WHITEPAPER

THE ROAD TO ENTERPRISE PaaS

Gordon Haff

EXECUTIVE SUMMARY

Platform-as-a-Service (PaaS) provides an abstraction that makes developers more productive by helping them focus on creating applications rather than dealing with infrastructures. This is important as information and its associated processing has become central to more businesses, even those in industries that haven't been historically thought of as dependent on information technology. And that makes writing custom applications in support of an organization's business objectives more important than ever.

PaaS offerings have generally evolved from point solutions that focused on a single programming model to ones that are more general. For example, multi-language ("polyglot") PaaS offerings that support both traditional enterprise tools like Java and newer-style scripting languages are increasingly the norm. However, it's become clear that many organizations want to also choose their PaaS operating models. That's why Red Hat® has expanded beyond its OpenShift Online by Red Hat solution to also offer OpenShift Enterprise by Red Hat, a commercial subscription version that IT shops can install in their own datacenters or on other dedicated hardware. As with all other Red Hat subscription offerings, OpenShift Enterprise takes the open source code that Red Hat has made available to the community through the OpenShift Origin project and makes it into a reliable and supported product for enterprise use.

Like OpenShift Online, OpenShift Enterprise leverages a variety of technologies in Red Hat Enterprise Linux to enable applications to run in multi-tenant environments in a secure and predictable manner. OpenShift Enterprise is also based on an extensible architecture that allows for the integration of a broad ecosystem of middleware and tools. In short, OpenShift Enterprise delivers everything needed to develop the applications that are increasingly central to today's information-driven businesses. And it does so in a way that maintains application portability across hybrid and heterogeneous IT infrastructures.

OPENSHIFT ENTERPRISE by Red Hat

INTRODUCTION

The PaaS moniker covers a lot of ground. At its broadest, it's almost a generic term for web APIs. At its narrowest, it's a set of programmatic interfaces to a specific hosted application – essentially a way to extend Software-as-a-Service (SaaS). Perhaps a better definition of PaaS is the abstraction that lets developers focus on writing, running, and managing applications without having to unduly concern themselves with low-level plumbing such as provisioning and tuning operating system images.[1] It's a means to make developers more productive by letting them focus on developing rather than the requisite infrastructure.

1 As a first approximation, you can think of IaaS, PaaS, and SaaS as successively abstracting the hardware and software of a datacenter—creating simplicity in exchange for reduced ability to fine tune the layers that have been abstracted away.

Given that context, it's not surprising that most initial PaaS offerings were hosted services. After all, a hosted service is almost always going to serve as an easier and preferred on-ramp for developers who don't want to worry about operational details. That's certainly been Red Hat's experience with its OpenShift Online offering, which makes deploying an application as easy as pushing it to a code repository. A PaaS like OpenShift Online handles details like auto-scaling, self-service, and monitoring applications, leaving developers to focus on creating applications with familiar tools, languages, and frameworks.

However, a hosted service is, well, hosted by someone else. And, for many organizations, that loss of control isn't acceptable, at least not for all of their applications. Using a PaaS that makes use of standard languages and development frameworks helps to a degree; once developed in that manner, an application can be deployed in any environment provisioned with appropriate operating system or middleware runtimes. But because such a transplanted application loses access to the operational automation of a PaaS platform, a hosted service may still not be seen as ideal.

The best alternative for many enterprises is a hybrid PaaS approach, which allows organizations to gain the advantages of a PaaS while operating it in the manner of their choosing – whether hosted, on-premise, or a combination of the two. This paper takes a look at the operational decisions associated with a PaaS in the enterprise. It also details OpenShift by Red Hat, the company's on-premise and hosted PaaS offerings, to let you see how a PaaS can provide automated services, multi-tenant quality-of-service and security, "polyglot" mutli-language/framework support, and a choice of operational models.

But, first, let's take a deeper look at why developers (and, increasingly, IT departments) love PaaS.

STREAMLINING APPLICATION DEVELOPMENT WITH PaaS

A lot of the attention around cloud applications tends to focus on Software-as-a-Service (SaaS), which is to say complete applications delivered directly to the users of those applications. And SaaS is indeed an important cloud story. Whether talking about consumers and smaller businesses gaining access to applications they didn't have the resources or skills to run in-house in the past or talking about largely undifferentiated applications like email,[2] SaaS eliminates the need for a lot of routine IT that doesn't add value.

2 Though at scale, even seemingly routine apps often come with their own unique integration and IT governance considerations.

However, the infiltration of technology into more and more types of businesses — even ones that we didn't historically think of as especially dependent on information or computing, such as agriculture — has also fueled a huge appetite for custom applications. Arguably, the only thing staunching that appetite is the money and time it takes to develop applications. And that's what makes PaaS so interesting given how it's focused on increasing developer productivity. Such productivity has a direct relationship to how quickly businesses can bring new services and products online — and how quickly they can start making money for the organization.

Some of this productivity increase comes from faster access to resources; no waiting for IT to order servers, provision them, and provide access. But that's really a benefit of cloud computing more broadly — whether a public cloud or a properly planned private one.[3] The big benefit of PaaS specifically is that it lets developers focus on the tasks that matter in application development and ignore those that don't. This means designing compelling user interfaces and appropriate database architectures, not configuring firewall settings or tuning operating system resource limits.

3 The Red Hat webinar series 4 Steps to a Private or Hybrid Cloud details the planning and implementation process in detail. http://www.redhat.com/solutions/cloud/resources/fromvirttocloud/

The benefits of PaaS aren't just limited to developers though. The industry is experiencing a shift away from strictly segregated operations and developer roles. The term "devops" was coined to capture this idea that application development increasingly embeds operational concepts such as availability and scaling. In part, this is the result of architectures that deal with these concepts at the application layer, rather than the underlying infrastructure.

PaaS is a great match for a devops model because the PaaS itself can help provide auto-scaling, self-service, standard pre-configured services, and other features that historically would have been considered part of IT operations. Under a hosted model, these capabilities are all provided directly to the developer and the underlying plumbing (e.g., adding server resources as needed, remediating infrastructure failures, monitoring for security issues) is handled by the service provider. With an on-premise PaaS, on the other hand, IT operation teams control and are responsible for the infrastructure and can choose to pass a degree of that control over to the developers. PaaS can therefore be a win for admins as well as developers by providing a framework and tools for delegating responsibilities while retaining overall control of policy.

Many organizations see PaaS as an opportunity to standardize development workflows as one important path to increased consistency and productivity. Standardization (within a given organization) can only become more important as the number and scale of applications increase and as more organizations move to more agile development models and more rapid, incremental updates to production systems.

BRINGING PaaS TO THE ENTERPRISE

The tremendous excitement around PaaS can be seen both through the growing adoption within the developer community, and through the amount of vendors entering the space with both online and on-premise PaaS solutions. Although initially used primarily by early adopters and innovators — including individual developers, start-up companies, and small companies — PaaS is entering the mainstream.

What is enabling this mainstream adoption by larger enterprises and a broader section of the market?

One change is that PaaS is becoming more multi-lingual. Red Hat has conducted surveys at several events that consistently reveal that most respondents intend to develop software for cloud environments using languages similar to the ones they use today.[4] PaaS platforms that limit developers to a specific language on a specific hosting platform have often been criticized by developers because they constrain this choice. It's telling that a number of language- and framework-specific PaaS offerings have shifted toward a more polyglot (multiple languages/frameworks) approach.

4 While there's much ongoing innovation in middleware, tools, and languages, there's also considerable inertia in practices and established skills that any organization takes into account when planning development methodologies.

OpenShift, for example, has cartridges that serve as the mechanism through which platform services are exposed. The initial release of Red Hat's on-premise offering included cartridges for Java, PHP, Ruby, Perl, and Python, as well as MySQL and Postgres for data services. OpenShift also has community-supported node.js and MongoDB flavors.

And Red Hat is expanding that community. Red Hat wants to see new cartridges, both for newer versions of the technology that it already has, as well as new technologies that people want to incorporate. New languages, new frameworks, new data services, and other add-ons.

Another important consideration is the support for different operational models. An online service serves as a simple on-ramp for a developer who just wants to try out the service. Certain new-style Internet businesses will even be comfortable using an online service for all their production applications — especially if, as in the case of OpenShift, they can have confidence that the code they write can be moved elsewhere without making changes.

However, many organizations still prefer or require that at least a subset of their applications run in an environment over which they have full control. They want the operational benefits of PaaS but they want to be in charge of the infrastructure. They may also prefer to expose fewer options to developers than is typically the case under a devops model. Arguably, this more traditional "ITOps" philosophy will morph into more functionally integrated organizational structures over time. Nonetheless, such changes will happen incrementally and, therefore, PaaS offerings such as OpenShift that can support a variety of operational philosophies have significant benefits.

ABOUT OPENSHIFT ENTERPRISE

OpenShift Enterprise is a commercial product, offered through a subscription like all Red Hat products, that lets IT organizations set up and manage their own PaaS. Red Hat curates, hardens, tests, and certifies the open source code from the OpenShift Origin project (under the Apache 2.0 license) to offer the reliable, supported, enterprise-ready OpenShift Enterprise product.

Architecturally, the only dependency for OpenShift Enterprise is Red Hat Enterprise Linux. OpenShift Enterprise can run on a baremetal system running Red Hat Enterprise Linux. It can run on virtual infrastructure provided by Red Hat Enterprise Virtualization or another virtualization platform. The Red Hat Enterprise Linux infrastructure can be managed and provisioned using tools such as Red Hat CloudForms or Red Hat Network Satellite. But OpenShift Enterprise is a PaaS orchestration layer that is independent of any specific tooling or infrastructure (other than Red Hat Enterprise Linux).

OpenShift Enterprise runs in one instance of Red Hat Enterprise Linux (the "broker")[5], which manages one or more other instances of Red Hat Enterprise Linux (the "nodes"). OpenShift Enterprise then provides the mechanisms for multiple applications to run within those nodes securely and with predictable performance, as well as providing monitoring and auto-scaling capabilities for those applications.

Application multi-tenancy, whereby multiple applications (each consisting of one or more "gears") can co-exist within each node, is provided through a variety of Red Hat Enterprise Linux features and managed by the OpenShift Enterprise broker. These features include:

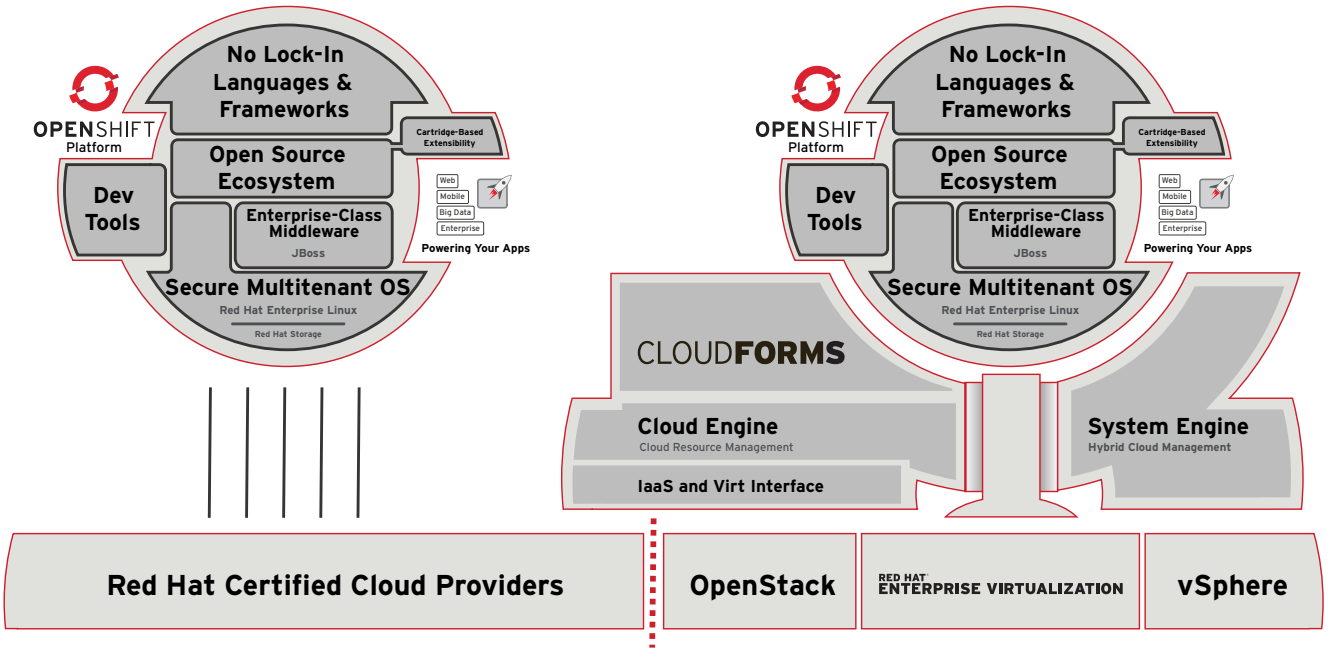
- **SELinux**, an implementation of a mandatory access control (MAC) mechanism in the Linux kernel, checks for allowed operations at a level beyond what standard discretionary access controls (DAC) can provide. It was initially created by the U.S. National Security Agency and can enforce rules on files and processes in a Linux system, and on their actions, based on defined policy. SELinux provides a high level of isolation between applications running within OpenShift Enterprise.
- **Control Groups** (cgroups) offer a powerful way to allocate processor, memory, and I/O resources among applications. They provide fine grained control of resource utilization in terms of memory consumption, I/O (storage and networking) utilization, and process priority, enabling the establishment of policies that provide quality-of-service guarantees.
- **Kernel namespaces** separate groups of processes so that they cannot see resources in other groups. From the perspective of an application running in OpenShift, for example, the application has access to a running Red Hat Enterprise Linux system even though, in reality, it may be one of many applications running within a single instance of Red Hat Enterprise Linux.

Collectively, these technologies implement an approach that is similar in certain respects to what are sometimes called containers, but with greater isolation and minimal resource overhead. The philosophy behind OpenShift was informed by its genesis as an online service, now called OpenShift Online, in which high security and high efficiency are both paramount.

Auto-scaling, which is built into the OpenShift Online offering, is also available in the on-premise OpenShift Enterprise offering. When you deploy an application through OpenShift, you can deploy it with scaling either enabled or disabled. If you deploy your application with scaling disabled, what you deploy is what you'll have, whether it's in one gear or several.

If you deploy with auto-scaling enabled, then the application can scale up and consume additional gears based on the usage of that application and its needs for more resources. For example, instead of, say, putting a MySQL database and JBoss Enterprise Application Platform into one gear, you would put MySQL into a separate gear, put JBoss Enterprise

5 The broker can alternatively run across multiple instances to avoid having a single point of failure.



Application Platform in its own gear, and then set up a software-based load balancer, which such as HAProxy, as a third gear. HAProxy would then detect the level of requests coming to the application. If it sees traffic exceeding a certain threshold, it will make a request to the broker to spin up an additional JBoss Enterprise Application Platform gear and configure it using JBoss clustering techniques.

In addition to a variety of programming languages and frameworks – including the latest Java EE 6 technologies – OpenShift also supports popular development tools. One such tool is Apache Maven, a software project management tool. Maven can manage a project's build, reporting, and documentation from a central model. Another is Jenkins, which is often used as a continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build – thereby increasing productivity. LDAP directory service and Kerberos network authentication protocol plug-ins let IT teams integrate OpenShift into their enterprise authentication systems so that when developers come to OpenShift, they can authenticate with their LDAP and/or Kerberos credentials.

OpenShift Enterprise's architectural approach has big benefits for developers and their organizations. It's built on a proven, secure multi-tenant operating system, Red Hat Enterprise Linux, and provides maximum flexibility in how that operating system is deployed and managed. It provides support for tools and languages that are both powerful and familiar. And it gives a great

deal of control to administrators, making the move from Red Hat Enterprise Linux administrator to PaaS administrator as seamless as possible with default policies and settings that will satisfy many requirements out of the box.

CONCLUSION

OpenShift Enterprise leverages important technology and operational learnings from Red Hat's hosted OpenShift Online service and delivers them in a form that organizations can control and consume in their own datacenters or other dedicated environments. Like other Red Hat subscription offerings, OpenShift also has an associated open source community project, OpenShift Origin, to which Red Hat contributes code through its philosophy of developing in upstream projects first whenever possible.

This is an important step because cloud computing — and even IT more broadly — is increasingly seen by CIOs, IT industry analysts, and others as being on a path to hybrid IT and infrastructure. One size doesn't fit all and neither does one deployment model. The most effective technologies are those that are flexible enough to adapt to a given organization's needs today while preserving portability of applications and tools to potentially different environments tomorrow.

This discussion is particularly germane to PaaS, which started out in many cases as essentially a point hosted offering for a narrowly defined development environment. OpenShift, on the other hand, has always been about choice of languages and frameworks and about application portability. With OpenShift Enterprise (and OpenShift Origin), there's now a choice of operational models that helps ensure organizations aren't locked into one provider or one way of doing things. And this choice will only become more important as PaaS increasingly represents the way that software development happens in cloud computing.

For additional information on OpenShift, visit **www.redhat.com/solutions/cloud-computing/paas**.

SALES AND INQUIRIES

NORTH AMERICA
1-888-REDHAT1
www.redhat.com

EUROPE, MIDDLE EAST AND AFRICA
00800 7334 2835
www.europe.redhat.com
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
www.apac.redhat.com
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
latammktg@redhat.com