PATHFINDER REPORT

451 Research | Advisory

Enterprise DevOps and Containers: Exclude Data and Storage at Your Own Risk

OCTOBER 2016

COMMISSIONED BY

Hewlett Packard Enterprise

About this paper

A Pathfinder paper navigates decision-makers through the issues surrounding a specific technology or business case, explores the business value of adoption, and recommends the range of considerations and concrete next steps in the decision-making process.

About 451 Research

451 Research is a preeminent information technology research and advisory company. With a core focus on technology innovation and market disruption, we provide essential insight for leaders of the digital economy. More than 100 analysts and consultants deliver that insight via syndicated research, advisory services and live events to over 1,000 client organizations in North America, Europe and around the world. Founded in 2000 and headquartered in New York, 451 Research is a division of The 451 Group.

NEW YORK
20 West 37th Street
New York, NY 10018
+1 212 505 3030

SAN FRANCISCO
140 Geary Street
San Francisco, CA 94108
+1 415 989 1555

LONDON
Paxton House
30, Artillery Lane
London, E1 7LS, UK
+44 (0) 207 426 1050

BOSTON
One Liberty Square
Boston, MA 02109
+1 617 598 7200

I. Enterprise IT Sea Change and Response

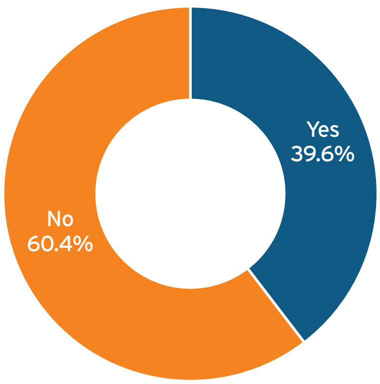
Many concurrent trends are driving dramatic changes in today's enterprise IT. Among the most disruptive trends are DevOps and application containers such as Docker. While enterprise organizations are rapidly adopting DevOps and containers, they often fail to devote adequate attention to data persistence and storage requirements. Consequently, they may encounter challenges in moving these container-based workloads into production. As DevOps and containers become more mainstream, these approaches will increasingly be used for applications with more complex data persistence and storage needs. To enable effective transformation, enterprises must find ways to embrace these new approaches while managing existing infrastructure, applications and processes in an integrated manner.

Containers were initially designed to support stateless applications, with minimal storage requirements. However, recent developments are enabling data to be persisted in external shared data storage volumes that can be connected to application containers, then disconnected and reconnected as the containers are created and relocated. This provides the data persistence required to enable containers to support production enterprise applications that need to continuously write to and read from data storage, such as analytics and BI applications.

Enterprise organizations are already moving faster – releasing new applications, services and features, while more effectively and efficiently managing IT infrastructure. Some of this is driven by changes already initiated in virtualization, open source software, agile methodology and cloud computing. But there is also a growing impact from the widespread adoption of DevOps and application containers. Organizations are now more aware of the need to usher in new technology and methodology, rather than resist it. 451 Research data indicates that nearly 40% of organizations are currently utilizing a DevOps approach, with enterprises typically citing improved speed, efficiency, reduced costs, and improved performance and quality among the biggest advantages.

Figure 1: Adoption of DevOps

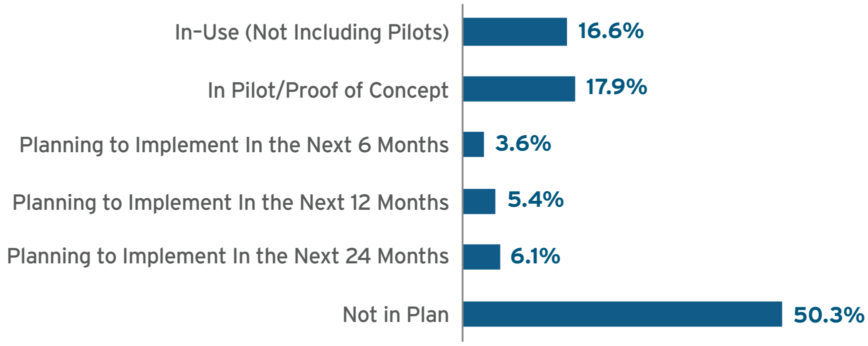
Q. Does your organization currently utilize DevOps approaches? [n=568]
Source: 451 Research, Software Defined Infrastructure, Q4 2015



There is further evidence of increasingly aggressive enterprise adoption and deployment of application containers for production use. 451 Research data indicates initial implementation of production container applications at 9%, up from 4% in Q1 2015, while broad implementation of production container applications stood at 5%, up from 2% in Q1 2015. This is impressive overall, given that modern application containers are barely three years old.

Figure 2: Implementation of Containers

Q. Please indicate your organization's implementation status for Containers (e.g. Docker, CoreOS). [n=553]
Source: 451 Research, Software Defined Infrastructure, Q4 2015



As the term implies, DevOps is primarily about software developers and IT operations teams working more closely together. Successful DevOps implementations in the enterprise achieve this collaboration, but also quickly pull in additional stakeholders, including security and line-of-business teams. Database and storage administrators should be critical stakeholders in successful enterprise DevOps deployments, but they are often overlooked. Accordingly, data persistence and storage considerations are not prioritized to the same extent as agility and automation.

The typical pattern is for DevOps backers to establish a successful effort – proving time, cost and other advantages – and then spread the effort throughout the organization. As DevOps and containers spread within an enterprise, data persistence and storage challenges eventually come into focus, sometimes after unnecessary complexity and pain. Data overload, data loss, and compliance and regulatory issues, for example, may surface downstream if organizations aren't thinking about data persistence and storage from the start.

II. Lessons Learned From Security

When the DevOps trend first emerged, the mantra of those pushing this accelerated model of developing and deploying applications was: 'If you're not embarrassed by your code, you waited too long to release it.' In other words, if there aren't a few bugs, vulnerabilities and defects in your application, you're fussing over it too much. Simply ship the code… now that we can deploy quickly, we can fix problems as they occur. For obvious reasons, this approach was largely incongruent with modern approaches to enterprise security.

Today security is one of the main drivers of DevOps, and a poster child for its benefits. A key reason for this change is the evolution of DevOps beyond cutting-edge organizations to mainstream enterprises. Incidents such as Shellshock and Heartbleed and the rise of ransomware also helped elevate the priority on security within DevOps. These situations illustrate the benefit of a more proactive DevOps capability, which means updating systems and clusters in a matter of hours rather than the typical reactive triage approach that is a reality for many enterprises.

When we consider data persistence and storage in the context of DevOps and containers, the pattern we saw with security is likely to repeat itself, with data persistence and storage concerns becoming imperative as applications transition from dev/test into production, and from stateless to stateful. Thus, we see an opportunity for enterprises to more effectively and efficiently embrace these disruptive trends by considering early on in this transition how data is managed and where and how it is stored. DevOps and containers help developers move more rapidly and effectively, but they can be burdensome to enterprise IT operations teams that must deal with the application of IT best practices and governance. By considering data persistence and storage early on, organizations can make DevOps and container implementations more palatable for administrators dealing with security, compliance, regulatory issues and other concerns. Enterprises are then better positioned to normalize, standardize and spread the successful deployment of DevOps and containers.

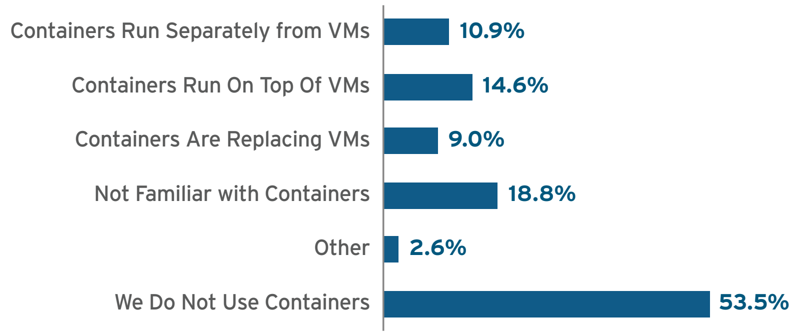
III. Proliferation of Containers: Mixed Mode Is the Norm

Among enterprises there is a mix of containers deployed inside VMs, containers replacing VMs and containers running on bare metal, adding to an already complex proposition for organizations and their database and storage administrators. This is one example of mixed-mode and intermodal IT within enterprise organizations, whereby traditional application management (mode 1) interfaces with emergent technology and methodology (mode 2). It is also becoming common to see mode 2 applications with mode 1 data.

Despite the dismay of pure container advocates, who argue that the greatest benefit from containers comes from running them on bare metal, the reality for many enterprises today is an intermodal approach whereby containers run inside VMs. This gives these organizations some of the speed, simplicity and manageability of containers while still providing the solid tooling, security and management of VMs that they know and trust. This highlights the utility of software and stacks that can accommodate both new applications and the modernization and migration of existing applications to SaaS, hosted infrastructure and the cloud.

Figure 3: Where Are Containers Running?

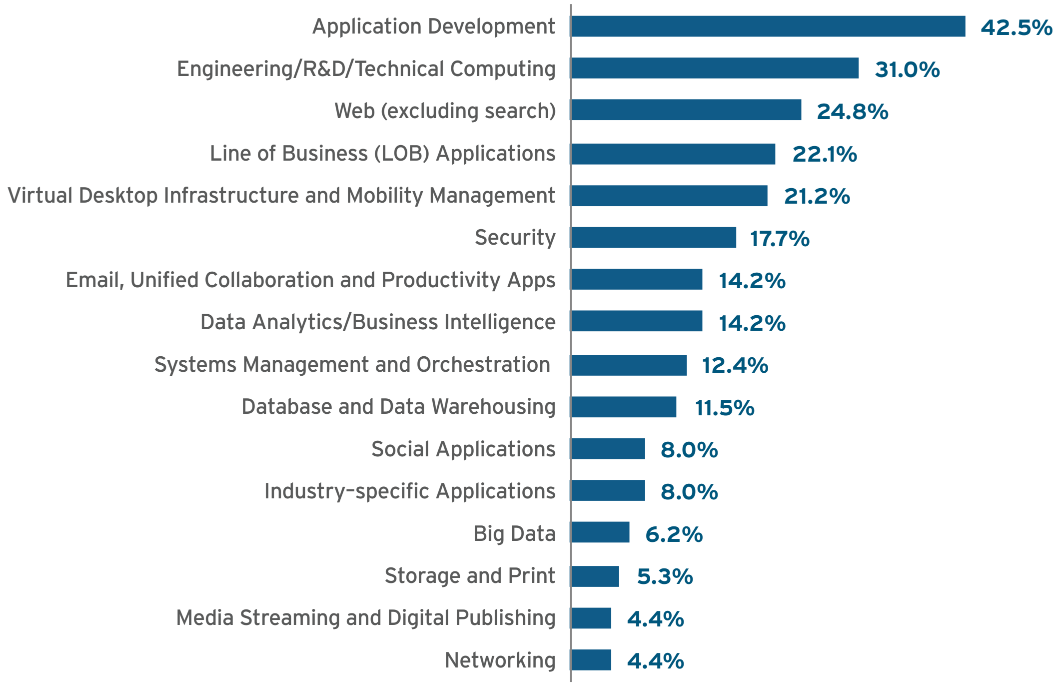
Q. What is your organization's strategy (if any) around the use of containers (e.g. Docker)? [n=458]
Source: 451 Research, Software Defined Infrastructure, Q4 2015



When we consider what types of workloads are deployed in containers, we see mainly development and testing, engineering, line-of-business and web applications, which is indicative of the stateless applications most prevalent with container use. However, the array of additional workloads in containers – including security, data analytics and big data – indicates that we're increasingly headed toward more mission-critical and stateful applications.

Figure 4: Top Container Workloads

Q. What are your organizationi's top 3 workloads that utilize container technologies? [n=113]
Source: 451 Research, Voice of the Enterprise: Cloud Computing, Q3 2015



IV. Consistency Between Dev/Test and Production Is Key

Proper and effective data persistence and storage is a critical part of maintaining consistency between dev/test and production environments – a key demand of organizations undertaking DevOps and container initiatives. While development and test environments are by nature transient, it is critical that the test environment closely replicates the conditions required for the application in production. This includes using similar data sets, as well as applying the same security, governance, compliance and data lifecycle requirements. For example, using sensitive data sets with test applications is poor practice, since the application is not well tested or hardened, and thus far more vulnerable to attack.

The need for dev/test and production consistency is also a function of the shorter distance and faster time for applications transitioning from development to production under a DevOps-and-containers model. While enterprise organizations must face cultural challenges in their implementation of DevOps, this consistency among environments is more of a technical issue, and one where we've seen containers as an effective part of the solution.

Enterprise DevOps and containers use is also very much about the need to satisfy developers' appetites for consuming 'infrastructure as code,' enabling rapid provisioning and tear-down of infrastructure – which can accelerate time to market by improving infrastructure provisioning turnaround for IT ops teams.

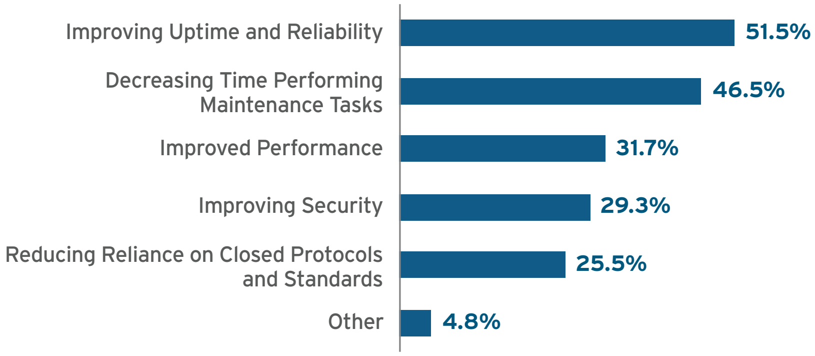
V. Container Applications, Drivers and Impediments

We can also get a sense of how containers are impacting enterprise IT practices and the market by considering the primary application workloads among enterprise users. With an increasing number of line-of-business, data analytics/business intelligence, big-data, database and data warehousing applications in containers, there will be a fairly significant connection between containers and data persistence/storage.

According to 451 Research data, key drivers of container adoption include: improved uptime and reliability, reduced time in performing tasks, improved performance, improved security, and reduced reliance on closed protocols and standards. These drivers, centered on both speed and efficiency, are fairly consistent with the forces pushing enterprises to adopt cloud computing and DevOps. They also intersect with effective and reliable management of data in applications and infrastructure.

Figure 5: Top Container Drivers

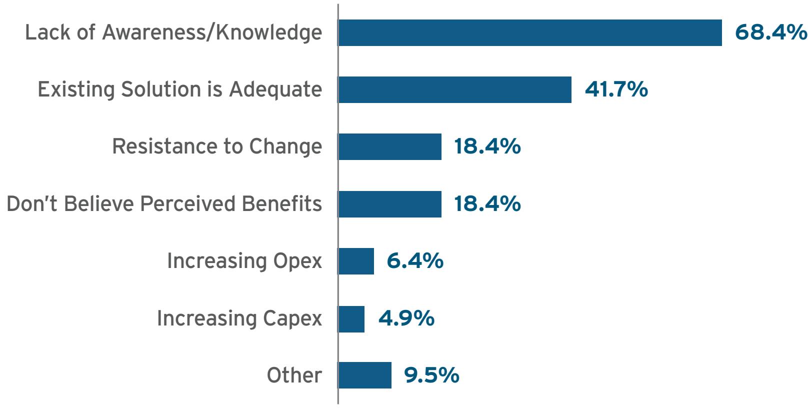
Please select top two technology drivers from the following for container technologies [n=501]
Source: 451 Research, Voice of the Enterprise: Cloud Computing, Q3 2015



In terms of inhibitors to container use, our research indicates that enterprise IT inertia and cost are significant factors. Despite these challenges, we see a number of vendors, including startups and established IT giants alike, rushing to respond to enterprise container-related gaps. Thus, we expect to see more tangible, technical container challenges emerge – including data persistence and storage issues – as awareness and cultural issues are addressed.

Figure 6: Top Container Inhibitors

Please select top two business inhibitors from the following for container adoption [n=326]
Source: 451 Research, Voice of the Enterprise: Cloud Computing, Q3 2015



VI. Container Persistence

Containers – and Docker, in particular – represent the latest shiny thing for many in the industry as they explore the potential benefits of lightweight approaches to packaging and deploying applications. The initial focus for containers had been on stateless web applications and microservices that can largely be spun up and down and moved around at will. Now enterprises are increasingly running data-intensive workloads on container technology.

However, standard Docker containers can include data volumes tied to an individual server. This means that if a container fails, or is moved from one server to another (which is likely, given that portability is one of the key benefits of containers), its connection with the data volume is lost. As such, containers are not designed to be persistent – a key requirement for any data-intensive workload.

That is not stopping database providers from working on Docker support, and there are a number of projects under way that attempt to solve the persistence problem. There are two approaches within Docker itself to circumvent this issue: data volumes and data volume containers. A data volume enables data to be stored in a separate container outside the boot volume, but within the root file system. Data volume containers take this a step further by creating a dormant container for the volume that can be exported to a new application container when it is created.

While data volume containers enable data to be persisted in a separate container from the application, the approach is still fairly rudimentary – the data volumes and data volume containers are still stored on the Docker host machine, and manual intervention is required to manage the connection between containers and associated volumes. Furthermore, there is no support for security beyond standard Unix file permissions, and there has traditionally been limited support for external storage.

That changed with the delivery of Docker v1.8, which added Docker volume plugins enabling users to take advantage of third-party storage systems, such as ClusterHQ's Flocker and others, which allows users to create, remove, mount and unmount data volumes. In addition, storage vendors are increasingly delivering their own native Docker volume plugins to enable Docker containers to take advantage of persistent data volumes.

VII. Flocker at Swisscom

Flocker is a project created by British company ClusterHQ that is designed to enable databases to run in Docker containers. Flocker can be used to run any Linux-compatible database in a Docker container, including Redis, MongoDB, MySQL and PostgreSQL. It works by creating a data volume from shared storage that can be reattached to a container as it is relocated.

Communications firm Swisscom is a named Flocker customer, and serves as a major validation of ClusterHQ's core mission. The company has been a longtime advocate of virtualization and cloud technologies, and is in the process of building a platform-as-a-service (PaaS) offering based on Cloud Foundry.

Swisscom saw Docker containers as the next step in delivering its Swisscom Application Cloud, but recognized that in order to achieve its desired density benefits, persistent databases would also have to be part of the containerized platform. Given that data services are currently managed outside Cloud Foundry, to deliver this Swisscom is now taking advantage of Flocker and the OpenStack Cinder API to create database containers from shared block storage.

VIII. fuboTV

Another example of a company running database software on Docker containers in production comes from fuboTV, which provides live streaming of soccer matches to subscribers in North America. It uses the MongoDB document database to manage customer data, store the program guide and metadata for the content catalog, and manage user data.

The company recently migrated its managed clusters from Compose.io to Google's Container Engine, which features Docker containers on the Kubernetes orchestration engine, in order to gain greater scalability and resource utilization.

For resiliency, fuboTV takes advantage of MongoDB's replica sets capability to provision data across Kubernetes pods in multiple Google Cloud regions. While data could be recovered from other nodes in the replica set, a container failure would reduce the speed of recovery to full resiliency. Instead, fuboTV takes advantage of Kubernetes volume abstraction to map each ephemeral container's MongoDB data directory to persistent network-attached storage.

IX. Capital One

Another interesting container use case is Capital One, which was able to implement containers successfully even with data-intensive production workloads in its highly regulated industry. Capital One's inclusion of data management in its container environment also helped the company to efficiently and adequately address storage requirements. Capital One integrated enterprise data management capabilities – including data sovereignty and geo-fencing, data governance and compliance – into its container environment. The company also integrated security into its container implementation with authentication, authorization and accounting included.

Capital One leveraged Docker containers and software-defined storage for its Analytics Garage, which supports big-data analytics application development. Key components to Capital One's effort included a scalable, fast, self-service platform for internal developers. Capital One also credits other components and open source software, including Hadoop, for not only its successful implementation of containers, but also for its ability to work with sample data and learn new tools and techniques.

X. Conclusion

It is clear that data management, persistence and storage are increasingly being considered in emergent DevOps and container implementations. Enterprise organizations undertaking digital transformation and embracing these trends must consider the time, cost and other risks of overlooking data and storage concerns. Furthermore, as security incidents illustrate, it pays to consider elements and stakeholders beyond developers and IT operations teams, including data persistence and storage issues, from the start. So enterprises must ask themselves whether they are adequately considering the fundamental changes in data management, governance and storage required for DevOps and container deployments.

Based on our research, there are several key findings regarding the intersection of data and storage with these emergent trends:

- We are seeing rapid adoption of application containers such as Docker among enterprises, including production implementations. Although organizations are embracing these technologies, they are also facing significant challenges pertaining to data management and storage.
- Enterprise organizations must balance the use of new technologies and methodologies such as DevOps and containers with existing infrastructure, processes and people. This means collaborative and combined support for both newer technologies and methodologies as well as existing and legacy applications and infrastructure in an integrated, intermodal way.
- Successful DevOps and container implementations in the enterprise require fundamental changes in the way organizations approach data management, governance and storage. There continue to be gaps in container readiness for enterprise production, particularly around data as it pertains to container migration, failover and backup. For success, it is imperative to engage security, storage and data governance teams up-front.
- Typical of enterprise hybrid cloud and mixed environments that leverage a broader array of application-layer components and infrastructures, containers often run inside VMs in today's enterprise implementations. However, there are scale, performance and other drivers for containers on bare metal, and our research reveals a mix of container deployment types.
- At present, containers are not designed to be persistent, which is a key requirement for any data-intensive workload. This is less of an issue with the stateless web applications today, but will become a bigger challenge for enterprises as they extend their container strategies to data-intensive applications and workloads, which vendors are working to support.
- We anticipate growing enterprise support for more data-intensive applications and workloads using containers, but lack of container functionality in terms of data persistence will remain an issue for the next 6-12 months at least.