Multi-cloud Container Platform with Hitachi Unified Compute Platform RS and VMware Tanzu

Reference Architecture Guide

Feedback

Hitachi Vantara welcomes your feedback. Please share your thoughts by sending an email message to SolutionLab@HitachiVantara.com. To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.

Revision history

Changes	Date
Fixed an error on this page.	May 6, 2022
▪ Changed block tracking (CBT) is not supported by Velero Plugin for vSphere. ▪ Do not run object stores within the same TKG cluster. ▪ Included a link to install Velero client and vSphere Operator CLI v1.1.1.	October 1, 2021
Minor updates	August 30, 2021

Reference Architecture Guide

This paper demonstrates combining VMware Cloud Foundation (VCF), VMware Tanzu Kubernetes Grid (TKG), VMware Tanzu Mission Control (TMC), and Hitachi Unified Compute Platform RS (UCP RS) to create, protect, and manage on-premises Kubernetes clusters and workloads that require persistent storage. Use these VMware TKG deployment options for Hitachi UCP RS in a hybrid converged/hyperconverged configuration to achieve the most flexible capabilities for persistent storage in a VMware vSphere-based container environment. You can now provide multiple storage and data protection options to consumers of your container-based virtual infrastructure, using well-known and supported storage integrations between Hitachi Vantara and VMware.

Your business is under pressure to deal with increased customer demands and increasing competition. Manage your business pressure using digital transformation, where agility, processes, and architecture are challenges that you need to address. Containerizing applications is one of the key initiatives in digital transformation.

Containers and their platforms enable increased speed from development to deployment by increasing operational efficiency. From development, to building, to testing, and then to deployment, containers streamline these operations. Containers also provide less overhead compared to traditional units of hardware or virtual hardware. Containers are highly portable with less dependency on the operating system and the underlying hardware platform.

A key element in the successful deployment of a container platform is having a robust and flexible infrastructure that can meet a wide variety of requirements in a highly dynamic environment. Hitachi infrastructure with VMware Tanzu provides highly available and high-performance infrastructure for container applications. Some specific challenges of providing an infrastructure for a container platform are:

- Data persistence

  Data is at the core of any application. Many applications require data persistence, such as MariaDB, PostgreSQL, MongoDB, and MySQL, among others. Continuous integration and continuous delivery (CI/CD) pipelines require data persistency at every level.

  Using well-known and proven vSphere storage integrations, you can provide persistent storage for stateful container applications. Using vSphere storage policies in combination with VASA Provider, you can provide dynamic ReadWriteOnce (RWO) vVols-based persistent volumes to container applications running within VMware TKG guest clusters and VMware Tanzu Supervisor clusters. Hitachi UCP RS with VMware Tanzu provides the storage infrastructure and integrations needed for your organization to successfully adopt container technology, while taking advantage of an existing VMware vSphere infrastructure within your organization.

- Data protection

  Data protection is a critical aspect of any data center infrastructure. VMware Cloud Foundation 4.2 includes the built-in Velero operator to enable your organization to protect any container-related entity, including Kubernetes Persistent Volumes.

  VMware Tanzu Mission Control also provides the capability to deploy and configure Velero to any CNCF-compliant Kubernetes clusters attached to TMC and back them up to an S3-compliant storage such as MinIO or AWS S3 itself.

  Hitachi Content Platform for cloud scale (HCP for cloud scale) provides standard AWS-compliant S3 storage that can be used with either on-premises or TMC-based Velero implementations.

- Computing platform

  With a wide range of applications that are stateful or stateless, a wide range of flexible computing platforms are necessary to match both memory and CPU requirements.

  The type of computing technology is also a consideration for licensing costs. Hitachi Vantara provides different computing options from the 1U dual socket Hitachi Advanced Server DS120 to the 2U quad socket Hitachi Advanced Server DS240.

▪ Network connectivity

As with any infrastructure, a reliable network is needed to provide enough bandwidth and security for container architectures. Hitachi UCP RS uses a spine and leaf design using Cisco Nexus or Arista switches.

VMware Cloud Foundation includes VMware NSX-T networking, enabling a flexible and secure software-defined networking (SDN) solution that easily integrates with VMware solutions.

▪ Infrastructure management

Having a robust and flexible infrastructure without efficient lifecycle management decreases efficiency exponentially as the infrastructure scales.

Orchestration and automation are the key to operational efficiencies. Hitachi UCP Advisor (UCP) Advisor provides a single pane of glass management and lifecycle manager for converged infrastructure, with automation for compute, network, and storage infrastructure. Hitachi Ops Center is also available with Hitachi Virtual Storage Platform (VSP) for storage management.

VMware Cloud Foundation offers tested and approved infrastructure designs that adhere to the VMware Validated Designs family of solutions in addition to automated deployment and configuration workflows. These features combine to ensure consistency and efficiency of infrastructure deployment in your organization.

VMware Tanzu Mission Control (TMC) offers single pane of glass management of disparate Kubernetes clusters across your organization. You can configure data protection, monitoring, and management of any VMware Tanzu resources or CNCF-compliant Kubernetes clusters to gain a holistic view of your Kubernetes-based infrastructure throughout your organization.

Solution overview

Hitachi UCP RS with VCF and VMware TKG with TMC provide a flexible on-premises container-based platform for your organization. Using this solution, you can easily deploy VMware Tanzu Kubernetes Grid guest clusters to VMware vSphere environments. You can also use well-known vSphere storage integrations into your container-based applications to provide dynamic persistent volume storage options within Kubernetes.

You can combine integrated Velero data protection and HCP for cloud scale with the listed components to complete your persistent data deployment and protection solution for container-based infrastructures.

Kubernetes is a powerful container orchestration platform. It provides a physical platform for different container management platforms such as VMware Cloud Foundation with Tanzu. Kubernetes Grid Services is one of the solutions for UCP RS.

The following figure shows a high-level diagram of VMware Tanzu Kubernetes Grid Services, container deployment types, and persistent volume types on the UCP RS stack running VMware Cloud Foundation.

Hitachi Unified Compute Platform Rack Scale (UCP RS) Stack
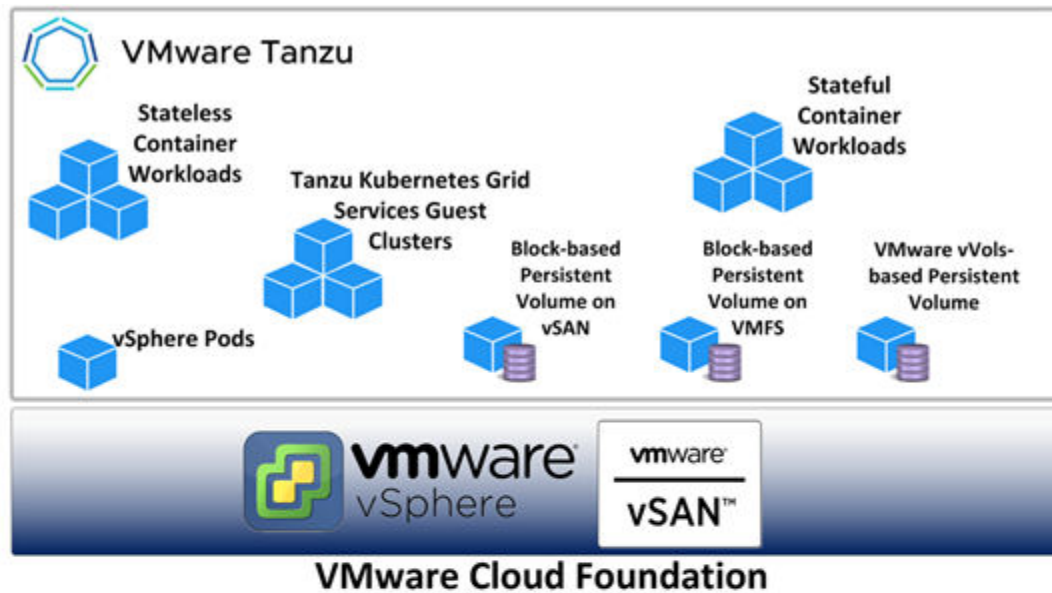
This reference architecture focuses on the deployment of VMware Tanzu Kubernetes Grid Services and associated services onto a VCF-based UCP RS infrastructure. This takes advantage of existing Hitachi and VMware storage integrations that can be extended into Kubernetes ecosystems such as Tanzu Kubernetes Grid Services. Hybrid Hitachi all-NVMe compute nodes with Fibre Channel HBAs were used in this reference architecture to provide the most flexible underlying storage capabilities to the Tanzu Kubernetes Grid Services ecosystem.

The following figure illustrates the VCF architecture and components used on top of UCP RS to enable Tanzu Kubernetes Grid Services (credit to VMware).
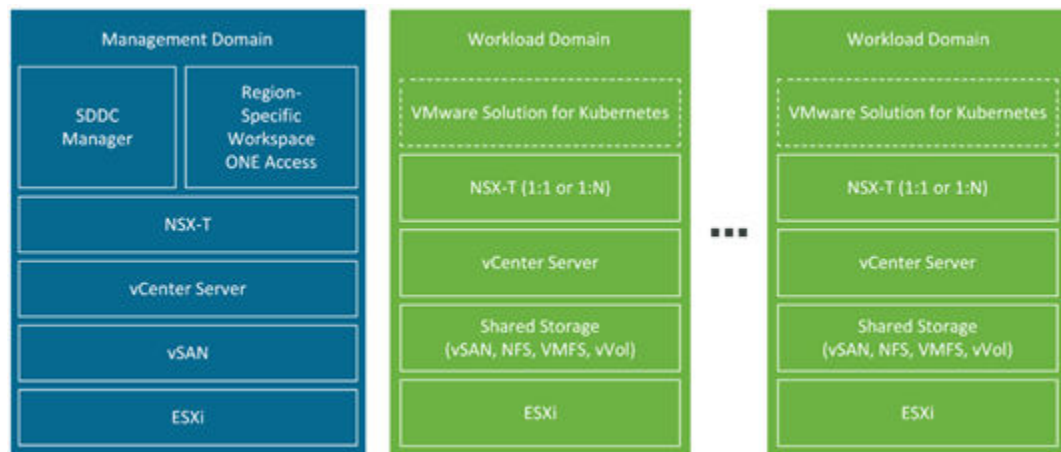
The solution validation of this reference architecture consists of a stateful business application service such as Wordpress, running within a TKG guest cluster.

This reference architecture guide describes how to deploy a TKG guest cluster through YAML definition and kubectl, as well as how to deploy a TKG guest cluster through TMC.

Both TKG guest clusters are deployed onto an on-premises UCP RS running VCF 4.2, illustrating local administration and SaaS-based management using kubectl and TMC. All Velero data protection options validated in this paper use Hitachi Content Platform for cloud scale as an S3-compliant target for backups and restores.

Follow the steps in the *Solution design* and *Solution validation* sections to learn about the storage, data protection, and management operations available with VMware and Hitachi when using VCF and Tanzu Kubernetes Grid Services.

Solution components

These are the key hardware and software components used for this environment.

Hardware components

The following tables list the versions of hardware and software tested in the Solution Validation section of this reference architecture. These components comply to the current UCP CI 4.5 firmware versions for UCP RS running VCF 4.2.

The solution tested used specific features based on the following hardware. You can use either Hitachi Advanced Server DS120/DS220/DS225/DS240 or any qualified server platform for UCP RS.

For more information, see the UCP CI Interoperability Matrix and UCP Product Compatibility Guide. For VCF specific compatibility see the UCP RS Interoperability Matrix.

Table 1 Hardware Components

Hardware	Description	Version	Quantity
Hitachi Advanced Server DS120 (compute)	▪ 2 × Intel Xeon 6240 18-core 2.60GHz processor ▪ 16 × 16 GB DIMM, 256 GB memory ▪ 32 GB SATADOM (boot) ▪ Emulex LPe3200 32 Gb/sec dual port PCIe HBA ▪ 2 × Mellanox CX4 dual port 10/25G NIC ▪ vSAN Cache Tier: 2 × Intel Optane SSD DC P4800X (375 GB, U.2) NVMe ▪ vSAN Capacity Tier: 10 × Intel SSD DC P4510 (4 TB, U.2) NVMe	BMC: 4.68.06 BIOS: 3B19.H00	8
Hitachi Virtual Storage Platform 5500	▪ 2 TB cache ▪ 8 × 1.9 TB NVMe drives ▪ 4 × 32 Gbps Fibre Channel ports	90-05-02-00/01	1
Hitachi Virtual Storage Platform G600	▪ 86 GB cache ▪ 8 × 1.2 TB SAS 10K drives ▪ 4 × 8 Gbps Fibre Channel ports	83-05-33-40/00	1
Cisco Nexus 9332C switch (spine)	▪ 32-port 40/100 GbE ▪ 2-port 1/10 GbE	NXOS 9.3.5	2
Cisco Nexus 93180YC-FX switch (leaf)	▪ 48-port 10/25 GbE ▪ 6-port 40/100 GbE	NXOS 9.3.5	2
Cisco Nexus 92348	▪ 48-port 1 GbE ▪ 4-port 1/10/25 GbE ▪ 2-port 40/100 GbE	NXOS 9.3.5	1
Brocade G620	▪ 48-port 16/32 Gbps Fibre Channel switch	9.0.0b	2

Software components

The following table lists the key software components.

Table 2 Software Components

| Software	Version
Hitachi Storage Virtualization Operating System RF	90-05-02-00/01 83-05-33-40/00
Hitachi UCP Advisor	3.10
Storage Provider for VMware vCenter (VASA)	3.6
VMware Cloud Foundation	4.2
VMware vSphere	7.0 Update 1d

Solution design

This is the detailed solution example for the Multi-cloud Container Platform with Hitachi Unified Compute Platform RS and VMware Tanzu.

UCP RS infrastructure components

The following figure shows a high availability configuration of Hitachi Unified Compute Platform RS used to validate the VMware Tanzu Kubernetes Grid Services solution. It includes the following components:

- Two Cisco 9332C or Arista 7050CX3 spine Ethernet switches.

- Two Cisco 93180YC-FX or Arista 7050SX3 leaf Ethernet switches.

- One Cisco 92348 or Arista 7010T management switch.

- Four Hitachi DS120 servers for VCF management nodes. vSAN ReadyNode certified as UCP HC V120-series, UCP HC V120F, UCP HC V121F, UCP HC V123F, or UCP HC V124N.

- Four or more Hitachi DS120, DS220, DS225, or DS240 for VCF virtual infrastructure workload domain nodes.

  - For vSAN compute nodes, leverage supported internal drives. These compute nodes are vSAN Ready Node Certified as UCP HC V120-series/V120F/V121F/V123F/V124N, UCP HC V220-series/V220F, UCP HC V225G, or UCP HC DS240, respectively.

  - For vVols or VMFS compute nodes, leverage the HBA PCIe card, which is optionally configured together with the UCP HC vSAN ReadyNodes, or when configuring UCP Fibre Channel-only nodes in UCP RS.

- One or more Hitachi VSP storage systems.
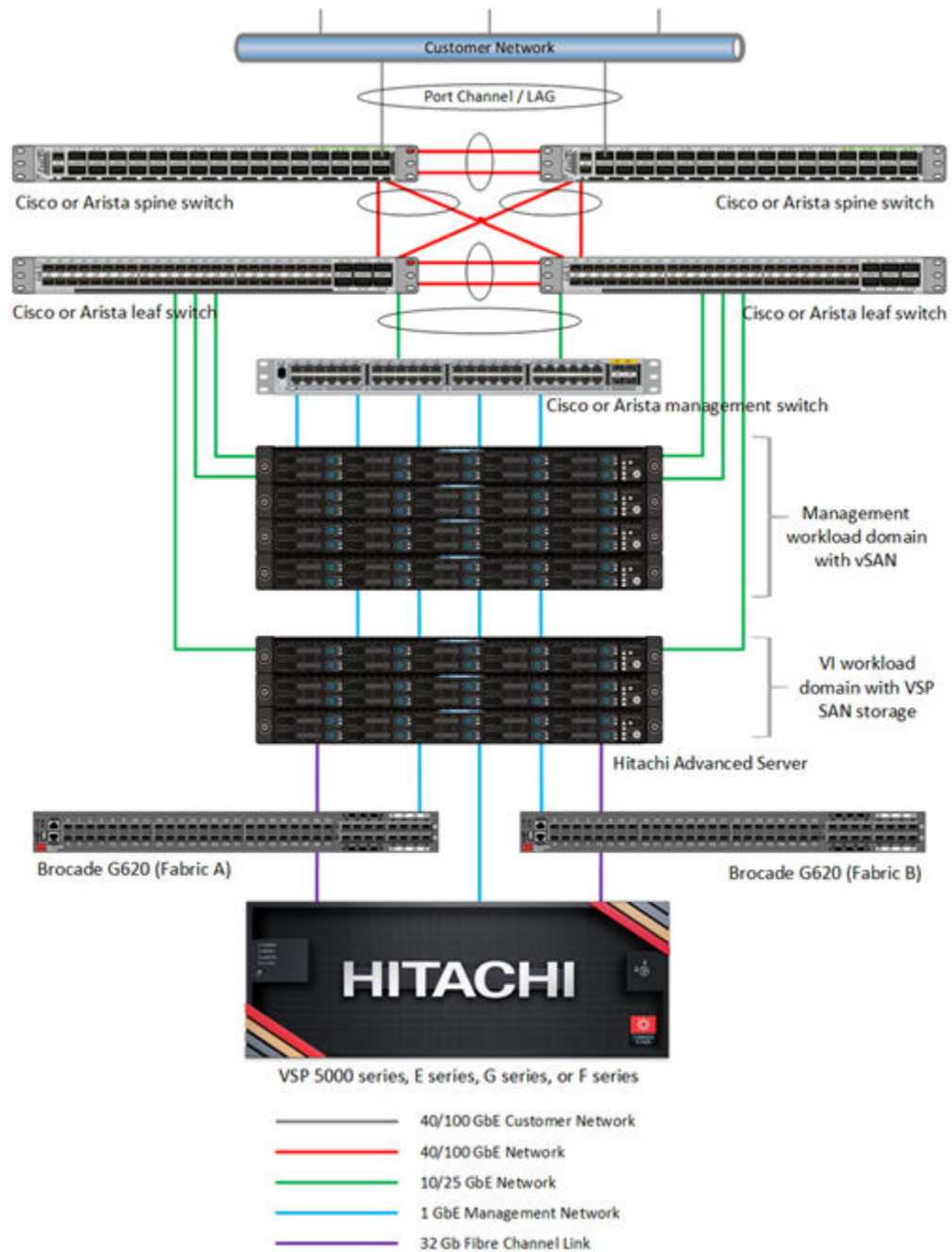
Deploy your UCP RS management workload domain by following the deployment instructions available at Unified Compute Platform RS with VMware Cloud Foundation Quick Start Guide - Hitachi Vantara Knowledge.

The configuration with Hitachi Virtual Storage Platform is described in Unified Compute Platform CI. For more information regarding Unified Compute Platform CI configurations, see Hitachi Unified Compute Platform CI for VMware vSphere Reference Architecture Guide.

VMware vVols and storage policy-based management (SPBM)

Storage Provider for VMware vCenter (VASA Provider) enables organizations to deploy Hitachi Storage infrastructure with VMware vSphere Virtual Volumes (vVols) to bring customers on a reliable enterprise journey to a software-defined, policy-controlled data center.

Hitachi storage policy-based management allows automated provisioning of virtual machines (VMs) and quicker adjustment to business changes. Virtual infrastructure (VI) administrators can make changes to policies to reflect changes in their business environment, dynamically matching storage-policy requirements for VMs to available storage pools and services. The vVols solution reduces the operational burden between VI administrators and storage administrators with an efficient collaboration framework leading to faster and better VM and application services provisioning.

To use VMware vVols with Hitachi storage, install Hitachi Storage Provider for VMware vCenter. For details, see VMware vSphere Virtual Volumes (vVols) with Hitachi Virtual Storage Platform Quick Start and Reference Guide.

Deploy the Hitachi Storage Provider for VMware vCenter into the management workload domain within your UCP RS VCF 4.2-based infrastructure. To deploy this environment, see Storage Provider for VMware vCenter (VASA).

Hitachi UCP Advisor

When you use VMFS datastores as part of a workload domain, all VSP storage operations are performed outside of VMware SDDC Manager. These LUNs are not managed by VMware Cloud Foundation.

An advantage of a UCP RS with VSP architecture, when compared with other VMware Cloud Foundation plus external storage validated solutions, is in the ease of SAN storage provisioning and SAN storage management using Hitachi UCP Advisor, even when vVols are not used.

Deploy Hitachi UCP Advisor into the management workload domain within your UCP RS VCF 4.2-based infrastructure. For more information, see Hitachi Unified Compute Platform Advisor.

Understanding VMware Tanzu Kubernetes Grid Services concepts and entities

VMware Tanzu allows you to run containers directly on VMware ESXi hosts as well as deploy dedicated Kubernetes clusters (Tanzu Kubernetes Grid Services guest clusters), while consuming standard vSphere resources controlled by the vSphere administrator. This is achieved by creating entities called vSphere Namespaces in the vSphere infrastructure. These vSphere Namespaces allow the vSphere administrator to create resource groupings to limit the underlying infrastructure resources that Kubernetes resources can consume.

You can create vSphere Namespaces by enabling workload management on a standard VMware vSphere ESXi cluster. To enable Tanzu functionality, the Workload Control Plane (WCP) service on the vCenter that hosts the cluster, where the workload management is enabled, interfaces with VMware components such as, SDDC Manager, NSX-T, and VMware vCenter.

When workload management is enabled on an ESXi cluster, vSphere pods, that is containers running on the ESXi host itself, TKG guest clusters, and virtual machines can be deployed as part of the container-based infrastructure to run workloads.

The following figure illustrates the various components that comprise a Tanzu Kubernetes Grid Services deployment (credit to VMware).
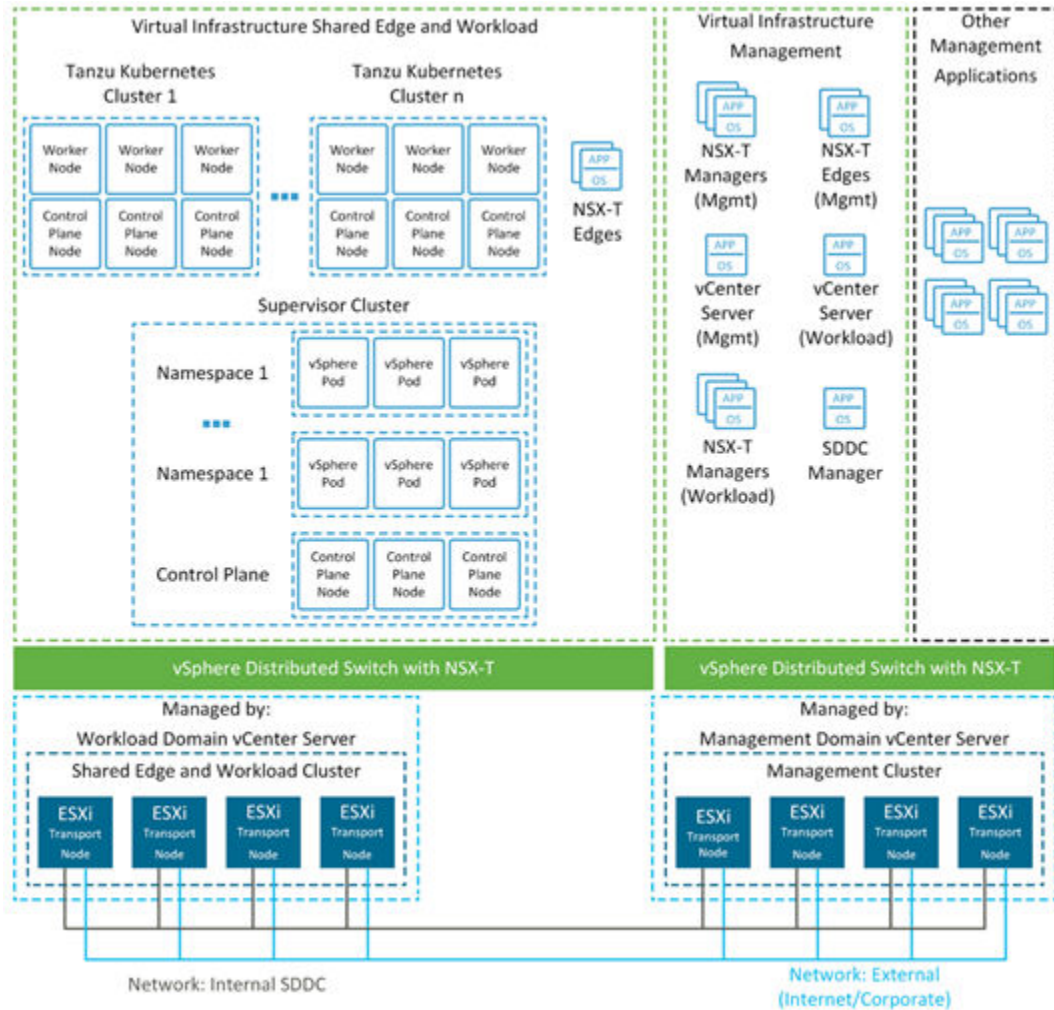
For more information on VMware Tanzu Kubernetes Grid Services, see the VMware product documentation link.

CSI and pvCSI storage concepts

Vanilla Kubernetes deployments on VMware vSphere typically require manual installation of the vSphere CSI driver to take advantage of the vSphere SPBM to Kubernetes StorageClass mapping. With Tanzu Kubernetes Grid Services, the vSphere CSI driver is pre-installed inside the Tanzu Supervisor cluster. Specific vSphere SPBM policies are then authorized for use at the vSphere Namespace level, which are translated into Kubernetes StorageClasses automatically.

For Tanzu Kubernetes Grid Services guest clusters, the paravirtual CSI (pvCSI) driver comes automatically installed inside the cluster. Any vSphere SPBM policies that were authorized by the vSphere administrator within the vSphere Namespace are automatically passed through as StorageClasses within the guest cluster, where the Tanzu Kubernetes Grid Services guest cluster is deployed.

Provisioning behavior is different depending on whether you are provisioning a persistent volume (PV) to a vSphere pod running natively on the Supervisor cluster, or if you are provisioning a PV to a pod running within a Tanzu Kubernetes Grid Services guest cluster.

Because the native vSphere CSI driver is installed in the Supervisor cluster, provisioning operations are similar to provisioning a vanilla Kubernetes cluster. A Persistent Volume Claim (PVC) is created that references an available StorageClass, which maps to a vSphere SPBM policy. A first class disk (FCD) is created within vSphere, and a resultant PV is presented to the Kubernetes layer from the CSI driver. The FCD is then mounted to the vSphere pod when requested for use as a PV.

The following figure illustrates the differences between presenting a volume to a vSphere pod and a Tanzu Kubernetes cluster (credit to VMware).
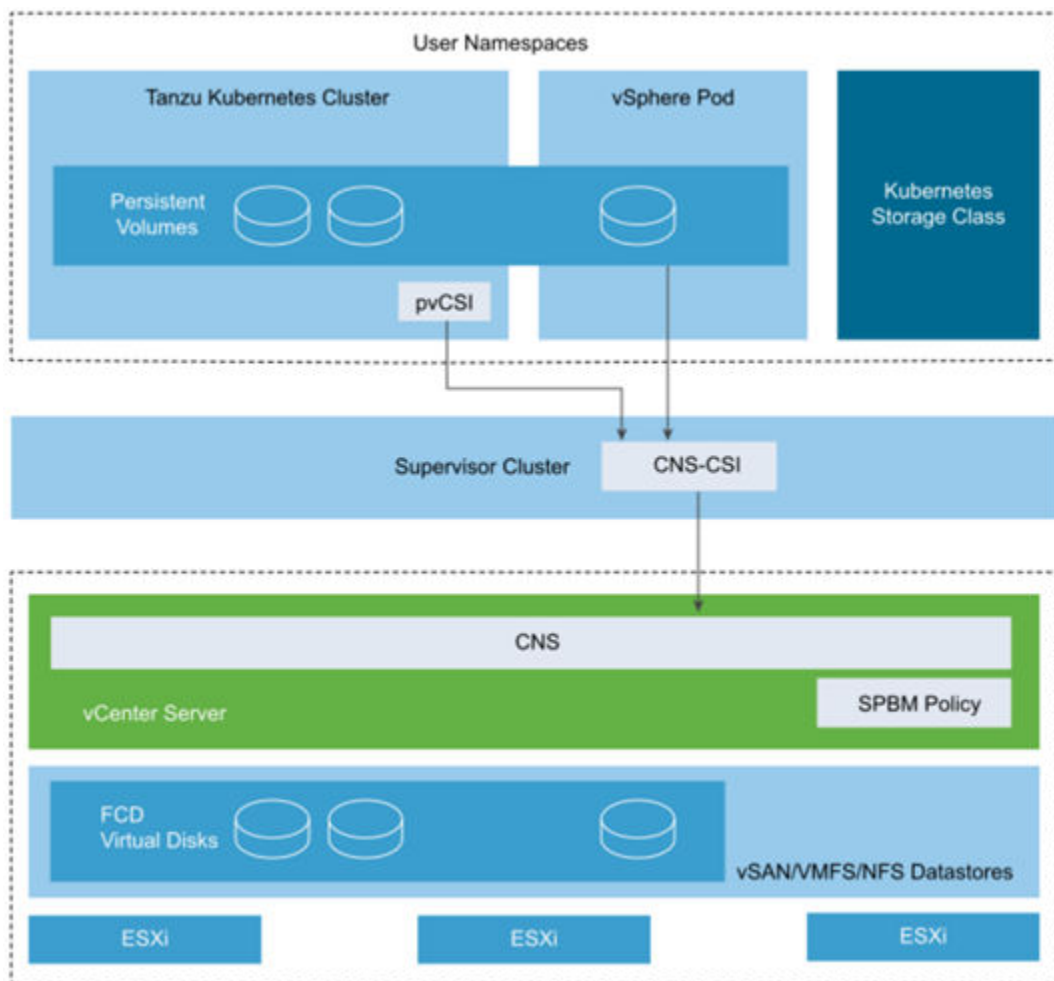
In a Tanzu Kubernetes Grid Services guest cluster, the pvCSI driver issues requests from the native CSI driver in the Supervisor cluster. The resultant PV is attached to a worker machine in the Tanzu Kubernetes Grid Services guest cluster when requested by a pod for use as a PV.

The following figure illustrates the PV provisioning workflow in Tanzu Kubernetes Grid Services (credit to VMware).
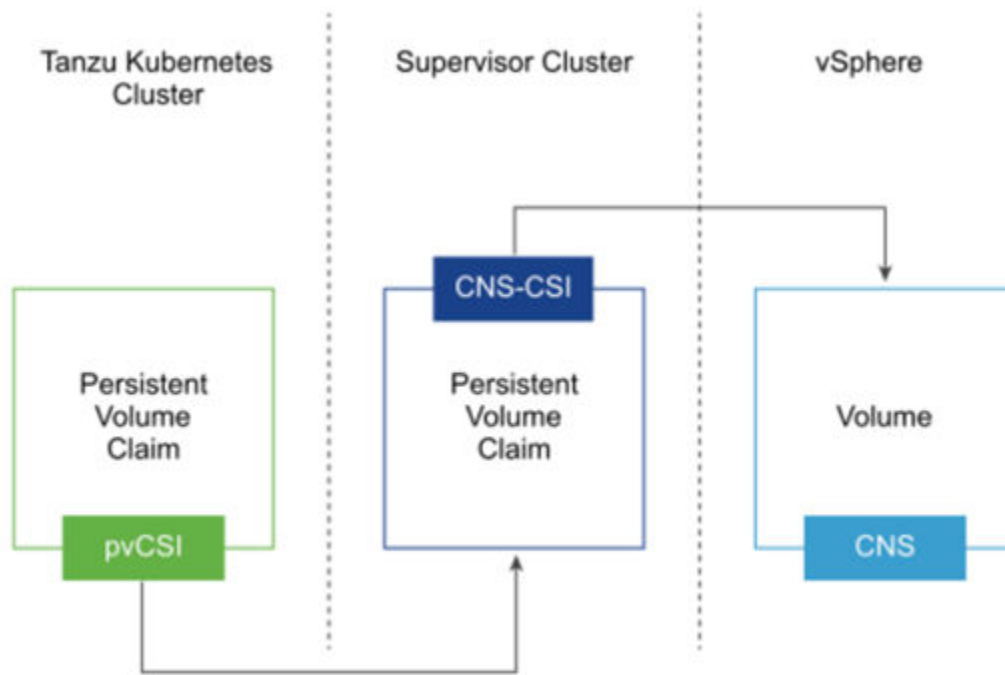
For more information on vSphere CSI and pvCSI in VMware Tanzu, see the VMware product documentation link.

Understanding native Tanzu data protection options

With VCF 4.2, VMware introduced built-in operators for specific services within the supervisor and guest clusters. One of these is the Velero data protection operator, which allows backup and restore of Kubernetes elements including PVs to S3-compliant object storage. This manual installation option uses the Velero Plugin for vSphere, which enables VADP-based backups with native vSphere snapshots against FCDs in vSphere. However, it requires deployment of the Velero Data Mover OVA into your vSphere environment.

Tanzu Mission Control offers the capability to automatically deploy Velero as a data protection option, with the capability to configure centralized AWS S3 and S3-compatible storage targets for multiple attached clusters or cluster groups. This option uses the restic plugin for Velero and does not use VADP-based vSphere snapshots for copying the data. Instead, it uses restic snapshots and built-in data movers to copy PV data to the S3 target.

TMC also manages the Velero lifecycle in the cluster. Customers do not need to install or manually configure Velero in each cluster because TMC will manage the fleet of clusters in a centralized way. If you choose to run Velero backup commands directly instead of using TMC-installed Velero, proceed with care. For example, if you use the same S3 bucket configured in TMC for backups using Velero CLI directly, and if you disable data protection from TMC with the option to remove backup files, it will remove Velero from the cluster and it will delete all the data from the S3 bucket.

Use the TMC-enabled Velero with restic installation if you:

- Want to easily assign a target for S3-compliant backup storage for multiple clusters.

- Have backup set sizes that are less than 100 GB.

- Are aware that large file deduplication (such as database files) can take a long time to scan, potentially adding a considerable amount of time to the backup.

- Do not want to deploy additional infrastructure to your vSphere environment to enable PV backups with Velero.

Use the manual installation method with the Velero Plugin for vSphere and Velero Data Manager OVA if you:

- Want to use VADP-based backups of PVs within vSphere TKG guest clusters.

- Have backup set sizes that are larger than 100 GB.

- Have the capability to deploy additional infrastructure to complement Velero for data mover functionality (Velero Data Manager OVA).

Deploy the target VI workload domain for VMware Tanzu Kubernetes Grid Services

Follow these procedures to deploy the target virtual infrastructure (VI) workload domain within your VCF 4.2 environment. These procedures assume you have already completed the following:

- Deployed your VCF 4.2 management workload domain.

- Successfully configured your BGP network environment.

- Deployed and configured Storage Provider for VMware vCenter (VASA Provider).

- Deployed and configured Hitachi UCP Advisor (UCP Advisor).

To successfully deploy this solution, you must complete the following:

- A minimum of four physical hosts for the VI workload domain configured with:

  - Local storage for vSAN storage.

  - Fibre Channel HBA zoned to a Hitachi Virtual Storage Platform storage system supported by Hitachi Storage Provider for VMware vCenter.

- IP address ranges and segments to support Tanzu Kubernetes Grid Services ingress and egress networking from the underlying Kubernetes infrastructure.

📄 **Note:** Several of the screenshots in this reference architecture show three hosts in the VI workload domain. During validation of this solution, three hosts were deployed into our initial VI workload domain, and then a fourth host was added later to verify host addition/cluster expansion workflows. Deploy a minimum of four hosts if you are deploying this solution.

Prepare hosts

You must prepare the hosts used in the VI workload domain used for Tanzu Kubernetes Grid Services in the same manner used for hosts when you deployed the management workload domain:

- Ensure vmnic0/1 are available for network assignment.
- Clear all partitions on local disks to be used for the vSAN datastore in the VI workload domain.
- Tag disks to be used as the cache tier in vSAN.

For more information, see the VMware documentation link.

Configure network pools

You must configure network pools in SDDC Manager for the VI workload domain that you want to create. For this solution, a minimum of vMotion and vSAN pools must be created for the VI workload domain as illustrated in the following figure.
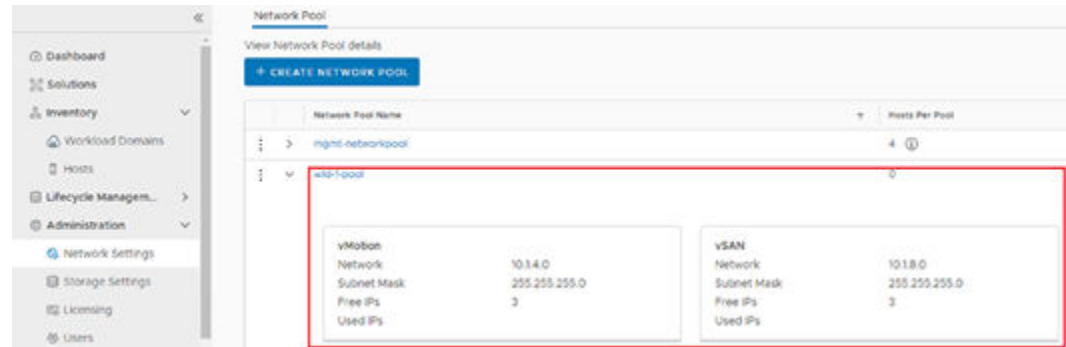
Commission hosts

Before creating the VI workload domain, you must commission the hosts so that they are available for deployment within SDDC Manager:

Procedure

1. Run the Commission Hosts wizard in SDDC Manager.
2. Select vSAN for the primary storage type.
3. Configure the hosts to use the network pool previously created for the VI workload domain.

   The following figure illustrates an example of the correct Commission Hosts wizard settings.

4. Verify the fingerprint on each host: select all the hosts, and then click **Validate All**.

5. Complete the Commission Hosts wizard and let the task complete in SDDC Manager.

Deploy a base VI workload domain

After the hosts are added and are in the Available status, you can create a new VI workload domain in the SDDC Manager.

Procedure

1. Select vSAN as the primary storage type for the VI workload domain.

2. Provide a name and organization for your VI workload domain cluster.
3. Ensure you select **Enable vSphere Lifecycle Manager Baselines**.

> ❗ **Important:** vSphere Lifecycle Manager Images workload domain types cannot have workload management enabled on them for Tanzu functionality

4. Provide a name for the vSphere ESXi cluster that will be created in the VI workload domain.

5. Enter the fully qualified domain name and the root password for the vCenter to be deployed in the VI workload domain.

The IP address field cannot be modified, DNS records must exist, and the name must be resolvable from SDDC Manager to continue in the VI workload domain deployment wizard.

6. Create a new NSX-T instance for the VI workload domain.

7. Enter your NSX-T virtual and node names, along with passwords and VLAN information specific to your environment.

8. Configure the vSAN FTT and deduplication parameters for the vSAN to be deployed.

9. Select the hosts to add to the VI workload domain.

   Only hosts in the Available status are shown; if all of your hosts do not appear see
   Prepare hosts (on page 16) and Commission hosts (on page 16).
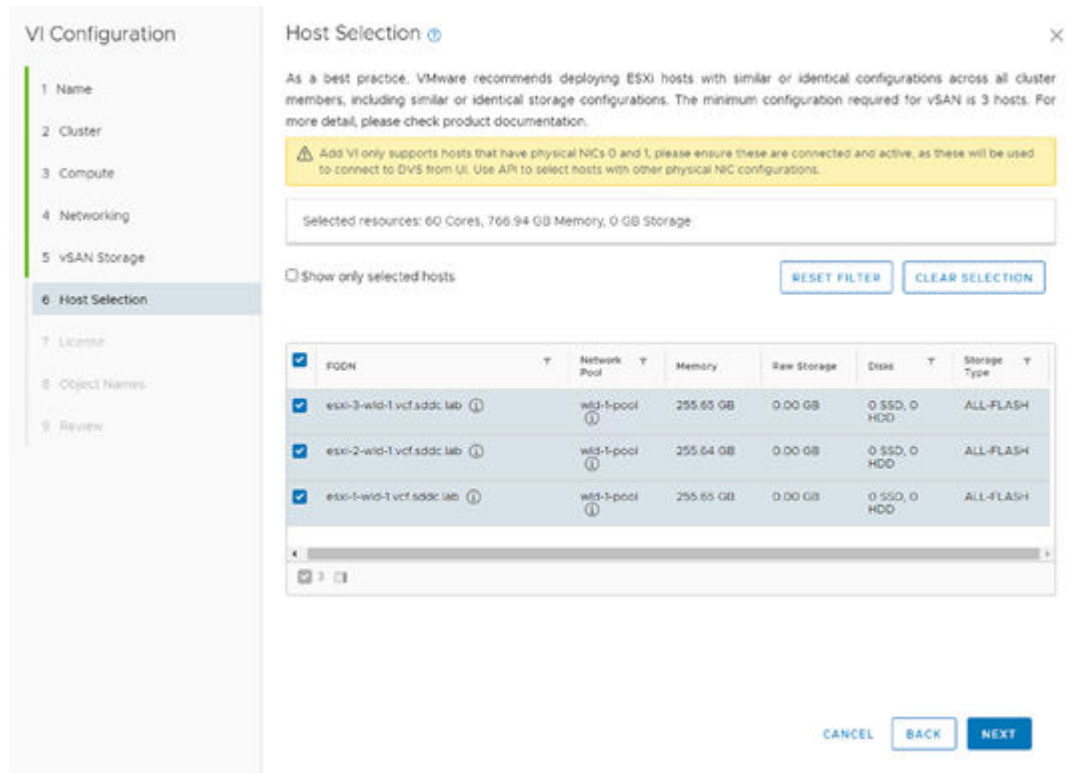
10. Enter the license information for NSX-T Data Center, VMware vSAN, and vSphere for Kubernetes (license information has been cleared in the figure).
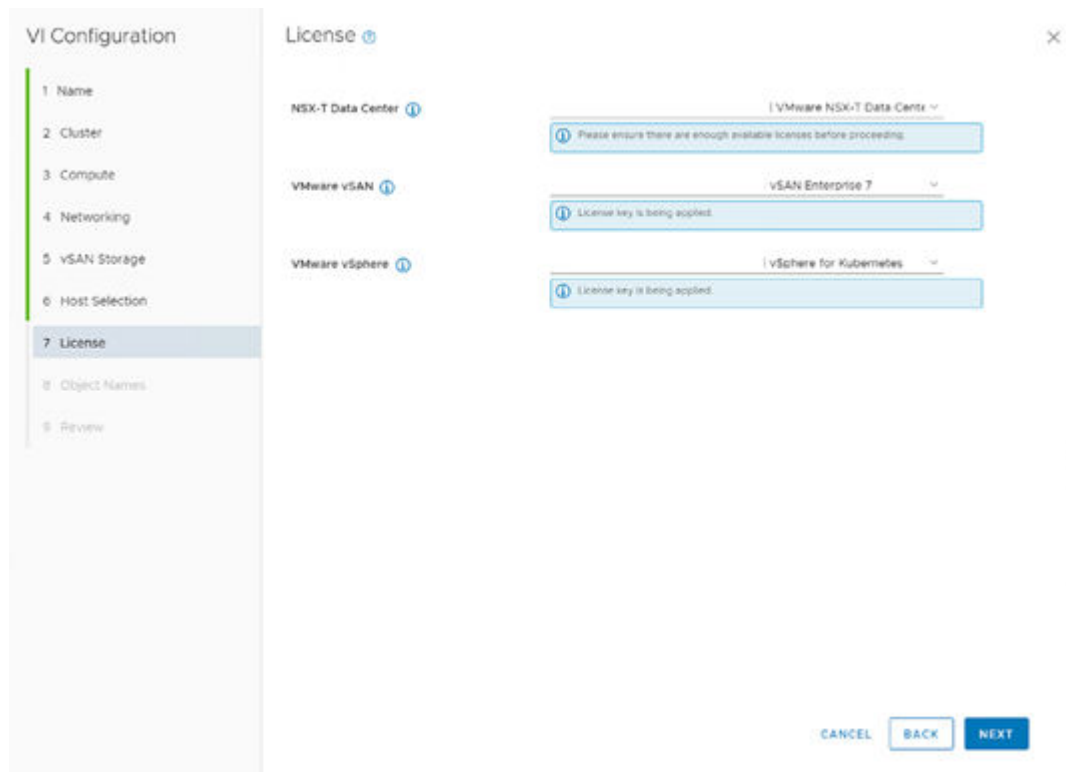
11. Review the object names that will be created for the VI workload domain and associated vSphere entities.

You can change either the name of your VI workload domain or the vSphere ESXi cluster to be deployed.

An example of auto-generated names based on the VI workload domain and vSphere ESXi cluster is shown in the following figure.

12. Review the complete configuration of the new VI workload domain and associated details.
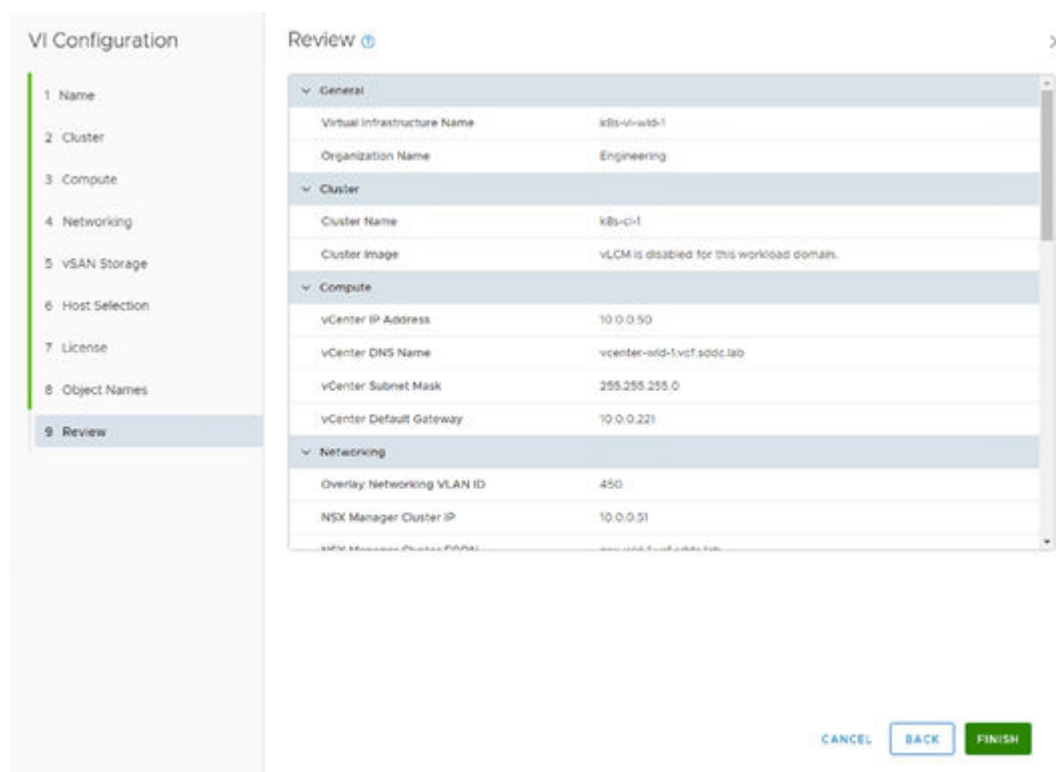
13. Click **Finish** and wait for the VI workload domain creation task to complete in SDDC Manager.

Deploy and enable Hitachi Storage (VASA) Provider for VMware vCenter

Deploy the Hitachi Storage (VASA) Provider for VMware vCenter to the management workload domain.

Procedure

1. Use UCP Advisor to create the necessary zone sets in the Fibre Channel fabrics.
2. Use Hitachi Storage Navigator to provision the necessary ALU targets from each storage system.
3. Register the Hitachi VSP storage systems in the VASA Provider.
4. Register VASA Provider as a Storage Provider within the vCenter associated with the target VI workload domain.
5. Create vVol datastores or VMFS (LDEV) datastores and associated SPBM policies for the Hitachi VSP storage systems configured for use by the target VI workload domain.

   For details, see VMware vSphere Virtual Volumes (vVols) with Hitachi Virtual Storage Platform Quick Start and Reference Guide for more information on deploying Hitachi Storage Adapter for VMware vCenter.

   For details see Using Hitachi Virtual Storage Platform with VMware Cloud Foundation and VMware Virtual Volumes for more information on using UCP Advisor to configure Hitachi VSP storage systems for use by Hitachi Storage Adapter for VMware vCenter.

Deploy additional NSX-T Manager Appliances

When SDDC Manager deployed the target VI workload domain with a new NSX-T instance, a single NSX-T Manager was deployed. Two additional NSX-T Manager must be deployed in the VI workload domain.

Procedure

1. Log in to the NSX-T Manager administrator interface. Notice the warning about the number of nodes in your NSX-T cluster.

2. Click the **Deploy Nodes** link next to the warning.

   The appliance view for NSX-T Manager is displayed. You should see the existing appliance in the target VI workload domain listed.
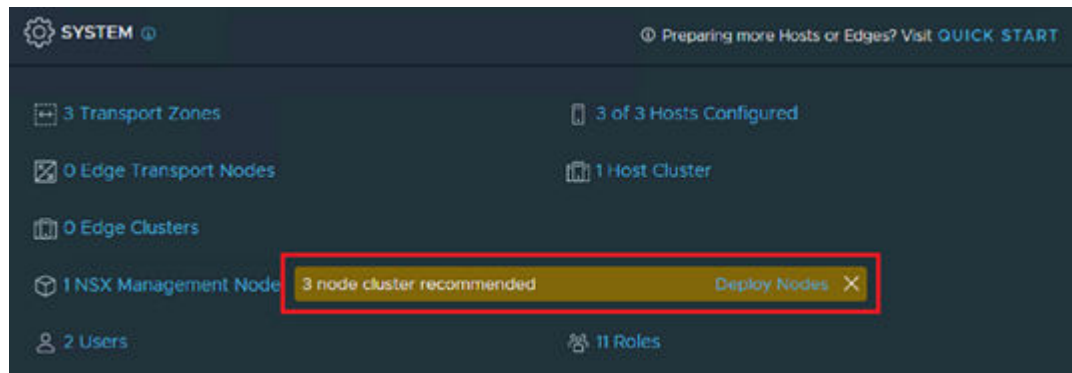
3. Click **Add NSX Appliance** to add additional NSX-T Manager appliances.
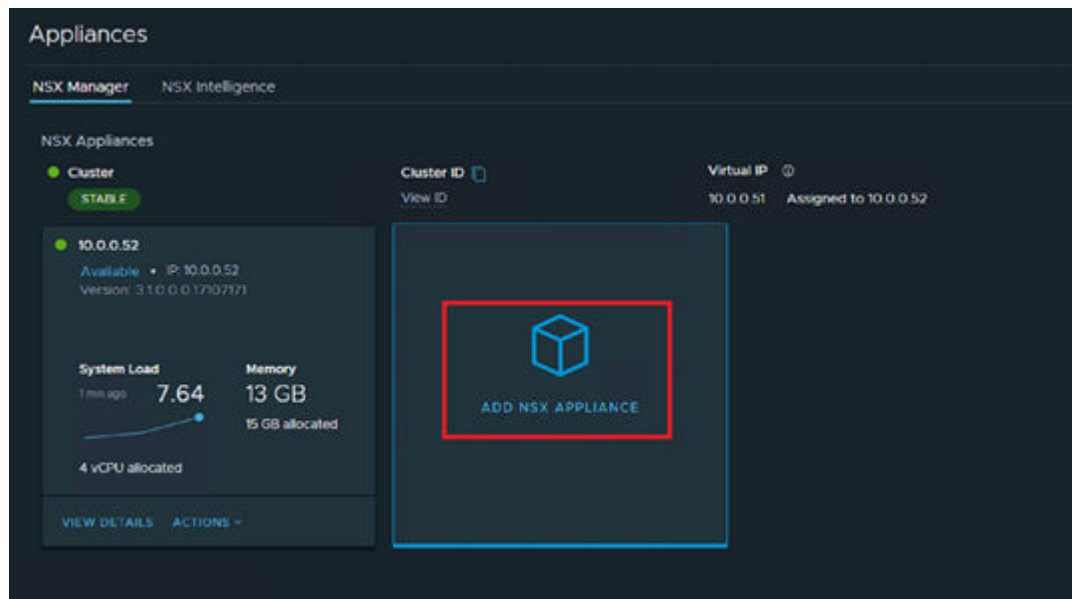
4. Enter the details for your second NSX-T Manager appliance. Ensure that you select a medium sized appliance. This is critical otherwise enough resources will not be available in NSX-T for proper enablement of workload management.

   The following figure shows an example of node details.

Reference Architecture Guide

5. Specify resource allocations within vSphere for the NSX-T Manager appliance.

    The vCenter and Compute Cluster should be pre-filled with correct values. For the **Datastore** and **Network** entries, configure parameters for your environment.

    The following figure shows an example of using the vSAN datastore and the management vDS port group for the selected VI workload domain.

6. Configure access and authentication credentials for the NSX-T Manager being deployed. Enter the same credentials used when configuring the new NSX-T instance during VI workload domain deployment.

The following figure shows an example of an NSX-T Manager appliance being configured to allow SSH and root login.
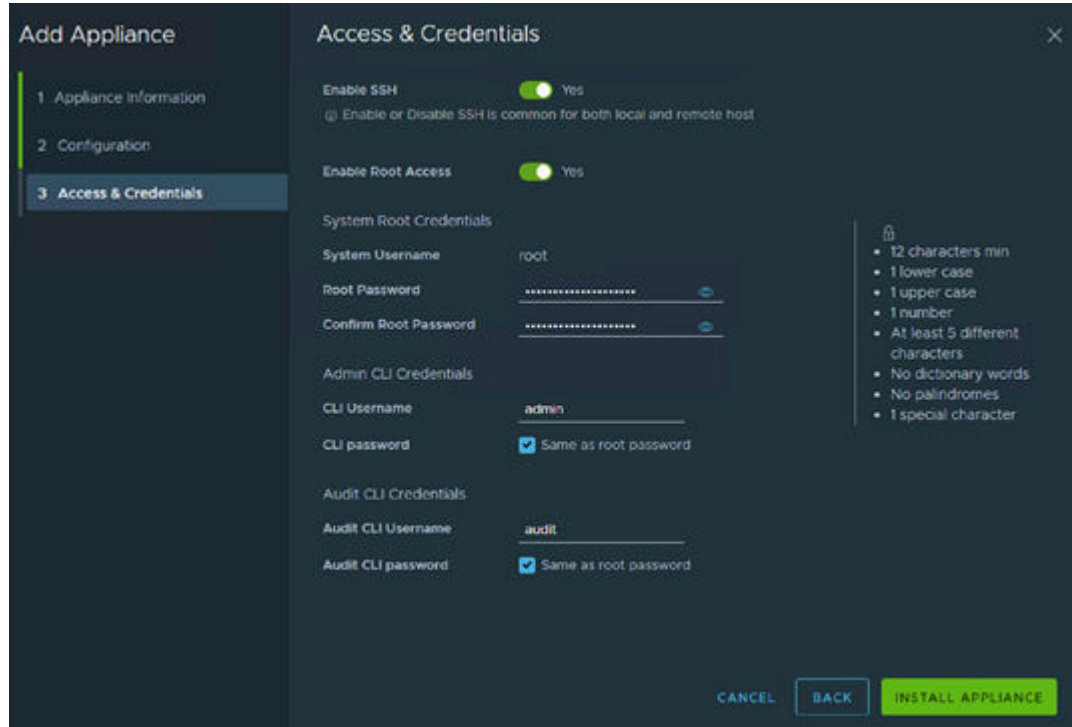
7. Deploy a third NSX-T Manager in the VI workload domain by repeating the process, replacing details such as FQDN and IP addresses with the correct values for the third node.

After you initiate the installation of the second and third NSX-T Manager appliances, you will see progress indicators in the appliance view within NSX-T Manager.
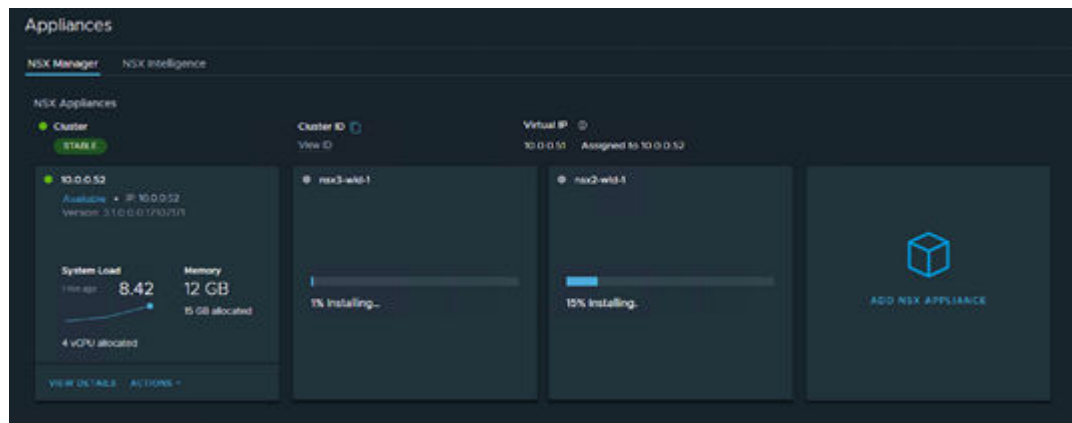
After the installation of the additional appliances is complete, you will see green health status indicators and associated metrics for all three NSX-T Managers.

Reference Architecture Guide
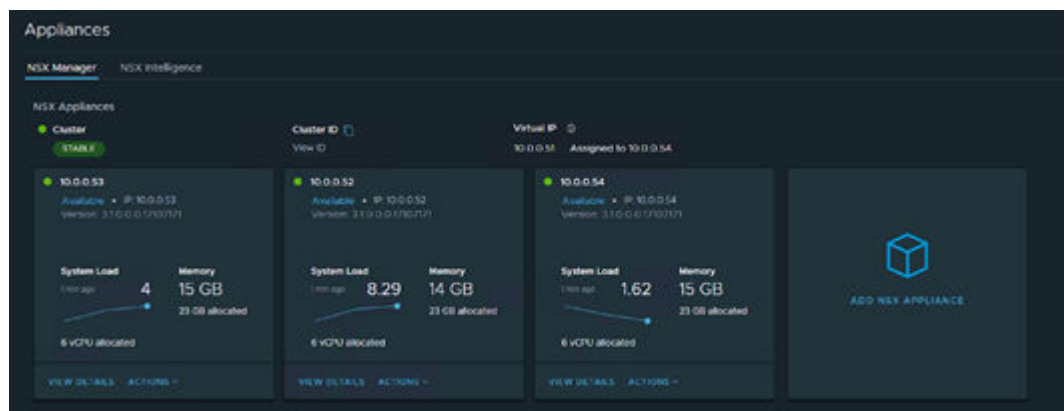
When all three nodes are showing healthy status and the cluster is stable, double-check resource allocations for the first NSX-T Manager appliance that was deployed by SDDC Manager during VI workload domain creation.

If the allocations do not match a medium appliance size for NSX-T (6vCPU/24 GB RAM), then:

- Shut the node down.
- Modify vCPU and memory.
- Power on the node.

Allow the NSX-T Manager appliance to fully boot and the NSX-T cluster to become stable again. This will take five to ten minutes.

> **Note:** You can log in to any of the NSX-T Manager appliances from the virtual machine console or over the network (if enabled) and run the get cluster status command to observe the status of each node's cluster services and overall cluster health.

Prepare BGP infrastructure for NSX-T Edge

Since a new NSX-T Edge cluster is necessary for enabling workload management on the target VI workload domain, you must prepare your BGP routing and network infrastructure to support the NSX-T Edge cluster. You must ensure that all of the following prerequisites are met:

- Separate VLANs and subnets are available for Host TEP VLAN and Edge TEP VLAN.
- VLANs for Host TEP and Edge TEP must be routable in the network infrastructure.
- BGP peer configurations must be configured on the Cisco or Arista ToR switches with interface IPs and matching ASN and BGP passwords.
- An ASN to assign to the NSX-T Edge cluster Tier0 interfaces is available.
- DNS entries are created for NSX-T Edge components (the two Edge cluster VMs that will be deployed).
- The vSphere cluster that will host the NSX-T Edge cluster must be L2 Uniform (must have identical management, uplink, Edge, and Host TEP networks).

Deploy an NSX-T Edge cluster

When all the prerequisites for the NSX-T Edge cluster are verified and are available, you can start the NSX-T Edge cluster addition wizard.

Procedure

1. From SDDC Manager, navigate to the details page of the VI workload domain.
2. From the **Actions** menu, select **Add Edge Cluster**.
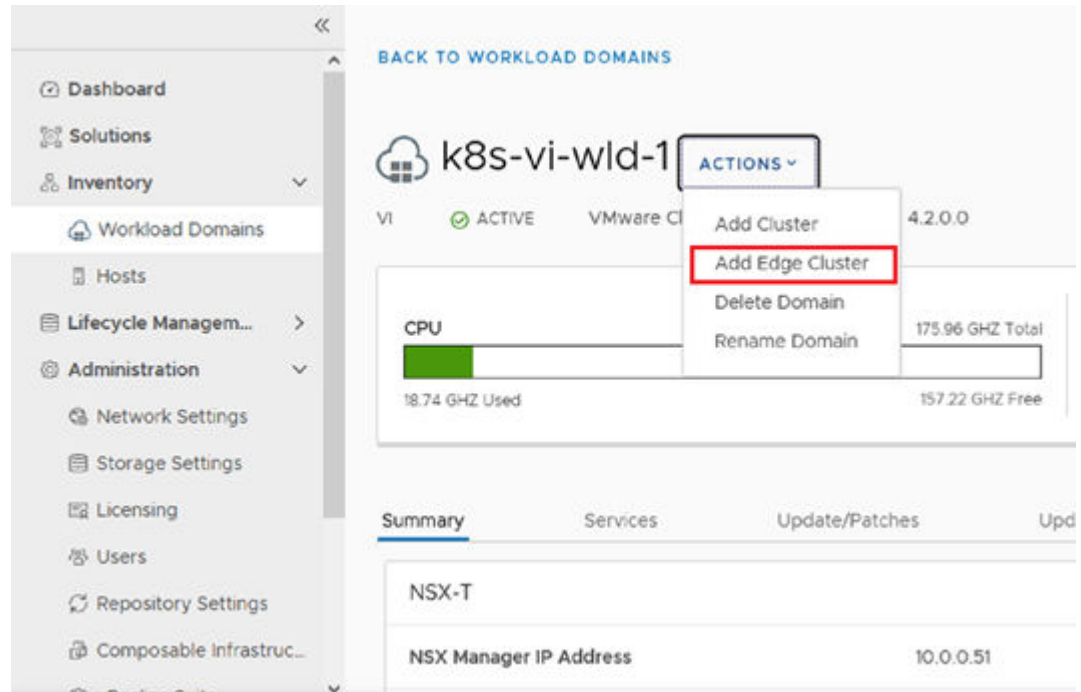
3. Enter the following:

   ▪ The name of your cluster.

   ▪ The MTU for your Edge cluster nodes.

   ▪ The ASN you reserved for assignment to the Tier0 interfaces.

   ▪ The names of the Tier 0 and Tier 1 routers.

   ▪ Credentials to be used for the NSX-T Edge cluster infrastructure.

4. For Use Case, select **Workload Management** for the use case, and ensure that the **Tier0 Routing Type** is configured for EBGP.

> **Note:** The **Edge Form Factor** and **Tier0 Service High Availability** configuration options are automatically set when selecting the Workload Management use case.

5. Enter details for the following:

   - The first Edge node Host and Edge TEP VLANs.

   - IP addresses, BGP interfaces.

   - ASN, and BGP Peer passwords.

6. When details for the first node are added, click **Add Edge Node** to commit the details for the first edge node.

7. Repeat this process for the second edge node with details specific to it and commit the details for the second edge node.

   When both nodes have been added successfully, you should see them at the bottom of the **Edge Node** pane of the Add Edge Cluster wizard.

8. View the summary of the edge cluster details, and then proceed to the Validation phase of the wizard. Wait until all items in the checklist turn green and the **Finish** button becomes available for use.

9. Click **Finish**.

10. If there are any issues with the edge cluster details, expand to view the reason for the failure and address the issue, and then run the Add Edge Cluster wizard again until it is successful.

    The following figure shows a configuration that has been successfully validated and is ready for SDDC Manager to initiate the workflow to add the NSX-T Edge cluster.
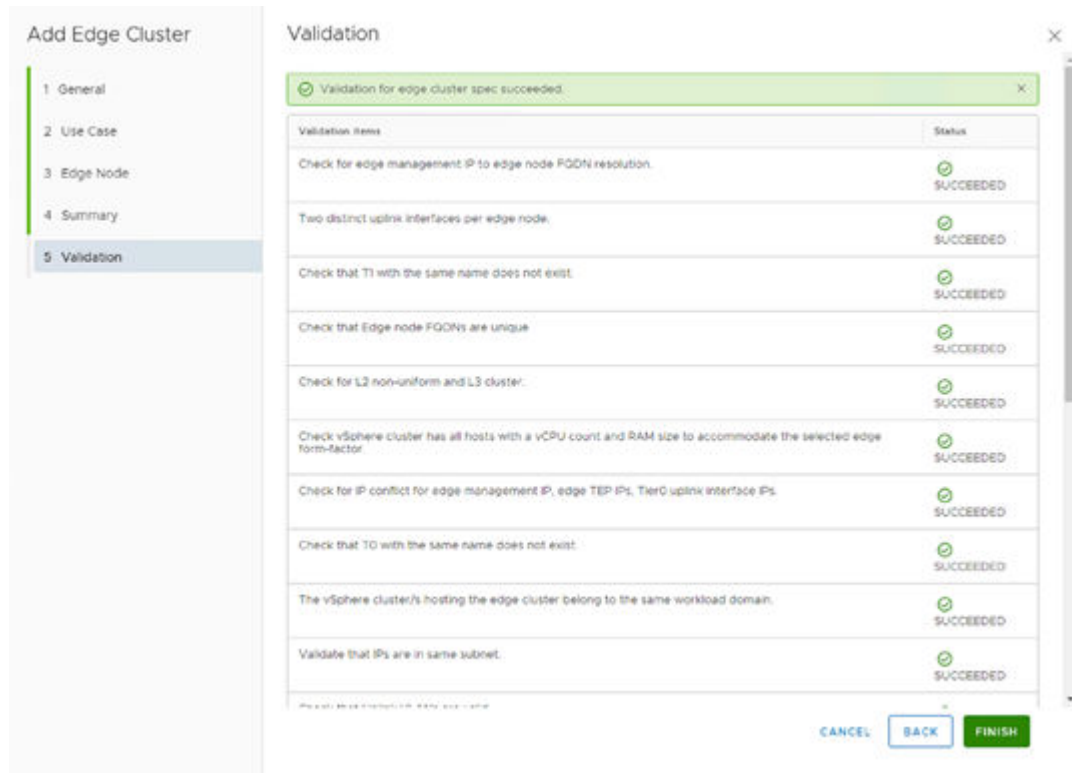
Wait for the SDDC Manager add NSX-T Edge cluster workflow to complete successfully.

Configure and enable Workload Management

Before enabling Workload Management, complete the following prerequisites:

- Ensure that the vSphere ESXi cluster that has Workload Management enabled has the proper vSphere for Kubernetes licensing applied.

- Deploy an NSX-T based VI workload domain as a target for Workload Management.

- Deploy an NSX-T Edge Cluster and confirm that it is available for the VI workload domain.

- Configure the following IP Addresses:

    - A subnet for pod networking in Tanzu that is non-routable and a minimum of /22.

    - A subnet for Service IP addresses in Tanzu that is non-routable and a minimum of /24.

    - A subnet for Ingress in Tanzu that is routable and a minimum of /27.

    - A subnet for Egress in Tanzu that is routable and a minimum of /27.

- Create an empty Content Library defined in the target vCenter hosting the VI workload domain and backed by storage accessible to each ESXi host in the VI workload domain that will have Workload Management enabled.

- Create a subscribed Content Library that subscribes to https://wp-content.vmware.com/v2/latest/lib.json.

Prepare the network and NSX-T infrastructure for VMware Tanzu

The Ingress and Egress network segments need to be routable not only on your BGP infrastructure, but also need to be defined as network segments within NSX-T and an interface defined on the NSX-T Edge Cluster Tier-1 gateway.

Procedure

1. Open a browser and navigate to the NSX-T Manager admin interface.
2. Log in with the Administrator account.
3. Click on the **Networking** tab at the top of the page, then navigate to Connectivity->Segments.
4. Click **Add Segment** and then configure the following options:

   - Segment Name: The name of the segment in NSX-T.

   - Connected Gateway: Select TIER1-Tier1.

   - Gateway CIDR IPv4: Assign the top IP address in your CIDR range in CIDR notation (that is, for 10.50.0.0/24, enter 10.50.0.254/24).

5. Click **Save**.

   The following figure shows example details of the new Ingress segment used when validating this solution.
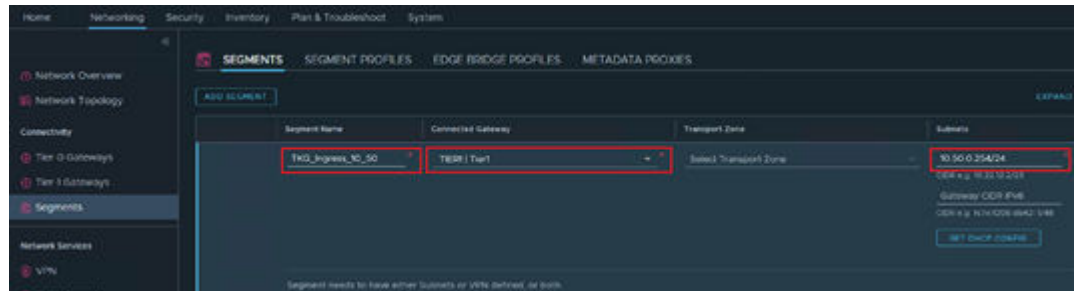
6. Repeat the previous step for the Egress network segment and configure the details for the name and Gateway CIDR IPv4 values. Keep the Connected Gateway value the same as the Ingress segment, and save the Egress segment in NSX-T.

> **Note:** Before you continue verify that you can ping the gateway CIDR IPv4 value from a machine that should be able to access the Ingress and Egress network segments using BGP routing. This ensures all Ingress and Egress networking is correctly setup prior to enabling Workload Management.

Enable Workload Management

Procedure

1. Open a browser and navigate to SDDC Manager, and then log in.
2. Navigate to the Solutions section in the left pane and click **Deploy** within the **Kubernetes Workload Management** box.
3. Review the prerequisites that appear, select them all, and then click **Begin**.
4. Select the target VI workload domain from the **Workload Domain** list.

5. Select the option for the compatible, available vSphere ESXi cluster that will be enabled.
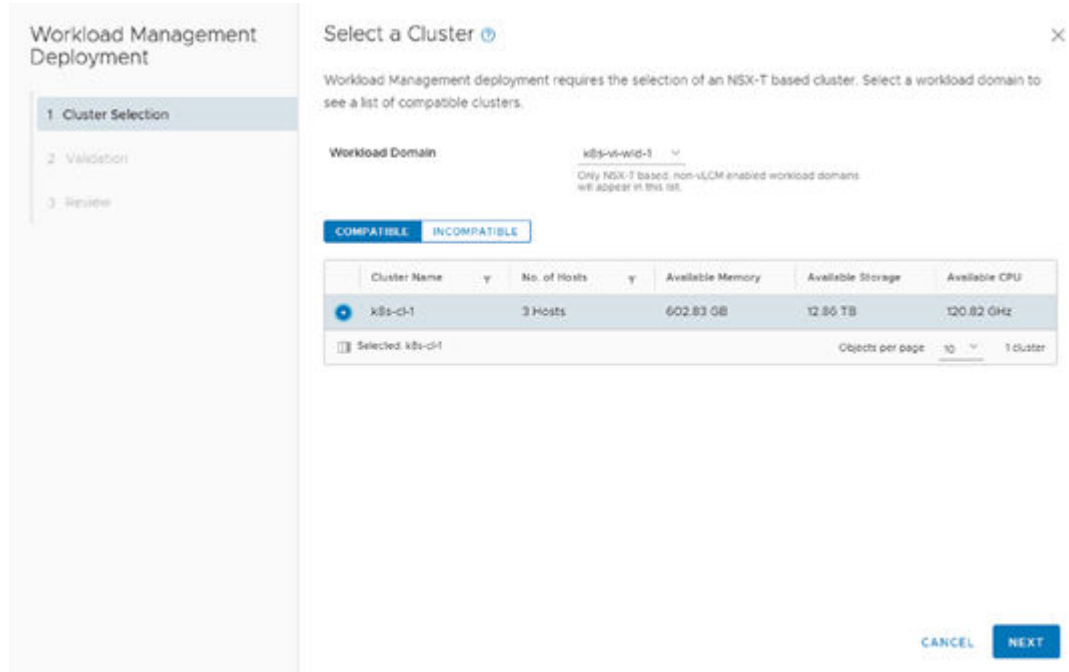
6. Wait for validation to complete successfully, and then click **Next**.

If any validation errors exist, review the reasons for failure and resolve the issue. Run the Workload Management Deployment wizard again.

The following figure illustrates a successful validation of the target VI workload domain and vSphere ESXi cluster on which Workload Management is being enabled.
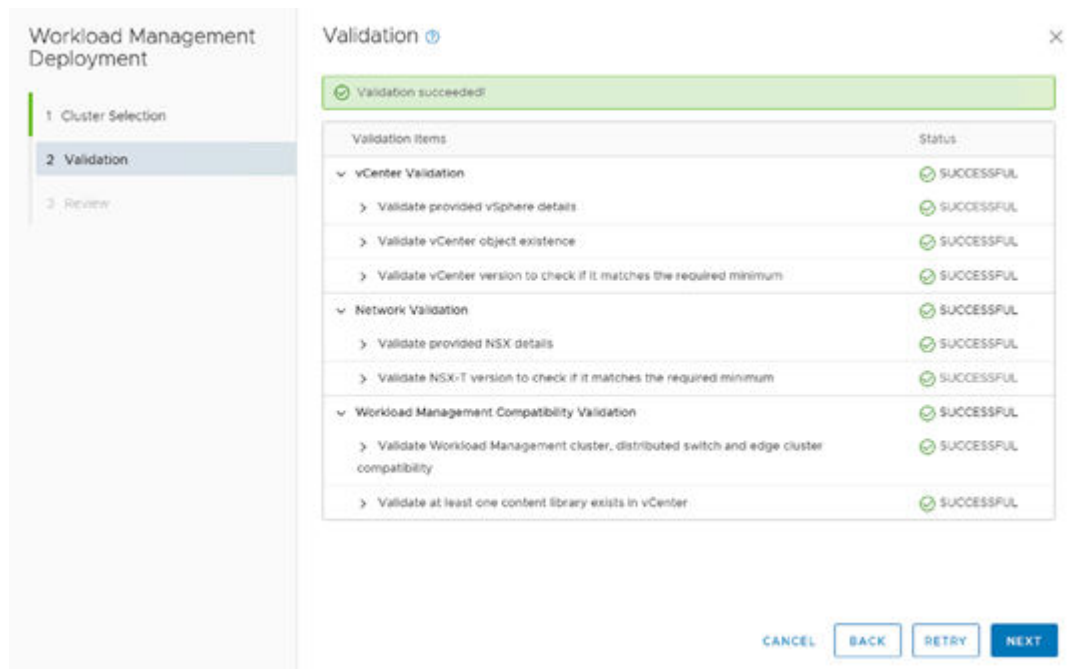
7. Review the summary, and then click **Complete in vSphere**.

This will open a new window to the vSphere Web Client.

8. In the Workload Management section of the vSphere client, complete enabling Workload Management on the target VI workload domain.

   The vSphere client Workload Management wizard appears automatically and prompts you to select a vCenter and networking stack option.

9. Select the target VI workload domain vCenter, ensure the NSX-T networking option is selected, and then click **Next**.
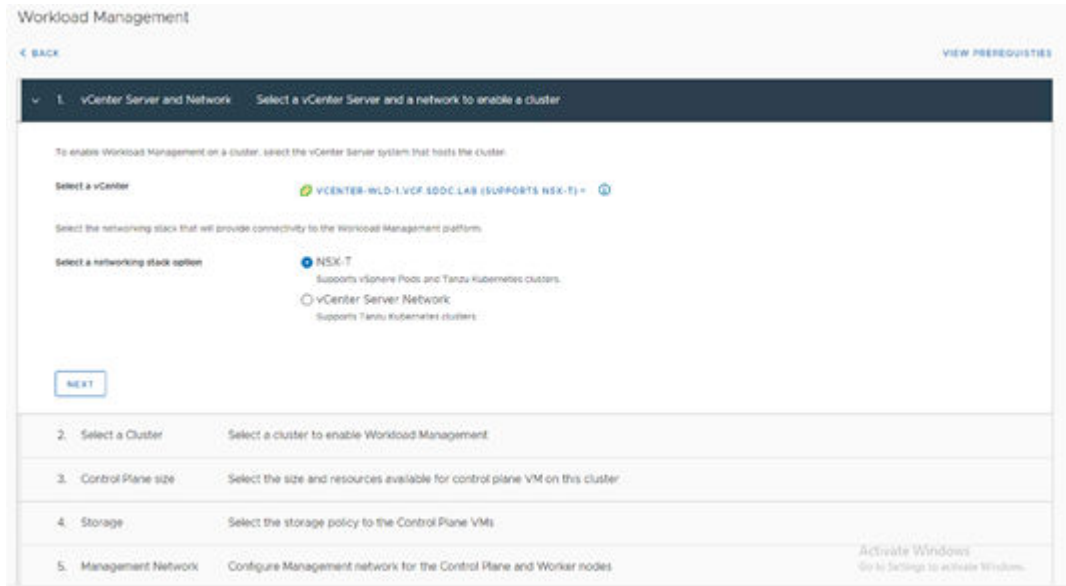
10. Select the compatible, available vSphere ESXi cluster that you want to enable Workload Management on, and then click **Next**.

    The following figure illustrates an example selection of a vSphere ESXi cluster.
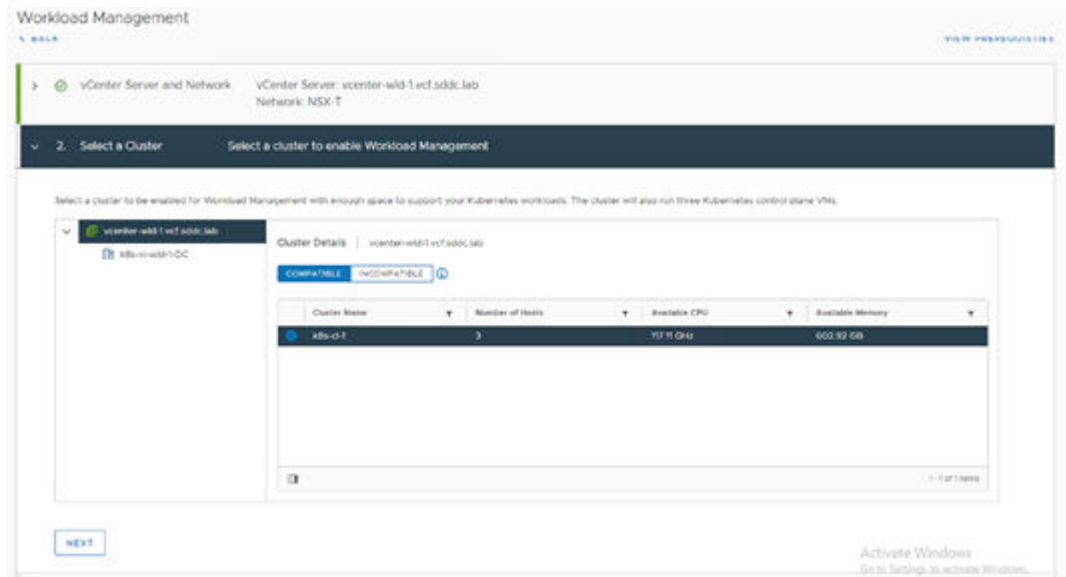
11. Select a size for the Control Plane of the Supervisor cluster within Workload Management, and then click **Next**.

    The following figure illustrates selecting a Small Control Plane size.
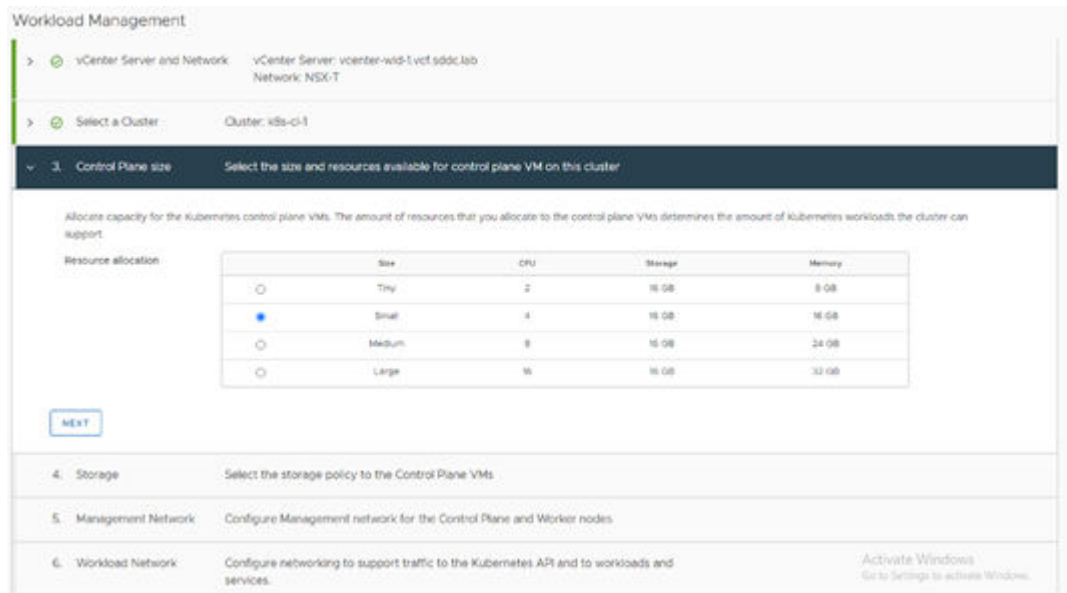
You can select different vSphere SPBM policies to control storage placement for different components within the Supervisor cluster. This allows you to select different storage targets for the Control Plane Nodes, Ephemeral Disks, and the Image Cache.

12. Select the appropriate storage policies for your environment, and then click **Next**.

The following figure shows a simple example of all components using the vSAN datastore via SPBM policies.
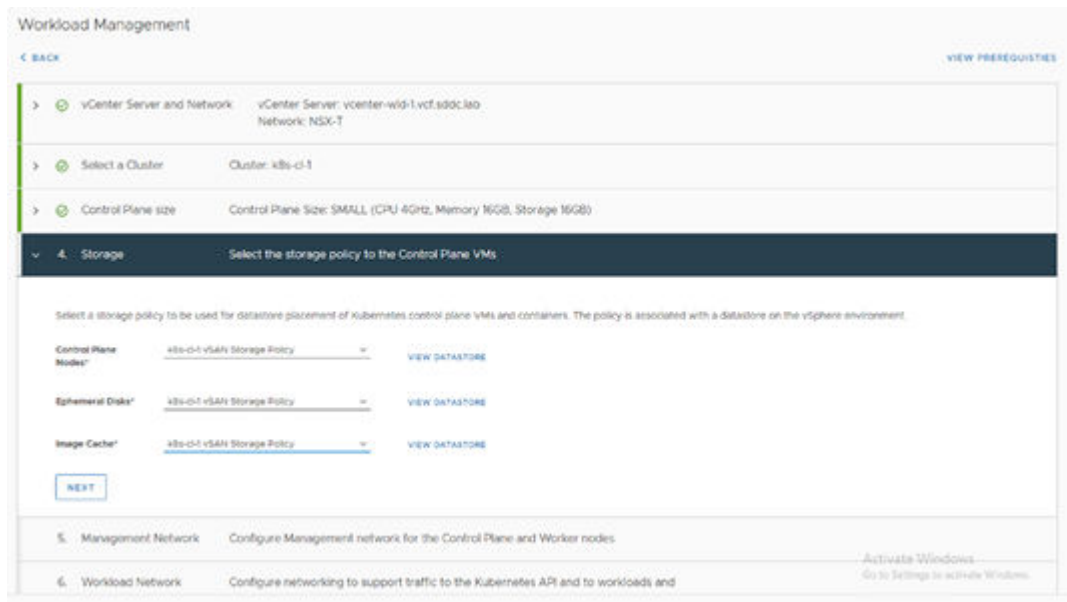
The Supervisor cluster Control Plane nodes need access to the management network on which the vCenter resides.

13. In the Management Network details section of the wizard, select the vDS port group that has access to your vCenter Server.

14. Enter a starting IP address that will be used to assign five consecutive IP addresses from this base address to the Supervisor Control Plane nodes and the Kubernetes API service.

    For example, a starting IP address of 10.0.0.170 would require that the IP addresses 10.0.0.170-174 be reserved for the Supervisor cluster.

15. Enter details for the remaining networking configuration for the management network components, and then click **Next**.
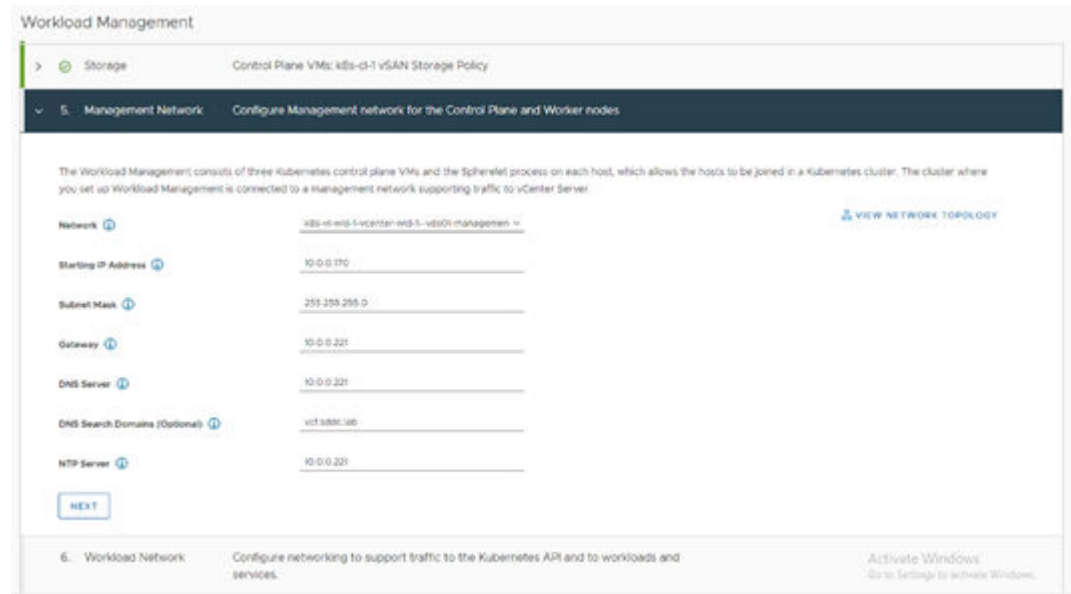
16. Enter details for the workload network.

17. Select the vSphere distributed switch for the vSphere ESXi cluster, the Edge Cluster name associated with the NSX-T deployment, and DNS server information.

    The subnets for the Pod CIDR and Service CIDR ranges are auto-populated based upon the size of the Control Plane nodes you selected earlier. You can modify these CIDR ranges if necessary, but most deployments should use the default ranges.

18. Enter your Ingress and Egress CIDR ranges that correspond to the Ingress and Egress network segments previously created in NSX-T and your BGP routing configuration, and then click **Next**.
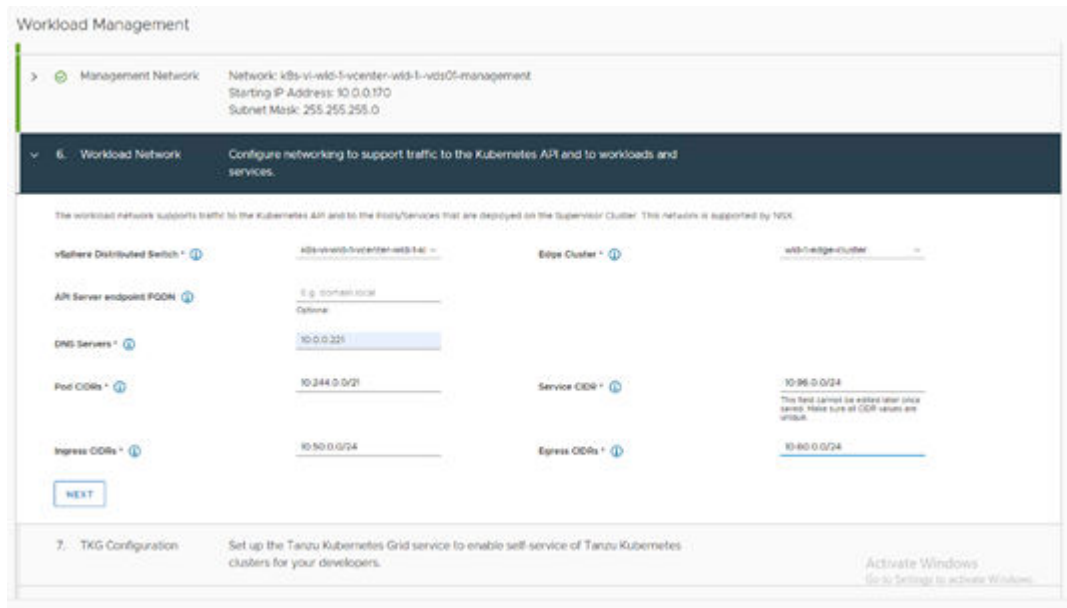
19. Click the **Edit** link to select the subscribed Content Library that you created earlier, and then click **Next**.
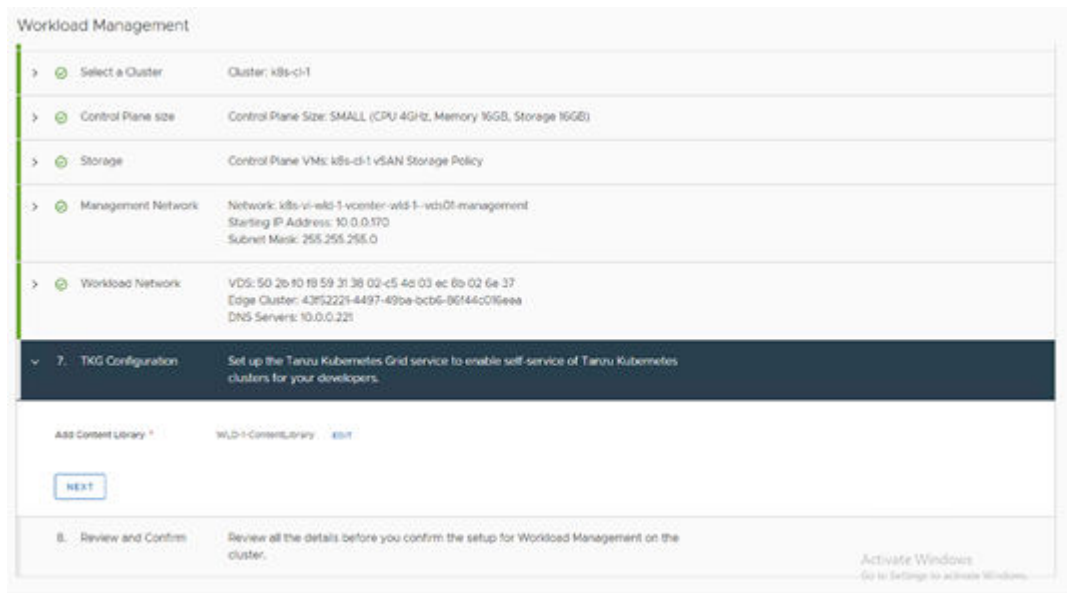
20. Click **Finish** to begin enabling Workload Management on the vSphere ESXi cluster.

    You can monitor the progress in the vSphere Web Client by highlighting the vSphere ESXi cluster, selecting the **Monitor** tab, and selecting **Namespaces** > **Overview**.

📄 **Note:** Workload Management can take between 30-60 minutes to enable, depending on hardware and network speeds. While errors may appear in the Overview section, they are typically transitory and will not persist. To monitor the process in detail or to troubleshoot, you can tail the wcpsvc.log file located at /var/log/vmware/wcp on the vCenter server to which the vSphere ESXi cluster is registered.

Configure vSphere Namespaces

Follow these procedures to create and configure the vSphere Namespace that will host vSphere pods or Tanzu Kubernetes Grid Services guest clusters.

Create vSphere Namespaces

Procedure

1. Open a browser and navigate to the vSphere Web Client for the vCenter hosting the Workload Management-enabled cluster.

2. Select **Menu** > **Workload Management**.

3. Click on the **Namespaces** link, and then click **New Namespace**.

4. Select the Workload Management enabled cluster and provide a name and optional description for the new vSphere Namespace, and then click **Create**.
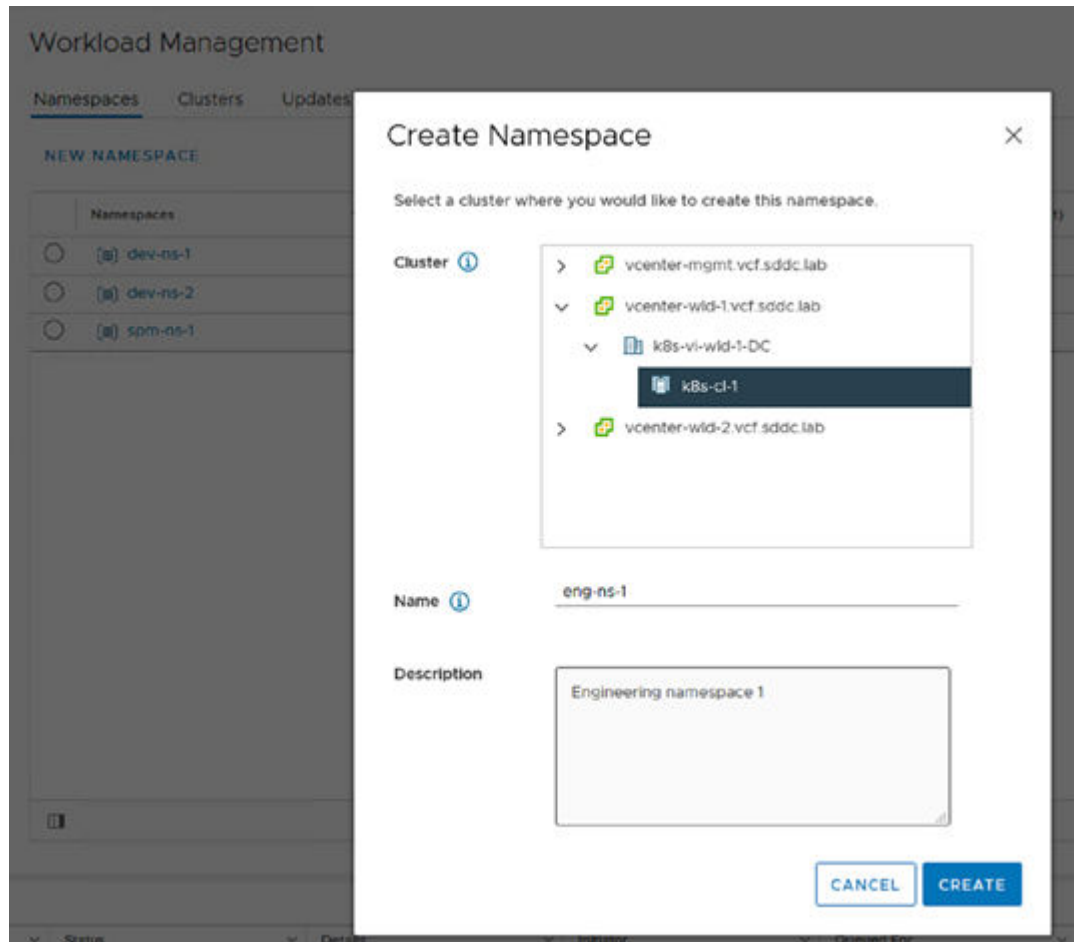
The following figure illustrates the Summary view of a newly created vSphere Namespace.
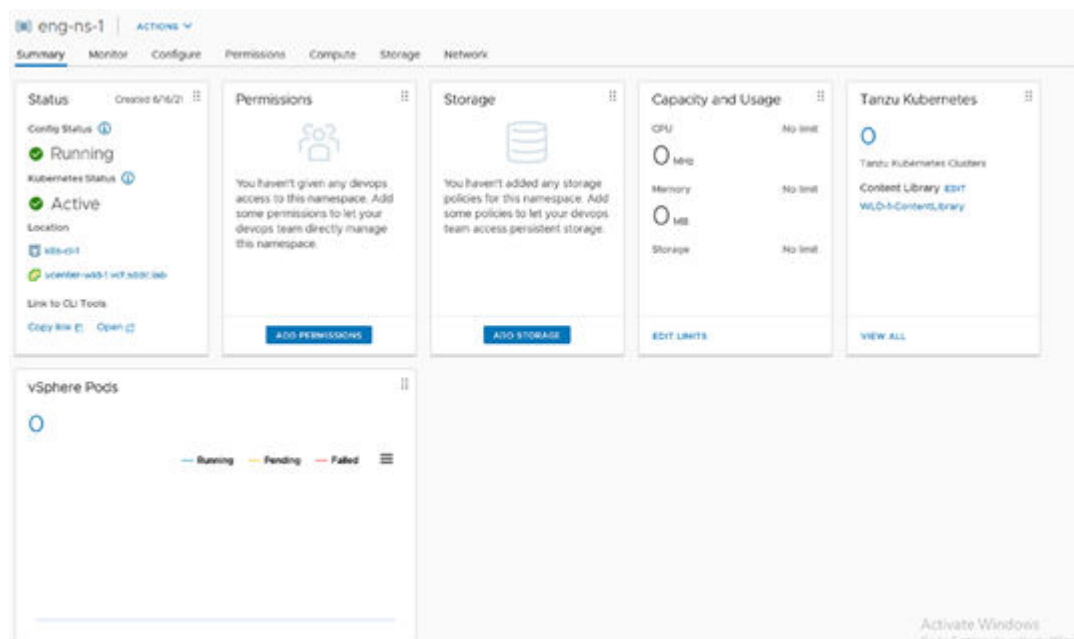
Assign vSphere Namespace permissions to vSphere users

You can assign two permission roles to vSphere Namespaces: Edit and View. To deploy resources and configure TKG using standard kubectl commands, you must assign a user that has edit permissions to the vSphere Namespace.

Procedure

1. Within the Permissions pane of the Namespace summary, click **Add Permissions**.
2. Select a user from the vSphere SSO domain.
3. Select the **Can edit** role.

Add Permissions ✕

Add a user or a group to give access to this namespace

Identity source	vsphere.local ⌄
User/Group Search	🔍 Administrator
Role	Can edit ⌄

CANCEL   OK

Configure storage policy mapping to vSphere Namespace

vSphere SPBM policies are used to automatically create StorageClasses in the vSphere Namespaces.

Procedure

1. From the Summary view of the vSphere Namespace, click **Add Storage**.

   All vSphere SPBM policies that are configured for the underlying vSphere ESXi cluster are displayed.

2. Select the checkbox next to the policies that you want to be passed down to this namespace and be made available from StorageClasses at the Kubernetes layer.

   The following figure shows an example selection of policies that target the default local vSAN datastore, vVol datastores on two separate Hitachi Virtual Storage Platform storage systems, and VMFS datastores on two separate Hitachi Virtual Storage Platform storage systems.

Select Storage Policies   ✕

		Storage Policy	Total Capacity	Available Capacity
☐	>	VM Encryption Policy	16.65 TB	15.89 TB
☑	>	vSAN Default Storage P...	10.92 TB	10.26 TB
☐	>	VVol No Requirements ...	1.39 TB	1.35 TB
☐	>	Management Storage P...	10.92 TB	10.26 TB
☐	>	Management Storage p...	10.92 TB	10.26 TB
☐	>	Management Storage P...	10.92 TB	10.26 TB
☐	>	Management Storage p...	10.92 TB	10.26 TB
☐	>	k8s-cl-1 vSAN Storage P...	10.92 TB	10.26 TB
☑	>	Hitachi vVol - Tier2 IOPS	713.93 GB	703.75 GB
☑	>	Hitachi vVol - Tier1 IOPS	713.93 GB	675.62 GB
☑	>	Hitachi VMFS - Tier1 IOPS	511.75 GB	510.34 GB
☑	>	Hitachi VMFS - Tier2 IOPS	1,023.75 GB	1,022.33 GB
☑ 5			1 - 10 of 12 items |< < [1] / 2 > >|

CANCEL   OK

3.

Configure resource limits for vSphere Namespace (optional)

You can limit the underlying compute and storage resources consumed by Kubernetes entities within the vSphere Namespace by configuring resource limits on the vSphere Namespace.

Procedure

1. From the Summary view of the vSphere Namespace, click the Edit Limits link in the Capacity and Usage box to configure the resource limits for the vSphere Namespace.

2. You can place limits on maximum CPU consumption (GHz), memory consumption (MB/GB), and storage consumption (MB/GB) as needed.

   Note that the storage limit is against all of the storage backing the SPBM policies configured for the Namespace, and not on an individual per-SPBM policy basis.

3.

Download CLI tools and connect to vSphere Namespace

Access to Tanzu Kubernetes Grid Services is performed through a modified kubectl tool that is available for download from the infrastructure that you just deployed.

1. From the Summary view of the vSphere Namespace, click the Open link in the Status box under the Link to CLI Tools heading.

2. Download the correct package for your workstation type and follow the installation instructions.

   If you are planning to use Velero as documented in this solution, it is recommended that you use a Linux workstation because some of the Velero tools are only available on that platform.

To log in to the new vSphere Namespace, issue the following command with information specific to your environment:

```
kubectl-vsphere login --insecure-skip-tls-verify --server https://<IP of Kube API> --
tanzu-kubernetes-cluster-namespace=<vSphere Namespace> --vsphere-username <username
of user with "Can edit" permissions in Namespace>
```

📄 **Note:** The --insecure-skip-tls-verify flag is only necessary if you are using self-signed certificates. If you are using certificates issued by your organization that are trusted by your workstation, you should remove this flag to ensure proper secure communications within your organization.

An example of logging in to the vSphere Namespace at Kube API IP 10.50.0.1, Namespace eng-ns-1, and username first.last@vsphere.local is shown in the following figure.

Notice the multiple contexts listed in the output. You should have access to the Supervisor cluster context (10.50.0.1 in the example) as well as the vSphere Namespace you have created and configured. Try switching contexts by using the `kubectl config use-context <your Namespace context>` command.

This switches the target of your `kubectl` commands to the newly created vSphere Namespace. Issue the `kubectl get sc` command to see the StorageClasses that were automatically created based on the vSphere SPBM policies that you configured for use in the vSphere Namespace.

The following figure shows an example of switching `kubectl` contexts and listing the StorageClasses in the vSphere Namespace.
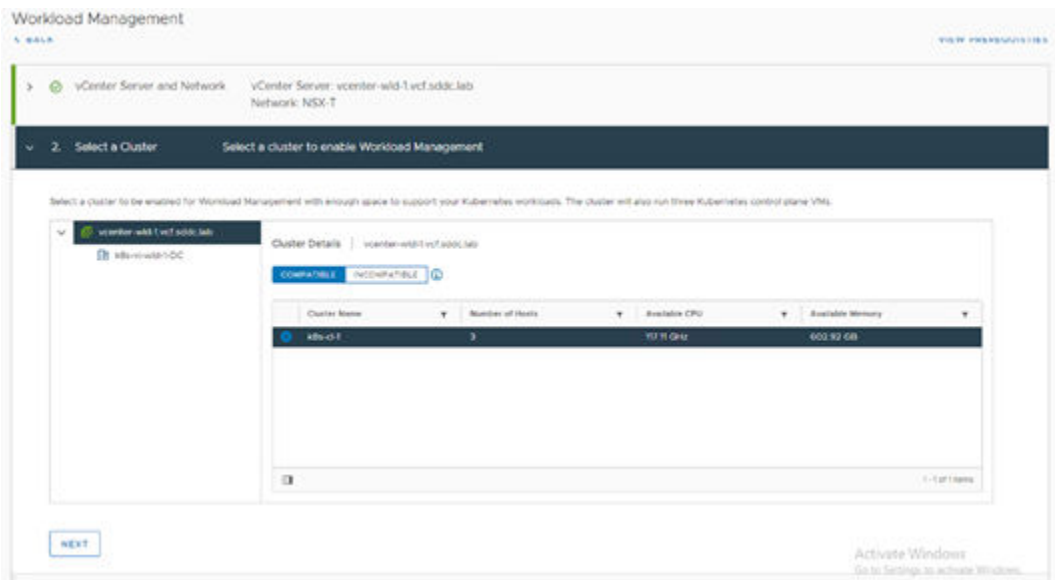
Register management clusters with Tanzu Mission Control

Register your Tanzu Kubernetes Grid Services Supervisor cluster with Tanzu Mission Control for a comprehensive view and control of your Kubernetes resources across multiple infrastructure types. This will allow you to deploy a TKG guest cluster from the TMC SaaS console later in the deployment.

Create Cluster Groups in Tanzu Mission Control

Procedure

1. Open a web browser and log in to your Tanzu Mission Control portal.
2. In the left pane, select **Cluster Groups**, and then click **Create Cluster Group**.
3. Enter a name and description for your cluster group, and optionally add any tags for the group.
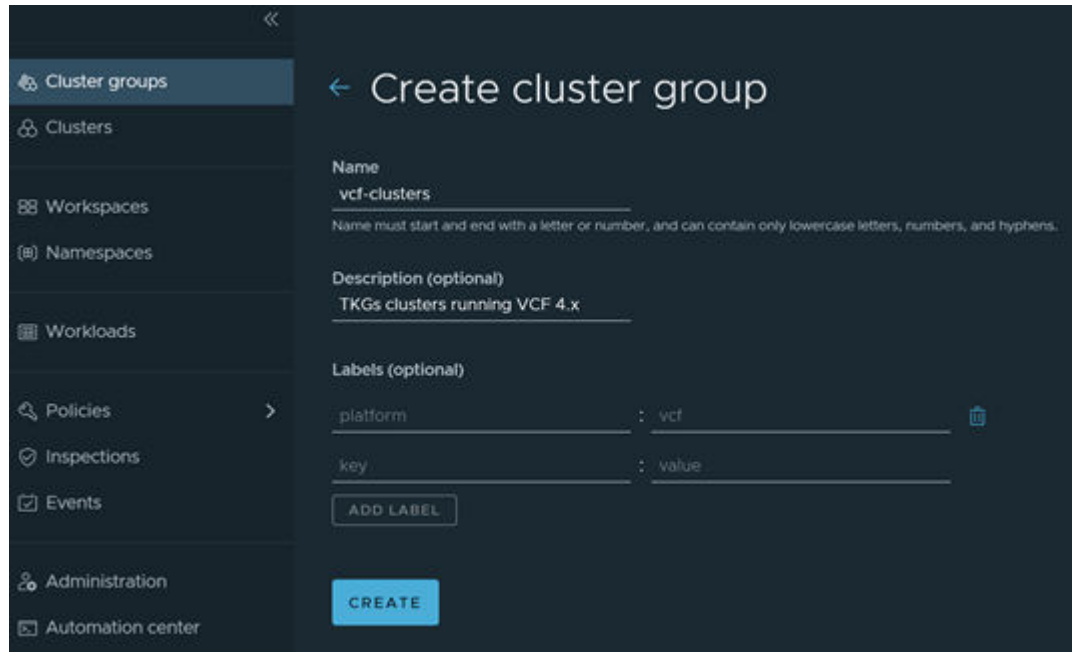
Register Supervisor clusters to Tanzu Mission Control

Procedure

1. Navigate to **Administration** > **Management clusters** > **Register Management Cluster** > **vSphere with Tanzu**.
2. Enter a name for the management cluster, select a default cluster group for it, optionally enter a description and tags, and then click **Next**.
3. Copy the registration URL that is displayed, and save it for later use. On your Linux workstation, log in to your Supervisor cluster context and issue the following command:

```
kubectl get ns
```

4. Look for a Kubernetes namespace beginning with tmc-svc (that is, tmc-svc-c8) and note the full name for later use.
5. Create a file named `tmc-cluster-register.yaml` on your Linux workstation with the following content, replacing the namespace name and the registration link with your specific information:

The following is the text of the *yaml* file:

```
apiVersion: installers.tmc.cloud.vmware.com/v1alpha1
kind: AgentInstall
metadata:
  name: tmc-agent-installer-config
  namespace: svc-tmc-c8
spec:
  operation: INSTALL
  registrationLink: https://hitachivantara.tmc.cloud.vmware.com/installer?
id=02aa8fe593009ff8516cb00eec97c7a8c
6dca9a07e6e527f40473a90a5df8c97&source=registration&type=tkgs
```

6. To begin registration of your Supervisor cluster to TMC, enter the following command using your Supervisor cluster context:

```
kubectl apply -f tmc-cluster-register.yaml
```

7. Check local registration status by entering the following command:

```
kubectl -n <your tmc-svc namespace> describe agentinstall tmc-agent-installer-
config
```

8. Switch back to your browser with your TMC session, and click **View Management Cluster**.

9. On the following page, click **Verify Connection** until the summary view of the Management Cluster is displayed. Wait for all components to turn green and the details to be displayed.
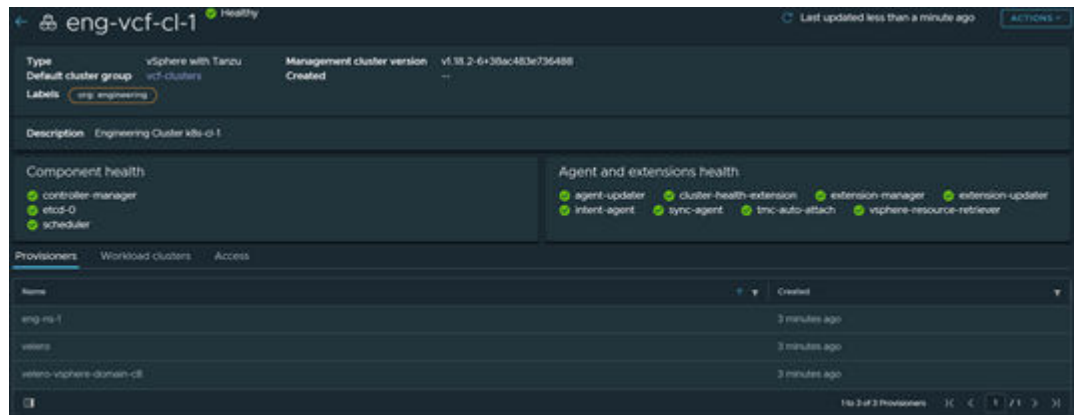
Deploy Tanzu Kubernetes Grid guest clusters

Now that a vSphere Namespace is deployed, configured, and accessible, you can deploy Tanzu Kubernetes Grid Services guest clusters to run workloads. Two TKG guest clusters are deployed as part of this solution. You must deploy a TKG guest cluster from your Linux workstation. Later, you must create a TKG guest cluster from the TMC portal.

Determine TKG Guest Cluster Resources

Tanzu Kubernetes Grid Services guest cluster node VMs are comprised of PhotonOS-based virtual machines, and TKG uses *virtualmachineclasses* to define the VM CPU/memory configuration used for the underlying node. To explore these classes, enter the following command with your `kubectl` context set to your vSphere Namespace:

```
kubectl get virtualmachineclasses
```


Note that the *virtualmachineclasses* are broken up into best-effort (no dedicated allocation) and guaranteed (full dedicated allocation) as well as different sizes for each type. To view resource allocation details for each *virtualmachineclass*, you can describe a specific *virtualmachineclass* using `kubectl` and view the CPU/memory allocation as well as any CPU/memory reservations that will be enforced at the vSphere layer.

The following figure shows example output from the following command, which shows 2vCPU/2 GB configuration for the VM, and a reservation of 2000MHz and 2Gi of RAM:

```
kubectl describe virtualmachineclass guaranteed-xsmall
```

Create Tanzu Kubernetes Grid Services guest cluster YAML definition and deploy cluster

To create a TKG guest cluster, you must define your cluster configuration in YAML format, and then apply it using kubectl to create the cluster within your vSphere Namespace. Different configuration options are available for deploying TKG guest clusters. For details, see the VMware documentation.

To validate this solution, we deployed a 3/3 master/worker TKG guest cluster into our namespace, and dedicated the *etcd* database on each master node to have a persistent volume. The following figure shows the YAML script used to create our TKG guest cluster.

```
apiVersion: run.tanzu.vmware.com/v1alpha1
kind: TanzuKubernetesCluster
metadata:
  name: eng-cluster-1
  namespace: eng-ns-1
spec:
  distribution:
    version: v1.18
  topology:
    controlPlane:
      count: 3
      class: best-effort-medium
      storageClass: vsan-default-storage-policy
      volumes:
        - name: etcd
          mountPath: /var/lib/etcd
          capacity:
            storage: 4Gi
    workers:
      count: 3
      class: best-effort-medium
      storageClass: vsan-default-storage-policy
  settings:
      network:
        cni:
            name: antrea
        services:
          cidrBlocks: ["10.112.0.0/12"]
        pods:
          cidrBlocks: ["10.128.0.0/16"]
```

The following is the text of the *yaml* file:

```
apiVersion: run.tanzu.vmware.com/v1alpha1
kind: TanzuKubernetesCluster
metadata:
  name: eng-cluster-1
  namespace: eng-ns-1
spec:
  distribution:
    version: v1.18
  topology:
    controlPlane:
      count: 3
      class: best-effort-medium
      storageClass: vsan-default-storage-policy
      volumes:
        - name: etcd
```

```
          mountPath: /var/lib/etcd
        capacity:
          storage: 4Gi
  workers:
    count: 3
    class: best-effort-medium
    storageClass: vsan-default-storage-policy
 settings:
     network:
       cni:
          name: antrea
       services:
         cidrBlocks: ["10.112.0.0/12"]
       pods:
         cidrBlocks: ["10.128.0.0/16"]
```

Note that this example cluster is named *eng-cluster-1* and will be deployed in the *eng-ns-1* namespace. It will be deployed with the latest authorized version of 1.18 Kubernetes from VMware. The three control plane master nodes will be deployed using the best-effort-medium *virtualmachineclass*. They will be deployed on the vSAN datastore, and then each will have a 4Gi PV for its respective copy of the etcd database. The three worker nodes will also be deployed using the best-effort-medium *virtualmachineclass* and will be deployed on the vSAN datastore. The Antrea CNI network plugin will be used, and the pod/service CIDR blocks to be used for the cluster are defined.

To create the cluster, run the following command:

```
kubectl create -f <your TKGs cluster YAML file>
```

You will see the nodes starting to be deployed in the vSphere client for your master/worker nodes. To check the progress of the cluster deployment, switch context to the vSphere Namespace where the TKG guest cluster is being deployed, and then enter the following command:

```
kubectl describe tanzukubernetescluster <your TKGs guest cluster name>
```

The following figure shows output from the `kubectl describe` command for viewing the progress of the cluster deployment and shows the TKG guest cluster deployment is complete.

```
root@vcf-ubuntu-1: ~
    Name:       CoreDNS
    Status:     applied
    Version:    v1.6.7_vmware.8
  Proxy:
    Name:       kube-proxy
    Status:     applied
    Version:    1.18.15+vmware.1
  Psp:
    Name:       defaultpsp
    Status:     applied
    Version:    v1.18.15+vmware.1-tkg.2.ebf6117
 Cluster API Status:
   API Endpoints:
     Host:   10.50.0.3
     Port:   6443
   Phase:    Provisioned
 Node Status:
    eng-cluster-1-control-plane-g9glb:             ready
    eng-cluster-1-control-plane-n888b:             ready
    eng-cluster-1-control-plane-tgblc:             ready
    eng-cluster-1-workers-bl5jc-645c968b67-6tdhr:  ready
    eng-cluster-1-workers-bl5jc-645c968b67-m82mx:  ready
    eng-cluster-1-workers-bl5jc-645c968b67-p6kmw:  ready
  Phase:                                           running
  Vm Status:
    eng-cluster-1-control-plane-g9glb:             ready
    eng-cluster-1-control-plane-n888b:             ready
    eng-cluster-1-control-plane-tgblc:             ready
    eng-cluster-1-workers-bl5jc-645c968b67-6tdhr:  ready
    eng-cluster-1-workers-bl5jc-645c968b67-m82mx:  ready
    eng-cluster-1-workers-bl5jc-645c968b67-p6kmw:  ready
Events:                                            <none>
```

To control and view resources within the new Tanzu Kubernetes Grid Services guest cluster, you must add an additional flag (`--tanzu-kubernetes-cluster-name=`) to your `kubectl vsphere login` command.

Run the following command to log out of the Supervisor cluster:

```
kubectl vsphere logout
```

Then log in again, adding the `--tanzu-kubernetes-cluster-name=<your TKGs cluster name>` flag.

The following figure shows the new login and the new eng-cluster-1 context that is available. It also shows the TKG guest cluster context displaying the deployed nodes in the TKG guest cluster.

Configure on-premises data protection operators

With VCF 4.2, the Velero operator is available as a native supervisor service that can be enabled for backup/restore of container-based applications and their associated PVs and data. Complete the following procedures to configure and enable on-premises Velero functionality.

Choosing a target S3 object store for backups

While any AWS-compliant S3 target can be used as a Velero backup target, it is recommended that you use HCP for cloud scale due to the enterprise-grade compliance, security, retention, and replication features available. Built-in MinIO and Cloudian object stores are now available as operators in VCF 4.2 in addition to Velero; however, they must run within the TKG infrastructure on which you are performing data protection actions.

Similar to vSAN witness appliances and associated VMware placement recommendations, it is recommended that you back up to a remote S3 target separate from your source infrastructure. This ensures maximum protection of your data in case of a site or infrastructure-specific failure.

Configure Hitachi Content Platform for cloud scale

You can obtain a 60-day trial of Hitachi Content Platform for cloud scale (HCP for cloud scale) by visiting https://trycontent.hitachivantara.com/. Follow the directions in your trial access email after registering to generate credentials and to create an S3 bucket to be used as a target for Velero backup data.

See Hitachi Content Platform for Cloud Scale Architecture Fundamentals for more information.

Enable Velero Operator on the Supervisor cluster

Procedure

1. Open a browser and navigate to the vSphere Web Client for the vCenter hosting the Workload Management enabled cluster.

2. Select the cluster in the left pane, and then navigate to **Configure** > **Supervisor Services** > **Services**.

3. Select the radio button next to the **Velero vSphere Operator** and then click **Enable**.

4. Leave all information blank, and then click **Next**.

5. Accept the terms and license agreement, and then click **Finish**.

   You should see a new vSphere Namespace created in your vSphere ESXi cluster named *velero-vsphere-domain-c8* (the last two digits might be different).

6. From the vSphere Web Client, highlight the new velero-vsphere-domain Namespace in the left pane and navigate to **Compute** > **Core Kubernetes** > **vSphere Pods**.

   In the right pane, you should see four Velero pods running.
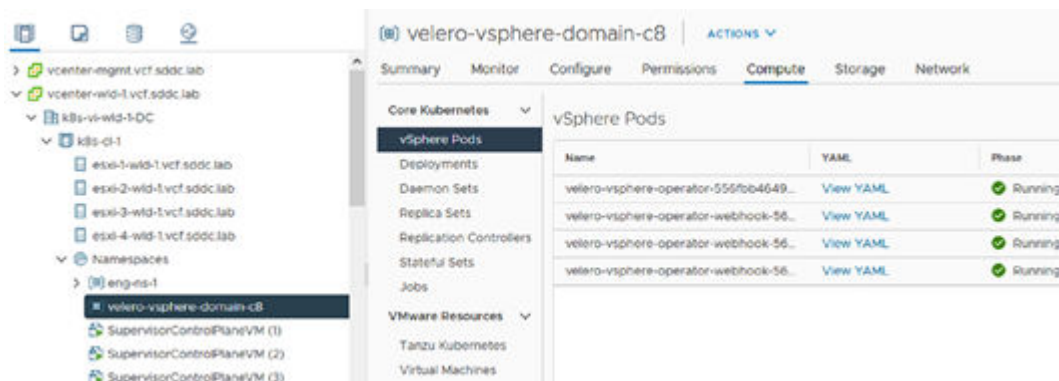
Install Velero into the Supervisor cluster

Procedure

1. From the **Menu** dropdown in the vSphere Web Client, select Workload Management.

2. Click on the **Namespaces** link, and then click **New Namespace**.

3. Create a new vSphere Namespace called *velero* in the target VI workload domain cluster, add the vSAN Default Storage Policy for storage, and assign the user administrator@vsphere.local**Can edit** permissions.

   This creates the target vSphere Namespace to install the Velero pod and backup driver into the Supervisor Cluster.

4. Install the Velero client and Velero vSphere Operator CLI onto your Linux-based workstation that has the vSphere `kubectl` binary installed. Installation steps are listed at https://github.com/vmware-tanzu/velero-plugin-for-vsphere/blob/main/docs/velero-vsphere-operator-user-manual.md#installing-velero-on-supervisor-cluster.

   You can find the latest Velero client at https://github.com/vmware-tanzu/velero/releases, and v1.1.1 of the Velero vSphere Operator CLI at https://github.com/vmware-tanzu/velero-plugin-for-vsphere/releases/tag/v1.1.1.

5. Install and copy to your /usr/bin filesystem location or add their locations to your $PATH variable.

6. Create a file with the HCP for cloud scale S3 access key and secret key from your HCP for cloud scale trial for the Velero vSphere installer.

The following figure shows an example of a credentials file that will be referenced during installation of Velero saved to /root/velero-hcpcs-credentials on the Linux workstation.

The following is the text of the *yaml* file:

```
[default]
aws_access_key_id = 38x9excf1agzUw1K8ieBCXT9JB5RXmjCQEl3TlfUVDwRv5ReHrWHBUoigVM-
VDtzcADL2ffiB8Nx3ak
aws_secret_access_key = 4e918208-bdd3-4e95-9fde-f39612280dde
```

To install Velero into the new namespace, you can create a script instead of typing out all of the credentials and details for your S3 target. The following figure shows an example script for installing Velero into the velero vSphere Namespace.

The following is the text of the *yaml* file:

```
#!/bin/sh
NAMESPACE="velero"
BUCKET="onprem-vcf-velero-target"
REGION="us-west-1"
S3URL="https://tryhcpforcloudscale.hitachivantara.com/"
VSPHERE_PLUGIN=vsphereveleroplugin/velero-plugin-for-vsphere:1.1.0
SECRET_FILE=/root/velero-hcpcs-credentials
velero-vsphere install \
--namespace $NAMESPACE \
--provider "aws" \
```
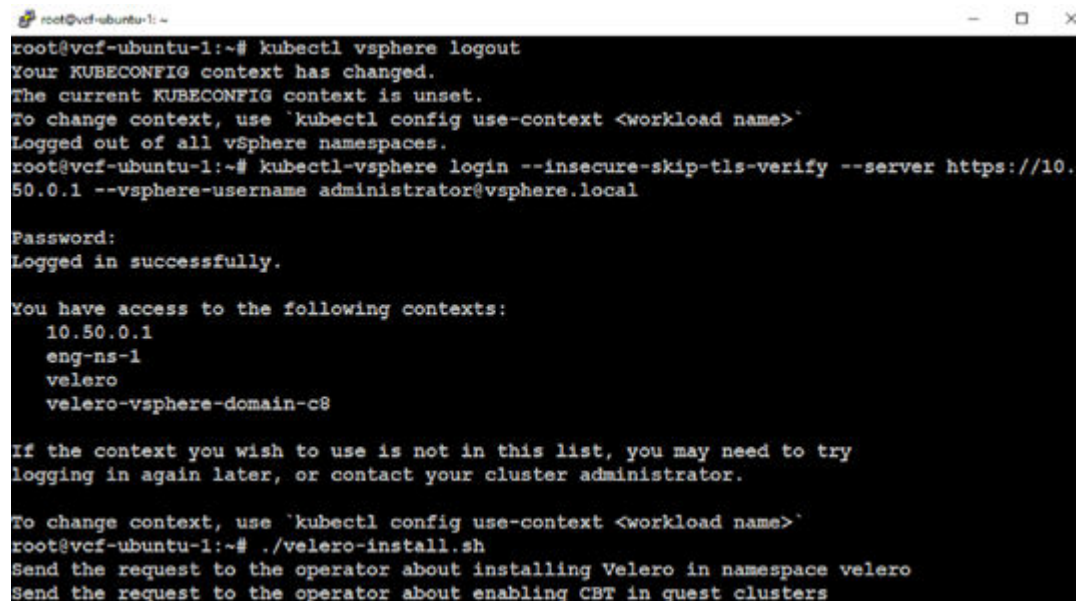
```
--plugins velero/velero-plugin-for-aws:v1.1.0,$VSPHERE_PLUGIN \
--bucket $BUCKET \
--secret-file $SECRET_FILE \
--snapshot-location-config region=$REGION \
--backup-location-config region=$REGION,s3ForcePathStyle="true",s3Url=$S3URL
velero-vsphere configure --enable-cbt-in-guests
```

If you use this example script, modify the following:

- BUCKET - This variable should be set to the S3 bucket name you configured in HCP for cloud scale.

- REGION - You may choose any region for this variable because HCP for cloud scale will accept any region name provided.

- S3URL - If you are using the 60-day trial of HCP for cloud scale, you do not need to change the value shown in the script. Otherwise, set this URL to the S3 endpoint URL of your HCP for cloud scale system.

- SECRET_FILE - Set this to the local path of the file you created previously with your HCP for cloud scale secret and access keys.

Run the `kubectl vsphere logout` command, and then log in directly to your Supervisor cluster using the administrator@vsphere.local account. When logged in, run your Velero installation script.

When Velero is installed, you should have two pods running within your Velero vSphere Namespace. Run the `kubectl get pods -n velero` command and wait until both the Velero and backup-driver pods are running successfully. The following figure shows both Velero pods running within the Velero vSphere Namespace.

Reference Architecture Guide

Velero is now enabled and installed on your Supervisor Cluster.

Install Velero into a TKG Guest Cluster

📄 **Note:** Do not delete or modify these objects. If you want to do so, do it through the Velero CLI.

Installation into your Tanzu Kubernetes Grid Services guest cluster is very similar to installation into the Supervisor namespace. You can either create a target namespace called *velero or Velero will create it for you during installation*, but this time it will be in the context of the TKG guest cluster to which you are installing Velero

Run the `kubectl vsphere logout` command, log in again including your `--tanzu-kubernetes-cluster-namespace` and `--tanzu-kubernetes-cluster-name` flags as you did when testing the creation of the initial vSphere Namespace, and then switch contexts to your Tanzu Kubernetes Grid Services guest cluster. Run the `kubectl create ns velero` command to create the Kubernetes namespace into which you will install Velero.

Create another script for the TKG guest cluster, and modify the same values that you did for the Supervisor cluster installation (you can use a different bucket/region for the TKG guest cluster as long as the new bucket is created in HCP for cloud scale).

Note the use of *velero install* as opposed to *velero-vsphere install* as the main binary called. An example script for installing into a TKG guest cluster is shown in the following figure.

The following is the text of the *yaml* file:

```
#!/bin/sh
NAMESPACE="velero"
BUCKET="onprem-vcf-velero-target"
REGION="us-west-1"
S3URL="https://tryhcpforcloudscale.hitachivantara.com/"
VSPHERE_PLUGIN=vsphereveleroplugin/velero-plugin-for-vsphere:1.1.0
SECRET_FILE=/root/velero-hcpcs-credentials
velero install \
--namespace $NAMESPACE \
--provider "aws" \
--plugins velero/velero-plugin-for-aws:v1.1.0,$VSPHERE_PLUGIN \
--bucket $BUCKET \
--secret-file $SECRET_FILE \
--snapshot-location-config region=$REGION \
--backup-location-config region=$REGION,s3ForcePathStyle="true",s3Url=$S3URL
```

When you have created your script and modified the necessary information, run it from the current context you used when creating the *velero* Kubernetes namespace within the Tanzu Kubernetes Grid Services guest cluster. An example of the expected output is shown in the following figure.

Deploy Velero Data Manager virtual machines

When Velero is deployed in a vanilla Kubernetes environment, a data mover pod is deployed that assists in moving the contents of any PVs being backed up or restored. In a Tanzu Kubernetes Grid Service deployment, a virtual machine called the Velero Data Manager must be used due to networking limitations in a vSphere environment.

The following are some specific requirements for deploying the Velero Data Manager:

- The Data Manager appliances should be deployed co-resident on the vSphere ESXi cluster hosting the Supervisor cluster where Velero is used.

- A vmkernel interface on each vSphere ESXi host should have the vSphereBackupNFC service enabled on that vmk interface.

- The Data Manager appliances should be connected to the same network as the vSphere ESXi host vmkernel interfaces that have the vSphereBackupNFC service enabled.

- The network used for backup should have a route to both the network that vCenter resides on as well as the Workload Management Ingress network hosting the TKG Control Plane.

- The Velero Data Manager must have Internet access to pull from Docker hub.

Best practice is to create a dedicated backup network for the Velero Data Manager traffic. For more information on creating and enabling the dedicated backup network, see *Setting up Backup Network* in Data manager configuration for vSphere with Tanzu in the documentation available from VMware.

📄 **Note:** If you cannot create a dedicated VLAN-backed backup network, you can configure the management vmkernel NICs on each host to have the vSphereBackupNFC service enabled, and then connect the Velero Data Manager appliances to the management vDS port group.

When the backup network requirements are met, you can deploy a Velero Data Manager OVA and configure it. You can download version 1.1 of the Velero Data Manager OVA from Data manager configuration for vSphere with Tanzu. Using the vSphere Web Client, deploy the OVA into the target vSphere ESXi cluster hosting the Supervisor cluster where Velero is installed. Connect it to the vDS port group that has ESXi vmk interfaces connected and enabled for the *vSphereBackupNFC* service.

Do not power on the resultant VM that gets deployed yet.

When the OVA is deployed, edit the settings of the VM, and then navigate to VM Options > Advanced > Configuration Parameters and click Edit Configuration. You must modify the following values before powering on the VM:

- guestinfo.cnsdp.vcUser - set this value to administrator@vsphere.local.

- guestinfo.cnsdp.vcAddress - set this value to the FQDN of the vCenter hosting the vSphere ESXi cluster configured with the Supervisor cluster where Velero is installed.

- guestinfo.cnsdp.vcPasswd - set this value to the administrator@vsphere.local password.

- guestinfo.cnsdp.wcpControlPlaneIP - set this value to the IP address of the Kubernetes API service on your Supervisor cluster.

These are the minimum values that you must modify before powering on the Velero Data Manager virtual machine. For more information about these values and the other configuration options available for the Velero Data Manager, see *Data Manager Virtual Machine Install* in Data manager configuration for vSphere with Tanzu in the documentation available from VMware.

Close the VM settings after modifying the *guestinfo* parameters, power on the virtual machine, and then open a web console to the VM from the vSphere Web Client. Log in to the VM through the console using the username/password combination of *root/changeme*. You will be prompted to change the password upon first login; set the password to one that meets your organizational security requirements.

The default networking stack for the Velero Data Manager is set for DHCP. To statically assign an IP address on the backup network and for DNS resolution to work (necessary in a VCF environment), do the following:

1. Edit the file */etc/systemd/network/99-dhcp-en.network* and change DHCP=yes to DHCP=no, and save the file.

2. Create a new file /etc/systemd/network/10-static-en.network and populate it with the following, using the correct static IP/netmask/gateway and DNS server for your environment:

```
[Match]
Name=eth0

[Network]
Address=10.0.0.47/24
```

```
Gateway=10.0.0.221
DNS=10.0.0.221
```

3. Run the `chmod 644 /etc/systemd/network/10-static-en.network` command to set the proper permissions on the new network configuration file.

4. Reboot the virtual machine by running the `reboot now` command.

5. When the virtual machine has rebooted, log in to the VM as *root* and issue the `systemctl status velero-datamgr.service` command. You should see the text *code=exited, status=0/SUCCESS* in the output, which means that the docker image for Data Manager has been pulled and started, and that registration with the target vSphere Namespace has succeeded. Do not be concerned that the service is not running, because this is a bootstrap method for the docker container performing the actual data moves. The following figure highlights the expected output of this command.

1. Run the `docker logs velero-datamgr` command to view the logs of the docker container running the actual Velero Data Manager components. Ensure that there are no errors in the last lines of the log output. You should see several success messages for initializing the various components of the DataMover within the Velero Data Manager.

Velero Data Managers only handle single uploads of PV data at a time, unless multiple Velero Data Managers are installed and registered. You can deploy additional Velero Data Manager appliances to achieve concurrent upload of PVs to S3 storage during a multi-PV backup. However, restore of PVs using the Velero Plugin for vSphere is restricted to serial restoration of PVs at this time.

Register guest clusters and configure data protection targets in Tanzu Mission Control

After your Supervisor cluster is registered in Tanzu Mission Control (TMC) as a Management cluster, you can automatically manage any Tanzu Kubernetes Grid Services guest clusters running within vSphere Namespaces on them. You can also configure a common HCP for cloud scale target for Velero backups that various cluster groups in TMC can use for data protection.

Register TKG guest clusters to Tanzu Mission Control

Procedure

1. Open a web browser and navigate to your TMC portal.

2. Select **Administration** from the left pane, navigate to the **Management clusters** tab, and then click on your Management cluster.

3. From the Summary view of your Management cluster, select the **Workload Clusters** tab.

   Note that your TKG guest cluster deployed from kubectl is shown but listed as **No** in the **Managed** column.

4. Select the checkbox next to it, and click **Manage 1 Cluster**.

5. Select the cluster group where you want to place the guest cluster, and then click **Manage**.
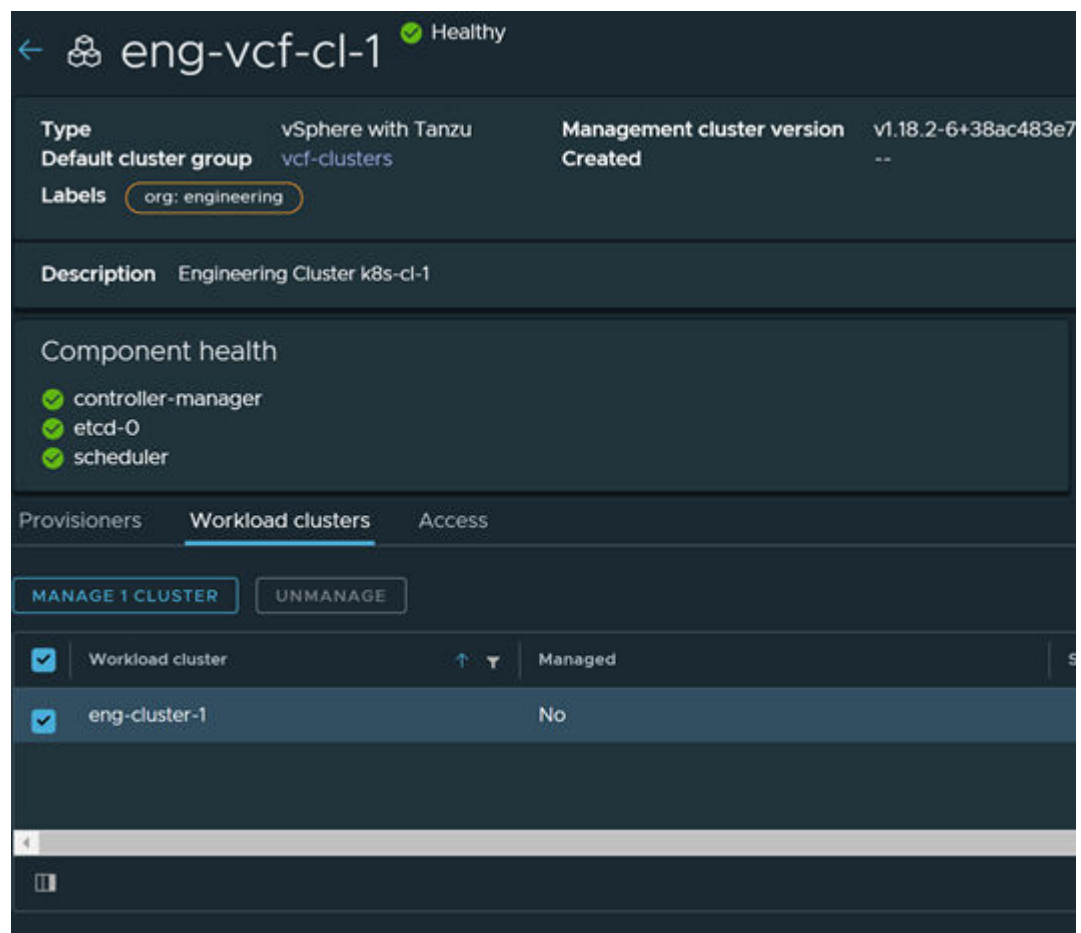
6. Click **Cluster Groups** in the left pane, and then click on the cluster group to which you just added your guest cluster.

   You should see your guest cluster listed and can click on it to view its summary page.
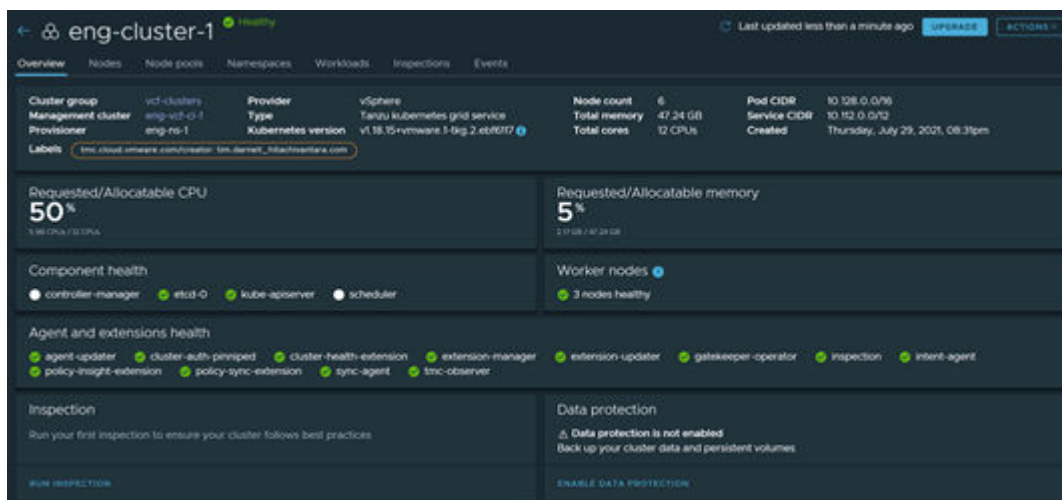
Register non-Tanzu Kubernetes clusters to Tanzu Mission Control

You can register any CNCF-compliant Kubernetes cluster to TMC.

Procedure

1. In the left pane, select **Clusters**, and then click **Attach Cluster**.

2. Enter a name for your cluster, select a cluster group for it, optionally enter a description and labels, and then click **Next**.

3. Copy the `kubectl create` command on the next page of the workflow, and then run it on your target Kubernetes cluster. The following figure shows example output from running the `kubectl create` command on a CNF-compliant Kubernetes cluster.

4. Click **Verify Connection** within TMC until you see a success message: *Attaching the CNCF-compliant Kubernetes cluster is complete*.

5. If you are registering a RedHat OpenShift cluster into TMC, replace the `kubectl` command with the oc command when logged in to your OpenShift cluster from the CLI.

6. Click **Verify Connection** within TMC until you see a success message: *Attaching the RedHat OpenShift cluster is complete*.

    You can view the details of your attached clusters as you would a Tanzu cluster resource.

7. Click on **Clusters** in the left pane, and the CNCF-compliant clusters that you have attached are displayed. The following figure shows a summary view of a RedHat OpenShift cluster attached to TMC.
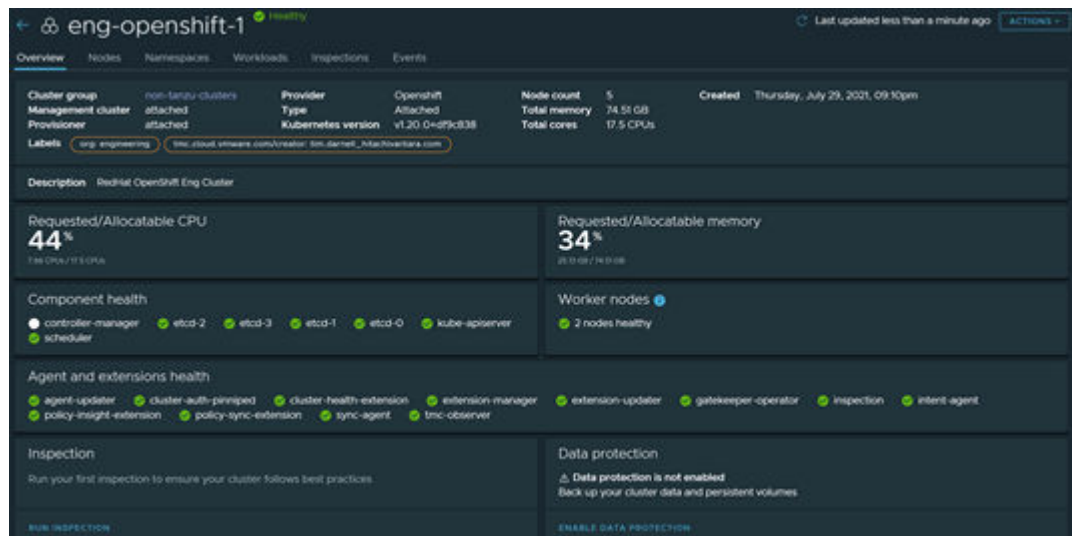
Configure S3 data protection target credentials

Configure TMC to use your trial or fully licensed edition of HCP for cloud scale as an S3 backup target for any attached or managed clusters in TMC once Data Protection has been enabled. A prerequisite for enabling data protection on a cluster is configuring the S3 target credentials and location in TMC.

Procedure

1. Click **Administration** in the left pane, and then select the **Accounts** tab.

2. From the **Create Account Credential** list, select **Create Customer Provisioned Storage Credential S3**. Enter a name for your credential set.

3. Enter your access and secret key information, and then click **Create**.

Configure an S3 data protection target

After your HCP for cloud scale credentials are configured, add the actual S3 target endpoint details as a target location in TMC.

Procedure

1. Click **Administration** in the left pane, and then navigate to the **Target locations** tab.

2. From the **Create Target Location** list, select **Customer provisioned S3-compatible storage**. Select the account credentials that you just created, and then click **Next**.

3. Enter the S3 endpoint URL for your HCP for cloud scale instance, the bucket name created for TMC-based Velero backups, a region to use in HCP for cloud scale, and then click **Next**.

4. Select the cluster groups that will be configured to have this target location configured as their data protection target, and then click **Next**.

5. Enter a name for your target location, and then click **Create**.

Solution Validation

If you have followed the guidance in the Solution Design section, your infrastructure is prepared and you can try these example deployments. This reference architecture was validated by:

▪ Deploying the Wordpress application with persistent volumes using Helm and proving data protection functionality in VCF 4.2 with Velero and Hitachi Content Platform for cloud scale.

▪ Deploying a Tanzu Kubernetes Grid Services guest cluster using Tanzu Mission Control.

▪ Proving data protection functionality in TMC for attached and managed clusters with Velero and Hitachi Content Platform for cloud scale.

On-premises persistent volume and data protection validation

Use the following procedures to:

- Deploy a Wordpress application with persistent volumes backed by vSAN and Hitachi VSP storage systems.

- Explore the vSphere CNS layer.

- Modify the Wordpress application, back it up to HCP for cloud scale, delete it, and then restore the application from HCP for cloud scale using Velero vSphere Operator and verify the data.

The following figure shows the solution architecture with the solution to be validated.
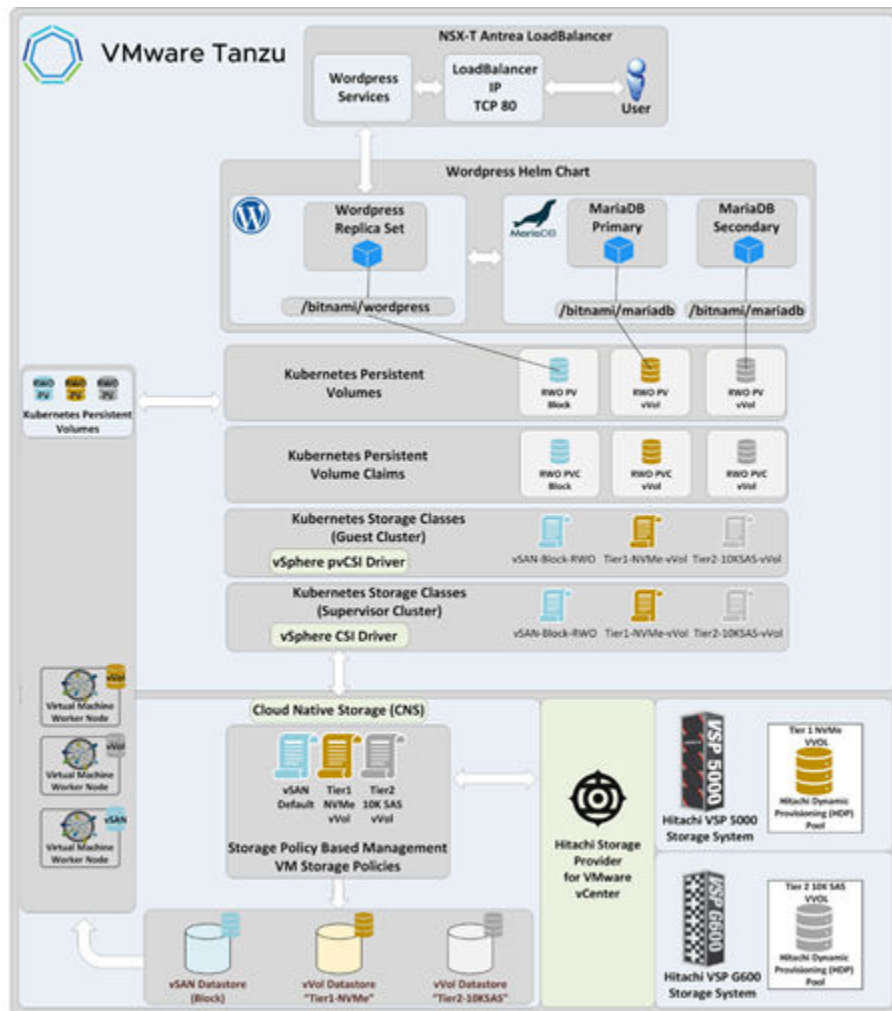
Install and configure Helm utilities

Helm allows you to install complex container-based applications easily with the ability to customize the deployment to your needs. On your Linux workstation, install the Helm binary by following the Helm documentation for your distribution.
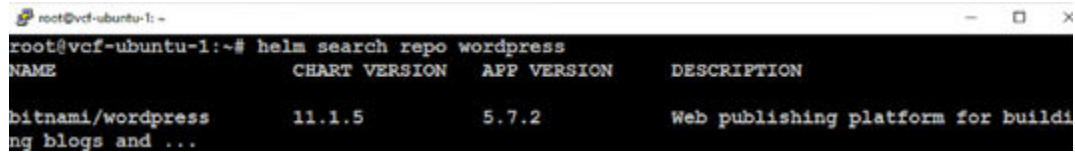
Add the *Bitnami* repository to your Helm configuration by running the following command:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

Search for the Wordpress Helm chart by running the following command:

```
helm search repo wordpress
```

The following figure shows example output, showing that the Helm binary is installed properly and the *bitnami* repository has been added with a Wordpress Helm chart available for use.

Modify the pod security policy on Tanzu Kubernetes Grid Services guest cluster

By default, TKG guest clusters have a restrictive pod security policy. You must create a *ClusterRoleBinding* policy to allow authenticated users to deploy workloads into the guest cluster. Run the following command to allow applications to be deployed by authenticated users:

```
kubectl create clusterrolebinding default-tkg-admin-privileged-binding --
clusterrole=psp:vmware-system-privileged --group=system:authenticated
```

For more information about pod security policies in Tanzu Kubernetes Grid Services, see the VMware documentation.

Verify StorageClasses in Tanzu Kubernetes Grid Services guest cluster

Verify that your kubectl context is set to your TKG guest cluster, and enter the command kubectl get sc. You should see the vSphere SPBM policies that you have authorized for use in the vSphere Namespace which you created and which the TKG guest cluster is running in.

For this example, we used the following three StorageClasses:

▪ One for the frontend Wordpress pod

▪ One for the primary MariaDB instance

▪ One for the secondary MariaDB instance

The following figure shows an example listing of StorageClasses available on a TKG guest cluster.

Customize and deploy Wordpress Helm chart with persistent storage

You can customize a Helm char deployment by downloading the chart values to a YAML file and using that file during Helm chart installation. You can also specify the custom values for a deployment on the command line or in a script.

The following figure shows an example script used in this deployment.

The following is the text of the yaml file:

```
#!/bin/sh
kubectl create ns test-app
helm install -n test-app wordpress \
--set wordpressUsername=admin \
--set wordpressPassword=wordpress \
--set replicaCount=1 \
--set persistence.storageClass=vsan-default-storage-policy \
```

Reference Architecture Guide

```
--set persistence.size=200Mi \
--set mariadb.architecture=replication \
--set mariadb.primary.persistence.storageClass=hitachi-vvol-tier1-iops \
--set mariadb.primary.persistence.size=256Mi \
--set mariadb.secondary.persistence.storageClass=hitachi-vvol-tier2-iops \
--set mariadb.secondary.persistence.size=256Mi \
bitnami/wordpress
```

You must modify these values to match your environment and the StorageClasses that are available in your TKG guest cluster. The values and their corresponding impact to the Wordpress deployment are as follows:

- wordpressUsername - Sets the admin username for the Wordpress application.

- wordpressPassword - Sets the password for the admin user in the Wordpress application.

- replicaCount - Configures the number of frontend Wordpress pods.

- persistence.storageClass - Sets the StorageClass to be used for the frontend Wordpress pods.

- persistence.size - Sets the size of the persistent volume to be assigned to the frontend Wordpress pods.

- mariadb.architecture - Indicates whether Helm should deploy a single backend database (standalone, single pod) or a high-availability backend database (replication, two pods).

- mariadb.primary.persistence.storageClass - Sets the StorageClass to be used for the primary MariaDB instance.

- mariadb.primary.persistence.size - Sets the size of the persistent volume to be assigned to the primary MariaDB instance.

- mariadb.secondary.persistence.storageClass - Sets the StorageClass to be used for the secondary MariaDB instance.

Run the script and Helm will begin to deploy your Wordpress deployment to your TKG guest cluster.

The following figure shows output from Helm at the beginning of the deployment after the `install` command has been issued.

```
root@vcf-ubuntu-1: ~                                                    -  □  ×
NAME: wordpress
LAST DEPLOYED: Fri Jul 30 23:37:53 2021
NAMESPACE: test-app
STATUS: deployed
REVISION: 1
NOTES:
** Please be patient while the chart is being deployed **

Your WordPress site can be accessed through the following DNS name from within your cluste
r:

   wordpress.test-app.svc.cluster.local (port 80)

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

  NOTE: It may take a few minutes for the LoadBalancer IP to be available.
        Watch the status with: 'kubectl get svc --namespace test-app -w wordpress'

   export SERVICE_IP=$(kubectl get svc --namespace test-app wordpress --template "{{ range
(index .status.loadBalancer.ingress 0) }}{{.}}{{ end }}")
   echo "WordPress URL: http://$SERVICE_IP/"
   echo "WordPress Admin URL: http://$SERVICE_IP/admin"

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

  echo Username: admin
  echo Password: $(kubectl get secret --namespace test-app wordpress -o jsonpath="{.data.w
ordpress-password}" | base64 --decode)
```

You can monitor the Wordpress deployment by viewing the resources in your *test-app* Kubernetes namespace. Run the `kubectl get all -n test-app` command to display the readiness of the pods, deployment, replicasets, and statefulsets of the Wordpress deployment.

The following figure shows an example of the output of this command for a fully running, healthy Wordpress deployment.

```
root@vcf-ubuntu-1: ~                                                    -  □  ×
root@vcf-ubuntu-1:~# kubectl get all -n test-app
NAME                                    READY    STATUS    RESTARTS    AGE
pod/wordpress-6bcc4744dc-q22jz          1/1      Running   0           2m58s
pod/wordpress-mariadb-primary-0         1/1      Running   0           2m58s
pod/wordpress-mariadb-secondary-0       1/1      Running   0           2m58s

NAME                                    TYPE          CLUSTER-IP      EXTERNAL-IP    PORT(S
)                         AGE
service/wordpress                       LoadBalancer  10.127.32.255   10.50.0.2      80:326
25/TCP,443:31518/TCP     2m58s
service/wordpress-mariadb-primary       ClusterIP     10.117.249.206  <none>         3306/T
CP                       2m58s
service/wordpress-mariadb-secondary     ClusterIP     10.114.13.64    <none>         3306/T
CP                       2m58s

NAME                          READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/wordpress     1/1      1             1            2m58s

NAME                                    DESIRED    CURRENT    READY    AGE
replicaset.apps/wordpress-6bcc4744dc    1          1          1        2m58s

NAME                                            READY    AGE
statefulset.apps/wordpress-mariadb-primary      1/1      2m58s
statefulset.apps/wordpress-mariadb-secondary    1/1      2m58s
```

## Verify the Wordpress application deployment

In the previous step you displayed the resources in the *test-app* namespace, which included the output of the services exposed for the Wordpress application. Find the *EXTERNAL-IP* for the service/wordpress in the test-app namespace.

From a machine that can access your TKG Ingress network, open a browser and enter the IP address of the Wordpress service (http, not https). The Wordpress application should display.

The following figure shows an example of the deployed Wordpress application accessed from a web browser using the Ingress IP address.
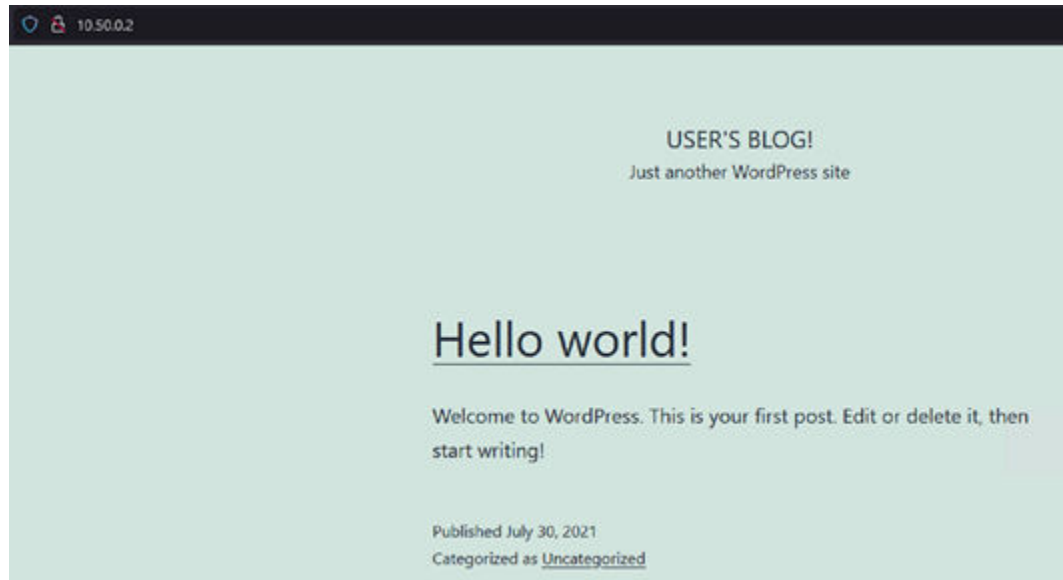
1. Browse to the admin interface of Wordpress (/admin), and log in using the username and password you set in the Helm installation script.

2. Click the Create your first post link.

3. Enter information to create a blog post, and then publish the post.

4. Navigate back to the root IP address of the Wordpress application to verify that your post was committed to the database.
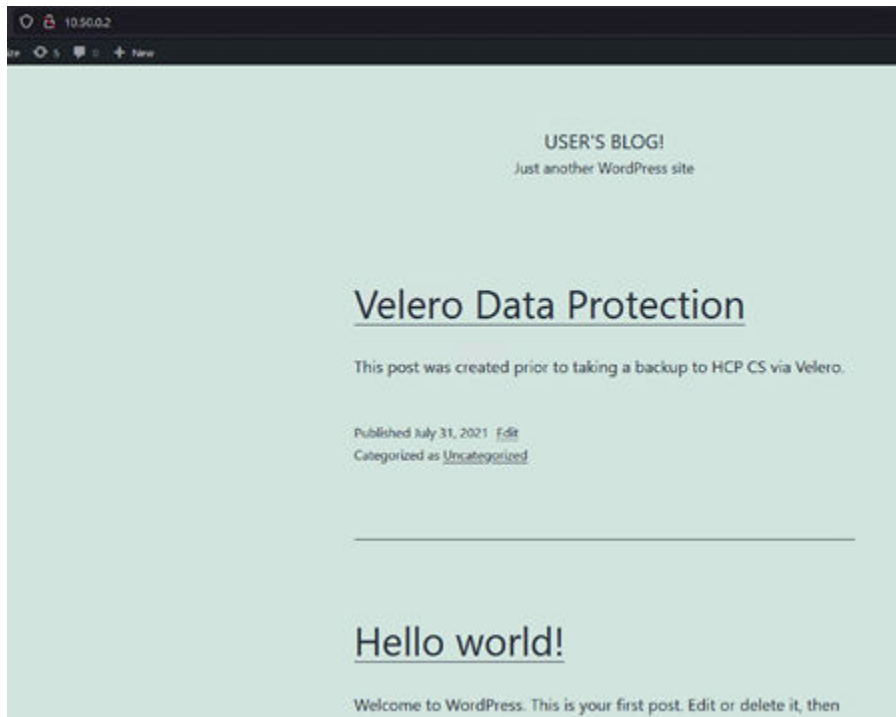
Explore backend storage resource mapping and allocations

When modifying the Helm chart values for the Wordpress deployment, you provided three different StorageClasses that map back to vSphere SPBM policies for persistent volume allocation to the Wordpress and MariaDB pods.

Using the VMware vSphere Cloud Native Storage and First Class Disk (FCD) features, you can follow the storage paths from the Kubernetes persistent volume layer to the vSphere vSAN/vVol layer. Complete the following procedures to validate the storage path from your running pods to the allocated backend storage.

Verify the vSAN ReadWriteOnce persistent volume data path

Starting at the Kubernetes layer, you can explore the PVC and corresponding PVs that were provisioned by the vSphere pvCSI driver.

Procedure

1. To list the PVCs created during the Wordpress Helm chart deployment, run the following `kubectl` command:

```
kubectl get pvc -n test-app
```

2. Get details on the vSAN RWO PVC by running the following `kubectl` command:

```
kubectl describe pvc wordpress -n test-app
```

The following figure shows the output of this command, including the details of the persistent volume claim and the access mode that was specified during Helm chart deployment.

3. Note the volume identifier and copy it for the next step.
4. Now that you have viewed the details of the PVC for the Wordpress frontend, explore the associated PV to the claim by running the following `kubectl` command, entering your Volume identifier from the previous step as the PV ID:

```
kubectl describe pv <PV ID>
```

The following figure shows the output of this command, including the details of the PV created for the PVC. Note the VolumeHandle and copy it for the next step.

You need both the PVC ID and the VolumeHandle ID for the vSAN RWO volume from previous steps in order to observe details about the volume within vCenter.

5. Open a browser and open a vSphere web client session to the vCenter hosting your Supervisor cluster, and then log in.

6. Highlight the vSphere cluster hosting the Supervisor cluster and navigate to its **Monitor** tab.

7. In the left pane expand **Cloud Native Storage**, and then click **Container Volumes**.

   You will see container volumes provisioned to your cluster in the right pane.

8. Find the volume that matches your PVC ID from the previous step, and then click on the **Details** icon.

   This displays the details about the volume that are surfaced from Kubernetes, including your TKG guest cluster ID, the persistent volume ID, namespace, labels, and pod allocation from within Kubernetes.
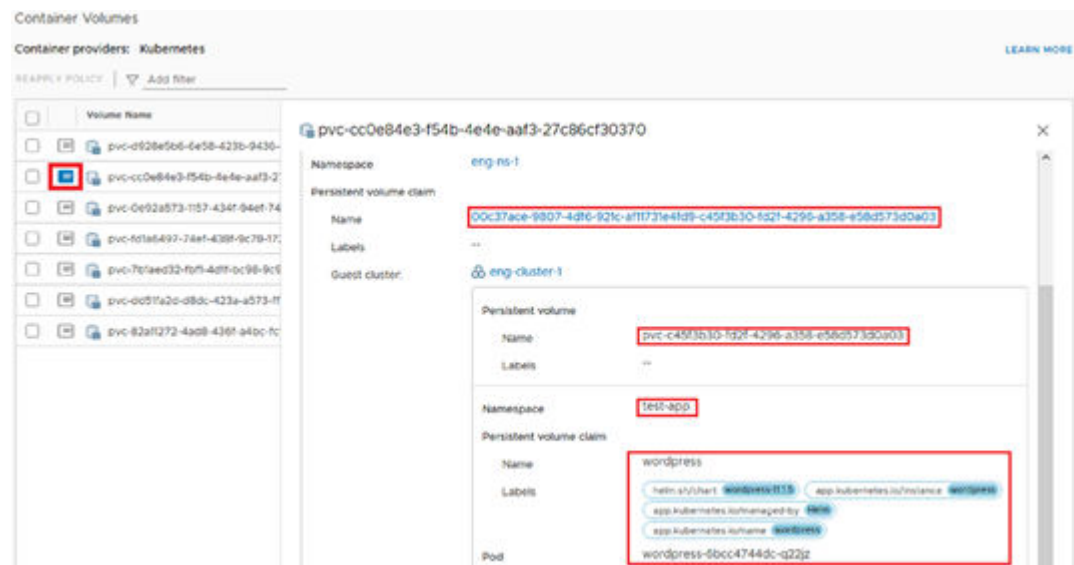
Verifying the primary MariaDB ReadWriteOnce persistent volume data path

When you observed the data path for the vSAN RWO PV, you ran `kubectl` commands to observe details about the associated PVC and PV at the Kubernetes layer.

Procedure

1. Run the same commands against the *data-wordpress-mariadb-primary-0* PVC in the test-app namespace.

   The following figure shows example output of the `kubectl describe pvc data-wordpress-mariadb-primary-0 -n test-app` command.

The following figure shows example output of the `kubectl describe pv <PV ID>` command for the PV details associated to the data-wordpress-mariadb-primary-0 PVC.

2. Note the VolumeHandle along with the PV name for correlation to the vSphere objects.

3. Open a browser and open a vSphere web client session to the vCenter hosting your Supervisor cluster, and then log in.

4. Highlight the vSphere cluster hosting the Supervisor cluster, and then navigate to its **Monitor** tab.

5. Expand **Cloud Native Storage**, and then click **Container Volumes**.

   You will see container volumes provisioned to your cluster in the right pane.

6. Find the volume that matches your PVC ID from the previous step, and then click **Details**.

   This displays the details about the volume that are surfaced from Kubernetes, including your TKG guest cluster ID, the persistent volume ID, namespace, labels, and pod allocation from within Kubernetes.
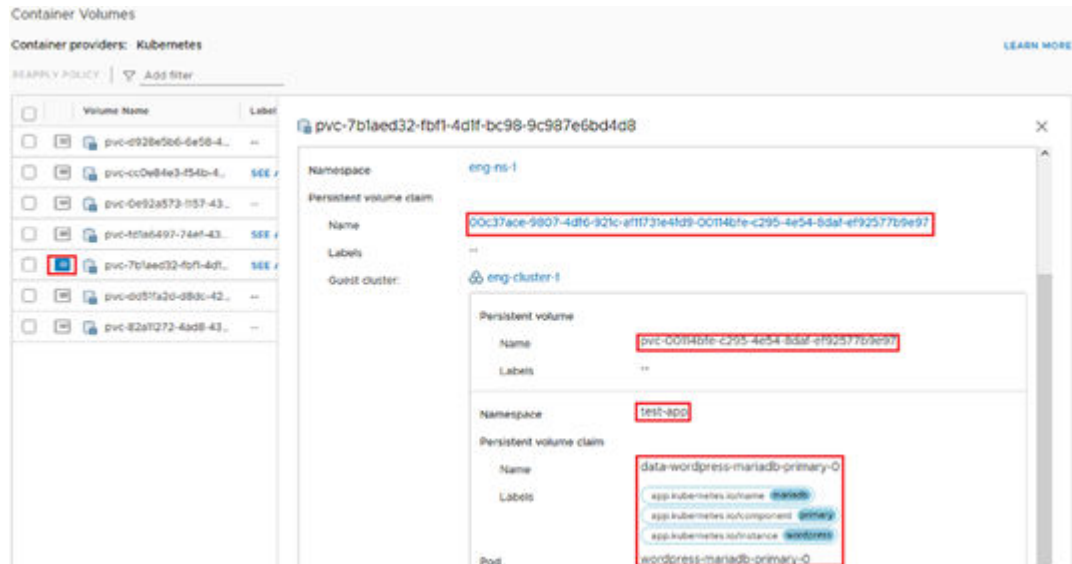
You can perform the same verification of the *data-wordpress-mariadb-secondary-0* PVC and associated PV.

> 📄 **Note:** Newer versions of vCenter and CNS-CSI (7.0 Update 2) provide information on the Volume Backing ID which provides the actual LDEV path to the Hitachi VSP, so you can trace the full data path to the VSP storage system.

Back up the Wordpress application using the built-in Velero operator

Procedure

1. From your Linux workstation, log in to your TKG guest cluster context.
2. Issue the following command to back up the Wordpress application.

```
velero backup create wordpress-backup  --selector app.kubernetes.io/
instance=wordpress  --snapshot-volumes
```

3. Take note of the usage of the label selector used.
4. When the Velero backup is initiated, you can run the `velero backup describe wordpress-backup` command to show the progress of the Velero backup.

   Even if the backup shows that it is completed, if there are persistent volumes that were backed up, you need to verify that the uploads for those volumes were processed and successful.

```
root@vcf-ubuntu-1:~
root@vcf-ubuntu-1:~# velero backup describe wordpress-backup
Name:         wordpress-backup
Namespace:    velero
Labels:       velero.io/storage-location=default
Annotations:  velero.io/source-cluster-k8s-gitversion=v1.18.15+vmware.1
              velero.io/source-cluster-k8s-major-version=1
              velero.io/source-cluster-k8s-minor-version=18

Phase:  Completed

Errors:    0
Warnings:  0

Namespaces:
  Included:   *
  Excluded:   <none>

Resources:
  Included:          *
  Excluded:          <none>
  Cluster-scoped:  auto

Label selector:  app.kubernetes.io/instance=wordpress

Storage Location:  default

Velero-Native Snapshot PVs:  true

TTL:  720h0m0s

Hooks:  <none>

Backup Format Version:  1.1.0

Started:    2021-07-31 02:04:55 +0000 UTC
Completed:  2021-07-31 02:06:04 +0000 UTC

Expiration:  2021-08-30 02:04:55 +0000 UTC

Total items to be backed up:  26
Items backed up:              26

Velero-Native Snapshots: <none included>
```

5. Switch your `kubectl` config context to your Supervisor cluster.

Because the Velero Data Manager components are registered to the *velero* namespace in the Supervisor cluster, the associated uploads for all PV backups (including from TKG guest clusters) are created in the Supervisor cluster *velero* namespace.

6. Issue the `kubectl get upload -n velero` command to list the uploads generated from the Velero backup you just created.

```
root@vcf-ubuntu-1:~
root@vcf-ubuntu-1:~# kubectl config use-context 10.50.0.1
Switched to context "10.50.0.1".
root@vcf-ubuntu-1:~# kubectl get uploads -n velero
NAME                                          AGE
upload-8efa19de-c4e5-4655-a257-bbf6b9cbb2ea   11m
upload-ad02057a-a20c-4ee0-bbe8-302fc8ce04e8   10m
upload-deec9db1-dd26-4e1d-9bd2-f54a81456716   11m
```

7. Run the `kubectl describe upload <Upload ID> -n velero` command, where the upload ID is the name of one of the uploads listed from the previous command output.

If you see *Upload completed*, the PV and its data have been successfully backed up to HCP for cloud scale as part of the Velero backup. The status progression that you should see for each upload is **New** > **In Progress** > **Completed**.

8. Verify the other two uploads for the Wordpress backup before you continue.

When all the uploads show a status message of **Completed**, the Velero backup of Wordpress is complete.

Explore the Hitachi Content Platform for cloud scale S3 object store

Procedure

1. Open a web browser, navigate to your Hitachi Content Platform for cloud scale web interface, and log in.

2. Browse the contents of your bucket, locate the *wordpress-backup* folder, and open it.

You will see the contents of all of the Kubernetes objects that were backed up during the Velero backup, with the exception of the PVs and their data.
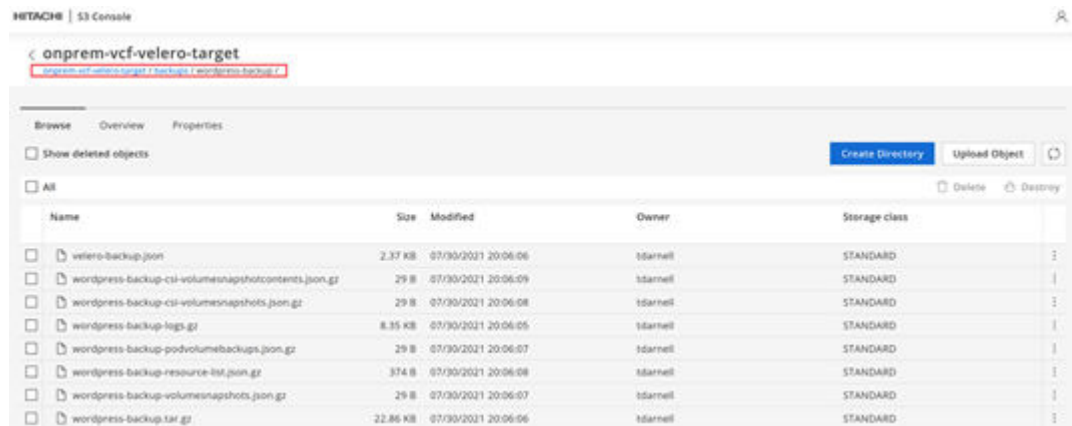
3. Navigate back to the root of your bucket and select the plugins folder.

4. Navigate to *vsphere-astrolabe-repo/ivd/data*.

5. Select an *ivd* record, and observe the different size objects in each folder.

   These correspond to the PVs that were backed up from the vSphere snapshots taken by the vSphere Plugin for Velero and uploaded to HCP for cloud scale by the Velero Data Manager appliances running on the ESXi cluster hosting the Supervisor cluster.

   > **Note:** Users should not delete or modify these objects. If they want to do so, they need to do it through the Velero CLI.
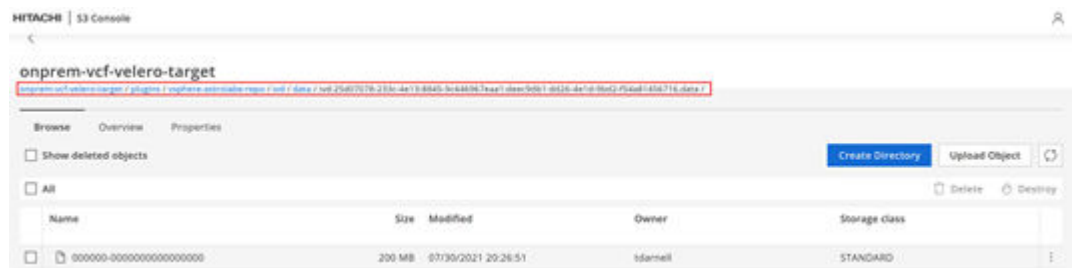
Delete the Wordpress application from TKG Guest Cluster

Procedure

1. From your Linux workstation, log in to your TKG guest cluster context.

2. Run the `helm delete wordpress -n test-app` command to remove the Wordpress application.

3. When the application is removed, run the `kubectl delete ns test-app` command to remove all of the resources including PVCs and PVs from the TKG guest cluster.

Restore the Wordpress application using the built-in Velero operator

Procedure

1. Run the `velero backup get` command to list the backups in the Velero database.

   You can see the backups of the Wordpress application that you took previously.

2. Run the `velero restore create wordpress-restore --from-backup wordpress-backup --restore-volumes` command to begin the restore of the Wordpress application and its associated PVs and data.

3. After the Velero restore is submitted, run the `velero restore describe wordpress-restore` command to view the progress of the restore. Note that the restore will not show as completed until the application and all of its PVs and associated data are restored.

   Similar to viewing the uploads during Velero backup, you can view the downloads of the PVs and their data from Hitachi Content Platform for cloud scale through the Supervisor cluster.

4. Switch your `kubectl` config context to your Supervisor cluster, and run the `kubectl get downloads -n velero` command to view the downloads associated with the restore from Hitachi Content Platform for cloud scale.

5. Run the `kubectl describe download <Download ID> -n velero` command, where the download ID is the name of one of the downloads listed from the previous command output.

6. Switch your `kubectl` config context to your TKG guest cluster, and run the `velero restore describe wordpress-restore` command until you see that the status is Completed.

7. Run the `kubectl get all -n test-app` command to view all of the Wordpress components that were restored.
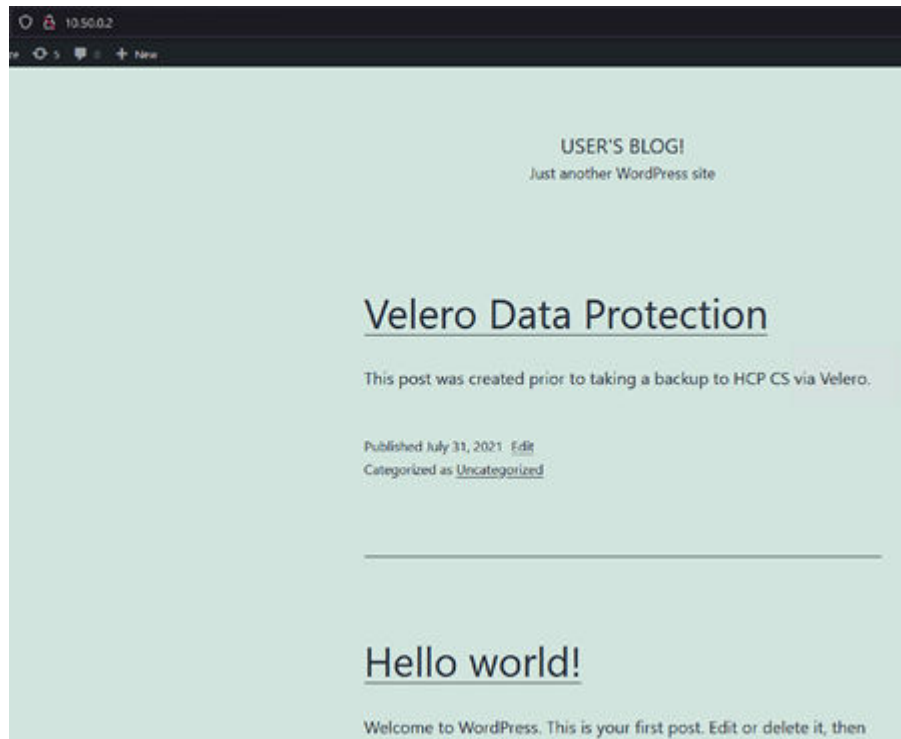

Verify Wordpress application persistent data

Procedure

1. Open a web browser and navigate to the EXTERNAL-IP of the service/wordpress listed in the output of the `kubectl get all` command.

   You will see the blog entry that you created before deleting the Wordpress application.

Tanzu Mission Control validation

Complete these procedures to do the following:

- Deploy a Tanzu Kubernetes Grid Services guest cluster to a registered Management cluster using Tanzu Mission Control.

- Deploy a Wordpress application with persistent volumes backed by vSAN and Hitachi Virtual Storage Platform storage systems.

- Enable data protection on the new TKG cluster through TMC.

- Modify the Wordpress application, back it up to Hitachi Content Platform for cloud scale, delete it, and then restore the application from Hitachi Content Platform for cloud scale using TMC, and then verify the data.

Deploy a TKG guest cluster to a registered supervisor cluster

Procedure

1. Open a web browser to your TMC web portal and log in.
2. Click **Clusters** > **Create Cluster**.
3. Select the Management cluster you want to create the TKG guest cluster on, and then click **Continue to Create Cluster**.
4. Select the vSphere Namespace where you want to create the new TKG guest cluster, and then click **Next**.
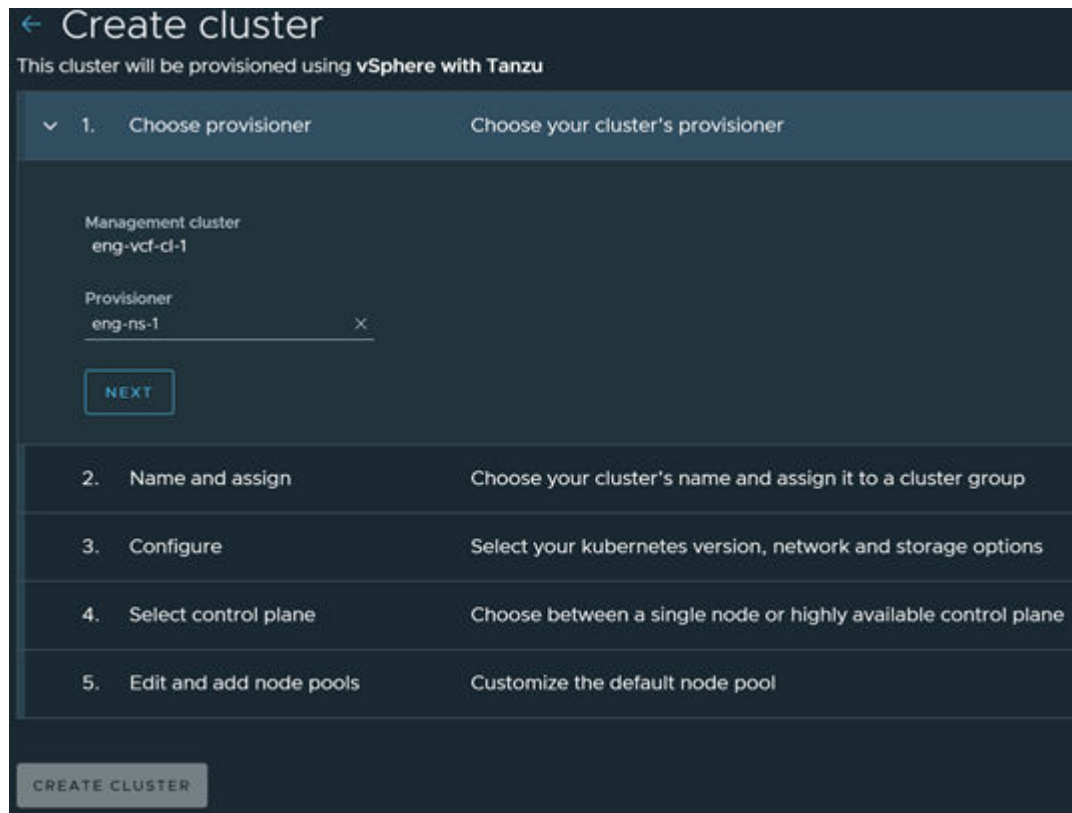
5. Enter a name and cluster group for the new TKG cluster, and optional description and labels, and then click **Next**.

6. Select the most recent version of 1.18 Kubernetes (VCF 4.2 comes with vSphere 7.0 Update 1d, which supports up to Kubernetes 1.18).

   You can leave the Pod and Service CIDR ranges at the default values.

7. From the **Allowed storage classes** list, select the SPBM policies you want to expose to the TKG guest cluster, and then click **Add Storage Class**.

8. Repeat this for each policy you want exposed.

9. Select a default storage class and click **Next**.
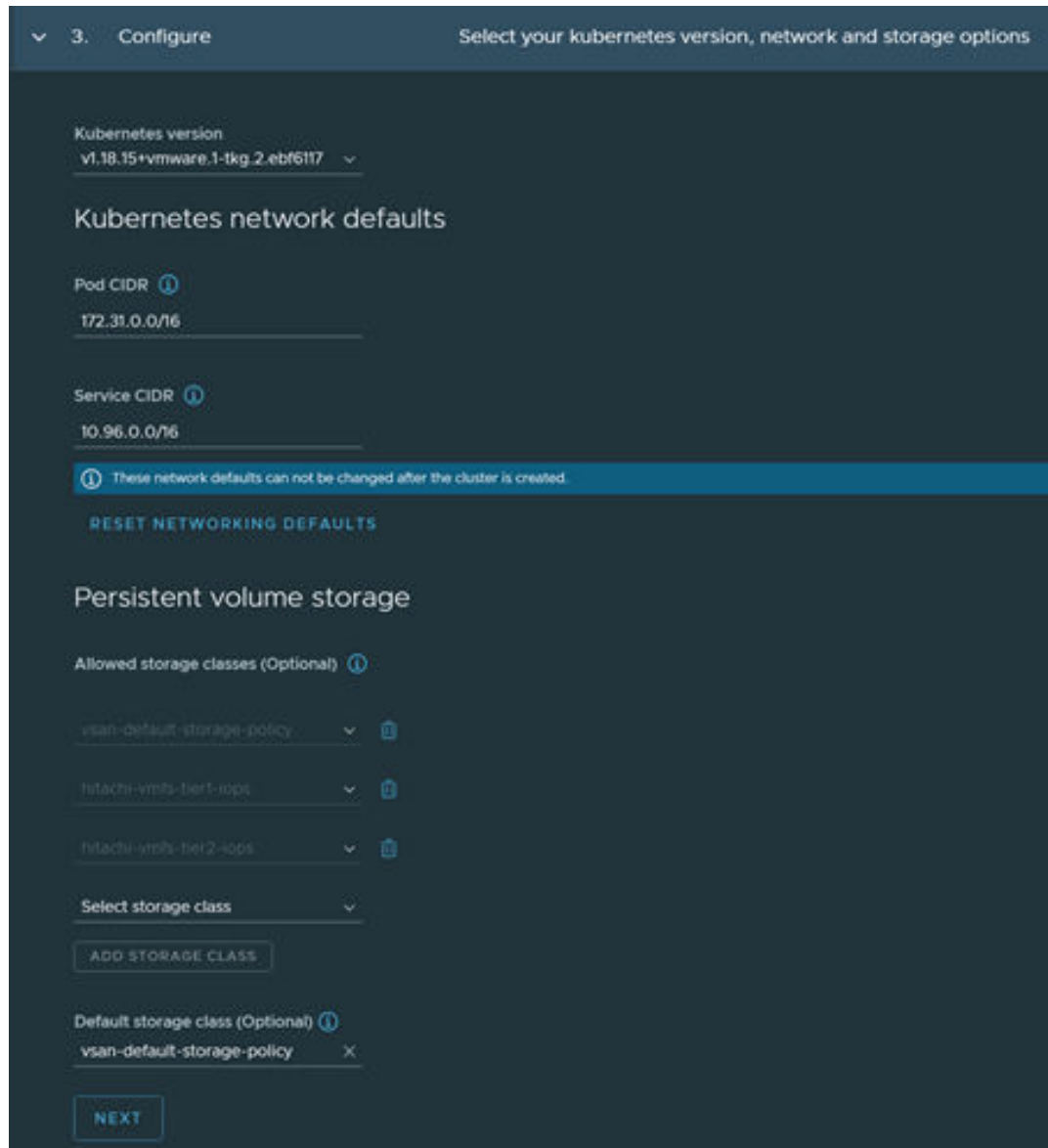
10. Select a highly available control plane, and then select an instance type for the node VMs that will be deployed as part of the TKG guest cluster.

11. Select a StorageClass for the node VMs to use, and then click **Next**.
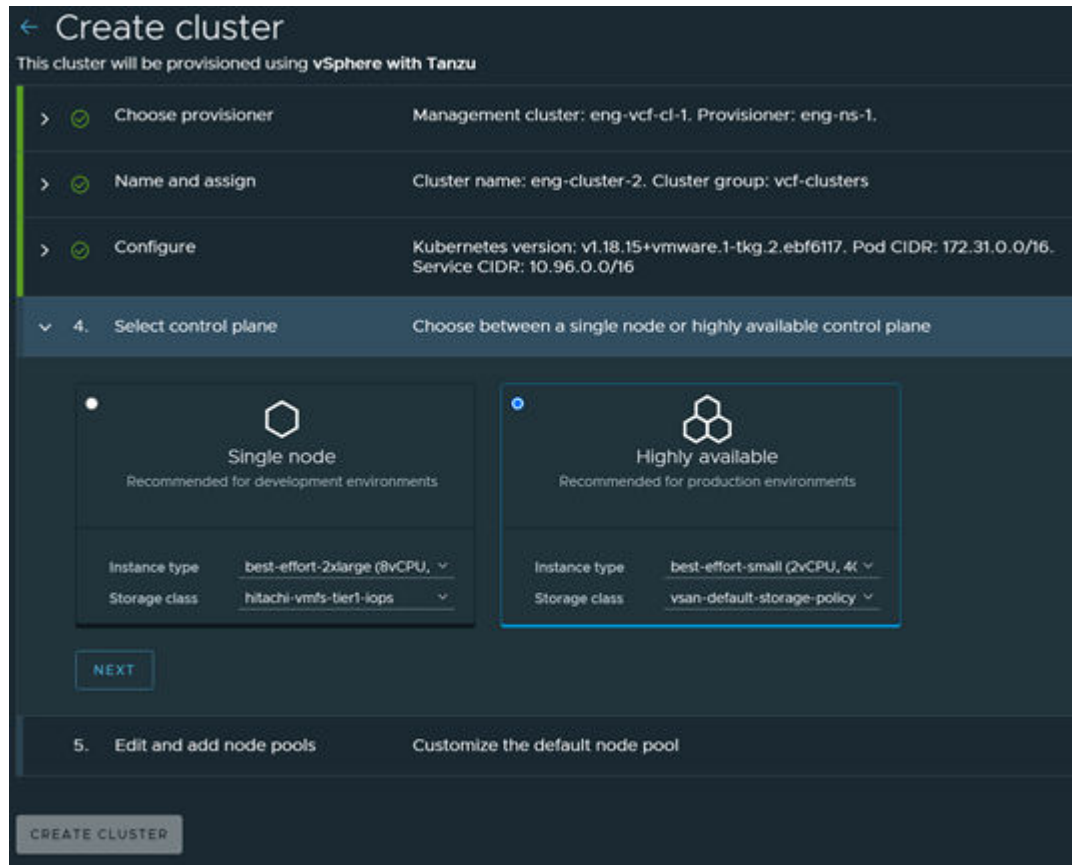
12. Select the worker instance type for your worker pool, the StorageClass for the worker node VMs, the number of worker nodes in the new TKG guest cluster, and then click **Create Cluster**.
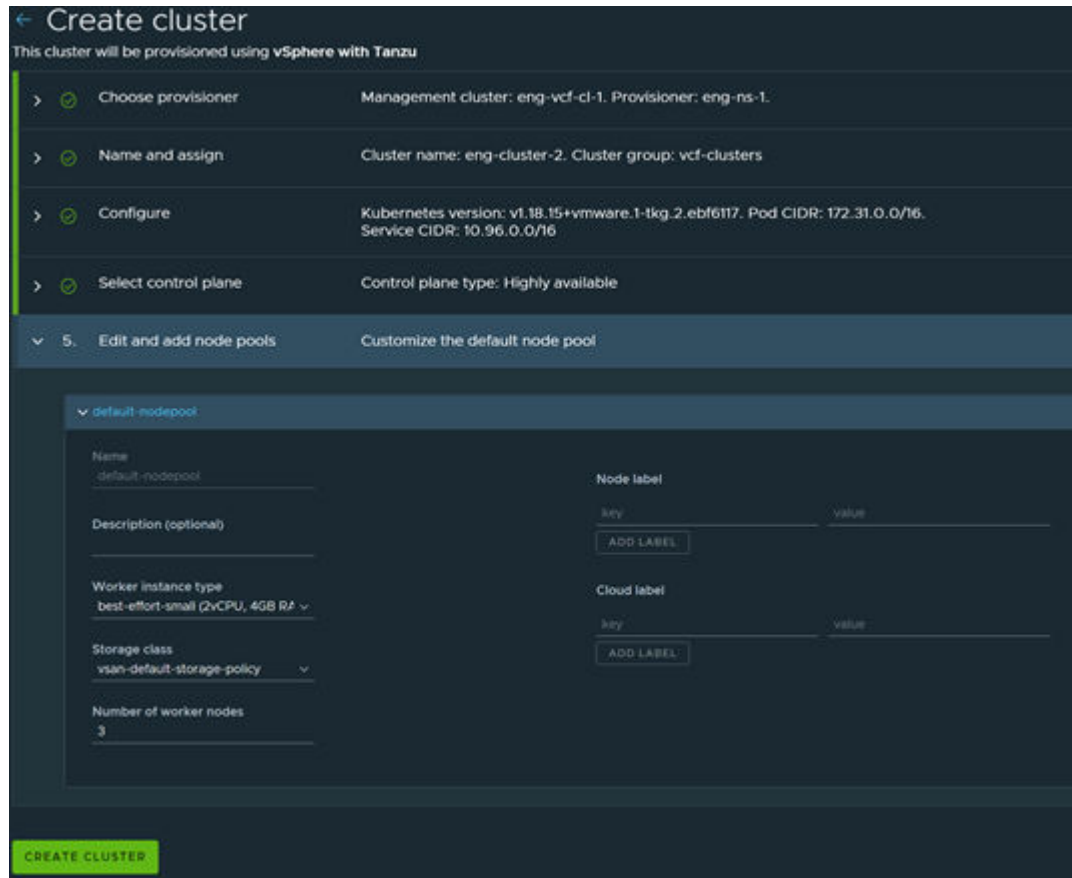
13. Wait for the new TKG guest cluster to deploy and show healthy status.
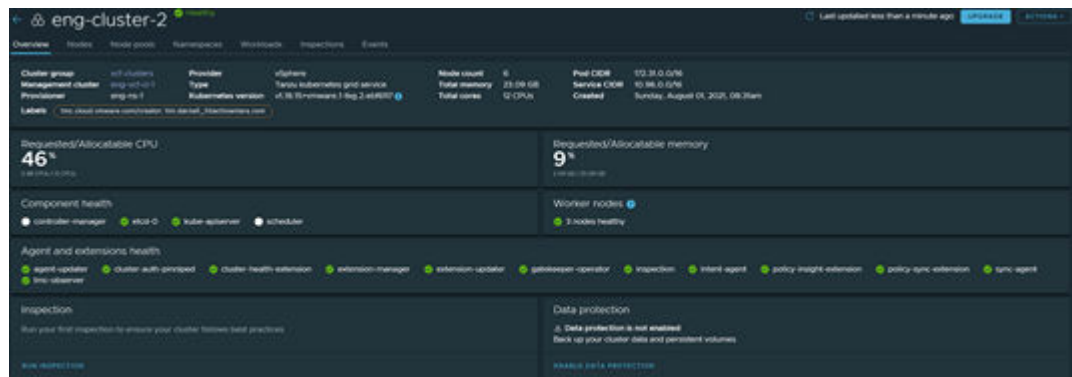
Modify the pod security policy on Tanzu Kubernetes Grid Services guest clusters

By default, TKG guest clusters have a restrictive pod security policy. You must create a *ClusterRoleBinding* policy to allow authenticated users to deploy workloads to the guest cluster. Run the following command to allow applications to be deployed by authenticated users:

```
kubectl create clusterrolebinding default-tkg-admin-privileged-binding --
clusterrole=psp:vmware-system-privileged --group=system:authenticated
```

See Using Pod Security Policies with Tanzu Kubernetes Clusters from VMware for more information about pod security policies in Tanzu Kubernetes Grid Services.

Customize and deploy a Wordpress Helm chart with persistent storage

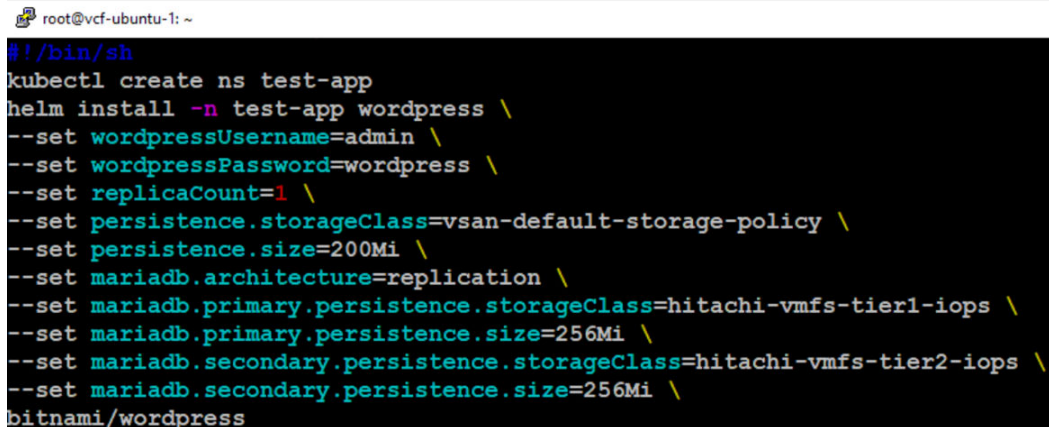Similar to the previous Wordpress Helm chart deployment, create a script to deploy Wordpress and configure the necessary options. Ensure that you are using StorageClasses that you selected in TMC when creating the TKG guest cluster.

The following figure shows an example script, referencing StorageClasses that were configured in TMC as part of the TKG guest cluster creation.

The following is the text of the yaml file:

```
#!/bin/sh
kubectl create ns test-app
helm install -n test-app wordpress \
--set wordpressUsername=admin \
--set wordpressPassword=wordpress \
--set replicaCount=1 \
--set persistence.storageClass=vsan-default-storage-policy \
--set persistence.size=200Mi \
--set mariadb.architecture=replication \
--set mariadb.primary.persistence.storageClass=hitachi-vmfs-tier1-iops \
--set mariadb.primary.persistence.size=256Mi \
--set mariadb.secondary.persistence.storageClass=hitachi-vmfs-tier2-iops \
--set mariadb.secondary.persistence.size=256Mi \
bitnami/wordpress
```

Run the script and Helm starts to deploy your Wordpress deployment to your TKG guest cluster. Monitor the Wordpress deployment by viewing the resources in your test-app Kubernetes namespace. Run the `kubectl get all -n test-app` command to display the readiness of the pods, deployment, replicasets, and statefulsets of the Wordpress deployment.

The following figure shows an example of the output of this command for a fully running, healthy Wordpress deployment.

Verify the Wordpress application deployment

In the previous step you displayed the resources in the *test-app* namespace, which included the output of the services exposed for the Wordpress application. Find the *EXTERNAL-IP* for the service/wordpress in the test-app namespace. From a machine that can access your TKG Ingress network, open a browser and enter the IP address of the Wordpress service (http, not https). The Wordpress application is displayed.

The following figure shows an example of the deployed Wordpress application accessed using a web browser from the Ingress IP address.
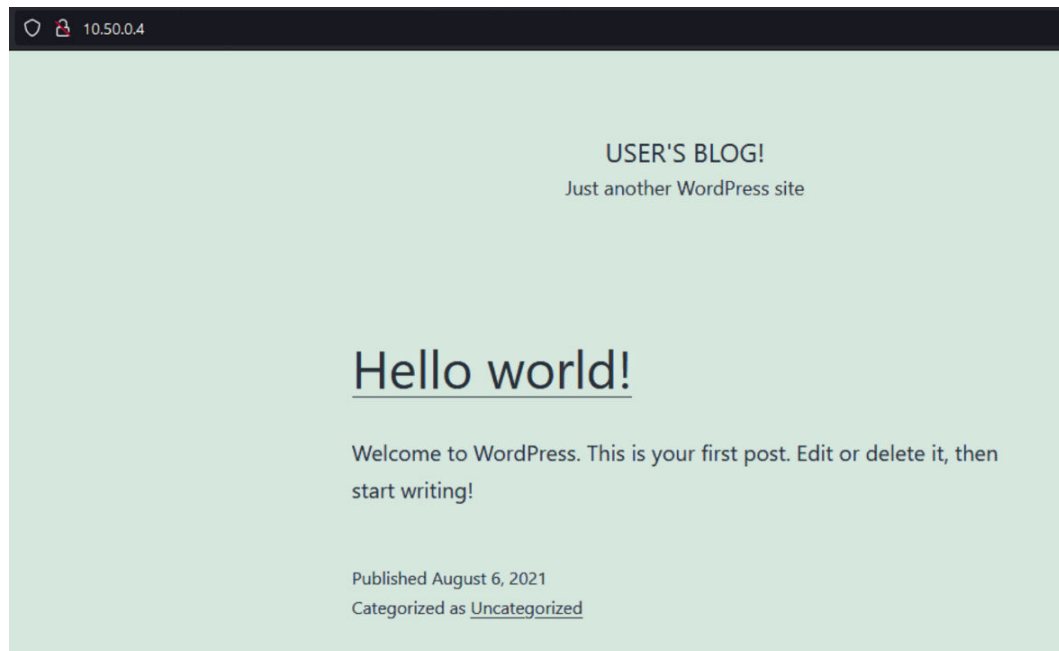
Procedure

1. Browse to the admin interface of Wordpress (/admin), and log in using the username and password you set in the Helm installation script.

2. Click the **Create your first post** link, enter information to create a blog post, and then publish the post.

3. Navigate back to the root IP address of the Wordpress application to verify that your post was committed to the database.

Back up the Wordpress application to HCP for cloud scale using TMC Data Protection

Procedure

1. Open a web browser, navigate to your TMC portal, and log in.

2. Click **Clusters** in the left pane, and then click on the cluster where you installed Wordpress.

   This displays the summary of your TKG guest cluster that was deployed using TMC.

3. Locate the **Data protection** box in the bottom right corner of the Summary view.

4. Click **Enable Data Protection** and acknowledge the notification that Velero will be installed on your cluster.

> 📄 **Note:** Make sure that Velero is not installed or Velero Operator and Velero CRD are not present in guest clusters while enabling the Enable Data Protection option.
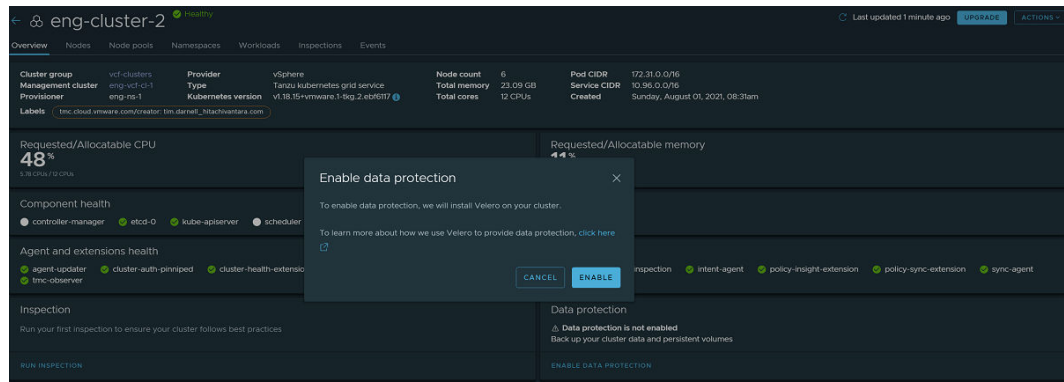
A message displays in the **Data protection** box in TMC stating that *Data protection is being enabled*.

You should annotate pod volumes before performing backups to include volumes in the backups. This approach is called Opt-In and the details and commands to annotate the pods are listed here: https://velero.io/docs/v1.6/restic/#using-opt-in-pod-volume-backup.

5. Refresh the cluster summary view until you see the message *Run a backup to protect your cluster data*, and then click the **Create Backup** link.

6. Select the **Back up selected namespaces** option, select the box *test-app* namespace, and then click **Next**.

7. Select the S3 backup target that you created in earlier steps.
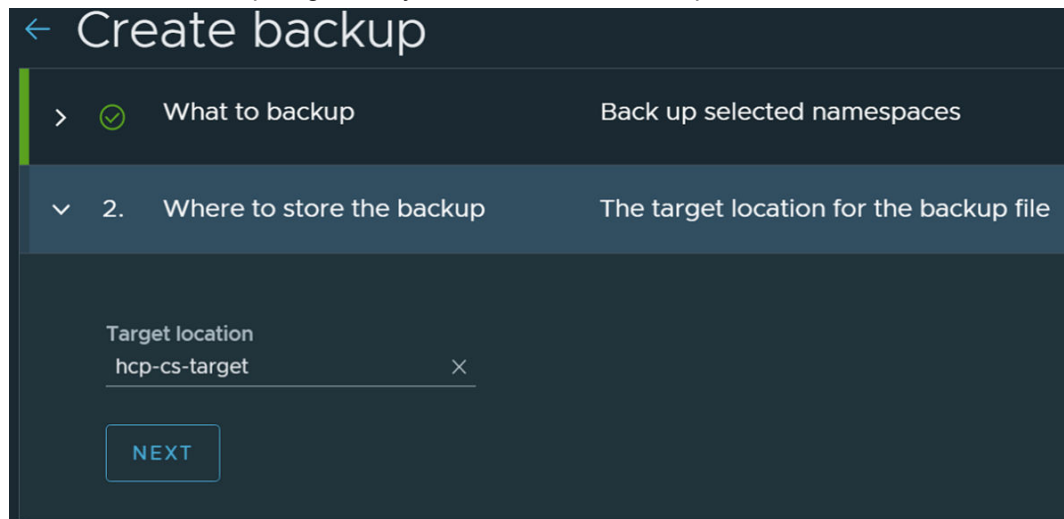
8. Select **Now** for the schedule type, and then click **Next**.

9. Enter the number of days for the retention period for the backup, and then click **Next**.

10. Provide a name for the backup, click **Create**, and then your backup will begin.

When you click **Create**, you are redirected to the **Data protection** tab of your cluster where the backup you created is listed.

11. Refresh the page until you see the backup status listed as **Completed**.

Delete the Wordpress application from TKG guest clusters

Procedure

1. From your Linux workstation, log in to your TKG guest cluster context.

2. Run the `helm delete wordpress -n test-app` command to remove the Wordpress application.

3. When the application is removed, run the `kubectl delete ns test-app` command to remove all the resources including PVCs and PVs from the TKG guest cluster.

Restore the Wordpress application from HCP for cloud scale using TMC Data Protection

Procedure

1. Switch back to your TMC web browser, which should still show the **Data protection** tab of your TKG guest cluster.

2. Select the *eng2-wordpress-backup* listed (or the name of your Wordpress backup), and then click the **Restore** link.

3. Select the **Restore selected namespaces** option, select the *test-app* namespace, and then click **Next**.

4. Enter a name for the restore job, and then click **Restore**.
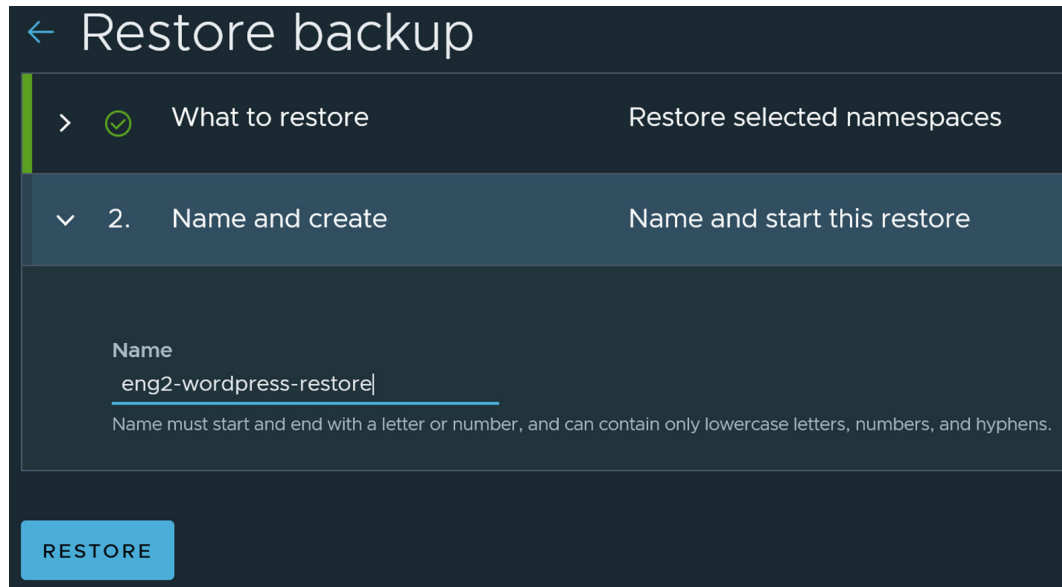
5. From your Linux workstation, run the `kubectl get all -n test-app` command until you see all resources running and in a healthy status.

   The following figure shows the restored Wordpress application resources all running and healthy with deployments, replicasets, and statefulsets in 1/1 Ready status along with all pods in Running status and an external IP assigned to the Wordpress service.

Verify the Wordpress application persistent data

Procedure

1. Open a web browser and navigate to the *EXTERNAL-IP* of the service/Wordpress listed in the output of the previous command.

   You should see the blog entry that you created before deleting the Wordpress application.

Conclusion

VMware Cloud Foundation, VMware Tanzu Kubernetes Grid Services, VMware Tanzu Mission Control, Hitachi Virtual Storage Platform, Hitachi Content Platform for cloud scale, and Hitachi Unified Compute Platform RS combine to create a powerful and flexible Kubernetes ecosystem.

This reference architecture has shown how to provide a simple, real-world example of multiple VMware Cloud Foundation and Tanzu Kubernetes Grid Services features with Hitachi storage integrations and Velero data protection backed by Hitachi Content Platform for cloud scale. You can now provide multiple storage and data protection options to consumers of your container-based virtual infrastructure, using well-known and supported storage integrations between Hitachi Vantara and VMware.

Product descriptions

This is information about the hardware and software components used in this solution for Tanzu Kubernetes Grid Services on Hitachi Unified Compute Platform Rack Scale with VMware Cloud Foundation.

Hitachi Advanced Server DS120

Optimized for performance, high density, and power efficiency in a dual-processor server, Hitachi Advanced Server DS120 delivers a balance of compute and storage capacity. This 1U rack-mounted server has the flexibility to power a wide range of solutions and applications.

The highly-scalable memory supports up to 3 TB using 24 slots of high-speed DDR4 memory. Advanced Server DS120 is powered by the Intel Xeon Scalable processor family for complex and demanding workloads. There are flexible OCP and PCIe I/O expansion card options available. This server supports up to 12 small form factor storage devices with up to 4 NVMe drives.

This solution allows you to have a high CPU-to-storage ratio. This is ideal for balanced and compute-heavy workloads.

Multiple CPU and storage devices are available. Contact your Hitachi Vantara sales representative to get the latest list of options.

Hitachi Advanced Server DS220

With a combination of two Intel Xeon Scalable processors and high storage capacity in a 2U rack-space package, Hitachi Advanced Server DS220 delivers the storage and I/O to meet the needs of converged solutions and high-performance applications in the data center.

The Intel Xeon Scalable processor family is optimized to address the growing demands on today's IT infrastructure. The server provides 24 slots for high-speed DDR4 memory, allowing up to 3 TB of memory per node when 128 GB DIMMs are used. This server supports up to 12 large form factor storage devices and an additional 2 small form factor storage devices.

This server has three storage configuration options:

- 12 large form factor storage devices and an additional 2 small form factor storage devices in the back of the chassis

- 16 SAS or SATA drives, 8 NVMe drives, and an additional 2 small form factor storage devices in the back of the chassis

- 24 SFF devices and an additional 2 SFF storage devices in the back of the chassis

Hitachi Advanced Server DS225

Choose Hitachi Advanced Server DS225 to ensure you have the flexibility and performance you need to support your business-critical enterprise applications.

Advanced Server DS225 delivers compute density and efficiency to meet the needs of your most demanding high-performance applications. It takes full advantage of the Intel Xeon scalable processor family with up to four dual-width 300 W graphic accelerator cards, up to 3 TB memory capacity, and additional PCIe 3.0 expansion slots in a 2U rack space package.

Front-side accessible storage bays supports up to eight hot-pluggable, serial-attached SCSI (SAS) or serial-ATA (SATA) devices. These bays also support flexible configuration, which allows Advanced Server DS225 to deliver high I/O performance and high capacity.

Hitachi Advanced Server DS240

Meet the needs of your most demanding high-performance applications with Hitachi Advanced Server DS240. With up to four Intel Zeon Scalable Processors and up to 6 TB memory capacity in a 2U rack-space package, this server delivers unparalleled compute density and efficiency.

The Advanced Server DS240 architecture takes full advantage of the Intel Xeon Scalable Processor family, including the highest performance options, to address the growing demands of your IT infrastructure.

Hitachi Virtual Storage Platform 5000 series

This enterprise-class, flash array evolution storage, Hitachi Virtual Storage Platform 5000 series (VSP) has an innovative, scale-out design optimized for NVMe and storage class memory. It achieves the following:

- **Agility using NVMe:** Speed, massive scaling with no performance slowdowns, intelligent tiering, and efficiency.

- **Resilience:** Superior application availability and flash resilience. Your data is always available, mitigating business risk.

- **Storage simplified:** Do more with less, integrate AI (artificial intelligence) and ML (machine learning), simplify management, and save money and time with consolidation.

Hitachi Virtual Storage Platform E990

Hitachi Virtual Storage Platform E990 supercharges business application performance with all-NVMe storage. It uses Hitachi Ops Center, so you can improve IT operations with the latest AI and ML capabilities. Advanced data reduction in Virtual Storage Platform E990 enables you to run data reduction with even the most performance hungry applications.

The all-NVMe architecture in Virtual Storage Platform E990 delivers consistent, low-microsecond latency to reduce latency costs for critical applications. This predictable performance optimizes storage resources.

With Virtual Storage Platform E990 and the rest of Hitachi midrange storage family, you have agile and automated data center technology. These systems allow you to cost-effectively meet your current digital expectations and give you the ability to address future challenges, as your application data needs and service levels evolve. With time-tested, proven availability and scalability, Hitachi Vantara delivers infrastructure solutions that help you maximize your data center advantage.

Hitachi Virtual Storage Platform F Series family

Use Hitachi Virtual Storage Platform F series family storage for a flash-powered cloud platform for your mission critical applications. This storage meets demanding performance and uptime business needs. Extremely scalable, its 4.8 million random read IOPS allows you to consolidate more applications for more cost savings.

Hitachi Virtual Storage Platform F series family delivers superior all-flash performance for business-critical applications, with continuous data availability.

Hitachi Virtual Storage Platform G series family

The Hitachi Virtual Storage Platform G series family enables the seamless automation of the data center. It has a broad range of efficiency technologies that deliver maximum value while making ongoing costs more predictable. You can focus on strategic projects and consolidating more workloads while using a wide range of media choices.

The benefits start with Hitachi Storage Virtualization Operating System RF. This includes an all new enhanced software stack that offers up to three times greater performance than our previous midrange models, even as data scales to petabytes

Hitachi Virtual Storage Platform G series offers support for containers to accelerate cloud-native application development. Provision storage in seconds, and provide persistent data availability, all the while being orchestrated by industry leading container platforms. Move these workloads into an enterprise production environment seamlessly, saving money while reducing support and management costs.

Arista Data Center switches

Arista Networks builds software-driven cloud networks for data center, cloud, and campus environments. Arista delivers efficient, reliable and high-performance Universal Cloud Network architectures, based on 10 GbE, 25 GbE, 40 GbE, 50 GbE, and 100 GbE platforms delivered with an extensible operating system - Arista EOS.

- Arista 7050CX3-32S is a 1RU sized spine switch with 32 (downlink) and 4 (uplink) 100 GbE QSFP ports for multiple-rack solutions. Each QSFP port supports a choice of five speeds, with flexible configuration between 100 GbE, 40 GbE, 4 × 10 GbE, 4 × 25 GbE, or 2 × 50 GbE modes.

- Arista 7050SX3-48YC8 is a 1RU sized switch with 48 × 25 GbE SFP and 8 × 100 GbE QSFP ports. The high density SFP ports can be configured in groups of 4 to run either at 25 GbE or a mix of 10 GbE/1 GbE speeds. The QSFP ports allow 100 GbE or 40 GbE high speed network uplinks.

- Arista 7010T is a 1RU sized, 48-port 1 GbE management switch for single-rack and multiple-rack solutions.

Cisco Nexus switches

The Cisco Nexus switch product line provides a series of solutions that make it easier to connect and manage disparate data center resources with software-defined networking (SDN). Leveraging the Cisco Unified Fabric, which unifies storage, data and networking (Ethernet/IP) services, the Nexus switches create an open, programmable network foundation built to support a virtualized data center environment.

Brocade switches from Broadcom

Brocade and Hitachi Vantara have partnered to deliver storage networking and data center solutions. These solutions reduce complexity and cost, as well as enable virtualization and cloud computing to increase business agility.

Brocade Fibre Channel switches deliver industry-leading performance, simplifying scale-out network architectures. Get the high-performance, availability, and ease of management you need for a solid foundation to grow the storage network you want.

Hitachi Storage Virtualization Operating System RF

Hitachi Storage Virtualization Operating System RF powers the Hitachi Virtual Storage Platform (VSP) family. It integrates storage system software to provide system element management and advanced storage system functions. Used across multiple platforms, Storage Virtualization Operating System includes storage virtualization, thin provisioning, storage service level controls, dynamic provisioning, and performance instrumentation.

Flash performance is optimized with a patented flash-aware I/O stack, which accelerates data access. Adaptive inline data reduction increases storage efficiency while enabling a balance of data efficiency and application performance. Industry-leading storage virtualization allows SVOS RF to use third-party all-flash and hybrid arrays as storage capacity, consolidating resources for a higher ROI and providing a high-speed front end to slower, less-predictable arrays.

Hitachi Vantara