White Paper

The Role of Virtualization in the Era of Containers and Cloud

Sponsored by: Red Hat

Gary Chen
December 2018

IDC OPINION

Virtualization is the foundation of the datacenter, with 88% of workloads being virtualized today, according to IDC's virtual machine (VM) forecast. However, while virtualization has matured and become ubiquitous, it is still evolving and will play a key role in the datacenter for years to come. Traditional virtualization, which dramatically improved server utilization via consolidation and greatly reduced server provisioning times, will still exist to serve legacy workloads. But virtualization is playing new roles in cloud and next-generation applications. Customers are beginning to supplement traditional virtualization management with private cloud architectures, which emphasize automation, orchestration, and self-service. On the public cloud side, virtualization serves as the infrastructure foundation of nearly all public clouds. For next-generation applications, customers are shifting to containers and new microservices-based architectures. Containers are today used most often with server virtualization technology, which is better at handling hardware provisioning and multitenancy, while containers act as a package for applications. This growing range of hypervisor use cases is changing hypervisor deployments and requirements. Virtualization has embedded itself firmly into the software stack and will remain the foundational component of the datacenter, handling both traditional workloads and next-generation cloud-native workloads.

SITUATION OVERVIEW

Virtualization is firmly entrenched in nearly every datacenter infrastructure today and is a ubiquitous part of the software stack. Few technologies have seen a ramp-up in enterprise IT like virtualization. The cost savings from consolidation were obvious and immediate. As virtualization matured, customers began to realize many agility benefits as well. Virtualization became the foundation for the software-defined datacenter and clouds of all types, private and public.

The KVM Hypervisor

Kernel-Based Virtual Machine (KVM) is the leading open source hypervisor and enables Linux to become a virtualization host for both Linux- and Windows-based virtual guests. The project, hosted by the Linux Foundation, is an integral part of Linux. KVM has found an important place in IT technology because of several key factors:

- KVM is available anywhere that Linux runs. Linux is used in a tremendous number of systems and use cases, and KVM has benefited from making itself available to nearly every Linux distribution or build.
- Because KVM is part of Linux, KVM is highly integrated with key open source projects such as containers, Kubernetes, and OpenStack, which rely on Linux. As hypervisors play an increasingly embedded role in many of these new open source software, KVM integration is key to creating an end-to-end solution.

- Much of the innovation around cloud and containers is done in open source. Linux, and by default KVM, is often a part of, or integrated with, these solutions as foundational elements because Linux and KVM are the standard for open source infrastructure.
- KVM has been used and proven in a wide variety of use cases over time, including on-premises enterprise virtualization, private cloud (such as OpenStack), hyperconverged and other hardware systems, public cloud, and telco networks for network functions virtualization (NFV). The ongoing evolution of KVM demonstrates its maturity, flexibility, and near-universal compatibility with anything that can run Linux.

Shifting Infrastructure Requirements

The IT industry is constantly evolving and today there is a shift happening in how apps are architected and built. Digital transformation is pushing businesses to invest more in software and deliver this software increasingly faster. Modern applications are beginning to use a cloud-native microservices-based architecture. The use of microservices decomposes an application into smaller logical services that can be developed in parallel and updated and scaled independently. Developers are also increasingly moving to Agile development methodologies, allowing them to ship code continuously, and becoming more integrated with operations teams (DevOps).

As the application architectures and development methodologies change, it is driving the requirement for different infrastructures optimized to support them. The pace of software development and deployment is increasing dramatically, and infrastructure shouldn't hold back that pace. This is one of the reasons why the industry has been seeing a shift to cloudlike infrastructures, both private and public. These infrastructures are API driven, highly scalable, available on demand, and fully automated. Most of these clouds have historically been VM based but now are evolving to also support containers.

What Is a Container?

Containers use various operating system (OS) abstraction and virtualization technologies to provide an isolated, resource-controlled environment to run applications. It is basically a type of sandbox around a normal application OS process and is generally considered to be much more isolated than an uncontainerized process, but not as isolated as a VM. Container images define how applications are packaged and only contain the application and its dependencies such as libraries, configurations, runtimes, and tools, making it more lightweight than a VM. The container image and runtime are standardized through the Open Container Initiative (OCI), making containers highly portable and universal. Kubernetes, a container management system, is a de facto standard in the industry today, which makes the container control plane standardized as well.

Containers are extensions of operating systems, a better way of sandboxing an application process, making the OS the foundation for container execution. Containers have a long history with Linux. Some of the key functions needed for containers began to be developed in the Linux kernel a decade ago. When Docker began developing the modern container, now part of OCI, it was developed on Linux. While today containers support other operating systems such as Windows, IDC research shows that nearly 80% of containers are on Linux.

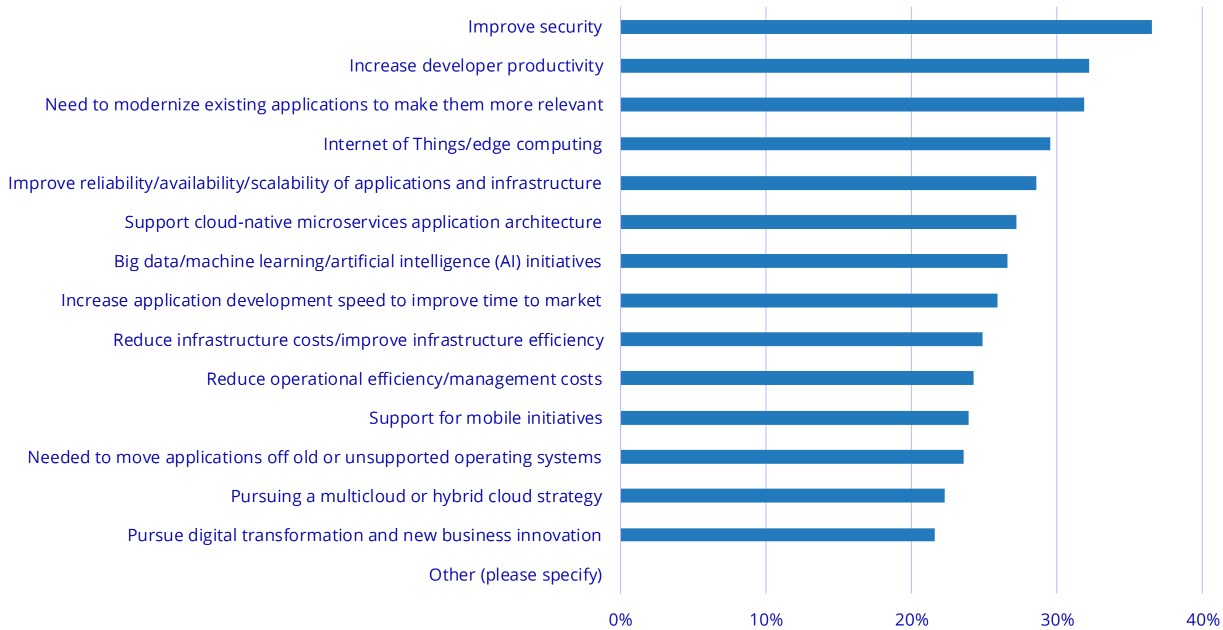
Why Are Enterprises Adopting Containers?

Containers are deployed to meet a wide variety of application development and infrastructure needs. Figure 1 summarizes findings from a recent IDC survey that uncovered 14 of these drivers. We summarize a few key drivers in the sections that follow.

FIGURE 1

Top Container Drivers

Q. What were the primary drivers that caused your organization to initially deploy containers?



n = 301

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

Containers Are Highly Synergistic with Microservices and Agile Development

Containers are an efficient way to encapsulate a microservice into highly portable units for deployment to a variety of infrastructures and clouds.

With Agile development, the workflow and software pipelines are becoming accelerated, often with the use of continuous integration/continuous deployment (CI/CD) systems. Containers are lightweight and portable, making them a good vessel to put code into in order to push them through these new software pipelines and reduce the time to deployment.

Containers Enable Application Modernization

Containers can also be used for some existing applications, though some may require some level of code changes or refactoring. This allows the application to be more portable for migration to the public cloud and can also retrofit the application into newer developer workflows and software pipelines.

Containers Enable Automated Operations at Web Scale

Container orchestration and management systems are designed to deploy and manage distributed, cloud-native applications. To do this at any scale, full automation is a core mantra of these systems. These systems are also designed to operate at web scale, providing scalability and resiliency capabilities to modern applications. However, scalability and resiliency are still highly dependent on the application's architecture, with cloud-native applications designed to better handle these aspects.

FUTURE OUTLOOK

The Role of KVM in the Era of Containers and Cloud

Virtualization and KVM are core technologies that currently underpin multiple types of open infrastructure and will continue to power new infrastructure for the future. KVM is seen in several different deployment scenarios:

- **As part of a traditional virtualization solution.** Customers deploy fleets of hypervisors on servers, managed through a centralized console.
- **As part of a private or public cloud.** OpenStack is primarily used in private clouds, and KVM is the default hypervisor for OpenStack. Public cloud providers largely prefer open source hypervisors for reasons of cost and customizability, and KVM is the preferred choice today.
- **As part of a container platform.** Containers are an OS-level virtualization technology and still require infrastructure to run on, like any other OS or application. And like other workloads, virtualized infrastructure is generally preferred for containers today because it is much easier to manage than bare metal and is widely available in today's datacenters. However, the performance benefit of running containers on bare metal is leading to increasing interest in this deployment model for the future.

Containers Versus Virtualization?

One common misconception today is that containers replace virtualization. While some of the functions of containers resemble virtualization, they actually operate very differently, and containers generally complement virtualization. Virtualization operates at the hardware level, virtualizing server hardware and carving it up into smaller pieces. Containers operate at the operating system level and application packaging level. IDC data shows that 72% of enterprise containers run virtualized today versus bare metal. Hypervisors provide key functionality to containers:

- **Hypervisors provide much stronger isolation than containers.** This is key especially in the public cloud where you would never see different tenants separated only by a container boundary. Inside enterprises also, there are many reasons to use a hypervisor for isolation. Different business units or workloads may have different policies or regulatory requirements, and a hypervisor provides additional separation in these cases.
- **Containers don't absolve organizations from having to provide a stable and robust infrastructure underneath.** Hypervisors are standard today for hardware provisioning and have very mature orchestration tools for this task, and they are already in place in nearly all enterprises.
- **More flexible OS choices are available.** If a container host OS was installed on bare metal, all containers would share that same kernel. Using VMs to run multiple container host OSs allows more flexible mixing and matching of OSs. This could be a mix of Windows and Linux, different versions of the same OS, and different patch levels that might be needed by different users

and applications. The reality today is that enterprise customers run many different patch levels and versions of an OS and use both Windows and Linux containers. Managing dedicated physical nodes for each type would create a lot of management overhead.

While IDC believes that most containers will continue to run on hypervisors, that does not mean that the dynamics of the hypervisor market will not be affected. Hypervisors as an embedded part of a container stack play a much different role today than traditional VMs. Many of the requirements and value of the hypervisor in a container stack will be different, and this could alter the feature set and cost of hypervisors going forward.

Container and VM Coexistence

Most enterprises will run a mixed mode of VMs and containers for the foreseeable future. Not every application can be easily containerized, as container technology is not fully transparent and backward compatible like virtualization was. Figure 2 illustrates recent customer data from an IDC study showing what levels of modification were required of existing applications to containerize them.

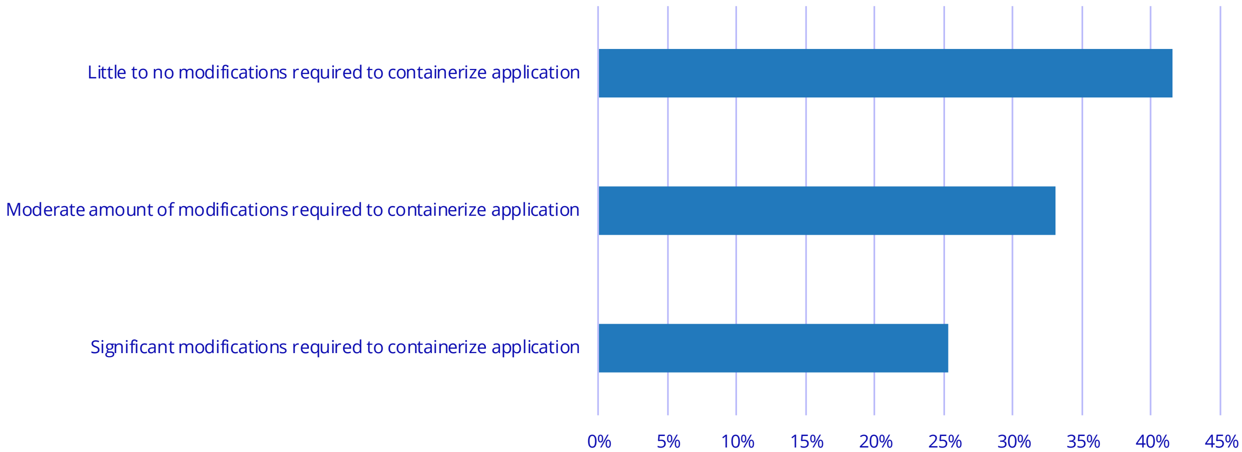
The reality is that many existing applications will never be containerized. The opportunity to containerize some applications may not come until that application is decommissioned and a whole new replacement is brought in. For a lot of applications, this is a very very long time.

Enterprises have always accumulated and managed multiple generations of technology in parallel. It would certainly be nice to always be using the latest and greatest, but this isn't financially viable. Enterprise applications will be deployed into both VMs and containers for the foreseeable future, and enterprises will need a converged platform to support both. Why a converged platform? Having VMs and containers in separate silos creates integration and management problems. For example, applications commonly talk to or integrate with other applications. Some of these applications might be VM based, while others will be containerized. Even different parts of the same applications might be running in different technologies, with say the database tier in a VM and the web tier in containers. To effectively manage these applications and optimize and monitor them, customers need to manage and integrate across multiple generations of technology. KVM has the opportunity to underpin all of these deployment types, whether it is managed with traditional virtualization management, private cloud, or a container system like Kubernetes.

FIGURE 2

Modifications Required for Containerization of Existing Apps

Q. For your existing applications that were migrated to a container, please indicate the percentage breakdown of what level of modification was required to containerize these applications.



n = 301

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

VMs in a Container?

KVM opens up some interesting possibilities because it is just another process in the Linux OS. Because of this, a Linux instance can serve multiple roles simultaneously, serving as both a hypervisor and a container host (along with other general-purpose OS functions). A recent open source project called KubeVirt leverages this to bring together the world of VMs and containers in a new way, by putting a VM into a container.

How does this work? Well, remember that KVM is simply a kernel module that enables the Linux kernel to become a hypervisor and reuses many of the OS features already there for virtualization. Thus, with KVM, a VM is just another Linux process. Now remember that a container on Linux is just a normal process that has additional sandboxing applied to it. Using Linux and KubeVirt, we can take that KVM VM process and apply a container sandbox around it. But why would someone want to do that?

- By wrapping a VM in a container, the VM can be managed with any choice of modern tooling such as Kubernetes. This allows VMs to leverage the modern APIs, automation, and management paradigms that are prevalent in container platforms.
- VMs can be put into modern workflows and software pipelines to modernize VM-based applications.
- It's possible to integrate traditional assets into modern container applications, creating hybrid VM- and container-based apps.
- Converging VMs and containers onto the same platform simplifies management, operations, troubleshooting, and monitoring.

RED HAT PROFILE

Red Hat has been a central figure in fostering the development of KVM. With Red Hat's influence in Linux and open source and the company's long history of successfully commercializing many open source projects, KVM powers many forms of modern Red Hat infrastructure:

- The Red Hat Virtualization offering consists of an enterprise implementation of the KVM hypervisor that is part of an optimized version of Red Hat Enterprise Linux (RHEL). It also includes traditional virtualization management based on the open source oVirt project, which is a Red Hat JBoss Middleware Java application, allowing it to run on a fully open source stack of Linux and Java.
- Red Hat OpenStack Platform brings a modern private cloud environment to enterprises by virtualizing resources from industry-standard hardware, organizes those resources into clouds, and manages those resources so users can access what they need, on demand. RHEL and KVM play an instrumental role in OpenStack, powering the underlying compute nodes.
- Red Hat OpenShift Container Platform, Red Hat's enterprise distribution of Kubernetes, can run on either bare metal or virtualized servers. While there are pros and cons to both, the majority of container deployments are on virtualized servers like Red Hat Virtualization for several reasons:
  - Easier management than bare metal with the virtualization management web UI and REST API
  - Coexistence with existing VM-based workloads
  - Infrastructure scalability for OpenShift nodes
  - Modernizes the existing virtualization platform with containers and cloudlike services
- Red Hat Container-Native Virtualization (CNV) is a new technology currently in incubation status that helps development teams that have adopted or want to adopt Kubernetes as part of the Red Hat OpenShift Container Platform but have existing virtual machine–based workloads that cannot be easily containerized. This technology provides a unified platform where developers can build, modify, and deploy applications in containers and VMs within a shared environment. CNV is based on the KubeVirt open source project and powered by the same KVM technology in Red Hat Virtualization and Red Hat OpenStack Platform.

By leveraging KVM, Red Hat is uniquely positioned to help businesses grow beyond traditional virtualization to containers, private cloud, and hybrid cloud management. All parts of the Red Hat stack are open source, but they are all integrated with the same KVM hypervisor, tested as one, and supported by a single vendor.

Red Hat Infrastructure Migration Solution

The Red Hat infrastructure migration solution helps organizations reduce the cost of their existing infrastructure investments. The solution analyzes the cost of the existing infrastructure and provides an alternative virtualization platform for both new growth and existing workloads. New workloads can be delivered on the alternative platform in an automated fashion with self-service, ensuring that IT can quickly recreate the workloads on future platforms, such as public clouds, without a heavy burden. Existing workloads are analyzed and migrated using a proven methodology to ensure all business requirements are satisfied for the workload, such as migrating within proper change control windows and having approvals from needed stakeholders.

A typical migration journey has three phases:

- **Discovery session.** In this first phase, Red Hat Consulting engages with an organization in a complimentary session to fully understand and document the requirements and scope of the migration journey.
- **Migration pilot.** In this phase, the alternative platform is deployed and operationalized, and several pilot migrations take place to ensure all requirements are met for a large-scale migration.
- **Migration at scale.** In this final phase, the customer migrates workloads at scale following the documented plan that was created in the previous phases with support from Red Hat Customer Experience and Engagement.

Red Hat Subscription Model

Like all Red Hat products, Red Hat Virtualization is available through a subscription model that includes software access, support, patches, and community participation. The standalone Red Hat Virtualization subscription is a single edition that consists of the manager and hypervisor and includes all core enterprise virtualization features built in, including live migration, high availability, user portal, and reports. Red Hat Virtualization is also now a key part of larger Red Hat bundles and suite products in areas including OpenStack private clouds, containers, and platform-as-a-service (PaaS) solutions. Red Hat OpenStack Platform and Red Hat OpenShift Container Platform are also available through an annual subscription model.

Red Hat-Microsoft Interoperability

Even though KVM is a Linux-based virtualization solution, Windows is treated as a first-class guest. Red Hat and Microsoft have entered into an interoperability and support agreement that ensures that customers will be able to use a combination of solutions from the two companies with support from both vendors.

Within the Red Hat Virtualization environment, Windows VMs are able to use the same full scalability features as Linux VMs, such as the number of vCPUs and the amount of vRAM. In addition, Red Hat provides Windows drivers for Red Hat Virtualization, which are conveniently available through the Windows Update service. Red Hat has attained Microsoft's Server Virtualization Validation Program (SVVP) certification, which means that Windows and all Microsoft software are validated for and supported on Red Hat Virtualization.

In addition, Red Hat certifies Red Hat Enterprise Linux and all Red Hat software on Hyper-V. Red Hat Enterprise Linux 6.4 and higher include the Microsoft Hyper-V Linux drivers, which are now in the upstream Linux code. These drivers improve the overall performance of Red Hat Enterprise Linux when running as a guest on Microsoft Hyper-V. Installation support for the Hyper-V paravirtualization drivers enables easy deployment of Red Hat Enterprise Linux as a guest in these environments.

The Red Hat and Microsoft collaboration goes well beyond just virtualization support. The two companies have entered into a broad corporate-level partnership that includes engineering collaboration, certification, and joint support across a number of areas, including Azure, .NET, and management. The most recent area is support for Windows containers within the OpenShift Container Platform. Both Red Hat and Microsoft engineering teams have publicly committed to the project, with a Developer Preview Program launching soon.

CHALLENGES/OPPORTUNITIES

Challenges

- **Expanding beyond the Linux install base.** KVM's close ties with Linux create some challenges because customers perceive KVM-based products as a solution for Linux only, although Windows is well supported. Customers not familiar with Linux may be hesitant to try Red Hat KVM-based products, fearing that they do not have Linux skills and knowledge.
- **Entrenched virtualization competitors.** Even with technologies like CNV, migration of VMs is likely to be from VMware, and traditionally, migrations from VMware have been difficult to pull off for a variety of reasons. While CNV has interesting benefits for running mixed-mode applications, migrating off VMware to KVM and then into a container may be difficult for customers to accomplish at large scale.
- **Building ecosystem and independent software vendor (ISV) support and certification.** The ecosystem is critical in adding value to any system platform and is a force multiplier for market success. Red Hat Virtualization and Red Hat OpenStack are competing with other hypervisors and cloud platforms for partner attention. However, OpenShift has carved out an early position in the container market and has drawn good attention from partners.

Opportunities

- **Cloud, web, and telco providers.** Open source has experienced great early success in the cloud as service providers have liked the customizable code and the low cost. While this opens doors for commercial open source vendors, it has also been historically challenging to convert these service providers to use paid subscription support services for solutions based on open source software. Telcos doing NFV have gravitated toward KVM and OpenStack, a bright spot for those technologies.
- **The reach of Linux.** As an integral part of Linux, the KVM hypervisor exists wherever Linux exists. Linux users already expect KVM virtualization services to always be available to them. The wide and varied distribution model for Linux will spread KVM broadly, with Linux vendors tasked with getting users to embrace it and pay for it.
- **New deployment models and stacks for cloud-native applications.** Technologies such as OpenStack, containers, KVM, and open source are heavily preferred in cloud-native ecosystems. While traditional virtualization is a very mature market, the needs and economics of new emerging cloud stacks are very different, and these new applications have no existing dependencies. KVM has already established key positions in these markets, the de facto hypervisor for anything Linux or open source based. Further, the potential to manage KVM-based virtual machines with the Kubernetes container orchestration platform via Container-Native Virtualization is appealing for these complex cloud-native environments.
- **Open source cloud and containers.** Open source is more accepted than ever and driving much of the innovation today in the industry. Red Hat Virtualization, OpenStack, and OpenShift can capitalize on the growing wave of open source affinity to grow adoption.

CONCLUSION

With the many changes happening to infrastructure in the cloud era, virtualization is still the foundation for infrastructure — tomorrow's and yesterday's. Virtualization is finding new and different roles under next-generation infrastructure, supporting private and public clouds and new compute models such as containers while still maintaining its massive base of traditional infrastructure.

Red Hat Virtualization is a mature open virtualization platform that builds off KVM, the open source standard hypervisor. Red Hat's history of commercializing Linux and other open source software positions the company to be a leader in bringing KVM to enterprises. Red Hat Virtualization is the common foundation across traditional and cloud stacks. Red Hat's KVM-based virtualization technology can be managed through traditional oVirt virtualization management, OpenStack cloud infrastructure, or the very modern OpenShift Container Platform. Red Hat Virtualization can power all these platforms and is highly integrated through common management technologies, such as Red Hat Ansible Tower and Red Hat CloudForms. Red Hat, a trusted vendor in enterprise open source, is well positioned to deliver open source innovation in a fully supported, tested, and integrated stack that can span traditional and cloud scenarios.

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