Platform as a service

How platform as a service (PaaS) empowers modern agile and DevOps teams

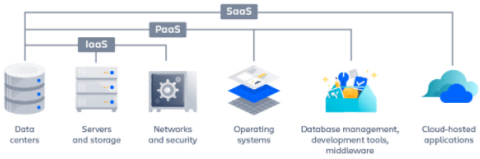
BY KEV ZETTLER

Summary: *Platform as a service (PaaS) is a cloud infrastructure layer that provides resources to build user-level tools and applications. It includes the underlying infrastructure including compute, network, and storage resources, as well as development tools, database management systems, and middleware.*

Today's software development teams now have immediate access to all the resources needed to support the entire application lifecycle, including to design, develop, test, deploy, and host applications. This can all be done directly from the cloud with platform as a service (PaaS). Teams have access to the resources they need, on-demand, without the need to purchase and maintain a complex on-premise infrastructure.

PaaS is a primary tier of modern cloud infrastructures. The base stack is infrastructure as a service (IaaS), which provides compute, network, and storage resources. PaaS is at the middle of the stack between IaaS and software as a service (SaaS). PaaS is dependent on IaaS but also enables SaaS.

What is platform as a service?



PaaS is a cloud infrastructure layer that provides resources to build user-level tools and applications. Like IaaS, these resources include servers, storage, and networking. PaaS also includes development tools, database management systems, middleware, email or notification systems, and more.

Since PaaS is a cloud-based infrastructure, it allows organizations to avoid the cost and complexity of buying and managing infrastructure resources including software licenses, application infrastructure, and development tools.

The "platform" in PaaS refers to a programming language ecosystem, or "tech stack." Popular language-based application tech stacks include Ruby on Rails, Django Python, Node.js MEAN stack, Java SpringMVC, among others. These language stack examples depend on an attached database system.

Benefits of platform as a service

Like other cloud services, PaaS provides resources on-demand that can scale as needed. Prior to this, teams needed to purchase their own hardware, then configure the servers, databases, firewalls, and all other components themselves. Teams would then have to maintain, scale, and monitor their custom infrastructure. There are also a host of other benefits to PaaS.

Empowers agile and DevOps practices

PaaS enables faster coding, testing, and deployment -- some of the key practices of agile and DevOps teams. Plus, PaaS is directly complementary to a CI/CD release workflow and helps to enable a full DevOps release cycle. Given the software lifecycle of building, testing, deploying, managing, and updating, PaaS handles the deployment phase.

Faster time to market

Faster time to market

Using platform as a service, developers can focus purely on code, rather than building, configuring, and provisioning infrastructure and platforms, or building auto-scaling functionality. This significantly shortens development time on new projects. Also, PaaS offers access to tools, templates, and code libraries that can reduce development time and simplify processes.

Scale as needed

PaaS hosts generally offer elastic scaling features, which lets teams rapidly add capacity in peak times and scale down as needed. Scaling is handled automatically by the PaaS provider.

Cost-effective development

PaaS allows teams to add development capabilities without adding staff, which can reduce engineering costs. Organizations no longer need to install and manage underlying development infrastructures.

Platform as a service providers offer helpful dashboards to analyze and manage infrastructure cost. These cost analysis tools help teams audit any areas of unexpected or wasteful expenses. Furthermore, these tools help teams optimize the cost of their deployments. Without these cost insights, teams may find their infrastructure expenses unexpectedly growing.

Supports distributed development teams

Since platform as a service is a cloud service, it supports collaboration between distributed teams. Platform as a service providers have globally distributed hardware, which means applications deployed to PasS can be accessed at lower costs and from any location.

Security and access control

Most PaaS providers have granular security and access control tools that enable teams to quickly configure access to PaaS resources. This is important for both company and customer security team access levels. This gives teams assurance that their infrastructure is locked down to prevent data breaches or other undesirable security failure scenarios.

Platform as a service use cases

PaaS is used to deploy user-ready application code executables including full SaaS web applications like CRMs, dashboards, chat rooms, and more. The PaaS execuatables also include backend APIs or microservices. In addition to application code, supplementary tools like load balancers, notification pipelines, and delayed job systems can be deployed alongside application code.

API development and management

APIs are an essential component of any modern distributed application, and the built-in frameworks provided by a PaaS greatly simplify API development and management. APIs are commonly used to enable external systems to connect with internal application resources, or to connect the different components in a microservice application architecture.

Microservices

PaaS is complementary to microservices and a great aid in deployment. PaaS makes it easy to deploy multiple microservice applications and configure them to communicate with each other. Most PaaS providers have user interface dashboards that provide a visual description of the current deployment. This allows teams to better see and grasp the layout of their live microservice deployment.

Multi-phase environments

PaaS can quickly deploy multi-phase environments like development, staging, and production. This gives teams added quality assurance, since they can verify correct application behavior through multiple phases. If the application behaves as expected in staging, it is then trivial to deploy it to the production environment.

Database hosting

Most applications depend on some kind of persistent data store. Because this is so common, PaaS providers offer database deployment and management as a core functionality. Teams can bypass the PaaS application hosting and instead use PaaS purely for database access. This pattern is so common that some PaaS providers offer specific billing plans for database only functionality.

Business analytics/intelligence

Most PaaS offerings make developer's lives significantly easier by including applications and frameworks for performing business analytics and intelligence. These solutions make it much easier for application developers to provide users with all the data and metrics they need to make informed business decisions.

Communications

Communications capabilities are essential to any modern application, and a PaaS provides communication tools and frameworks to support technologies such as sms, email, voice, etc. that make it simple for developers to add communication functionality to their applications.

Internal tools and private dashboards

Internal tools are perfect candidates for PaaS deployment since they don't require special one-off infrastructure dependencies. Internal tools aggregate views of internal metrics and don't need advanced, elastic scaling because they're accessed by a subset of internal stakeholders. Even so, PaaS provides elastic scaling by default.

In conclusion...

PaaS gives infrastructure and development resources to small and large development teams alike. It allows teams to access the resources they need, when they need it, and to scale accordingly. PaaS enables faster execution and more frequent software releases, making it a favorite technology for teams practicing agile and DevOps.

Ready to integrate PaaS with your DevOps workflow? Get started with DevOps

KEV ZETTLER

Kev is a lead full stack web developer and serial entrepreneur with over a decade of experience building products and teams with agile methodologies. He is a passionate contributor, author, and educator on emerging open source technologies like DevOps, cryptocurrency, and VR/AR. In his free time, he participates in indie game development jams.

ARTICLE

How infrastructure as a service empowers the modern enterprise

Learn about infrastructure as a service, including the pros and cons and how to use it.

Read this article →

ARTICLE

How infrastructure as code (IaC) manages complex infrastructures

In order to reap the benefits of agile, you need to be agile through all phases of the software development lifecycle.

Read this article →

Up Next
Continuous Delivery Pipeline 101 →