White Paper

Red Hat OpenShift 3: A Modern Application Platform

By Stephen D. Hendrick
Principal Analyst, Application Development and Deployment Research

June 2015

This ESG White Paper was commissioned by Red Hat and is distributed under license from ESG.

Contents

The Shift to Modern Application Development and Deployment

There are a growing number of examples of how software and the cloud have the power to be disruptive technologies. Amazon and Alibaba in retail, Square and Apple Pay in finance, Uber and Lyft in transportation, and Netflix and Spotify in media are all examples of recent software-based services that have revolutionized how we do business. This potential for disruption across all vertical industries is driving a requirement for enterprises and their IT organizations to be more agile and responsive to changing business environments.

Cloud computing is also enabling a more distributed approach to application architecture. Applications are designed around smaller independent components called microservices. The decomposition of applications into smaller components has distinct advantages of developing applications faster and delivering them in a more resilient way—both of which are better aligned with today's business needs. However, this design adds complexity in the application layer for IT operations. For IT operations to support this new application architecture, the underlying middleware, runtimes, and other software services needed for application lifecycle management should be highly automated and have their complexity abstracted away. Standardization and platform abstraction not only make operations more efficient in handling developer needs, but they also help streamline the deployment pipeline, thereby accelerating the delivery of microservice-based applications. The shift to modern application development and the need for rapid and continuous deployment make a strong DevOps-enabling platform a key component in the IT arsenal.

The large-scale automation used underneath this platform abstraction is necessary to ensure that deployments are handled efficiently at potentially massive scale and volatility. Automation is also a prerequisite for immutability where components and services are never changed—just replaced. Support for immutable infrastructure enables a modern IT environment to simplify maintenance and keep pace with the dynamic demands of cloud applications, where instances are created and destroyed instead of being patched and rebooted. This approach to application deployment yields highly agile operations and is consistent with microservices application development.

Linux containers are key building blocks for this modern platform. Today, the most popular Linux container format to build a robust DevOps environment is based on the Docker project. The Docker project provides an abstraction layer that enables multiple containerized images to run on a single operating system instance. The utility of the Docker packaging model is closely tied to its ecosystem, which includes over 1,000 contributors, support for the leading vendors, and thousands of images on the Docker Hub. Containerization is having a profound impact on IT for two primary reasons. First, containers provide a simpler and more efficient architectural model for hosting an image by providing the benefits of virtualization without the need for a hypervisor and its attendant guest operating system. In other words, multitenancy is managed within the operating system itself, enabling more efficient use of underlying infrastructure. Second, the increasingly pervasive use of the Docker image format by leading software vendors eliminates much of the lock-in that exists today by enabling choice of underlying platforms for customers.

While containers are effective at the heavy lifting when instantiating an image or application component on a single hosting environment, most modern applications are comprised of many components that span multiple hosts that in turn could be located in various geographic regions. This is where Kubernetes, a Google-initiated project, comes into play as a means to address container orchestration and management. Kubernetes handles container deployments and orchestration as a cluster manager by using a declarative model that enables the user to define her application needs. The policies and instructions defined in this model are carried out by Kubernetes, which instantiates, restarts, redistributes, and replicates containers as necessary. The benefit of Kubernetes is that it provides a standardized and repeatable approach to container orchestration and microservices management at scale.

Finally, a modern platform needs a developer interface that facilitates usage of only the pieces needed to quickly prototype, test, and deploy an application, in many cases directly to production from source. Having immediate access to a wide array of development tools and frameworks allows the developer to get started with little to no friction, and concentrate on the task of innovation. At the same time, this choice still needs to meet the control and security requirements of the IT operations team. In previous years, the concepts of choice and control were counter

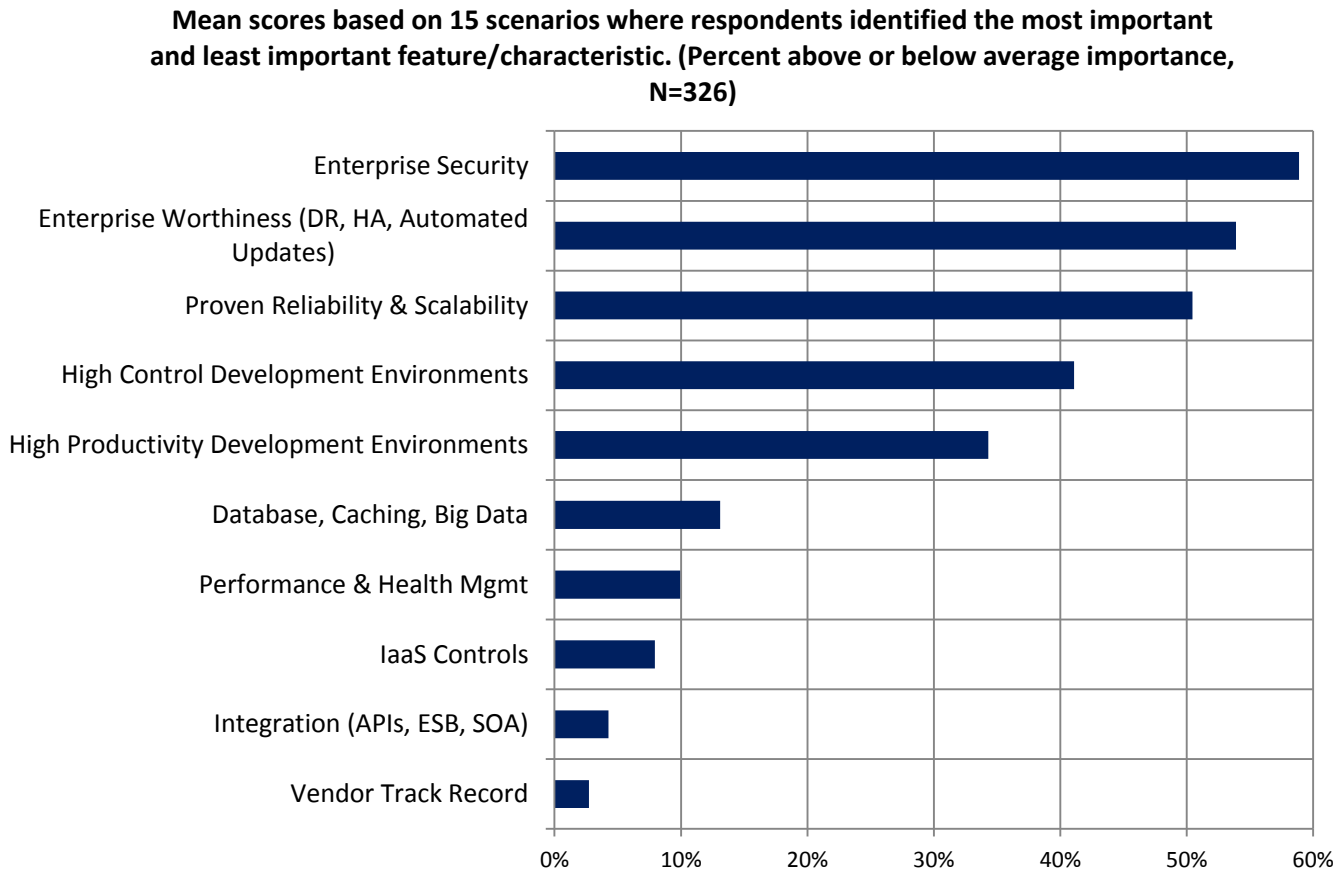
to each other, but thanks to the modern application platform, the ideas and people involved can operate in harmony.

OpenShift 3: Enabling DevOps and Driving Innovation

Red Hat's OpenShift 3 represents the next stage in the evolution of OpenShift and is a prime example of the modern application platform described in the previous section. Despite the pioneering work that Red Hat had invested in its own approach to containerization, the benefits from standardizing on the Docker image format were viewed as more compelling and beneficial for the ecosystem in the long term. Although Red Hat, its partners, and its customers had amassed hundreds of cartridges that would run on OpenShift, by adopting the Docker format as its container strategy, OpenShift allowed users to have access to thousands of images available on the Docker Hub, in addition to any custom images users may hold privately. Additionally, in order to meet the needs of the enterprise, Red Hat and its partners are actively working on image certification as a way to verify image content.

Today's focus on microservices, abstraction, automation, containers, orchestration, and ecosystems is echoed by ESG's platform-as-a-service (PaaS) research. Figure 1 shows that capabilities related to application deployment and development rank as the most important features that developers and IT managers alike look for in a platform. Enterprise security, enterprise worthiness, proven reliability and scalability, and development environment choice were identified as the most important PaaS features by a considerable margin.[1] These features are widely considered key technologies for enabling DevOps and are characteristic of a modern application platform.

Figure 1. Platform-as-a-service (PaaS) Top Ten Features in Order of Importance



Source: Enterprise Strategy Group, 2015.

[1] Source: ESG Research Report, *Platform-as-a-service Feature Preference Study*, May 2015.

Under the Hood of OpenShift 3

Looking underneath the platform abstraction, one can understand how OpenShift 3 is architected for enterprise IT, taking advantage of emerging industry standards and Red Hat's enterprise grade operating system. The major features provided by OpenShift 3 include:

- **Trusted and Optimized Container Host:** The foundation for OpenShift 3 is Red Hat Enterprise Linux (RHEL) 7, bringing enterprise grade security and reliability via security-enhanced Linux (SELinux) in the container host. Moreover, OpenShift 3 can also leverage RHEL Atomic Host as this secure base, which provides the additional advantage of being a minimal footprint of RHEL specifically optimized to run and manage containers based on the Docker project. Combined with a technology called OSTree, this further simplifies host operating system provisioning and maintenance while limiting server downtime.
- **Standard Container Format:** OpenShift 3 utilizing the Docker packaging model represents a standardized approach for application packaging and deployment of Linux containers. Standardized containers enable the platform to provide higher application density and provisioning speed while also enabling portability across physical, virtual, and cloud (both public and private) environments.
- **Web-scale Orchestration:** By utilizing the Kubernetes project, OpenShift 3 provides cluster management for the microservices environment that can span multiple logical and physical host locations. Additionally, with native health management, applications are highly resilient.
- **Container Networking:** OpenShift 3 provides both internal and external container networking services along with the ability to plug in third-party SDN solutions. This networking layer can be integrated with existing DNS, routing, and load balancing capabilities that customers may currently have available.
- **Container Storage:** Not every application needs immutability, so for those instances OpenShift 3 enables stateful services to run inside containers by providing shared storage volumes for containers that is extensible with plugins for NFS, iSCSI, Ceph, and other popular platform storage solutions, including AWS and GCE.
- **Container Registry:** OpenShift 3 allows users to deploy their own private container register to allow authentication and access control to images based on their own policies. This registry can pull images from multiple sources, including Red Hat CDN, Red Hat Satellite, Docker Hub, and most other third-party enterprise registries.
- **Simplified Administration:** Management capabilities allow operations to easily manage resource quotas, project creation, authentication and authorization, and installation of the environment, including third-party infrastructure service integrations where desired.
- **Developer Interface:** Developers have a choice of web console, CLI, and IDE integration to manage their applications, including built-in collaboration with fine-grained access control.
- **Source to Image:** In addition to the capability to run existing container images, another innovation in the OpenShift 3 feature set is the option to use the Source-To-Image (STI) build tool. The STI tool is used for building container images by injecting user source code into the image and building a new image for rapid deployment. There is a wide choice of supported runtimes and frameworks for building these, including PHP, Python, Ruby, and Java.
- **Ecosystem of Application Services:** Beyond the Red Hat-provided and -supported runtimes and frameworks and the ability to run containers from multiple sources, OpenShift 3 also allows users to take advantage of fully certified third-party containers available from the Red Hat Container Certification program. By using certified containers from partners, customers get peace of mind that they are running trusted and secure containers for mission-critical applications.
- **Community and Collaboration:** OpenShift Commons is an online experience with the charter to build connections across OpenShift communities, projects, and stakeholders. It goes beyond developer

contributions and provides an environment to foster sharing of best practices and peer-to-peer communication to drive success and adoption of OpenShift.

The Bigger Truth

Modern application platforms are designed to address a wide variety of IT needs ranging from the development and deployment of traditional applications to newer microservice applications leveraging highly scalable and immutable infrastructure. Enterprises should consider transitioning to a modern application platform because it offers developers design and development flexibility, IT managers secure and automated deployment capabilities, and IT environments an agile and highly abstracted platform from which to deliver increasing IT value to the enterprise.

Based on ESG's PaaS research, the combination of a carefully defined set of application development, deployment, and operational capabilities is essential to the success of IT meeting growing business demands. OpenShift 3 allows business and IT organizations the right combination designed to build, deploy, and manage applications more efficiently and at scale. The choice of development environments and extensive middleware services offered by OpenShift 3 to support a wide variety of application architectures, including microservices, provides developers the ability to optimize application efforts by selecting the right language and services to meet the functionality needed. With access to thousands of images through OpenShift 3's standardized support for containers, and by leveraging the orchestration capabilities included in OpenShift 3, your ability to simplify operations requirements while providing significantly enhanced levels of service flexibility and application portability lets your team get applications to market quickly. In ESG's opinion, the evaluation of OpenShift 3 is essential for enterprises looking to modernize application processes to meet growing demands, to stay ahead of the ever-changing market landscape, and to leverage the advantage of emerging industry standards.

Enterprise Strategy Group | **Getting to the bigger truth.**

20 Asylum Street | Milford, MA 01757 | Tel: 508.482.0188 Fax: 508.482.0218 | www.esg-global.com