Containers

NetApp Solutions

NetApp
March 06, 2021

Table of Contents

Containers

Red Hat OpenShift

NVA-1149: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization

Alan Cowles and Nikhil M Kulkarni, NetApp

NetApp HCI for Red Hat OpenShift on Red Hat Virtualization (RHV) is a best-practice deployment guide for the fully automated install of Red Hat OpenShift through the Installer Provisioned Infrastructure (IPI) method onto the verified enterprise architecture of NVA-1148: NetApp HCI with Red Hat Virtualization. The purpose of this NetApp Verified Architecture deployment guide is to provide a concise set of verified instructions to be followed for the deployment of the solution. The architecture and deployment methods described in this document have been validated jointly by subject matter experts at NetApp and Red Hat to provide a best-practice implementation of the solution.

Use Cases

The NetApp HCI for Red Hat OpenShift on RHV solution is architected to deliver exceptional value for customers with the following use cases:

- Infrastructure to scale on demand with NetApp HCI
- Enterprise virtualized workloads in RHV
- Enterprise containerized workloads in Red Hat OpenShift

Business Value

Enterprises are increasingly adopting DevOps practices to create new products, shorten release cycles, and rapidly add 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, such as the following:

- High availability at all layers in the stack
- Ease of deployment procedures
- 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

NetApp HCI for Red Hat OpenShift on RHV acknowledges these challenges and presents a solution that helps address each concern by implementing the fully automated deployment of RedHat OpenShift IPI on the RHV enterprise hypervisor. The remainder of this document details the components used in this verified architecture.

Technology Overview

NetApp HCI

NetApp HCI is an enterprise-scale, disaggregated hybrid cloud infrastructure (HCI) solution that delivers compute and storage resources in an agile, scalable, and easy-to-manage two-rack unit (2RU), four-node building block. It can also be configured with 1RU compute and server nodes. The minimum deployment depicted in the figure below consists of four NetApp HCI storage nodes and two NetApp HCI compute nodes. The compute nodes are installed as Red Hat Virtualization Hosts (RHV-H) hypervisors in a high-availability (HA) cluster. This minimum deployment can be easily scaled to fit customer enterprise workload demands by adding additional NetApp HCI storage or compute nodes to expand available resources.

[Error: Missing Graphic Image]

The design for NetApp HCI for Red Hat Virtualization consists of the following components in a minimum starting configuration:

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes running the Red Hat Virtualization RHV-H hypervisor

For more information about compute and storage nodes in NetApp HCI, see NetApp HCI Datasheet.

NetApp Trident

Trident is a NetApp open-source and fully supported storage orchestrator for containers and Kubernetes distributions, including Red Hat OpenShift. It works with the entire NetApp storage portfolio, including the NetApp Element storage system that is deployed as a part of the NetApp HCI solution. Trident provides the ability to accelerate the DevOps workflow by allowing end users to provision and manage storage from their NetApp storage systems, without requiring intervention from a storage administrator. An administrator can configure a number of storage backends based on project needs, and storage system models that allow for any number of advanced storage features, such as: compression, specific disk types, or QoS levels that guarantee a certain performance. After they are defined, these backends can be leveraged by developers as part of their projects to create persistent volume claims (PVCs) and attach persistent storage to their containers on demand.

[Error: Missing Graphic Image]

Red Hat Virtualization

RHV is an enterprise virtual data center platform that runs on Red Hat Enterprise Linux (RHEL) and uses the KVM hypervisor.

For more information about RHV, see the Red Hat Virtualization website.

RHV provides the following features:

- **Centralized management of VMs and hosts.** The RHV manager runs as a physical or virtual machine (VM) in the deployment and provides a web-based GUI for the management of the solution from a central interface.
- **Self-hosted engine.** To minimize the hardware requirements, RHV allows RHV Manager (RHV-M) to be deployed as a VM on the same hosts that run guest VMs.
- **High availability.** In event of host failures, to avoid disruption, RHV allows VMs to be configured for high availability. The highly available VMs are controlled at the cluster level using resiliency policies.
- **High scalability.** A single RHV cluster can have up to 200 hypervisor hosts enabling it to support requirements of massive VMs to hold resource-greedy, enterprise-class workloads.

- **Enhanced security.** Inherited from RHV, Secure Virtualization (sVirt) and Security Enhanced Linux (SELinux) technologies are employed by RHV for the purposes of elevated security and hardening for the hosts and VMs. The key advantage from these features is logical isolation of a VM and its associated resources.

Red Hat Virtualization Manager

RHV-M provides centralized enterprise-grade management for the physical and logical resources within the RHV virtualized environment. A web-based GUI with different role-based portals are provided to access RHV-M features.

RHV-M exposes configuration and management of RHV resources via open-source, community-driven RESTful API. It also supports full-fledged integration with Red Hat CloudForms and Red Hat Ansible for automation and orchestration.

Red Hat Virtualization Hosts

Hosts (also called hypervisors) are the physical servers that provide hardware resources for the VMs to run on. Kernel-based Virtual Machine (KVM) provides full virtualization support, and Virtual Desktop Server Manager (VDSM) is the host agent that is responsible for communication of the hosts with the RHV-M.

Two types of hosts are supported in RHV are RHV-H and RHEL hosts:

- RHV-H is a light-weight minimal operating system based on RHEL, optimized for ease of setting up physical servers as RHV hypervisors.
- RHEL hosts are servers that run the standard RHEL operating system and are later configured with the required subscriptions to install the packages required to permit the physical servers to be used as RHV hosts.

Red Hat Virtualization Architecture

RHV can be deployed in two different architectures: with the RHV-M as a physical server in the infrastructure or with the RHV-M configured as a self-hosted engine. The self-hosted engine deployment, where the RHV-M is a VM hosted in the same environment as other VMs, is recommended and used specifically in this deployment guide.

A minimum of two self-hosted nodes are required for high availability of guest VMs and RHV-M as depicted in the figure below. For ensuring the high availability of the manager VM, HA services are enabled and run on all the self-hosted engine nodes.

[Error: Missing Graphic Image]

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform is a fully supported enterprise Kubernetes platform. Red Hat makes several enhancements to open-source Kubernetes to deliver an application platform with all the components fully integrated to build, deploy, and manage containerized applications. With Red Hat OpenShift 4.4, the installation and management processes have been streamlined through the IPI method which has been deployed in this solution. By leveraging this deployment method, a fully functional OpenShift cluster providing metering and monitoring at both the cluster and application level can be fully configured and deployed on top of Red Hat Virtualization in less than an hour. OpenShift nodes are based upon RHEL CoreOS, an immutable system image designed to run containers, based on RHEL, which can be upgraded or scaled easily on demand as the needs of the end user require, helping to deliver the benefits of the public cloud to the local data center.

[Error: Missing Graphic Image]

Next: Architectural Overview: NetApp HCI for Red Hat OpenShift on RHV

Abstract

This NetApp HCI for Red Hat OpenShift on Red Hat Virtualization (RHV) deployment guide is for the fully automated installation of Red Hat OpenShift through the Installer Provisioned Infrastructure (IPI) method onto the verified enterprise architecture of NetApp HCI for Red Hat Virtualization described in NVA-1148: NetApp HCI with Red Hat Virtualization. This reference document provides deployment validation of the Red Hat OpenShift solution, integration of the NetApp Trident storage orchestrator, and a solution verification consisting of an example application deployment.

Architectural Overview: NetApp HCI for Red Hat OpenShift on RHV

Hardware Requirements

The following table lists the minimum number of hardware components that are required to implement the solution. The hardware components that are used in specific implementations of the solution might vary based on customer requirements.

Hardware	Model	Quantity
NetApp HCI compute nodes	NetApp H410C	2
NetApp HCI storage nodes	NetApp H410S	4
Data switches	Mellanox SN2010	2
Management switches	Cisco Nexus 3048	2

Software Requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

Software	Purpose	Version
NetApp HCI	Infrastructure (compute/storage)	1.8
NetApp Element	Storage	12.0
NetApp Trident	Storage orchestration	20.04
RHV	Virtualization	4.3.9
Red Hat OpenShift	Container orchestration	4.4.6

Next: Design Considerations: NetApp HCI for Red Hat OpenShift on RHV

Design Considerations: NetApp HCI for Red Hat OpenShift on RHV

Network Design

The Red Hat OpenShift on RHV on HCI solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality. OCP uses the logical network on the RHV for the cluster management. This section describes the arrangement and purpose of each virtual network segment used in the solution and outlines the pre-requisites for deployment of the solution.

VLAN Requirements

The NetApp HCI for Red Hat OpenShift on RHV solution is designed to logically separate network traffic for different purposes by using virtual local area networks (VLANs). NetApp HCI requires a minimum of three network segments. However, this configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution, as well as the specific VLAN IDs that are used later in the verified architecture deployment.

VLANs	Purpose	VLAN ID
Out-of-band management network	Management for HCI nodes and IPMI	16
In-band management network	Management for HCI nodes, ovirtmgmt, and VMs	1172
Storage network	Storage network for NetApp Element	3343
Migration network	Network for virtual guest migration	3345

Network Infrastructure Support Resources

The following infrastructure should be in place prior to the deployment of the OpenShift Container Platform (OCP) on Red Hat Virtualization on NetApp HCI solution:

- At least one DNS server which provides a full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- (Optional) Outbound internet connectivity for both the in-band management network and the VM network.
- RHV cluster should have at least 28x vCPUs, 112GB RAM, and 840GB of available storage (depending on the production workload requirements).

Next: Deploying NetApp HCI for Red Hat OpenShift on RHV

Deployment Summary: NetApp HCI for Red Hat OpenShift on RHV

The detailed steps provided in this section provide a validation for the minimum hardware and software configuration required to deploy and validate the NetApp HCI for Red Hat OpenShift on RHV solution.

Deploying Red Hat OpenShift Container Platform through IPI on Red Hat Virtualization consists of the following steps:

1. Create storage network VLAN

2. Download OpenShift installation files
3. Download CA cert from RHV
4. Register API/Apps in DNS
5. Generate and add SSH private key
6. Install OpenShift Container Platform
7. Access console/web console
8. Configure worker nodes to run storage services
9. Download and install Trident through Operator

Next: Validation Results: NetApp HCI for Red Hat OpenShift on RHV

1. Create Storage Network VLAN: NetApp HCI for Red Hat OpenShift on RHV

To create a storage network VLAN, complete the following steps:

To support Element storage access for NetApp Trident to attach persistent volumes to pods deployed in OpenShift, the machine network being used for each worker in the OCP deployment must be able to reach the storage resources. If the machine network cannot access the Element storage network by default, an additional network/VLAN can be created in the Element cluster to allow access:

1. Using any browser, log in to the Element Cluster at the cluster’s MVIP.
2. Navigate to Cluster > Network and click Create VLAN.
3. Before you provide the details, reserve at least five IP addresses from the network that is reachable from the OCP network (one for the virtual network storage VIP and one for virtual network IP on each storage node).

   Enter a VLAN name of your choice, enter the VLAN ID, SVIP, and netmask, select the Enable VRF option, and enter the gateway IP for the network. In the IP address blocks, enter the starting IP of the other addresses reserved for the storage nodes. In this example, the size is four because there are four storage nodes in this cluster. Click Create VLAN.

[Error: Missing Graphic Image]

Next: 2. Download OpenShift Installation Files

2. Download OpenShift Installation Files: NetApp HCI for Red Hat OpenShift on RHV

To download the OpenShift installation files, complete the following steps:

1. Go to the Red Hat login page and log in with your Red Hat credentials.
2. On the Clusters page, click Create Cluster.

   [Error: Missing Graphic Image]

3. Select OpenShift Container Platform.

   [Error: Missing Graphic Image]

4. Select Run on Red Hat Virtualization.

[Error: Missing Graphic Image]

5. The next page allows you to download the OpenShift installer (available for Linux and MacOS), a unique pull secret that is required to create the `install-config` file and the `oc` command-line tools (available for Linux, Windows, and MacOS).

   Download the files, transfer them to a RHEL administrative workstation from where you can run the OpenShift installation, or download these files directly using wget or curl on a RHEL administrative workstation.

[Error: Missing Graphic Image]

Next: 3. Download CA Certificate from RHV

3. Download CA Certificate from RHV: NetApp HCI for Red Hat OpenShift on RHV

To download the CA certificate from RHV, complete the following steps:

1. In order to access the RHV manager from the RHEL machine during the deployment process, the CA certificate trust must be updated on the machine to trust connections to RHV-M. To download the RHV Manager's CA certificate, run the following commands:

```
sudo curl -k 'https://<engine-fqdn>/ovirt-engine/services/pki-
resource?resource=ca-certificate&format=X509-PEM-CA' -o /tmp/ca.pem
[user@rhel7 ~]$  sudo curl -k 'https://rhv-m.cie.netapp.com/ovirt-
engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA'
-o /tmp/ca.pem
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
100  1376  100  1376    0     0   9685      0 --:--:-- --:--:-- --:--:--
9690
```

2. Copy the CA certificate to the directory for server certificates and update the CA trust.

```
[user@rhel7 ~]$  sudo cp /tmp/ca.pem /etc/pki/ca-
trust/source/anchors/ca.pem
[user@rhel7 ~]$ sudo update-ca-trust
```

Next: 4. Register API/Apps in DNS

4. Register API/Apps in DNS: NetApp HCI for Red Hat OpenShift on RHV

To register API/Apps in DNS, complete the following steps:

1. Reserve three static IP addresses from the network being used for OCP: the first IP address for OpenShift Container Platform REST API, the second IP address for pointing to the wildcard application ingress, and

the third IP address for the internal DNS service. The first two IPs require an entry in the DNS server.

> The default value of the `machineNetwork` subnet as created by IPI during OpenShift install is `10.0.0.0/16`. If the IPs you intend to use for your cluster's management network fall outside of this range, you might need to customize your deployment and edit these values before deploying the cluster. For more information, see the section Use a Custom Install File for OpenShift Deployment.

2. Configure the API domain name by using the format `api.<openshift-cluster-name>.<base-domain>` pointing to the reserved IP.

   [Error: Missing Graphic Image]

3. Configure the wildcard application ingress domain name by using the format `*.apps.<openshift-cluster-name>.<base-domain>` pointing to the reserved IP.

   [Error: Missing Graphic Image]

Next: 5. Generate and Add SSH Private Key

5. Generate and Add SSH Private Key: NetApp HCI for Red Hat OpenShift on RHV

To generate and add an SSH private key, complete the following steps:

1. For the installation debugging or disaster recovery on the OpenShift cluster, you must provide an SSH key to both the `ssh-agent` and the installation program. Create an SSH key if one does not already exist for password-less authentication on the RHEL machine.

```
[user@rhel7 ~]$  ssh-keygen -t rsa -b 4096 -N '' -f ~/.ssh/id_rsa
```

2. Start the `ssh-agent` process and configure it as a background running task.

```
[user@rhel7 ~]$  eval "$(ssh-agent -s)"
Agent pid 31874
```

3. Add the SSH private key that you created in step 2 to the `ssh-agent` , which enables you to SSH directly to the nodes without having to interactively pass the key.

```
[user@rhel7 ~]$  ssh-add ~/.ssh/id_rsa
```

Next: 6. Install OpenShift Container Platform

6. Install OpenShift Container Platform: NetApp HCI for Red Hat OpenShift on RHV

To install OpenShift Container Platform, compete the following steps:

1. Create a directory for OpenShift installation and transfer the downloaded files to it. Extract the OpenShift

installer files from the tar archive.

```
[user@rhel7 ~]$ mkdir openshift-deploy
[user@rhel7 ~]$ cd openshift-deploy
[user@rhel7 openshift-deploy]$ tar xvf openshift-install-linux.tar.gz
README.md
openshift-install
[user@rhel7 openshift-deploy]$ ls -la
total 453260
drwxr-xr-x.  2 user user       146 May 26 16:01 .
dr-xr-x---. 16 user user      4096 May 26 15:58 ..
-rw-r--r--.  1 user user  25249648 May 26 15:59 openshift-client-
linux.tar.gz
-rwxr-xr-x.  1 user user 354664448 Apr 27 01:37 openshift-install
-rw-r--r--.  1 user user  84207215 May 26 16:00 openshift-install-
linux.tar.gz
-rw-r--r--.  1 user user      2736 May 26 15:59 pull-secret.txt
-rw-r--r--.  1 user user       706 Apr 27 01:37 README.md
```

The installation program creates several files in the directory used for installation of the cluster. Both the installation program and the files created by the installation program must be kept even after the cluster is up.

The binary files that you previously downloaded, such as `openshift-install` or `oc`, can be copied to a directory that is in the user's path (for example, `/usr/local/bin`) to make them easier to run.

2. Create the cluster by running the `openshift-install create cluster` command and respond to the installation program prompts. Pass the SSH public key, select ovirt from the platform, provide the RHV infrastructure details, provide the three reserved IP addresses and the downloaded pull secret to the installation program prompts. After all the inputs are provided, the installation program creates and configures a bootstrap machine with a temporary Kubernetes control plane which then creates and configures the master VMs with the production Kubernetes control plane. The control plane on the master nodes creates and configures the worker VMs.

   It can take approximately 30–45 minutes to get the complete cluster up and running.

```
[user@rhel7 openshift-deploy]$ ./openshift-install create cluster
--dir=/home/user/openshift-deploy --log-level=info                    ?
SSH Public Key /home/user/.ssh/id_rsa.pub
? Platform ovirt
? oVirt cluster Default
? oVirt storage domain data_domain
? oVirt network ovirtmgmt
? Internal API virtual IP 10.63. 172.151
? Internal DNS virtual IP 10.63. 172.153
? Ingress virtual IP 10.63. 172.152
? Base Domain cie.netapp.com
? Cluster Name rhv-ocp-cluster
? Pull Secret [? for help]
************************************************************************
************************************************************************
************************************************************************
************************************************************************
********
INFO Obtaining RHCOS image file from 'https://releases-art-
rhcos.svc.ci.openshift.org/art/storage/releases/rhcos-
4.4/44.81.202004250133-0/x86_64/rhcos-44.81.202004250133-0-
openstack.x86_64.qcow2.gz?sha256=f8a44e0ea8cc45882dc22eb632a63afb90b4148
39b8aa92f3836ede001dfe9cf'
INFO The file was found in cache: /home/user/.cache/openshift-
installer/image_cache/e263efbc53c0caf612bcfaad10e3dff0. Reusing...
INFO Creating infrastructure resources...
INFO Waiting up to 20m0s for the Kubernetes API at https://api.rhv-ocp-
cluster.cie.netapp.com:6443...
INFO API v1.17.1 up
INFO Waiting up to 40m0s for bootstrapping to complete...
INFO Destroying the bootstrap resources...
INFO Waiting up to 30m0s for the cluster at https://api.rhv-ocp-
cluster.cie.netapp.com:6443 to initialize...
INFO Waiting up to 10m0s for the openshift-console route to be
created...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run
'export KUBECONFIG=/home/user/openshift-deploy/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.rhv-ocp-cluster.cie.netapp.com
INFO Login to the console with user: kubeadmin, password: NtsqU-p3qUb-
8Hscu-JfAq7
```

3. When the cluster deployment is complete, the directions for accessing the OpenShift cluster, including a link to its web console and credentials for the kubeadmin user, are displayed. Make sure to take a note of

these details.

4. Log in to the RHV Manager and observe that the VMs relating to the OCP cluster are up and running.

[Error: Missing Graphic Image]

Next: 7. Access Console/Web Console

7. Access Console/Web Console: NetApp HCI for Red Hat OpenShift on RHV

To access the console or web console, complete the following steps:

1. To access the OCP cluster through the CLI, extract the `oc` command-line tools tar file and place its content in a directory that is in the user's path.

```
[user@rhel7 openshift-deploy]$ tar xvf openshift-client-linux.tar.gz
README.md
oc
kubectl
[user@rhel7 openshift-deploy]$ echo $PATH
/usr/local/bin: /usr/local/sbin:/sbin:/bin:/usr/sbin:/usr/bin

[user@rhel7 openshift-deploy]$ cp oc /usr/local/bin
```

2. To interact with the cluster through the CLI, you can use the `kubeconfig` file provided by the IPI process located in the `/auth` directory inside the folder from where you launched the installation program. To easily interact with the cluster, export the file that is created in the directory. After a successful cluster deployment, the file location and the following command are displayed.

```
[user@rhel7 openshift-deploy]$ export KUBECONFIG=/home/user/openshift-
deploy/auth/kubeconfig
```

3. Verify whether you have access to the cluster and whether the nodes are in the Ready state.

```
[user@rhel7 openshift-deploy]$ oc get nodes
NAME                                 STATUS   ROLES    AGE   VERSION
rhv-ocp-cluster-hdr7k-master-0       Ready    master   93m   v1.17.1
rhv-ocp-cluster-hdr7k-master-1       Ready    master   93m   v1.17.1
rhv-ocp-cluster-hdr7k-master-2       Ready    master   93m   v1.17.1
rhv-ocp-cluster-hdr7k-worker-0-ghskz Ready    worker   83m   v1.17.1
rhv-ocp-cluster-hdr7k-worker-0-xdl99 Ready    worker   86m   v1.17.1
rhv-ocp-cluster-hdr7k-worker-0-zkxmt Ready    worker   85m   v1.17.1
```

4. Log in to the web console URL by using the credentials, both of which were provided after the successful deployment of the cluster, and then verify GUI access to the cluster.

[Error: Missing Graphic Image]

Next: 8. Configure Worker Nodes to Run Storage Services

8. Configure Worker Nodes to Run Storage Services: NetApp HCI for Red Hat OpenShift on RHV

To configure the worker nodes to run storage services, complete the following steps:

1. To access storage from the Element system, each of the worker nodes must have iSCSI available and running as a service. To create a machine configuration that can enable and start the iscisd service, log in to the OCP web console and navigate to Compute > Machine Configs and click Create Machine Config. Paste the YAML file and click Create.

   [Error: Missing Graphic Image]

2. After the configuration is created, it will take approximately 20–30 minutes to apply the configuration to the worker nodes and reload them. Verify whether the machine config is applied by using `oc get mcp` and make sure that the machine config pool for workers is updated. You can also log in to the worker nodes to confirm that the iscsid service is running.

```
[user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                    UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False
[user@rhel7 openshift-deploy]$ ssh core@10.63. 172.22 sudo systemctl
status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
vendor preset: disabled)
   Active: active (running) since Tue 2020-05-26 13:36:22 UTC; 3 min ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
 Main PID: 1242 (iscsid)
   Status: "Ready to process requests"
    Tasks: 1
   Memory: 4.9M
      CPU: 9ms
   CGroup: /system.slice/iscsid.service
           └─1242 /usr/sbin/iscsid -f
```

It is also possible to confirm that the MachineConfig has been successfully applied and services have been started as expected by running the `oc debug` command with the appropriate flags.

Next: 9. Download and Install NetApp Trident

To download and install NetApp Trident, complete the following steps:

1. Make sure that the user that is logged in to the OCP cluster has sufficient privileges for installing Trident.

```
[user@rhel7 openshift-deploy]$ oc auth can-i '*' '*' --all-namespaces
 yes
```

2. Verify that you can download an image from the registry and access the MVIP of the NetApp Element cluster.

```
[user@rhel7 openshift-deploy]$ oc run -i --tty ping --image=busybox
--restart=Never --rm -- ping 10.63.172.140
If you don't see a command prompt, try pressing enter.
64 bytes from 10.63.172.140: seq=1 ttl=63 time=0.312 ms
64 bytes from 10.63.172.140: seq=2 ttl=63 time=0.271 ms
64 bytes from 10.63.172.140: seq=3 ttl=63 time=0.254 ms
64 bytes from 10.63.172.140: seq=4 ttl=63 time=0.309 ms
64 bytes from 10.63.172.140: seq=5 ttl=63 time=0.319 ms
64 bytes from 10.63.172.140: seq=6 ttl=63 time=0.303 ms
^C
--- 10.63.172.140 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.254/0.387/0.946 ms
pod "ping" deleted
```

3. Download the Trident installer bundle using the following commands and extract it to a directory.

```
[user@rhel7 ~]$ wget
[user@rhel7 ~]$ tar -xf trident-installer-20.04.0.tar.gz
[user@rhel7 ~]$ cd trident-installer
```

4. The Trident installer contains manifests for defining all the required resources. Using the appropriate manifests, create the TridentProvisioner custom resource definition.

```
[user@rhel7 trident-installer]$ oc create -f
deploy/crds/trident.netapp.io_tridentprovisioners_crd_post1.16.yaml

customresourcedefinition.apiextensions.k8s.io/tridentprovisioners.triden
t.netapp.io created
```

5. Create a Trident namespace, which is required for the Trident operator.

```
[user@rhel7 trident-installer]$ oc create namespace trident
namespace/trident created
```

6. Create the resources required for the Trident operator deployment, such as a ServiceAccount for the operator, a ClusterRole and ClusterRoleBinding to the ServiceAccount, a dedicated PodSecurityPolicy, or the operator itself.

```
[user@rhel7 trident-installer]$ oc kustomize deploy/ >
deploy/bundle.yaml
[user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

7. Verify that the Trident operator is deployed.

```
[user@rhel7 trident-installer]$ oc get deployment -n trident
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator    1/1     1            1           56s
[user@rhel7 trident-installer]$ oc get pods -n trident
NAME                              READY   STATUS    RESTARTS   AGE
trident-operator-564d7d66f-qrz7v  1/1     Running   0          71s
```

8. After the Trident operator is installed, install Trident using this operator. In this example, TridentProvisioner custom resource (CR) was created. The Trident installer comes with definitions for creating a TridentProvisioner CR. These can be modified based on the requirements.

```
[user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentprovisioner_cr.yaml
tridentprovisioner.trident.netapp.io/trident created
```

9. Approve the Trident serving CSR certificates by using `oc get csr -o name | xargs oc adm certificate approve`.

```
[user@rhel7 trident-installer]$ oc get csr -o name | xargs oc adm
certificate approve
certificatesigningrequest.certificates.k8s.io/csr-4b7zh approved
certificatesigningrequest.certificates.k8s.io/csr-4hkwc approved
certificatesigningrequest.certificates.k8s.io/csr-5bgh5 approved
certificatesigningrequest.certificates.k8s.io/csr-5g4d6 approved
certificatesigningrequest.certificates.k8s.io/csr-5j9hz approved
certificatesigningrequest.certificates.k8s.io/csr-5m8qb approved
certificatesigningrequest.certificates.k8s.io/csr-66hv2 approved
certificatesigningrequest.certificates.k8s.io/csr-6rdgg approved
certificatesigningrequest.certificates.k8s.io/csr-6t24f approved
certificatesigningrequest.certificates.k8s.io/csr-76wgv approved
certificatesigningrequest.certificates.k8s.io/csr-78qsq approved
certificatesigningrequest.certificates.k8s.io/csr-7r58n approved
certificatesigningrequest.certificates.k8s.io/csr-8ghmk approved
certificatesigningrequest.certificates.k8s.io/csr-8sn5q approved
```

10. Verify that Trident 20.04 is installed by using the TridentProvisioner CR, and verify that the pods related to Trident are.

```
[user@rhel7 trident-installer]$ oc get tprov -n trident
NAME      AGE
trident   9m49s

[user@rhel7 trident-installer]$ oc describe tprov trident -n trident
Name:         trident
Namespace:    trident
Labels:       <none>
Annotations:  <none>
API Version:  trident.netapp.io/v1
Kind:         TridentProvisioner
Metadata:
  Creation Timestamp:  2020-05-26T18:49:19Z
  Generation:          1
  Resource Version:    640347
  Self Link:
/apis/trident.netapp.io/v1/namespaces/trident/tridentprovisioners/triden
t
  UID:                 52656806-0414-4ed8-b355-fc123fafbf4e
Spec:
  Debug:  true
Status:
  Message:  Trident installed
  Status:   Installed
  Version:  v20.04
```

```
Events:
  Type    Reason      Age                     From
Message
  ----    ------      ----                    ----
-------
  Normal  Installing  9m32s                   trident-operator.netapp.io
Installing Trident
  Normal  Installed   3m47s (x5 over 8m56s)  trident-operator.netapp.io
Trident installed
[user@rhel7 trident-installer]$ oc get pods -n trident
NAME                               READY   STATUS    RESTARTS   AGE
trident-csi-7f769c7875-s6fmt       5/5     Running   0          10m
trident-csi-cp7wg                  2/2     Running   0          10m
trident-csi-hhx94                  2/2     Running   0          10m
trident-csi-l72bt                  2/2     Running   0          10m
trident-csi-xfl9d                  2/2     Running   0          10m
trident-csi-xrhqx                  2/2     Running   0          10m
trident-csi-zb7ws                  2/2     Running   0          10m
trident-operator-564d7d66f-qrz7v   1/1     Running   0          27m

[user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 20.04.0        | 20.04.0        |
+----------------+----------------+
```

11. Create a storage backend that will be used by Trident to provision volumes. The storage backend specifies the Element cluster in NetApp HCI. You also can specify sample bronze, silver, and gold types with corresponding QoS specs.

```
[user@rhel7 trident-installer]$ vi backend.json
{
    "version": 1,
    "storageDriverName": "solidfire-san",
    "Endpoint": "https://admin: admin- password@10.63.172.140/json-
rpc/8.0",
    "SVIP": "10.61.185.205:3260",
    "TenantName": "trident",
    "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS":
2000, "burstIOPS": 4000}},
              {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS":
6000, "burstIOPS": 8000}},
              {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS": 8000,
"burstIOPS": 10000}}]
}
[user@rhel7 trident-installer]$ ./tridentctl -n trident create backend
-f backend.json
+-------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME           | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+-------------------------+----------------
+--------------------------------------+--------+---------+
| solidfire_10.61.185.205 | solidfire-san  | 40f48d99-5d2e-4f6c-89ab-
8aee2be71255 | online |       0 |
+-------------------------+----------------
+--------------------------------------+--------+---------+
```

Modify the `backend.json` to accommodate the details or requirements of your environment for the following values:

- Endpoint corresponds to the credentials and the MVIP of the NetApp HCI Element cluster.
- SVIP corresponds to the SVIP configured over the VM network in the section titled Create Storage Network VLAN.
- Types corresponds to different QoS bands. New persistent volumes can be created with specific QoS settings by specifying the exact storage pool.

12. Create a StorageClass that specifies Trident as the provisioner and the storage backend as `solidfire-san`.

```
[user@rhel7 trident-installer]$ vi storage-class-basic.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
  provisioningType: "thin"

[user@rhel7 trident-installer]$ oc create -f storage-class-basic.yaml
storageclass.storage.k8s.io/basic created
```

In this example, the StorageClass created is set as a default, however an OpenShift administrator can define multiple storage classes corresponding to different QoS requirements and other factors based upon their applications. Trident selects a storage backend that can satisfy all the criteria specified in the parameters section in the storage class definition. End users can then provision storage as needed, without administrative intervention.

Next: Validation Results: NetApp HCI for Red Hat OpenShift on RHV

Validation Results: NetApp HCI for Red Hat OpenShift on RHV

This section provides the steps to deploy a continuous integration/continuous delivery or deployment (CI/CD) pipeline with Jenkins in order to validate the operation of the solution.

Create the Resources Required for Jenkins Deployment

To create the resources required for deploying the Jenkins application, complete the following steps:

1. Create a new project named Jenkins.

   [Error: Missing Graphic Image]

2. In this example, we deployed Jenkins with persistent storage. To support the Jenkins build, create the PVC. Navigate to Storage > Persistent Volume Claims and click Create Persistent Volume Claim. Select the storage class that was created, make sure that the Persistent Volume Claim Name is jenkins, select the appropriate size and access mode, and then click Create.

[Error: Missing Graphic Image]

Deploy Jenkins with Persistent Storage

To deploy Jenkins with persistent storage, complete the following steps:

1. In the upper left corner, change the role from Administrator to Developer. Click +Add and select From Catalog. In the Filter by Keyword bar, search jenkins. Select Jenkins Service, with Persistent Storage.

   [Error: Missing Graphic Image]

2. Click Instantiate Template.

   [Error: Missing Graphic Image]

3. By default, the details for the Jenkins application are populated. Based on your requirements, modify the parameters, and click Create. This process creates all the required resources for supporting Jenkins on OpenShift.

   [Error: Missing Graphic Image]

4. The Jenkins pods take approximately 10–12 minutes to enter the Ready state.

   [Error: Missing Graphic Image]

5. After the pods are instantiated, navigate to Networking > Routes. To open the Jenkins webpage, click the URL provided for the jenkins route.

   [Error: Missing Graphic Image]

6. Because the OpenShift OAuth was used while creating the Jenkins app, click Log in with OpenShift.

   [Error: Missing Graphic Image]

7. Authorize jenkins service-account to access the OpenShift users.

   [Error: Missing Graphic Image]

8. The Jenkins welcome page is displayed. Because we are using a Maven build, complete the Maven installation first. Navigate to Manage Jenkins > Global Tool Configuration, then in the Maven subhead, click Add Maven. Enter the name of your choice and make sure that the Install Automatically option is selected. Click Save.

   [Error: Missing Graphic Image]

9. You can now create a pipeline to demonstrate the CI/CD workflow. On the home page, click Create New Jobs or New Item from the left- hand menu.

   [Error: Missing Graphic Image]

10. On the Create Item page, enter the name of your choice, select Pipeline, and click Ok.

    [Error: Missing Graphic Image]

11. Select the Pipeline tab. From the Try Sample Pipeline drop- down menu, select Github + Maven. The code is automatically populated. Click Save.

    [Error: Missing Graphic Image]

12. Click Build Now to trigger the development through the preparation, build, and testing phase. It can take several minutes to complete the whole build process and display the results of the build.

[Error: Missing Graphic Image]

13. Whenever there are any code changes, the pipeline can be rebuilt to patch the new version of software enabling continuous integration and continuous delivery. Click Recent Changes to track the changes from the previous version.

    [Error: Missing Graphic Image]

Next: Best Practices for Production Deployments

Best Practices for Production Deployments - NetApp HCI for Red Hat OpenShift on RHV

This section lists several best practices that an organization should take into consideration before deploying this solution into production.

Deploy OpenShift to an RHV Cluster of at Least Three Nodes

The verified architecture described in this document presents the minimum hardware deployment suitable for HA operations by deploying two RHV-H hypervisor nodes and ensuring a fault tolerant configuration where both hosts can manage the hosted-engine and deployed VMs can migrate between the two hypervisors. Because Red Hat OpenShift initially deploys with three master nodes, it is ensured in a two-node configuration that at least two masters will occupy the same node, which can lead to a possible outage for OpenShift if that specific node becomes unavailable. Therefore, it is a Red Hat best practice that at least three RHV-H hypervisor nodes be deployed as part of the solution so that the OpenShift masters can be distributed evenly, and the solution receives an added degree of fault tolerance.

Configure Virtual Machine/Host Affinity

Ensuring the distribution of the OpenShift masters across multiple hypervisor nodes can be achieved by enabling VM/host affinity. Affinity is a way to define rules for a set of VMs and/or hosts that determine whether the VMs run together on the same host or hosts in the group or on different hosts. It is applied to VMs by creating affinity groups that consist of VMs and/or hosts with a set of identical parameters and conditions. Depending on whether the VMs in an affinity group run on the same host or hosts in the group or separately on different hosts, the parameters of the affinity group can define either positive affinity or negative affinity. The conditions defined for the parameters can be either hard enforcement or soft enforcement. Hard enforcement ensures that the VMs in an affinity group always follows the positive/negative affinity strictly without any regards to external conditions. Soft enforcement, on the other hand, ensures that a higher preference is set out for the VMs in an affinity group to follow the positive/negative affinity whenever feasible. In a two or three hypervisor configuration as described in this document soft affinity is the recommended setting, in larger clusters hard affinity can be relied on to ensure OpenShift nodes are distributed. To configure affinity groups, see the Red Hat 6.11. Affinity Groups documentation.

Use a Custom Install File for OpenShift Deployment

IPI makes the deployment of OpenShift clusters extremely easy through the interactive wizard discussed earlier in this document. However, it is possible that there are some default values that might need to be changed as a part of a cluster deployment. In these instances, the wizard can be run and tasked without immediately deploying a cluster, but instead outputting a configuration file from which the cluster can be deployed later. This is very useful if any IPI defaults need to be changed, or if a user wants to deploy multiple identical clusters in their environment for other uses such as multitenancy. For more information about creating a customized install configuration for OpenShift, see Red Hat OpenShift Installing a Cluster on RHV with Customizations.

Next: Videos and Demos: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization

Videos and Demos: NetApp HCI for Red Hat OpenShift on RHV

The following video demonstrates some of the capabilities documented in this document:

[] | *NetApp HCI for Red Hat OpenShift on Red Hat Virtualization*

Next: Additional Information: NetApp HCI for Red Hat OpenShift on Red Hat Virtualization

Additional Information: NetApp HCI for Red Hat OpenShift on RHV

To learn more about the information described in this document, review the following websites:

- NetApp HCI Documentation https://www.netapp.com/us/documentation/hci.aspx
- NetApp Trident Documentation https://netapp-trident.readthedocs.io/en/stable-v20.04/
- Red Hat Virtualization Documentation https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.3/
- Red Hat OpenShift Documentation https://access.redhat.com/documentation/en-us/openshift_container_platform/4.4/

Google Anthos

NVA-1141: NetApp HCI with Anthos, Design and Deployment

Alan Cowles

The program solutions described in this document are designed and thoroughly tested to minimize deployment risks and accelerate time to market.

This document is for NetApp and partner solutions engineers and customer strategic decision makers. It describes the architecture design considerations that were used to determine the specific equipment, cabling, and configurations required to support the validated workload.

NetApp HCI with Anthos is a verified, best-practice hybrid cloud architecture for the deployment of an on-premises Google Kubernetes Engine (GKE) environment in a reliable and dependable manner. This NetApp Verified Architecture reference document serves as both a design guide and a deployment validation of the Anthos solution on NetApp HCI. The architecture described in this document has been validated by subject matter experts at NetApp and Google to provide the advantage of running Anthos on NetApp HCI within your own enterprise data-center environment.

NetApp HCI, is the industry’s first and leading disaggregated hybrid cloud infrastructure, providing the widely recognized benefits of hyperconverged solutions. Benefits include lower TCO and ease of acquisition, deployment, and management for virtualized workloads, while also allowing enterprise customers to independently scale compute and storage resources as needed. NetApp HCI with Anthos provides an on-premises, cloud-like experience for the deployment of containerized workloads managed by Anthos GKE on-premises. This solution provides simplified management, detailed metrics, and a range of additional functionalities that enable the easy movement of workloads deployed both on-site and in the cloud.

Features

With NetApp HCI for Anthos, you can deploy a fully integrated, production-grade Anthos GKE environment in your on-premises data center, which allows you to take advantage of the following features:

- NetApp HCI compute and storage nodes
  - Enterprise-grade hyperconverged infrastructure designed for hybrid cloud workloads
  - NetApp Element storage software
  - Intel-based server compute nodes, including options for Nvidia GPUs
- VMware vSphere 6.5
  - Enterprise hypervisor solution for deployment and management of virtual infrastructures
- Anthos GKE in Google Cloud and On-Prem
  - Deploy Anthos GKE instances in Google Cloud or on NetApp HCI

The NetApp Verified Architecture program gives customers reference configurations and sizing guidance for specific workloads and use cases.

Next: Solution Components

Solution Components

The solution described in this document builds on the solid foundation of NetApp HCI, VMware vSphere, and the Anthos hybrid-cloud Kubernetes data center solution.

NetApp HCI

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 of both compute and storage resources, 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 storage quality-of-service (QoS) limits that are available natively with NetApp Element® software. Element enables the granular control of every application and volume, helps to eliminate noisy neighbors, and satisfies enterprise performance SLAs. NetApp HCI multitenancy capabilities can help eliminate many traditional performance-related problems.
- **Flexible.** Previous generations of hyperconverged infrastructures often required fixed resource ratios, limiting deployments to four-node and eight-node configurations. NetApp HCI is a disaggregated hyper-converged infrastructure that can scale 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. NetApp HCI is available in mix-and-match small, medium, and large storage and compute configurations.
  The architectural design choices offered enable you to confidently scale on your terms, making HCI viable for core Tier 1 data center applications and platforms. NetApp HCI is architected in building blocks at either the chassis or the node level. Each chassis can hold four nodes in a mixed configuration of storage or compute nodes.
- **Simple.** A driving imperative within the IT community is to simplify deployment and automate 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 both simplifying deployment and ongoing management. The NetApp Deployment Engine (NDE) tool eases the configuration and deployment of physical infrastructure, including the installation of the VMware vSphere environment and the integration of the NetApp Element Plug-in for vCenter Server. With NDE, future scaling operations can be performed without difficulty.

NetApp HCI Configuration

NetApp HCI is an enterprise-scale disaggregated hybrid cloud infrastructure (HCI) solution that delivers compute and storage resources in an agile, scalable, and easy-to-manage two-rack unit (2RU) four-node building block. It can also be configured with 1RU compute and server nodes. The minimum deployment consists of four NetApp HCI storage nodes and two NetApp HCI compute nodes. The compute nodes are installed as VMware ESXi hypervisors in an HA cluster without the enforcement of VMware DRS anti-affinity rules. This minimum deployment can be easily scaled to fit customer enterprise workload demands by adding additional NetApp HCI storage or compute nodes to expand available storage.

[NetApp HCI minimum Configuration] | *netapp_hci_min_config.png*

Figure 1. NetApp HCI minimum configuration.

The design for NetApp HCI for Anthos consists of the following components in a minimum starting configuration:

- NetApp H-Series all-flash storage nodes running NetApp Element software
- NetApp H-Series compute nodes running VMware vSphere 6.5U2

For more information about compute and storage nodes in NetApp HCI, see the NetApp HCI Datasheet.

NetApp Element Software

NetApp Element software provides modular, scalable performance, with each storage node delivering guaranteed capacity and throughput to the environment. You can also specify per-volume storage QoS policies to support dedicated performance levels for even the most demanding workloads.

iSCSI Login Redirection and Self-Healing Capabilities

NetApp Element software uses the iSCSI storage protocol, a standard way to encapsulate SCSI commands on a traditional TCP/IP network. When SCSI standards change or when Ethernet network performance 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 in the cluster. As a part of the iSCSI login process, 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 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, a 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. The following 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 specific volume.
- **Burst IOPS.** The maximum number of IOPS allowed in a short burst scenario. The burst duration setting is configurable, with a default of 1 minute. If a volume has been running below the maximum IOPS level, burst credits are accumulated. When performance levels become very high and are pushed, short bursts of IOPS beyond the maximum IOPS are allowed on the volume.

Multitenancy

Secure multitenancy is achieved with 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.
- **VPN routing/forwarding (VFR)-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 service provider environments where scale and preservation of IP-space are important.

Enterprise Storage Efficiencies

The NetApp Element software cluster increases 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 only stores 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 Element 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, stored in 4K blocks, and 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 mentioned above can be managed with APIs. These APIs are the only method that the UI uses to control the system whether actions are performed directly through Element or through the vSphere plug-in for Element.

VMware vSphere

VMware vSphere is the industry leading virtualization solution built on VMware ESXi hypervisors and managed by vCenter Server, which provides advanced functionality often required for enterprise datacenters. When using the NDE with NetApp HCI, a VMware vSphere environment is configured and installed. The following features are available after the environment is deployed:

- **Centralized Management.** Through vSphere, individual hypervisors can be grouped into data centers and combined into clusters, allowing for advanced organization to ease the overall management of resources.
- **VMware HA.** This feature allows virtual guests to restart automatically if their host becomes unavailable. By enabling this feature, virtual guests become fault tolerant, and virtual infrastructures experience minimal disruption when there are physical failures in the environment.
- **VMware Distributed Resource Scheduler (DRS).** VMware vMotion allows for the movement of guests between hosts nondisruptively when certain user-defined thresholds are met. This capability makes the virtual guests in an environment highly available.
- **vSphere Distributed Switch (vDS).** A virtual switch is controlled by the vCenter server, enabling centralized configuration and management of connectivity for each host by creating port groups that map to the physical interfaces on each host.

Anthos

Anthos is a hybrid-cloud Kubernetes data center solution that enables organizations to construct and manage modern hybrid-cloud infrastructures, while adopting agile workflows focused on application development. Anthos on VMware, a solution built on open-source technologies, runs on-premises in a VMware vSphere-based infrastructure, which can connect and interoperate with Anthos GKE in Google Cloud.
Adopting containers, service mesh, and other transformational technologies enables organizations to experience consistent application development cycles and production-ready workloads in local and cloud-based environments. The following figure depicts the Anthos solution and how a deployment in an on-premises data center interconnects with infrastructure in the cloud.
For more information about Anthos, see the Anthos website located here.

Anthos provides the following features:

- **Anthos configuration management.** Automates the policy and security of hybrid Kubernetes deployments.
- **Anthos Service Mesh.** Enhances application observability, security, and control with an Istio-powered service mesh.
- **Google Cloud Marketplace for Kubernetes Applications.** A catalog of curated container applications available for easy deployment.
- **Migrate for Anthos.** Automatic migration of physical services and VMs from on-premises to the cloud.
- **Stackdriver.** Management service offered by Google for logging and monitoring cloud instances.

[Anthos Architecture.] | *anthos_architecture.png*

Figure 2. Anthos architecture.

Containers and Kubernetes Orchestration

Container technology has been available to developers for a long time. However, it has only recently become a core concept in data center architecture and design as more enterprises have adopted application-specific workload requirements.

A traditional development environment requires a dedicated development host deployed on either a bare-metal or virtual server. Such environments require each application to have its own dedicated machine, complete with operating system (OS) and networking connectivity. These machines often must be managed by the enterprise system administration team, who must account for the application versions installed as well as host OS patches. In contrast, containers by design require less overhead to deploy. All that is needed is the packaging of application code and supporting libraries together, because all other services depend on the host OS. Rather than managing a complete virtual machine (VM) environment, developers can instead focus on the application development process.

As container technology began to find appeal in the enterprise landscape, many enterprise features, such as fault tolerance and application scaling, were both requested and expected. In response, Google partnered with the Linux Foundation to form the Cloud Native Computing Foundation (CNCF). Together, they introduced Kubernetes (K8s), an open-source platform for orchestrating and managing containers. Kubernetes was designed by Google to be a successor to both the Omega and Borg container management platforms that had been used in their data centers in the previous decade.

Anthos GKE

Anthos GKE is a certified distribution of Kubernetes in the Google Cloud. It allows end users to easily deploy managed, production-ready Kubernetes clusters, enabling developers to focus primarily on application development rather than on the management of their environment. Deploying Kubernetes clusters in Anthos GKE offers the following benefits:

- **Simplifying Deployment of Applications.** Anthos GKE allows for rapid development, deployment, and updates of applications and services. By providing simple descriptions of the expected system resources (compute, memory, and storage) required by the application containers, the Kubernetes Engine automatically provisions and manages the lifecycle of the cluster environment.
- **Ensuring Availability of Clusters.** The environment is made extremely accessible and easy to manage by using the dashboard built into the Google Cloud console. Anthos GKE clusters are continually monitored by Google Site Reliability Engineers (SREs) to make sure that clusters behave as expected by collecting regular metrics and observing the use of assigned system resources. A user can also leverage available health checks to make sure that their deployed applications are highly available and that they can recover easily should something go awry.
- **Securing Clusters in Google Cloud.** An end user can ensure that clusters are secure and accessible by customizing network policies available from Google Cloud’s Global Virtual Private Cloud. Public services can be placed behind a single global IP address for load balancing purposes. A single IP can help provide high availability for applications and protect against Distributed Denial of Service (DDOS) and other forms of attacks that might hinder service performance.
- **Easily Scaling to Meet Requirements.** An end user can enable auto-scaling on their cluster to easily counter both planned and unexpected increases in application demands. Auto-scaling helps make sure that system resources are always available by increasing capacity during high-demand windows. It also allows the cluster to return to its previous state and size after peak demand wanes.

Anthos on VMware

Anthos on VMware is an extension of Google Kubernetes Engine that is deployed in an end user’s private data center. An organization can deploy the same applications designed to run in containers in Google Cloud in

Kubernetes clusters on premises. Anthos on VMware offers the following benefits:

- **Cost Savings.** End users can realize significant cost savings by utilizing their own physical resources for their application deployments instead of provisioning resources in their Google Cloud environment.
- **Develop Then Publish.** On-premises deployments can be used while applications are in development, which allows for testing of applications in the privacy of a local data center before being made publicly available in the cloud.
- **Security Requirements.** Customers with increased security concerns or sensitive data sets that cannot be stored in the public cloud are able to run their applications from the security of their own data centers, thereby meeting organizational requirements.

Next: Design Considerations

Design Considerations

This section describes the design considerations necessary for the successful deployment of the NetApp HCI Anthos solution.

Port Identification

NetApp HCI consists of NetApp H-Series nodes dedicated to either compute or storage. Both node configurations are available with two 1GbE ports (ports A and B) and two 10/25 GbE ports (ports C and D) on board. The compute nodes have additional 10/25GbE ports (ports E and F) available in the first mezzanine slot. Each node also has an additional out-of-band management port that supports Intelligent Platform Management Interface (IPMI) functionality. The following figure identifies each of these ports on the rear of an H410C node.

[NetApp HCI network ports (compute node).] | *netapp_hci_network_ports_compute_node.png*

Figure 3. NetApp HCI network ports (compute node).

Network Design

The NetApp HCI with Anthos solution uses two data switches to provide primary data connectivity at 25Gbps. It also uses two additional management switches that provide connectivity at 1Gbps for in-band management for the storage nodes and out-of-band management for IPMI functionality.

Cabling Storage Nodes

The management ports A and B must be active on each storage node to run NDE, configure the NetApp HCI cluster, and provide management accessibility to Element after the solution is deployed. The two 25Gbps ports (C and D) should be connected, one to each data switch, to provide physical fault tolerance. The switch ports should be configured for multi-chassis link aggregation (MLAG) and the data ports on the node should be configured for LACP with jumbo-frames support enabled. The IPMI ports on each node can be used to remotely manage the node after it is installed in a data center. With IPMI, the node can be accessed with a web-browser-based console to run the initial installation, run diagnostics, and reboot or shut down the node if necessary.

Cabling Compute Nodes

The 25Gbps ports on the compute nodes are cabled with one onboard port © cabled to one data switch, and an additional port from the PCI slot (E) cabled to the second switch to provide physical fault tolerance. These ports should be configured to support jumbo frames. Connectivity for the node is managed by the vDS after VMware vSphere is deployed in the environment. The IPMI ports can also be used to remotely manage the node after it is installed in a data center. With IPMI, the node can be accessed via a web-browser-based console to run diagnostics and to be rebooted or shut down if necessary.

[Network cabling reference diagram.] | *network_cabling_reference_diagram.png*

Figure 4. Network cabling reference diagram.

VLAN Requirements

The solution is designed to logically separate network traffic for different purposes by using Virtual Local Area Networks (VLANs). NetApp HCI requires a minimum of three network segments. However, this configuration can be scaled to meet customer demands or to provide further isolation for specific network services. The following table lists the VLANs that are required to implement the solution, as well as the specific VLAN IDs that are used later in the validated architecture deployment.

VLANs	Purpose	VLAN Used
Out-of-band management	Management for HCI nodes	16
In-band management	Management for HCI nodes and infrastructure virtual guests	3480
Storage Network	Storage network for NetApp Element	3481
vMotion network	Network for VMware vMotion	3482
VM network	Network for virtual guests	1172

Network Infrastructure Support Resources

The following infrastructure should be in place prior to the deployment of the Anthos on NetApp HCI solution:

- A DHCP server providing addresses for both the in-band management network and the VM network. The DHCP pool must be large enough to support at least 10 VMs for an initial deployment and should be scaled as necessary.
- At least one DNS server providing full host-name resolution that is accessible from the in-band management network and the VM network.
- At least one NTP server that is accessible from the in-band management network and the VM network.
- Outbound internet connectivity for both the in-band management network and the VM network.

Best Practices

Install a Second F5 Big-IP Virtual Edition Appliance

In a production environment, it is a best practice to avoid single points of failure in your environment. For this validation, a single F5 BIG-IP Virtual Edition Load Balancer appliance was used to validate connectivity to the control plane and the ingress VIP addresses for the Anthos on VMware clusters. While this works fine for a simple validation, loss of communication with the control plane VIP for a cluster can make a cluster inaccessible or unable to be managed from the admin workstation or the Google Cloud console. F5 BIG-IP Virtual Edition supports application-based HA to make sure disruptions do not happen. Although this issue is mentioned briefly, setup procedures for this functionality are not described in detail in this document. However, NetApp recommends investigating this feature further before deploying the NetApp HCI for Anthos solution into production.

Enable VMware vSphere DRS and Configure Anti-Affinity Rules

VMware vSphere provides a feature that makes sure that no single node in the cluster runs low on physical resources available to virtual guests. The Distributed Resource Scheduler (DRS) can be configured on vSphere clusters consisting of at least three ESXi nodes. The NetApp HCI minimum configuration described in this deployment guide consists of two compute nodes and is unable to make use of this feature. As a result of this limitation, we were also forced to disable anti-affinity rules for the Anthos on VMware clusters that we deployed.

Anti-affinity rules ensure all masters or all workers for a specific user cluster run on different nodes, so that a single node failure cannot disable an entire user cluster or the pods that it is hosting. As the NetApp HCI system is both easily and rapidly scalable, and considering the minimum deployment described in this validation has two open chassis slots for immediate expansion of HCI 410C nodes, NetApp suggests adding additional compute nodes into the empty chassis slots prior to deploying the solution into production, and enabling DRS with Anti-Affinity rules.

Leverage SnapMirror to Copy Data Remotely for Disaster Recovery

NetApp Element storage systems can use NetApp SnapMirror technology to replicate storage volumes to systems running the NetApp ONTAP system, including AFF, FAS, and Cloud Volumes ONTAP. You can set up regularly scheduled SnapMirror operations to back up the VMware datastores and restore from a remote site in the event of a disaster. It is also possible to use SnapMirror to back up or migrate the persistent volumes provisioned by Trident and reattach them to Kubernetes clusters deployed in other environments and in the cloud.

Next: Hardware and Software Requirements

Hardware and Software Requirements

This section describes the hardware and software requirements for the NetApp HCI and Anthos solution.

Hardware Requirements

The following table lists the minimum number of hardware components that are required to implement the solution. The hardware components that are used in specific implementations of the solution might vary based on customer requirements.

Hardware	Model	Quantity
NetApp HCI compute nodes	NetApp H410C	2
NetApp HCI storage nodes	NetApp H410S	4
Data switches	Cisco Nexus 3048	2
Management switches	Mellanox NS2010	2

Software Requirements

The following table lists the software components that are required to implement the solution. The software components that are used in any implementation of the solution might vary based on customer requirements.

Software	Purpose	Version
NetApp HCI	Infrastructure (compute/storage)	1.6P1
VMware vSphere	Virtualization	6.5U2
Anthos on VMware	Container orchestration	1.1
F5 Big-IP Virtual Edition	Load balancing	15.0.1
HashiCorp Terraform	Automation and provisioning	0.12.12
NetApp Trident	Storage management	19.10

Next: Workflow Summary

Deployment Steps

This section provides detailed protocols for implementing the NetApp HCI solution for Anthos.

This deployment is divided into the following high-level tasks:

1. Configure management switches
2. Configure Data Switches
3. Deploy NetApp HCI with the NetApp Deployment Engine
4. Configure the vCenter Server
5. Deploy and Configure the F5 Big-IP Virtual Edition Appliance
6. Complete Anthos Prerequisites
7. Deploy the Anthos Admin Workstation
8. Deploy the Admin and the First User Cluster
9. Deploy additional user clusters
10. Enable access to the cluster with the GKE console
11. Install and configure NetApp Trident storage provisioner

Video Demos: NetApp HCI with Anthos

The following videos demonstrate some of the capabilities documented in this NVA.

- Deploying an application from the Google Cloud Application Marketplace to Anthos:

▶ https://docs.netapp.com/us-en/netapp-solutions/containers/media/Anthos-Deploy-App-Demo.mp4 *(video)*

- Dynamic scaling of Kubernetes clusters deployed on Anthos on VMware:

▶ https://docs.netapp.com/us-en/netapp-solutions/containers/media/Anthos-Scaling-Demo.mp4 *(video)*

- Using NetApp Trident to provision and attach a persistent volume to a Kubernetes pod on Anthos:

▶ https://docs.netapp.com/us-en/netapp-solutions/containers/media/Anthos-Trident-Demo.mp4 *(video)*

Where to Find Additional Information: NetApp HCI with Anthos

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

- Anthos Documentation
- NetApp HCI Documentation
- NetApp NDE 1.6 Deployment Guide
- NetApp Trident Documentation
- VMware vSphere 6.5 Documentation
- F5 Big-IP Documentation