White Paper

Transforming a Corporate Datacenter into a Modern Environment: Kubernetes as a Foundation for Hybrid Cloud

Sponsored by: IBM

Gary Chen Al Gillen
November 2019

IDC OPINION

Information technology (IT) professionals face a changing world where substantially new technologies emerge every few years, and these technologies come with broad and deep ramifications for the typical corporate compute environment. Most recently, the industry has embraced the notion of a cloud-based deployment model as being the long-term ideal deployment scenario, but the difficulties of moving both applications and data directly from a classic on-premises compute environment, which incorporates a mix of server architectures, operating systems (OSs), and virtualization and platform services, to a cloud-centric deployment environment are not simple and without their challenges.

For many organizations, the practical way to transform from an on-premises computing model to a public cloud-centric platform is to use a hybrid cloud computing model, giving the customer access to both on-premises resources and public cloud resources in a relatively seamless manner. This approach can help bridge the gaps between classic and modern computing architectures and can make it possible for a business to take a series of smaller steps over time — as opposed to a single giant leap — to transition key workloads fully to the public cloud. The good news is that the industry is already on a path toward technologies that offer customers the ability to create a hybrid cloud that incorporates common operational attributes across on-premises and off-premises resources.

The key enablement for hybrid cloud is made possible through the availability of a portable, multiplatform cloud platform. The industry is moving in that direction, with Linux, containers, and Kubernetes as the basis of a universal abstraction layer. Amazon Web Services (AWS), Google, IBM Cloud, Microsoft, Pivotal, Red Hat, and VMware are among the vendors working to develop such a platform, which would make it possible for a customer to use a hybrid mix of servers and clouds, as well as a hybrid mix of locations.

IBM's recent launch of its next generation of the LinuxONE III system extends the capabilities that help enhance the private cloud portion of a hybrid cloud environment through enhanced performance, a new on-chip compression accelerator that boosts throughput of the pervasive encryption capability, and extensive software integration. With the LinuxONE system, Red Hat OpenShift will be a widely available software resource, along with IBM's new Cloud Paks — a set of developer and deployment services — to accelerate the deployment of modern development tools, middleware, artificial intelligence (AI) data services, and management resources.

SITUATION OVERVIEW

For all the efforts in the industry to simplify information technology, each wave of simplification brings additional complexity in that it is additive in nature. This is the case with the transition to cloud computing. Today, IDC sees cloud computing itself as being a collection of closely related technologies, including the following dimensions:

- **On-premises private cloud:** This is a combination of traditional hardware and software that is capable of being used as a traditional enterprise server but also can deliver an experience that closely parallels that of a public cloud environment from a management plane perspective. Ideally, an on-premises private cloud system will have a work-alike behavior to a public cloud resource, meaning there is no meaningful learning curve for a customer to transition to a hybrid cloud deployment scenario.
- **Off-premises public cloud:** This is a third-party cloud environment made available by a hyperscale provider or a smaller cloud services vendor. Examples include Alibaba Cloud, AWS, Google, IBM Cloud, and Microsoft Azure. In addition, public cloud services are offered by software-only solutions, which are deployed on top of multiple hyperscale cloud provider environments. Examples include Pivotal Cloud Foundry (and Cloud Foundry Foundation open source software solutions), Red Hat OpenShift, SUSE Cloud, and VMware Cloud.
- **Hybrid cloud:** This is more of a use case than a fundamentally different technology. Hybrid cloud services offer the ability to operate across multiple clouds, which may include on-premises private clouds and public clouds, or the use of multiple public clouds (often called multicloud), as a common infrastructure that is managed by a common control plane. Most of the public cloud vendors are working on a hybrid cloud strategy, usually without substantial platform diversity. Conversely, some of the software-only solutions, including those from Pivotal, Red Hat, VMware, and SUSE, support both hybrid cloud and multiplatform hybrid cloud deployment scenarios.

While the on-premises portion of a hybrid environment might not have equal scale-out attributes, and some of the public cloud services that may be of interest to developers may not be available locally (although can still be consumed remotely, assuming latency is not a concern), a hybrid deployment scenario that incorporates a private cloud has become an interesting and highly viable alternative. For many customers, the movement to a hybrid cloud that includes a strong private cloud component is the preferred scenario.

One concern that customers have historically faced is portability of applications. However, the industry has made strides in accomplishing that exact objective, including through the following means:

- Growth is seen in the use of interpretative languages, such as Python and JavaScript. Historically, compiled languages were popular because compiler optimization for a given platform maximized performance. Today, with abundant compute resources, the use of interpretive languages no longer presents a performance concern. Modern continuous integration/continuous delivery (CI/CD) pipelines can be configured to create multiple sets of binaries for different deployment scenarios for compiled languages in use.
- Containers are a widely used abstraction layer across different infrastructures, clouds, and system architectures today. Both new and existing applications are being repackaged using containers, whether designed as a set of microservices or not. Containers can now also include multi-architecture support.

- The industry is quickly moving toward Kubernetes as the foundational container management and orchestration layer for next-generation applications, with a number of portable Kubernetes environments available today for both private and public cloud deployment. The availability of consistent Kubernetes platforms on key systems and clouds makes it possible to build a hybrid cloud environment that spans resources in different physical locations.

Not all containers contain software that could be described as greenfield microservices. Realistically, there will be a wide variety of applications housed in containers that in turn will require a broad variety of external services. For example, some containers will contain rehosted monolithic applications, while other containers may have partially refactored applications that are stateful, require high security, and have other attributes reflective of the environment from which they originated. These applications will need systems and infrastructure software that can support their full set of requirements.

Containers and Kubernetes

The notion of a containerized Linux application is not new and was originally made possible under the Linux container (LXC) technology, as part of the Linux operating system. However, the adoption of LXC was very limited until Docker (the company and the technology) emerged and created a compelling and easy way to package applications into a container.

Docker and Open Container Initiative (OCI) containers package an application and all its dependencies into a single portable container image. These images are then shared on a centralized container registry where they can be iterated on by other developers or pushed to production. Container images also use the concept of layers within the image so that developers can easily build on top of existing images. When containers are executed, they run in their own sandbox so that each container is isolated from each other and each container appears to have the entire OS to itself. The developer-friendly tools and APIs made building, sharing, and running containers easy and quickly became popular.

Containerized applications are only the first step, as a way to operate a complex mix of containerized services was needed. Google took the concepts and learnings of its internal "Borg" container orchestration technology and created the open source software Kubernetes project with active participation from IBM, Red Hat, and others. Governance of the Kubernetes project was handed over to the newly formed Cloud Native Compute Foundation (CNCF) to make the project truly open and available to the IT industry. The industry subsequently embraced Kubernetes as a preferred next-generation infrastructure and as a deployment and orchestration platform for containerized applications.

Containers are gaining tremendous traction as a portable, universal application packaging standard. One reason for this is that containers are standardized as part of the Open Containers Initiative. OCI defines how containers should be executed (runtime) and how the container image should be formatted. Standardization also means that containers are consistent, interoperable, and portable between various container platforms and cloud services. Standardization also extends further up the stack with Kubernetes for orchestration and management.

While Kubernetes is not a codified standard like OCI, it is an independently governed open source project with very broad industry participation. So not only do containers have a standardized format, a great deal of the management stack is consistent across implementations. Other components such as the Istio service mesh and Knative for serverless are also starting to gain traction and may be ubiquitous one day as well, providing an even more common stack. However, while standardization

accelerated container adoption, the primary reason why containers drew excitement in the first place was for the speed containers provided to developers both in development life cycles and for DevOps-enabled deployments.

For developers, containers are a perfect fit to efficiently encapsulate new cloud-native microservices and to push these changes down increasingly automated software build pipelines using CI/CD. The developer-friendly APIs made working with complex software faster and more convenient, improving developer workflows. Containers can also improve code quality by helping enable more automated test systems and better environmental control as containers typically include all dependencies. Ultimately, containers enable faster development of software, faster deployment of changes, and more developer productivity.

For operators, containers and Kubernetes offer a highly modern, scalable, and automated way of running large web-scale applications. Kubernetes embeds much of the knowledge and experience of the largest web companies in running fast-changing apps reliably at very large scale. It has established deployment patterns such as blue/green upgrades, A/B testing for new app features, and multiple automated scaling options. In addition, container-style deployments help solve configuration headaches by leveraging immutable infrastructure. This means that a container state is defined in its image, and it never changes during runtime. If a change is desired, the old container instances are brought down and a new image is started instead of making patches or configuration changes to a running image. Container repositories also help centralize container images and maintain versioning. The lightweight and reactive nature of containers, combined with a modern control plane, enables IT to efficiently deploy and manage modern applications.

As customers move to hybrid cloud and multicloud, containers can play a key role in portability and consistency across different environments. As previously discussed, the broad adoption of the OCI format and the Kubernetes control plane by the IT industry inherently makes container platforms similar at their core. The CNCF provides a Kubernetes conformance test and certification, which means that all Kubernetes must behave the same for core functions. This means that customers can largely use any Kubernetes product and expect a certain level of compatibility.

Developers can largely work with containers and Kubernetes using the exact same APIs and tooling that they wish, no matter the distribution or cloud service or infrastructure underneath. This can help provide a consistent developer environment across on-premises and various public clouds. In addition, containers can help abstract across different system architectures as the container interfaces remain the same regardless of the hardware underneath and the developer does not have to know the ins and outs of the system or OS to develop applications for it. For operators who manage Kubernetes, there are some system-specific knowledge and deployment tools they will have to install and integrate, but operating Kubernetes and the apps managed by it remains largely consistent across any distro.

MULTI-ARCHITECTURE CONTAINERS AND HYBRID CLOUDS

Containers make it easier to abstract away the differences of underlying systems for developers. However, an enterprise platform like IBM LinuxONE can offer additional benefits to container operations and applications, and container developers don't have to learn many new skills to take advantage of them.

The LinuxONE contains a firmware-based hypervisor that works in conjunction with the z/VM or KVM software hypervisor to provide secure and flexible compute, memory, and I/O provisioning. Today,

most containers are run in virtual machines (VMs) rather than on a bare metal orchestration environment. While these technologies seem to overlap, they mostly work at very different levels. Hypervisors virtualize and partition hardware, while containers virtualize the OS.

Running containers in VMs offers multiple benefits:

- Allows an additional layer of isolation and separation, as the VM boundary is much stronger than the container boundary
- Carves up today's large hardware system into more consumable chunks (This can also improve security and reliability by not consolidating a massive number of containers on a single OS kernel.)
- Allows more flexible mixing of different operating systems or even different version/patch levels of the same operating system

The unique firmware-based hypervisor within the IBM LinuxONE platform can help Kubernetes with scaling. IBM LinuxONE can both scale up and scale out in a nondisruptive way with the hypervisor. For example, this complements Kubernetes' vertical pod autoscaling feature that helps rightsize CPU and memory provisioning for containers.

As enterprises transition to microservices architecture, much of what is containerized today is traditional legacy apps that may or may not be refactored. Containerizing existing applications still brings benefits, and enterprises are doing this widely today, with about half of containerization attached to legacy applications. The flexibility and reliability of LinuxONE and its hypervisors also make this platform suited to host large monolithic containers during this transition.

IBM's latest release of the LinuxONE system offers the developers and administrators new software solutions in easy-to-consume formats called IBM Cloud Paks. Enterprise ready and delivered in a container, IBM Cloud Paks offer developer tools and data and AI services, along with open source middleware software, and run on Red Hat OpenShift Cloud Platform.

IBM also offers IBM Cloud Hyper Protect Services, a portfolio of public cloud services built on IBM Secure Service Container technology to enable developers to easily build applications with highly sensitive data. These include:

- **IBM Cloud Hyper Protect Crypto Services:** Allows customers to keep their own keys for cloud data encryption through customer-controlled Hardware Security Modules
- **IBM Cloud Hyper Protect DBaaS:** Provides highly secure and easy-to-use enterprise cloud database environments for complete data confidentiality in the public cloud
- **IBM Cloud Hyper Protect Virtual Servers:** Delivers complete authority over customer-managed virtual servers for sensitive workloads

While containers bring many security benefits through features such as immutable infrastructure and centralized image repositories, it is still a new layer in the stack to be dealt with, and security is a major challenge for adopters. LinuxONE offers many security features that can be exploited by containers. The overall design of the LinuxONE platform is geared toward security from hardware to software. The IBM cryptographic coprocessor adapter achieves the highest Linux security certification with an HSM providing FIPS 140-2 Level 4 compliance. LinuxONE also provides a feature called IBM Secure Service Containers, a type of highly secure logical partition that delivers a secure application execution environment for containerized applications:

- Tamper-resistant, trusted boot sequence in firmware with isolated memory.
- Restricted administrator access that helps prevent misuse of privileged credentials, to provide data protection in use
- Transparent, automatic, and overhead-free encryption of all data at rest and in flight
- Reliability and dramatic scale-up capabilities
- Colocation with existing data/applications

Application Portability

In the past, applications were tied not only to a specific operating system but to a specific operating system on a specific server because of optimizations, configurations, and tuning, along with a select suite of middleware and data management software products. Widespread use of virtualization on x86 servers allowed that full stack — operating system, application, and all relevant dependencies — to become more portable from one virtualized server to another. While that was an important step in moving toward today's compute environment, applications ultimately needed to be abstracted from the specific operating system instance.

Successfully accomplishing such an abstraction requires that an application and its dependencies are collectively abstracted from the specific instance of the underlying operating system. Successfully achieving this goal also has the benefit of reducing the overhead associated with a one VM and one operating system-per-workload deployment model.

Platform independence is further enhanced by the use of interpreted languages, through common language choices such as JavaScript, Perl, and Python, as well as through bytecode-compiled languages such as Java. Interpreted languages offer the benefit of being interpreted on a just-in-time basis and, as such, are interpreted by the platform on which they run, eliminating any concerns of bytecode ordering — which is one of the differentiations between LinuxONE family (big endian) and x86 solutions (little endian).

By comparison, widely used compiled languages such as C, C++, Go, and Haskell are precompiled prior to execution. This means the compiler will create binary code for the platform to which the application is destined to be deployed. This means code complied for x86 Linux platforms will not be executable on Linux on an IBM LinuxONE system, even on a software cloud platform such as OpenShift Container Platform.

To address that issue, most modern continuous integration/continuous delivery systems can manage multiple sets of binaries compiled from common source code and deploy the correct binary to the appropriate platform. Cross-compilers for Linux are available that make it possible to compile a Z Linux binary on an x86 Linux system. In addition, OCI container images have multi-architecture capabilities,

where a single image can contain binaries for multiple system types instead of having to juggle multiple images.

FUTURE OUTLOOK

Industry Direction

Information technology has a tendency to over-rotate, at least in terms of enthusiasm and embrace, for exciting emerging technologies. The reality, though, is that any technology has support implications that potentially extend out for decades. As such, customers should look toward embracing technologies that are likely to offer the best long-term choices for portability, flexibility, and supportability.

Because the industry has settled on Kubernetes for the container control plane, this becomes the nexus for innovation and integration, leading to multiple benefits. First, Kubernetes itself will benefit from a massive community that is focused on one platform. Second, there is a growing innovation of projects that are not a part of Kubernetes per se, but integrate with and optimize for Kubernetes.

For example, the Istio service mesh and Knative for serverless computing build directly on Kubernetes technology. These projects are serving to foster a collectively large community and will become an essential part of Kubernetes deployments, creating a broader, more common container platform. Having more common components across various Kubernetes distributions and cloud services will benefit users as they increasingly become more hybrid and multicloud based.

IBM LinuxONE directly participates in this community in a variety of ways, including:

- **Red Hat OpenShift Container Platform on IBM LinuxONE:** One of the real diamonds that IBM wanted from Red Hat's portfolio was the OpenShift Kubernetes Platform. This technology, which can be used as the foundation for a hybrid cloud, has the ability to span multiple hyperscale cloud environments including Amazon Web Services, Google Cloud Platform, and Microsoft Azure and offers support for Istio and Knative. Support for OpenShift is done in conjunction with Red Hat Enterprise Linux, which has been available on LinuxONE servers for years.
- **IBM Cloud Paks on LinuxONE:** In early August, IBM announced the notion of "Cloud Paks" for OpenShift. These containerized offerings include separate products intended to support data and artificial intelligence (Cloud Pak for Data), application modernization and cloud-native application development support (Cloud Pak for Applications), enterprise application integration (Cloud Pak for Integration), business process and decision-making automation (Cloud Pak for Automation), and one specifically designed for multicloud management (Cloud Pak for Multicloud Management).
- **IBM Cloud Private on LinuxONE:** IBM's existing IBM Cloud Private technology will continue to see support and investment protection from IBM in parallel with the new technologies that entered IBM's portfolio through the Red Hat acquisition.
- **Linux distributions:** They are available for LinuxONE, including Canonical Ubuntu, Red Hat Enterprise Linux, and SUSE Linux Enterprise Server. In addition, several open source community distributions are available for customers that are interested in using them for noncritical workloads. Distributions with lightweight kernels such as Alpine and Red Hat Enterprise Linux CoreOS are also supported for container use.

CHALLENGES/OPPORTUNITIES

Challenge: Containers abstract many things from developers, allowing them to work the same way with the same APIs, regardless of the system underneath. However, the underlying systems still have a significant impact on how containers are executed, scaled, and secured. The challenge today for systems vendors is messaging these differences and benefits to an audience that is increasingly viewing infrastructure as a commodity.

Opportunity: Containers open IBM LinuxONE systems to a new modern audience: the cloud-native developer. These developers can develop for and deploy to LinuxONE systems in the same way they do any other infrastructure with containers. This can bring LinuxONE development into modern workflows and to participate in container deployments, without having to train developers specifically for the platform. However, there will be a learning curve for Z system operators, as they will be tasked with deploying and integrating container platforms there.

Challenge: Most developers have deep experience in developing for x86 environments, less so for alternative architectures.

Opportunity: Interpreted languages offer better portability; compiled languages are increasingly able to support multiple compile scenarios, cross compilers, and multi-architecture containers. Today, the use of a variety of non-x86 processor types, including GPUs, ASICs, custom processors optimized for artificial intelligence, ARM processors, and other technologies like Raspberry Pi at the edge, means developers are employing tools to help them support a heterogeneous deployment environment — where Kubernetes is the common denominator.

Challenge: IBM LinuxONE is lumped into the traditional IT infrastructure segment.

Opportunity: IBM LinuxONE systems offer scalability, security, and reliability far beyond commodity platforms and are a valuable dimension to a larger multicloud deployment opportunity. Also, there is a growing ecosystem of hundreds of open source tools and software already available on the LinuxONE. While developers are largely insulated from the underlying system with containers and special skills are not required, some awareness of these capabilities is still crucial to understand what type of applications might be possible in containers given various requirements. The challenge for systems vendors and customers in site reliability engineer roles will be how to surface the value and differences of the underlying system through the container platform in a way that IBM LinuxONE can capture containers that are well suited to its capabilities.

Challenge: Customers that are trying to move to off-premises solutions won't want a large system in their datacenter.

Opportunity: IBM offers the IBM Cloud Hyper Protect Services powered by LinuxONE running in the IBM Cloud. IBM's adoption of the industry-standard 19in. frame for its new LinuxONE III system also enables the system to easily fit into a standard datacenter.

CONCLUSION

IT executives are facing an era of accelerated innovation, and to support the increased demand, they need to have a viable plan for a hybrid cloud and multicloud infrastructure that will support their specific application needs and provide scale, security and, equally important, portability to other

dimensions of the hybrid cloud environment. The standardization of infrastructure and deployment software layers means that customers can — and should — focus more of their resources on creating differentiation for their applications, including optimizing user experience, support for multiple user devices, and functionality/completeness of the application itself. Of course, this assumes differentiation would also include core attributes such as reliability, scalability, security, and cost of operation.

IBM has brought innovative new platforms to the industry that offer many of these software and deployment experiences and attributes that today's developers want and need. These services are largely abstracted from the underlying platform, enabled in particular by the Red Hat OpenShift Container Platform and a set of secure and scalable container deployment services on IBM's platforms, opening the IBM LinuxONE to a broader potential audience that is looking for behavior attributes and outcomes from its IT investments.

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