Running Red Hat OpenShift Container Platform on VMware Cloud Foundation

Reference Architecture

vmware

DELL Technologies

Table of contents

Note: This solution provides general design and deployment guidelines for running OpenShift Container Platform on VMware Cloud Foundation. It is showcased in this paper running on Dell EMC VxRail. The reference architecture applies to any compatible hardware platforms running VMware Cloud Foundation.

Executive Summary

Business Case

Red Hat® OpenShift® offers automated installation, upgrades, and lifecycle management throughout the container stack—the operating system, Kubernetes and cluster services, and applications on any cloud. OpenShift helps teams build with speed, agility, confidence, and choice. OpenShift is focused on security at every level of the container stack and throughout the application lifecycle. It includes long-term and enterprise support from one of the leading Kubernetes contributors and open-source software companies.

The manageability of operating an OpenShift environment with virtualized infrastructure can be improved over the management of traditional IT infrastructure on bare metal, since the demand for resources can fluctuate with business needs, leaving the OpenShift cluster either under-powered or over-provisioned. IT needs a more flexible, scalable, and secure infrastructure to handle the ever-changing demands of OpenShift. With a single architecture that is easy to deploy, VMware Cloud Foundation™ can provision compute, network, and storage on demand. VMware Cloud Foundation protects network and data with micro-segmentation and satisfies compliance requirements with data-at-rest encryption. Policy-based management delivers business-critical performance. VMware Cloud Foundation delivers flexible, consistent, secure infrastructure and operations across private and public clouds and is ideally suited to meet the demands of OpenShift.

Dell EMC VxRail is the only fully integrated, preconfigured, and tested HCI system optimized for VMware vSAN and is the standard for transforming VMware environments. VxRail invests in the complete lifecycle, including advanced automation, making it easier for you from day one forward, allowing you to further simplify IT infrastructure and operations.

In this solution, we provide the generic design and deployment guidelines for OpenShift on VMware Cloud Foundation on VxRail.

Technology Overview

Solution technology components are listed below:

- VMware Cloud Foundation
    - VMware vSphere®
    - VMware vSAN™
    - VMware NSX® Data Center
- Dell EMC VxRail
    - VxRail HCI System Software
- Red Hat OpenShift

VMware Cloud Foundation

VMware Cloud Foundation is an integrated software stack that combines compute virtualization (VMware vSphere), storage virtualization (VMware vSAN), network virtualization (VMware NSX), and cloud management and monitoring (VMware vRealize® Suite) into a single platform that can be deployed on-premises as a private cloud or run as a service within a public cloud. This documentation focuses on the private cloud use case. VMware Cloud Foundation bridges the traditional administrative silos in data centers, merging compute, storage, network provisioning, and cloud management to facilitate end-to-end support for application deployment.

VMware vSphere

VMware vSphere is VMware's virtualization platform, which transforms data centers into aggregated computing infrastructures that include CPU, storage, and networking resources. vSphere manages these infrastructures as a unified operating environment and provides operators with the tools to administer the data centers that participate in that environment. The two core components of vSphere are ESXi™ and vCenter Server®. ESXi is the hypervisor platform used to create and run virtualized workloads. vCenter Server is the management plane for the hosts and workloads running on the ESXi hosts.

VMware vSAN

VMware vSAN is the industry-leading software powering VMware's software defined storage and HCI solution. vSAN helps customers evolve their data center without risk, control IT costs, and scale to tomorrow's business needs. vSAN, native to the market-leading hypervisor, delivers flash-optimized, secure storage for all of your critical vSphere workloads, and is built on industry-standard x86 servers and components that help lower TCO in comparison to traditional storage. It delivers the agility to scale IT easily and offers the industry's first native HCI encryption.

vSAN simplifies Day 1 and Day 2 operations, and customers can quickly deploy and extend cloud infrastructure and minimize maintenance disruptions. vSAN helps modernize Hyperconverged Infrastructure (HCI) by providing administrators a unified storage control plane for both block and file protocols and provides significant enhancements that make it a great solution for traditional virtual machines as well cloud-native applications. vSAN helps reduce the complexity of monitoring and maintaining infrastructure and enables administrators to rapidly provision a file share in a single workflow for Kubernetes-orchestrated cloud native applications.

VMware NSX Data Center

VMware NSX Data Center is the network virtualization and security platform that enables the virtual cloud network, a software-defined approach to networking that extends across data centers, clouds, and application frameworks. With NSX Data Center, networking and security are brought closer to the application wherever it's running, from virtual machines to containers to bare metal. Like the operational model of VMs, networks can be provisioned and managed independently of the underlying hardware. NSX Data Center reproduces the entire network model in software, enabling any network topology—from simple to complex multitier networks—to be created and provisioned in seconds. Users can create multiple virtual networks with diverse requirements, leveraging a combination of the services offered via NSX or from a broad ecosystem of third-party integrations ranging from next-generation firewalls to performance management solutions to build inherently more agile and secure environments. These services can then be extended to a variety of endpoints within and across clouds.

Dell EMC VxRail

The only fully integrated, pre-configured, and pre-tested VMware hyperconverged integrated system optimized for VMware vSAN and VMware Cloud Foundation, VxRail transforms HCI networking and simplifies VMware cloud adoption while meeting any HCI use case - including support for many of the most demanding workloads and applications. Powered by Dell EMC PowerEdge server platforms and VxRail HCI System Software, VxRail features next-generation technology to future proof your infrastructure and enables deep integration across the VMware ecosystem. The advanced VMware hybrid cloud integration and automation simplifies the deployment of a secure VxRail cloud infrastructure.

VxRail HCI System Software

VxRail HCI system software is integrated software that delivers a seamless and automated operational experience, offering 100% native integration between VxRail Manager and vCenter. Intelligent lifecycle management automates non-disruptive upgrades, patching, and node addition or retirement while keeping VxRail infrastructure in a continuously validated state to ensure that workloads are always available. The HCI System Software includes SaaS multi-cluster management and orchestration for centralized data collection and analytics that uses machine learning and AI to help customers keep their HCI stack operating at peak performance and ready for future workloads. IT teams can benefit from the actionable insights to optimize infrastructure performance, improve serviceability, and foster operational freedom.

Red Hat OpenShift

Red Hat OpenShift ships with Red Hat Enterprise Linux® CoreOS for the Kubernetes control plane nodes and supports both Red Hat Enterprise Linux CoreOS and Red Hat Enterprise Linux for worker nodes. OpenShift supports the Open Container Initiative (OCI), which is an open governance structure around

container formats and runtimes. OpenShift includes hundreds of fixes to defect, security, and performance issues for upstream Kubernetes in every release. It is tested with dozens of technologies and is a robust tightly integrated platform supported over a 9-year lifecycle. OpenShift includes software-defined networking and validates additional common networking solutions. OpenShift also validates numerous storage and third-party plug-ins for every release.

See *https://www.openshift.com/products/container-platform* for detailed information regarding OpenShift Container Platform.

VMware NSX-T Container Plug-in for OpenShift

VMware NSX Container Plugin (NCP) provides the integration between NSX-T Data Center and container orchestrators such as Kubernetes, as well as integration between NSX-T Data Center and container-based PaaS (platform as a service) software products such as OpenShift.

The main component of NCP runs in a container and communicates with NSX Manager and with the OpenShift control plane. NCP monitors changes to containers and other resources and manages networking resources such as logical ports, switches, routers, and security groups for the containers by calling the NSX-T Policy API.

The NSX CNI plug-in runs on each OpenShift node. It monitors container life cycle events, connects a container interface to the guest vSwitch, and programs the guest vSwitch to tag and forward container traffic between the container interfaces and the vNIC.

Kubernetes vSphere CSI Driver

Cloud Native Storage (CNS) is a vSphere and Kubernetes (K8s) feature that makes K8s aware of how to provision storage on vSphere on-demand, in a fully automated, scalable fashion as well as providing visibility for the administrator into container volumes through the CNS User Interface within vCenter. Run, monitor, and manage containers and virtual machines on the same platform—in the same way:

- Simplify your infrastructure needs, lifecycle, and operations.
- Lower costs, using a platform you already know for consistent operations across workloads and across clouds.
- Spend less time managing infrastructure and more time building apps that provide business value.

The main goal of CNS is to make vSphere and vSphere storage, including vSAN, a platform to run stateful Kubernetes workloads. vSphere has a great data path that is highly reliable, highly performant, and mature for enterprise use. CNS enables access of this data path to Kubernetes and brings an understanding of Kubernetes volume and pod abstractions to vSphere. CNS was first released in vSphere 6.7 Update 3.

Test Tools

We leveraged the following monitoring and benchmark tools in this solution.

Monitoring Tools

vSAN Performance Service

vSAN Performance Service is used to monitor the performance of the vSAN environment through the vSphere Client. The performance service collects and analyzes performance statistics and displays the data in a graphical format. You can use the performance charts to manage your workload and determine the root cause of the problems.

vSAN Health Check

vSAN Health Check delivers a simplified troubleshooting and monitoring experience of all things related to vSAN. Through the vSphere client, it offers multiple health checks specifically for vSAN including cluster, hardware compatibility, data, limits, and physical disks. It is used to check the vSAN health before the mixed-workload environment deployment.

This is only for vSAN health check. We can also enable VxRail cluster health monitoring for overall health monitoring.

Application Validation Tools

Jenkins

Jenkins is an open-source automation server that lets you flexibly orchestrate your build, test, and deployment pipelines. A Kubernetes cluster adds a new automation layer to Jenkins. Kubernetes makes sure that resources are used effectively and that your servers and the underlying infrastructure are not overloaded. Kubernetes' ability to orchestrate container deployment ensures that Jenkins always has the right amount of resources available.

We used Jenkins as one of the applications for OpenShift's functional validation.

Gitlab

GitLab is the community version and a GitHub like service that organizations can use to provide internal management of git repositories. It is a self-hosted Git-repository management system that keeps the user code private and can easily deploy the changes of the code. It manages projects, not tools. With GitLab, you get an open DevOps platform delivered as a single application—one interface, one conversation thread, and one data store.

We used Gitlab as one of the applications for OpenShift's functional validation.

Refer this *Gitlab's document* for installing Gitlab on Kubernetes.

Solution Configuration

This section introduces the resources and configurations:

- Architecture diagram
- OpenShift installation
- Virtual machine placement and VMware vSphere Distributed Resource Scheduler™
- VMware vSphere High Availability
- vSAN Fault Tolerance setting
- Hardware resources
- Software resources
- Network configuration
- vSAN configuration

Architecture Diagram

The VMware Cloud Foundation test environment was composed of a management domain and a workload domain. We deployed the OpenShift Container Platform in the workload domain, and all other infrastructure VMs were in the separate management workload domain (figure 1).

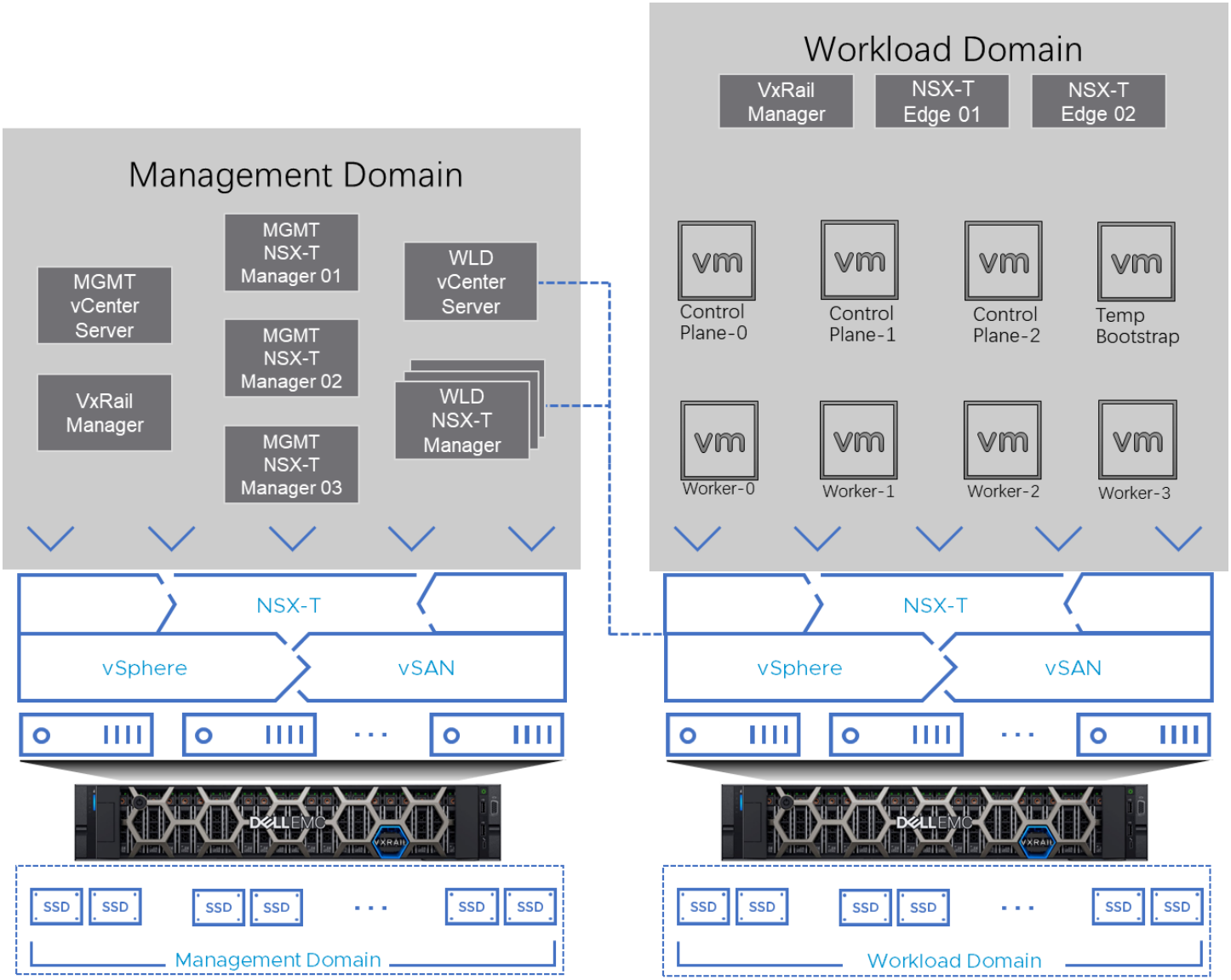

Figure 1. OpenShift on VMware Cloud Foundation Solution Architecture

Notation in Figure 1:

- Temp Bootstrap: This is the temporary bootstrap node. It can be safely deleted after the OpenShift Container Platform is fully deployed.
- Control plane-0,1,2: These are the control plane nodes of Kubernetes deployed and managed by OpenShift.
- Worker-0,1,2,3: These are the worker nodes of Kubernetes deployed and managed by OpenShift. We deployed 4 worker nodes as the starting point. More worker nodes can be added on demand through the OpenShift control plane.

In our solution, we created a 4-node VxRail P570F cluster for the VMware Cloud Foundation management domain, running management virtual machines and appliances. The management domain can be shared with other workload domains.

Table 1. Management Domain VMs

VM Role	vCPU	Memory (GB)	VM Count
Management Domain vCenter Server	4	16	1
SDDC Manager	4	16	1
Management Domain NSX-T Manager	6	24	3
Workload Domain NSX-T Manager	12	48	3
Workload Domain vCenter Server	8	28	1
VxRail Manager Appliance	2	8	1

For the workload domain, we created another 4-node VxRail P570F cluster with a separate NSX-T Fabric, deployed an NSX Edge Cluster, and deployed the OpenShift VMs in the workload domain.

Table 2 shows the deployment of the workload domain edge nodes and OpenShift VMs. For the workload domain edge node, we recommend that NSX Edge transport nodes are deployed with "Large" form factor.

Table 2. Workload Domain VMs

VM Role	Minimum vCPU	Minimum Memory (GB)	Storage	Deployment Size	VM Count
Workload Domain Edge node	8	32	200 GB	Large	2
OpenShift Control Plane Nodes	4	16	120GB for OS	n/a	3
OpenShift Compute Nodes	2	8	120GB for OS	n/a	Minimum of 2 for a standard cluster
OpenShift Bootstrap Node (Temprory)	4	16	120GB for OS	n/a	1

This is called a building block for a basic installation of OpenShift with VMware Cloud Foundation on VxRail. Based on the customer demands and database size, we can expand the workload domain to include more physical hosts. A cluster with vSAN enabled supports up to 64 physical hosts for non-stretched cluster. With adding more hosts to the vSAN cluster, not only the capacity of CPU and memory is increased for computing but also the capacity of vSAN storage is increased accordingly. This is one of the benefits of HCI that we can increase the capacity of computing and storage at the same time and proportionally.

OpenShift Installation

In this solution, we used OpenShfit's 'User Provisioned Infrastructure' (UPI) approach to install OpenShift. We followed the official *OpenShift documentation* for the installation.

The key part is that we must manually create virtual machines with vSphere and fill in their names in the corresponding OpenShift configuration YAML files.

For the network configuration, we used the VMware NSX Container Plugin. We followed the *official documentation* for the NCP installation and configuration. The sample customized ncp operator installation yaml files are in this *github page* for reference.

Pay attention to that in the OpenShift YAML file configuration, we should specify 'ncp' as the networking type similar to the following:

```

networking:
  networkType: ncp
  clusterNetwork:
  - cidr: 10.4.0.0/16
    hostPrefix: 23
  machineCIDR: 10.114.16.0/24
  serviceNetwork:
  - 172.30.0.0/16

```

Then, we followed *this documentation* for the CSI driver installation and deployment.

Also refer this *github page* for some installation scripts and caveats.

Virtual Machine Placement and vSphere DRS

VMware vSphere Distributed Resource Scheduler (DRS) is a feature included in the vSphere Enterprise Plus. In this solution, if DRS is enabled in the cluster, the rule of thumb is:

- Place the control plane nodes on three different physical hosts to accommodate one host failure. We must create DRS Anti-Affinity rules to separate the VMs to different physical hosts.
- Let DRS do the automatic placement of compute node virtual machines.

For DRS Anti-Affinity rules, see *the DRS documentation.*

VMware vSphere High Availability

We recommend enabling vSphere High Availability for the workload domain cluster.

If vSphere HA is enabled, in case of a physical host failure and there are enough remaining resources to satisfy the resource reservation like having a spare host, vSphere can automatically power on the impacted virtual machines on the other surviving hosts.

In case of a physical host failure and if there are not enough remaining resources to satisfy the resource reservation, vSphere HA would not restart the impacted virtual machines, which is by design. Because forcing a virtual machine restart on a surviving host may cause resource contention and imbalanced performance among the OpenShift nodes. We suggest that the resource reservation should at least be set to all the control plane nodes.

Hardware Resources

In this solution, for the workload domain of OpenShift, we used a total of four VxRail R570F nodes. Each server was configured with two disk groups, and each disk group consisted of one cache-tier write-intensive SAS SSD and four capacity-tier read-intensive SAS SSDs.

Each VxRail node in the cluster had the following configuration, as shown in table 3.

Table 3. Hardware Configuration for VxRail

PROPERTY	SPECIFICATION
Server model name	VxRail P570F
CPU	2 x Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz, 28 core each
RAM	512GB
Network adapter	2 x Broadcom BCM57414 NetXtreme-E 25Gb RDMA Ethernet Controller
Storage adapter	1 x Dell HBA330 Adapter
Disks	Cache - 2 x 800GB Write Intensive SAS SSDs Capacity - 8 x 3.84TB Read Intensive SAS SSDs

Software Resources

Table 4 shows the software resources used in this solution.

Table 4. Software Resources

SOFTWARE	VERSION	PURPOSE
VMware Cloud Foundation on VxRail	4.2	A unified SDDC platform on VxRail that brings together VMware vSphere, vSAN, NSX, and optionally, vRealize Suite components, into a natively integrated stack to deliver enterprise-ready cloud infrastructure for the private and public cloud. See *BOM of VMware Cloud Foundation on VxRail* for details.
Dell EMC VxRail	7.0.131	Turnkey Hyperconverged Infrastructure for hybrid cloud
VMware vSphere	7.0	VMware vSphere is a suite of products: vCenter Server and ESXi.
VMware vSAN	7.0	vSAN is the storage component in VMware Cloud Foundation to provide low-cost and high-performance next-generation HCI solutions.
NSX-T	3.1	NSX-T is the key network component in VMware Cloud Foundation on VxRail and is deployed automatically. It is designed for networking management and operation.
OpenShift	4.7	The version of OpenShift software being tested in this solution.

Network Configuration

Figure 2 shows the VMware vSphere Distributed Switch™ network configuration for OpenShift cluster in the workload domain of the VMware Cloud Foundation on VxRail. NSX-T, which underlies the vSphere infrastructure, is used for the OpenShift cluster networking. To enable external access for the

OpenShift cluster, an NSX-T edge cluster is required to deploy. Also it is required to configure the BGP peering and route distribution of the upstream network. For more details, refer to *VMware Cloud Foundation 4.2 on VxRail Planning and Preparation Guide*.

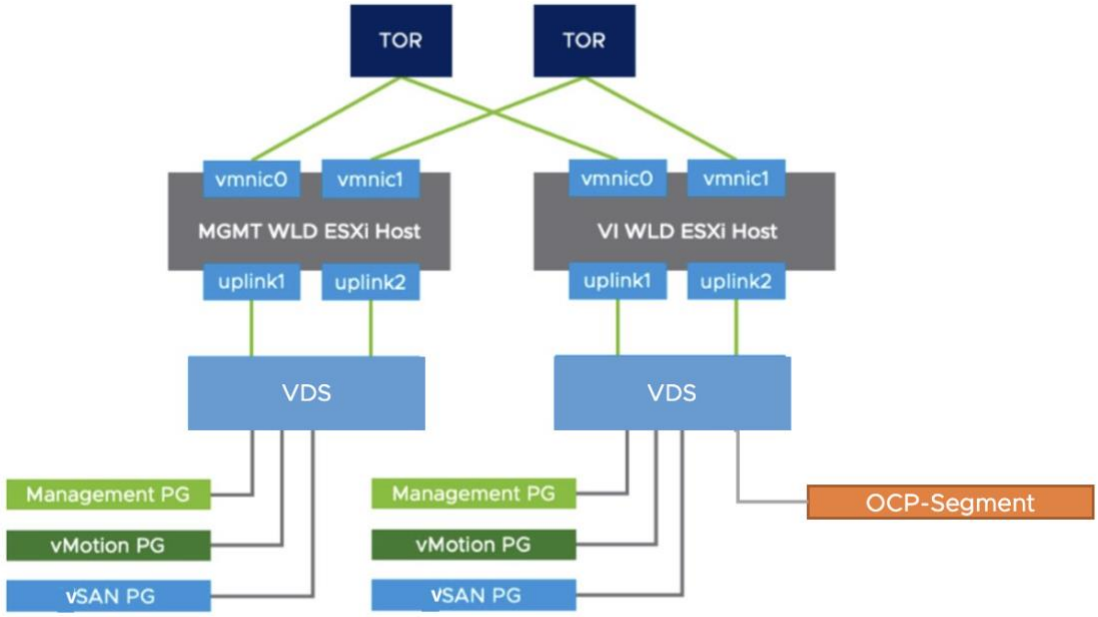

Figure 2. The Overall NSX-T Networking Architecture

Figure 2 shows the VMware vSphere Distributed Switches configuration for both the management domain and the workload domain of the VMware Cloud Foundation. For each domain, two 25 GbE vmnics were used and configured with teaming policies. The management domain can be shared among different workloads.

The NSX-T controllers resided in the management domain. The OpenShift virtual machines were configured with a VM network called 'OCP-Segment' on an NSX-T segment. VMware vSphere vMotion®, vSAN, and VXLAN VTEP for NSX-T had another dedicated segment created. In the workload domain's uplink setting, we used a dedicated physical NIC for vSAN traffic. The other physical NIC was dedicated to OpenShift virtual machines' traffic. The reason is that both vSAN and OpenShift may use high network traffic, so both of them need the dedicated NIC.

Jumbo Frame (MTU=9000) was enabled on the physical switches, vSAN VMkernel, and all the virtual switches to improve performance.

NSX-T managers and edges have more than one instance to form NSX clusters to achieve HA and better load balancing. Besides, based on workloads, the vCPU and memory may be adjusted to achieve better performance. Table 5 shows the configuration of the NSX-T managers and edge nodes virtual machines. The NSX-T managers reside in the management workload domain, so it will not cost the compute resources for OpenShift VMs. However, the NSX-T edge nodes reside in the OpenShift workload domain and it will cost some CPU and memory resources. This should be taken into consideration while doing the sizing of the cluster before OpenShift is deployed.

Table 5. NSX-T VM Configuration

NSX-T VM ROLE	INSTANCE	VCPU	MEMORY (GB)	VM NAME	VIRTUAL DISK SIZE	OPERATING SYSTEM
NSX-T Manager	3	12	48	NSX-unified-appliance-<version>	200GB	Ubuntu
NSX-T Edge Nodes	2	4	8	Edge-<UUID>	120GB	Ubuntu

vSAN Configuration

The solution validation was based on a 4-node vSAN cluster as a building block.

The validation tests were conducted using the default vSAN datastore storage policy of RAID 1 FTT=1, checksums enabled, deduplication and compression deactivated, and no encryption. In the below sections, we explained the detailed configurations of the vSAN cluster and some items in the Storage Policy Based Management (SPBM).

Deduplication and Compression

The 'Deduplication and Compression' option was configured on the cluster level and it can be enabled or deactivated for the whole vSAN cluster. While in our testing we deactivated it, by enabling it we can reduce the vSAN storage usage but induce higher latencies for the OpenShift application. This is a tradeoff for customers' choices.

Failures to Tolerance (FTT)

Failures to Tolerance (FTT) is a configuration item in vSAN's storage policy. For the 'StorageClass' in OpenShift and the corresponding vSAN's storage policy, we recommended setting vSAN's Failures to Tolerate (FTT) to 1. In our testing, we set FTT to 1 as the baseline. Do not set the FTT to 0 in an OpenShift with vSAN deployment because FTT=0 may possibly cause the data of the replications of the same pod to be stored in the same physical disk. This may cause data loss in case of a physical disk failure.

In the case of using RAID 1 in vSAN policy, there are two copies for each piece of data in vSAN. So, the estimated database capacity requirement should not exceed half of the vSAN's overall capacity. In the case of RAID 5, vSAN consumes 1.33 times of the raw capacity and you can calculate the storage usage accordingly. If the capacity increase is needed, the additional machines can be added to the cluster and vSAN can increase the data capacity storage for OpenShift online without the service interruption to OpenShift users.

Checksum

Checksum is a configuration item in vSAN's storage policy. We compared the Kubernetes performance between enabling and disabling checksum. By disabling vSAN's checksum, there is barely any performance improvement for applications deployed by OpenShift, while by enabling it, we can ensure the data is correct from the vSAN storage hardware level. So, we recommend keeping the checksum enabled, which is the default value.

Erasure Coding (RAID 1 vs. RAID 5)

Erasure Coding is a configuration item in vSAN's storage policy. It is also known as configuring RAID 5 or RAID 6 for vSAN objects. With FTT=1 and RAID 1, the data in vSAN is mirrored and the capacity cost would be 2 times of the raw capacity. With FTT=1 and RAID 5, the data is stored as RAID 5 and the capacity cost would be 1.33 times of the raw capacity.

In our testing, we used FTT=1 without Erasure Coding (RAID 1). By enabling Erasure Coding, we could save some vSAN storage spaces but induce higher latencies for the Kubernetes applications. Again, this is a tradeoff for customers' choices.

Encryption

vSAN can perform data at rest encryption. Data is encrypted after all other processing, such as deduplication. Data at rest encryption protects data on the storage devices.

Encryption is not used in our testing. Use encryption as per your company's Information Security requirements.

Failure Testing

This section introduces the failure scenarios and the behavior of failover and failback. This section includes:

- Physical host failure

- Physical cache disk failure
- Physical capacity disk failure

Physical Host Failure

We mimicked one physical host failure by directly powering off one of the physical hosts through the VxRail server iDRAC. We observed the following items:

- If there is a spare host, vSphere HA automatically powered on the impacted virtual machines on other surviving hosts. If there is no spare host or the surviving hosts do not meet the requirements of the resource reservation, the failed virtual machines would not be restarted until more resources are added to the cluster.
- Regardless of restarting the failed virtual machines or not, the host failure immediately caused the Kubernetes pod failure. If the pods' replication in Kubernetes is enabled, the failed pods can still be restarted in some surviving OpenShift worker nodes and the service can resume running.
- For data in vSAN, there were always two copies of a virtual disk created by CNS, and they must reside on different hosts. If one copy of a virtual disk resided on the impacted host, the virtual disk was marked as degraded while it was still working from the OpenShift and applications' perspective. The virtual machines and pods impacted by the failed virtual disks were still running even though the virtual disks were marked as degraded.

  - If the failed host was brought back in one hour, vSAN will resync the newly written data from other healthy hosts to this restarted host.

  - If the failed host was brought back after an hour or requires more time to recover, vSAN will rebuild the data of the degraded virtual disks to healthy hosts to ensure there are two copies of data on the surviving healthy hosts in the case of FTT=1.

Physical Cache Disk Failure

We mimicked a physical cache disk failure by injecting an error to one of the cache disks following the *vSAN 7.0 Proof of Concept Guide*.

If a cache disk fails, the vSAN disk group that contains the failed cache disk will be marked as failed. All the impacted data were immediately rebuilt to other healthy disk groups. In the meantime, all the OpenShift virtual machines, services, and pods kept running without any interruption because there was still one copy of data working from the vSAN storage level. OpenShift service was not interrupted. vSAN will intelligently control the networking traffic for data rebuilding so this only had minimum impact on the OpenShift performance.

Physical Capacity Disk Failure

We mimicked a physical capacity disk failure by injecting an error to one of the capacity disks as above.

The behavior is similar to a cache disk failure. All the OpenShift virtual machines, Kubernetes services and pods kept running without any interruption, and OpenShift service was not interrupted.

If 'Deduplication and Compression' is enabled on the vSAN cluster, the vSAN's behavior is exactly the same as a cache disk failure. The vSAN disk group that contains the failed capacity disk will be marked as failed. All the impacted data are immediately rebuilt to other healthy disk groups.

If the 'Deduplication and Compression' option is deactivated on the vSAN cluster, the difference from a cache disk failure is that a capacity disk failure would only impact the data on this specific failed disk. Only the impacted data on this failed capacity disk would be rebuilt on other healthy capacity disks.

Best Practices

- Use the same server model for the physical hosts in the workload domain.
- Follow the guidelines from *OpenShift documentation* for the detailed deployment and optimization items.
- Follow the *VMware NSX-T Container Plug-in* for the NCP installation and configuration.
- Followed the *VMware CSI driver documentation* for the CSI driver installation and deployment.

- Enable Jumbo Frame on the physical switches. Use Jumbo Frames on the vSAN VMKernel and all virtual switches.
- Set Failures to Tolerate (FTT) to at least 1 in vSAN's storage policy for data protection.
- Enable vSAN's checksum.
- Enable vSphere HA in the cluster.
- Enable vSphere DRS in the cluster.

Conclusion

VMware Cloud Foundation on VxRail delivers flexible, consistent, secure infrastructure and operations across private and public clouds. It is ideally suited to meet the demands of modern applications running on Red Hat OpenShift Container Platform in a virtualized environment.

With VMware Cloud Foundation, we can easily manage the lifecycle of the hybrid cloud environment. Besides, we have a unified management plane for all applications including OpenShift. With VMware Cloud Foundation, we can leverage the leading virtualization technologies including vSphere, NSX-T, and vSAN.

In this solution paper, we demonstrated the architecture of running OpenShift Container Platform with VMware Cloud Foundation on VxRail. We showed the configuration details, the hardware resources, and the software resources used in the solution validation. We showed the various configuration options in addition to the best practices. VxRail Manager and VMware Cloud Foundation Manager provided the lifecycle management. vSAN provides reliable, high-performance, and flexible storage to OpenShift. NSX-T provided the fine-grained, secured, and high-performance virtual networking infrastructure to OpenShift. Also, vSphere DRS and vSphere HA provided efficient resource usage and high availability. All the above lead to a consolidated solution of running OpenShift Container Platform with VMware Cloud Foundation on VxRail.

References

- *VMware Cloud Foundation*
  - *Announcing VMware Cloud Foundation 4.2*
  - *Get the Facts of VMware Cloud Foundation – Part 6*
- *VMware vSphere*
- *VMware vSAN*
- *VMware NSX Data Center*
- *Dell EMC VxRail*
- *VMware Cloud Foundation on Dell EMC VxRail Admin Guide*
- *VMware Cloud Foundation on VxRail Architecture Guide*
- *Red Hat OpenShift Container Platform*

Appendix

Sample of install-config.yaml for OpenShift User Provisioned Infrastructure Installation

https://github.com/vsphere-tmm/OpenShift-on-VMware-Cloud-Foundation-Reference-Architecture/blob/main/install-config.yaml

Sample yaml file of Jenkins Deployment during Solution Validation

https://github.com/vsphere-tmm/OpenShift-on-VMware-Cloud-Foundation-Reference-Architecture/blob/main/jenkins-deployment.yaml

About the Author

Victor (Shi) Chen, Solutions Architect in the Solutions Architecture team of the Cloud Platform Business Unit, wrote the original version of this paper.

The following reviewers also contributed to the paper contents:

- Ka Kit Wong, Staff Solutions Architect in the Solutions Architecture team of the Cloud Platform Business Unit in VMware
- William Leslie, Sr. Manager of VxRail Technical Marketing in Dell EMC
- Vic Dery, Sr. Principal Engineer of VxRail Technical Marketing in Dell EMC