

Ten Advantages of Platform-as-a-Service

WHITE PAPER

ENGINEYARD.COM / 1-866-518-9273

Looking to develop, deploy, and manage applications in the cloud?

Learn how PaaS empowers you to innovate faster, more cost effectively and with less risk.



Executive Summary

To get an application running in the cloud you can spend long hours downloading, compiling, installing, configuring, and connecting all sorts of components—and that is just to get your application up and running on a single virtual server instance. Not only can this be time consuming and costly, but it takes away from time you could have spent innovating and improving your application.

But now there is a better way. In the last several years, technologies between the operating system and your application—the platform layer—have evolved and matured to make cloud computing easier.

Rather than downloading and building all of those platform-level technologies on each server instance, and then later having to repeat the process as you scale, you can go to a simple Web user interface, click a few options, and have your application automatically deployed to a fully provisioned cluster. As application usage grows, you can later add more capacity by clicking a few more buttons. As you need to set up increasingly sophisticated architectures such as high availability and disaster recovery, you can do this from the same Web interface. And as all the constituent platform components evolve along their own respective paths, they are updated for you, with no required effort on your part. That is what Platform-as-a-Service (PaaS) is all about.

You get a reliable comprehensive application platform that lets you retain full access to all the environment components with “white-glove” expertise and support if you ever have questions or issues with your system. You can focus on improving your application while Engine Yard takes care of your platform.

There are numerous benefits to using a PaaS rather than “doing it yourself” on an IaaS, including agility, efficiency, quality, and other needs. We briefly outline the most important of these benefits below.



PaaS consists of three main components. First, there are the layers of software that your application runs on—“the stack”. These are the various libraries, frameworks, and services that the developer uses to build the application and which are assumed to be present in the runtime environment. The stack consists of components such as the language interpreter or language virtual machine (VM), the application framework (e.g. Rails, Lithium), the HTTP server, the load balancer, caching mechanisms, databases and so on. A given PaaS may offer several stack combinations to choose from, such as different stacks for different languages or different frameworks within a language domain. The stack remains with your application as it runs; logically, it becomes essentially a part of your application.

The second major component of a PaaS is the deployment machinery that instantiates virtual servers and provisions them with instances of the stack and your application code. This machinery operates when you click the deploy button (or perhaps invoke it from a command line), and it gets out of the way once deployment is complete and your application is up and running. This machinery is itself code, perhaps a combination of scripts and Web services and perhaps uses an off-the-shelf technology such as Puppet or Chef. The way this machinery is architected, the particular parameters it exposes, and the functions it makes available to an overlying graphical user interface (GUI) or command line interface (CLI) are important differentiators between a good PaaS and a bad one.

The third component is the user interface and the overall user experience (UX). A particular PaaS may provide Web GUI, a CLI or both. The ordering of the screens, the choices made available, the logic of how multiple applications and environments are organized and presented—all these factors are make-or-break for the learnability and usability of a given PaaS. The goal is to make it easy to change the things you care about and hide the things you don't care about.

The right trade-offs between simplicity and flexibility, constraint and freedom, and opaqueness and transparency are all critical.

Choosing the right PaaS takes some thought and understanding of your needs and goals. It's not trivial, but making the right choice can be of huge benefit to your business.



Ten Reasons to Use a PaaS Instead of Doing it Yourself

No matter what business you're in, and no matter what stage your business is in, PaaS can help you innovate new offerings, expand your capacity, extend your customer base, and grow quickly. Here are a few examples of the value PaaS can bring to your business.

1. Innovate Faster

First and foremost, using a PaaS to deploy and run your application enhances your agility.

The Web is accelerating the pace of innovation. To compete, you need to quickly transform new ideas into real applications and evolve those applications with agility in order to meet fast-changing business and technical requirements. Market opportunities exist very briefly. Your business needs to build, deploy, and iterate in days or weeks, not months or years.

Setting up platform-level software to run your application is time-consuming and complex. By simplifying, automating, and in many cases eliminating the steps associated with setting up the foundation for your application, you can get your application deployed much more quickly in the first place, and you can iterate, adapt, and extend it more rapidly over time.

"It takes us 50% less time to deploy on Engine Yard Cloud because it's so easy to configure servers. We can deploy with just a few clicks and add another instance with just one more click—and on top of it all, everything is pre-configured by Rails experts at Engine Yard who could write the book on best practices for rapid deployment."

—Josh Krall, CTO, TransFS



Ten Reasons to Use a PaaS Instead of Doing it Yourself

Do-it-yourself on EC2

Initialize

1. Build app
2. Set up EC2 account
3. For each desired instance:
 - a. Install/configure OS
 - b. Install desired language version, runtime
 - c. Install frameworks and other libraries/gems
 - d. Install/configure application server
 - e. Install/configure HTTP server
 - f. Install/configure load balancer
 - g. Install/configure other components (DB, cache)
 - h. Debug integration of stack
 - i. Install/configure app on stack
4. Get instances working together

Update

1. Repeat items a-i above, per instance
2. Get instances working together

Scale

1. Do a-i above for new instances
2. Reconfigure application servers, load balancers, DB, etc.
3. Ensure consistent stacks
4. Get instances working together

Use Engine Yard PaaS

Initialize

1. Build app
2. Set up Engine Yard account
3. Click several configuration choices
4. Click "Boot"

Update

1. Repeat items a-i above, per instance
2. Get instances working together

Scale

1. Use Add Instances wizard



"By deploying on Engine Yard Cloud, we've saved at least 60% on engineering resources—this productivity boost translates directly to more features and faster time to market."

- Chris McNeilly
CTO, Motista

#2 Focus Resources

Eliminating much of the overhead to deploy and manage applications doesn't just mean you can do certain things faster. It means you don't have to do certain things at all. Which means that you don't even need to know how to do them. Which allows you to be even better at knowing how to do the things that differentiate your business, like building applications with innovative features and exceptional user experiences.

Let's say you're an entrepreneur with a great idea and some seed funding, or you're an enterprise line-of-business manager with some spare budget you'd like to put on a special project. Do you want to spend your precious funding on some generalist developers who can code moderately well and also do system administration, or would you rather get top-notch coders who don't necessarily have deployment expertise? Would you rather spend a dedicated headcount on an ops person or have an additional application developer?

#3 Save Money

Focusing development resources and spending less headcount on unneeded expertise are both benefits that intuitively translate into reducing costs. But beyond these obvious things are even more ways that PaaS saves real money compared with doing it yourself on IaaS.

With PaaS you are tapping into a real economy of scale. Imagine the number of hours it would take to set up the core stack—the platform-level components—for your applications. Imagine the hours consumed on an ongoing basis to maintain the stack. Imagine the cost of those hours, and consider the incremental value from this work to your application. Now consider those same costs amortized across thousands of applications. There is very little differentiating value from doing this low-level work yourself, so clearly, buying platform from a provider is more efficient than building it yourself.

There are also less obvious hidden costs, such as the cost of downtime when one of your administrators makes a mistake configuring your application server and no one can access your Web application for hours. According to a study by Uptime Institute, 70 percent of data center downtime is caused by human error. Consider both the hard costs of downtime, such as lost business and unexpected support costs, and the soft costs, such as idled employees and a tarnished reputation.



"With Engine Yard, we can deploy our application in five minutes, and we know we're running on the best technology stack. If we were managing our own servers, we'd have to take resources away from development to invest in a dedicated IT/infrastructure engineer. Instead, we're able to focus on building new features."

- Damon Danieli
Founder and CTO, Z2Live

#4 Get the Best Technology

The benefit of economies of scale doesn't simply stop at getting the same thing for less money. What you actually end up with is something better, for less money.

The stack and platform-level technology you would build yourself will almost never be as good as what a top PaaS will provide. Few companies have both the ability to pay and the attractiveness to hire the world's best platform builders. The top platform builders are in companies whose main business is platform. People who are world-class in a given discipline for the most part want to work somewhere where that discipline is core to the business, not in an ancillary or supporting role.

A PaaS typically employs specialists who constantly tune, optimize, load-balance, reconfigure, and so on. The result of faster page loads is often a reduction in bounce rates, because customers are more satisfied with the service level they experience. And since search engines use bounce rates and page load times to prioritize paid search rankings, faster application performance can substantially improve your application visibility and business performance.

We've mentioned the stack itself several times, and getting the best stack means getting a stack with the latest versions of all components, configured optimally to work with each other and as the foundation for applications such as yours. In addition to the stack itself, there is the deployment mechanism, the platform software that instantiates virtual servers on the infrastructure and installs instances of the stack on them. Many aspects differentiate a good deployment mechanism from a bad one: what configuration parameters are exposed, what component options exist, how stack versions are managed, what activities are automated, what components are pre-built into binaries versus compiled at deployment time, and so on.

Above all of this, there is the user interface, which may include GUI and CLI variants. Getting the user interface and user experience right is make-or-break for platform interaction. The best platform user interfaces will be simple yet flexible—the things you don't care to customize shouldn't get in the way, and the things you do want to customize should be easy to do so. The interface needs to be both learnable and usable—i.e. quick to get started with but also powerful enough to support the expert user.



"Engine Yard Cloud is a scalable, reliable platform that's been developed and fine-tuned by cloud experts. We have an easy and repeatable deployment process that is 200% better than managing our own datacenter."

- John Schult
Director of Product
Engineering, Vitruve

"We were the victims of our own success—we had so many people using our applications we couldn't keep up with the need to expand. On top of that, we're a game studio, not an infrastructure company, so we were inexperienced when it came to configuring servers to scale based on changing traffic patterns. This is where Engine Yard really shines because scaling Rails applications is one of their specialties."

- Eric Peng
CEO, PlayMesh



#5 Stay Up to Date

A particular challenge of deploying your application on a self-built stack is the sheer number of components that need to be tracked, maintained, updated, and re-integrated over time. It's one thing to get it all set up and humming along in the first place, but the first time you need to swap in an update to the app server or the load balancer or the cache you may find yourself in a nightmare of reconfiguration. One bad experience like this leads many do-it-yourselfers to remain indefinitely on an increasingly outdated stack for fear of rocking the boat. The downside of course is that you end up missing out on the latest security updates, performance improvements, and new features—and what may have started as competitive advantage becomes a weighty impediment to keeping up with your competitors.

With PaaS, you not only get the best possible stack as of the moment you deploy, you also get a stack that keeps up with you over time, ensuring that your application is always running on the latest and greatest. You don't fall behind your competitors, and you also don't waste time and incur risk by doing it yourself. The PaaS experts constantly incorporate and test component updates and bring them into the platform. At Engine Yard, updates are rolled out in a way that minimizes risk of incompatibility and gives you complete control over how updates are brought into your production applications.

#6 Maximize Uptime

PaaS offerings can help you achieve your availability goals and give you innovative new disaster recovery/business continuity options. PaaS vendors have the tools, technologies, and experience to help you avoid the unplanned outages that cause downtime. The best PaaS vendors embed technologies and techniques in their products to keep availability high enough that they can offer service-level agreements (SLAs) at or above 99.9% availability. Engine Yard, for example, employs application templates and configuration recipes that minimize human error, ensure up-to-date snapshots of content assets, and allow fast and easy rollbacks if something goes awry.

Beyond basic data backup and OS hardening, PaaS vendors can protect your data in other ways. With Engine Yard, for example, completed transactions can be recovered in the case of database failures, so you never lose data from a committed transaction. And because application configuration is all captured in application templates and recipes, it's easy to reproduce your application in another availability zone if the primary zone has a disruption. Engine Yard automatically spreads your application across multiple availability zones, enhancing application resilience.

"The best thing about relying on Engine Yard for our deployments is that they are dedicated stack experts, so I can trust that they've selected battle-tested production configurations for Rails applications. Because of this, we've reduced the time we spend each week on system operations."

- Nathan Phelps
Co-founder & CTO,
HealthLeap

#7 Scale Easily

It's one thing to get the best technology at the best cost for a given size of business. It's another thing to achieve that at many different sizes—potentially spanning orders of magnitude—as your business grows. When building a platform yourself, you basically have three choices: you can optimize for the scale you're at now, you can optimize for a scale you expect to be at a later date, or you can invest a lot in building your own scaling mechanism. In the first case you risk having to redo your platform and incur downtime when you outgrow your initial set-up. In the second case you will likely waste resources due to overprovisioning. And in the third case, you will like spend a lot of opportunity cost building something that ends up not nearly as good as what you can get from a PaaS.

With a PaaS, on the other hand, you get the benefit of a great scaling mechanism developed by experts over time and in response to the needs of many customers. On top of that, the PaaS scaling mechanism leverages the underlying infrastructure's elasticity but presents it in an easy-to-use way, abstracting the complexity of the mechanism's details. For example, to add instances in the Engine Yard Cloud you simply click the 'Add' button, and a wizard walks you through a couple of checkboxes.

#8 Strengthen Security

Security showcases another distinct advantage of the PaaS model. With the sheer volume and the diversity of security threats on an upward spiral, protecting against attacks is best left to specialists. A PaaS offering provides continual security updates for individual stack components as they are issued. At Engine Yard the stack engineering team maintains fully updated Rails and PHP technologies, so security vulnerabilities in core language or framework components are quickly remedied and customers are automatically notified following the patch.



#9 Do Your Project Right

An interesting behavior happens in real-world enterprises when departments or lines of business can't get corporate approval or fast enough turnaround to build and deploy a new Web application; they tend to do it anyway. Through the back door. They go to a digital media agency; they re-channel some discretionary budget and hire consultants; they get it done using the "easier to get forgiveness than permission" principle. These unofficial or back-door efforts usually result in a mess, sooner or later. At the very least, they add to the complexity and inefficiency of the environment, because they introduce new and unapproved tools, processes, and/or infrastructure. In other cases they create an even bigger mess: what if the new Web service really takes off, and suddenly the LOB manager needs to procure budget to scale up? What if the new Web service unexpectedly disrupts, corrupts, or causes downtime in legacy applications? It might not be as easy to get "forgiveness" as presumed.

PaaS dramatically cuts the risk, cost, and complexity of new projects. It brings predictability to both the cost and the ramifications of introducing new Web applications and services. To the extent needed, a PaaS offering can complement and add value to existing development and IT operations. Simply put, it doesn't need to come in through the back door.

#10 Get the Best Support

As outlined above, when you build and run on a PaaS, you use technology that has been developed and refined in response to the needs of thousands of customers. But it's not just the technology that embodies that aggregated expertise. It's the people themselves. When you call Engine Yard Support, you speak to someone who has dealt with hundreds of problems in the same domain as yours. You speak to someone who has access to—may even be sitting next to—some of the leading experts in the community, whether for core Ruby or PHP language stack components or complementary open source projects.



Why Engine Yard?

Once you've made the decision to adopt the PaaS model, the next question is which provider to choose? Engine Yard, one of the pioneers of PaaS, has been running cloud-based platforms for more than five years and has some of the deepest expertise and over 2000 successful production customers.

With a unique combination of Ruby on Rails and PHP expertise, operations orchestration, open source community involvement, and world-class service, Engine Yard empowers you to build and run cloud-based applications easily and cost-effectively. Thousands of customers, from Web startups to Fortune 500 enterprises, trust Engine Yard for fast deployment, rapid scaling, high performance, and 24×7 uptime, allowing them to leverage cloud agility for competitive advantage. Three main offerings are available:

Engine Yard Cloud is a self-service, elastic PaaS that runs atop Amazon EC2. With an easy-to-use deployment and management dashboard, you can have your Ruby applications up and running in minutes and scaling easily as your business grows. With deep support for Rails as well as other frameworks, plus a range of database and caching options including MySQL, PostgreSQL, memcached, and Redis, Engine Yard Cloud is the go-to place in the cloud for running Ruby Web applications.

Orchestra is a self-service, elastic PaaS that runs atop Amazon EC2 and provides all you need to get PHP applications up and running quickly. A free tier allows you to experiment with small applications as long as you wish, and when you're ready to move into production and grow, the platform scales with you. Numerous PHP frameworks and supporting technologies are provided with click-button simplicity.

Engine Yard Managed is a proven enterprise-grade managed service that delivers everything you need to get started quickly and run your applications, including operating system, database, application server and more. Running on Terremark's highly customizable and fully auditable infrastructure, our massively scalable platform provides you the flexibility and white-glove service you need to grow and run your business while allowing you to focus on building great applications.

A key advantage of Engine Yard's offerings is superior customer service. We recruit top-caliber Ruby and PHP talent—people who have vast experience with applications and who contribute to related open source projects—and we make their expertise available to you 24 hours a day, seven days a week. Have a question about how to use an Engine Yard product or how to choose the best runtime? Our team is ready and waiting to help you out.



Engine Yard also offers a range of support options to help you deploy and maintain your applications in a production environment. Several levels of support are available, from self-support to standard and premium packages. We also provide a broad spectrum of application services, such as database and performance analysis, as well as customized professional services to help you develop, deploy, manage, and scale Web applications.

In addition, Engine Yard gives you access to a thriving partner ecosystem. Hand-picked for their experience and technical capabilities, these developers, digital agencies, and technology providers in our various partner programs represent some of the best and brightest in cloud computing today.

Conclusion: Innovate Faster with More Reliability, Flexibility and Less Risk

If you're looking to accelerate innovation and reduce cost, risk, and complexity as you move to the cloud, it's time to take a close look at PaaS. By taking advantage of PaaS offerings from Engine Yard, you'll eliminate the inefficiency and excessive expense of trying to do everything yourself. You'll leapfrog the learning curve of IaaS and the limitations of traditional managed services offerings. And you'll do so without sacrificing flexibility and without compromising. In short, you'll transform Web application development and deployment from a resource to be maintained into a source of competitive advantage.

Learn More

For additional information about the Engine Yard PaaS offerings, technical details about the Engine Yard platform, and customer references, please visit our website at www.engineyard.com.





Engine Yard™

© ENGINE YARD 2014, ALL RIGHTS RESERVED

Contact

EngineYard.com

1-866-518-9273

Engine Yard
PO Box 77130
San Francisco, CA 94107-1713



13 TEN ADVANTAGES OF PLATFORM-AS-A-SERVICE