container, monitoring resources for auto-scale triggers, provisioning a data

In this e-guide

model and connecting containers to the network. Infrastructure as a service is the layer below.

"IaaS just gives you network, storage and compute," IDC analyst Larry Carvalho said. "It's not going to get you what you need. You need some services on top of that."

The distinction brings to mind the old app server concept from the SOA world, where developers could get higher-level APIs to system services such as transaction management, object-relational mapping or message queuing. Only now, we are talking about cloud-based services. PaaS also includes provisioning, deployment and orchestration.

In a blog post, Gnani Dathathreya, director and distinguished engineer at Capital One, wrote that with a monolithic app server, "A single faulty service can bring down the entire app server and all the services running on it. Microservice differs in that each service runs in its own container."

Carvalho said every PaaS provider "is going toward a cloud-native approach, which is containers, container orchestration with Kubernetes, container security and all the features that you used to use in the traditional PaaS but now moving to a microservices approach. So microservices and containers and cloud functions are all in there together."

Gartner agreed. "By 2018, most high-control application (hca) PaaS providers will include microservices infrastructure capabilities in their [app PaaS]," Thomas said in her report.

In this e-guide

Features to look for

How do you decide between IaaS and PaaS? Despite a more recent shift toward PaaS for microservices architectures, there are still plenty of companies that go the IaaS route, using platforms such as Amazon Web Services (AWS) or Microsoft Azure. Netflix was an early pioneer in migrating to AWS for microservices, building a slew of open source software along the way to make up for some of the things missing in the IaaS layer.

Capital One's Dathathreya noted that his organization also had to add components. "Part of creating our microservices architecture involved developing infrastructure automation, container management solutions and a DevOps pipeline for our microservices," he said.

Moreover, Dathathreya said the company's DevOps pipeline components for microservices include source code management, a build server, code repository, image repository, cluster manager, container scheduler, dynamic service discovery, software load balancer and a cloud load balancer.

All of those are among the features needed for deploying microservices on IaaS environments. Other features required include the following:

- A mature Agile and DevOps practice for either IaaS or PaaS platforms. As most organizations are using microservices to support continuous delivery, a solid DevOps team is mandatory.

In this e-guide

- Continuous delivery support. In an IaaS environment, DevOps engineers must independently install and configure build automation tools.
- Developer independence. PaaS platforms provide developer independence, whereas with IaaS, DevOps engineers may need to create multiple IaaS environments for each development group.
- Application runtime support. An application runtime is automatically deployed in a PaaS container. But with IaaS, the DevOps team needs to configure an application runtime on IaaS instances.

The bottom line

When weighing IaaS and PaaS, Gartner recommended that organizations approach microservices with caution since "most organizations don't have the type of web-scale agility and scalability requirements" to adopt a microservices architecture. However, if you do embark on a microservices journey, you should host your microservices in a modern PaaS that supports containers and provides container management services.

Next article

In this e-guide

What's the difference between IaaS and PaaS to developers?

Fred Churchville, Site Editor | SearchMicroservices.com

IaaS vs. PaaS: The differences

IaaS can be referred to as the *build-it-yourself* approach. By working with an IaaS offering, like Amazon Web Services (AWS) or Google Compute Engine, developers and application management personnel will have access to an integrated set of tools and other resources that will help with development and app management.

"IaaS is sort of low-level; I've got all these resources available to me -- compute, storage, network -- and I assemble them together myself," explained Randy Shoup, VP of engineering at the online subscription and personal shopping service Stitch Fix, a company that has invested heavily in microservices development. "So, instead of getting the car, get the pieces of the car, and I build it myself."

IaaS provides the tools and resources needed for development, but the organization is responsible for provisioning and managing its own development platform. This means developers will need to orchestrate their own execution processes, manage their own data services and maintain the underlying OS. For some teams, this will mean more flexibility when adding

In this e-guide

unique features and tweaking apps and services for maximum performance. But this may also be a more time-consuming approach because more work will be involved for developers.

PaaS, on the other hand, provides the complete application development package for developers: tooling, support and a preconfigured development platform. PaaS can be likened to the full service gas station of application development management services. Examples of PaaS options include Red Hat's OpenShift platform, Heroku, Microsoft Azure and Google App Engine. Engine Yard, OpenStack and CA Technologies are other examples of vendors in this space. PaaS customers work on a platform and with services that are created and maintained by the PaaS provider, rather than on their own.

"PaaS is more like getting a car and just driving," Shoup explained. "So, all the pieces are already assembled for you. They work together, and it makes it easier."

This means that application execution services, data services and underlying OS functions are handled through the vendor's software. The PaaS vendor will also provide a suite of development tools, frameworks and other resources that are uniquely integrated with the platform the vendor provides. PaaS providers may be able to customize the platform for an organization's specific needs, but it may take time or require that the organization wait for a new version of the platform software.

In this e-guide

"The only difference is really whether or not the development tools are actually built into the environment or if you're going to host them separately," explained Chris Tozzi, a TechTarget contributor and DevOps analyst.

More together than separate

While it is possible for both of these application development approaches to function independently of the other, it is often the case that organizations will run both IaaS-based and PaaS-based operations alongside the other or in conjunction.

"That's kind of where we are at Stitch Fix right now," Shoup explained. "We are in a bit of a transition period from [where] we were fully on a platform as a service, using Heroku, and now, we are halfway between that and building our own platform on lower-level infrastructure as a service, things from Amazon Web Services."

Shoup explained that an organization might employ a strategy where user-facing aspects of an application are developed using a PaaS, while back-end functions are handled using an IaaS approach.

"You can absolutely imagine, for example, that all the user-facing pieces, all the user-facing applications, use an off-the-shelf platform as a service, like Heroku, App Engine or Cloud Foundry," Shoup said. "And then, maybe the back-end services aren't served well by that thing, so we write those more in infrastructure. That's a totally legitimate strategy, and all the modern platform microservice vendors enable that."

In this e-guide

Shoup explained that this crossover between IaaS and PaaS occurs even within a vendor's own offerings. Since the PaaS offering Heroku is built on AWS, theoretically, users will have access to those same AWS development tools in whatever other development projects they need it for, whether it is on or off the Heroku platform.

"From within Heroku, I mean, you have relatively convenient access to all the AWS services," he explained. "So, you can kind of have a foot in both worlds."

According to some experts, the lines may be blurring between what is considered IaaS and PaaS. New container orchestration offerings from large vendors, like Amazon EC2 Container Service, could be considered to "lie in the middle," between IaaS and a fully integrated development and deployment platform.

"The vendors would like you to believe that it's one or the other," explained George Lawton, an independent researcher and TechTarget contributor. "But really it's a spectrum, and they're sort of converging."

Next article

In this e-guide

IaaS vs. PaaS: The best choice for a DevOps environment

Jan Stafford, Features Writer | SearchMicroservices.com

PaaS or IaaS? That is the question

Along with many other advantages of microservices-based applications, small code base, scalability, simple deployment and more, come additional options in technology stacks. That gives DevOps teams the option to choose more flexible cloud platforms for development and deployment that fit business needs as well as DevOps environment, team size and skill levels. But selecting the right cloud services platform -- usually between IaaS and PaaS -- on which to deploy a microservices architecture and develop microservices is not so easy.

In this buyer's handbook, veteran IT journalist Darryl Taft explores the rationale for choosing infrastructure as a service and platform as a service. Essentially, it's a know-thyself exercise, Taft explains, because either choice comes with its own requirements for in-house expertise, customization and lifecycle management.

With IaaS, the organization gets the big package of infrastructure services, including hosted network, server, storage and development tools. The pros and cons of IaaS for microservices development and deployment are the

In this e-guide

same: The organization has to use IaaS and sometimes third-party tools to build its own DevOps platform. The big questions here are: Does the DevOps team have the skills and, more important, the time to build? And would the broader customization options of build your own be valuable?

That's not to say customization isn't possible with most PaaS offerings, but the key value of PaaS is its packaged and integrated one-stop toolkit. The need for speed is a top driver for choosing PaaS, which comes with automated development, deployment, provisioning, security and other application lifecycle management tools. While a build-your-own IaaS development platform requires maintenance by the in-house DevOps team, the PaaS user gets updates and maintenance services from the provider.

When deciding between IaaS and PaaS, we offer expert advice on which option best suits your DevOps environment.

Next article

In this e-guide

About SearchMicroservices

Over 4 million programmers, architects, IT managers, and developers turn to our site for industry news, expert advice and peer-to-peer learning opportunities around managing microservices and service-oriented architecture (SOA), application modernization and digital transformation, Business Process Management (BPM), application integration and APIs, software containers and microservices design and development, DevOps, and more.

Whether you're looking to solve a specific application architecture problem or just trying to stay on top of recent industry developments, our site is your online portal for in-depth and relevant information

For further reading, visit us at http://SearchMicroservices.com

Images; Fotalia