White Paper

Container-Based Environments Require Key Data Services to Meet Enterprise Requirements

Sponsored by: Red Hat

Eric Burgener Lucas Mearian

February 2021

IDC OPINION

The move by enterprises to employ containers and cloud-native applications in support of digital transformation (DX) has profound implications for how data infrastructures must be architected. While improving DevOps speed and efficiency as well as application portability, containers increase by orders of magnitude the number of deployable applications compared with traditional or virtual machine (VM)-based environments. Therefore, traditional data governance techniques do not scale well and do not offer the granularity needed to manage data effectively.

As part of operating in a cloud-native mode, enterprises want to run legacy applications using containers. As container adoption increases, so too does the amount of data captured on persistent storage. Enterprises must ensure that data is easily available for use to uncover meaningful insights that drive better business decisions. That means key data services such as data discovery, data resiliency, data security, and data governance are required to deliver true enterprise-class storage in container-based environments. And those services must apply to data in all of its states – whether it's in transit, being used in analytics or other processes, or it is at rest.

IN THIS WHITE PAPER

For many enterprises, the need for increased agility encouraged by DX is driving significant changes in how applications are developed and moved to production and how information technology (IT) infrastructure is architected, consumed, and deployed. The use of container technology is rapidly becoming a key factor in improving information technology agility, but it has significant implications for how data infrastructure needs to be crafted and managed. This white paper traces the rise of container technology, discusses the new data service requirements it imposes on IT, and introduces the key data services that enterprises need to provide with that data infrastructure. As they use container technology, enterprises will need to ensure that they are providing these data services as part of the supporting infrastructure.

SITUATION OVERVIEW

As they move into the digital era, enterprises have had to adapt to a much faster-paced, more dynamic business environment. The structural changes this drives comprehensively impact all areas of the business – including how enterprises explore new market opportunities; develop, deploy, and support products and services; define and manage internal processes and infrastructure; and interact with customers and suppliers. Above all else, this new era requires the agility to react faster to developing and/or changing business conditions, ultimately resulting in the ability to bring new products and services to market more quickly.

In response, most enterprises are undergoing DX (i.e., a move to much more data-centric and digitized business models). As part of this journey, more data than ever before is created, captured, stored, protected, and analyzed. The enterprises that do this most successfully view data as a critical business asset, finding new and innovative ways to leverage it to create competitive differentiation and drive faster, better business insights. Two areas in particular are being pursued by digitally transforming enterprises:

- **More agile methods for application development and deployment:** Next-generation applications are being added that leverage artificial intelligence (AI), machine learning, and big data and analytics to drive value for enterprises based on their data. Older, more static waterfall approaches to application development as well as legacy workload deployment methodologies are cumbersome and inflexible and clearly keep IT from being as agile as it needs to be. In response, IT organizations are implementing newer, more agile approaches. DevOps is the set of practices that combines agile application development and IT operations, shortening the systems development life cycle and providing for "continuous integration and continuous delivery" (CI/CD) with high software quality. These new approaches leverage microservices architectures and container-based deployment methodologies that have become the hallmark of "cloud-native applications." It is clear that these newer approaches deliver capabilities that are appreciated by both application developers and IT infrastructure administrators responsible for deploying and maintaining workloads.
- **Enterprises moving toward running mission-critical apps in containers:** Next-generation applications drive the need for heightened performance, availability, scalability, and security requirements, and as a result, enterprises are modernizing their IT infrastructures. According to IDC's 2020 *Cloud Pulse Worldwide Survey on Enterprise Cloud Management,* over the next two years, about 30% of enterprise applications will be using containers to enable deployment in both private and public clouds, and 31% of cloud-based containerized applications will consist of refactored legacy applications.

Defining Modernized Infrastructure

Modernized infrastructure leverages new technologies and architectural designs that deliver increased performance and capacity density, higher availability, improved efficiencies, simpler management, and much better agility. Technologies most in demand for enterprises undergoing data infrastructure refresh include software-defined storage (SDS), NVMe and other solid state storage advancements, cloud technologies, container-based application architectures, and much more comprehensive automation and orchestration capabilities. SDS is much more flexible than legacy, hardware-centric architectures, and infrastructure based on software-defined designs provides freedom of choice in hardware selection, incorporates self-managed data tenets that make systems much easier to manage, enables nondisruptive technology refresh that extends the life cycle of data platforms, and results in significantly lower total cost of ownership (TCO).

Cloud-Native Applications: Software Defined, Container Based, and Automated

Cloud-native applications have evolved over time to address the long-standing disconnect between how application developers want to work and how IT administrators want to deploy and manage applications (and the IT infrastructures that support them). As developers moved to more agile application development methodologies, IT's inability to rapidly respond to requests to provision or add resources, introduce enhancements and bug fixes for production workloads, recover data in the wake of failures, and make desired new services (like accelerated compute) available on demand pushed developers toward the public cloud. The public cloud's self-service-oriented "on-demand provisioning" of resources complemented the new, more agile development workflows very well. With developers provisioning and using infrastructure through the public cloud, however, IT was very concerned about meeting performance and availability service-level agreements (SLAs), maintaining data integrity, protecting data with backups, crafting disaster recovery strategies, addressing evolving security needs, and meeting governance and compliance requirements.

The use of containers as a deployment methodology for applications has helped address both developer and IT administrator needs, and their use is one of the key foundations of DevOps environments. Containers are a logical construct (conceptually similar to virtual machines) with two important differences; they are both lighter weight and more agile than operating system (OS)-bound, monolithic VMs. Container agility solves the problem of making applications easily portable and ensures they can be quickly and easily moved between traditional IT and private and public cloud infrastructure as needed. Containers allow developers to work with public cloud–based development environments and then hand the containerized application environment over to IT administrators, who can then much more easily and reliably deploy and maintain the containerized applications.

For those enterprises that are working with hybrid cloud environments — which will ultimately include most organizations — IT administrators will select from three different deployment models: traditional IT infrastructure, private cloud infrastructure, and public cloud infrastructure. IT administrators will choose the best deployment model, weighing workload requirements and costs as they seek to achieve optimal workload placement. Initial deployment often requires IT to move an application from the public cloud to one of the other two locations, and they may seek to move the workload again as it evolves over time. As IT administrators maintain and enhance a production workload, new releases will have to be deployed as they are made available. These requirements underlie their desire for a consistent, rapid, and low-risk method for deploying applications.

Microservices architectures move away from monolithic application designs to more composed applications that encompass a number of different features, each of which may be running in its own container. So an "application" becomes a collection of features, each of which may be running in its own container, communicating with other containers, and providing other features through defined APIs, all of which collaborate to provide the application "service." Each container encapsulates an entire runtime environment for its component and is completely agnostic to the underlying infrastructure (meeting the letter of the law for "software defined"). The use of microservices also significantly reduces the amount of development time and regression testing that has to be done to ensure reliable deployments because each "function" operates independently in its own container. There is far less code to write and test, a reality that makes it much easier to deploy new features and/or bug fixes into existing applications by just updating a single microservice while the rest of the application continues to run unimpacted. The use of containers, coupled with microservices-based application designs, enables IT organizations to build more efficient and reliable software-defined architectures at scale and increase the pace at which new applications and features can be deployed.

Unlike VMs, which contain their own operating system, multiple containers can share a single kernel or operating system (which can itself be running in a container). For this reason, they can be much lighter weight than VMs. As mentioned previously, a container might just contain a particular function (a microservice) and requires access to a container host (which contains a host operating system like Linux or Windows) to run. And microservices-based designs encourage the reuse of code (e.g., a "sort" routine or a "search" function) across multiple applications, potentially significantly increasing developer productivity. Cloud-native applications use both technologies — containers and microservices design — together to achieve a much more agile, efficient, and software-defined infrastructure.

In moving from monolithic applications, which might be deployed in their own dedicated VM, microservices architectures using containers do increase the number of objects (i.e., containers) that must in general be managed. Releasing a new application may require deploying 10–100+ containers. While the portability, flexibility, more efficient scaling, opportunity for code reuse, and much easier and more reliable maintenance and upgrade paths are desirable, the use of containers does somewhat increase management complexity. For this reason, IT organizations using containers typically also extensively use automation. Automation allows a multi-container application to be reliably deployed or redeployed with a single click, and it ensures that workload mobility, backup, and other administrative processes operate correctly at the application level by appropriately coordinating between all the relevant containers. As part of the DevOps journey for most enterprises, then, they are also investing heavily in automation and orchestration.

Containers Drive New Data Infrastructure Requirements

A move to the use of containers has profound implications for how data infrastructure must be architected. First, since the use of containers allows applications to be split into potentially many components, each of which runs in its own container, these environments can increase the number of data objects (i.e., storage volumes) that have to be managed by one to two orders of magnitude (relative to traditional or VM-based environments). Traditional data governance approaches do not scale well into this range and do not offer the granularity needed to manage data effectively in these types of environments.

For example, a traditional Microsoft SQL Server database running in a VM may require two storage volumes (one for the log and one for the data). A microservices-based implementation of that same database may have up to 100+ separate containers collaborating to provide the same application "service." Most of those containers will require their own (much smaller) storage volume, driving the need for an increase in volumes, roughly two orders of magnitude greater than a traditional VM-based deployment of that same application. As more microservices or workloads are added to the same environment, the need for additional volumes will increase accordingly. Traditional enterprise data solutions may be able to handle thousands to tens of thousands of volumes, but they were not designed to provision and manage hundreds of thousands of volumes very efficiently.

Second, to provide maximum agility, container images were designed to be stateless. As the use of containers has become more widespread, enterprises want to run both cloud-native and legacy applications using containers. While the stateless architecture is an excellent fit for cloud-native web-scale workloads, many traditional applications running on databases are stateful, and native container capabilities need to be augmented to support stateful applications requiring persistent data. Legacy storage systems impose limitations in supporting stateful applications in containers — they don't offer a RESTful API to access persistent data, they aren't designed to manage the large number of storage volumes required in container-based environments, and they don't support the extensive automation or portability needed.

Data Services in Container-Based Enterprise Environments

The goal of the data services required for container-based environments is to ensure that the data is appropriately accessible by authorized users with a consistent experience and that that consistent experience scales seamlessly as the environment grows. As the amount of data captured increases over time, enterprises must ensure that the latest data is easily available for use to uncover meaningful insights that drive better business decisions. There are four key data services that are required to deliver true enterprise-class data in container-based environments, and these must cover the data in all its states: data in motion, data in action, and data at rest. Data in motion refers to how data is captured and moved to various locations and workflows as part of its life cycle. Data in action refers to how data is used for analytics and other operational processes that actually drive the insights, while data at rest refers to how data is stored, protected, and retained when it is either not in motion or not in action. The four key data services are data discovery, data resiliency, data security, and data governance.

Keep in mind that the underlying hardware infrastructure in container-based environments is most likely to be constructed using server-based, software-defined storage. It may also, however, include an ability to leverage other designs like multi-controller arrays, hyperconverged infrastructure, cloud-based tiers, or other types of data in the storage layer. The software-defined infrastructure platform should be flexible enough to simultaneously accommodate a variety of different types of on-premises storage architectures, all of which support the industry-standard Container Storage Interface (CSI) specification for accessing persistent storage.

Data Discovery

Enterprise storage must be accurate, available, and accessible at all times. Storage vendors have implemented a number of features in their enterprise storage systems to imbue data with these characteristics. This also includes the ability to automatically discover new containers/apps for DevOps. Without knowing where the applications and associated data are, it's impossible to ensure it is protected by the proper data services.

Unorganized data is also difficult to manage, and there is a growing amount of it posing a security risk (i.e., not knowing where data resides could leave it open to unauthorized access and theft or manipulation). In addition, artificial intelligence and machine learning rely on the discovery of data. Understanding where data sets reside opens them up to analysis, but in order for that data to be useful, it must first be classified and identified using metadata so that it can be cataloged and made easy to find.

Data discovery is about the ability to find the right data at the right time; this is a key pain point for data engineers and data scientists. Classically, this has been the realm of enterprise search. Most enterprises see AI-enabled search as a key asset for research, analysis, and decision making. Search systems include departmental, enterprise, and task-based search and discovery systems as well as cloud-based and personal information access systems.

According to IDC's *AI-Enabled Enterprise Search 2019 Trends Survey* published in January 2020, nearly 40% of enterprises said that their AI-enabled search tool has substantially decreased or completely eliminated the need to access multiple systems for information. According to the same survey, most organizations see benefits from AI-enabled search within the first six months.

Data Resiliency

In software-defined environments, the underlying hardware becomes secondary as much of the functionality is performed in software. For example, erasure coding in SDS partitions data into fragments and writes them across multiple drives and servers; this enables the original data to survive multiple drive or server failures. N-way replication is another example where SDS enables multiple nodes in a cluster to replicate their data (e.g., a database) to each other and to a main target database. All nodes in the cluster are both database source and target and can push the data to other locations, such as an offsite location for backup. Regardless of which approaches are used, resiliency features to check for in the storage infrastructure supporting container-based environments include:

- **Resilient storage:** RAID, erasure coding, and/or local n-way replication
- **Data protection:** Leveraging existing backup and restore
- **Disaster recovery:** Cross-region replication

Data Security

Enterprise data must be secure, and this is enforced through encryption, access control, and key management. Depending on government regulation and/or business-specific governance, administrators may need to ensure that data is not only encrypted while at rest but also in flight.

Role-based access control (RBAC), which allows administrators to define varying levels of access based on different user roles, is also important. For block-based storage, there are some controls at the storage layer (such as LUN mapping, masking, and zoning) that can be used to make systems more secure. For file- and object-based storage, access control is usually defined at the individual file and/or object level. For unstructured storage environments, RBAC fits more into the data governance data service. Access control considerations are mentioned in the Data Governance section.

Malware and ransomware attacks are afflicting more enterprises than ever before. While encryption does provide one layer of protection against bad actors, another layer of defense referred to as "air gap" protection can be important. Air gap protection simply means data is stored offline with limited internal accessibility. For example, data could be stored in a backup copy. That way, if data does become corrupted or is encrypted by a bad actor demanding a ransom to unlock the data, the administrator can just recover the most recent "uncorrupted" copy of the data from an air-gapped version.

Data Governance

Given the scale at which most container-based environments need to operate in terms of storage volumes managed, data governance is a critical area. Container environments operate in clusters where different microservices will run (and can be moved to run) across a variety of different nodes.

While the term *data governance* often refers to the practice of improving the quality and reliability of data, it also encompasses the ability to ensure regulatory compliance in light of where data may reside.

Because containerized applications are lightweight and portable, they can take full advantage of cloud services, and their platform deployment can be dynamic, including private, public, and hybrid cloud and multicloud environments. One often cited attribute of containerized applications is portability; containers can move between OS platforms and between clouds. The analogy often used is to that of a cargo shipping container that can be loaded onto multiple transportation means (i.e., ships, railways, and tractor trailer trucks). The concept is similar for containerized applications, which can be spun up and moved between a myriad of environments. That means data produced by containerized apps is also portable and can be geographically dispersed; this means organizations should take a strict approach to data privacy, auditing, and regulatory compliance.

For example, governmental and industry regulations such as the Sarbanes-Oxley Act for financial auditing, HIPAA for healthcare data privacy, and the EU's GDPR protecting personally identifiable information could all apply to data produced by containerized applications.

To achieve compliance with these regulations, business processes and controls should include well-defined management controls to govern the data depending on the regulations that could apply. Data governance tools address the central management of data through policies, protocols, and procedures that control how data is managed and stored.

RED HAT DATA SERVICES

Red Hat is a multinational software company, operating as a wholly owned subsidiary of IBM, that provides commercial software-defined infrastructure products to enterprises based around open source technologies. Originally founded in 1993 and acquired by IBM in July 2019 for $34 billion, it is well recognized as the leading provider of commercial open source technologies in the industry. Red Hat OpenShift is a family of containerization software products whose flagship offering is Red Hat OpenShift Container Platform. OpenShift Container Platform is a software-defined infrastructure platform built around Docker containers, orchestrated by Kubernetes, and hosted on a foundation of Red Hat Enterprise Linux.

OpenShift Container Platform provides an infrastructure foundation for digitally transforming enterprises looking to move to container-based environments. The vendor started with a focus on the key data services that enterprises need to consolidate legacy and next-generation cloud-native applications under a single infrastructure. Red Hat Data Services is a portfolio of products and services that deliver simplified access, a consistent experience, and dynamic scale for data across hybrid cloud and multicloud. The portfolio supports a wide range of rich content (block, file, and/or object) and provides access to data from anywhere and for anyone, including application developers and data scientists. It also supports self-service capabilities that meet developer agility and IT governance requirements, and it scales seamlessly to hundreds of petabytes while meeting the critical enterprise application requirements of discovery, resiliency, security, and governance.

Red Hat Data Services offerings are integrated with Red Hat OpenShift, spanning all three states of the data. For data in motion, they decrease the time to find the right data, help accelerate processes by establishing automated data pipelines, enable new business models, provide the agility to cope with dynamic events and markets, and help organizations move to more real-time-oriented computing. For data in action, they enable DevOps CI/CD pipelines, enable more powerful and insightful data analytics, accelerate processes, increase the self-reliance of different constituencies (developers, data scientists, and IT managers) without endangering governance or compliance, and generally remove the artificial constraints to innovation. For data at rest, they protect critical datastores that unlock value for the organization, enable easier sharing of data, make internal IT processes more consistent and efficient, and increase the performance and scalability of data-centric business operations.

CHALLENGES/OPPORTUNITIES

Although microservices-based architectures offer the flexibility to create much more efficient IT infrastructure, they are still relatively new. Digitally transforming enterprises are clearly moving in the direction of DevOps, but it is a journey, and businesses are at varying stages along that path. IT organizations are already working with containers and slowly building out the long-term supporting infrastructure they will need for container-based environments. Several key strategic decisions will need to be made along the way, such as:

- What role does virtualization play with containers?
- How do we evolve our application portfolio and split it most effectively between replacing, rehosting, and refactoring existing workloads into container-based environments?

- What is the right orchestration platform for containers?
- What is the optimum location for each container-based workload?

Most enterprises will develop new applications on the cloud-native model but will look to move existing applications to microservices architectures more slowly depending on refactoring decisions.

Red Hat has an excellent opportunity with enterprises that decide to evolve away from the use of virtualization technologies to container-based environments based on Kubernetes for efficiency and agility reasons. OpenShift Container Platform is a mature platform, backed by thousands of Red Hat developers as well as the open source development community. Through OpenShift Container Platform, Red Hat Data Services delivers enterprise-class storage and data services for container-based environments and has 24 x 7 support today. Red Hat OpenShift is in use by thousands of customers, many of whom are bellwether accounts in the Fortune 1000. Enterprises that already know how they want to use containers in their production environment would do well to evaluate how Red Hat OpenShift (and the Red Hat Data Services it supports) offers their businesses the key to unlocking the value of their data to drive better business results.

CONCLUSION

Nearly three-quarters of enterprises are undergoing DX plans to refresh their server, storage, and/or data protection infrastructure to meet evolving requirements. And 91% of those organizations deem such technology refreshes critical success factors in achieving their DX objectives. Next-generation applications (i.e., containerized cloud-native apps) are driving the need for enterprises to rethink how data services can be applied in support of modern architectures. Today, containerization applies to both greenfield applications and legacy applications that enterprises want to move to a more cloud-native deployment model to achieve DevOps efficiency and speed, application portability, and IT flexibility. This strategy drives the need for a set of enterprise-class data services (data discovery, data resiliency, data security, and data governance) that can be consistently applied across both next-generation and legacy applications.

Red Hat OpenShift is an open source, software-defined, and container-based IT infrastructure platform that leverages Kubernetes orchestration and, with Red Hat Data Services, delivers the enterprise-class data services needed to manage enterprise applications across all three data states — data in motion, data in action, and data at rest. With thousands of deployments, Red Hat OpenShift is a proven platform that has already provided the foundation for DX for enterprises both large and small. Red Hat Data Services products integrated with OpenShift Container Platform provide cloud-native abstractions that unlock the business value of data in hybrid and multicloud environments for developers, data scientists, and IT managers alike while ensuring data is resilient, secure, and managed according to established governance policies. The combination of Red Hat's software-defined infrastructure and open source heritage also gives the company's solutions compelling economics, making Red Hat an excellent choice for enterprises that need cost-effective, scalable, and agile IT infrastructure to support their DX.

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

140 Kendrick Street
Building B
Needham, MA 02494
USA
508.872.8200
Twitter: @IDC
idc-community.com
www.idc.com