White Paper

The Rise of the Enterprise Container Platform

Sponsored by: Docker

Gary Chen
June 2018

IDC OPINION

Digital transformation is causing businesses to move faster in the digital realm, driving a wave of technology change. Companies are migrating to the public cloud to take advantage of on-demand resources and implementing DevOps and microservices architectures to increase application development speed.

Containers have rapidly evolved in the past few years into a perfect core technology to enable all of these technology changes. Lightweight, agile, and standardized, containers are the perfect vessel to encapsulate microservices, enable portability between on-premises and cloud, and push through continuous integration/continuous deployment (CI/CD) pipelines. Containers are also being used for existing applications, extending many of the benefits to traditional enterprise applications while preparing them for future modernization. While the next-generation applications are certainly cloud native and microservices based, enterprises have huge portfolios of applications that can't be rebuilt on a whim and the industry will be in transition for a decade or more. Containerizing these traditional applications provides some immediate benefits around resource consolidation and simpler patching and maintenance until further application modernization investments can be made.

While containers are a core enabling technology, the industry is witnessing the buildout of full container platforms as containers redefine the datacenter stack. Companies that want to operationalize at scale across the enterprise, create a common operating model across DevOps, and comply with organizational security policies are gravitating toward container platforms for security, management, and more governance capabilities over their container deployments.

SITUATION OVERVIEW

Containers are not a new concept, but until the recent introduction of Docker, they were an esoteric and niche technology primarily used by operating system (OS) admins to isolate workloads. Docker found success by taking a set of core kernel functions and making them highly relevant and user-friendly:

- Developed easy-to-use, API-enabled, and developer-friendly tooling
- Open sourced what became the de facto container format standard and formally standardized it through the Open Container Initiative (OCI)
- Focused on solving modern developer problems
- Developed innovative packaging technology to improve developer workflows and improve consistency in software packaging

- Addressed the application dependency problem, which simplified support/test matrixes and made applications more portable
- Leveraged the lightweight nature of containers for improved infrastructure and operations of modern and scale-out apps

The introduction of Docker also coincided perfectly with the digital transformation sweeping businesses across the globe, spurring companies of all sizes and industries to build more digital assets and do it faster and faster. Digital transformation, cloud migration, and web scale began to push the IT industry toward fast and agile development, DevOps, cloud-native microservices-based application architectures, and multicloud/hybrid cloud deployments. Docker was portable, efficient, and lightweight, which lent itself well to:

- Easy portability between all types of infrastructures including public cloud, private cloud, virtualized servers, and bare metal servers
- Efficient packaging that made containers a perfect choice to encapsulate new microservices and also ideal for packaging code to be pushed down CI/CD pipelines that many were building

Containers Are Production Ready

Containers are already being used in production and for mission-critical apps, which is a very fast maturation compared with other technologies. In fact, 85% of container adopters are using containers for production apps, with 76% using containers for mission-critical applications (see Figure 1).

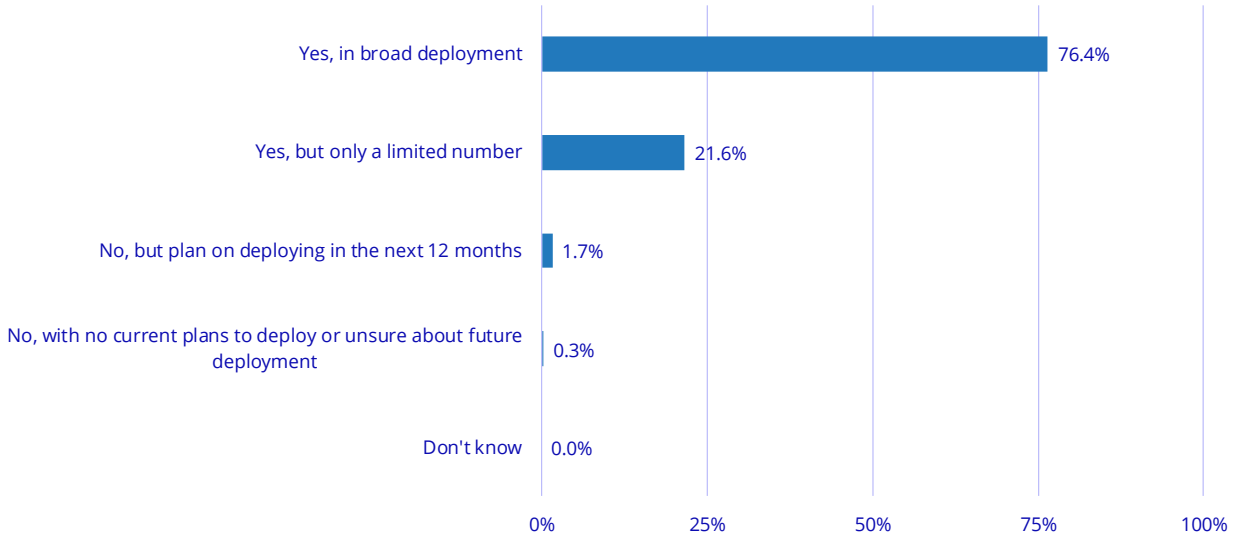
One reason that container execution is so stable is that the core Linux kernel functions for containerization have been in development for a while and are quite mature. This was helped by many web companies that pioneered and matured the core technology within the container engine for use with their own applications and services.

However, the overall percentage of applications running in containers in most enterprises is still very small. While enterprises have been successful and quick in putting containerized applications into production, running containers at larger scale will require more robust platforms to accommodate the full range of enterprise applications and requirements. Containerization is redefining the datacenter stack as server virtualization has, and enterprises today have built out only a small fraction of that stack.

FIGURE 1

Containers Are in Production and Being Used for Mission-Critical Apps

Q. Are any of your "tier 1" mission- or business-critical applications running in containers today?



n = 301

Source: IDC's *Container Infrastructure Software Survey,* January 2018

Consistency Across Diverse Environments

With the rise of public cloud, deployment options are more diverse than ever, and consequently users typically have multiple environments for their applications. Even on-premises is a combination of different generations of technology, a mashup of traditional bare metal, virtualized servers, and private cloud systems. The goal for organizations today is to stitch these disparate environments into a cohesive and seamless hybrid cloud, with standardized containers in a container platform environment as a starting point on a path to accomplish that goal.

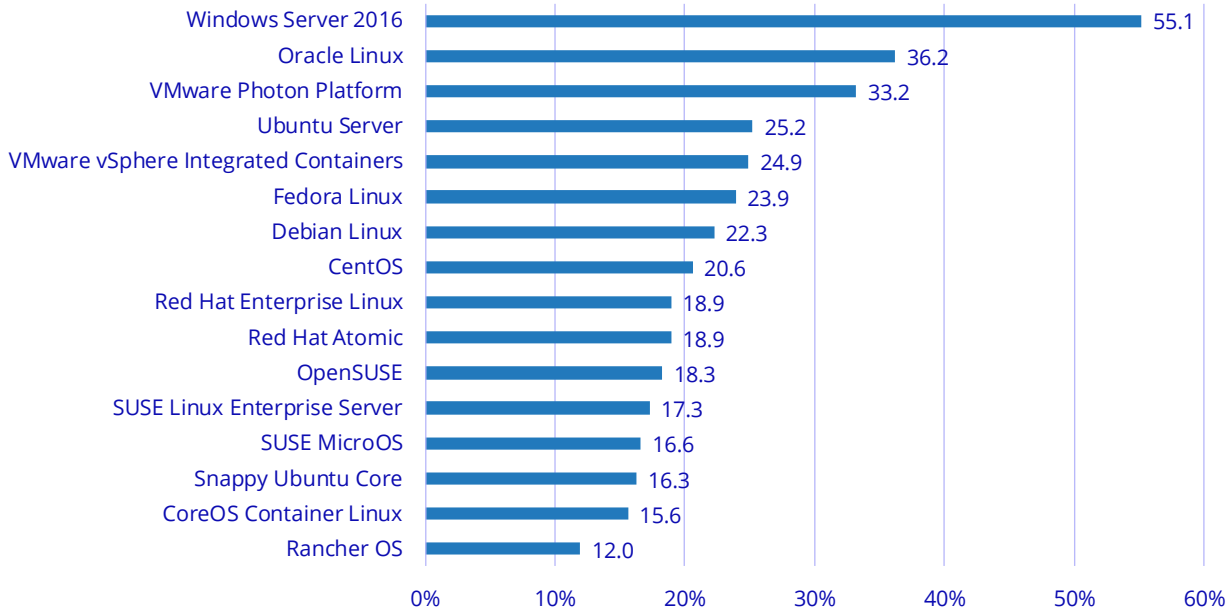
Figure 2 illustrates the diverse and hybrid environments that containers are deployed on:

- Multiple operating systems: 88% use containers with multiple OSs (4.1 OSs on average), and 51% of container deployers have both Windows and Linux containers.
- Multiple public cloud: 83% of container deployers use containers on multiple public clouds (3.7 clouds on average).
- On- and off-premises: Container deployers report that 55% of containers run on-premises, while 45% run in the public cloud.

FIGURE 2

Container Host OS

Q. Of all the container hosts you manage, whether on-premises or in the cloud, please indicate the following operating systems that you use.



n = 301

Note: A container host is the shared OS that hosts multiple user/application containers.

Source: IDC's *Container Infrastructure Software Survey,* January 2018

Container Workloads Span Existing and Greenfield Applications

Unlike web companies, most enterprises have limited greenfield opportunities. Technologies that can't be applied to existing applications have limited success in the enterprise, and containers have developed into a tool that can address both new and existing applications. In fact, 54% of containers run existing applications today versus 46% running new applications (see Figure 3).

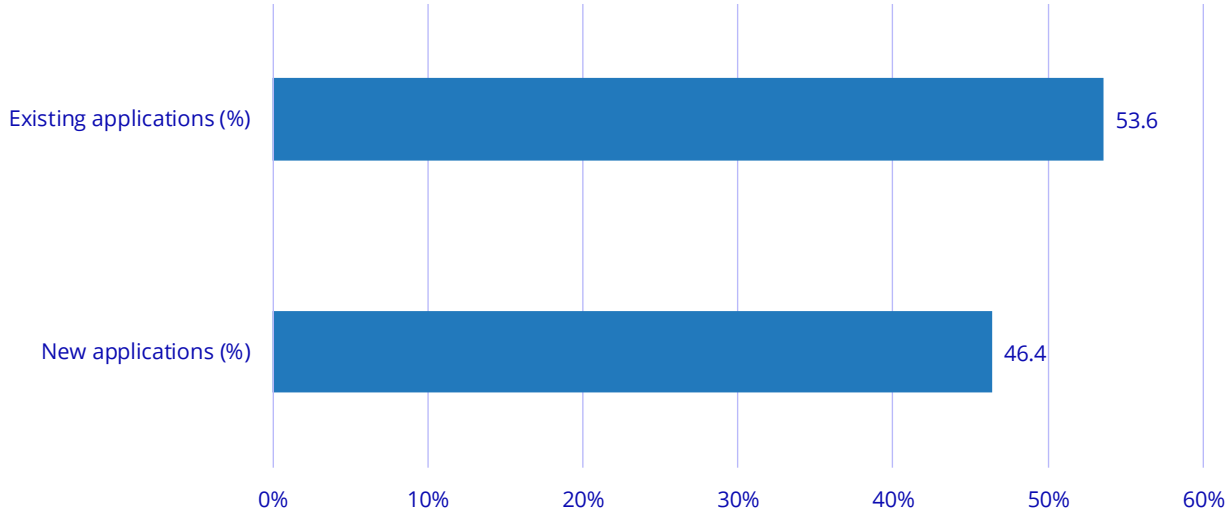
For those existing applications, there is a solid path to containerization; 75% of existing applications containerized today required no to moderate modifications to containerize, with only 25% requiring significant modifications. Many of these applications (58%) will also continue to be refactored and modernized over time. Initial containerization of existing applications is the first step for many when refactoring applications, with containers serving as an enabling technology for application modernization.

As containers become the de facto format for new applications and as organizations see benefits in containerizing existing apps, organizations need a platform solution that can support these different projects, which have very different needs and architectures.

FIGURE 3

Existing Versus New Applications

Q. *What percentage of your containerized applications are existing applications migrated to a container (from a VM or bare metal) versus a new application that started in a container from day 1?*



n = 301

Source: IDC's *Container Infrastructure Software Survey,* January 2018

FUTURE OUTLOOK

The Rise of the Container Platform

Containers have sparked a wave of innovation in IT. This new core compute primitive is having ripple effects on the entire stack, and users are embarking on a journey that will be multilayered. As the industry and deployments progress, maturation and development move up the stack. Just a few years ago, much of the focus was on developing the core container engine. Most users had container hosts running these engines and were using manual or simple tools to manage them. Today many users are looking at orchestration capabilities through tooling such as Docker Swarm and Kubernetes. Containers have progressed from being just a technology and point solution to becoming a core part of a company's IT strategy.

However, operationalizing at scale across the enterprise requires many more layers to be addressed to have a complete container platform. A full set of management functions is needed to operate containers at scale within enterprises and comply with company security policies. Some of these areas are:

- Integration and support of open source components
- Orchestration
- Access control

- Integration with and extensions to compute, storage, and networking via APIs
- Security

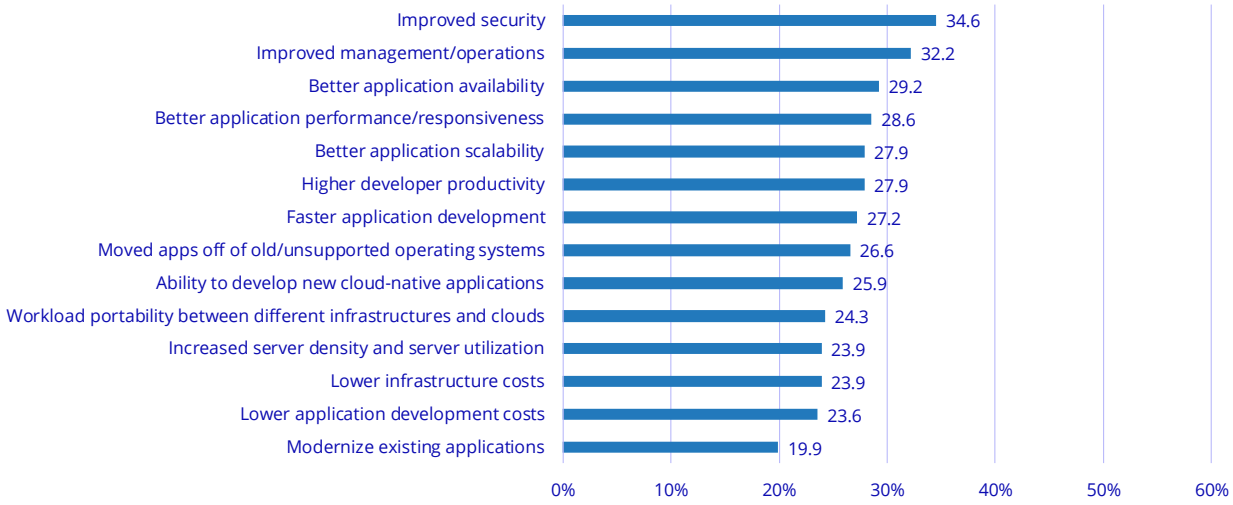
Container Security — A Top Driver for Containerization

Container platforms bring a more coordinated approach to security. Security is a top priority for enterprises and doubly so for containers. IDC's *Container Infrastructure Software Survey,* January 2018, shows that security is a top benefit and driver for containers, allowing customers the opportunity to build more secure applications and infrastructure (see Figure 4). It also allows customers to patch faster, improving response time to new threats. Security is also the top container challenge in that it is a new technology and customers have a learning curve to figure out how to secure this new layer.

FIGURE 4

Benefits of Containerization

Q. What are the top benefits your organization realized from containers?



n = 301

Source: IDC's *Container Infrastructure Software Survey,* January 2018

Container Platform Considerations

As with server virtualization that preceded it, the container wave started with a small, simple idea that operated at the lower layers of the stack. However, it was such a transformative idea that it changed many of the assumptions the rest of the stack was built on. Today we are seeing that transformative ripple effect as a full container platform is being built out to address the security, management, governance, and operational efficiency requirements enterprise customers have when adopting containers as the basis for the next generation of compute.

A container platform not only has multiple modules and end-to-end holistic functionality but also brings increasing abstraction from the underlying infrastructure. This is key as customers search for consistency across hybrid cloud and multicloud environments. A container platform should allow users to develop and manage applications the same way regardless of underlying cloud, virtualization, or OS. This brings

portability and guards against lock-in. It should also address multiple types of use cases from microservices to CI/CD to modernization of legacy applications.

A container platform also needs to integrate well with enterprise IT systems and processes while providing additional security capabilities to protect containerized applications. Because containers rely heavily on the source image files, securing the container should also include securing the image. A container platform should include solutions to manage and secure the entire container life cycle from development to production, including image management.

Finally, to drive growth and adoption inside the enterprise environment, a container platform needs to be simple to install, configure, and operate at scale. A platform that can quickly add value back to the company while serving the needs of developers and IT will be more successful in the long run. The ecosystem around a platform is also important. No platform can contain everything an enterprise may need, and the ecosystem market of add-on drivers, plug-ins, and complementary software and hardware is key to the value of a platform.

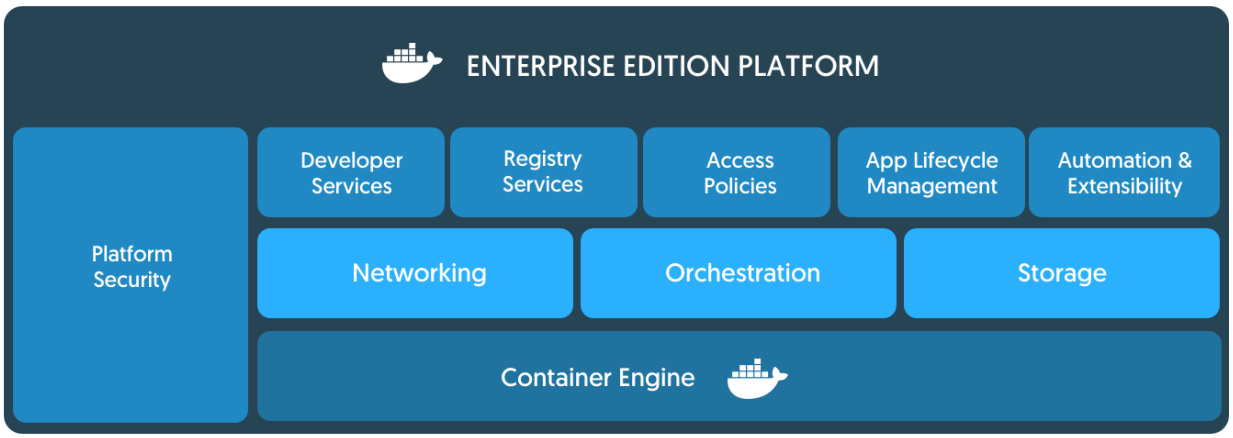
DOCKER PROFILE

Docker, the company that sparked the modern container trend with the Docker Engine, offers a full, enterprise-ready container platform called Docker Enterprise Edition. Docker Enterprise Edition is Docker's commercial offering that provides choice, agile operations, and integrated security. Docker has over 500 commercial customers using Docker Enterprise Edition today. Keeping with the Docker tradition of making complicated technology easier to consume, the Docker platform abstracts the complexity, making orchestration, security, and operations integrated components of the Dev to Ops workflow and easier to use.

According to Docker, there have been billions of downloads of its container technology, and it has been tested by millions of users over millions of nodes. Figure 5 depicts the Docker Enterprise Edition platform.

FIGURE 5

Docker Enterprise Edition



Source: Docker, 2018

Freedom of Choice

Docker Enterprise Edition includes a commercially supported version of the Docker Engine and has broad OS support. It runs on both Windows and Linux and supports most major Linux distributions such as CentOS, RHEL, Ubuntu, SUSE, and Oracle Linux.

Orchestration is one component of a container platform, and Docker Enterprise Edition supports both Docker Swarm and Kubernetes, enabling users to choose based on use case and business needs. Customers can operate both in the same cluster and can easily move applications between orchestrators if needs change.

Docker Enterprise Edition is a portable container platform that is deployable on nearly any infrastructure, whether bare metal, virtualized server, private cloud, or (multiple) public clouds. With Docker Certified Infrastructure and Certified Plugins for storage and networking, users get a validated and fully supported deployment that integrates with their chosen infrastructure. This enables organizations to reduce costs and increase operational efficiency by standardizing the way to build, manage, and secure applications across diverse infrastructures.

Security

Integrated security in Docker Enterprise Edition incorporates additional security at multiple points in the application delivery life cycle to allow customers to build a secure software supply chain. With the default isolation and security parameters, plus strong system-level encryption included in Docker Enterprise Edition, organizations can architect a customizable layer of protection for their existing application stack. Docker's secure, integrated supply chain provides protection against attacks, verifies content publishers, and scans against vulnerabilities and is delivered in a simplified workflow.

This ensures that applications receive greater protection and improved governance with centrally managed content and policy-driven automation. Some of the security features within Docker Enterprise Edition are:

- Role-based access control
- Container image scanning and signing
- Policy-based image promotions
- Secrets management
- Out-of-the-box container runtime defaults including seccomp
- Encrypted communications with mutual TLS

Agile, Efficient Operations

Docker Enterprise Edition enables organizations to reduce costs and increase operational efficiency by standardizing the way to build, manage, and secure applications across diverse infrastructures including multiple clouds. Docker Enterprise Edition was designed to scale and to be easy to operate and deploy. Organizations can start small and scale up, start with legacy or new applications, use a wide range of operating systems, and deploy on a multitude of infrastructures or clouds.

Integrated Solutions

Docker Enterprise Edition is backed by enterprise-level support, services, and training to help customers master the Docker container platform and create their transition to modern applications. The "Modernize Traditional Applications" program, or MTA, is designed to jump-start a customer's adoption of containers by providing the fastest path to cutting operational costs and to gaining agility by beginning containerization with existing applications. Docker Professional Services also help companies create the governance and application pipelines necessary to continue modernization for these applications and pave the way for microservices and continued innovation.

CUSTOMER STORIES

Finnish Railways

- Finnish government-owned company, but independently run
- Responsible for passenger traffic logistics and for tracking build/maintenance of the Finnish railway system
- Revenue: 1.2 billion EUR
- 8,000 employees, 70 IT staff
- Container deployment: About 800 container instances in total (300 are for production)

Over the past few years, Finnish Railways has embarked on a modernization initiative, seeking to migrate to the public cloud, implement DevOps, and revamp its application development infrastructure. The decision was made to use AWS for the public cloud, host everything in containers, and begin to build apps using a microservices architecture.

The primary application to be modernized was the ticket reservation system, which was running on a mainframe system. The decision was driven by the architecture team, and the goal was to rebuild the ticket reservation system from scratch with a microservices architecture and host it in Docker containers. Finnish Railways chose to use the public cloud because it wanted to pursue infrastructure as code, the backbone of the company's modernization strategy. It was more flexible than the company's on-premises datacenters, and Finnish Railways had no restrictions or regulations that mandated on-premises only. The reservation system renewal project was funded by business, and the Infrastructure as a Container (IaaC) project was funded by IT.

Docker was already the standard container type, but Finnish Railways had to evaluate and choose an orchestration solution. The company looked at AWS ECS, but it abstracted away much of the infrastructure underneath. While this was acceptable for pure microservices applications, it didn't work for some legacy applications that Finnish Railways wanted to migrate over. The team also evaluated Mesosphere, but Finnish Railways ultimately decided it didn't need the huge clusters that Mesosphere was known for. Kubernetes was on the radar as well, but it was early days for the project and Finnish Railways couldn't find a good enterprise distribution of it. Finnish Railways eventually chose to use Docker Enterprise Edition with integrated Swarm orchestration.

Finnish Railways found that Docker Swarm, Docker Enterprise Edition's integrated orchestration capability, was much simpler to set up, with apps needing just one compose file to set everything up. Kubernetes has grown steadily since then, and Docker Enterprise Edition now also supports Kubernetes alongside Swarm. For newer applications, Finnish Railways is considering using Kubernetes, but its existing applications will all stay on Swarm, and the company doesn't see any benefit in migrating.

Security is a top issue at nearly every organization, and Finnish Railways leverages several Docker Enterprise Edition security features in its deployment. The company has recently added image scanning to protect the integrity of its container code. Role-based access control is also a very key feature, and Finnish Railways depends on that a great deal. The company also uses the Docker Trusted Registry and does not allow the use of public registries.

Beyond completely rebuilding the reservation system, Finnish Railways also migrated many existing applications into Docker. They were primarily Java apps, based on WebLogic. Finnish Railways had to make changes to containerize these applications; in some instances, heavy changes were required. Many were using legacy methods such as file sharing and sticky sessions that were not supported in containers. The company also implemented persistent storage to help with compatibility, using AWS EFS through a Docker Volume Plugin. Finnish Railways made only the minimum amount of changes to containerize these applications. The company does not intend to further refactor the applications into microservices because the level of effort would be too great. Instead, it will let the applications age out and then rebuild them at that point from scratch to be cloud native.

Even with a traditional architecture, Finnish Railways still saw several benefits from containerizing applications. With containers, the company could increase the density and utilization of its Amazon compute instances, saving on its cloud bill. Finnish Railways is experiencing an average cost savings of 50% across its application portfolio with Docker Enterprise Edition. Deployment also became much faster. Finnish Railways has one vendor that manages its infrastructure and another vendor that develops the application. The application team would give the infrastructure vendor instructions on how to deploy, but if something went wrong, everything had to stop and start over again. Deployment time took a day, sometimes several days, but with containers, it now takes only minutes. Furthermore, Finnish Railways has been able to achieve standardization on a common platform and architecture that can be leveraged across all applications and by all of Finnish Railways' vendors.

For the newer microservices applications, the biggest change was transitioning to DevOps and agile development. This transition was significant, not just technically but also culturally, requiring many operational and mindset adjustments throughout the organization. However, that effort is starting to bear fruit now.

Through many lessons learned during its modernization, cloud, and container journey, Finnish Railways offers the following advice to peers starting out with containers:

- Start small and simple. Try building simple containers first and gradually add new features to the system as you gain knowledge and experience.
- Keep the big picture in mind, and don't let individual projects impose individual requirements. Develop general practices that can scale across all infrastructure and applications, not just for one project's needs.
- Create thorough documentation, even starting with very basic step-by-step instructions. Companies that deal with multiple vendors, contractors, and even a variety of internal staff will need to make sure everyone is trained and following the same procedure. This documentation will be very useful and used many times over and over.

Franklin American Mortgage Company

- United States–based mortgage banking firm licensed to provide residential mortgages across the nation
- Container deployment: About 1,000 container instances

Franklin American sees itself as a technology company that happens to do mortgages and as such has invested in its own proprietary software applications, making the software development process crucial to the business. Two years ago, the company created an innovation team to foster and advance innovation throughout the company by turning ideas into software products, revenue generators, or growth catalysts. The team was originally composed of about 15 staff members, consisting of developers, architects, product owners, test engineers, and DevOps engineers. The initial focus of the team was to reimagine a key component within the business and deliver it using a modern microservices architecture and agile development processes.

Containers were determined to be key to enabling a modern environment, and Franklin American began evaluating vendors and products. The company eventually chose Docker Enterprise Edition for a couple of key reasons. First, it was easy to maintain given the simplicity of Docker Swarm and the management tools for it. Second, it allowed isolation within a single cluster, avoiding the building and maintenance of multiple clusters. A combination of Docker resource collections, RBAC, Active Directory integration and node affinity made it possible to run dev, test, QA, UAT, staging, and production on a single cluster while isolating these environments from each other.

Another crucial set of features is part of Docker Trusted Registry, which is bundled with Docker Content Trust and image scanning, which respectively provide image provenance to authenticate where the image is from and that it has not been tampered with and validation that the image is free of critical vulnerabilities. These features were all included with Docker Enterprise Edition and are API enabled. The closest alternative would have entailed using a combination of products lacking integration and would have been much more expensive. Docker Enterprise Edition had the best combination of the features Franklin American needed — at the right price point.

Franklin American is also experimenting with Kubernetes and has upgraded its test cluster to Docker Enterprise Edition 2.0, which includes both Swarm and Kubernetes orchestrators. Franklin American is considering adding Kubernetes to integrate some parts of its infrastructure that have Kubernetes drivers. The company likes Docker Enterprise Edition 2.0 because it provides the option to use either orchestrator while managing both through a single management plane and uniform approach to operations. Franklin American doesn't have to grow its DevOps team significantly in order to manage both technologies.

Today, Franklin American runs one primary cluster for production as well as various other test zones. It has a separate test cluster primarily for testing Docker Enterprise Edition control plane updates. The main cluster runs approximately 1,000 container instances across 32 nodes. The nodes are virtualized and running on top of VMware vSphere. The container instances host about 300 logical services, most of which make up the reimagined component of the software.

By rebuilding and containerizing the software, Franklin American was able to dramatically increase the number of deployments. From inception, it quickly ramped up to do 120 deployments per month and increased that to 200 deployments per day today over the span of a year. Those deployments are spread across various environments, but it still equates to about 30 deployments into production per day. These results were achieved by automating every part of testing possible and automating deployments into every environment.

Another key to success was increasing developer productivity by being able to give developers individual sandboxes to test their changes. Previously developers shared one environment, and it was hard to test changes in isolation. Docker enabled creating these environments through its Docker Enterprise Edition management plane API. It was able to tie directly into Active Directory (AD) and associate AD groups with Docker user groups and give different groups their own access and resources. This environment helped with security audits by having one source of identity and access management. The environment also enabled everyone to have access to Docker and, more importantly, restrict what they have access to but not restrict what they could do within their own zones.

In terms of operational benefits, Docker Enterprise Edition enabled a much more secure and agile infrastructure. Upgrading a Docker cluster is nondisruptive because the Swarm schedule can do rolling upgrades one node at a time. If a security patch is needed, Franklin American has to update only the core base images. Downstream app images are automatically rebuilt, tested, and pushed through the stages until they are promoted to production.

For the future, Franklin American is experimenting with cloud bursting. Today all its container images run on-premises, but the company is working on the possibility of spinning up worker nodes in public clouds and adding them to the cluster. Franklin American is also standing up another cluster in a second datacenter location for DR.

Franklin American offers the following advice to peers starting out with containers based on some challenges it encountered:

- Do good capacity planning up front. Initially Franklin American had overprovisioned for its applications, causing the cluster to be underutilized.
- Bring in the experts. Companies that are migrating existing applications should collaborate with the experts who work with those applications to gain an understanding of how they do deployments and then build the containerization process from there.
- Fail fearlessly, and fail fast. Docker adopts, promotes, and enables that key mindset. It makes it easy to try something new and experiment without a lot of investment to do it. This allows businesses to quickly find out what works and what doesn't and iterate quickly. In fewer than 18 months, Franklin American was able to go from concept to about 1,000 containers.

CHALLENGES/OPPORTUNITIES

Challenges

- Container technology has been adopted as an initial first step for managing diverse applications and infrastructures, but as it becomes more universally deployed, it is clear that a container management platform is needed to handle the diverse enterprise requirements around security, management, governance, and operational efficiency.
- Containers involve a change not only in technology but also in how organizations approach application development and infrastructure management (i.e., skills, attitudes, and processes). In many ways, this is the real learning curve, and within the enterprise, there are very entrenched elements that are difficult and slow to change. Skills around cloud-native apps are also at a premium today in the enterprise, which can also hinder adoption.
- Vendors providing technology solutions is just one part of the puzzle. With enterprises, which are known for maintaining multiple generations of technology, the trick is always how to make the transition in a smooth manner. As applications transition to cloud-native apps, enterprises need guidance on how to slowly do that over time and on how to bridge between the old and the new during the transition.

Opportunities

- Docker Enterprise Edition has the opportunity to be platform neutral and offer freedom of choice because it does not have an established business in operating systems, hypervisors, or public cloud platforms that would bias it. As enterprise IT environments become diverse and more spread out, many will look for uniformity to prevent the spread of silos and lock-in.
- Docker created the modern container, which comes with a significant amount of industry recognition. Docker has the opportunity to be a visionary leader and define the road map for container platforms.

CONCLUSION

The container revolution is under way today as companies seek to move faster and build digital assets to support growth. IDC forecasts that there will be over 1.8 billion enterprise container instances deployed by 2021. Container platforms will be key to managing the ever-expanding diversity of IT environments (multiple operating systems, hypervisors, private clouds, and public clouds). While containers are the perfect vehicle to underpin cloud-native, microservices applications, containers are also well suited to many existing applications. These applications can reap immediate benefits from being modernized and prepare for possible refactoring over time. To successfully leverage containers in these diverse scenarios and do it at scale, enterprises will need a robust, integrated, and holistic container platform.

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1,100 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For 50 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-community.com
www.idc.com