NetApp Verified Architecture

Red Hat OpenShift Container Platform with NetApp HCI

NVA Design

Amit Borulkar, NetApp
October 2018 | NVA-1124-DESIGN | Version 1.0

Abstract

Red Hat OpenShift Container Platform with NetApp® HCI is a prevalidated, best-practice data center architecture for deploying microservices workloads at an enterprise scale. This document describes the architectural design and best practices for deploying the solution at production scale in a reliable and risk-free manner.

In partnership with

redhat.

NetApp®

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Executive Summary

To meet the growing customer demands and requests for new features, enterprises are adopting DevOps methodologies that allow an organization to operate in an agile manner. Processes such as continuous integration and continuous delivery result in frequent release cycles, which deliver the latest features and updates to customers in a very short time span. Containers have played a critical role in the success of DevOps. Containers package an application and its dependencies together, which abstract the underlying platform and allow an application to run anywhere. However, practicing DevOps at an enterprise scale has its own technical and cultural challenges. Even a minimum amount of downtime (seconds) can result in significant losses both financially and business-trust wise. An environment should be monitored for resource availability (compute, network, and storage) and application performance should be guaranteed. All the components in the stack, such as physical infrastructure, virtualization layers, and operating systems, should be updated with the latest security patches.

To address these challenges, NetApp and Red Hat partnered to provide a turnkey platform for running DevOps workloads in a reliable manner. NetApp HCI provides an intuitive, API-driven, programmable agile platform with enterprise-class features such as storage efficiencies and self-healing capabilities for complete high availability (HA) and guaranteed performance. NetApp HCI is based on VMware virtualization, which provides decades of virtualization innovation and enterprise-hardened features through a push-button deployment. Red Hat OpenShift Container Platform provides enterprise Kubernetes bundled CI/CD pipelines, automated builds, and deploy, which enables developers to focus on application logic while leveraging all best-of-class enterprise infrastructure.

2 Program Summary

2.1 NetApp Verified Architecture

Red Hat OpenShift Container Platform with NetApp HCI is a prevalidated, best-practice data center architecture for deploying microservices workloads at an enterprise scale. This document describes the enterprise requirements for deploying containerized applications and services, the various design choices and technical requirements to achieve a flexible, predictable, and reliable infrastructure that scales independently with your application demands. This architecture described in this document is codesigned and coengineered by subject-matter-experts (SMEs) from within NetApp and Red Hat to provide the advantages of open-source innovation with enterprise robustness.

NetApp HCI provides the widely recognized benefits of hyper converged solutions including lower TCO, ease of purchasing, deployment, growth and management for virtualized workloads. However, NetApp HCI is different because allows IT to scale storage and compute separately. This can be an enormous source of wasted resources because HCI scales to meet enterprise needs. NetApp HCI is based on the VMware platform, which is the enterprise standard for private cloud management. Red Hat OpenShift Container Platform is a comprehensive enterprise-grade Platform-as-a-Service (PaaS), based on Kubernetes. Red Hat OpenShift Container Platform is frequently deployed within a VMware virtualized infrastructure to be able to benefit from the native HA of VMware and dynamic resizing of VM resources that leads to greater security, efficiency, and agility.

Together these three technology solutions provide for ease of procurement, deployment, and ongoing management and growth of your business-critical hardware, virtual resources, and enterprise applications. They use a modern container-based application deployment model but leverage the common tools of the modern data center which preserves investments.

3 Solution Overview

3.1 Business Challenges

Enterprises are increasingly adopting DevOps practices to create new products, shortening the release cycles and rapidly adding new features. Because of their innate agile nature, containers and microservices play a crucial role in supporting DevOps practices. However, practicing DevOps at a production scale in an enterprise environment presents its own challenges and imposes certain requirements on the underlying infrastructure; for example:

- HA at all layers in the stack
- Nondisruptive operations and upgrades
- API-driven and programmable infrastructure to keep up with microservices agility
- Multitenancy with performance guarantees
- Ability to run virtualized and containerized workloads simultaneously
- Ability to scale infrastructure independently based on workload demands

3.2 Red Hat OpenShift Container Platform with NetApp HCI

To meet these business challenges, NetApp and Red Hat coengineered, designed, and validated a reference architecture that allows you to develop and to deploy an application in a reliable and risk-free manner. Red Hat OpenShift Container Platform with NetApp HCI enables you to run containerized applications by using enterprise-grade Kubernetes for container orchestration with performance guarantees. Further, the scale-out design of the solution allows you to start small and to scale out compute and storage resources independently of one another to meet your workload's requirements.

With enterprise-class support and security patches and bug fixes at all layers in the stack, you can now focus on your application development, leading to better business outcomes.

3.3 Target Audience

The target audience for the solution includes the following groups:

- DevOps practitioners
- Enterprise IT cloud and virtualization administrators
- Service providers
- NetApp and Red Hat partners

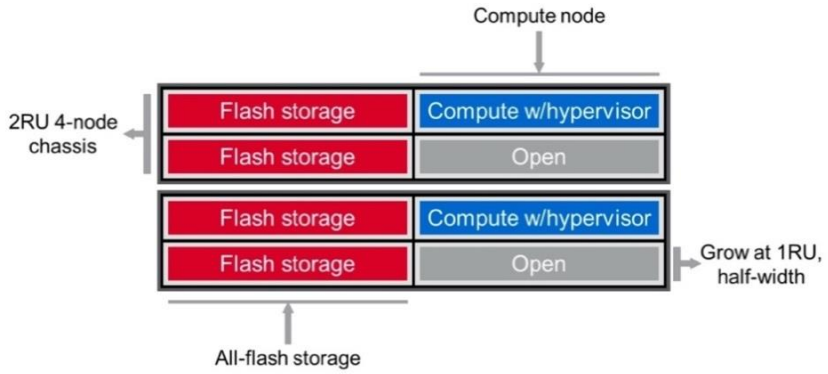
4 Solution Technology

4.1 NetApp HCI

NetApp HCI is an enterprise-scale hyper converged infrastructure (HCI) solution that delivers compute and storage resources in an agile, scalable, easy-to-manage two-rack unit (2RU), four-node building block. This solution is based on the following minimum configuration (as shown in Figure 1):

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes running VMware ESXi
- NetApp Deployment Engine (NDE) and NetApp Element® Plug-in for VMware vCenter Server , which enable deployment and management of NetApp HCI

Figure 1) NetApp HCI minimum configuration.



For details about and technical specifications for compute and storage nodes in NetApp HCI, see the NetApp HCI datasheet.

NetApp HCI Design Principles

By providing an agile turnkey infrastructure platform, NetApp HCI enables you to run enterprise-class virtualized and containerized workloads in an accelerated manner. At its core, NetApp HCI is designed to provide predictable performance, linear scalability, and a simple deployment and management experience.

Predictable

One of the biggest challenges in a multitenant environment is delivering consistent predictable performance for all your workloads. Running multiple enterprise-grade workloads can result in resource contention, in which one workload might interfere with the performance of another. NetApp HCI alleviates this concern with QoS limits that are available natively with NetApp Element software. NetApp Element software allows the granular control of every application and volume, eliminates noisy neighbors, and satisfies performance SLAs. NetApp HCI multitenancy capabilities can help eliminate more than 90% of traditional performance-related problems[1].

Flexible

Previous generations of HCIs required fixed resource ratios, limiting deployments to four- to eight-node configurations. NetApp HCI, however, scales compute and storage resources independently. Independent scaling prevents costly and inefficient overprovisioning, eliminates the 10% to 30% HCI tax from controller VM overhead, and simplifies capacity and performance planning.

With NetApp HCI, your licensing costs are reduced. NetApp HCI is available in mix-and-match small, medium, and large storage and compute configurations. The architectural design choices that are offered enable you to confidently scale on your terms, making HCI viable for core data center applications and platforms.

[1] Source: https://www.netapp.com/us/resources/esg-lab-report-quantifying-the-economic-value-of-a-solidfire-deployment

NetApp HCI is architected in building blocks at either the chassis or the node level. Each chassis can hold four nodes, made up of storage nodes, compute nodes, or both. A minimum configuration is two chassis with six nodes, consisting of four storage nodes and two compute nodes. Two additional blank spots can be used for expansion. If you follow best practices, you can mix compute and storage nodes. Resources can be scaled nondisruptively through a simple UI-driven process.

It is easy to resize the OpenShift VMs and to create new worker nodes as your scale needs change. You can also upgrade operating systems nondisruptively with rollback through NetApp Snapshot™ copies and maintain HA with native HA and anti-affinity rules.

Simple

A driving imperative within the IT community is to automate all routine tasks, eliminating the risk of user error while freeing up resources to focus on more interesting, higher-value projects. NetApp HCI can help your IT department become more agile and responsive by simplifying deployment and ongoing management.

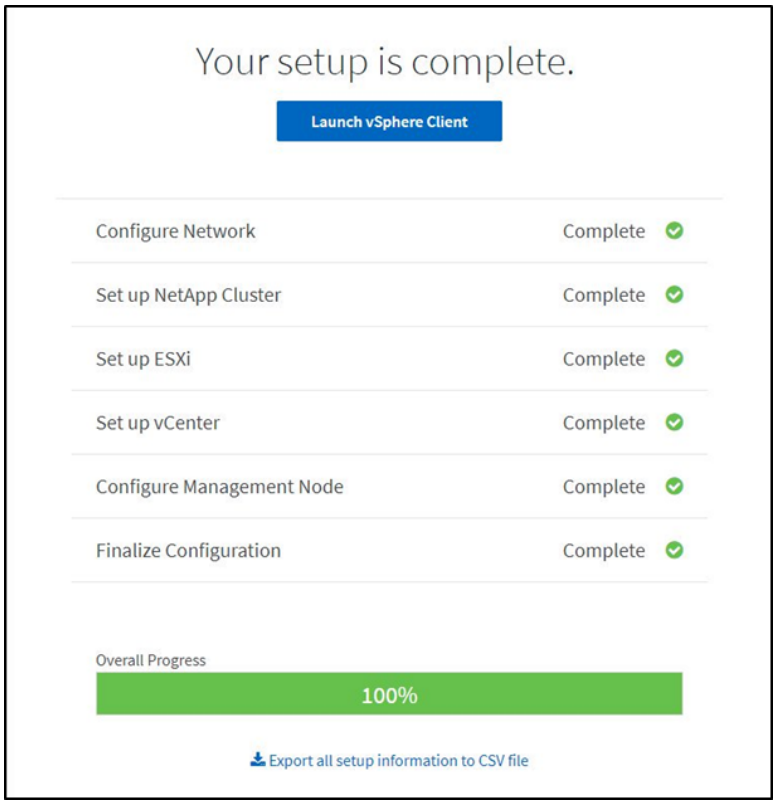
The new NDE eliminates most manual steps that are involved in deploying infrastructure, such as assigning names, network settings, and IP addresses, and provisioning ESXi hosts and VMware datastores. You can expect the infrastructure to be functional in less than 30 minutes.

The Element Plug-in for vCenter simplifies management, and it's intuitive. Also, with NetApp HCI, a robust suite of APIs enables integration into higher-level management, orchestration, backup, and disaster recovery tools.

4.2 NetApp Deployment Engine

The NDE enables the quick deployment of NetApp HCI, including the NetApp Element software cluster and the VMware virtualized infrastructure. NDE simplifies day 0 deployment by reducing the number of manual steps from over 400 to less than 30. NDE is intuitive and reuses data such as user name and password; therefore, you do not have to reenter information or set credentials at varying complexity levels. Likewise, assigning IP addresses is taken care of by NDE, allowing you to set a scheme and a pool for all resources before the actual configuration. Also, preinstallation checklists enable successful deployments because the system automatically checks for user errors, eliminating manual checks.

Figure 2) Successful NDE deployment.



As seen in Figure 2, NDE optimally configures the data and management networks, configures the cluster, and sets up VMware ESXi and vCenter and other required configurations. Your virtualized environment is up and running in a risk-free manner.

For more information about NDE, see the NetApp HCI NDE User Guide.

For more information about deploying NetApp HCI, see the NetApp HCI Deployment Guide 1.4

4.3 NetApp Element Software

NetApp Element provides modular, scalable performance with each storage node delivering guaranteed capacity and throughput to the environment.

iSCSI Login Redirection and Self-Healing Capabilities

NetApp Element software leverages the iSCSI storage protocol, a standard way to encapsulate iSCSI commands on a traditional TCP/IP network. When iSCSI standards change, or when the performance of Ethernet networks improves, the iSCSI storage protocol benefits without the need for any changes. Although all storage nodes have a management IP and a storage IP, NetApp Element software advertises a single storage virtual IP address (SVIP address) for all storage traffic of the cluster. As a part of the iSCSI login process, the storage can respond that the target volume has been moved to a different address and therefore it cannot proceed with the negotiation process. The host then reissues the login request to the new address in a process that requires no host-side reconfiguration. This process is known as iSCSI login redirection.

iSCSI login redirection is a key part of the NetApp Element software cluster. When a host login request is received, the node decides which member of the cluster should handle the traffic based on the IOPS and the capacity requirements for the volume. Volumes are distributed across the NetApp Element software

cluster and are redistributed if a single node is handling too much traffic for its volumes or if a new node is added. Multiple copies of a given volume are allocated across the array. In this manner, if a node failure is followed by volume redistribution, there is no effect on host connectivity beyond a logout and login with redirection to the new location. With iSCSI login redirection, the NetApp Element software cluster is a self-healing, scale-out architecture that is capable of nondisruptive upgrades and operations.

NetApp Element Software Cluster QoS

A NetApp Element software cluster allows QoS to be dynamically configured on a per-volume basis. You can use per-volume QoS settings to control storage performance based on SLAs that you define.

Three configurable parameters define the QoS:

- **Minimum IOPS:** the minimum number of sustained IOPS that the NetApp Element software cluster provides to a volume. The minimum IOPS configured for a volume is the guaranteed level of performance for a volume. Per-volume performance does not drop below this level.
- **Maximum IOPS:** the maximum number of sustained IOPS that the NetApp Element software cluster provides to a particular volume.
- **Burst IOPS:** the maximum number of IOPS allowed in a short burst scenario. If a volume has been running below the maximum IOPS, burst credits are accumulated. When performance levels become very high and are pushed to maximum levels, short bursts of IOPS beyond maximum IOPS are allowed on the volume.

Multitenancy

Secure multitenancy is achieved through the following features:

- **Secure authentication.** The Challenge-Handshake Authentication Protocol (CHAP) is used for secure volume access. The Lightweight Directory Access Protocol (LDAP) is used for secure access to the cluster for management and reporting.
- **Volume access groups (VAGs).** Optionally, VAGs can be used in lieu of authentication, mapping any number of iSCSI initiator specific iSCSI Qualified Names (IQNs) to one or more volumes. To access a volume in a VAG, the initiator's IQN must be in the allowed IQN list for the group of volumes.
- **Tenant virtual LANs (VLANs).** At the network level, end-to-end network security between iSCSI initiators and the NetApp Element software cluster is facilitated by using VLANs. For any VLAN that is created to isolate a workload or a tenant, NetApp Element Software creates a separate iSCSI target SVIP address that is accessible only through the specific VLAN.
- **VRF-enabled VLANs.** To further support security and scalability in the data center, NetApp Element software allows you to enable any tenant VLAN for VRF-like functionality. This feature adds these two key capabilities:
  - **L3 routing to a tenant SVIP address.** This feature allows you to situate iSCSI initiators on a separate network or VLAN from that of the NetApp Element software cluster.
  - **Overlapping or duplicate IP subnets.** This feature enables you to add a template to tenant environments, allowing each respective tenant VLAN to be assigned IP addresses from the same IP subnet. This capability can be useful for in-service provider environments where scale and preservation of IPspace are important.

Enterprise Storage Efficiencies

The NetApp Element software cluster leverages key features to increase overall storage efficiency and performance. The following features are performed inline, are always on, and require no manual configuration by the user:

- **Deduplication.** The system stores only unique 4K blocks. Any duplicate 4K blocks are automatically associated to an already stored version of the data. Data is on block drives and is mirrored by using

the NetApp Element software Helix® data protection. This system significantly reduces capacity consumption and write operations within the system.

- **Compression.** Compression is performed inline before data is written to NVRAM. Data is compressed and stored in 4K blocks, and after it has been compressed, it remains compressed in the system. This compression significantly reduces capacity consumption, write operations, and bandwidth consumption across the cluster.
- **Thin provisioning.** This capability provides the right amount of storage at the time that you need it, eliminating capacity consumption that caused by overprovisioned volumes or underutilized volumes.
- **Helix.** The metadata for an individual volume is stored on a metadata drive and is replicated to a secondary metadata drive for redundancy.

  Note: Element was designed for automation. All the storage features are available through APIs. These APIs are the only method that the UI uses to control the system.

For more information, see the Element Software product page.

4.4 Project Trident

Trident enables microservices and containerized applications to leverage enterprise-class storage services (such as QoS, storage efficiencies, and cloning) to meet the applications' persistent storage demands. Depending on an application's requirements, Trident can dynamically provision storage from:

- NetApp ONTAP® data management software (NetApp AFF, FAS, ONTAP Select, and Cloud Volumes ONTAP)
- NetApp Element software (NetApp HCI and SolidFire)
- NetApp SANtricity® software (NetApp E-Series and EF-Series)

Trident makes use of the StorageClass object that was introduced in Kubernetes 1.4 to dynamically provision Persistent Volumes (PV) when a PersistentVolumeClaim (PVC) object is created. A storage class provides a way for administrators to describe the classes of storage that they offer. A storage class might map to different QoS levels, backup policies, or other storage characteristics. In this reference architecture, NetApp recommends that you create a storage class that specifies Trident as the provisioner and SolidFire SAN as the Trident storage back end.

For more information, see the Trident documentation.

4.5 Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform unites development and IT operations on a single platform to build, deploy, and manage applications consistently across on-premises and hybrid cloud infrastructures. Red Hat OpenShift is built on open-source innovation and industry standards, including Kubernetes and Red Hat Enterprise Linux, the world's leading enterprise Linux distribution.

OpenShift is part of the Cloud Native Computing Foundation (CNCF) Certified Kubernetes program, providing portability and interoperability of your container workloads. OpenShift Container Platform provides the following capabilities:

- **Self-service provisioning.** Developers can quickly and easily create applications on demand from the tools that they use most, while operations retain full control over the entire environment.
- **Persistent storage.** By providing support for persistent storage, OpenShift Container Platform allows you to run both stateful applications and cloud-native stateless applications.
- **Continuous integration and continuous development (CI/CD).** This source-code platform manages build and deployment images at scale.
- **Persistent storage:** By providing support for persistent storage, OpenShift Container Platform allows users to run both stateful applications and cloud-native stateless applications.

- **Open-source standards.** These standards incorporate the Open Container Initiative (OCI) and Kubernetes for container orchestration, in addition to other open-source technologies. You are not restricted to the technology or to the business roadmap of a specific vendor.
- **CI/CD pipelines.** OpenShift provides out-of-the-box support for CI/CD pipelines so that development teams can automate every step of the application delivery process and make sure it's executed on every change that is made to the code or configuration of the application.
- **Role-Based Access Control (RBAC).** This feature provides team and user tracking to help organize a large developer group.
- **Automated build and deploy.** OpenShift gives developers the option to build their containerized applications or have the platform build the containers from the application source code or even the binaries. The platform then automates deployment of these applications across the infrastructure based on the characteristic that was defined for the applications. For example, how quantity of resources that should be allocated and where on the infrastructure they should be deployed in order for them to be compliant with third-party licenses.
- **Consistent environments.** OpenShift makes sure that the environment provisioned for developers and across the lifecycle of the application is consistent from the operating system, to libraries, runtime version (for example, Java runtime), and even the application runtime in use (for example, tomcat) in order to remove the risks originated from inconsistent environments.
- **Configuration management.** Configuration and sensitive data management is built in to the platform to make sure that a consistent and environment agnostic application configuration is provided to the application no matter which technologies are used to build the application or which environment it is deployed.
- **Application logs and metrics.** Rapid feedback is an important aspect of application development. OpenShift integrated monitoring and log management provides immediate metrics back to developers in order for them to study how the application is behaving across changes and be able to fix issues as early as possible in the application lifecycle.
- **Security and container catalog.** OpenShift offers multitenancy and protects the user from harmful code execution by using established security with Security-Enhanced Linux (SELinux), CGroups, and Secure Computing Mode (seccomp) to isolate and protect containers, encryption through TLS certificates for the various subsystems, and providing access to Red Hat certified containers (`access.redhat.com/containers`) that are scanned and graded with a specific emphasis on security to provide certified, trusted, and secure application containers to end users.

4.6 Solution Design Highlights

The solution provides the following key features:

- HA is available at all layers.
- A complete scale-out architecture with independent scale points is used for compute and storage.
- NetApp HCI allows containerized workloads and virtualized workloads to run alongside one other.
- VMware and NetApp Element QoS enable predictable, guaranteed performance for mixed workloads.
- Trident enables enterprise-class dynamic storage provisioning for containerized workloads.
- NetApp HCI enables multitenancy with resource requests and limits.
- Red Hat OpenShift Container Platform enables enterprise-grade Kubernetes with tested, hardened, and validated integrations.
- Red Hat Enterprise Linux provides an enterprise-class hardened Linux operating system, with all security patches and bug fixes.
- There is a single point of support for all the components in the stack: infrastructure, hypervisor, operating system, and container orchestrator.

5 Solution Design

5.1 Technology Requirements

This section covers the technology requirements for the Red Hat OpenShift Container Platform with NetApp HCI validated solution. Individual customer requirements might vary.

For more information about the technical requirements and for installation guidance on NetApp HCI, review the NetApp HCI Resources page.

Hardware Requirements

Table 1 lists the hardware components that are required to implement the solution. The hardware components that are used in any particular implementation of the solution might vary based on your organization's requirements.

Table 1) Hardware requirements.

Layer	Product Family	Quantity	Details
Compute	NetApp H500E	4	2 x Intel E5-2650 v4; 12 cores; 2.2GHz 512GB RAM
Storage	NetApp H500S	4	6 x 960GB encrypting/nonencrypting

Software Requirements

Table 2 lists the software components that are required to implement the solution. The software components that are used in any particular implementation of the solution might vary based on your organization's requirements.

Table 2) Software requirements.

Layer	Software	Version
Storage	NetApp Element software	10.4
	Trident	18.10
NetApp HCI engine	NetApp Deployment Engine	1.4
Hypervisor and later	Hypervisor	VMware vSphere ESXi 6.5 U2
	Hypervisor Management System	VMware vCenter Server 6.5
	Red Hat Enterprise Linux	7.5
	Red Hat OpenShift Container Platform	3.10

Note: NetApp HCI is switch vendor agonistic and relies on standard enterprise-class data center switching features. The network design is described in section 5.2, "Architectural Overview."

License Requirements

The following components in the solution have license requirements:

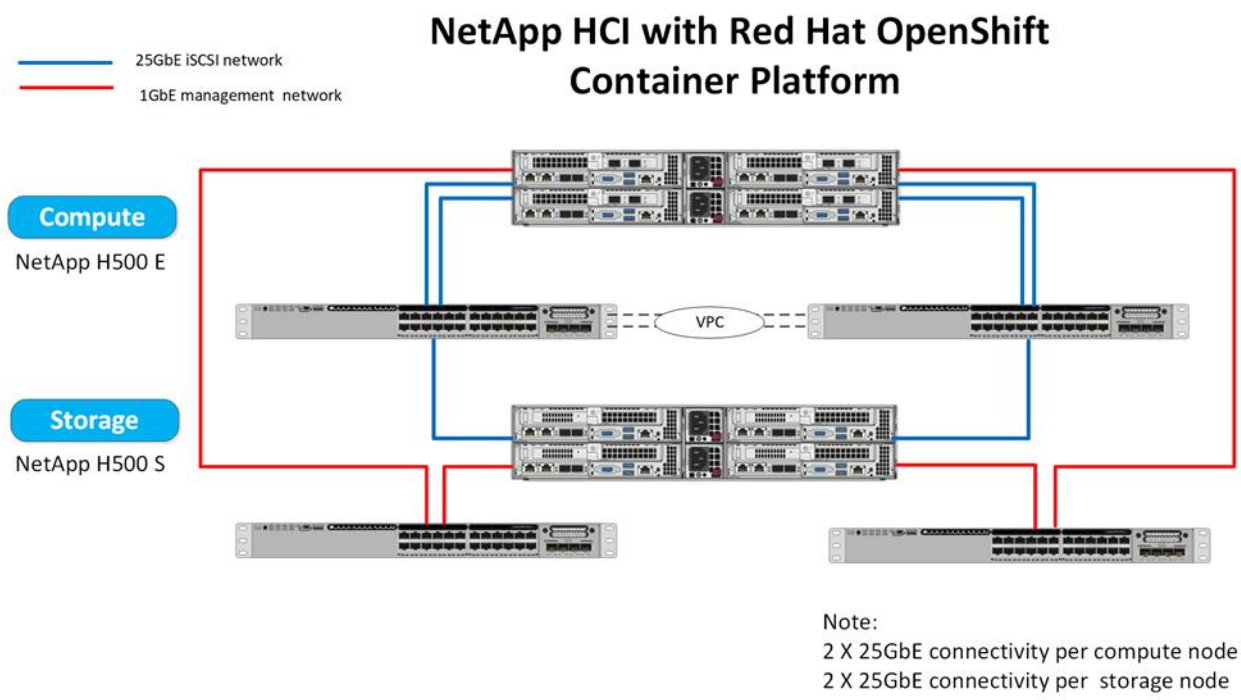
- VMware vSphere
- VMware vCenter Server Enterprise Plus
- Red Hat OpenShift Container Platform subscription

5.2 Architectural Overview

Physical Topology

Figure 3 shows the topology of NetApp HCI with Red Hat OpenShift Container Platform.

Figure 3) Topology of NetApp HCI with Red Hat OpenShift Container Platform.

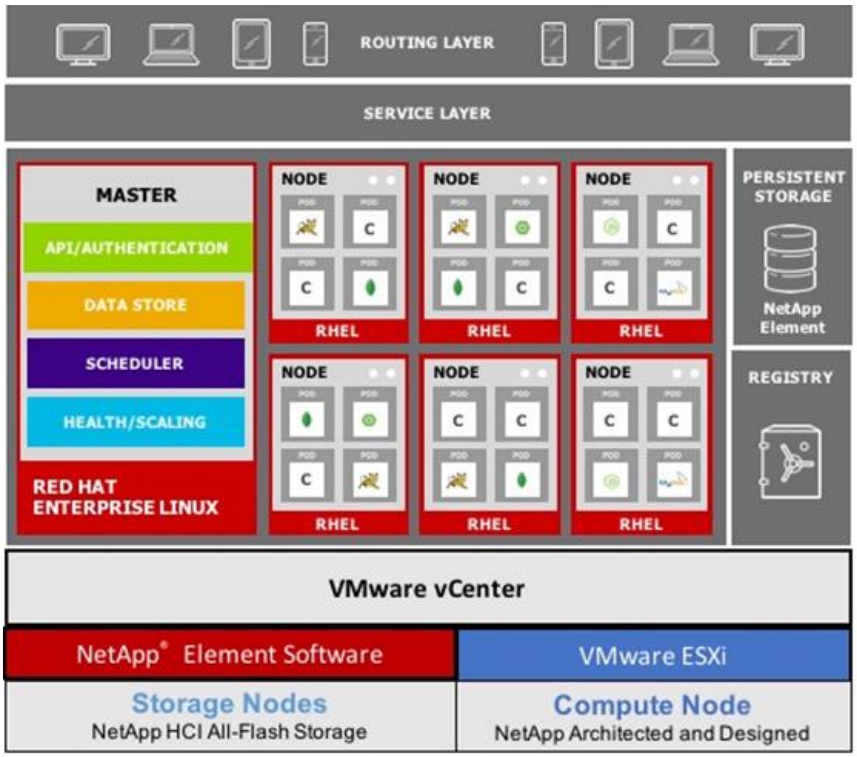


This reference architecture enables an end-to-end 25GbE iSCSI network. NetApp Deployment Engine configures the following out-of-the-box network configuration:

- The two 25GbE storage ports on the NetApp H500S nodes are configured in a Link Aggregation Control Protocol (LACP) mode, thereby providing higher throughput and resiliency.
- The two 25GbE compute ports on the NetApp H500E nodes are configured with VMware iSCSI port binding, thereby providing higher throughput and resiliency.
- NDE automatically configures the required port channels for the VM network and VMware vMotion.
- An additional port group with `OpenShift-Storage` VLAN tagging is created to map iSCSI storage directly from Red Hat Enterprise Linux nodes. This port group uses the Route Based on Physical NIC Load load-balancing policy.
- Jumbo frames are configured end to end

Figure 4 shows the layered infrastructure.

Figure 4) Layered architecture (image provided by Red Hat).



NetApp HCI leverages VMware ESXi as the hypervisor and VMware vCenter Server for centralized management of VMs, ESXi hosts, and NetApp Element software. All the OpenShift nodes are deployed as Red Hat Enterprise Linux 7.5 VMs on NetApp HCI.

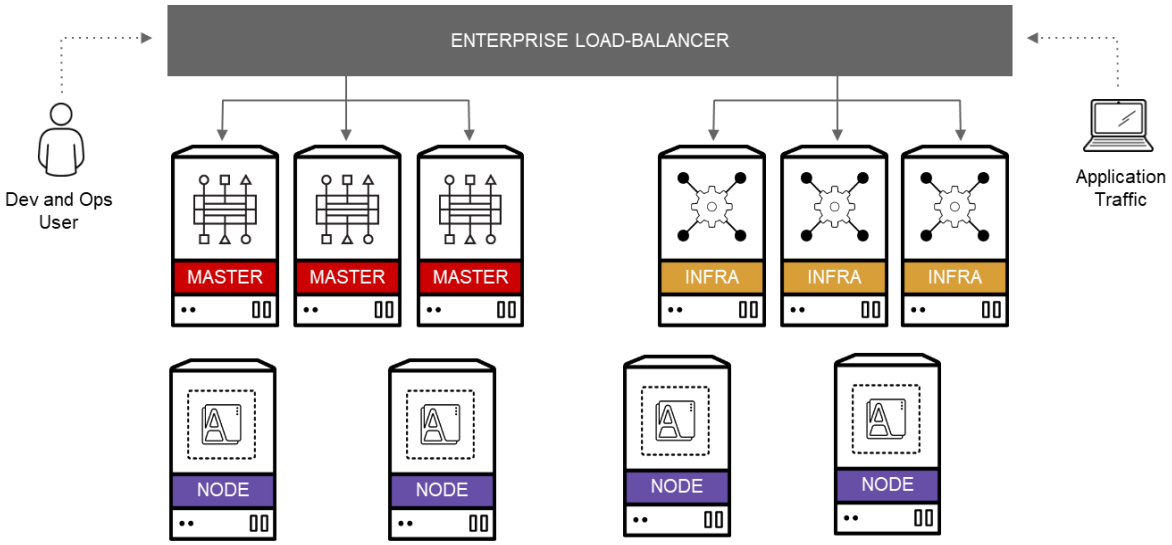
5.3 Red Hat OpenShift Container Platform Components

Red Hat OpenShift Container Platform uses Kubernetes to manage containerized applications across a set of hosts. A highly available deployment of OpenShift requires the following components:

- A deployment instance
- Three master instances
- Three infrastructure instances
- Four application instances

Figure 5 illustrates the HA Red Hat OpenShift Container Platform.

Figure 5) HA Red Hat OpenShift Container Platform (image provided by Red Hat).



VMware scheduler considers different scheduling filters and schedules the OpenShift nodes across the ESXi cluster. VMware HA makes sure that the OpenShift nodes are restarted on a different NetApp HCI compute node in the event of a physical node failure. VMware Distributed Resource Scheduler (DRS) also makes sure that the OpenShift nodes are migrated to a different ESXi host to optimize resource usage.

To avoid losing quorum among OpenShift master nodes in the event of a physical node failure, VMware VM-VM anti-affinity rules are used to separate OpenShift master nodes on different physical hosts. Similarly, OpenShift infrastructure nodes are separated on different physical nodes.

OpenShift Master Nodes

Master nodes run the control-plane components of the OpenShift cluster. Master nodes make global decisions about the cluster; for example, scheduling, monitoring, and responding to cluster events. The control-plane components such as the API server, controller manager server, and etcd run as static pods that are managed by a kubelet. Table 3 describes the different master components.

Table 3) OpenShift master nodes.

Component	Description
API server	The Kubernetes API server acts as a front end for the Kubernetes control plane. It accepts the desired cluster state (YAML or JSON configuration files). Examples of the cluster state include, but are not limited to: • Applications or other workloads to run • Container images for your applications and workloads • Allocation of network and disk resources
Etcd	Etcd stores the cluster state information persistently.

Component	Description
Controller manager server	The controller manager includes: • **Node controller:** notices and responds when nodes go down • **Replication controller:** maintains the correct number of pods for every replication controller object in the system • **Endpoints controller:** populates the endpoints object (that is, joins services and pods) • **Service account and token controllers:** create default accounts and API access tokens for new namespaces

Note: You can also label the OpenShift master nodes to run application workloads along with control-plane elements.

OpenShift Nodes

Typically, all the application workloads are scheduled to run on OpenShift nodes. A node is a worker machine in OpenShift. A node provides the run-time environments for containers. Each node in an OpenShift cluster has the required services that must be managed by the master node. Nodes also have the required services to run pods, including the container run time, a kubelet, and a service proxy. Table 4 describes the various components in OpenShift nodes.

Table 4) OpenShift node components.

Component	Description
Kubelet	• A kubelet is an agent that runs on each node in the cluster. It makes sure that containers are running in a pod. • The kubelet takes a set of PodSpecs that are provided through various mechanisms and makes sure that the containers that are described in those PodSpecs are running and are healthy. The kubelet does not manage containers that were not created by Kubernetes.
kube-proxy	kube-proxy is a service proxy that enables the Kubernetes service abstraction by maintaining network rules on the host and by performing connection forwarding.

OpenShift Infrastructure Nodes

Infrastructure nodes are dedicated to running applications that administrators deploy to provide services for the OpenShift Container Platform cluster. Infrastructure nodes run services such as OpenShift Container Registry (OCR) and the OpenShift Container Platform router. OpenShift logging stacks, such as ElasticSearch, Fluentd, and Kibana (EFK), are also deployed on the infrastructure nodes. Table 5 lists the OpenShift infrastructure node components.

Table 5) OpenShift Infrastructure node components.

Component	Description
OCR	OCR adds the ability to automatically provision new image repositories on demand. You get a preprovisioned endpoint to push the application build images.
OpenShift Container Platform router	This router enables routes that are created by developers to be used by external clients. The routing layer in OpenShift Container Platform router is

Component	Description
	pluggable, and several router plug-ins are provided and are supported by default.

5.4 Design Considerations for Deploying Red Hat OpenShift Container Platform with NetApp HCI

NetApp HCI leverages VMware ESXi as the hypervisor and VMware vCenter Server for centralized management of VMs, ESXi hosts, and the other dependent components. All the OpenShift nodes are deployed as Red Hat Enterprise Linux 7.5 VMs on NetApp HCI.

OpenShift Node Sizing

Red Hat provides guidance on the minimum resource requirements recommendation for OpenShift nodes. Some of the major factors that influence sizing decisions are described in this section.

OpenShift Nodes

Your sizing decisions should consider:

- The application profile of your containerized workloads.
- The number of pods that are expected to run in the cluster.

Note: NetApp strongly recommends that you disable swap memory on nodes so that you avoid memory pressure situations.

OpenShift Master Nodes

When you size your system, consider the following:

- One CPU core and 1.5GB of memory per 1,000 pods are required.
- A 50-node Kubernetes cluster requires two virtual CPUs (vCPUs) and 8GB of RAM for each etcd instance.

The solution was validated with the capacity that is shown in Table 6.

Table 6) OpenShift VMs specifications.

Node	Number of vCPUs	Memory (GB)
OpenShift master	8	16
OpenShift node	4	16
OpenShift infrastructure	8	16

Note: OpenShift Container Platform is scale-out by design. Additional OpenShift nodes can be provisioned on demand. Furthermore, the independent design points in NetApp HCI allow you to add more HCI compute nodes as you need them.

Note: It is recommended that you not overprovision vCPUs to allow VMware scheduler to provision VMs on a different host when a sufficient number of vCPUs are not available on a single host.

Setting Resource Limits for OpenShift Nodes

When OpenShift nodes are scheduled to full capacity, pods can consume all the available resources on a node by default . This scenario can cause an issue because nodes typically run quite a few system daemons that power the operating system and Kubernetes itself. Unless resources are set aside for these system daemons, pods and system daemons compete for resources, which can lead to resource

starvation issues on the node. The kubelet exposes a feature named Node Allocatable, which helps to reserve compute resources for system daemons. Some general guidelines include:

- Enforce `Allocatable` on pods.
- After adequate monitoring and alerting are in place to track kube-system daemons, attempt to enforce `kube-reserved` based on usage heuristics.
- If necessary, enforce `system-reserved` over time.

Using VMware Resource Pool for OpenShift Environment

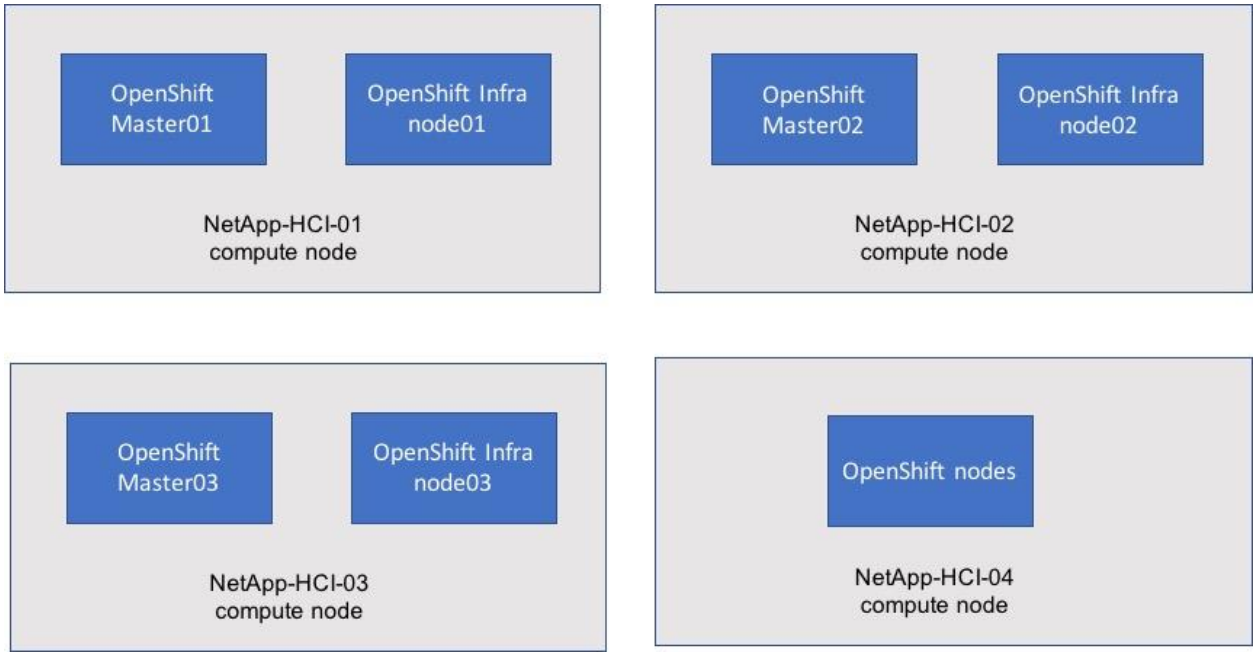
By default, VMware ESXi does not entirely reserve the resources that are requested by a VM. OpenShift scheduler detects the node's `Allocatable` resources (which correspond to a virtual hardware specification) for scheduling the pod. If other resource-intensive workloads are running on the VMware ESXi cluster, it might lead to resource starvation to the application pod despite being scheduled correctly by the Kubernetes scheduler. Reserving the VM resources can lead to inefficiencies. Hence, NetApp strongly recommends that you create a VMware resource pool, which is an aggregation of resources for all the OpenShift nodes. The OpenShift cluster then has guaranteed resources regardless of any other workloads that are running on the ESXi cluster.

VMware Anti-Affinity Rules

As discussed in section 5.3, "Red Hat OpenShift Container Platform Components," the underlying infrastructure is designed to tolerate physical node failures and is completely highly available by design. VMware HA provides migration of OpenShift nodes during physical node failures. So that you avoid losing quorum during a physical node failure, NetApp recommends that you not have more than one OpenShift master and one OpenShift infrastructure node on the same physical nodes. OpenShift Container Platform on NetApp HCI leverages VMware VM-VM anti-affinity rules to separate the OpenShift master nodes into separate failure domains. Similarly, OpenShift Infrastructure nodes are scheduled in separate failure domains.

Figure 6 shows the VM layout that uses VMware anti-affinity rules.

Figure 6) VM layout that uses VMware anti-affinity rules.



VMware ESXi also provides fair CPU shares to processes while accounting for shared resources in hyperthreading. VM resources are also scheduled to avoid nonuniform memory access (NUMA) boundaries.

5.5 Network Design Considerations for Red Hat OpenShift Container Platform

OpenShift uses virtual extensible LAN (VXLAN) overlay networking for pod networks. This deployment was validated with the `ovs-multitenant` plug-in that enables project-level isolation between pods and services.

The different network traffic types in OpenShift are isolated by using VLANs. The following port groups with corresponding VLAN tags are created:

- **Management:** In-band management network
- **Storage network:** iSCSI network for persistent storage
- **Intracluster traffic:** The overlay OpenShift network
- **Cluster egress traffic:** External traffic for services

5.6 Storage Elements in Red Hat OpenShift Container Platform

Persistent Storage for Containerized Applications

Trident enables dynamic provisioning of enterprise-class storage from NetApp Element software-based NetApp HCI all-flash storage nodes. Kubernetes 1.4 introduced support for storage classes, which allow you to specify the provisioner and additional details such as QoS levels and media types.

Trident is deployed and managed by an OpenShift administrator:

- NetApp recommends that you deploy Trident on OpenShift infrastructure nodes alongside other OpenShift services such as OCR or OpenShift Platform router.
- NetApp recommends that you create a `trident` user in Element software with access privileges to read, volumes, accounts, and ClusterAdmins resource types.
- NetApp recommends that you use CHAP authentication to provide access to PVs in the nodes. Set `UseCHAP` to `true` in the Trident storage back end.
- Depending on the workload requirements, use storage pools to create different storage classes that correspond to different QoS bands.
- For generic use cases without specific QoS requirements, NetApp recommends that you create a default storage class that specifies Trident as the storage provisioner and that uses the default QoS settings in Element software.

Multitenancy and Storage Quotas

OpenShift projects enable multitenancy in OpenShift deployments. Storage quotas provide resource limits for the storage resources in a multitenant environment. Storage limits can be set on the attributes listed in Table 7.

Table 7) Attributes for setting storage quotas.

Attribute	Description
`requests.storage`	Across all PV claims, the sum of storage requests cannot exceed this value.
`persistentvolumeclaims`	The value is the total number of PV claims that are allowed in the namespace.

Attribute	Description
`<storage-class-name>.storageclass.storage.k8s.io/requests.storage`	Across all PV claims that are associated with the storage-class name, the sum of storage requests cannot exceed this value.
`<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims`	The total number of PV claims that are associated with the namespace cannot exceed this value.

Note: Certain OpenShift projects can be configured to not use certain storage classes for provisioning. For example, `solidfire-gold. storageclass.storage.k8s.io/persistentvolumeclaims=0` does not use `solidfire-gold storageclass` for provisioning.

Persistent Volume Claim Attributes and Annotations

Access Modes

iSCSI block storage supports only Read Write One (`RWO`) and Read Only Many (`ROX`) modes. A PV can be attached to Read Write mode for only one host and can be attached to Read Only mode for multiple hosts. These modes are specified as `accessModes` in the PVC request.

Volume Lifecycle Management

Trident enables lifecycle management of PVs through annotations in a PVC object. The reclaim policy for the created PV can be determined by using the annotation `trident.netapp.io/reclaimPolicy`. Table 8 describes the PVC annotations for volume lifecycle management.

Table 8) PVC annotations for volume lifecycle management.

trident.netapp.io/reclaimPolicy	Description
Delete	Trident deletes the PV and the backing volume.
Retain	The administrator deletes the PV object and the backing volume.

Note: If the annotation is not unspecified, Trident uses the Delete Reclaim policy.

Cloning

Clones are not exposed through Kubernetes Persistent Volume Framework. However, these features have a critical use case in enterprise adoption of OpenShift, because data protection and storage efficiencies are crucial to achieve the agility that containers provide. Cloning can be set by using the PVC request annotations that are listed in Table 9.

Table 9) PVC annotations for cloning.

Annotation	Description
`trident.netapp.io/cloneFromPVC`	Provisions a new PV by cloning the existing volume

Notes:

- NetApp recommends that you clone off an idle volume.
- PVC and its clone must be in the same OpenShift project and must use the same storage class.

OpenShift etcd Storage

Etcd is a consistent and highly available key value store that is used as the Red Hat OpenShift Container Platform backing store for all cluster data. Etcd stores the persistent master state, and other components watch etcd for changes to bring themselves into the desired state.

Etcd uses the Raft algorithm to gracefully handle leader elections during network partitioning and the loss of the current leader. For a highly available Red Hat OpenShift Container Platform deployment, all three etcd instances are collocated on each of the OpenShift master nodes.

Because values that are stored in etcd are critical to the function of Red Hat OpenShift Container Platform, you should implement firewalls to limit the communication with OpenShift masters running etcd.

NetApp HCI all-flash nodes provide low-latency flash-based storage to etcd. NetApp strongly recommends that you create a separate Virtual Machine Disk (VMDK) for each OpenShift master node. This volume should be formatted to the `xfs` file system and should be mounted to `/var/lib/etcd`. This approach not only allows easy recovery of etcd data in node failure scenarios, but it also prevents the `/var` partition from becoming full. Separate volumes for etcd provides additional protection by using Snapshot copies.

Note: OpenShift Container Platform uses etcd for storing additional information beyond what Kubernetes itself requires. For example, OpenShift Container Platform stores information about images, builds, and other components in etcd as required by features that OpenShift Container Platform adds on top of Kubernetes.

Docker Graph Storage

Containers and the images from which they are created are stored in Docker's storage back end on the host. This storage is ephemeral and is separate from any persistent storage that is required for your applications. Docker stores images and containers in a graph driver, which is a pluggable storage technology such as Device Mapper, OverlayFS, and Btrfs.

For production use, NetApp strongly recommends that you create a separate VMDK (separate from the guest operating system root disk) and attach it to each OpenShift node for Docker graph storage. Docker should be configured with the overlay2 storage driver in thin-pool mode.

Note: Separate VMDK for Docker graph storage prevents the `/var` partition from becoming full on the OpenShift nodes.

OpenShift Container Registry

OpenShift can build container images from source code, deploy them, and manage their lifecycle. To enable this feature, OpenShift provides an internal, integrated registry, OCR, that can be deployed in the OpenShift environment to manage images. OCR stores images and metadata. For a production environment, persistent storage should be used for the registry; otherwise, any images that were built or were pushed into OCR would disappear if the pod were to restart.

OCR is deployed as a `DeploymentConfig` with `Repilcas` set to `1`. Therefore, OpenShift HA is responsible for moving the OCR pod to a different node in node failure scenarios. Because replica is set to `1`, a block-based PV from the Element software-based NetApp HCI all-flash nodes can be provisioned to provide persistent storage. OCR deployment (standalone or integrated) can be configured to dynamically provision persistent storage by using Trident.

Scaled OpenShift Registry

A scaled registry is an OpenShift Container Platform registry where three or more pod replicas are running. Therefore, the storage technology must support Read Write Many (RWX) mode. NetApp strongly recommends that you use NetApp StorageGRID® object storage with S3 protocol as the storage backend for scaled OpenShift registry. StorageGRID nodes are deployed as VMs on NetApp HCI. NetApp Element

volumes are attached to the StorageGRID storage nodes. StorageGRID with NetApp HCI has been validated to support over 17 million objects (Docker images) while consuming less than 15% of the compute capacity in your NetApp HCI cluster.

OpenShift Logging Services

Aggregated logging collects and aggregates logs from the pods that are running in the Red Hat OpenShift Container Platform cluster as well as `/var/log/messages` on nodes. Red Hat OpenShift Container Platform users can then use a web interface to view the logs of projects for which they have view access.

The Red Hat OpenShift Container Platform aggregated logging component is a modified version of the EFK stack, composed of a few pods that are running on the OpenShift Container Platform environment:

- **Elasticsearch** is an open-source, broadly distributable, enterprise-grade search engine.
- **Kibana** is a web UI for Elasticsearch.
- **Curator** automatically performs Elasticsearch maintenance operations on a per-project basis.
- **Fluentd** is a component that gathers logs from nodes and containers and feeds them to Elasticsearch.

A separate Elasticsearch cluster, a separate Kibana, and a separate Curator component can be deployed to form the operations cluster (OPS) cluster. Fluentd sends logs from the `default`, `openshift`, and `openshift-infra` projects as well as `/var/log/messages` on nodes into this different cluster.

Note: NetApp strongly recommends that you create a separate OPS cluster (a separate cluster of EFK) to manage logging for OpenShift components.

Elasticsearch requires persistent storage to persist the logging data. The logging services and OPS cluster should be configured to use a storage class that enables Trident to dynamically provision persistent storage from the `solidfire-san` back end.

OpenShift Metrics Services

Red Hat OpenShift Container Platform has the ability to gather metrics from the kubelet and to store the values in Heapster. Red Hat OpenShift Container Platform metrics services enable you to view CPU, memory, and network-based metrics, and the values are displayed in the UI. These metrics can allow horizontal autoscaling of pods based on parameters that you provide.

Red Hat OpenShift Container Platform metrics services are composed of a few pods that run on the OpenShift Container Platform environment:

- **Heapster** scrapes the metrics for CPU, memory, and network usage on every pod, then exports them into Hawkular Metrics.
- **Hawkular Metrics** is a metrics engine that stores the data persistently in a Cassandra database.
- **Cassandra** is a NoSQL database management system designed to handle large amounts of metrics data. It uses asynchronous masterless replication to allow low-latency operations for all clients.

You should use node selectors to specify where the metrics components should run. In the reference architecture environment, the components are deployed on nodes with the label `region=infra`.

Cassandra nodes use persistent storage so that metrics can be preserved. You should use the relevant `OpenShift-Ansible` variables to configure the metrics services deployment to use a storage class that enables Trident to dynamically provision persistent storage from the `solidfire-san` back end.

OpenShift EmptyDir

An emptyDir volume is first created when a pod is assigned to a node and exists as long as that pod is running on that node. As the name indicates, the volume it is initially empty. Containers in the pod can all read and write the same files in the emptyDir volume, however, that volume can be mounted at the same

or different paths in each container. When a pod is removed from a node for any reason, the data in the emptyDir is permanently deleted. Some emptyDir use cases include:

- Scratch space (for example, for a disk-based merge sort)
- Checkpointing a long computation for recovery from crashes

NetApp strongly recommends that you limit the size of emptyDir volumes and the volumes based on emptyDir volume, such as secrets and configuration maps, are on each node.

To limit the size of emptyDir volumes in an XFS file system, configure the local volume quota for each unique FSGroup by using the `node-config-compute` configuration map in the OpenShift node project.

This value represents the resource quantity representing the desired quota per [FSGroup], per node, such as 1Gi, 512Mi, and so on.

Note: This configuration requires that the volume directory be on an XFS file system mounted with the `grpquota` option. The matching security context constraint `fsGroup` type must be set to `MustRunAs`.

5.7 Security Recommendations

NetApp strongly recommends that you keep SELinux and firewalld enabled on all the nodes in your deployment.

6 Conclusion

Red Hat OpenShift Container Platform with NetApp HCI enables your organization to accelerate application development and deployment. With this enterprise-class, highly available, and scalable platform, you can run microservices and container workloads in a virtually risk-free manner.

Acknowledgements

The author of this document would like to thank and acknowledge the following people; they all had a positive impact on the quality of this document:

- Aaron Patten, Principal Architect, NetApp
- Andrew Sullivan, Technical Marketing Manager, Red Hat
- Justin Hover, Technical Marketing Engineer, NetApp
- Jeff Applewhite, Technical Marketing Engineer, NetApp
- Dave Cain, Senior Architect, Red Hat
- Thomas Hanvey, Technical Marketing Engineer, NetApp

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites:

- NetApp HCI
  https://www.netapp.com/us/products/converged-systems/hyper-converged-infrastructure.aspx
- NetApp HCI datasheet
  https://www.netapp.com/us/media/ds-3881.pdf
- NetApp product documentation
  http://docs.netapp.com

- Project Trident documentation
  https://netapp-trident.readthedocs.io/en/stable-v18.07/
- Red Hat OpenShift Container Platform 10
  https://access.redhat.com/documentation/en-us/openshift_container_platform/3.10/html/installing_clusters/
- Red Hat OpenShift Container Platform Scaling and Performance Guide
  https://access.redhat.com/documentation/en-us/openshift_container_platform/3.10/html-single/scaling_and_performance_guide/

Version History

Version	Date	Document Version History
Version 1.0	October 2018	Initial release.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Trademark Information

NVA-1124-1018-DESIGN