WHITE PAPER

Kubernetes on Hitachi Unified Compute Platform (UCP)

Containerized Workload Orchestration on Converged Infrastructure

By: Hitachi Data Systems

August 2015

Contents

Executive Summary 2

Introduction 3

The Container Market 5

The Solution 5

Workload Migration 8

Site Replication 10

Next Steps 11

Executive Summary

Hitachi Unified Compute Platform (UCP) is a converged infrastructure system that integrates compute, storage and network resources for the deployment of enterprise-class bare-metal or virtual workloads. Kubernetes is a container-orchestration open source solution that handles scheduling in a compute stack and actively manages containerized workloads. Using Kubernetes alongside Hitachi Unified Compute Platform Director suite of automation software for virtualized environments enables the management and sharing of physical and virtual infrastructure resources. It enables an IT administrator to maximize their environment inside the UCP compute stack by running an elastically scalable container cluster alongside their existing workloads all managed with a single unified control plane.

This white paper presents a summary of Kubernetes and UCP system architecture, integration points and workload benefits.

Introduction

Combining the infrastructure orchestration expertise of Hitachi Unified Compute Platform Director (UCP Director) with the power of Kubernetes brings out more resource potential for workloads and helps make developers more agile. Using Kubernetes to run modern applications and services alongside the enterprise-proven UCP Director allows innovative customers to migrate to and orchestrate a containerized system. This combined environment provides three main benefits. UCP Director: (1) simplifies and hides the infrastructure configuration for running Kubernetes, allowing workloads to operate with high reliability and availability; (2) provides an enterprise-level platform for Kubernetes to run flexibly and scale up to thousands of resources and nodes; and (3) allows an existing traditional bare-metal or virtualized workload running on UCP to coexist with a container-orchestrated Kubernetes environment. It reduces costs, removes error-prone deployment steps, simplifies scheduling logic, and decouples application dependencies, therefore accelerating software development for businesses.

The current market trend is containerization, pioneered by Google, adopted by large web-scale tech companies, and brought to the masses by Docker. When developing software composed of more than one container, it became apparent that an orchestration tool is necessary. Kubernetes originated from Google Borg, which is proven across Google's services. Containerization was previously a very complex technology with a high barrier for entry to many developers. The trend to containerize workloads and applications is rapidly gaining traction and disrupting the market. This disruption can be compared to the computing market with virtualization, where the way that computing was built and thought of changed completely. This lead companies and organizations to grow in ways previously inaccessible. The containerization disruption is more rapid due to the traction from both the engineering and business focused organizations. Many businesses are embracing the containerization movement, from Tutum, a container-as-a-service company, all the way to ING Bank, a global enterprise financial company.

There are two major hurdles to the traditional method of bringing up containerized workloads without UCP Director or Kubernetes: rapidly provisioning your infrastructure to prepare it for running workloads, as well as orchestrating containers once they are deployed.

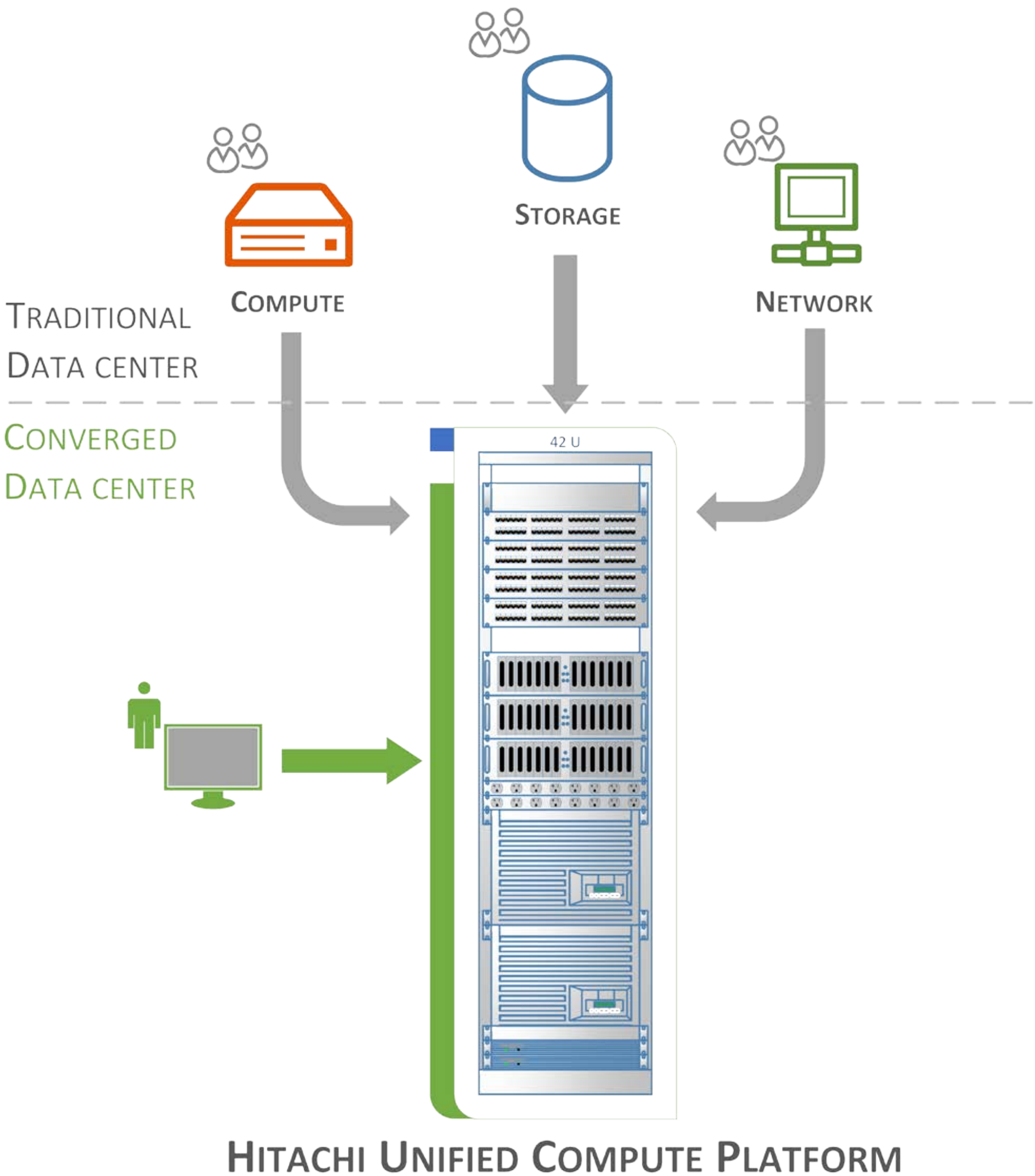
The first major hurdle in a traditional data center, is provisioning the data center infrastructure to prepare it to run workloads. Traditional IT administrators within data centers typically have four distinct teams to manage their resources;

- A storage team of subject matter experts manages and provisions the storage arrays containing their compute-side data.
- A networking team to manage switching for their infrastructure and compute workloads.
- A Fibre Channel team to manage data paths between their hosts and arrays
- A platform team to orchestrate their highly available workloads within virtual machines.

The handoffs between all these teams as well as the expertise in the subject matter for each, lends to high overhead, high operating expenditure (opex) returned to the company, and a nondeterministic process to provision an infrastructure.

Hitachi UCP solves this problem as a converged infrastructure system that allows automatic provisioning and management of infrastructure needs. Hitachi UCP Director automation leverages a feature-rich enterprise suite of tools to provision storage, networking and Fibre Channel resources to reduce time to value, directly on-premises (see Figure 1).

Figure 1. UCP Director converged infrastructure simplifies a traditional data center by orchestrating compute, storage and networking.



The second major hurdle is working with containers and coordinating the orchestration of containerized applications. Outside of basic scenarios and applications, scheduling, monitoring and scaling containers is very complex and elaborate, thus requiring an orchestrator. A current leader in this area is the Kubernetes project created by Google. The project is based on Googles' Borg container system that has been refined for over a decade.

Kubernetes is leading the technology development and making it easier for application developers to build distributed applications using the experience gained from Google Borg. Kubernetes has several technical benefits, such as scheduling, health monitoring, declarative application management, modularity and portability to accelerate development and simplify operations. Combining UCP Director with Kubernetes enables an orchestrated, monitored, flexible, containerized workload alongside other traditional workloads in UCP.

The Container Market

Why are containers the direction the community is backing, and why should they be considered on the road map for IT organizations? IT organizations will benefit from containers due to maximizing resource utilization by reducing duplication of application dependencies, self-contained images allow more predictable and portable applications, all of which increases the speed of which software developers build and ship code. Traditionally, large companies had to support an aging monolithic infrastructure and cannot afford to continue building production applications at normal speeds. By using containers, developers are able to observe, orient, decide and act on application development at a much faster and more reliable pace.

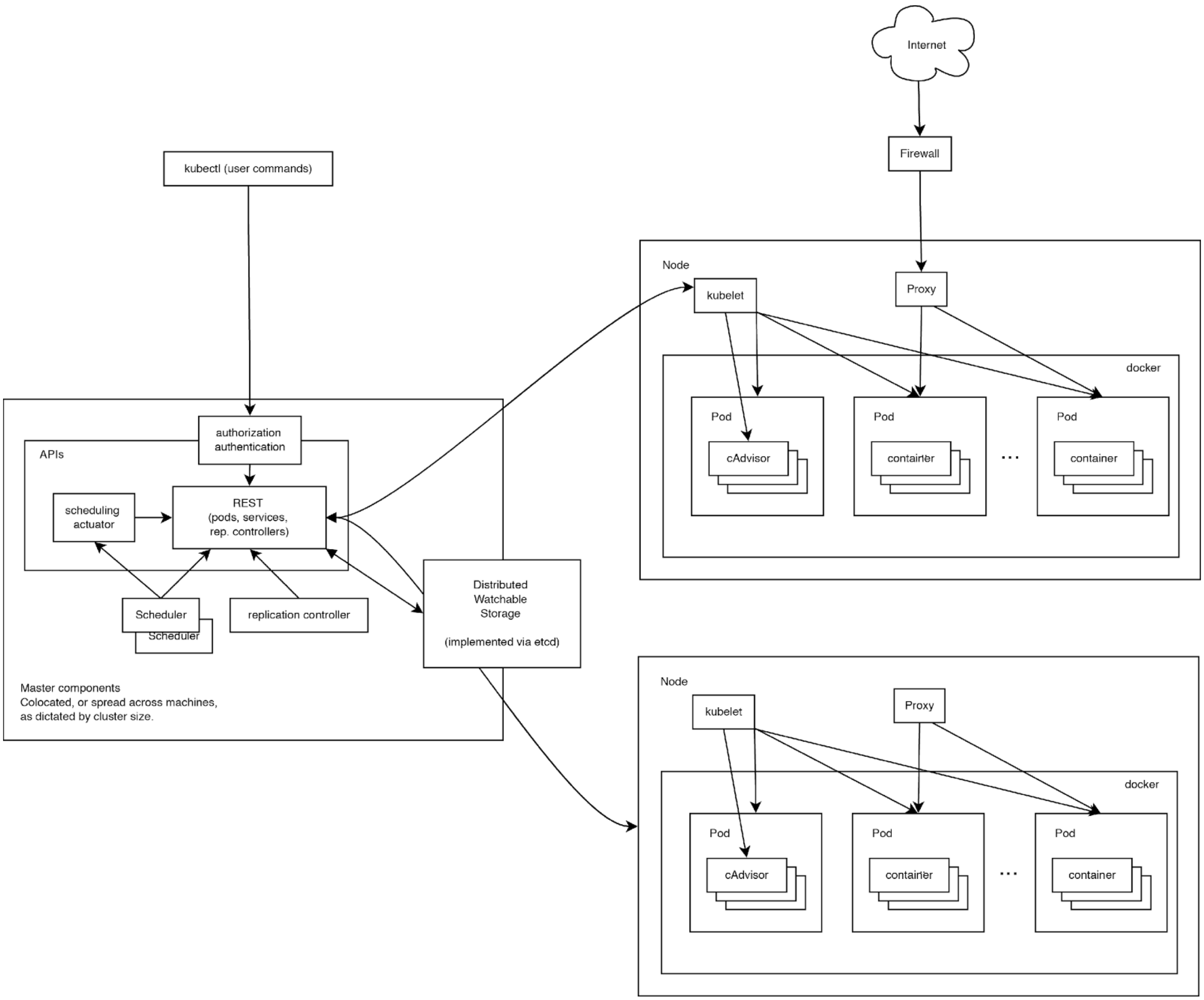
With containerization of production tools and applications, large enterprise companies can now move with higher speed and agility. The IT department also benefits because they don't need to replicate the developers' specific environmental needs. Instead, developers can just ask for a Kubernetes cluster to run their container applications. Developers have much faster feedback loops due to container images running in production exactly how they ran in development and test environments.

The Solution

The solution of running and managing containers in a converged ecosystem is to run Kubernetes on Hitachi Unified Compute Platform. UCP Director deploys and configures the underlying hardware and infrastructure, while Kubernetes is installed on that infrastructure, whether it is bare-metal servers or virtual machines. The Kubernetes architecture is a master node topology that allows a system to become part of the Kubernetes cluster as shown in Figure 2.

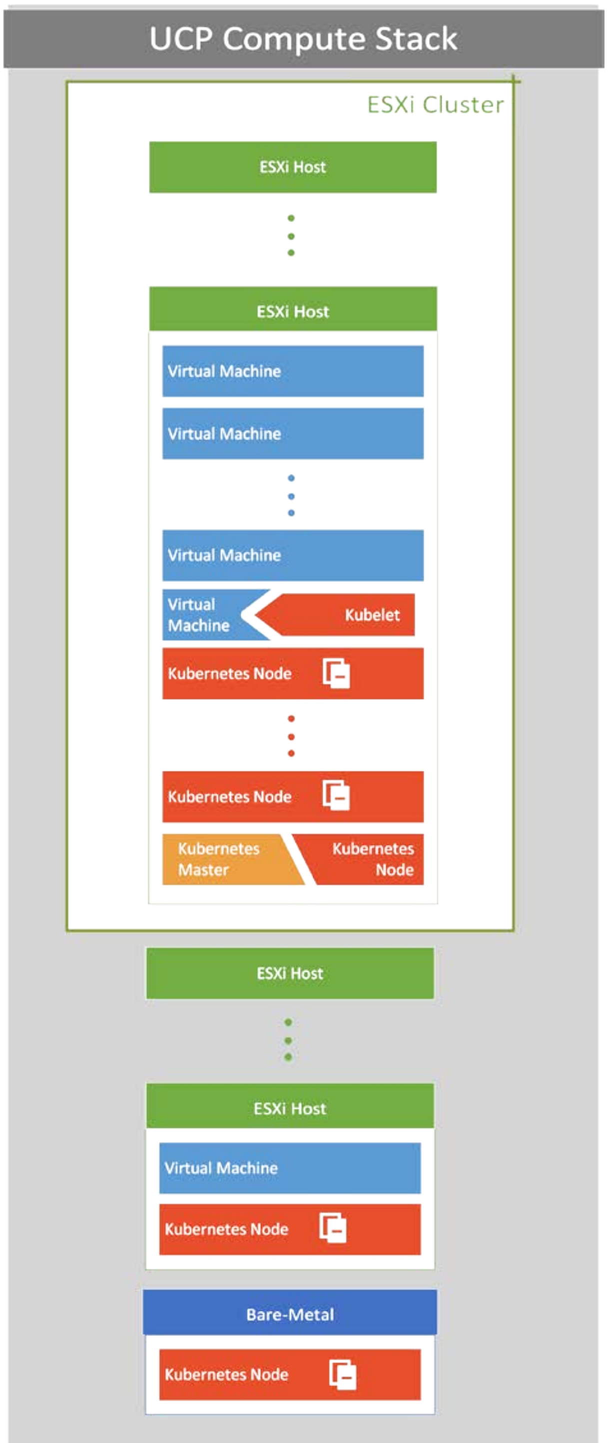
As seen from the architecture in Figure 2, the master components of the Kubernetes system need to be available and reliable for production needs, due to the fact it is a central point of communication. This can be done in many ways, but for the context of this paper the VMware high availability feature is the high availability solution. UCP Director Installs the VMware and/or bare-metal infrastructure that will be used to run Kubernetes master and nodes. To deploy this infrastructure, UCP Director orchestrates storage provisioning, deploy a VMware cluster via its service template cluster deploy feature, and provide networking connectivity by configuring VLANs on the associated switches. Based on Figure 3, the architecture includes deploying all components to the UCP compute stack. This cluster is configured for high availability to guarantee the Kubernetes master is highly available. When using UCP Director to provision VMware clusters, by default it will create a Virtual Dedicated Server (VDS) with the network configuration needed for this architecture.

Figure 2. Kubernetes architecture diagram illustrates the master node relationship.



One of the major benefits of using containers orchestrated by Kubernetes is its ability to manage and allocate resources on a host or cluster dynamically, to best utilize the resources available. The Kubernetes nodes are going to "fill the gaps" in the infrastructure managed by UCP Director. Figure 3 shows how containerized workloads can coexist and append to an existing environment without disrupting the current production workloads (typically run within virtual machines or VMs). The Kubernetes architecture allows the configuration and use of resources in the UCP compute cluster that would otherwise be underutilized. For Kubernetes to function on a node it deploys a process called a Kubelet. These Kubelets can be run on operating systems deployed to a virtual or bare-metal machine.

Figure 3. Flexible architecture: Kubernetes nodes can flexibly run within ESXi Clusters, ESXi standalone hosts, or bare-metal servers.



The Kubelet component can be deployed inside a pre-existing operating system that is capable of running Docker. An example of this is shown in Figure 3, where the virtual machine that is already running in your system can run a Kubelet process inline. Kubernetes allows the definition of resources with labels, enabling the user to select and control where the defined resources can be run. In the example of running a Kubelet inside a VM that already has workloads on it, the container can be limited so it will not consume more than is defined. Labeling also allows running pods on different tiers of configurations or hardware. For example, a set of production nodes with a higher priority label will allow Kubernetes to select and manipulate pods and services related to the label priority. This becomes extremely valuable as users scale out their containerized application to meet their dynamic workload demands. Kubernetes can orchestrate deployment of workloads on blades based on labeling to assure all cpu, memory and capacity resources are being utilized correctly and efficiently. As of today, Google recommends Kubernetes run on top of a hypervisor. This is for security reasons dealing with containerization technology and security boundaries. With the flexible architecture in Figure 3, the end user has the option to have a performant workload run on a bare-metal system, or to run securely within a hypervisor. The feature set for Kubernetes goes beyond the basics discussed in this paper, such as persistent storage, load balancing, auto-replication, rolling upgrades and a more advanced topic of hybrid clouds systems.

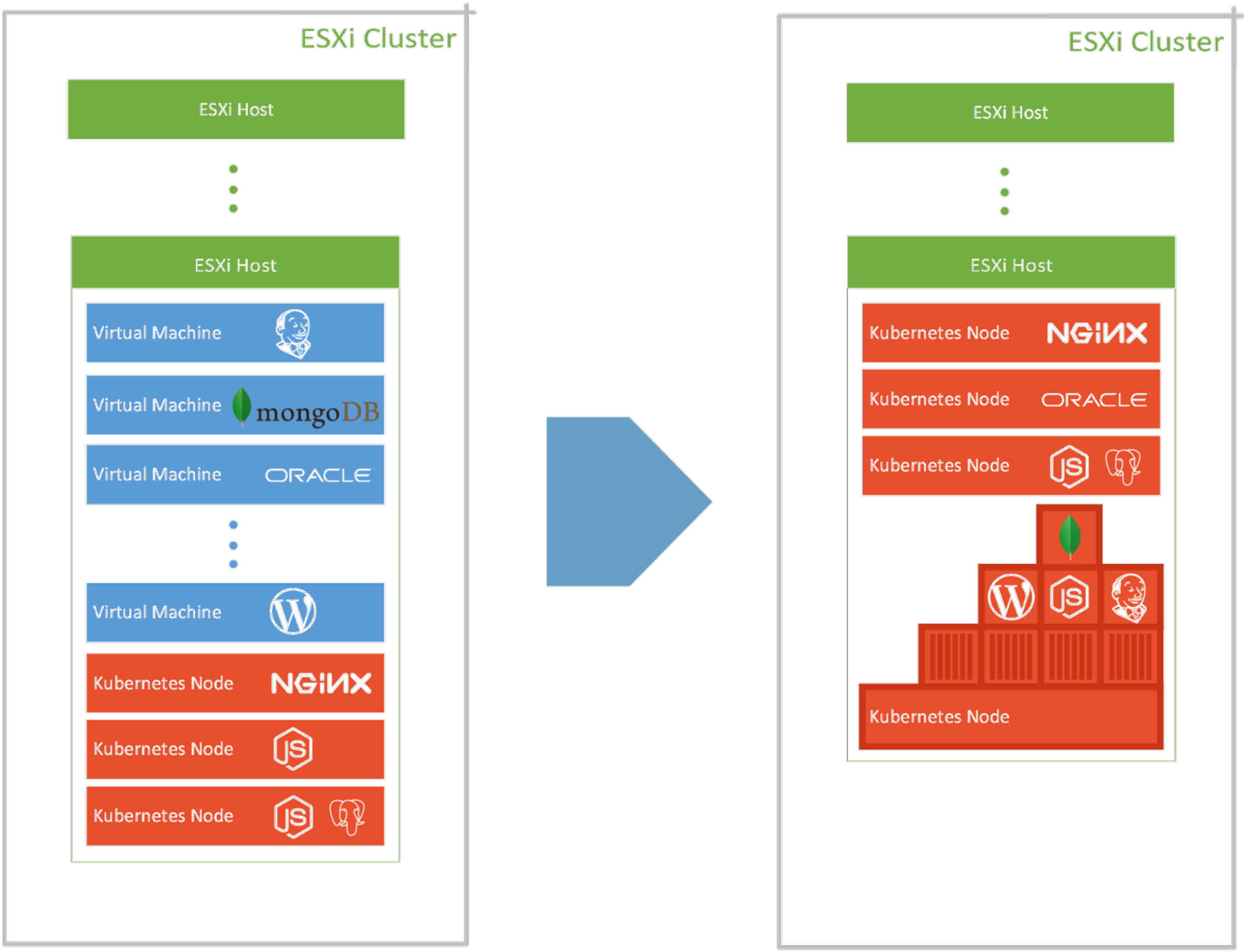
Another benefit of joining Kubernetes and UCP is the ability to leverage both API interfaces to extend enterprise application services. This provides a unique enterprise experience to developers, allowing them to automate orchestration of infrastructure and containerized applications. Developers can now declaratively and programmatically define the environments required and how they are run from the metal all the way up to the application layer. An example of this is programmatically creating and attaching infrastructure storage then leveraging Kubernetes to define and utilize persistent storage to containers in a pod. The ability to leverage Kubernetes' extensibility also enables developers to create their ideal ecosystem to meet their container orchestration requirements.

Workload Migration

The third major benefit is that existing enterprise applications running on UCP can be augmented or migrated into the Kubernetes application service model, over time, in a co-existent architecture. When maintaining production applications or processes that have dependencies on the system they are running on, end users can feel locked into their environment and will not have the agility they desire. Containerization allows maintaining and distributing applications without being burdened by environmental constraints. Kubernetes on UCP provides the platform for augmenting or migrating applications and processes over to containers. One strategy to migrate requires developers to decompose and write applications for containerized architectures. Figure 4 shows the end result of a migrated architecture, eliminating duplicate dependencies and allowing code to be built and shipped faster. Once applications have partially or totally migrated, Kubernetes will orchestrate containers to increase reliability, facilitate scaling workloads, and be resilient to single component failures. It is also possible to augment existing workloads to take advantage of containerization, such as building in new capabilities (that is, logging, monitoring, modifying integration endpoints) within a container.

The migration to containers can be done while co-existing with noncontainer workloads or services. This allows newly containerized workloads to scale with proper configuration and planning of resource requirements. Kubernetes will orchestrate deployment, scale and monitoring of those containerized services.

Figure 4. Kubernetes provides a migration strategy to run containerized workloads.



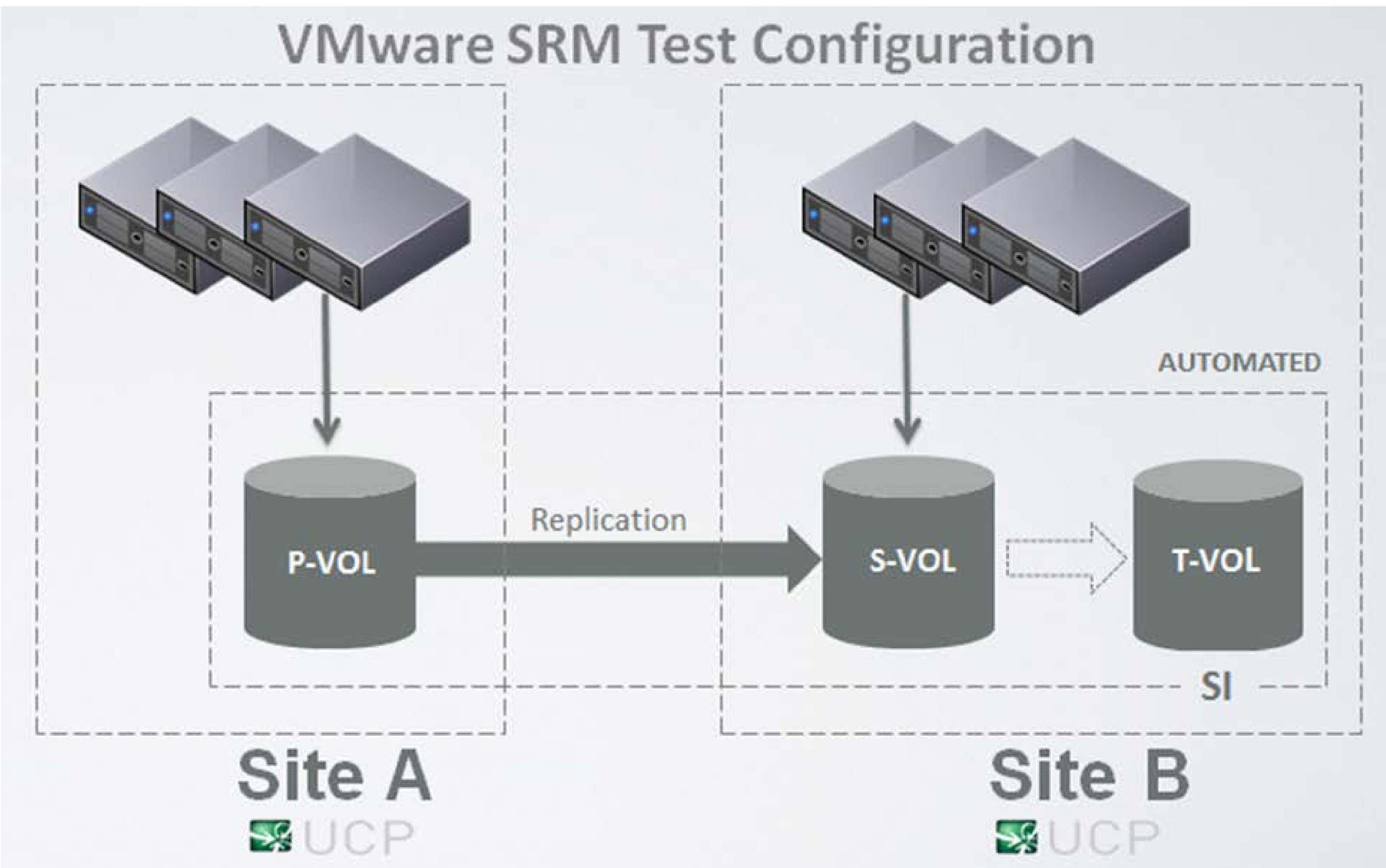
In an augmented, migrated, or newly deployed system with UCP and Kubernetes, the end user will enjoy reduced capital expenditure (capex) and opex costs. UCP greatly reduces opex as a turnkey converged system to deploy and provision data-center infrastructures, and containerizing can reduce capex due to the scaling on resources that were previously unachievable. As mentioned previously, opex reductions are realized with UCP by removing handoffs between specialized enterprise organizations to provision a deterministic infrastructure. In addition, cost savings will be achieved when running the same set of applications on far less physical space. When running on a container-based system an organization could also find cost savings through reduced licensing costs on virtual machines required to run their workloads. For Kubernetes to reduce costs on the end user, they must follow best practices when containerizing their workload.

Site Replication

Hitachi UCP Director also provides the site replication capability, to enable recovery and continuation of vital technology infrastructure in the case of disaster. With complete site replication it is possible to have a Kubernetes ecosystem fully redundant with UCP's disaster recovery feature. UCP Director disaster recovery supports both synchronous (Hitachi TrueCopy) and asynchronous (Hitachi Universal Replicator) replication modes, as well as adjustable Maximum Outage Tolerance (MOTs) to define recovery points. In addition, UCP Director supports test failovers, test failbacks, creating recovery plans, and dynamically allocating T-Vols in a novel simplified interface to support any workload requirements for recovery.

Illustrated by Figure 5, the running workload can be secure from failures. Containers have the connotation of expecting failures in services and processes but cannot sustain a disaster causing a site to go down. When an entire site goes offline in this architecture, running services and workloads can be recovered at a remote site. Site replication is a significant asset when designing and planning an enterprise-level containerized system to ensure critical virtual machine, bare-metal, and Kubernetes containerized workloads can be recovered.

Figure 5. This example of Hitachi UCP site replication ensures that Kubernetes and production workloads can be recovered.



UCP = Hitachi Unified Compute Platform

Next Steps

When you have made the decision to deploy a Kubernetes architecture an environment, use UCP Director to provision the storage, networking, host and resources needed. UCP Director provides a single pane of glass to easily provision these resources on-premises, and it is then used in the Kubernetes setup workflow.

Hitachi has a community-supported path to get Kubernetes running on a UCP system, running CentOS 7 with Kubernetes master and node components. To deploy the Kubernetes system use the Hitachi supported setup guide found here in the HDS community [1]. These scripts will install a framework of Kubernetes in the UCP compute cluster. For reference, there are other Kubernetes setup guides outside of the purview of this white paper at the open source repository found here. To order or learn more about a Hitachi UCP converged infrastructure system, please contact your Hitachi Data Systems sales representative: http://www.hds.com/contact-sales/.

Hitachi Data Systems

Corporate Headquarters
2845 Lafayette Street
Santa Clara, CA 95050-2639 USA
www.HDS.com **community.HDS.com**

Regional Contact Information
Americas: 866 374 5822 or info@hds.com
Europe, Middle East and Africa: +44 (0) 1753 618000 or info.emea@hds.com
Asia Pacific: +852 3189 7900 or hds.marketing.apac@hds.com

WP-541-A HDS August 2015