The Buyer's Guide to Platform-as-a-Service (PaaS)

2019 edition

TrustRadius

Table of Contents

1 Why Read This Guide?

PaaS is designed to accelerate application development and reduce operations overhead. But this huge advantage comes at a cost: the significant benefits of a PaaS are possible because of certain constraints. The challenge of PaaS technology is that there is some necessary loss of flexibility and PaaS users lose a degree of control over compute characteristics and workload-processing patterns. Development and DevOps teams must be comfortable with the particular set of constraints imposed by the PaaS and feel that the cost is reasonable.

This guide seeks to provide PaaS buyers with a free resource to help them make better informed purchasing decisions and let them know what PaaS users think of 7 popular solutions. Our goal is to help you differentiate between these 7 PaaS products, and find the one that will serve your business needs best.

84% of PaaS users said that their PaaS enabled them to bring new applications to market faster.

This guide is based on detailed information from 2,726 reviews and ratings published on TrustRadius, as well as 112 additional survey responses from end-users about the buying process. Each review and rating has been individually vetted by a TrustRadius researcher to ensure we're providing authentic feedback from real users of the software. Along with detailed reviewer feedback and survey responses, this guide is informed by interviews with domain experts and vendors.

In this guide readers can find:

- An overview of the category, including the differences between PaaS, IaaS, CaaS and FaaS.
- PaaS buying trends and what PaaS users are struggling with the most.
- At-a-glance summaries of 7 PaaS products highlighting reviewer demographics, pros and cons, and end-user feedback.

196,000 reviews from 100% verified users

About TrustRadius

TrustRadius is the most-trusted review site for business technology, bringing transparency to the $3.7 trillion B2B market. We help buyers make confident decisions with in-depth reviews and ratings from real software users. Every reviewer on TrustRadius is authenticated and every review vetted by our Research Team before publication. We also help vendors engage and convert buyers by putting their customer's voice to work, authentically and at scale.

Headquartered in Austin, TX, TrustRadius was founded by successful entrepreneurs and is backed by the Mayfield Fund and LiveOak Venture Partners.

To learn more, visit www.trustradius.com.

2

Factors to Consider When Choosing a PaaS

Not all PaaS are the same. When considering a purchase, buyers should understand some of the tradeoffs to be made. Key factors include the degree of abstraction provided by the platform, the deployment model, and the various different types of PaaS, particularly the differences between PaaS closely tied to a specific SaaS product or operating environment versus standalone platforms.

Cloud computing platforms provide access to a shared pool of configurable computing resources.

The first factor: understanding PaaS as a part of cloud computing

PaaS is one of the three best-known cloud computing models. Cloud computing defines the delivery of computing services over the Internet. To be more precise, cloud computing platforms provide access to a shared pool of configurable computing resources such as networks, servers, storage, applications and other services that can be rapidly provisioned over the Internet.

Most technology buyers are familiar with the most well-known cloud computing model, Software-as-a-Service (SaaS). The other two are Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS).

How are these different cloud computing models different?

	Example Products	Who Uses It?
SaaS	Gmail, Trello, Salesforce, CRM, EventPro, Office365, GoogleDocs	End Users
PaaS	Heroku, AWS Elastic Beanstalk, Google App Engine	Software Developers
IaaS	AWS Elastic Compute Cloud (EC2), Microsoft Azure, Google Compute Engine	IT Administrators

SaaS

Applications running in the cloud are offered to application consumers or end users as a service over the Internet. The SaaS provider handles all the infrastructure and upgrades Consumers of the software service typically access it via a web browser and pay a monthly or annual subscription. This pricing model provides lower upfront costs to users and a reliable income stream to vendors.

PaaS

PaaS is a development and deployment environment in the cloud that makes coding and deploying applications much more efficient for developers. It is essentially a platform for building scalable software delivered over the web with modest up-front investment.

SaaS, PaaS, and IaaS are the three main cloud computing models.

PaaS is designed to release application developers from the complexity of provisioning, configuring, managing, and updating hardware resources by providing the operational components underlying application development. These include operating systems, databases, container orchestration, middleware, and BI a as cloud-enabled services. Developers of business applications don't have to worry about scalability and security issues, and are free to focus on developing applications.

IaaS

IaaS, unlike its close cousin PaaS, is not aimed at application developers, but rather at IT professionals. An IaaS is a service providing the most fundamental layer of cloud services including compute, storage, and networking services. IaaS scales infrastructure components up and down with demand and avoids the complexity and expense of having to manage physical servers and other datacenter infrastructure. IaaS platforms allow users to install and configure operating systems, middleware and applications.

Higher levels of abstraction allow developers to focus on application dev. rather than maintenance and capacity planning.

Do you need IaaS, CaaS, PaaS, or FaaS?

The fundamental differences between these related concepts are best understood in terms of increasing abstraction. Abstraction is a term borrowed from object-oriented programming and, in this context, has to do with how different platforms handle complexity and control. This is a helpful way of thinking about the technology stack from the most fundamental bottom layer, with increasing abstraction and agility, as you move up the stack to the top layer. The price for this increasing agility is a loss of control and configurability.

Developers generally prefer to take advantage of higher levels of abstraction. This allows them to focus on application development and avoid distracting (but necessary) overhead work, such as server maintenance and capacity planning.

This continuum of increasing abstraction means that PaaS buyers may want to include emerging adjacent technologies like Container-as-a-Service (CaaS) and Functions-as-a-Service (FaaS) as part of their consideration set. The following diagram* illustrates how PaaS relates to other adjacent "as-a-Service" platforms, and sheds some light on the increasing popularity of emerging FaaS technology. Moving from left to right, from low abstraction to high abstraction, agility increases and configurability decreases.

	Infastructure-as-a-Service (IaaS)	Container-as-a-Service (CaaS)	Platform-as-a-Service (PaaS)	Function-as-a-Service (FaaS)
Example	Digital Ocean	Google Kubernetes	Red Hat OpenShift	AWS Lambda
Description	Virtual Infastructure	Containerized Apps	Focus on App Logic	Event-Driven Architecture
	Configurability			**Agility**
Base Layer Unit	Virtual Machine	Container	Application	Event-Driven Functions
Good For	Existing Systems	Running in Multiple Environments	Web-Facing Application Development	Extending Services

**This is a modified version of Google computing stack illustration*

IaaS

IaaS (Infrastructure-as-a-Service) platforms provide IT professionals with a robust computing infrastructure. But IT staff must select, configure and monitor the platform components that they want to use. There is considerable overhead to getting started with an IaaS, but these systems are highly configurable, providing administrators with granular control over servers, storage, and networking services.

Examples of IaaS platforms:

> Google Compute Engine

> Digital Ocean

> Amazon Web Services

> Rackspace Managed Hosting

An IaaS environment is an alternative to running a private data center.

When to choose an IaaS:

An IaaS environment is an alternative to running a private data center. Modern businesses increasingly do not want to invest in and manage servers, storage, and networking for private data centers for a number of reasons, notably cost. Building a private data center is extremely expensive. However, renting the infrastructure from a major cloud service provider reduces logistical and financial barriers to entering new markets without being constrained by IT resources. If your organization's IT resources are provided by an on-premise data center, switching to an IaaS replaces high costs for equipment and personnel with an easy-to-manage and much more affordable cloud services model.

A CaaS is a good option for development teams who want more control over containerization tasks.

CaaS

CaaS (Container-as-a-Service) is one step up the abstraction ladder. Where IaaS platforms use virtual machines or bare metal hardware as fundamental resources, CaaS platforms package up applications and all their dependencies into containers more lightweight than virtual machines. For this reason, it's possible to host more containers on a single host than fully-fledged virtual machines. CaaS is a suitable platform for developers who want more control over container orchestration. Using a CaaS allows developers to deploy applications on containers without worrying about the limitations on container orchestration provided by the typical PaaS.

Examples of container orchestration tools:

> AWS EC2 Container Service

> Google Kubernetes

> Docker Swarm

> Rackspace Karina

When to choose a CaaS:

Both CaaS and PaaS help developers deploy and run applications more efficiently. A CaaS is a good option for development teams when they prefer to have more granular control over containerization tasks. Your team may need greater control than a PaaS can provide for things like language support or multi-cloud hosting. In cases like this, a strong infrastructure team used to dealing with networking and operations issues may prefer a CaaS over a PaaS. A CaaS is typically less opinionated and more configurable than a PaaS.

PaaS

PaaS (Platform-as-a-Service) is still further up the abstraction ladder and therefore favors agility over configurability. As we have seen, PaaS is an integrated application development and deployment solution that releases application developers from much of the complexity of provisioning, configuring, and managing hardware resources. Developers benefit from far more automation at the expense of less granular control.

A PaaS will allow an in-house development team to develop a new application much faster than otherwise possible.

Examples of PaaS platforms:

> Salesforce Heroku

> Google App Engine

> AWS Elastic Beanstalk

> Rackspace OpenShift

When to choose a PaaS:

A PaaS is a good choice when many developers are working on a complex single application. It provides for the simultaneous use of a single source code and the ability to automate testing and deployment. Acquiring a PaaS will allow an in-house development team to develop a new application much faster than otherwise possible.

FaaS

FaaS (Function-as-a-Service) is a category of cloud computing services that is disrupting the way applications and systems have been built for decades. FaaS is a relatively new development that has emerged out of PaaS. It is frequently thought of part of PaaS, although it is a distinct technology which is often referred to as Serverless. The term Serverless gives a clue about the level of abstraction. Like PaaS, FaaS is a cloud computing model that abstracts server management and low-level infrastructure decisions away from developers, but it goes much further than PaaS. All allocation of resources is managed by the platform, allowing applications to be developed without any thought of implementation, load balancing, or scaling. It allows developers to execute small snippets of code in response to events without having to build complex infrastructure.

FaaS can be very cost-effective since developers are only charged for the time it takes to execute the code.

The advantages of a FaaS are primarily that developers can focus on writing code rather than thinking about infrastructure management. Another major advantage is that developers are only charged for the time it takes to execute the code. If the code is only executed a few times a day, the total charge per day might be for just a few seconds.

Examples of FaaS platforms:

> AWS Lambda

> Google Cloud Functions

> Microsoft Azure Functions

> IBM Cloud Functions

When to choose a FaaS:

FaaS is a good choice when developers want to release and iterate new applications very quickly and easily without having to worry about infrastructure issues like scaling. For tight budgets, FaaS is also attractive because users are only charged when the code executes. FaaS is a particularly good option for processes like ETL or time-based batch processing where expensive tools can be replaced by using function “runs” to only execute and pay for each run instance.

Consider PaaS deployment model options

The PaaS computing model is usually thought of as being deployed in a public cloud, i.e. computing services provided over the Internet by third-party providers. For example, customers subscribing to NetSuite or Salesforce usually access these applications over the public cloud. But this is not the only model.

Most PaaS solutions support public, private, hybrid, & community cloud deployment options.

Public cloud

A public cloud is the provision of computing services over the public Internet. Public cloud services are usually sold on-demand and help companies avoid the high cost of purchasing and maintaining an on-premise data center.

> Shifts CapEx to OpEx and supports multiple tenants
> Offers a Pay-as-You-Go Model

Private cloud

A private cloud is provided to a single organization rather than the general public. It offers the same advantages of scalability, resource utilization, and self-service as the public cloud, but is provided in a data center operated either by the organization itself or a third party.

> Leverages existing CapEx and can help reduce OpEx
> Intended for a Single Tenant

Hybrid Cloud

Hybrid deployments are a combination of on-premise, private cloud, and third-party public cloud services. Resources are orchestrated between the two clouds by management and automation software.

> Bridges one or more private, public, or community clouds
> Supports resource portability
> Allows maniupulation or CapEx and OpEx to reduce costs

Community Cloud

The community cloud is a recent variation of public cloud that provides a cloud platform for specific business communities. It is particularly useful for highly-regulated industries like health care, financial, or legal entities with a strong need to share resources.

> Allows sharing of CapEX and OpEx to reduce costs
> Supports resource portability
> Brings together groups or organizations with a common goal/interest

Implications for Buyers

Most PaaS software can run on premises, hosted in a private environment or natively in the major IaaS public cloud. The PaaS automatically configures infrastructure resources across these environments, making them a platform for hybrid cloud. However, not all products will be able to handle all three environments with equal ease. Ask vendors to demonstrate their platforms operating in each scenario.

Community Cloud PaaS are only relevant to buyers operating in specific communities. For example, a healthcare community cloud, tailored to provide specific security and regulatory requirements compliant with HIPAA data protection regulations, is of interest only to healthcare organizations.

3 Different Types of PaaS

There are three main types of PaaS: those tied to a particular SaaS vendor, a particular operating system, or open-cloud.

PaaS tied to a SaaS product

These PaaS offerings are tightly connected to commonly-used SaaS platforms like Salesforce, Workday, or Intuit. The purpose of this kind of PaaS is to create a developer ecosystem around a SaaS application by providing a platform allowing ISVs to create new capabilities that run on the core SaaS platform. Examples might be custom business processes, platform extensibility, data model customization, or a broad range of new functionality. Many of these platforms have low-code/no-code capabilities opening them up to less technical resources.

In this way, new capabilities can be rapidly created to meet market demand and then sold into the customer base directly or via an application marketplace.

Examples of this kind of PaaS are:

> Workday Cloud Platform

> Salesforce App Cloud/Lightning

> Google App Engine

> Intuit Developer Network

Most of these PaaS tools are provided to customers or ISVs free of cost by the vendor, although fees may be charged up to a certain level of resources consumed, such as storage, bandwidth, or hours used.

PaaS tied to an operating environment

The most common example of this is an Infrastructure-as-a-Service (IaaS) vendor including PaaS capabilities as part of the IaaS offering and encroaching further up the stack. These offerings may not have the same depth of functionality as standalone PaaS platforms, but they can work well if the customer is committed to running only on one specific IaaS. Some buyers may worry about vendor lock-in.

Examples of this type of PaaS are:

> AWS Elastic Beanstalk

> AT&T PaaS

> Microsoft Azure

Open-Cloud PaaS

This kind of PaaS is not tied to a SaaS product or operating environment but allows organizations to use a completely separate platform. This provides flexibility, but can also add cost.

Examples of open-cloud PaaS are:

> Engine Yard

> Cloud Foundry (open source)

> Jelastic

> Red Hat OpenShift

> Apprenda

aPaaS: the future of PaaS?

This guide focuses on the PaaS category, but in recent years a variant has appeared which frequently causes confusion: aPaaS. This stands for Application Platform-as-a-Service and is best understood as a further development of PaaS focused on development. While PaaS is a middleware platform that provides a number of services to developers including integrations, business intelligence, business process management, application development, operating systems, and a long list of other platform services, aPaaS is more specifically focused on application development alone.

As PaaS vendors add drag-and-drop capabilities to their platforms, we will begin to see convergence of PaaS with aPaaS (Low-Code)

An aPaaS exclusively provides services required to build and deploy apps. It does not provide any other PaaS capabilities. The primary benefit of an aPaaS is development velocity. An aPaaS can help organizations move from concept to finished product more quickly than if they were to develop on a PaaS. These tools are designed to help organizations become extremely agile at the expense of control. As developers become more difficult and more expensive to hire, the focus has shifted to tools that allow non-developers to create their own apps.

Examples include low-code and no-code development platforms like Mendix and OutSystems. These are a type of aPaaS to which Gartner has given the unlikely name HPaPaaS, or High Productivity Application Platform-as-a-Service.

The other kind of aPaaS is High Control Application-as-a-Service (HCaPaaS). These tools are suitable for professional developers building customer-facing, front-office, or back-office cloud-native software solutions. The category covers code-friendly third-generation programing language tools that are suitable when professional developers are available, and complex services are needed that are not based on standard models.

The aPaaS concept has overtaken standard middleware PaaS solutions in terms of industry excitement, and has also led to confusion between the two categories. The promise of low-code and no-code platforms is potentially ground-breaking: ease-of-use by non-technical staff, time-to-market from idea to implementation, and flexibility to integrate various business and IT functions mean PaaS capabilities will be accessible to more people than ever before. However, PaaS is still a remarkably useful technology for highly skilled development teams developing complex applications in a much more agile manner than was previously possible.

Note: This PaaS guide does not include low-code/no-code aPaaS products, but is focused on the more fully featured PaaS solutions.

3 Lessons from PaaS users

A survey of PaaS users on TrustRadius reveals high levels of customer satisfaction with these platforms across the board. 93% are happy with platform feature sets, and 90% of respondents indicate that they would buy the same platform again. Responses reveal a few useful tips, though, from those who have bought PaaS before.

The benefits of PaaS

Development velocity is one of the principal benefits of using a PaaS. Since developers can concentrate on coding, applications can be delivered to market much faster than would be possible otherwise. PaaS vendors are largely successful in delivering on this key benefit. 84% of respondents say their PaaS has enabled them to bring new applications to market faster.

84% of PaaS users said that their PaaS enabled them to bring new applications to market faster.

Here are some characteristic quotes from users on development velocity:

Allowed us to develop and deploy our app with only one engineer. [Our PaaS] has saved time to develop and do other things for our app.

Makes it a lot easier to get an incredibly developed application off the ground and into the hands of the relevant parties.

Speed increased greatly and it allowed for simultaneous creating and integration at a later point, rather than consecutive deployments.

The main benefit—time to market decreased.

Worry free from managing Infrastructure; more resources freed up to work on platform and application level projects.

63% of PaaS users also use an IaaS.

IaaS vs PaaS

Many companies use both an IaaS and a PaaS in conjunction. In fact, PaaS functionality is increasingly being bundled with IaaS offerings. However, this is not a requirement. When asked whether PaaS users also use an IaaS in conjunction, 37% of PaaS users report that they do not also use an IaaS.

Although some IaaS platforms include limited PaaS capabilities, many companies opt to run a PaaS platform independently of IaaS. If the most important problem your business is facing is to help application developers get up and running as quickly as possible without having to worry about things like server management, a PaaS is a great choice. It can be a particularly smart choice for early stages of development when companies are anxious to build a minimum viable product and are less concerned with infrastructure issues like scaling and redundancy.

Of the 63% of respondents who do use a PaaS in conjunction with an IaaS, the most common choice of IaaS is AWS. The most common order of operations is to adopt the IaaS first, and then the PaaS.

Choosing an IaaS and PaaS from different Vendors

Most respondents are running IaaS and PaaS from the same vendor. But the smaller number who have opted for heterogeneous platforms, in some cases, have experienced issues:

Yes, mainly due to security configurations between the providers preventing the required access. Issues also rise when the IaaS provider makes unannounced changes, temporarily affecting connectivity.

We used to run Heroku and AWS platform at the same time and we've had problems with too many CI/CD pipelines to monitor and maintain, having similar service on different platforms is not helping much.

We do have a mix of PaaS and IaaS. We do see compatibility and integration issues that we are currently resolving. For IaaS we use Microsoft Azure and AWS. For PaaS we use AWS Elastic and Google App engine.

Yes, massive issues with using Heroku with different platforms. You basically must purchase Heroku Enterprise to securely pair it with other platforms. The enterprise offering is very expensive and doesn't offer many compelling options besides security configurations that should be native in the non-enterprise product.

Yes, there are cross compatibility issues meaning systems deployed in one IaaS cannot access internal only systems from a PaaS from another vendor.

Tip for Buyers

Although issues do not always arise in running a PaaS independently of an IaaS, our survey data contains enough anecdotal commentary around running PaaS and IaaS from different vendors to indicate that this can be troublesome. If you already have an IaaS in place and are looking at acquiring a PaaS (or vice-versa), consider the vendor's own PaaS solution. If you do prefer to implement two different vendors' products, talk to the vendors about compatibility issues. Ideally, get the PaaS vendor to demo the platform working smoothly on top of an underlying IaaS from a different vendor to be certain that there are no significant compatibility issues.

If implementing IaaS and PaaS solutions from different vendors, talk to vendors about compatibility issues.

35% of PaaS users have hired or are hiring new teams to manage these cloud platforms.

Will you need new organizational roles?

35% of respondents are either actively considering creating new roles or have already hired new teams to manage these cloud platforms. One third of these expressed the need to hire or train DevOps engineers. Some organizations have created entirely new development teams to take advantage of the technology. One respondent indicated that engineers working on legacy technologies were replaced by "CloudOps" teams specialized in setting up IaaS and PaaS systems.

Training is also key, as existing resources may require re-training focused on cloud computing paradigms. 16% of survey respondents felt that they did not receive adequate training from their PaaS vendor.

Tip for Buyers

Talk to vendors about the impact on organizational structure and required skill sets before acquiring a PaaS. Not all platforms are the same. Some require considerably more resources to create real organizational value. The necessary discussion to have is around DevOps and required resources to handle this relatively new function.

However, vendors should be able to provide accurate recommendations on organizational structure and total number of resources required. They should also be able to help with recommendations for re-training of existing resources. But as many users have found, training resources by vendors are not always adequate. Be sure to ask vendors to outline their training resources and plans in detail.

4

Spotlight on 7 Leading PaaS Solutions

Development velocity is the driving force behind the adoption of PaaS. By removing some of the infrastructure complexities of developing applications, these platforms increase overall development agility and speed to market. They also help to solve the development skills crunch by making it possible for less-skilled developers to concentrate on business logic and coding, leaving infrastructure issues to be handled automatically by the platform.

PaaS platforms increase overall development agility and speed to market.

This guide aims to give you a snapshot of 7 of the most reviewed PaaS products on TrustRadius. An exhaustive list of platforms can be found on the PaaS category on our website. The product profiles in this report showcase aggregate data from reviews collected on TrustRadius, including the most commonly listed pros and cons, feedback from reviewers, and customer demographic information.

In order to be included in this guide, products must have met the following requirements:

> The product must be in our PaaS software category.

> The product must have at least 30 reviews.

> The product must be 'customer verified'—indicating there have been at least 10 new or updated reviews within the past year.

Products are presented in order of momentum or popularity, based on research frequency—i.e., the number of pageviews they've received on TrustRadius in the last 30 days.

Product Comparison Tables

	Overall Satisfaction Rating	Product Metrics		Reviewer Company Size			Comparison Information
	trScore	# of Reviews	Share of Traffic	Small	Midsize	Entreprise	Most Compared To
Microsoft Azure	8.3	63	32.1%	29%	37%	34%	Splunk Enterprise Oracle Cloud Infrstructure Hadoop
Heroku Platform	8.7	41	9.2%	58%	29%	14%	Microsoft Azure Google App Engine AWS Lambda
AWS Lambda	8.7	30	8.9%	28%	36%	36%	Google App Engine Red Hat OpenShift Cloud Foundry
Google App Engine	8.0	26	7.1%	44%	29%	27%	AWS Lambda Microsoft Azure AWS Elastic Beanstalk
IBM PaaS Solutions	6.8-7.6	54	7.0%	22%	26%	52%	Red Hat OpenShift Google App Engine Kubernetes
AWS Elastic Beanstalk	8.2	189	5.8%	38%	29%	32%	Amazon S3 Google App Engine CloudFoundry
Salesforce Lightning Platform	8.1	10	2.1%	24%	47%	29%	Microsoft Azure K2 Platform Google App Engine

	Overall Satisfaction Rating	Feature Rating										
	trScore	Ease of Building User Interfaces	Scalability	Platform Mgmt Overhead	Workflow Engine Capability	Platform Access Control	Services-Enabled Integration	Development Environment Creation	Development Environment Replication	Issue Monitoring and Notification	Issue Recovery	Upgrades & Platform Fixes
Microsoft Azure	8.3	8.2	8.9	7.6	7.6	8.2	8.4	8.7	8.6	7.8	7.7	8.0
Heroku Platform	8.7	8.6	9.5	7.9	8.1	8.2	8.8	9.0	8.7	8.1	8.3	8.6
AWS Lambda	8.7	8.1	9.3	8.2	8.3	8.5	8.9	7.7	7.5	7.9	7.9	8.6
Google App Engine	8.0	8.4	8.9	8.1	7.9	8.1	7.7	8.3	8.0	8.1	8.0	8.3
IBM PaaS Solutions	--	--	--	--	--	--	--	--	--	--	--	--
AWS Elastic Beanstalk	8.2	8.2	9.4	6.9	7.9	8.5	8.7	8.4	8.1	7.8	7.9	8.4
Salesforce Lightning Platform	8.1	7.5	7.9	6.7	7.4	7.4	7.8	7.4	7.6	7.4	6.2	8.3

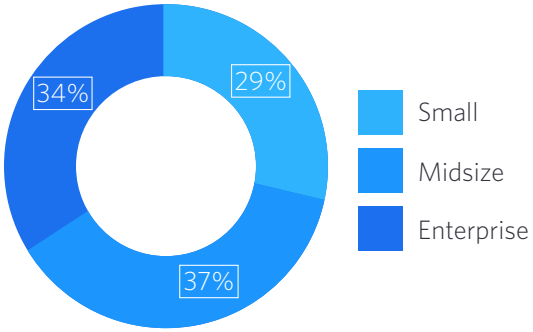
Note: IBM PaaS Solutions comprise three different products. Data on individual products can be viewed on www.trustradius.com.

Microsoft Azure

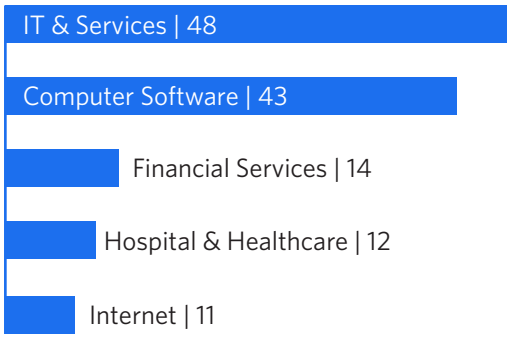
Score 8.3 out of 10

Microsoft Azure is a combination IaaS and PaaS cloud computing platform. The PaaS platform as a service is built on top of IaaS so it provides all the features of IaaS along with its own features, like business intelligence, middleware, development tools, etc. The core of the PaaS platform is Azure App Service for building, deploying and scaling web apps. App Service comprises Web Apps and Mobile Apps for development, and Logic Apps for business logic orchestration. The other key services are Web App for Containers for running Docker container services on the Linux platform, and App Service Environment, providing an extra layer of security and compliance services, and Azure Functions, a Serverless platform for executing code based on events.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



Great Fit For: Development teams that wish to create web apps quickly and securely. It is a good choice for development teams already working in .NET and/or with a need to integrate other Microsoft products like Office 365.

PROS

Great platform for testing, and deploying
(mentioned in 8 reviews)

PaaS support in from developing an application to cloud till deploying it. PaaS support complete life cycle of an application: Building Testing, Deploying, Managing and Updating.

Aditya S. | Application Consultant
Computer Software

ActiveDirectory (AD) integration
(mentioned in 5 reviews)

Active Directory integration. Azure allows for solutions outside your environment to pull your AD information in a non-intrusive fashion to help you automate certain functions such as single sign-on.

Verified User | C-Level IT Executive
Financial Services

CONS

Tech support
(mentioned in 9 reviews)

Support quality can be hit-or-miss, seemingly at random based on the experience and ability of the technician that your tickets are assigned to.

Verified User | IT Director
Architecture and Planning

Documentation
(mentioned in 7 reviews)

Poor documentation for some areas.

Verified User | IT Employee
Financial Services

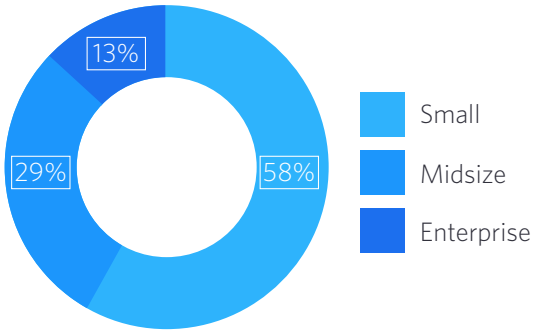
Interested in learning more about Microsoft Azure? Read reviews on TrustRadius.

Heroku Platform

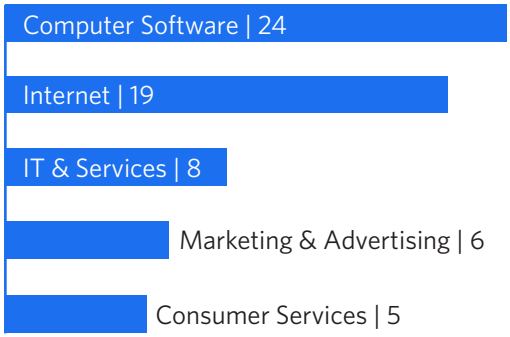
Score 8.7 out of 10

Heroku is one of two PaaS offerings available on the Salesforce platform. The other, App Cloud, is now called the Lightning Platform. While Lightning is really an aPaaS designed for non-coders, Heroku is a code-based platform designed for companies that want to build external customer-facing applications. The platform supports multiple languages such as Ruby, Java, Node.js, Scala, Python, and others. It also provides support for multiple databases including Postgres, Cloudant, Couchbase, and MongoDB.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



Great Fit For: Development teams that want a, code-based PaaS for building customer-facing external applications in a variety of languages that is exceptionally easy to use and deploy. The free option is particularly helpful for rapid development of simple applications.

PROS

Exceptionally easy to deploy
(mentioned in 16 reviews)

Heroku has a very simple deployment model, making it easy to get your application up-and-running with minimal effort. We can focus on our efforts the unique aspects of our application.

Verified User | C-Level Executive
Consumer Services

Great community and large number of add-ons
(mentioned in 15 reviews)

Add-ons: Heroku has a rich add-on library that further saves a lot of time we would spend building things from ground-up.

Verified User | Engineer
Internet

CONS

Expensive at scale
(mentioned in 19 reviews)

The price is not so affordable when you start growing. For small companies, needing small containers, it works quite well but for large applications, it may be too expensive.

William M. | Engineer
Internet

Needs better insight tools to help with dyno scaling
(mentioned in 4 reviews)

Better reporting on how apps scale and whether I should add more dynos or less. At times our site was growing slower and slower and we upped our dynos. It wasn't until we lowered our dynos that the site sped up.

Adam F. | Engineering Director
Computer Software

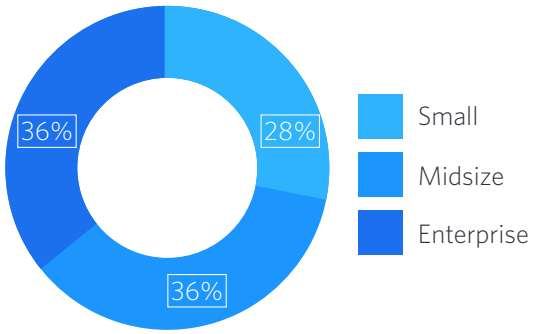
Interested in learning more about Heroku? Read reviews on TrustRadius.

AWS Lambda

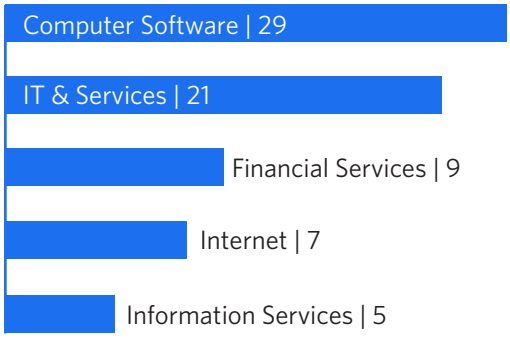
Score 8.7 out of 10

AWS Lambda is a FaaS serverless computing platform which represents an extension of the idea behind PaaS technology. AWS Lambda lets developers run code without provisioning or managing servers. Like all Serverless tools, Lambda is an event-driven computing platform, which means that it runs when triggered by an event and executes code that's been loaded into the system. To give a simple example, evervy time an image is uploaded into a repository, a Lambda function could automatically resize the image. The advantage for developers is that they can run code on Lambda without having to manage servers or even containers. It automatically scales according to the volume of work fed into it.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



Great Fit For: Developers that work predominantly with an AWS infrastructure. Ideal for a bootstrapped environment where keeping costs low and development velocity high is paramount.

PROS

⊕ **Deep integration with AWS services**
(mentioned in 11 reviews)

Well integrated with other AWS services like S3, SQS, IAM, and SNS.

Quang V. | Solution Architect
Computer Software

⊕ **Inexpensive (only charged for the time code executes)**
(mentioned in 13 reviews)

Pay for only what you use. Because Lambda is billed by the 100ms of execution time, you can run low volume services extremely cheaply.

Andrew R. | Software Engineer
Sports

CONS

⊖ **Cold Start issue**
(mentioned in 11 reviews)

Cold starts are a persistent problem which can be mitigated with a simple cron job but would ideally be handled by the platform.

Verified User | Information Technology
Automotive

⊖ **Poor user interface**
(mentioned in 7 reviews)

The UI and Developer experience is not so great. IF you use an abstraction like Serverless Application Model (SAM), things get pretty easy, but it's still AWS UI/DX you're working with after that (which is to say, not their strength).

Verified User | Engineer
Fundraising

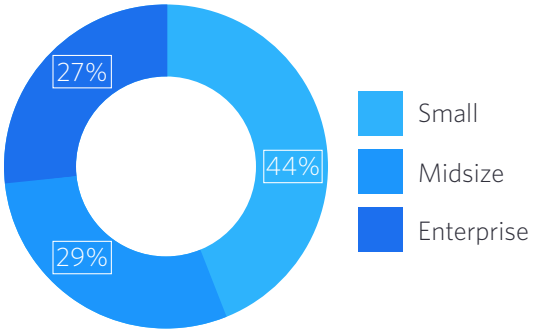
Interested in learning more about AWS Lambda? Read reviews on TrustRadius.

Google App Engine

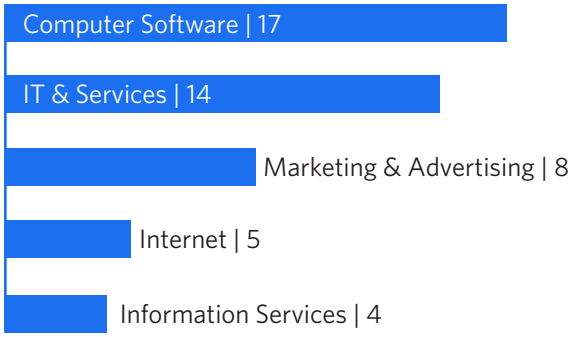
Score 8.0 out of 10

Google App Engine has a high level of abstraction offering a more constrained, sandboxed environment than some competing platforms. It is a platform where developers can begin writing code immediately without dealing with things like web servers or databases. Scalability is also handled automatically, and there is a data-storage API which can be used without worrying about which database it is stored in. The advantage for developers is clear: they can focus on actually building an application, and making users happy since Google has removed most of the hassle from operations work. App Engine makes a lot of sense for developers also using other Google services, such as Google Compute Engine, Google Cloud SQL, or Google Cloud Storage.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



Great Fit For: Developers that want to quickly prototype ideas and have operations such as scaling up/down, fault-tolerance, zero-downtime releases, and host server management all handled automatically. It is also a good choice for development teams already using other Google services.

PROS

Automatic scaling
(mentioned in 10 reviews)

The scalability testing of Google App Engine is top notch. You can quickly and efficiently test if your new app will support millions of users.

Verified User | Project Manager
Information Technology

Integration with other Google products
(mentioned in 5 reviews)

Easy integration with other Google Cloud products, e.g. Datastore, Pub/Sub, Cloud Storage, etc.

Dmitry S. | Software Engineer
Legal Services

CONS

Documentation
(mentioned in 9 reviews)

Documentation does not always keep up with the latest changes to the service... At times, we were surprised to find out that something we didn't think was possible was, or, conversely, something that was supposed to work fine which had been deprecated.

Jonah D. | Web Developer
Computer Software

Pricing
(mentioned in 6 reviews)

Price. Google App Engine isn't cheap. But, you get what you pay for. Rock solid service, great tools, at a hefty price.

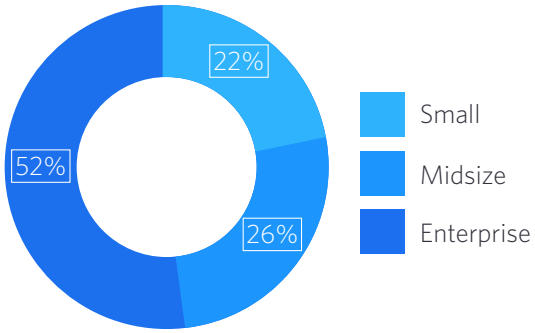
Jonah D. | Web and Mobile Developer
Computer Software

Interested in learning more about Google App Engine? Read reviews on TrustRadius.

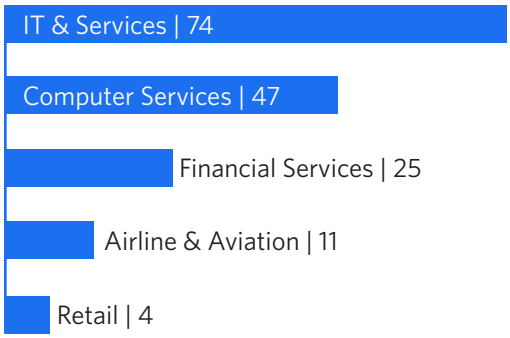
IBM PaaS Solutions

IBM's PaaS offerings are based on a number of different services including the open-source cloud platform CloudFoundry, IBM Functions for event-driven FaaS capabilities, IBM Cloud Kubernetes Service for Containers-as-a-Service (and an alternative to CloudFoundry), a Cloud Kubernetes add-on called Istio for the c ollection and display of application-related metrics, and IBM Cloud Private which is a platform for developing and managing on-premise containerized applications. This section will cover IBM Cloud Kubernetes Service, IBM Cloud Foundry, and IBM Cloud Functions.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



IBM Cloud Kubernetes Service

Score 7.6 out of 10

IBM Cloud Kubernetes Service is an open-source tool for managing containerized software at scale. It is a Container-as-a-Service platform that is an alternative to the Cloud Foundry PaaS. IBM offers two versions: IBM Cloud Private for handling containerization of on-prem apps, and a managed service in the public cloud. Although Kubernetes is easy to install and set up it can create headaches in managing multiple clusters. For this reason, IBM has built a multi-cloud manager to help manage resources across many different clouds.

Great Fit For: Developers who wish to bundle and run their applications in containers. Containers eliminate many of the manual processes involved in deploying and scaling containerized applications, and IBM Cloud Kubernetes helps easily and efficiently manage those clusters.

PROS

⊕ **Strong scalability**
(mentioned in 7 reviews)

Highly Scalable. We keep adding nodes when needed.

Lokesh V. | Subject Matter Expert
Logistics and Suply Chain

⊕ **Excellent Support**
(mentioned in 3 reviews)

Great support when needed.

Grant G. | Product Development Scientist
Consumer Services

CONS

⊖ **Can be unreliable: some users report service issues**
(mentioned in 5 reviews)

Breakdown of service may occur at any time, which leads to malfunction of applications.

Verified User | Analyst
Computer Software

⊖ **Documentation could be better**
(mentioned in 4 reviews)

Our developers felt that the documentation was difficult to navigate. Time spent in the platform cleared the problem for the team.

Damani B. | Managing Consultant
Management Consulting

Interested in learning more about IBM Cloud Kubernetes Service? Read reviews on TrustRadius.

IBM Cloud Foundry

Score 6.8 out of 10

IBM Cloud Foundry is used by developers to create, manage, run and deploy applications for both the public and private cloud. As with other PaaS, the main advantage for developers is that they do not have to worry about how to manage cloud resources to run their apps as the platform does that for them.

Great Fit For: Larger enterprises that need to increase development velocity and manage the entire application development lifecycle from development, through testing to deployment. The platform is also well-suited to companies that want to run applications in multiple programming languages, allowing developers to use the cloud platform that best suits specific application workloads.

PROS

⊕ **Easy to deploy applications**
(mentioned in 4 reviews)

Using Cloud Foundry allows us to deploy quickly and easily with no hassles. It is simple to manage multiple environments such as staging and production.

Kim D. | Full-stack Developer
Computer Software

CONS

⊖ **Weak documentation**
(mentioned in 4 reviews)

Needs better documentation for all functionality.

Priya M. | Software Engineer
Information Technology and Services

Interested in learning more about IBM Cloud Foundry? Read reviews on TrustRadius.

IBM Cloud Functions

Score 6.9 out of 10

IBM Cloud Functions is a serverless programming platform based on Apache OpenWhisk. Developers use their favorite programming language to write code ("actions") that responds to external events. Actions are hosted and executed in IBM Cloud, and scale on demand based on the number of events. There are no servers or other infrastructure to provision and manage. Developers can focus on writing app logic instead of worrying about infrastructure issues like scaling and availability.

Great Fit For: Development of lightweight microservices for cloud apps, and event-driven data processing. IBM Cloud Functions is also a good choice when there are no long-running processes and when budgets are tight, as users are only charged when the code executes.

PROS

⊕ **Easy to trigger when needed**
(mentioned in 3 reviews)

Good for triggering IoT apps when certain criteria/ validations are met.

Verified User | Project Manager
Internet

CONS

⊖ **Platform still unstable**
(mentioned in 4 reviews)

Breakdown of service may occur at any time, which leads to malfunction of applications.

Verified User | Analyst
Computer Software

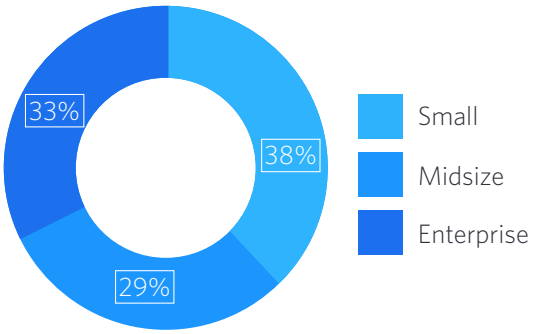
Interested in learning more about IBM Cloud Functions? Read reviews on TrustRadius.

AWS Elastic Beanstalk

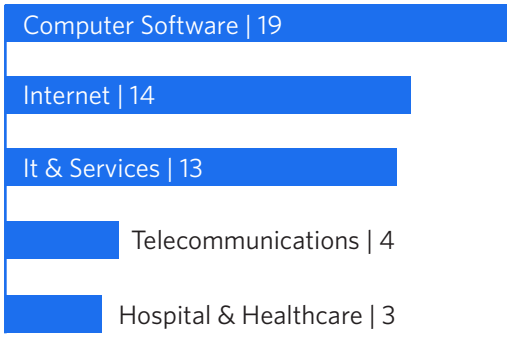
Score 8.2 out of 10

AWS Elastic Beanstalk is a good example of a PaaS tied to a specific operating environment. AWS Elastic Beanstalk is a cloud deployment and provisioning service that automates the deployment of applications on the Amazon Web Services (AWS) infrastructure. Once developers upload their applications, load balancing, scaling and application health monitoring are all handled automatically. There is no extra charge for Elastic Beanstalk. Customers are only required to pay for resources used to store and run their applications. The platform has powerful customization features and provides lower level access and control than some competing platforms.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



Great Fit For: Developers working on AWS infrastructure. Elastic Beanstalk has been designed specifically for the deployment of applications on AWS and is not a general purpose PaaS that can be used with alternative infrastructure.

PROS

Excellent auto-scaling capabilities
(mentioned in 11 reviews)

Scalability: The ability to autoscale based on traffic helps with availability and overall cost.

Jordan L. | IT Manager
Marketing and Advertising

Easy to set up and get apps deployed
(mentioned in 9 reviews)

Easy to set up deployment of many technologies.

Richard R. | Software Developer
Computer Software

CONS

Steep learning curve
(mentioned in 6 reviews)

Difficult to understand. No matter how cute and easy the AWS videos sound, I find it difficult to understand. There are just too many configurations.

Rahul C. | Software Engineer
Information Technology and Services

Poor documentation
(mentioned in 4 reviews)

The product is generally very good, but if there is one thing I'd improve it is the web-based user interface for managing instances.

Verified User | Information Technology Executive
Computer Software

Interested in learning more about AWS Elastic Beanstalk? Read reviews on TrustRadius.

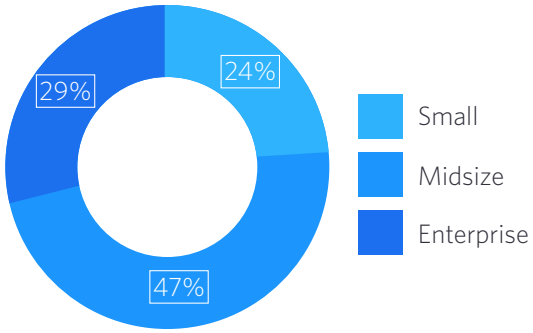
Salesforce Lightning Platform

(formerly Salesforce App Cloud)

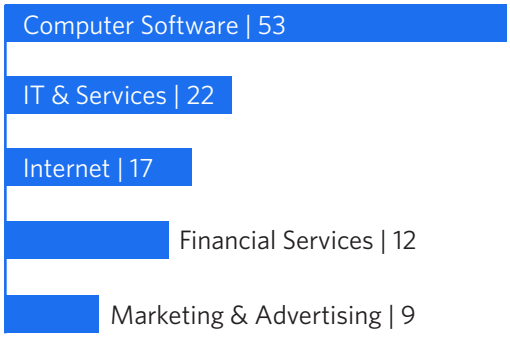
Score 8.1 out of 10

Salesforce Lightning Platform is a PaaS platform well-suited to the development of internal, employee-facing apps, rather than external, customer-facing applications. It has been designed for the business user who wants to create internal apps and doesn't have a technical background in development; the tools are point-and-click and easy to use. Users create apps by dragging and dropping components into the Lightning App Builder, and then automate business processes with the Lightning Process Builder. Salesforce Platform is a good example of a PaaS strongly tied to a SaaS product to encourage 3rd-party development on that product, rather than a general purpose PaaS.

Reviewer Company Size Distribution



Top 5 Reviewer Industries



Great Fit For: Business users with no coding experience who need to create internal Salesforce apps using a simple drag and drop interface.

PROS

⊕ Excellent dev. environment for mobile and web apps
(mentioned in 5 reviews)

Powerful enterprise development platform for mobile and desktop. For enterprise customers already using Salesforce, App Cloud is a viable solution to develop and deploy for both mobile and desktop.

Verified User | Engineer
Health, Wellness and Fitness

⊕ Access to pre-built apps on AppExchange
(mentioned in 4 reviews)

The Salesforce App Cloud was well suited when it came to installing new packages (apps) to help our environment accomplish things that traditional Salesforce Sales and Service Cloud couldn't accomplish.

Verified User | QA Administrator
Non-Profit

CONS

⊖ Can be expensive
(mentioned in 5 reviews)

Very pricey, as are all things Salesforce. Hard for startups/emerging companies to justify.

Verified User | Sales
IT and Services

⊖ Support and training limited
(mentioned in 3 reviews)

Training and access to non-admins seems very limited and hard to find. Resources are hard to locate at times.

Hayden A. | Account Development
Computer Software

Interested in learning more about Salesforce Lightning Platform? Read reviews on TrustRadius.

We hope that you find this guide helpful for your software buying journey! If you have any questions about this Buyer's Guide or its contents, please contact us anytime.

research@trustradius.com
facebook.com/trustradius
twitter.com/trustradius
linkedin.com/company/trustradius
Instagram: @trustradius

About TrustRadius Reviews

TrustRadius scores and visuals are designed to create the most accurate, data-based picture of products in the marketplace. Learn more about our research methodology data integrity standards here. Find more information on licensing this guide or your company's individual TrustBrief profile page here.