TECHNICAL WHITE PAPER:
February 2024

# VMware vSAN™ Features

## Proof of Concept (PoC) Guide

## Table of contents

## Introduction

The vSAN Features guide represents one of a series of vSAN Proof of Concept Guides covering a variety of vSAN related topics. The other guides being:

- vSAN Proof of Concept: vSAN Architecture Overview & Setup
- vSAN Proof of Concept: vSAN Management, Monitoring & Hardware Testing
- vSAN Proof of Concept: vSAN Performances Testing
- vSAN Proof of Concept: vSAN Stretched Cluster & Two-Node Overview & Testing

This guide is designed to stand largely separate from the other documents. That said, the assumption is that the reader has working knowledge of vSAN cluster creation and Storage Policy Management. Especially since the steps documented herein often assume a vSAN Cluster already exists in your test environment. If you require a refresher, please review the vSAN Proof of Concept: vSAN Architecture Overview & Setup guide

The particular focus of this guide is discussion and walkthrough of specific vSAN features such as:

- Space efficiency features (e.g., compression, deduplication, RAID-5/RAID-6 erasure coding, and Trim/Unmap)
- Encryption
- File Services

This document primarily focuses on vSAN Express Storage Architecture™ (ESA) cluster environments. vSAN Original Storage Architecture™ (OSA) environments are covered where they differ from vSAN ESA.

# vSAN Space Efficiency Features

## Overview

Space efficiency technologies in enterprise storage play an important role improving value and decreasing costs. VMware vSAN has several technologies in place to help improve storage efficiency.

Space efficiency techniques can be categorized into the following:

- Opportunistic
  - These space efficiency techniques are dependent on conditions of the data, and not guaranteed to return a predetermined level of savings
  - vSAN offers several types of opportunistic space efficiency features such as Deduplication & Compression (in vSAN OSA), Compression-only, TRIM/UNMAP space reclamation, and thin provisioning
- Deterministic
  - These space efficiency techniques can be relied upon to deliver a guaranteed level of capacity savings
  - vSAN offers deterministic space efficiency capabilities through data placement schemes that are optimized for storing data in a resilient but efficient manner, including RAID-5/6 erasure coding

In vSAN, opportunistic and deterministic space efficiency features can be used independently or together.

For a deeper discussion vSAN space efficiency please refer to:

https://core.vmware.com/resource/vsan-space-efficiency-technologies

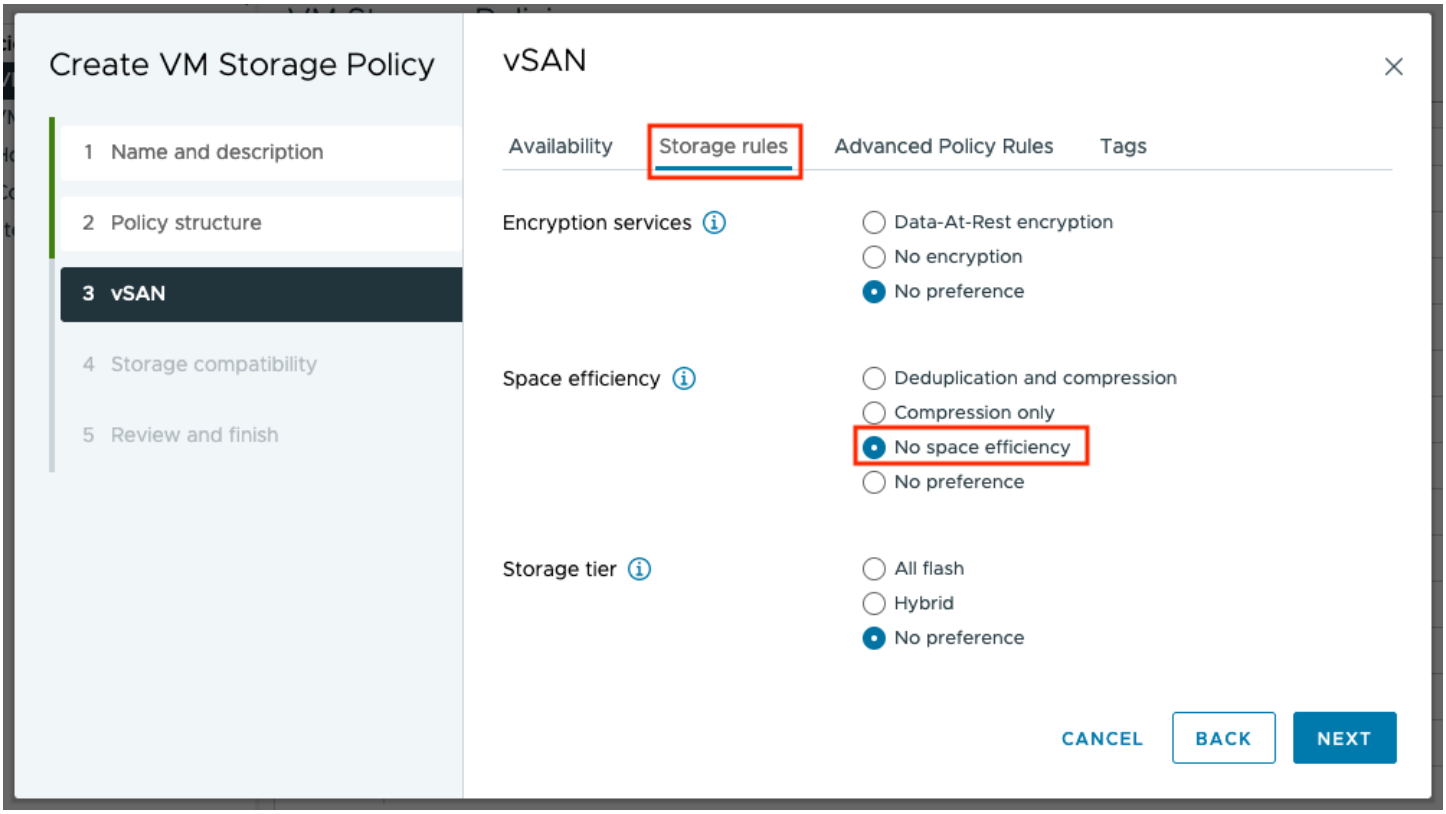
## Compression Only

### vSAN ESA Compression

With vSAN ESA, **compression is defined at a storage policy level**, and is **enabled by default**. The compression mechanisms in vSAN ESA evaluate and compress data differently than in vSAN OSA. In vSAN ESA, each incoming 4KB block is evaluated on a 512 Byte sector size (vSAN OSA, uses larger 2KB sector sizing). The 512 Byte size equates to 8 sectors per 4KB block (8*512 Bytes = 4096 Bytes or 4KB). This allows the 4KB block to reduce in increments of 512 bytes. Depending on how compressible the 4KB block, one can realize up to an 8:1 compression ratio. For example, a 4KB block could be compressed down to 7/8ths its original size is if it is not very compressible, or all the way down to 1/8th its original size, if it is h highly compressible. Ultimately, compressible data in vSAN ESA storage can be compressed at finer levels of granularity.

To ***disable*** compression on an vSAN ESA cluster, navigate to **Home > Policies and Profiles > VM Storage Policies** and create or amend an existing policy.

Under the **storage rules > space efficiency** setting, select '**no space efficiency**':




*Note: This will update new writes only on the vSAN ESA cluster. Existing data will not be affected.*

### vSAN OSA Compression

vSAN OSA supports "Compression Only". Compression is applied directly at the cluster-level and implemented per disk group. The compression algorithm will take a 4K block and try to compress it to 2KB or less (2:1). If this is successful, the compressed block is then written to the capacity tier. If the compression algorithm cannot compress the block by this amount, the full 4KB will be written to the capacity tier. More information on enabling compression in vSAN OSA clusters is in the vSAN OSA cluster deduplication and compression section of this guide.

## Deduplication and Compression

### vSAN ESA Cluster

vSAN ESA does not support deduplication as of the publication of this guide. That said, vSAN ESA supports a variety of space efficiency features:

- Up to 8:1 compression ratio
- Improvements to RAID-5/RAID-6 erasure coding
- Default trim/unmap support

Depending on the overall vSAN ESA design, actual space efficiency may, in fact, exceed a comparable vSAN OSA cluster leveraging deduplication.

vSAN OSA Cluster

In addition to just compressing the data, further savings may be achieved with deduplication in vSAN OSA. When data is destaged from the cache to capacity tier, vSAN will check to see if a match for that block exists. If true, vSAN does not write an additional copy of the block, and metadata is updated. However, if the block does not exist, vSAN will attempt to compress the block.

To demonstrate the effects of Deduplication and Compression, this exercise shows the capacity after deploying a set of identical VMs. Before starting this exercise, ensure that the Deduplication and Compression service is enabled on the cluster. When enabling the Deduplication and Compression service, vSAN will go through a rolling update process: vSAN will evacuate data from each disk group in turn and the disk group will be reconfigured with the features enabled. Depending on the number of disk groups on each host and the amount of data, this can be a lengthy process.

*Note: Administrators have the option of enabling "Compression Only" or "Deduplication and compression" simultaneously. In this example we will select the "Deduplication and compression" option.*

To enable Deduplication and Compression complete the following steps:

Navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **EDIT** button that corresponds to the **Data Services**:

Toggle 'Compression only' or 'Deduplication and Compression' and select **APPLY**:

## Testing Compression

Compression rates, and the associated space savings) are very much dependent on the stored data. Further, data change rates mean that compression ratios are dynamic for a given system. This makes testing compression particularly challenging.

However, we can make reasonable attempts at a repeatable test, given a static, freely available dataset. One such dataset is from the human genome project, hosted by Ensembl (https://www.ensembl.org).

The top-level human genome data consists of a very large text file (consisting of a long string of letters), compressed using gzip. Our test here would be to distribute the data over several VMs on a vSAN datastore and then uncompress the data. Upon enabling compression (and later deduplication and compression for OSA clusters) we can see vSAN compression in action.

### Importing the Dataset

First, create a VM that we can later template. A fast, repeatable method using an Ubuntu image is detailed in Appendix A.

Before the final steps marking the VM as a template and cloning, power on and open an SSH session to the VM. Download the human genome file (around 1GB in size):

```
curl -u anonymous:password 'https://ftp.ensembl.org/pub/release-
108/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna_sm.toplevel.fa.gz' -o dna.fa.gz
```

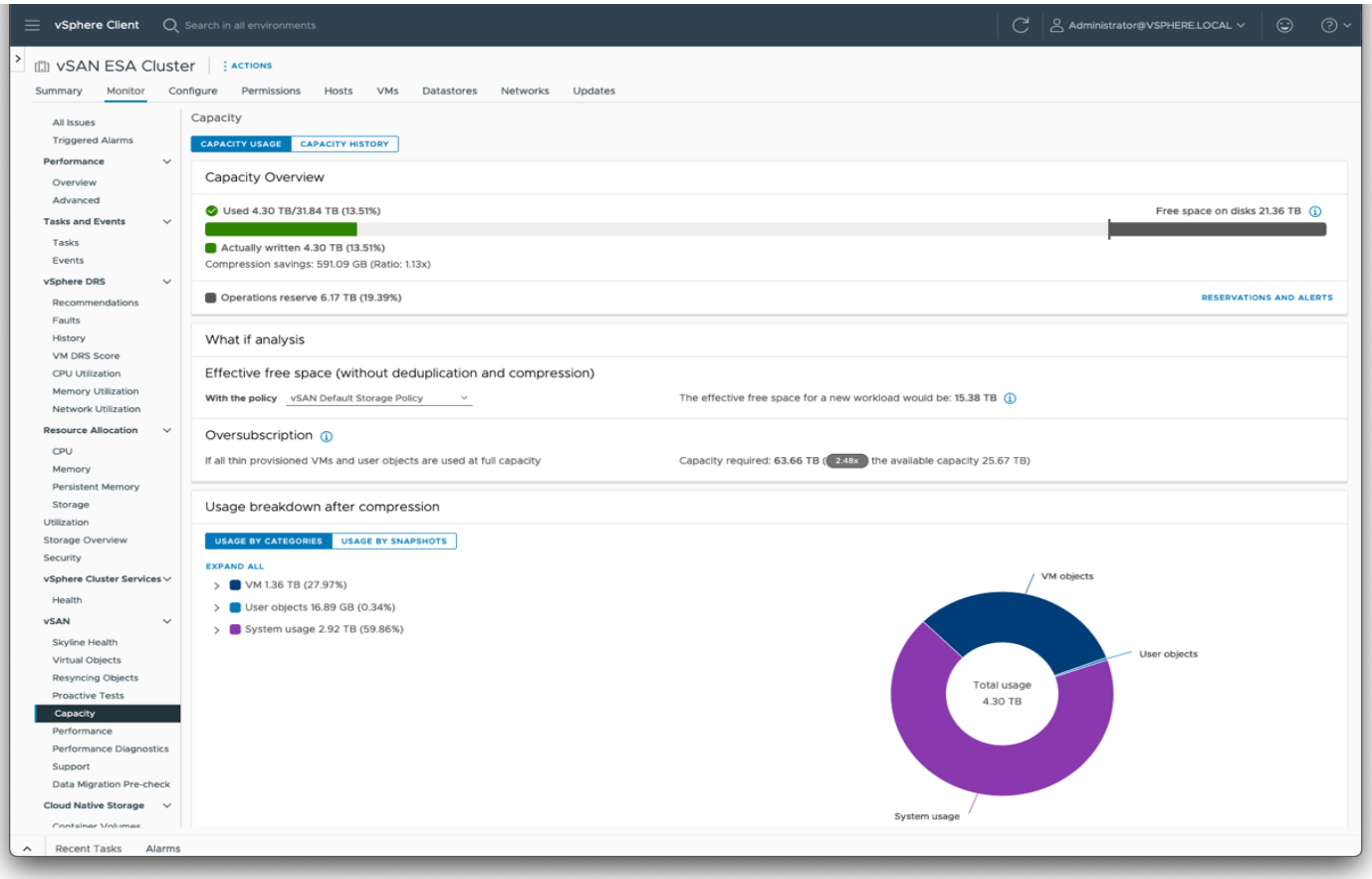
Once this has been downloaded, shutdown the VM and mark as a template. Then clone the VM: the number of clones will, of course, depend on the size of the vSAN datastore.

After cloning, we can inflate the downloaded genome file. The command below uses 'govc' utility, see Appendix A details on downloading the utility. Note, this will take some time:

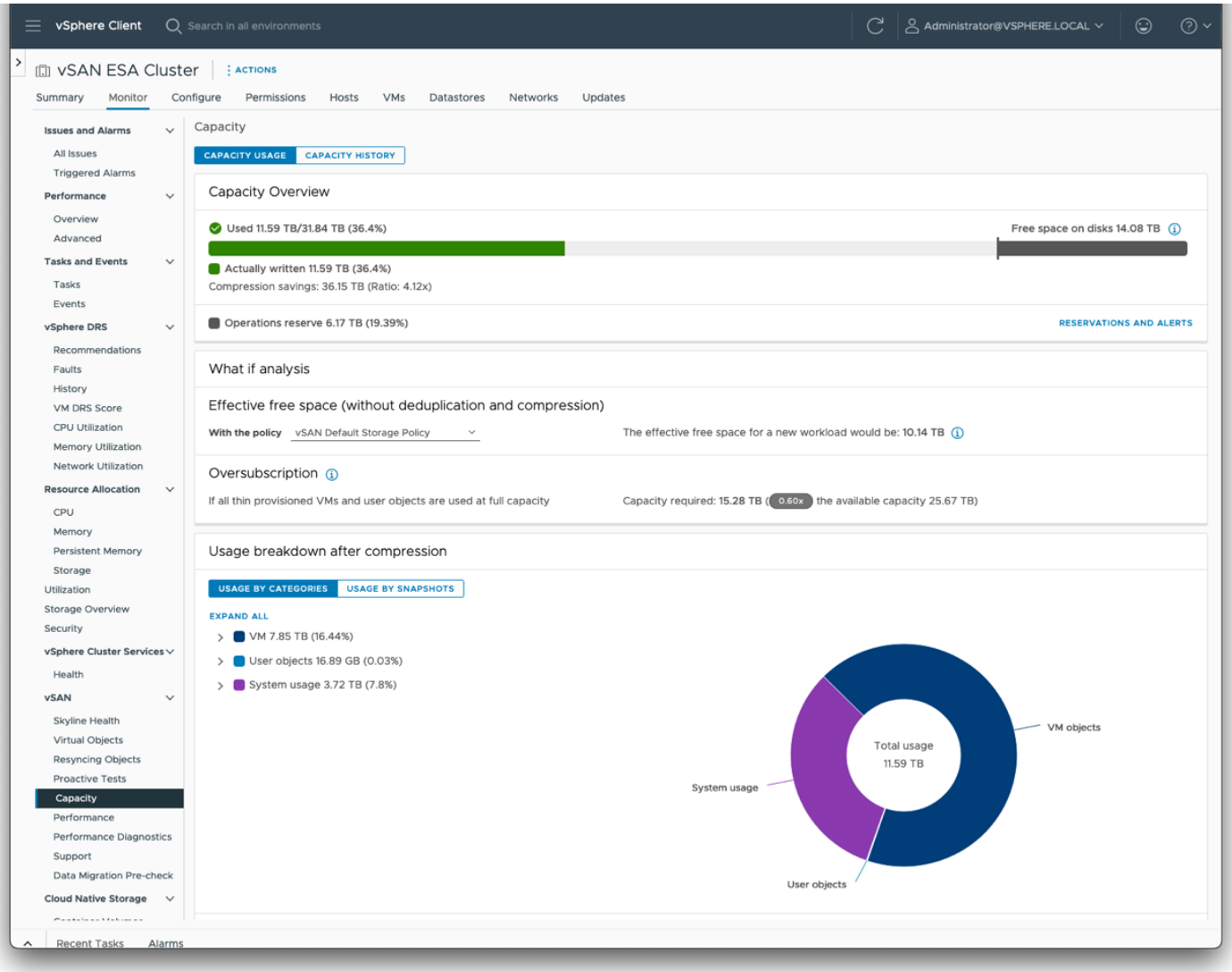
```
govc find -type m -name 'ubuntu-vm*' | xargs -P0 -I '{}' bash -c 'ssh -o "StrictHostKeyChecking=no"
ubuntu@$(govc vm.ip {}) gzip -dv dna.fa.gz'
```

## ESA Cluster

Below is an example on an ESA cluster. Before the decompression phase (i.e. when the VMs have just been cloned), we have around 4.3TB used, which has been compressed to a ratio of around 1.13x (remember that for ESA compression is enabled by default):

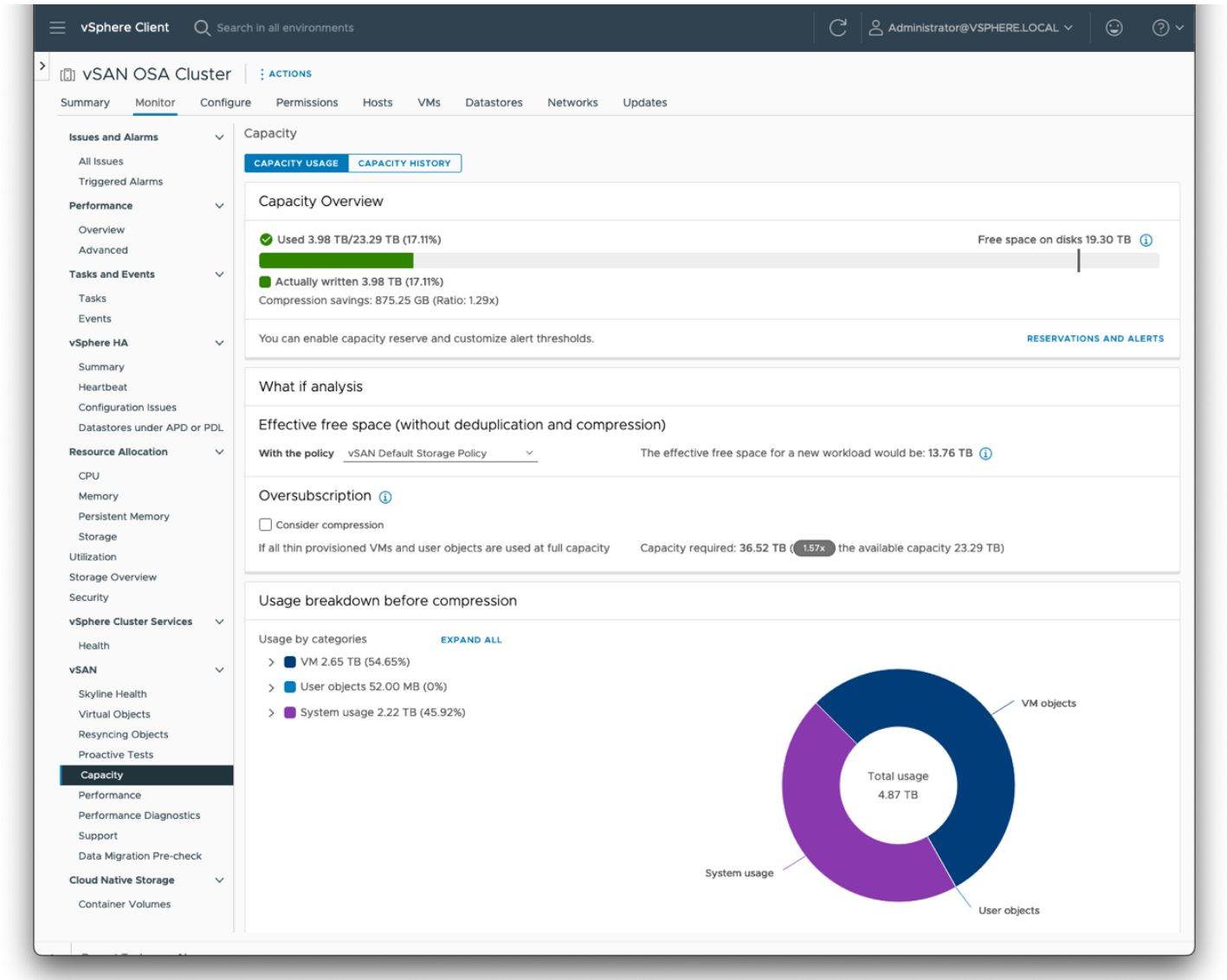



After decompressing the dataset on all the VMs, we now have around 11.6TB used, with a compression ratio of around 4x (the consumption here with vSAN compression is greater than with gzip of course. This is balanced with the advantage of accessing and manipulating the data with greater performance):

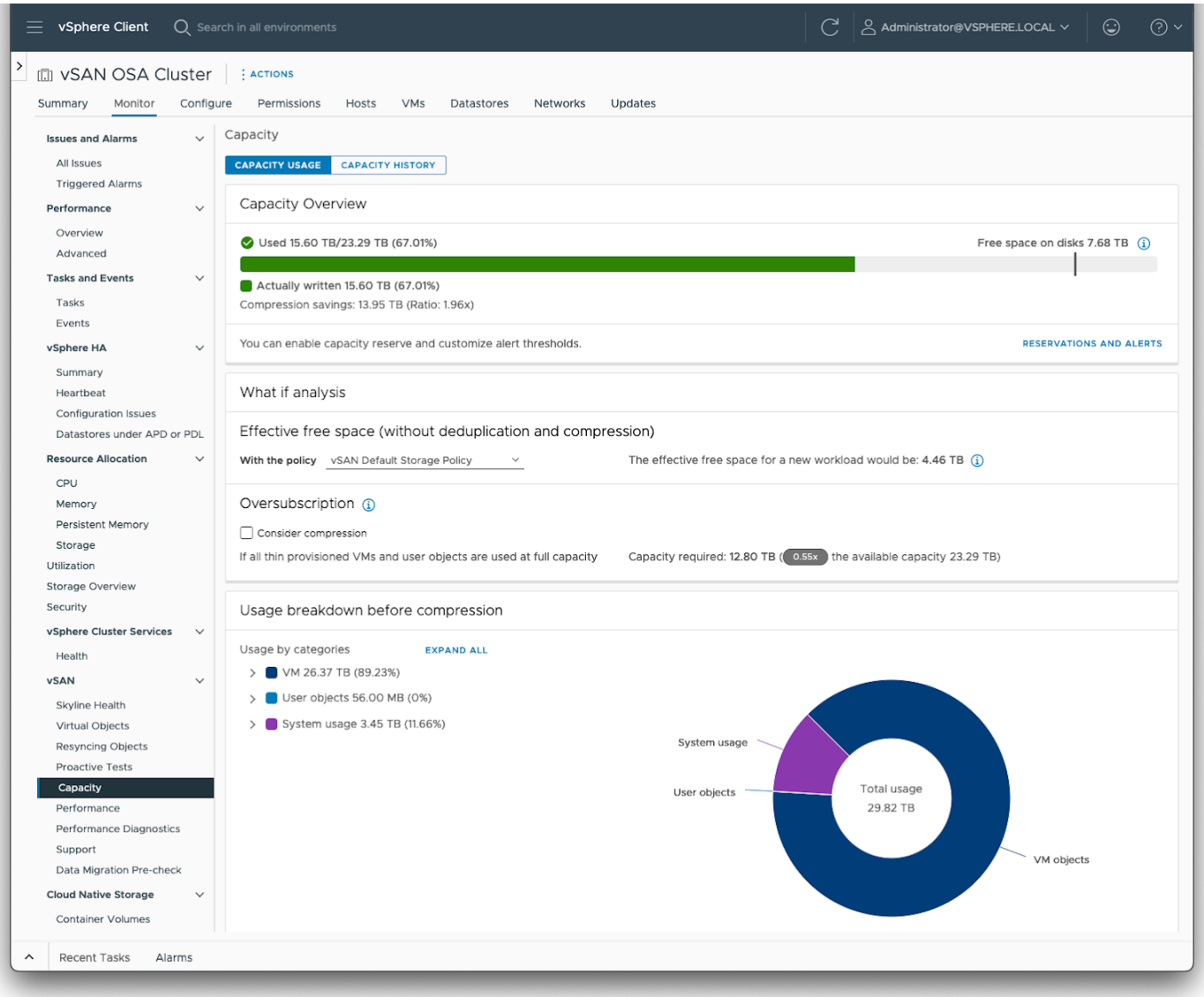

vSphere Client
Search in all environments
Administrator@VSPHERE.LOCAL
vSAN ESA Cluster
ACTIONS
Summary
Monitor
Configure
Permissions
Hosts
VMs
Datastores
Networks
Updates
Issues and Alarms
All Issues
Triggered Alarms
Performance
Overview
Advanced
Tasks and Events
Tasks
Events
vSphere DRS
Recommendations
Faults
History
VM DRS Score
CPU Utilization
Memory Utilization
Network Utilization
Resource Allocation
CPU
Memory
Persistent Memory
Storage
Utilization
Storage Overview
Security
vSphere Cluster Services
Health
vSAN
Skyline Health
Virtual Objects
Resyncing Objects
Proactive Tests
Capacity
Performance
Performance Diagnostics
Support
Data Migration Pre-check
Cloud Native Storage
Capacity
CAPACITY USAGE
CAPACITY HISTORY
Capacity Overview
Used 11.59 TB/31.84 TB (36.4%)
Free space on disks 14.08 TB
Actually written 11.59 TB (36.4%)
Compression savings: 36.15 TB (Ratio: 4.12x)
Operations reserve 6.17 TB (19.39%)
RESERVATIONS AND ALERTS
What if analysis
Effective free space (without deduplication and compression)
With the policy
vSAN Default Storage Policy
The effective free space for a new workload would be: 10.14 TB
Oversubscription
If all thin provisioned VMs and user objects are used at full capacity
Capacity required: 15.28 TB (
0.60x
the available capacity 25.67 TB)
Usage breakdown after compression
USAGE BY CATEGORIES
USAGE BY SNAPSHOTS
EXPAND ALL
VM 7.85 TB (16.44%)
User objects 16.89 GB (0.03%)
System usage 3.72 TB (7.8%)
Total usage
11.59 TB
VM objects
System usage
User objects
Recent Tasks
Alarms

## OSA Cluster – Compression only

Similarly, on an OSA cluster with just compression enabled, we have around 4TB used in the first instance, when the VMs have just been deployed:



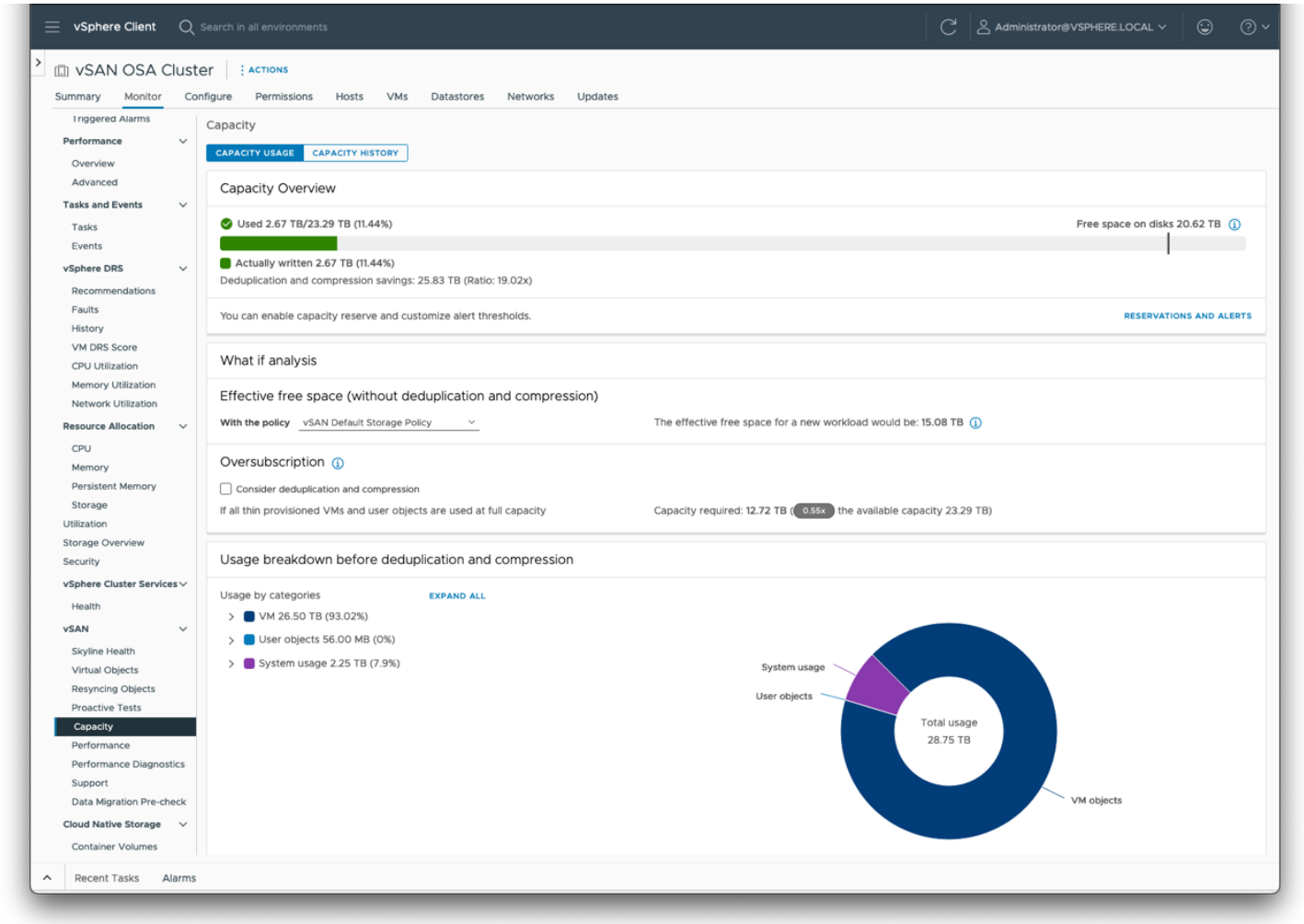

After decompression we have around 15.6TB used and a compression rate of around 2x. This is around half of the ratio that we saw from the ESA cluster:


vSphere Client
Search in all environments
Administrator@VSPHERE.LOCAL
vSAN OSA Cluster
ACTIONS
Summary
Monitor
Configure
Permissions
Hosts
VMs
Datastores
Networks
Updates
Issues and Alarms
All Issues
Triggered Alarms
Performance
Overview
Advanced
Tasks and Events
Tasks
Events
vSphere HA
Summary
Heartbeat
Configuration Issues
Datastores under APD or PDL
Resource Allocation
CPU
Memory
Persistent Memory
Storage
Utilization
Storage Overview
Security
vSphere Cluster Services
Health
vSAN
Skyline Health
Virtual Objects
Resyncing Objects
Proactive Tests
Capacity
Performance
Performance Diagnostics
Support
Data Migration Pre-check
Cloud Native Storage
Container Volumes
Capacity
CAPACITY USAGE
CAPACITY HISTORY
Capacity Overview
Used 15.60 TB/23.29 TB (67.01%)
Free space on disks 7.68 TB
Actually written 15.60 TB (67.01%)
Compression savings: 13.95 TB (Ratio: 1.96x)
You can enable capacity reserve and customize alert thresholds.
RESERVATIONS AND ALERTS
What if analysis
Effective free space (without deduplication and compression)
With the policy
vSAN Default Storage Policy
The effective free space for a new workload would be: 4.46 TB
Oversubscription
Consider compression
If all thin provisioned VMs and user objects are used at full capacity
Capacity required: 12.80 TB (
0.55x
the available capacity 23.29 TB)
Usage breakdown before compression
Usage by categories
EXPAND ALL
VM 26.37 TB (89.23%)
User objects 56.00 MB (0%)
System usage 3.45 TB (11.66%)
System usage
User objects
Total usage
29.82 TB
VM objects
Recent Tasks
Alarms

## OSA Cluster – Deduplication and Compression

Here we see the effects of enabling deduplication as well as compression. Note that this can take a long time to complete (as each disk group, in turn, will need to be taken offline, formatted, and brought online again) to enable the service.

As our dataset is the same across all VMs, we achieve an impressive 19x storage saving across the cluster.




It is important to remember that this is very much an ideal scenario for deduplication. Realistically, it may be very rare to have such a highly compressible dataset that is exactly homogenous across the cluster.

## RAID-5/RAID-6 Erasure Coding

Storage policies that direct vSAN to use RAID-5/6 with erasure coding can provide better space efficiency compared to RAID-1 without erasure coding. Instead of the 200% or 300% overhead with traditional RAID-1 (assuming FTT = 1 or 2 respectively) , RAID-5 requires only 33% additional storage, and RAID-6 requires only 50% additional overhead.

### RAID-5/RAID-6 Erasure Coding - vSAN OSA

In vSAN OSA, to support RAID-5 and RAID-6, the following host requirements must be met:

- RAID-5 (3+1): minimum of four hosts; 1.3x space capacity consumed
- RAID-6 (4+2): minimum of six hosts; 1.5x space capacity consumed

### RAID-5/RAID-6 Erasure Coding - vSAN ESA

vSAN ESA, replaced the traditional 3+1 with RAID-5 scheme with two separate options:

- RAID-5 (2+1): three to five hosts; 1.5x space capacity consumed
    - Opens opportunities to reduce capacity usage for smaller vSAN clusters that relied on RAID-1 topologies requiring 2x space capacity consumed
    - For more information on RAID-1 Performance using RAID5/RAID-6 - https://core.vmware.com/blog/raid-56-performance-raid-1-using-vsan-express-storage-architecture
- RAID-5 (4+1): minimum of five hosts; 1.25x space capacity consumed

vSAN ESA includes new ***Adaptive RAID-5*** functionality. Depending on the number of hosts in the cluster, vSAN ESA will automatically adjust the RAID-5 mode. vSAN ESA presents a single RAID-5 storage policy rule for you to select and will adapt the RAID-5 scheme based on the host count of the cluster. Additionally, it will determine which RAID-5 scheme to use not by the minimum hosts required, but by the minimum hosts ***recommended*** to ensure there is a spare fault domain (host) whenever possible. Adaptive RAID-5 automatically re-arranges data as the cluster size increases or decreases.

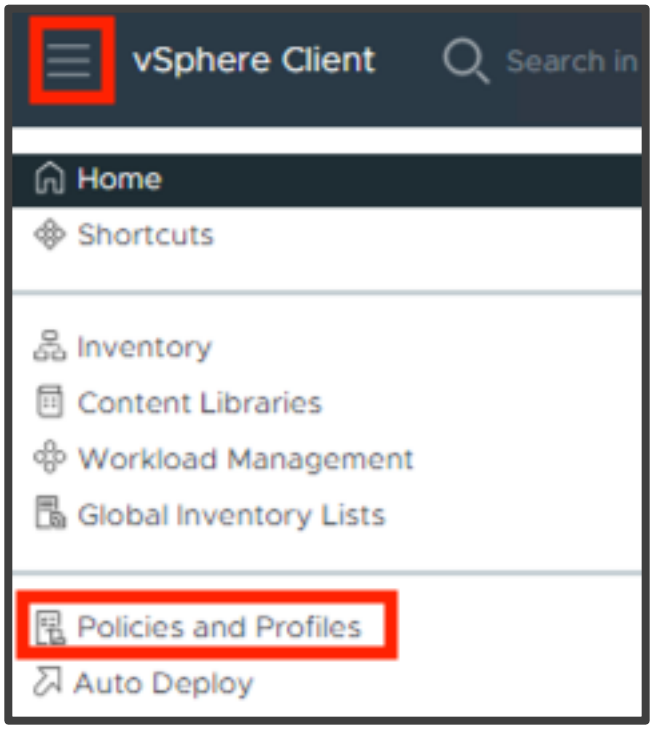
For more details on Adaptive RAID-5 Erasure Coding in vSAN ESA, visit:
https://core.vmware.com/blog/adaptive-raid-5-erasure-coding-express-storage-architecture-vsan-8

*Note: The erasure coding architecture in vSAN ESA provides the space savings with the same level of performance as RAID-1. Therefore, for most clusters, the recommended storage policy applied to the VMs should be RAID-5. For even higher levels of resilience and space efficiency without compromising performance, consider standardizing on FTT=2 using RAID-6 in clusters with 7 or more hosts.*

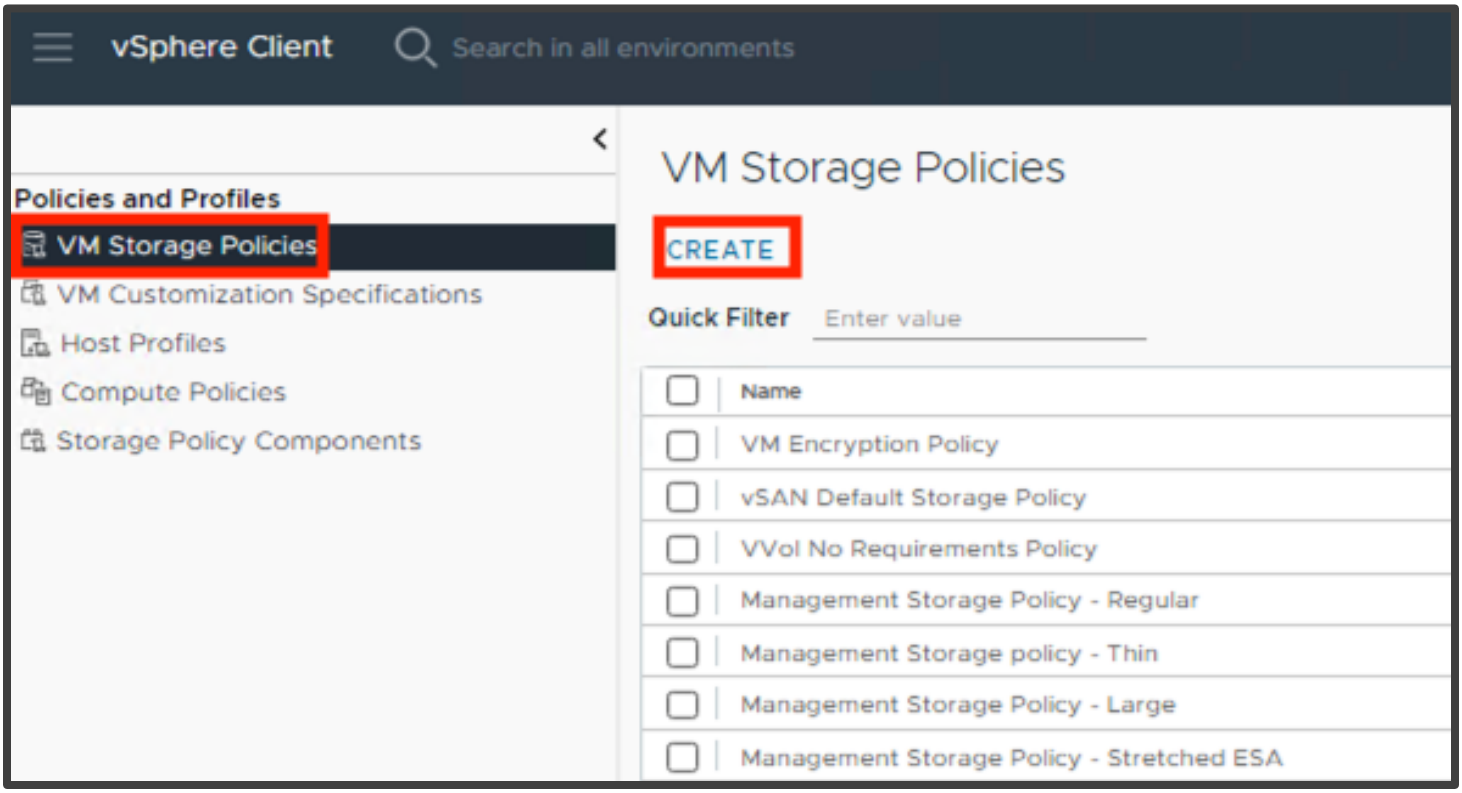
Below we show how to create a RAID-5 policy and how components are distributed with vSAN OSA and ESA.

### Create RAID-5 Storage Policy and Apply

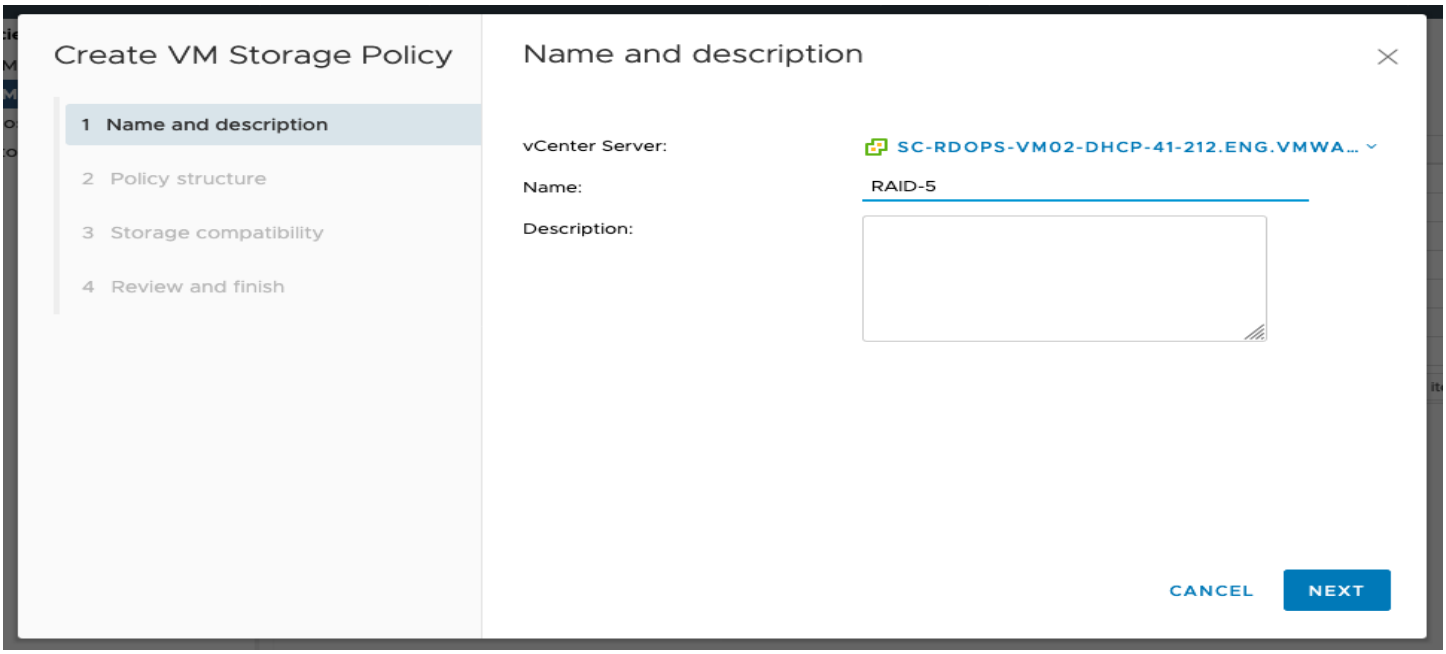
We can easily create a RAID-5 storage policy through vCenter. Navigate to: **Menu > Policies and Profiles > VM Storage Policies**:



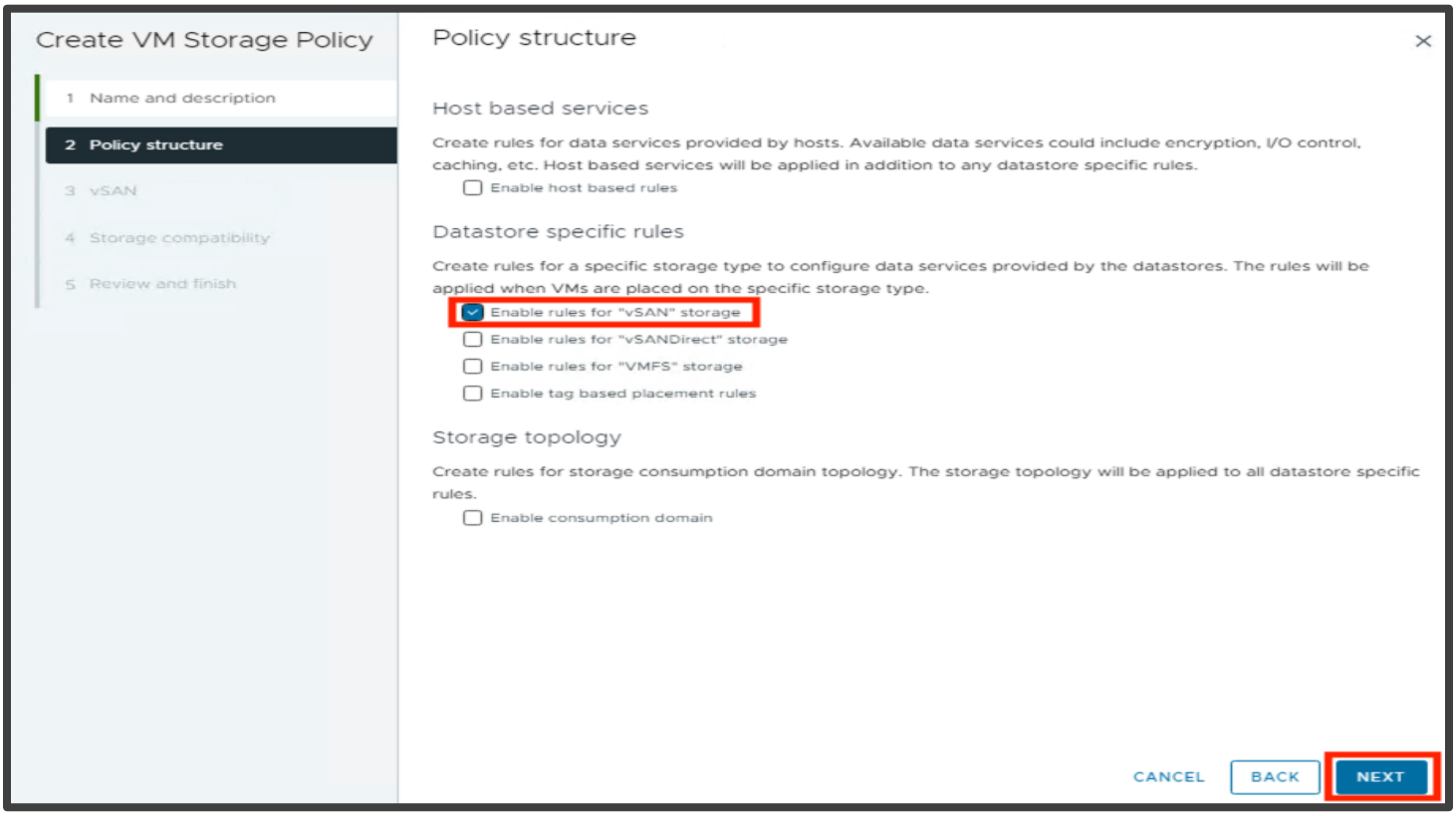

Next, navigate to 'VM Storage Policies' and click **Create**:



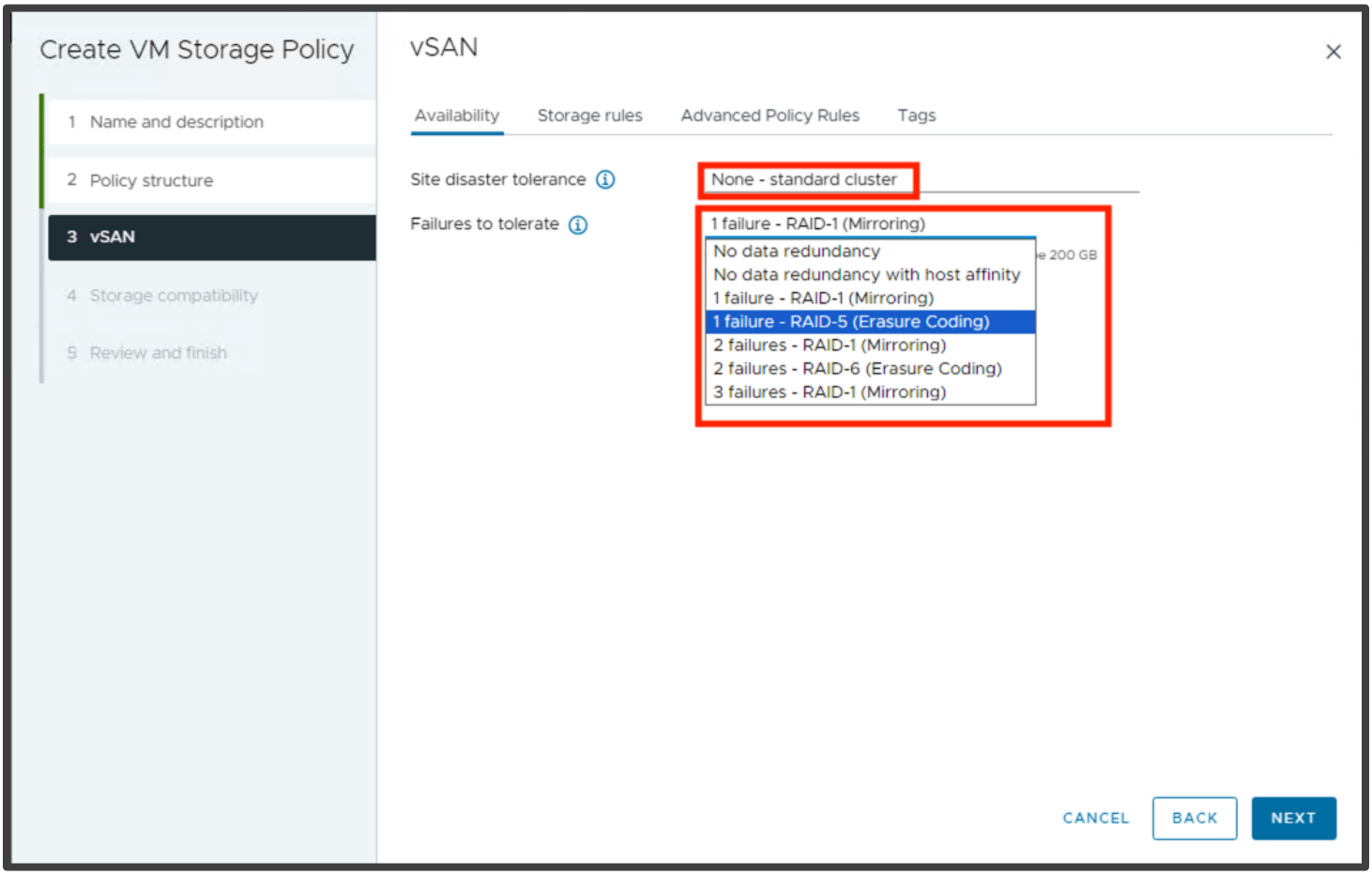
Select the appropriate vCenter Server, create a name and click **Next**:



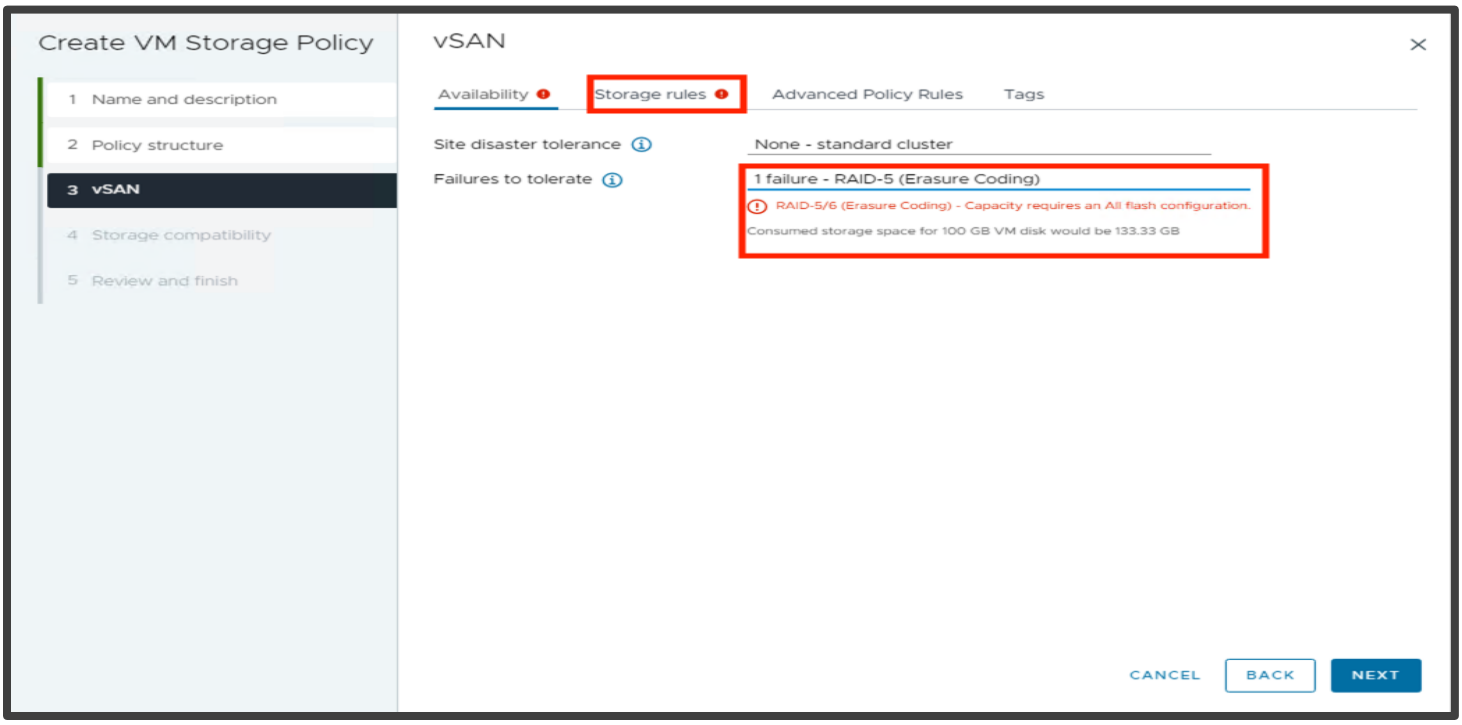

Ensure 'Enable rules for "vSAN" storage' is checked and click **Next**:



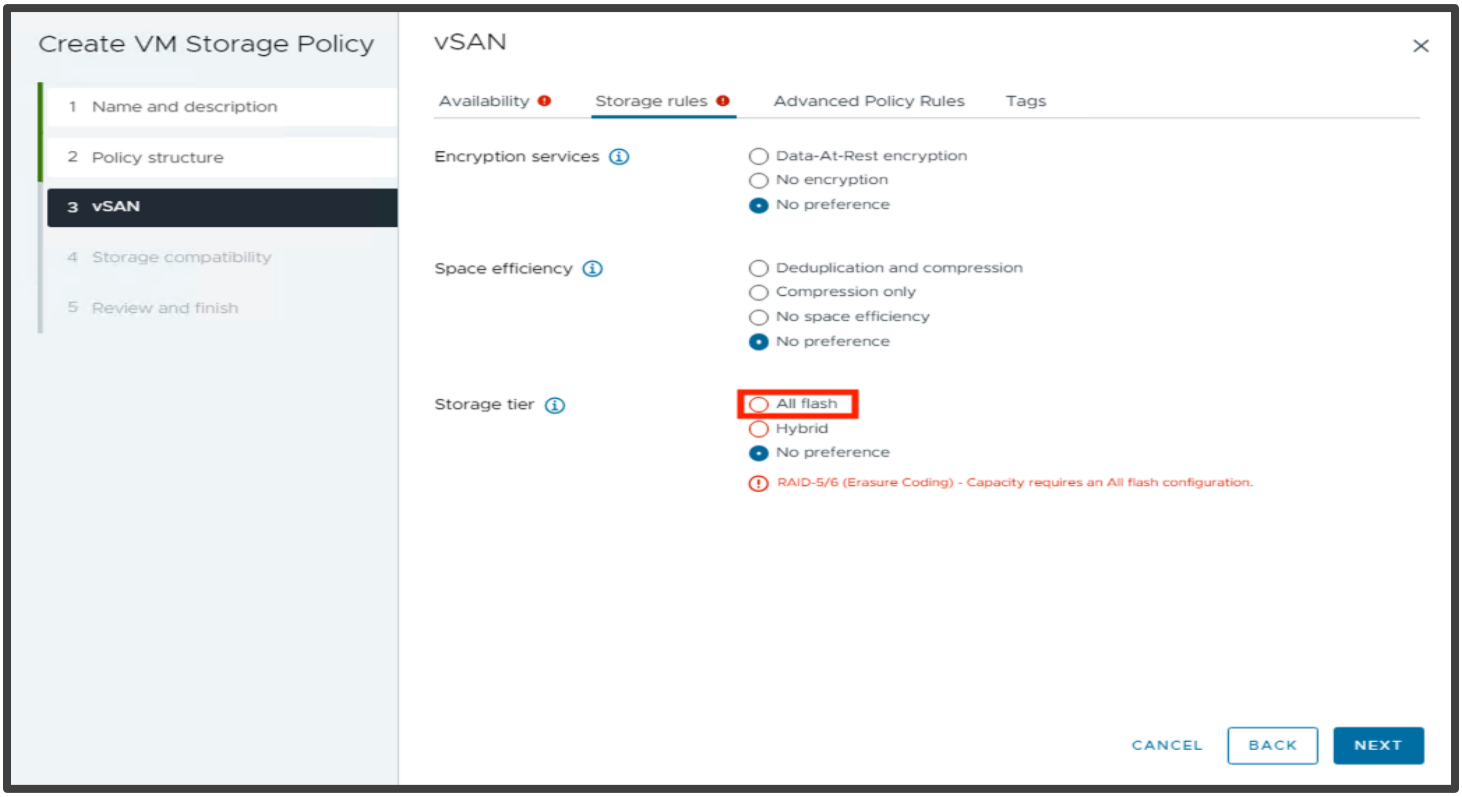
Select 'None - standard cluster' from the 'Site disaster tolerance' drop-down' and '1 failure - RAID-5 (Erasure Coding)' for 'Failures to tolerate" drop-down, then click **Next**:



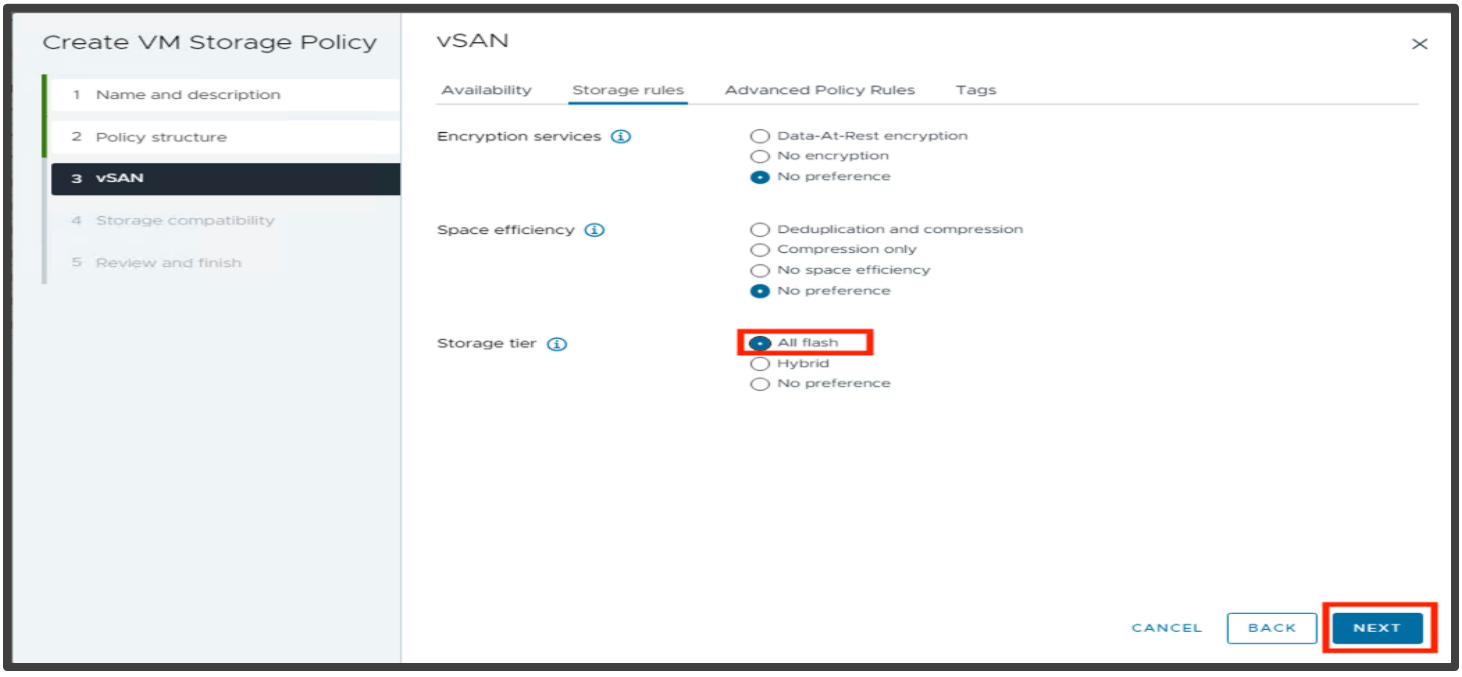
Once "1 failure - RAID-5 (Erasure Coding)' is selected, you may see this warning 'RAID-5/6 (Erasure Coding) - Capacity requires an All-flash configuration." If so, select the 'Storage rules" tab:



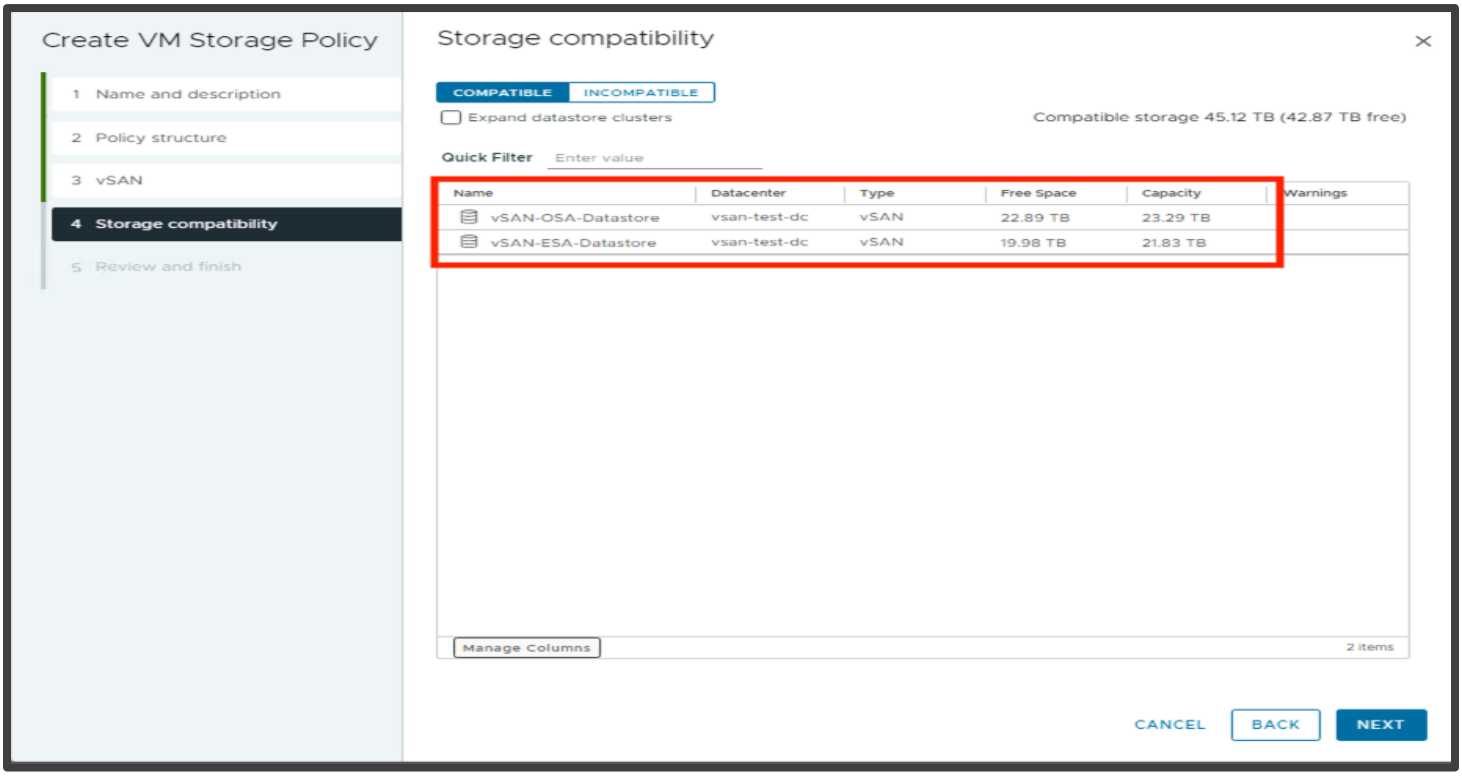

Once in the Storage rules tab, you will see:



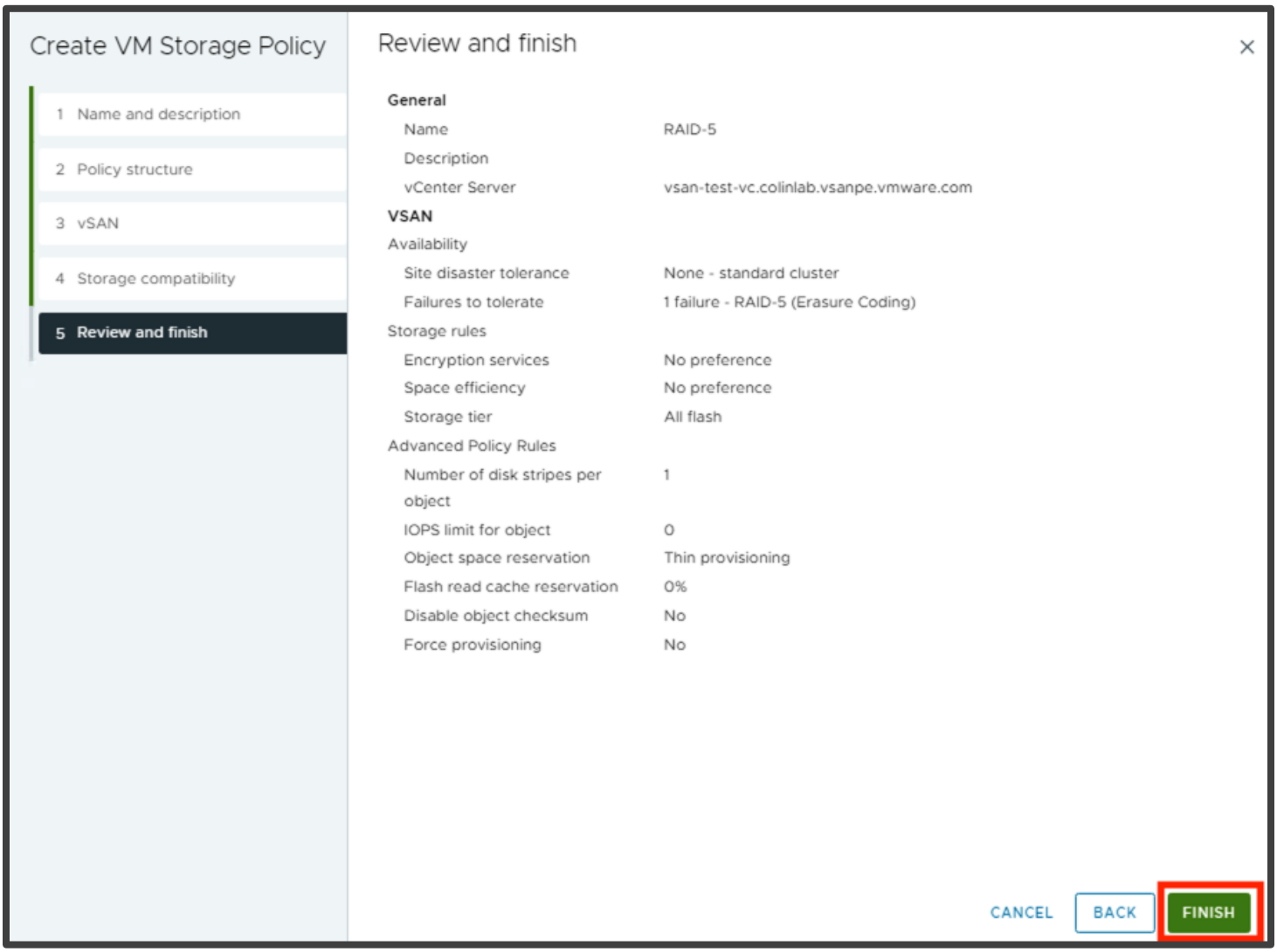
Check the 'All flash' radio, then click **Next**:



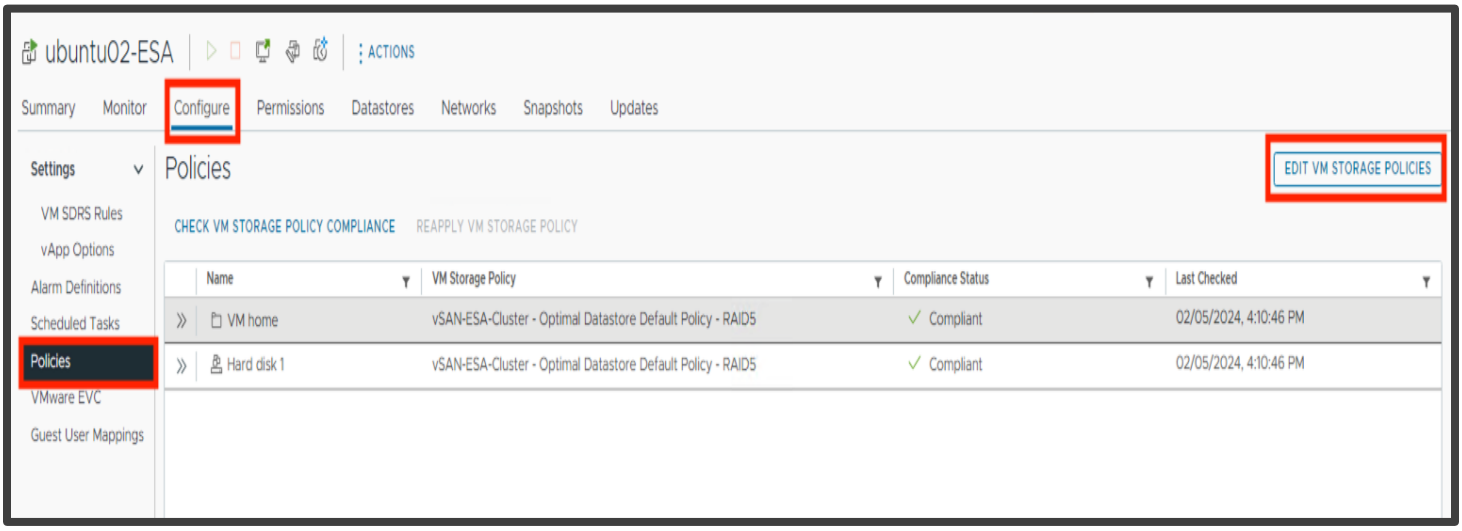

We see that the vSAN datastore is compatible with this policy (in this example there is both an OSA and ESA datastore listed), click **Next**:



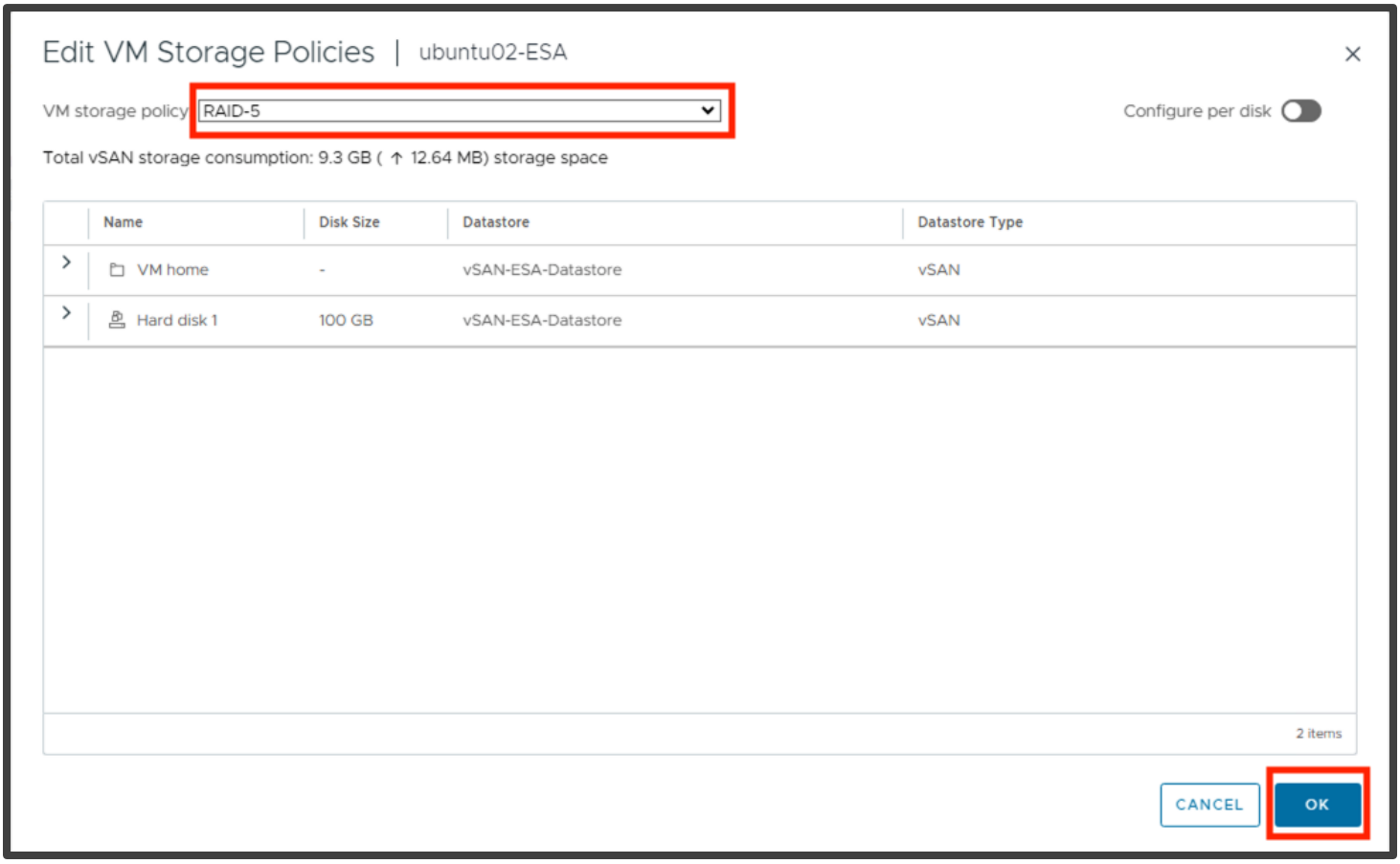
Finally, review and click **Finish**:



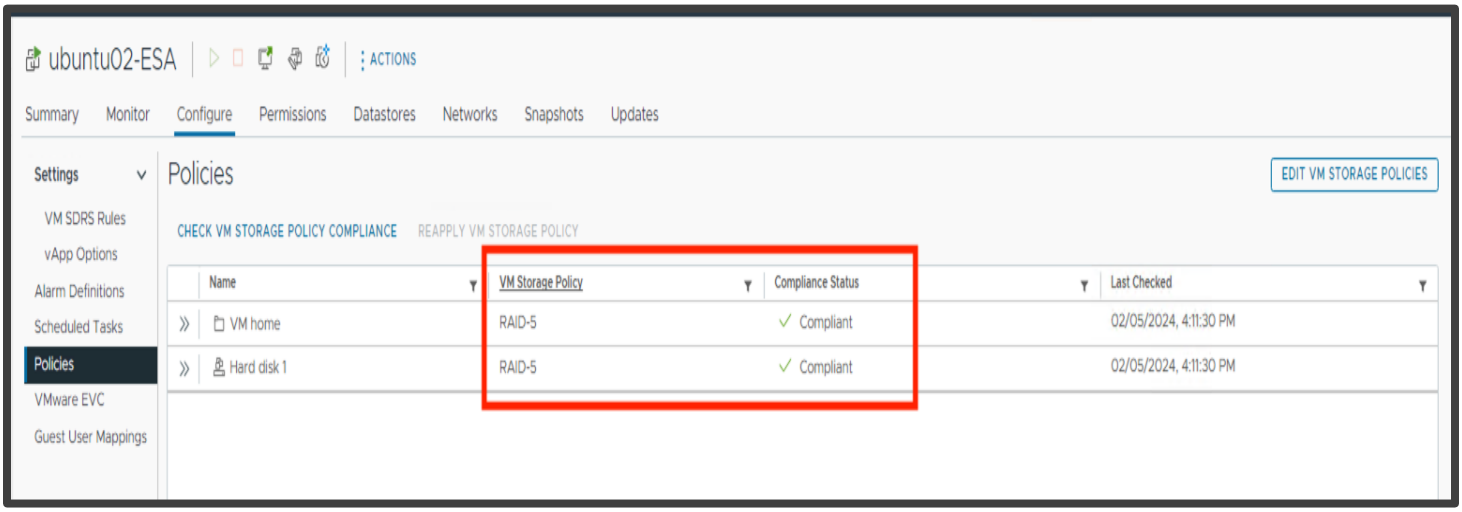
Next, we apply this storage policy to an existing VM. Navigate to **[Virtual Machine] > Configure > Policies** and click **Edit** VM storage policies:




Select the newly created storage policy and click **OK**:



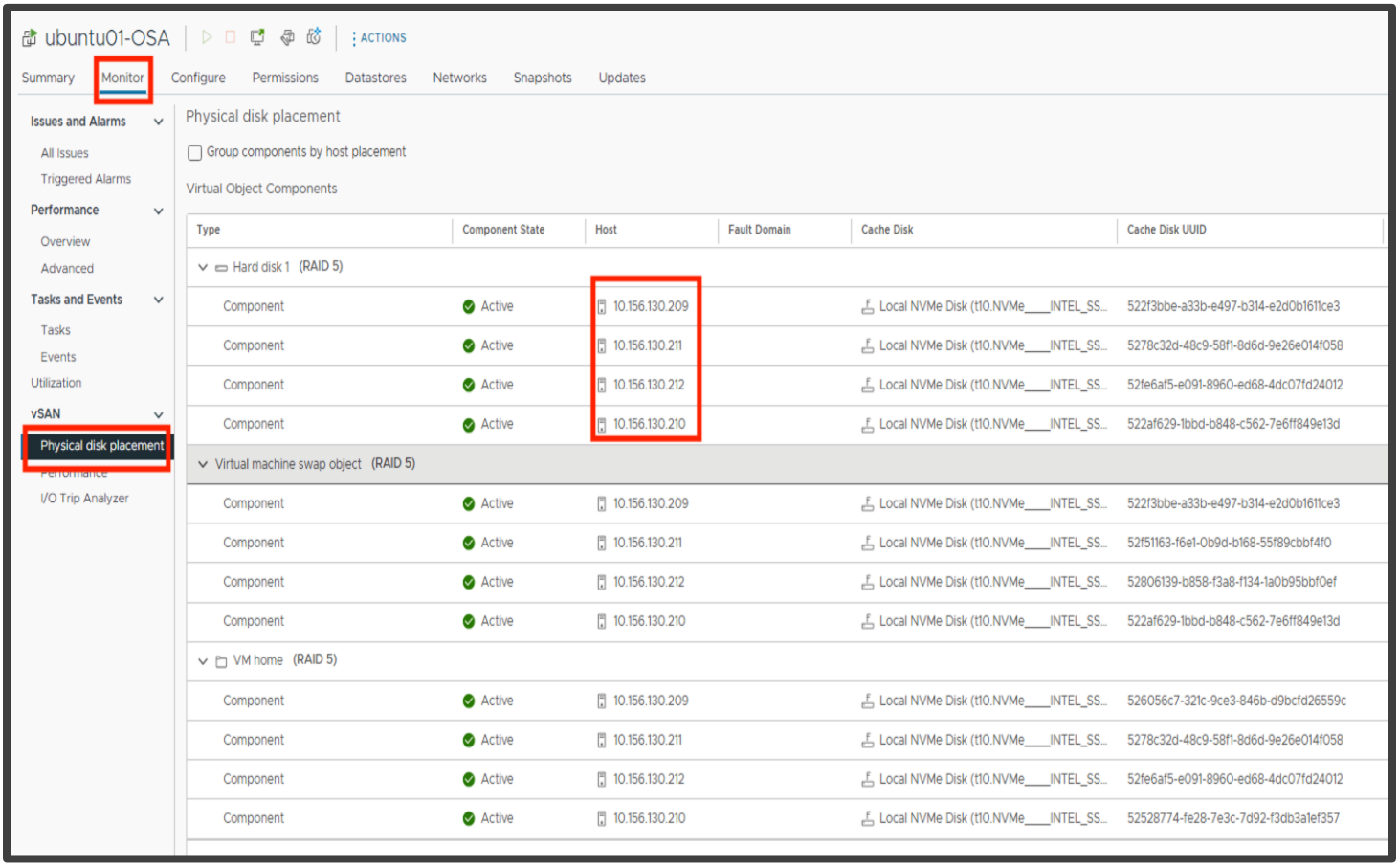
After this has been set, vSAN will move the data components as per the policy. Once this has been completed, the VM's disks will show as compliant to the policy:




We can now observe the data objects are arranged in vSAN OSA and ESA clusters

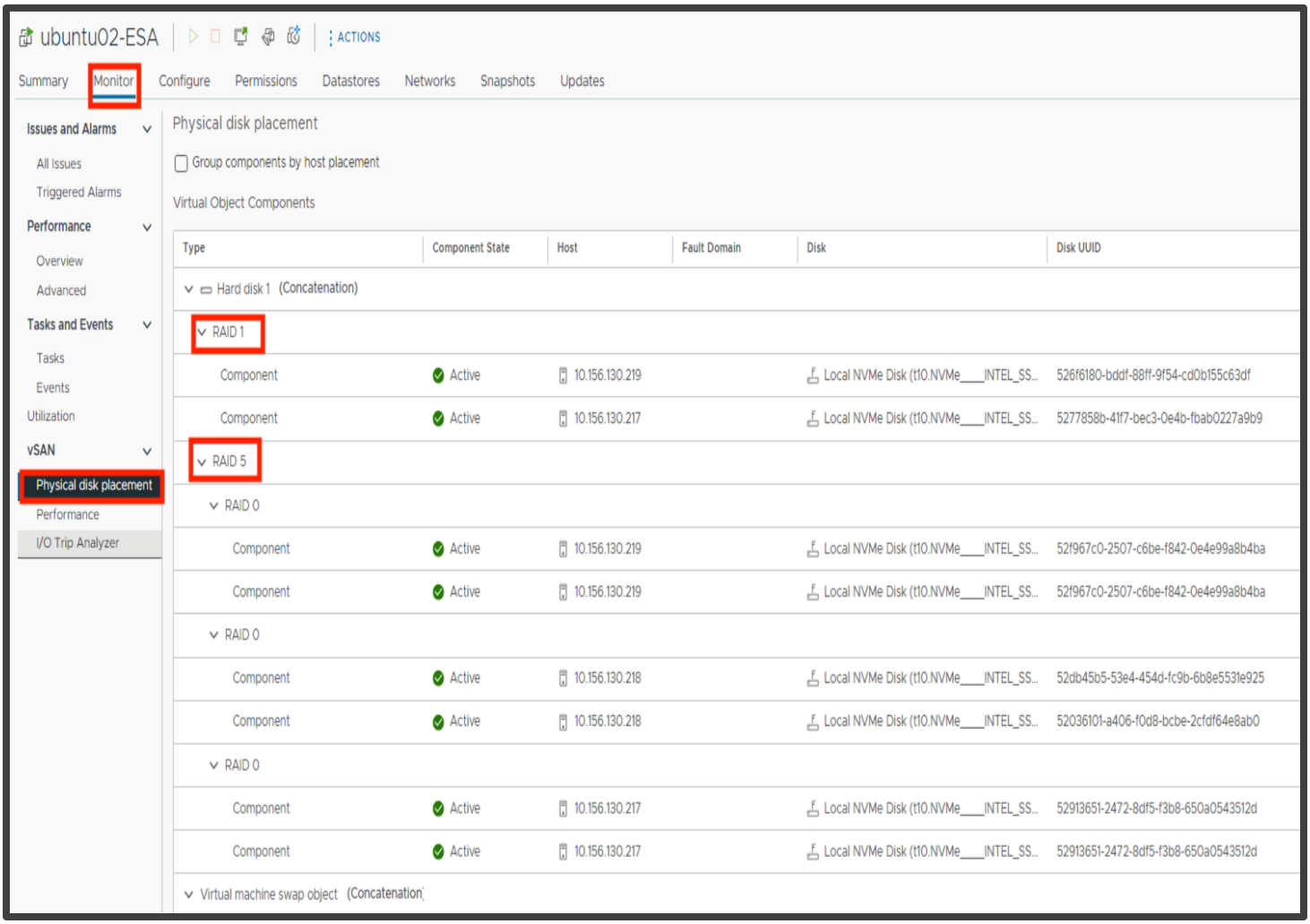
### RAID-5 Data Placement in vSAN OSA

Navigate to **[Virtual Machine] > Monitor > Physical disk placement**. This screen shows that the components are now spread over four hosts, i.e. RAID-5 3+1:



### RAID-5 Data Placement in vSAN ESA

As above, we navigate to **[Virtual Machine] > Monitor > Physical disk placement**. As expected, we see the performance leg remain as RAID-1. Moreover, as we have four hosts, vSAN adaptive RAID-5 will select the 2+1 layout for the capacity leg. As shown below, the capacity leg is RAID-5, spread over three hosts, i.e., RAID-5 2+1:



## Trim/Unmap

vSAN supports space reclamation on virtual disks using trim commands issued from the guest VM operating system.

### Guest Requirements

The following should be met for trim/unmap to work:

- At least VM hardware version 11 (Windows) or version 13 (Linux)
- The setting 'disk.scsiUnmapAllowed' in the VM's VMX file set to true (default)
- The VM's operating system recognizes the disk as 'thin'

For more details, visit:
https://core.vmware.com/resource/vsan-space-efficiency-technologies#sec19560-sub5

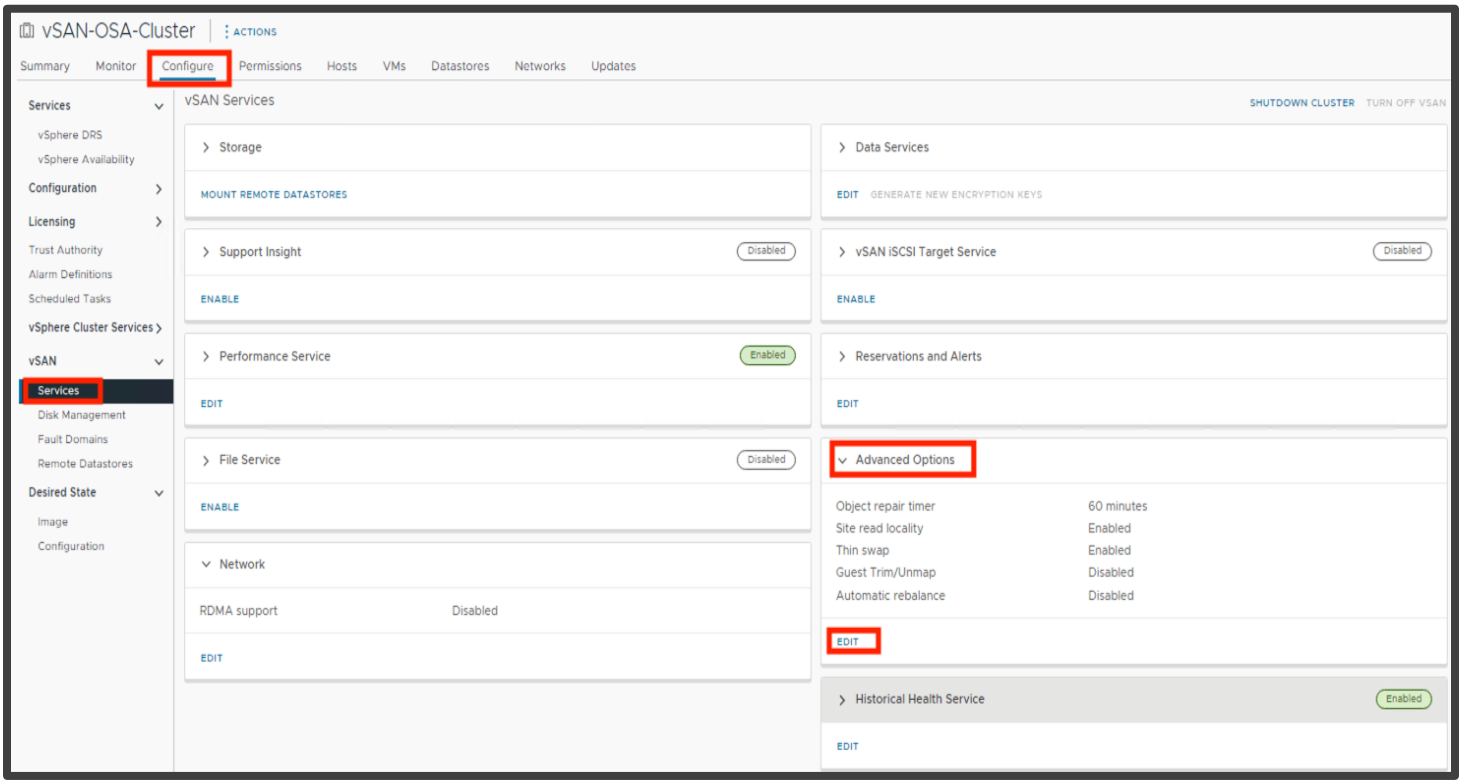
### Enabling Trim/Unmap on an ESA Cluster

Trim/Unmap functionality is enabled by default in vSAN ESA clusters.

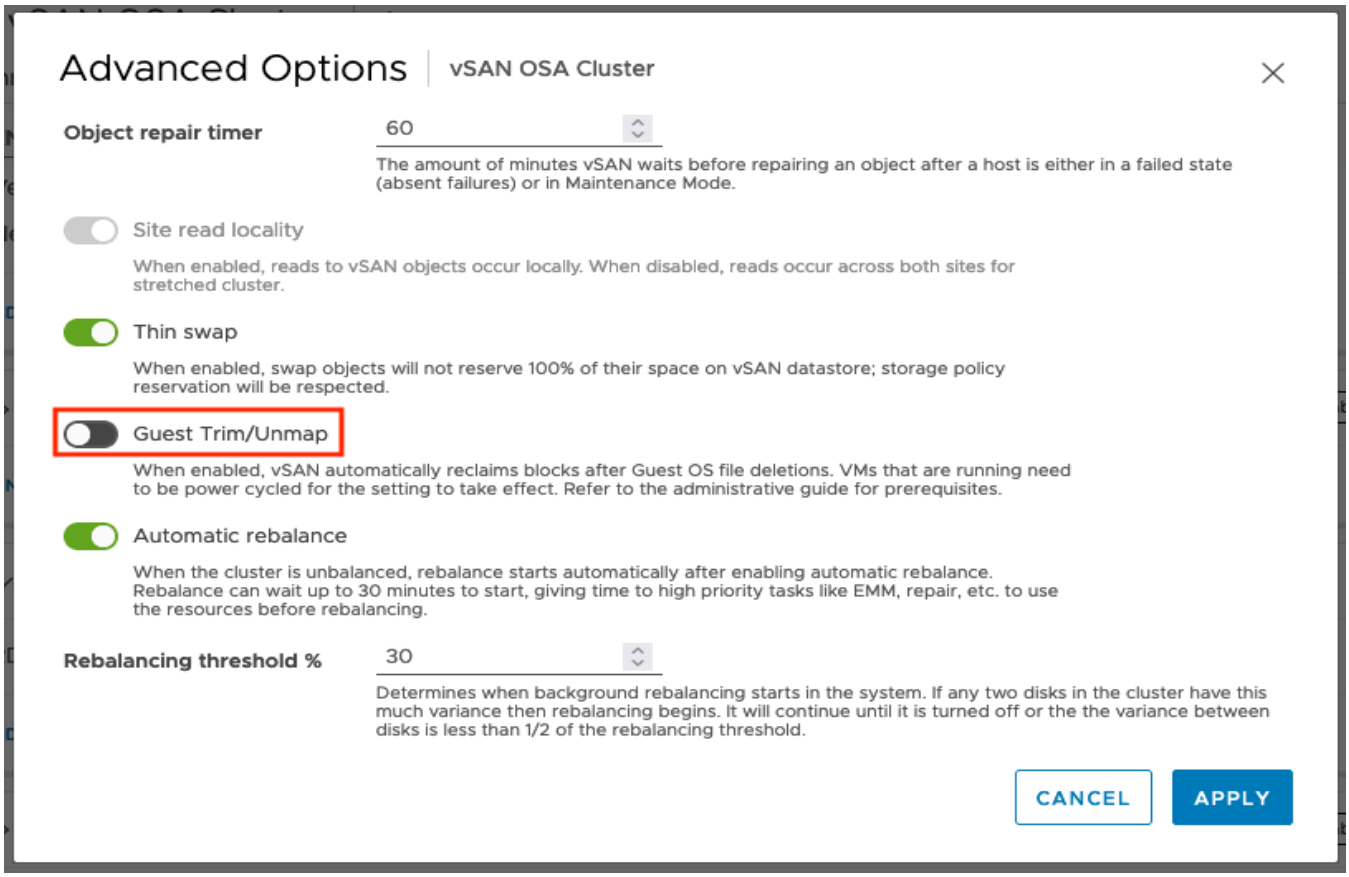
Trim/Unmap functionality can be explicitly disabled in the VMX file by the setting *disk.scsiUnmapAllowed* set to *false*.

### Enabling Trim/Unmap on an OSA Cluster

To enable this feature for vSAN OSA, a cluster-wide setting for enabling unmap is set by navigating to **[vSAN Cluster] > Configure > Services > Advanced Options** then click the **EDIT** button that corresponds to the **Advanced Options** section:



Then toggle the 'Guest Trim/Unmap' setting. Note that VMs will need to be power cycled for this setting to be effective:

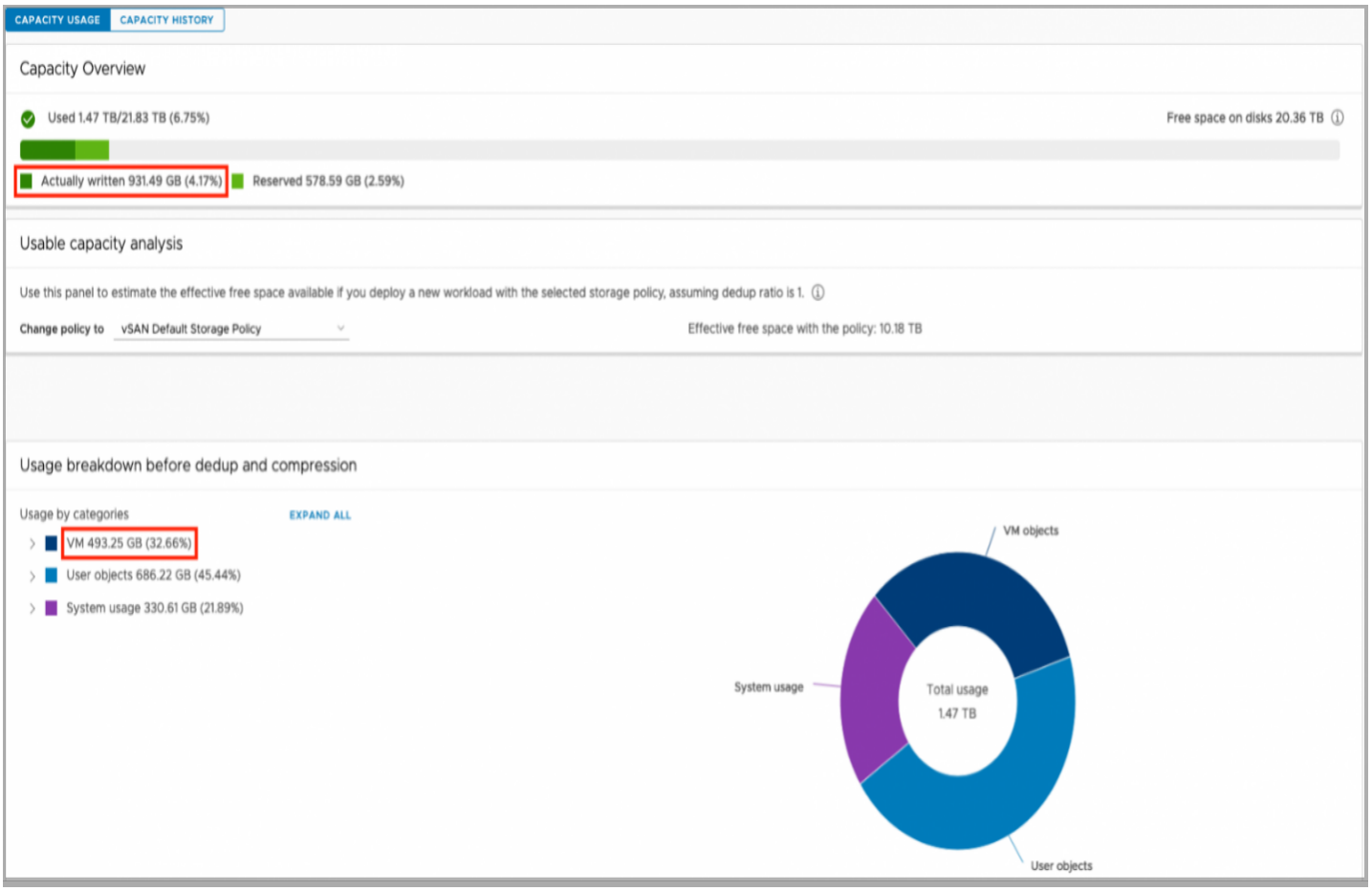



Once unmap is enabled on the cluster, guest VMs can issue commands (such as fstrim) to free any previously deleted data. Trim/unmap is enabled by default on vSAN ESA.

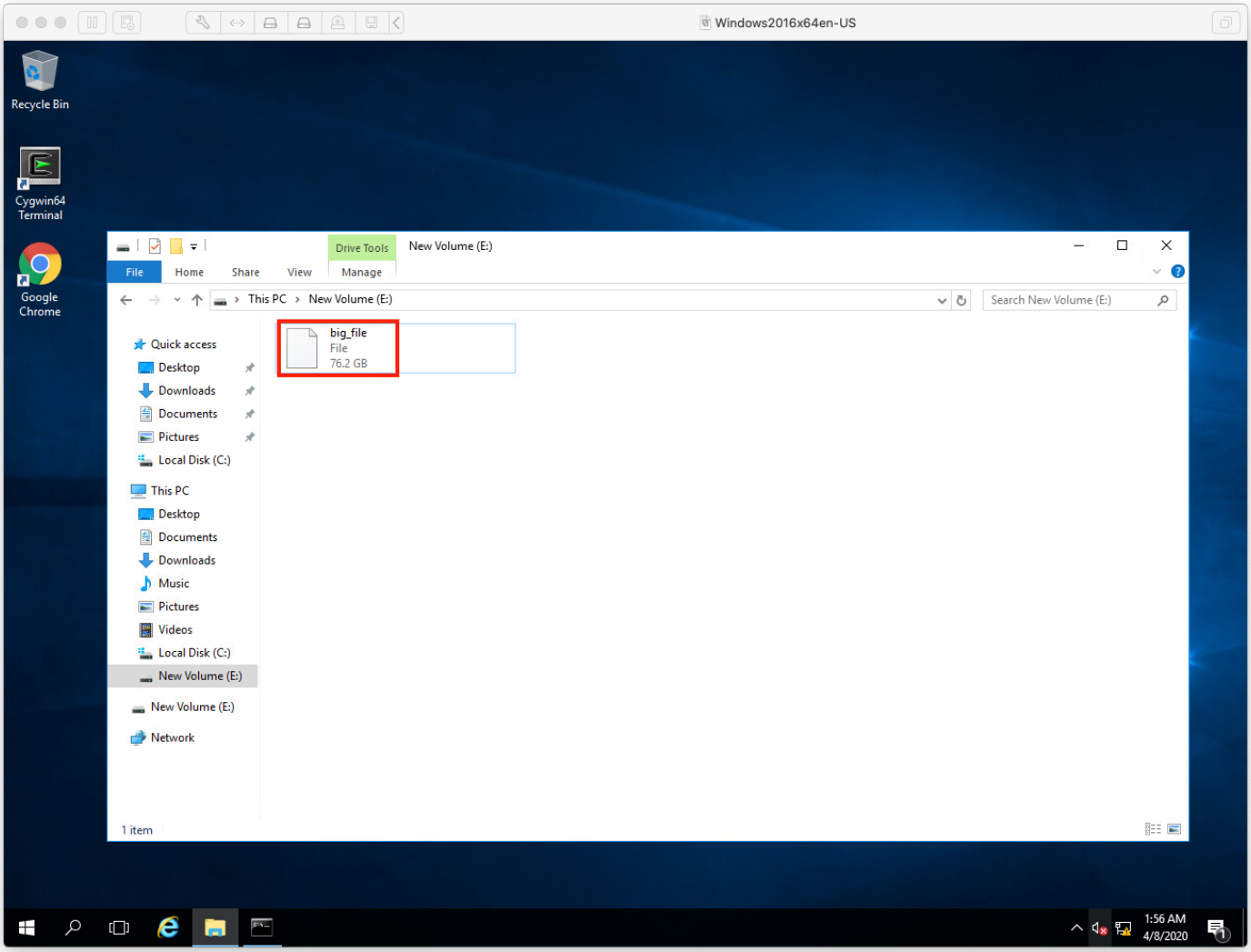
### Example on an OSA Cluster

To demonstrate the effects of this on a vSAN OSA cluster, firstly we observe how much space is in use currently, by navigating to **[vSAN Cluster] > [Monitor] > [Capacity]**.

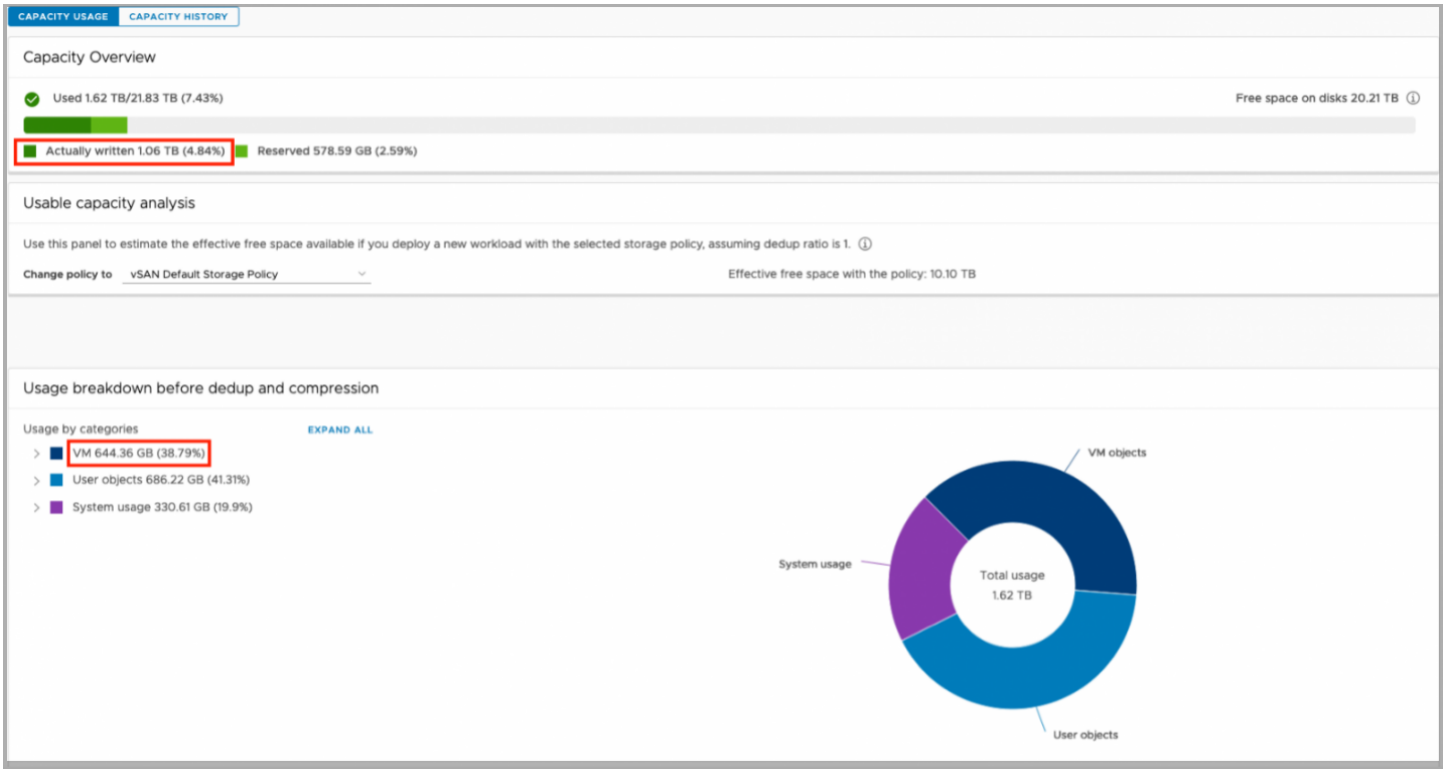
In this example, we can see that around 931GB of space is currently in use, with around 493GB of VM data:



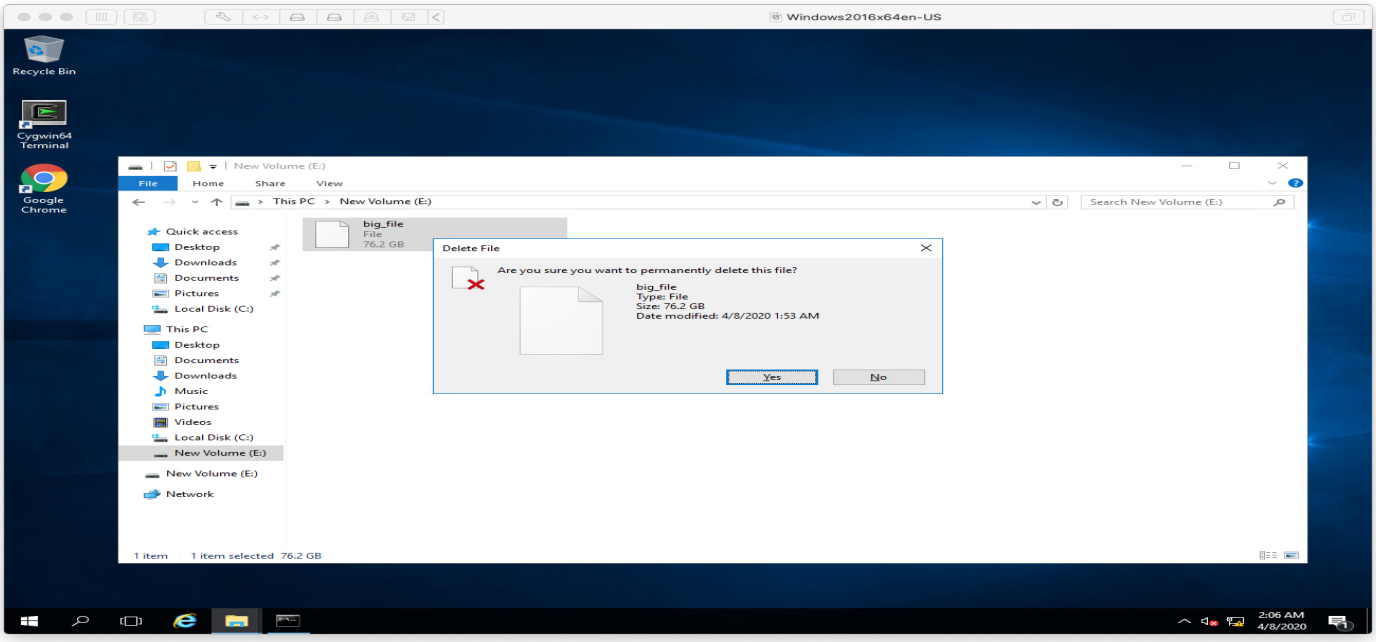
Next, we create or copy a large file on our guest VM. In this case a Windows 2016 VM is used, and a large (~76GB) file has been created:



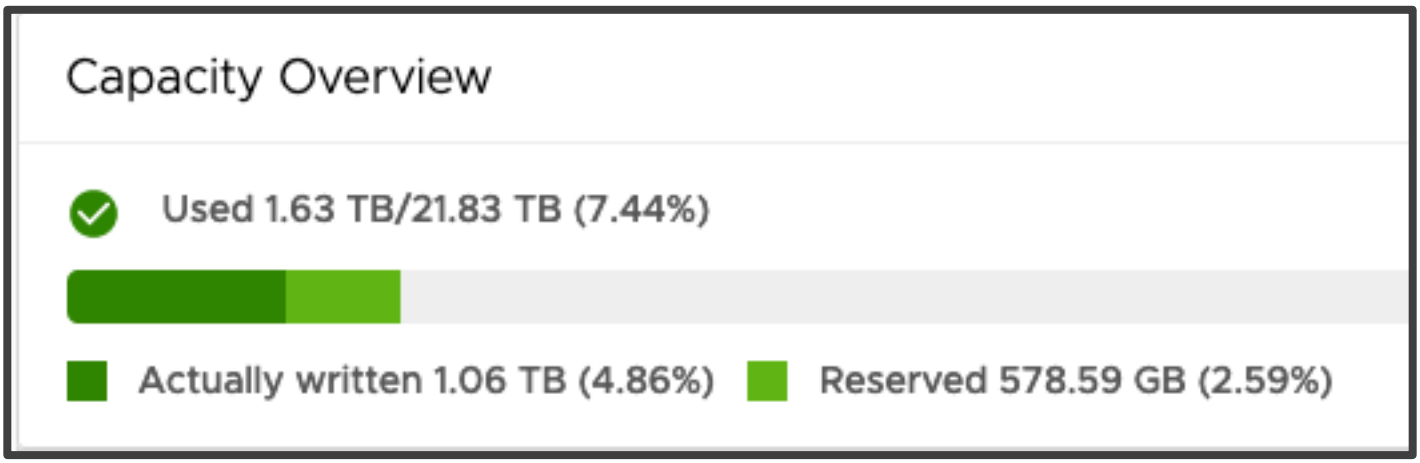
As expected, our space utilization increases by around 76GBx2 (as this is a RAID-1 object). Thus, 76GBx2 + 493GB gives us around 644GB, as we see below:




We now delete the file:

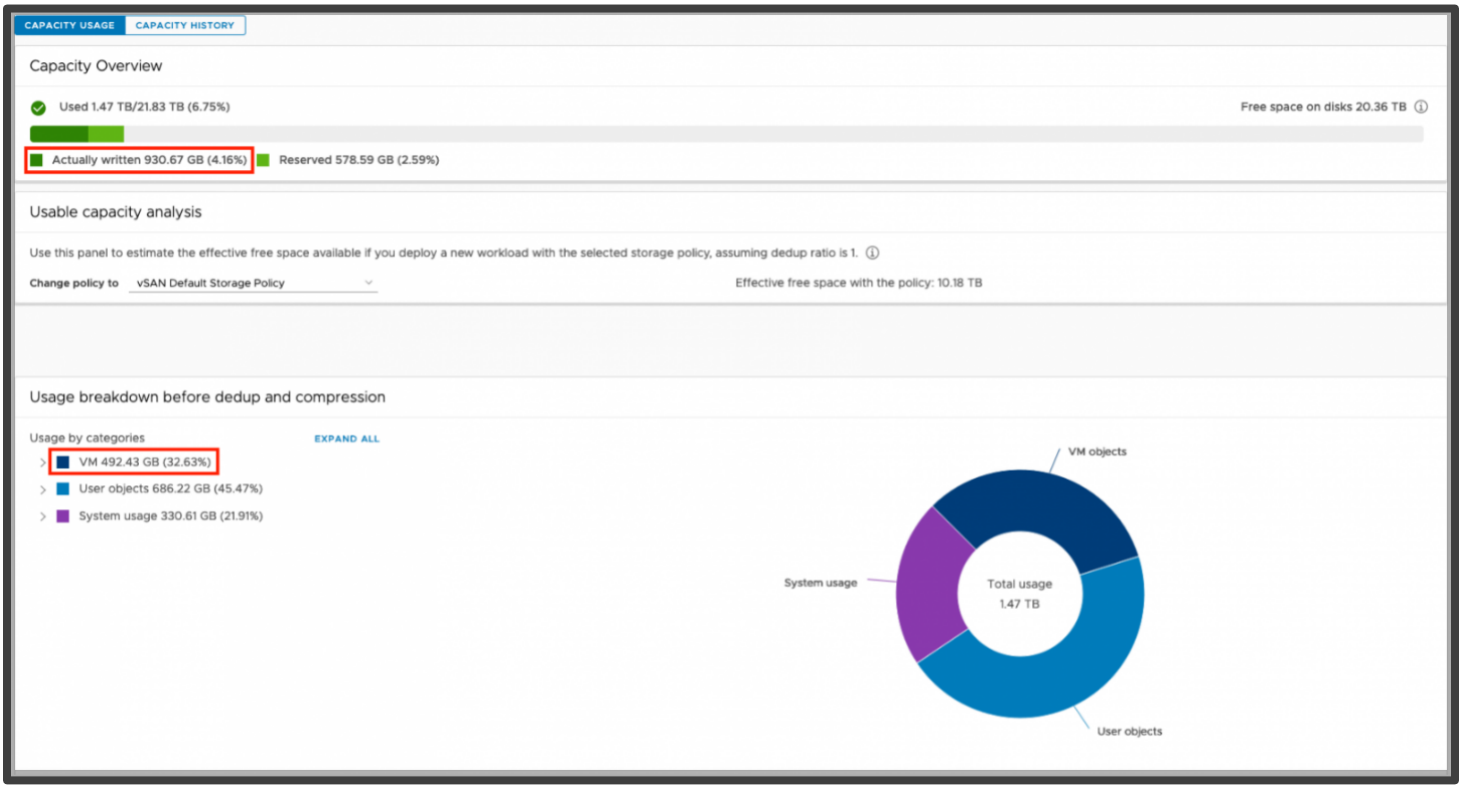


Looking back at the capacity view, we can see that the space consumed is still the same:




We now enable the guest trim/unmap advanced setting for the cluster (see above).

Looking back at vCenter, we see that the used space has been freed:



## vSAN Max™ - Disaggregated Storage

What is vSAN Max? VMware's new disaggregated storage offering that provides Petabyte-scale centralized shared storage for your VMware vSphere® (vSphere) clusters.

Built on the foundation of vSAN ESA, vSAN Max is a fully distributed architecture, where access to data is NOT funneled through centralized I/O controllers. Instead, vSAN Max uses the full power of each node (host) in the vSAN Max cluster. Aggregated resources across all hosts in a vSAN Max cluster to the process I/O. Each additional host linearly scales capacity and performance.

For more additional technical information please review:

- Main vSAN Max informational page - https://core.vmware.com/vsan-max
- vSAN Max Design & Operations Guide - https://core.vmware.com/resource/vsan-max-design-and-operational-guidance
- vSAN Max introduction blog post - https://core.vmware.com/blog/introducing-vsan-max
- vSAN Max scalability blog post - https://core.vmware.com/blog/vsan-max-and-advantage-scalability

### vSAN Max Sizing Considerations

When sizing vSAN Max deployments, consider that vSAN Max clusters support:

- A maximum of 32 ESXi hosts in the cluster (24 ESXi hosts recommended)
- A maximum of 10 compute clusters mounting to a single vSAN Max cluster
- A maximum of 128 total ESXi hosts (both within the vSAN Max cluster and the vSAN Compute clusters connecting to a single vSAN Max datastore)

*Note: Limiting the vSAN Max cluster size to 24 ESXi hosts will allow for up to 104 ESXI hosts from vSAN compute clusters to mount the datastore, offering a 4.3:1 ratio. A vSAN Max cluster size of 32 ESXi hosts would allow for up to 96 ESXI hosts from vSAN compute clusters to mount the datastore, offering a storage ratio of 3:1.*

### Disaggregated Storage for vSAN OSA (AKA: HCI Mesh)

Although the vSAN Max is explicitly a vSAN ESA function, vSAN OSA deployments still support disaggregated storage.

vSAN OSA datastores can be shared between two vSAN clusters, utilizing vSAN's native data path for cross-cluster connections. Compute Only Clusters are also supported.

Each vSAN OSA client cluster can mount a maximum of ten remote vSAN OSA datastores. A vSAN OSA server cluster can export its datastore up to a maximum of ten client clusters.

All vSAN features are supported except for Data-in-Transit encryption, Cloud Native Storage (including vSAN Direct), Stretched Clusters, and 2-Node Clusters. Additionally, HCI Mesh will not support remote provisioning of File Services Shares, iSCSI volumes, or First-Class Disks (FCDs). File Services, FCDs, and the iSCSI service can be provisioned locally on clusters participating in a mesh topology but may not be provisioned on a remote vSAN datastore.

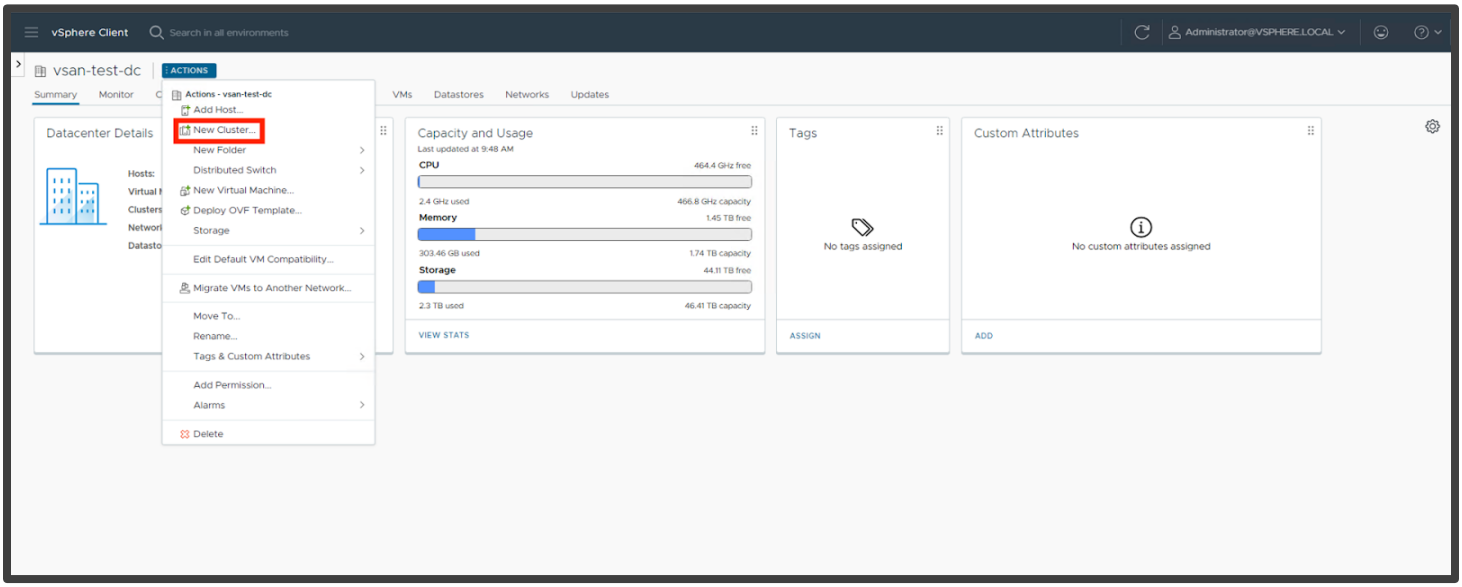
The same MTU sizing is required for both the Client and Server clusters.

## Using Quickstart to Enable vSAN Max Cluster

vSAN Max leverages vSAN ESA, as such the initial enablement process is very similar to the steps reviewed in the Using Quickstart to Enable Single vSAN HCI Cluster section of the vSAN Proof of Concept: vSAN Architecture Overview & Setup Guide.

### Initialize Cluster

Navigate to your **Datacenter** > Click **Actions** > **New Cluster**.

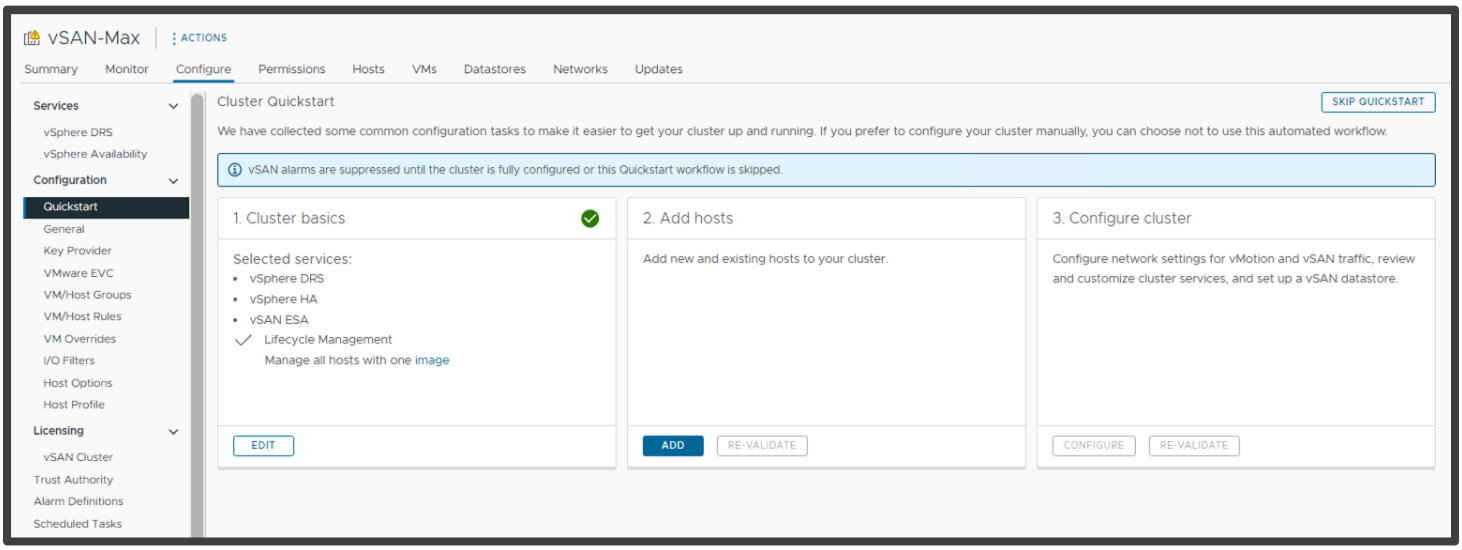



The New Cluster screen pops-up and we are presented with a dialog to enable services. Provide a name for the cluster and select vSAN from the list of services. Ensure that vSAN ESA (the default). For the Quickstart workflow to configure the vMotion VMkernel, vSphere DRS must be set to enabled.

We can also setup the cluster to use a single image (thereby enabling vLCM). For more information on vLCM, see: https://core.vmware.com/resource/introducing-vsphere-lifecycle-management-vlcm.

### Quickstart – Cluster Basics

The initial cluster creation above initializes the Quickstart process. Once the cluster has been created, navigate to **[vSAN Cluster] > Configure > Quickstart**. On this screen you will be able to confirm the basic services selected previously then move to the add hosts and configuration phases.



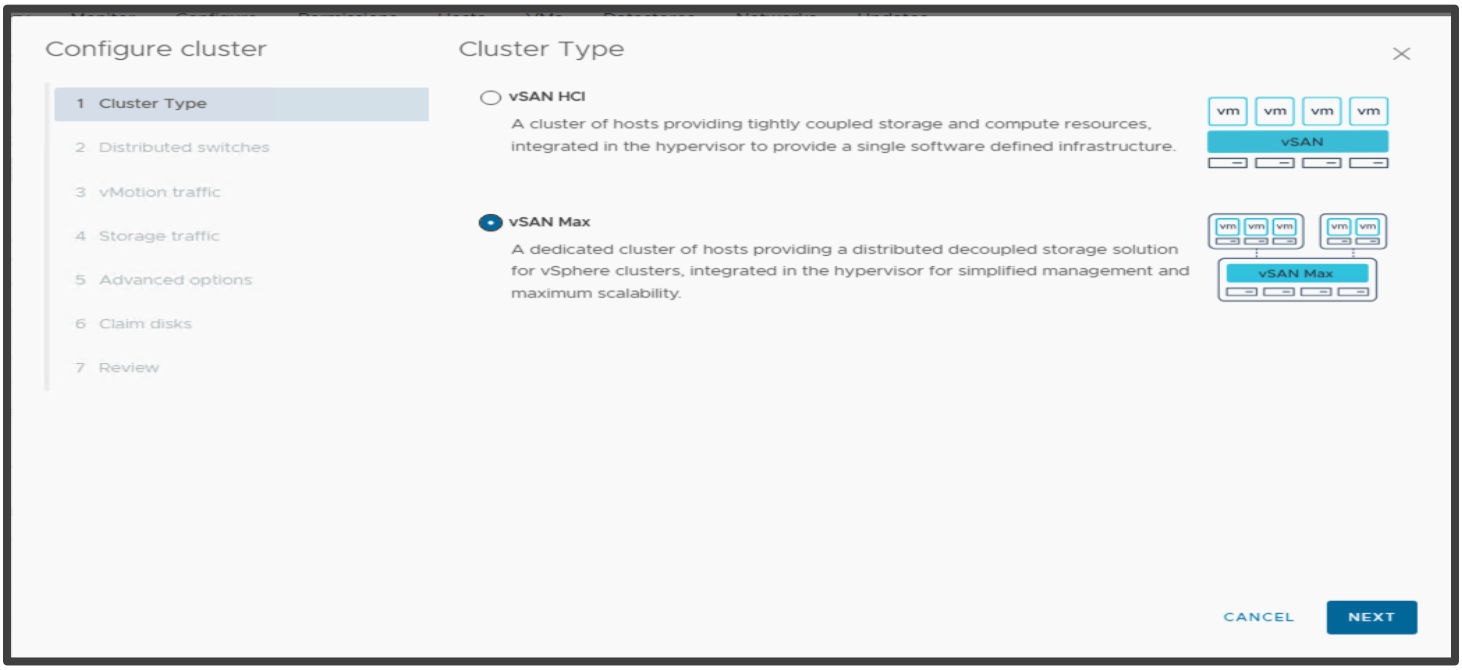

### Quickstart – Add Hosts

The Adding Hosts steps for vSAN Max are identical to those for a single vSAN HCI cluster. Refer to the steps documented in the Enable a Single vSAN HCI, Quickstart – Add Hosts section of the vSAN Proof of Concept: vSAN Architecture Overview & Setup Guide.

### Quickstart – Configure Cluster

The next step is to configure the vSAN Max cluster. After clicking on **Configure** under Step 3: Configure Cluster, the Configure Cluster workflow will start. Ensure that **vSAN Max** is selected.

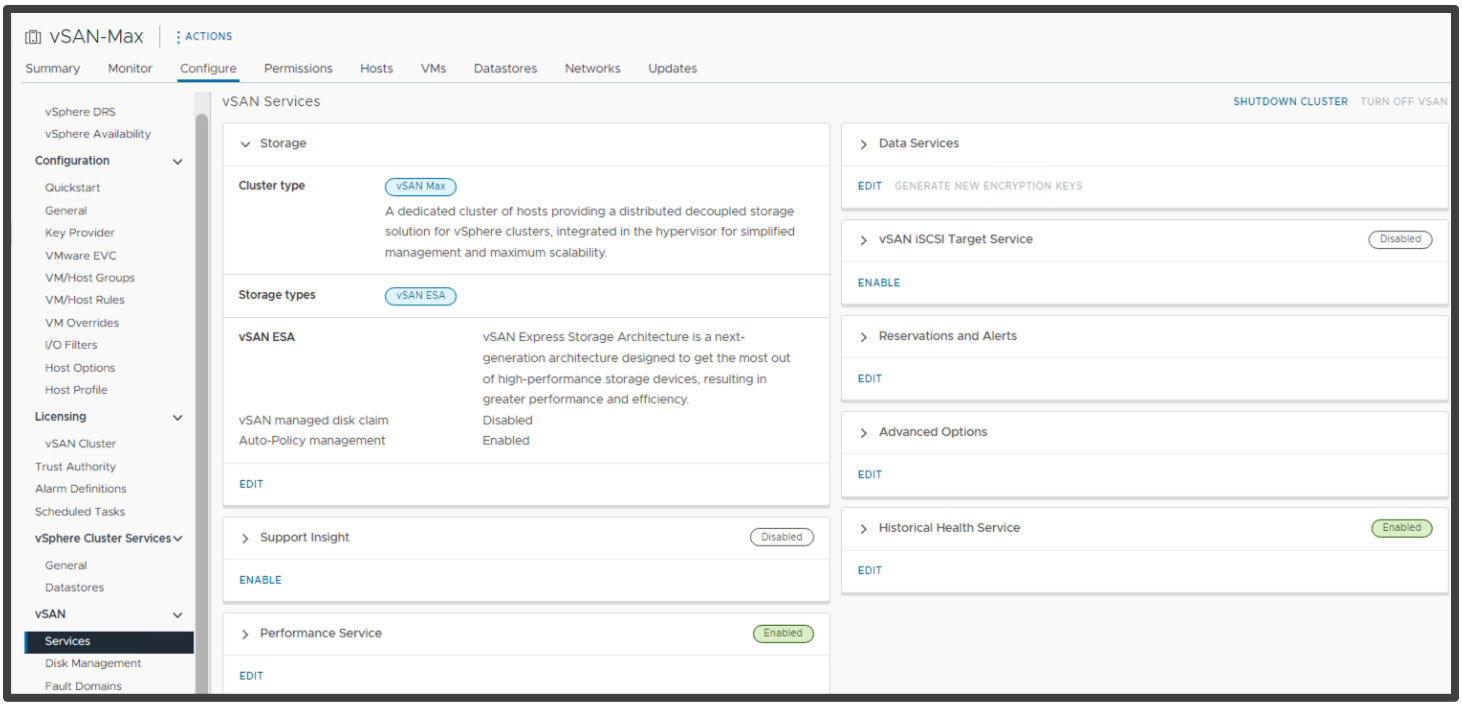
*Step 1: Select Cluster Type*



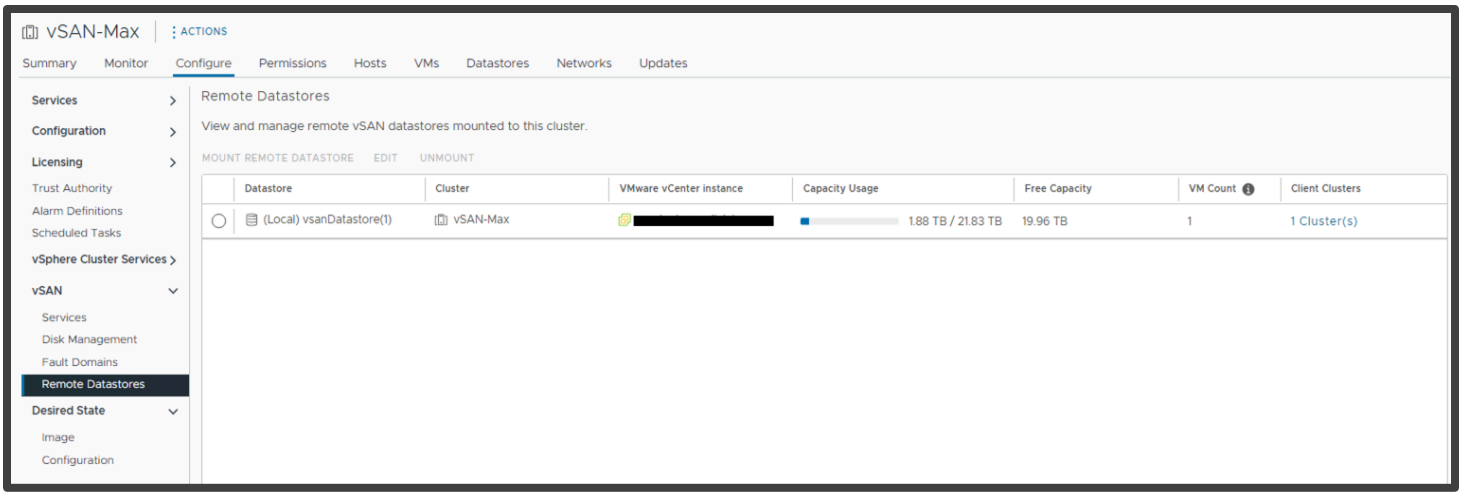

*Steps 2 -7: Configuring the Cluster*

The remaining steps to configure the vSAN Max cluster are identical to those for a single vSAN HCI cluster. Refer to the steps documented in the Enable a Single vSAN HCI, Quickstart – Configure Cluster section of section of the vSAN Proof of Concept: vSAN Architecture Overview & Setup Guide.

After the new vSAN Max cluster creation completes, navigate to **[vSAN Cluster] > Configure > vSAN > Services**. The screen will show that the vSAN Max cluster is ready to provide disaggregated storage to vSAN Computer clusters.



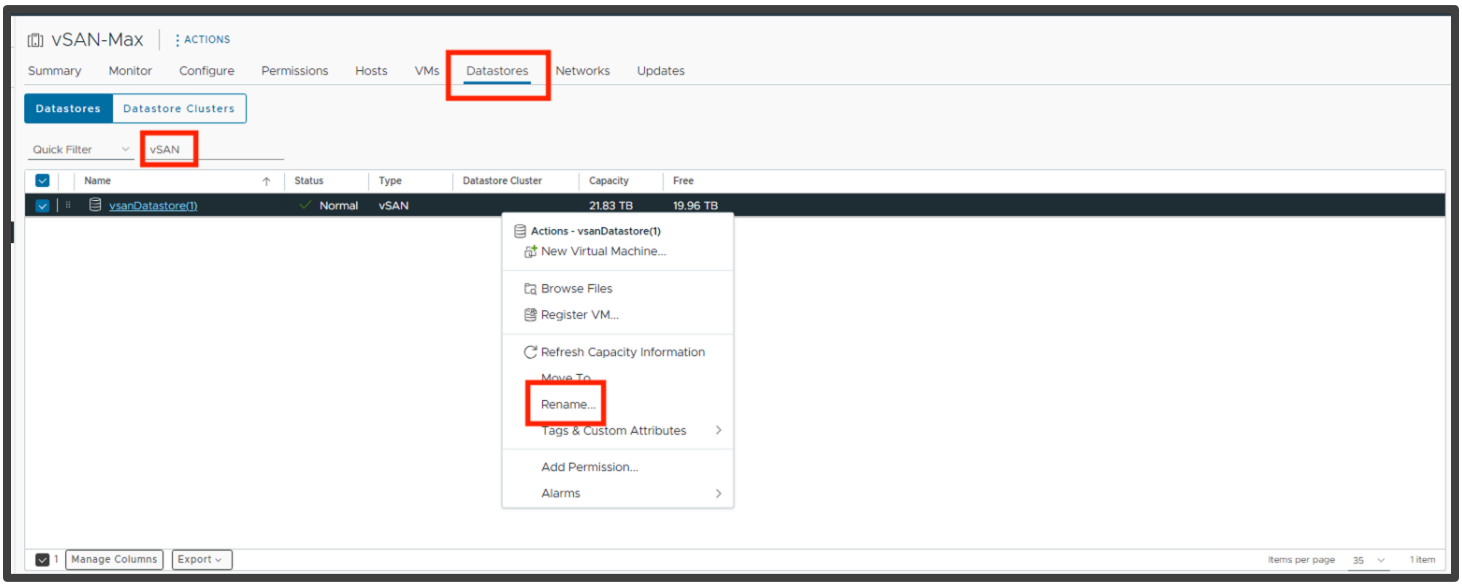

Now navigate to **[vSAN Cluster] > Configure > vSAN > Remote Datastore**. This screen shows the name of the remote datastore created by the vSAN Max cluster configuration workflow. The datastore name is a default name. If you wish to rename this datastore please refer to the Post-Configuration – Renaming vSAN Datastore section of this document.



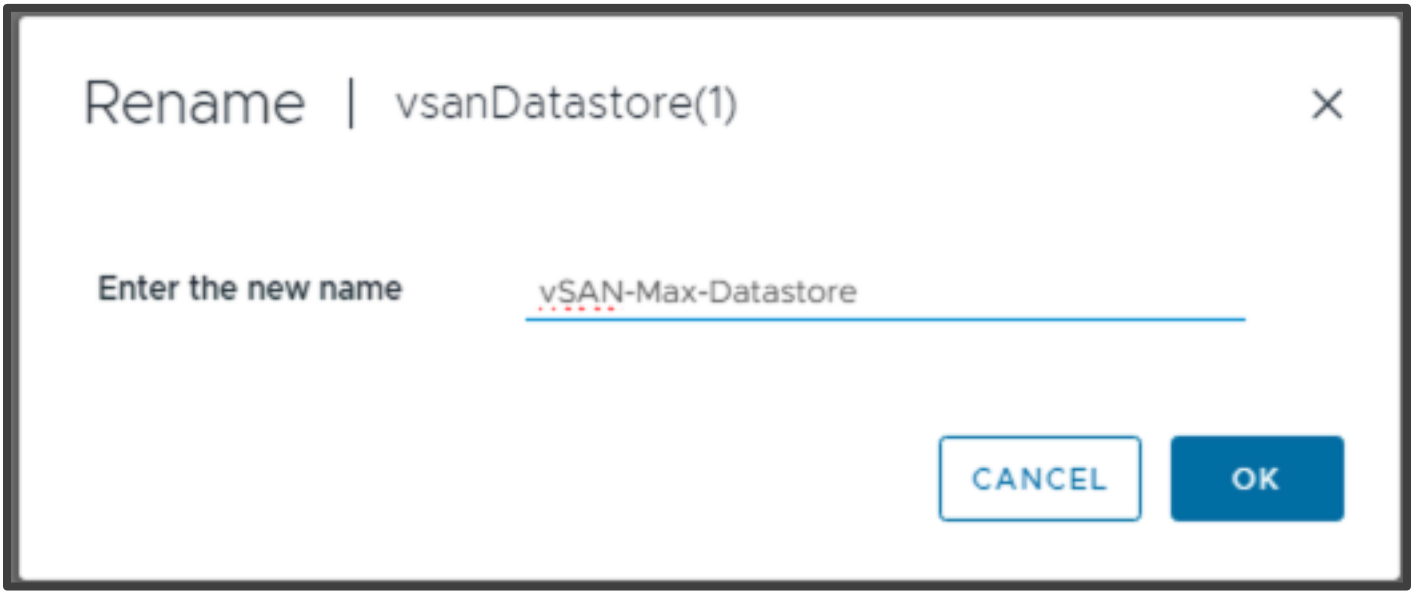
### Post-Configuration – Renaming vSAN Datastore (optional)

Once the vSAN Max cluster creation completes, the vSAN Max datastore is ready to be shared with vSAN Compute Clusters. The datastore will have the default name of "vsanDatastore." If the default name is not suitable for your environment, use these steps to rename the datastore as needed.

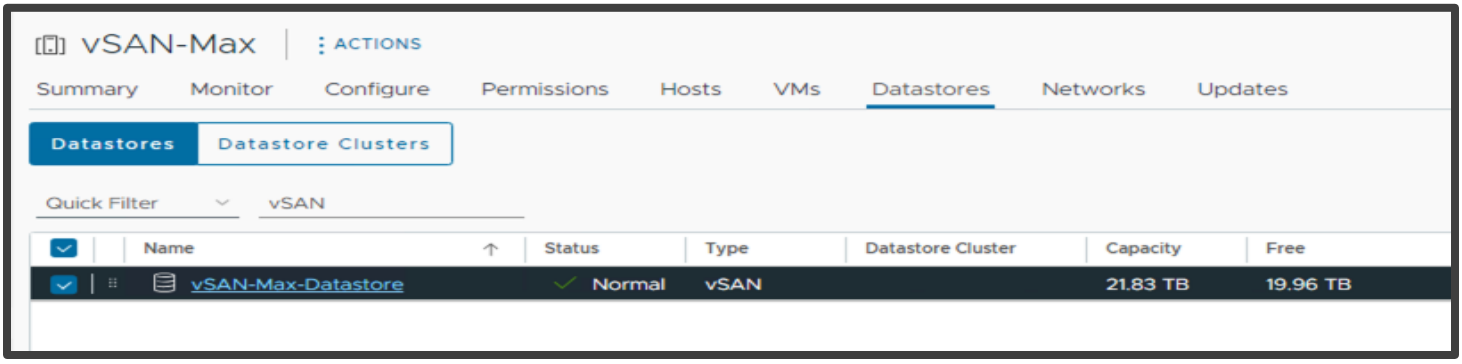
Navigate to **[vSAN Cluster] > Datastores**. Once on that screen, filter on vSAN (to make it easier to find the new datastore otherwise one may see the local datastore for each cluster host as well). Then right-click on the **vSAN datastore** and select **Rename**.




This will open the Rename workflow. In the workflow rename the datastore as needed then select **OK**



The datastore will now reflect the newly assigned name.




## Manually Enabling vSAN Max on a Cluster

**Note: If Quickstart was used (as per the earlier section) then this section can be skipped.**

Manual vSAN Max enablement is available for those that do not wish to use the Quickstart process.

For this scenario, please follow the vSAN Max Cluster Services Configuration instructions in the vSAN Max Design and Operational Guidance document. Direct link to the section listed below:

https://core.vmware.com/resource/vsan-max-design-and-operational-guidance – sec32263-sub1

## Enabling vSAN Max/HCI Mesh Services on a VMware Cloud Foundation™ based Cluster

VCF includes dedicated processes to automate the deployment and configuration of core infrastructure including vSAN services. In fact, these processes are required and are the only supported methods within VCF.

As of the writing of this guide, VCF 5.1 supports HCI Mesh. For more information, please review below.

HCI Mesh with VCF - https://docs.vmware.com/en/VMware-Cloud-Foundation/5.1/vcf-admin/GUID-1F86850D-E95E-40A8-AFC5-BE58D504D739.html

## Encryption in vSAN

There are two (mutually exclusive) modes of encryption available with vSAN, namely:

- Data-at-Rest encryption - Encrypts data on the configured physical devices within the vSAN cluster
- Data-in-Transit encryption
    - Encrypts data as it moves across the network between hosts in the vSAN cluster
    - When you enable data-in-transit encryption, vSAN encrypts all data and metadata traffic between hosts

Either encryption is usable alongside all other vSAN features (such as deduplication and compression, RAID-5/6 erasure coding, and stretched cluster configurations among others). Additionally, all vSphere features, such as VMware vSphere® vMotion® (vMotion), VMware vSphere® Distributed Resource Scheduler™ (DRS), VMware vSphere® High Availability (HA), and VMware vSphere® Replication™ are supported.

Note: Although not covered in this guide, one can also encrypt at the virtual machine level via vSphere Virtual Machine Encryption. For more information please review:

https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-security/GUID-8D7D09AC-8579-4A33-9449-8E8BA49A3003.html

### vSAN Data-at-Rest Encryption

vSAN can encrypt data at rest in your vSAN datastore. Data is encrypted after all other processing, such as deduplication, is performed. Data at rest encryption protects data on storage devices in case a device is removed from the cluster.

Be aware that:

Self-encrypted drives are **not required**.

- vSAN OSA
    - Data is encrypted when it is written to persistent media and the encryption step occurs just before the write to the capacity device
    - Data-at-Rest encryption can be enabled in an existing cluster in vSAN OSA
    - If there is enough space in the cluster, data is evacuated from each device, which are then, in turn, formatted
- vSAN ESA
    - The encryption step is higher in the stack (compared to OSA), just after the guest write
    - Once Data-at-Rest encryption is enabled it cannot be disabled

#### Key Management Server

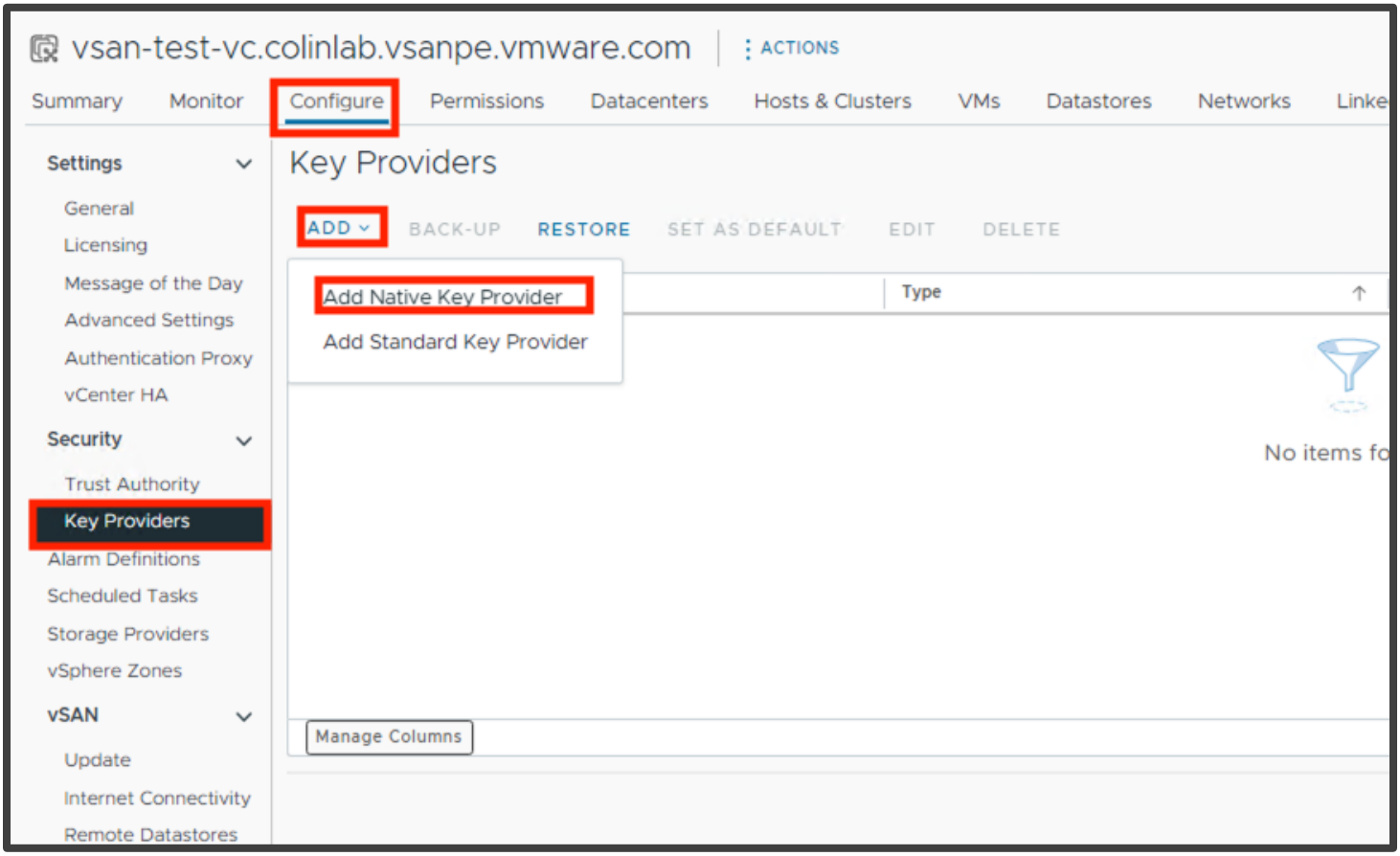
A Key Management Server (KMS) is required to enable and use data-at-rest encryption, whether vSAN ESA or OSA. Either a native (built-in) key provider in vCenter or a third-party KMS solution can be used. Third-party KMS are commonly deployed in clusters of hardware appliances or virtual appliances for redundancy and high availability.

VMware maintains a HCL for KMS servers here:
https://www.vmware.com/resources/compatibility/search.php?deviceCategory=kms

*Internal Key Management Server*

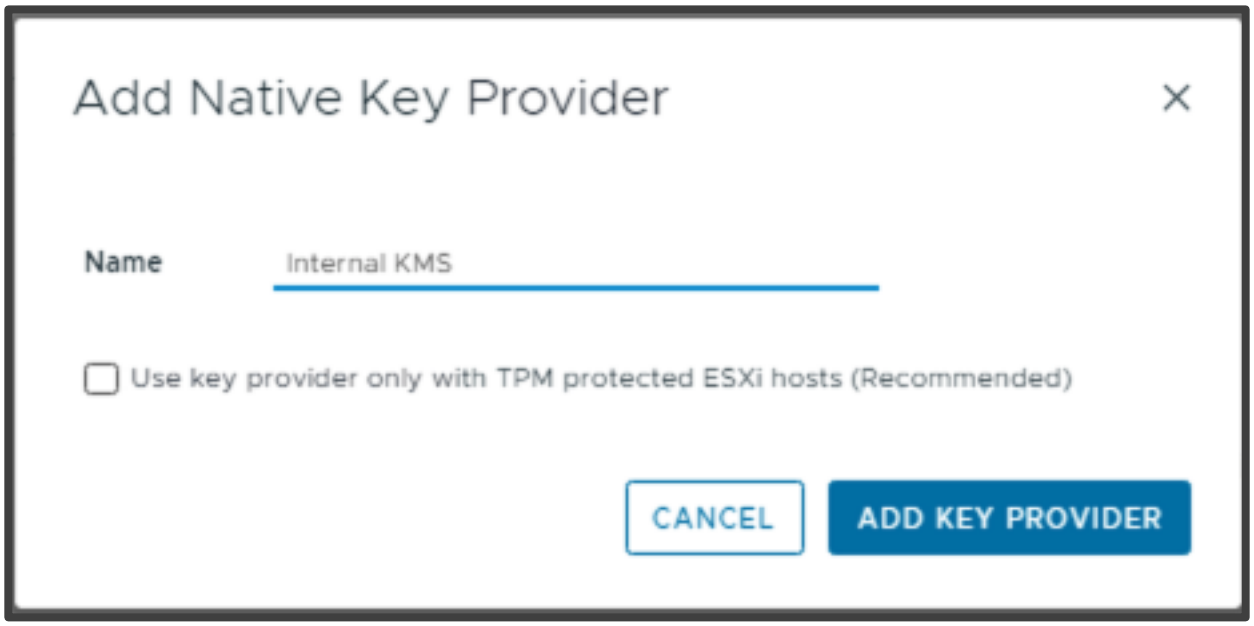
Integrated with vCenter, a built-in 'native' key provider is available to use, providing basic key functionality.

To enable, navigate to **[Top Level-vCenter Server] > Configure > Key Providers > Add > Add Native Key Provider**:

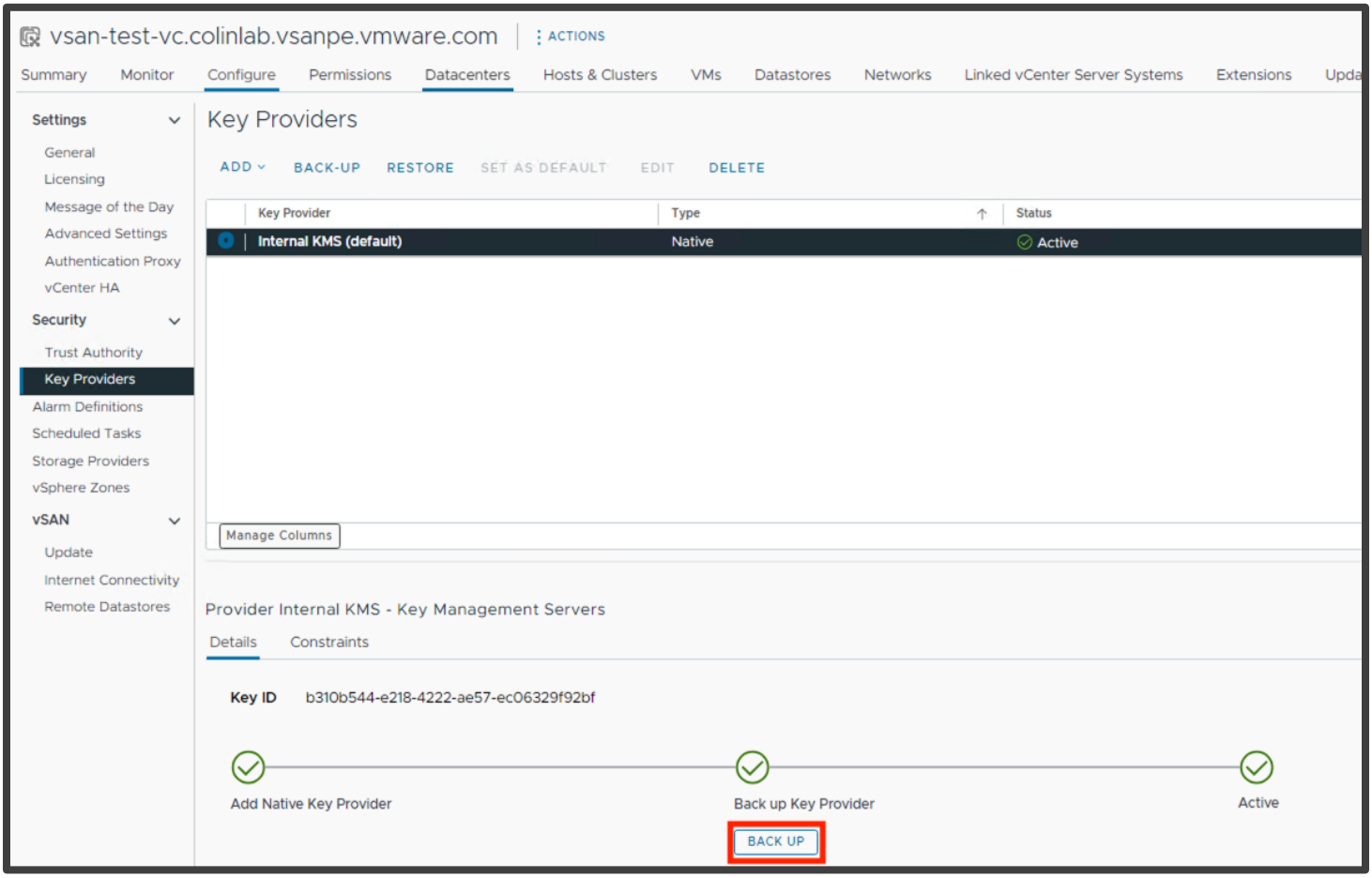


During the naming of the Key Provider, you are presented with the recommended option to only “Use the key provider with TPM protected VMware ESXi™ (ESXi) Hosts.” Note that it is advisable to use a TPM protected host. That said, If the checkbox is selected, this Key Provider will not work on non-TPM protected hosts. In that scenario, the workflow to enable data-at-rest encryption will fail (discussed later in this section). For the purposes of this walkthrough, in the Add Native Key Provider, the TPM selection checkbox is unchecked.

For more details visit:
https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-security/GUID-31F2B3D0-259A-4EFD-B675-F89BC27ACE55.html



To function, the native key provider must be backed up (a file will be downloaded locally):



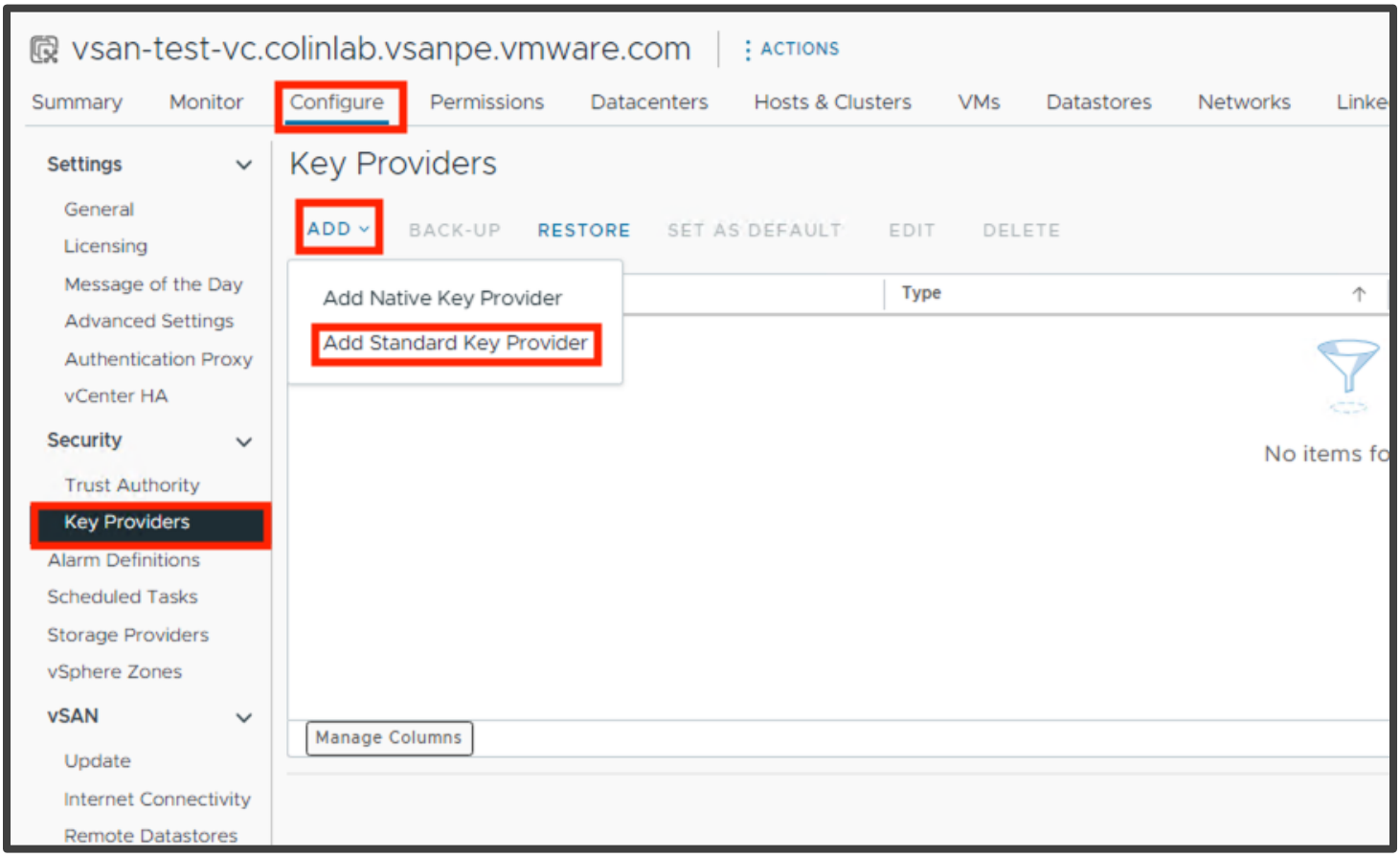

When enabling vSAN Encryption, the added Native Key Provider will be available to select. You can add both the internal and external KMS servers to vCenter and use them in different clusters.

Utilizing the Native Key Provider is a quick and easy way to test the vSAN Encryption services. For more advanced functionality, such as redundancy across KMS servers (possibly located in different locations) you may want to consider utilizing external KMS in a cluster configuration.

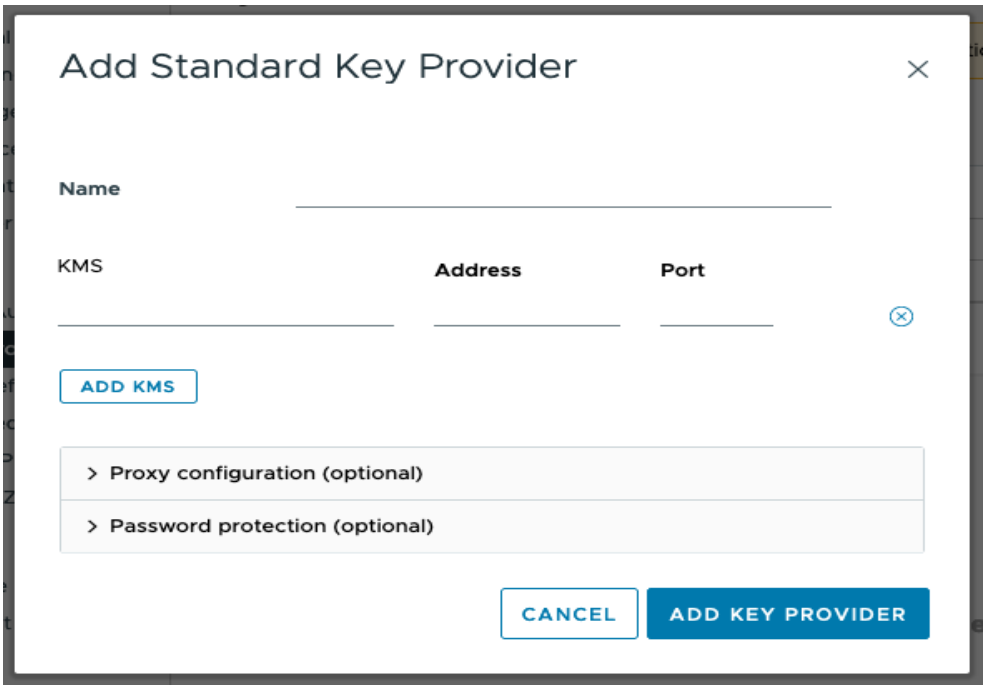
### *External Key Management Server*

Given the multitude of Key Management Server (KMS) vendors, the setup and configuration of a KMS server/cluster is out of scope for this document. However, it is a prerequisite prior to enabling vSAN encryption. The initial configuration of the KMS server is done through vCenter; the KMS cluster is, and a trust relationship is established. The process for doing this is vendor-specific, so please consult your KMS vendor documentation prior to adding the KMS cluster to vCenter.

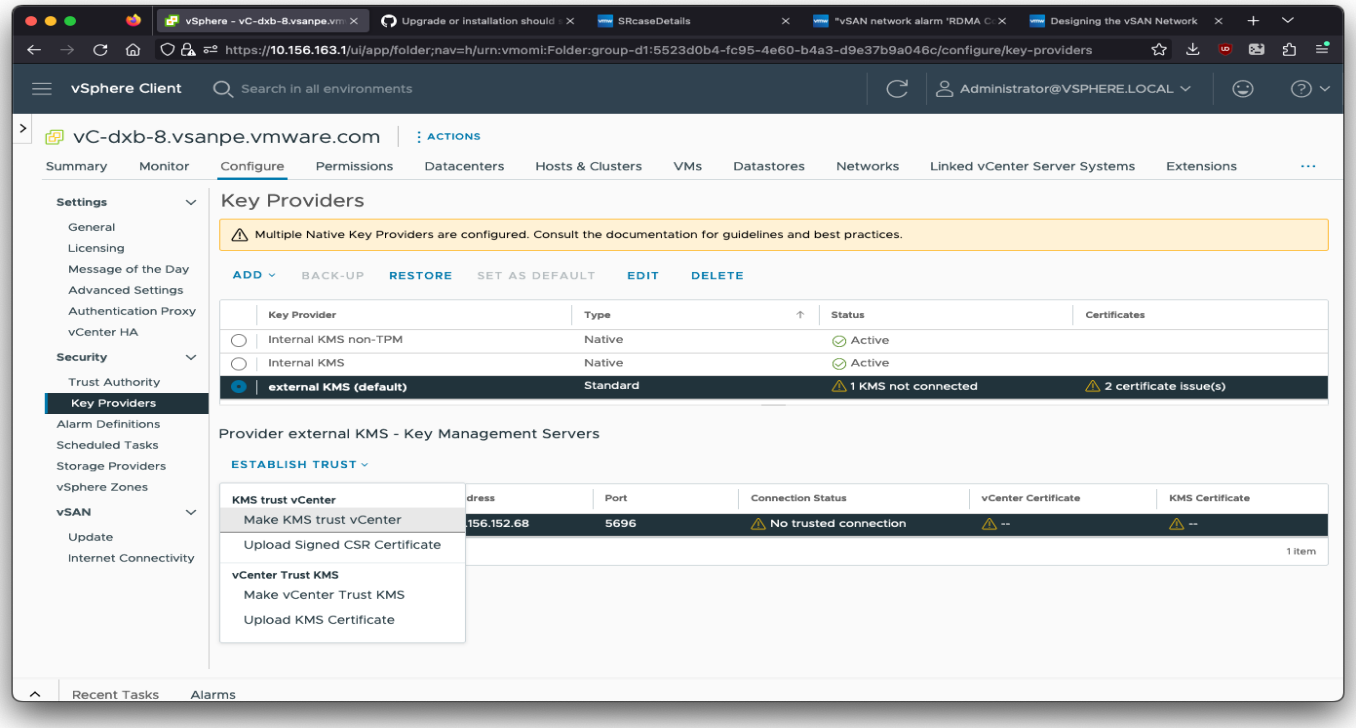
To add an external KMS cluster to vCenter, like above, navigate to **[Top Level-vCenter Server] > Configure > Key Providers > Add > Add Standard Key Provider**.



Enter the information for your specific KMS cluster/server:




Once the KMS cluster/server has been added, you will need to establish trust with the KMS server. Follow the instructions from your KMS vendor as they differ from vendor to vendor.



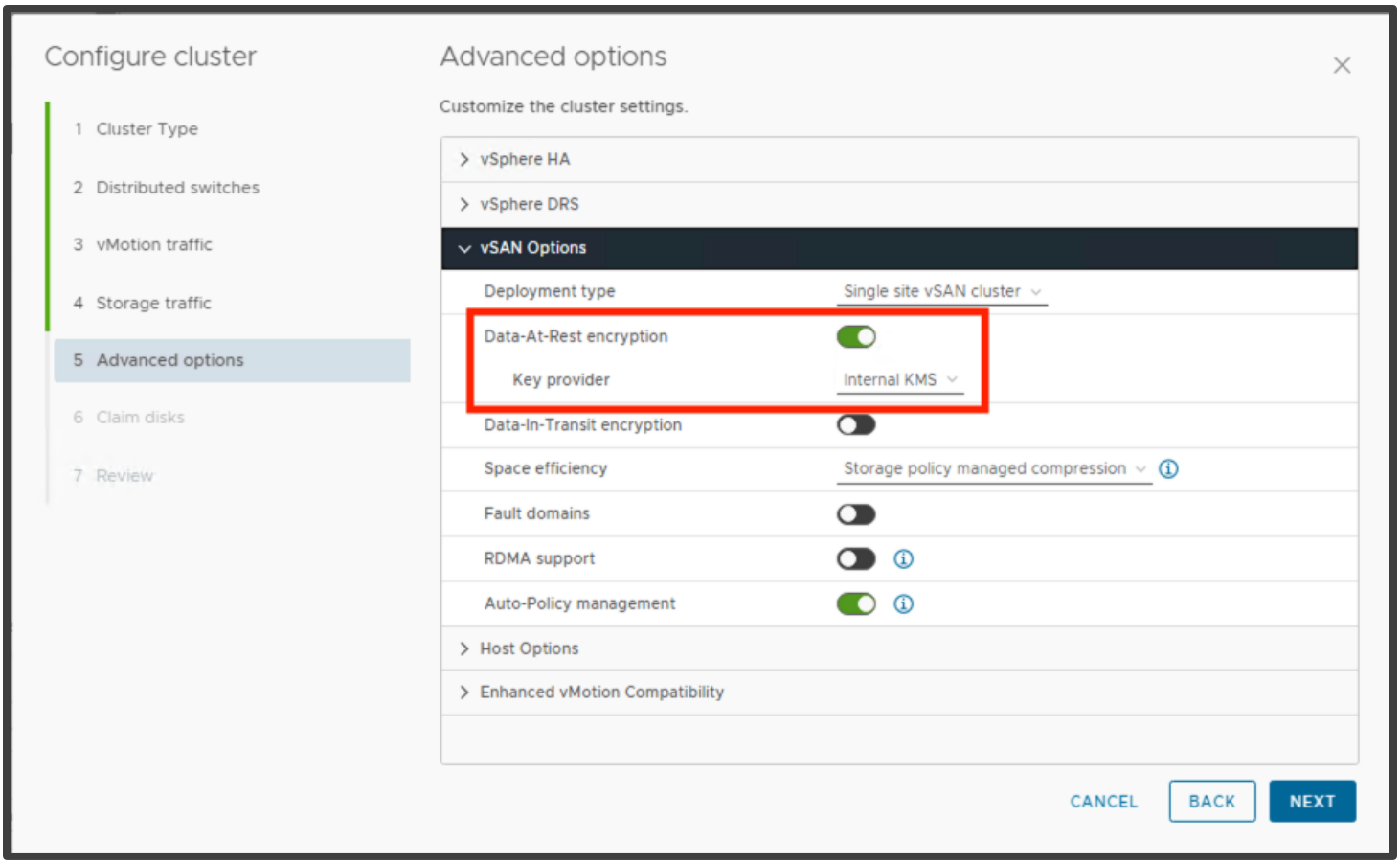
After the KMS has been properly configured, you will see that the connection status and the certificate have green checks, meaning we are ready to move forward with enabling vSAN encryption.

### Enabling vSAN Data-at-Rest Encryption - vSAN ESA

Prior to enabling vSAN ESA encryption, a KMS must have been deployed (and trusted by vCenter). Review the Key Management Server section for more information.

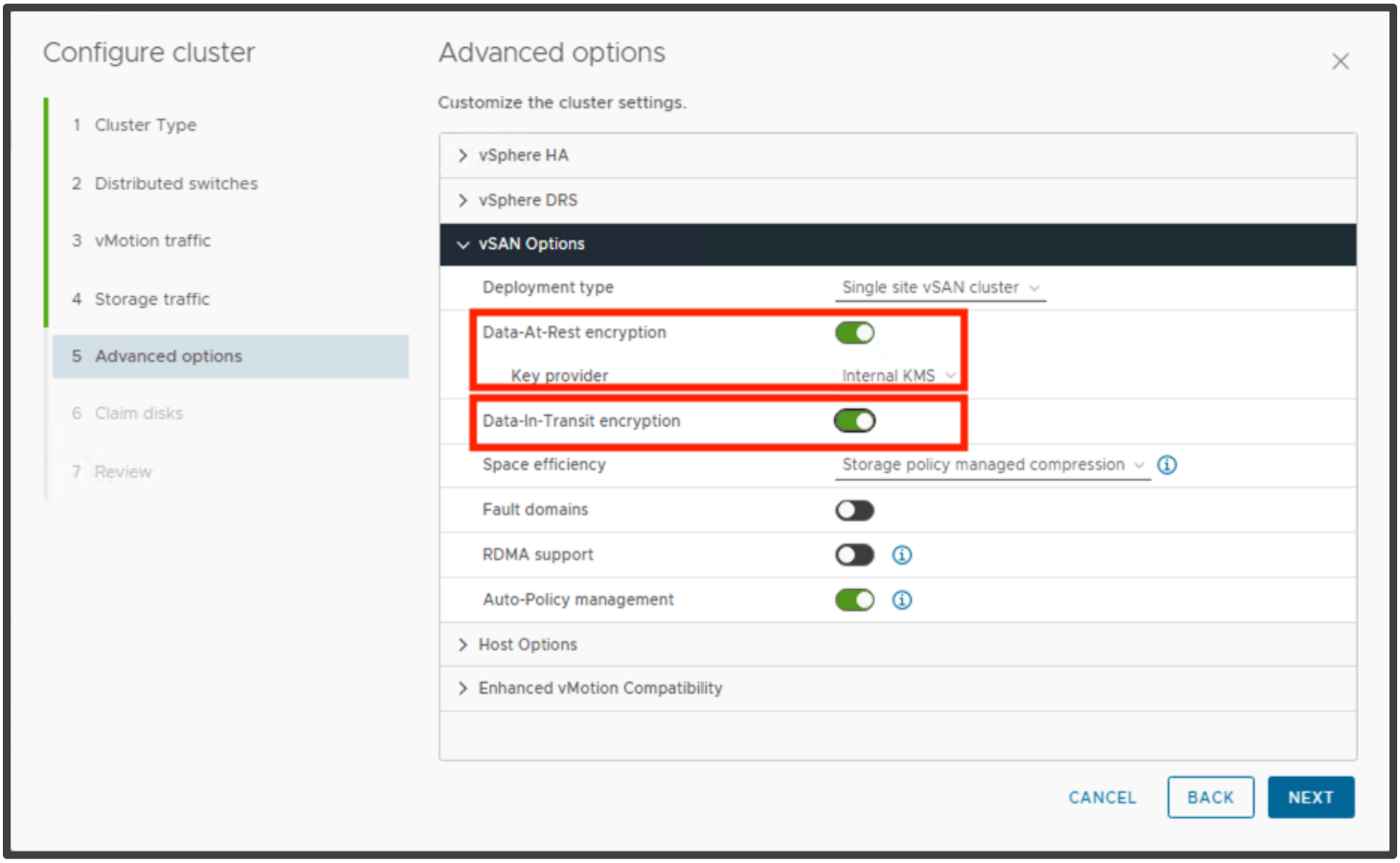
For a complete discussion of vSAN ESA cluster creation is available in the "vSAN Proof of Concept: vSAN Architecture Overview & Setup" guide.

Example of the Configure Cluster: Advanced Options screen:




Also note that Data-in-Transit encryption can be set in parallel to Data-at-Rest during initial vSAN ESA cluster creation. More details on Data-in-Transit encryption are available in the vSAN ESA Data-in-Transit Encryption section.

Example of the Configure Cluster: Advanced Options screen:




### Enabling vSAN Data-at-Rest Encryption - vSAN OSA

Prior to enabling vSAN OSA encryption, a KMS must have been deployed (and trusted by vCenter). Review the Key Management Server section for more information.

There are two options for enabling Data-at-Rest encryption:

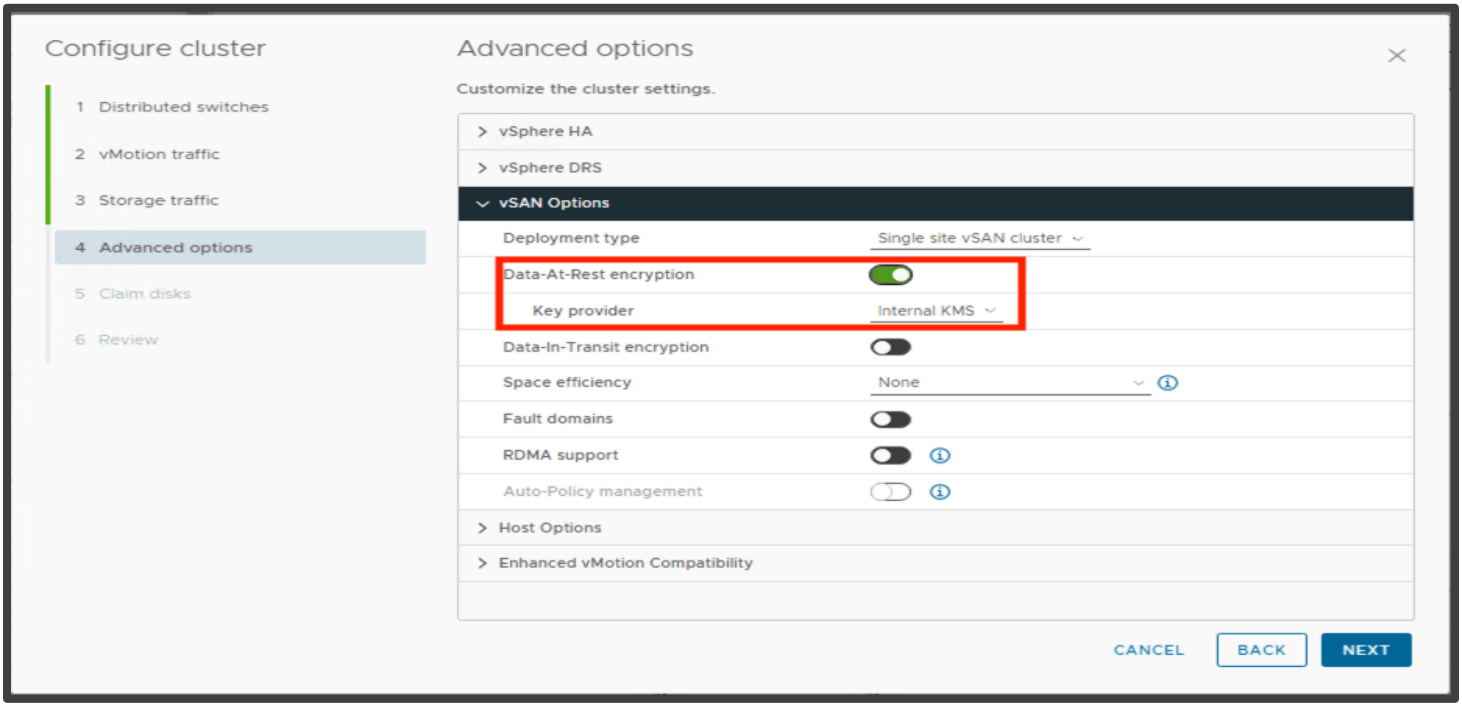
- Day 0 - During vSAN OSA cluster creation
- Day N - Post vSAN OSA cluster creation

Encrypting the vSAN OSA cluster after cluster creation can take quite some time. Especially if the cluster is in active use. The exact time varies depending on the amount of data that needs to be migrated during the rolling reformat. If you know encryption at rest is a requirement, go ahead and enable encryption during vSAN OSA cluster creation.

#### *Data-at-Rest Encryption Day 0 - During vSAN OSA Cluster Creation*

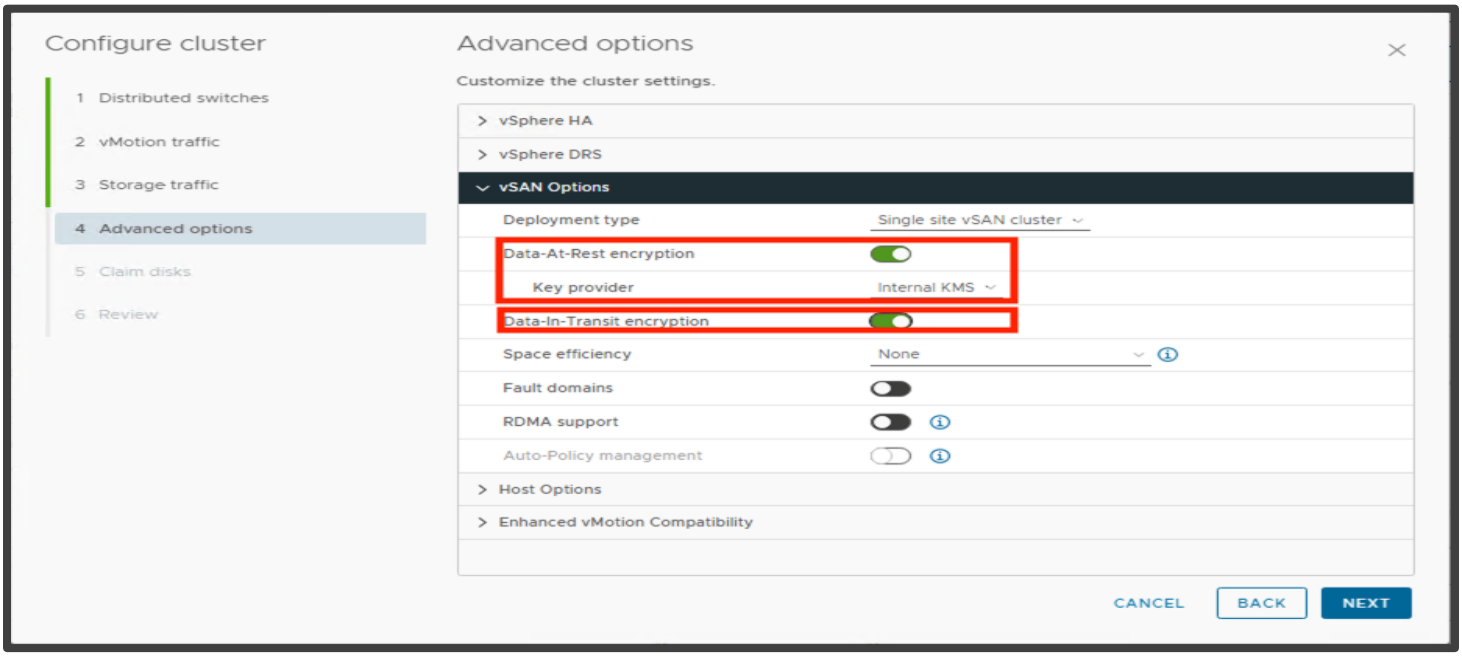
As with vSAN ESA, one can configure vSAN OSA cluster Data-at-Rest encryption at initial cluster creation. Specifically, during the Advanced Options step. A complete discussion of vSAN OSA cluster creation is available in the "vSAN Proof of Concept: vSAN Architecture Overview & Setup" guide.

Example of the Configure Cluster: Advanced Options screen:



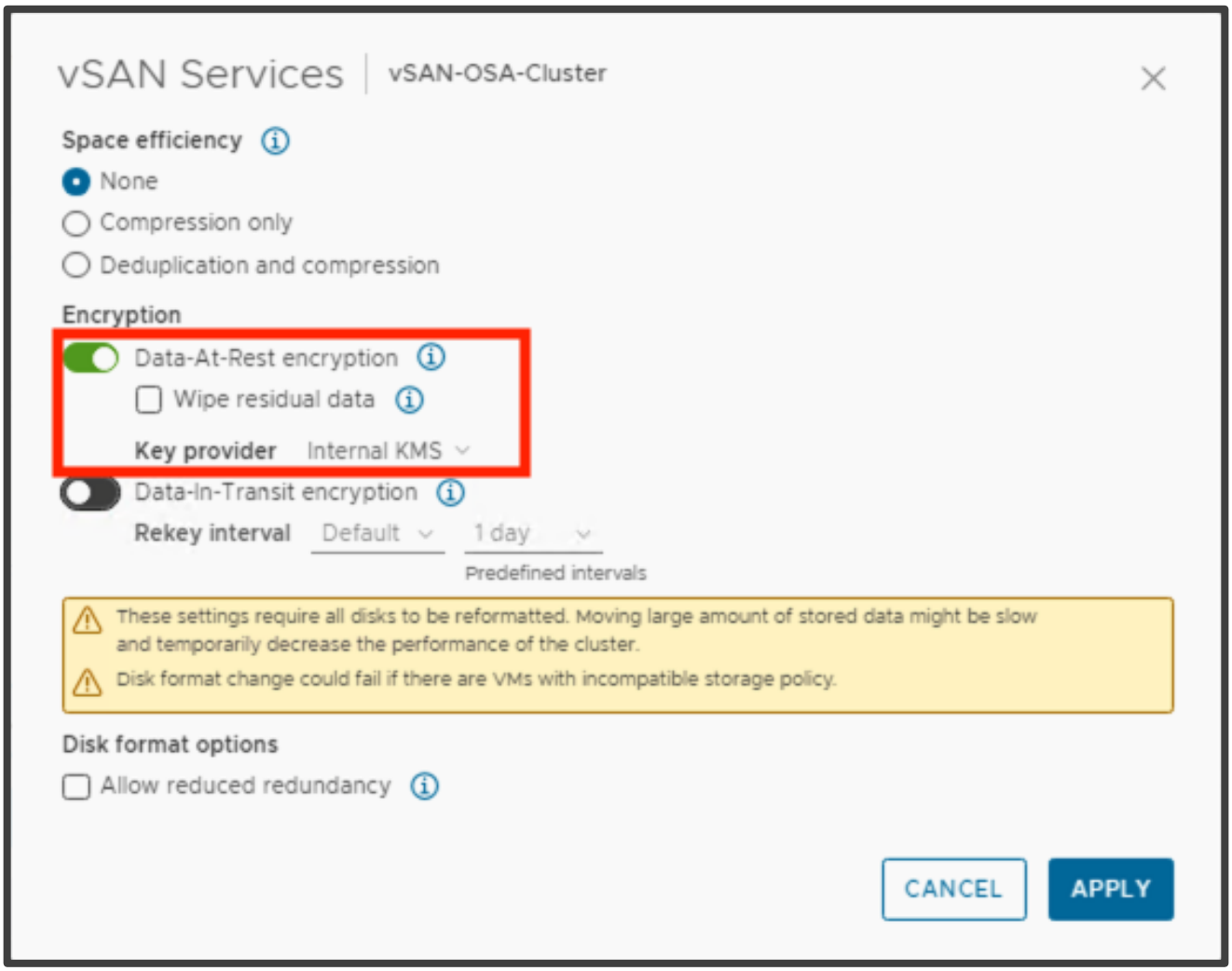

Also note that Data-in-Transit encryption can be set in parallel to Data-at-Rest during initial vSAN ESA cluster creation as well. More details on Data-in-Transit encryption are available in the vSAN OSA Data-in-Transit Encryption section.

Example of the Configure Cluster: Advanced Options screen configuring Data-at-Rest and Data-in-Transit:



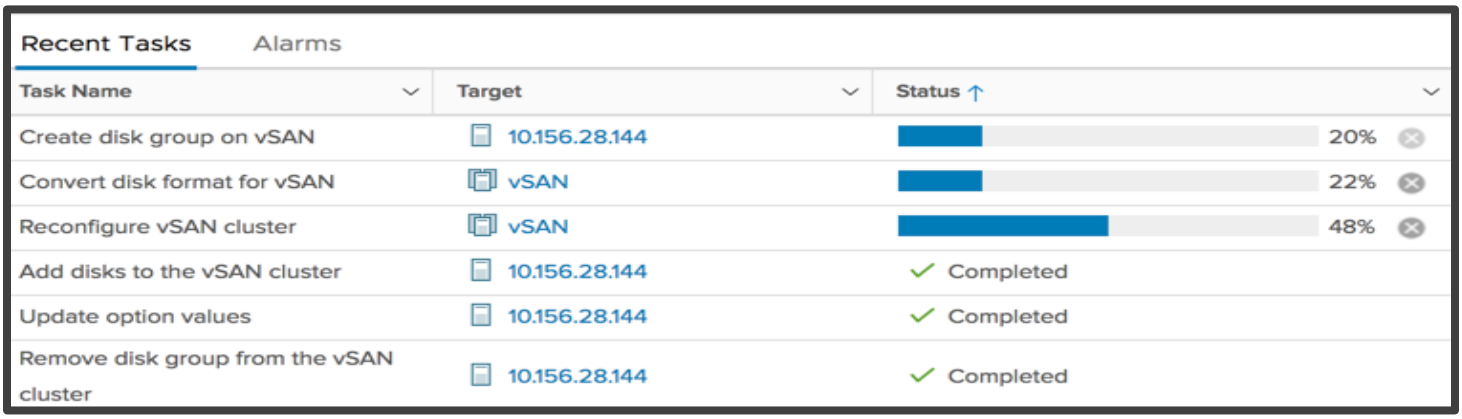
*Data-at-Rest Encryption Day N - Post vSAN OSA Creation*

To enable vSAN OSA Data-at-Rest encryption on an existing vSAN OSA cluster, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the EDIT button that corresponds to the **Data Services** section. Here we have the option to erase all disks before use (Wipe residual data). This will increase the time it will take to do the rolling format of the devices, but it will provide better protection. There is also an option to speed up the process by formatting more than one disk at a time (allow reduced redundancy).




As the example indicates, Data-in-Transit encryption can be initialized at the same time (in parallel) with Data-at-Rest or separately as required.

After you click **APPLY**, vSAN will remove one disk group at a time, format each device, and recreate the disk group once the format has completed. It will then move on to the next disk group until all disk groups are recreated, and all devices formatted and encrypted. During this period, data will be evacuated from the disk groups, so you will see components resyncing:



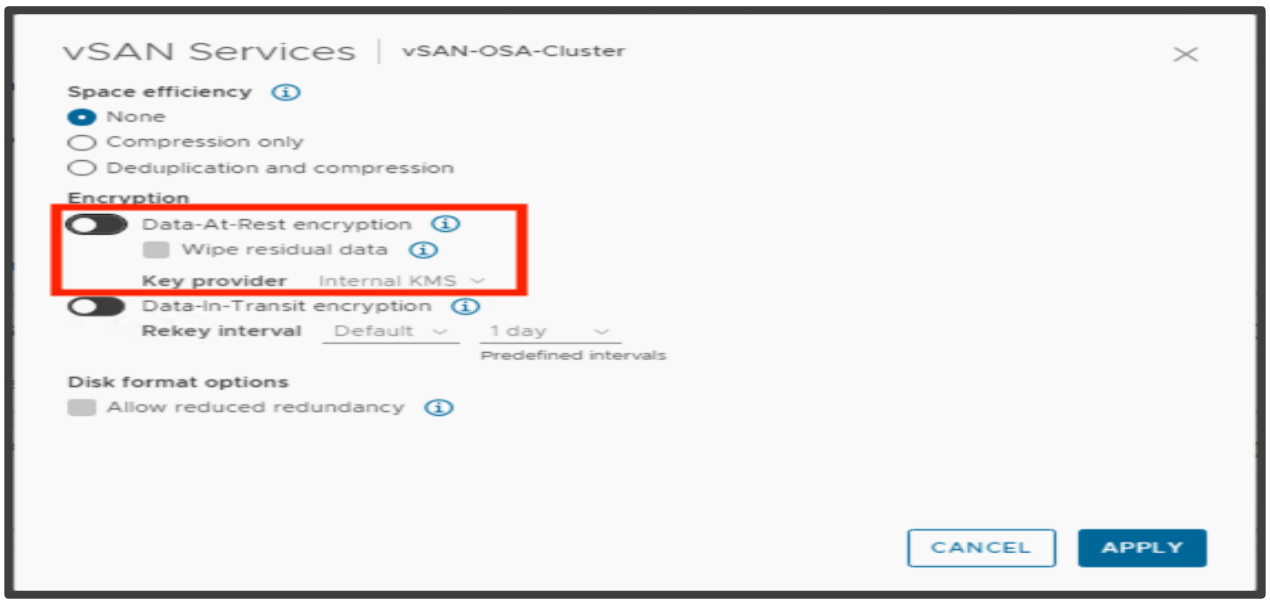

### Disabling vSAN Data-at-Rest Encryption - vSAN ESA

The only method to disable Data-at-Rest encryption on vSAN ESA clusters is a full rebuild of the cluster. For more information on cleanly removing a vSAN cluster, use the steps discussed in Appendix B.

### Disabling vSAN Data-at-Rest Encryption - vSAN OSA

Disabling vSAN OSA Data-at-Rest encryption follows a similar procedure as its enablement. Since the encryption is done at the disk group level, a disk reformat will also be conducted while disabling encryption.

To disable vSAN encryption on a vSAN OSA cluster, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **EDIT** button that corresponds to the **Data Services** section. Once in the configuration pop-up screen, simply toggle **Data-at-Rest encryption** to off, then click **Apply**.



Keep in mind that vSAN OSA will conduct a rolling reformat of the devices by evacuating the disk groups first, deleting the disk group and re-creating the disk group without encryption, at which point it will be ready to host data. The same process is conducted on all remaining disk groups until the vSAN OSA datastore is no longer encrypted.

Since the disk groups are evacuated, all data will be moved within the disk groups, so it may take a considerable amount of time depending on the amount of data present on the vSAN datastore.

### Encryption Rekey

You have the capability of generating new encryption keys. Both vSAN ESA and OSA support:

- **Shallow Rekey** - High-level rekey where the data encryption key is wrapped by a new key-encryption key
- **Deep Rekey**: - A complete re-encryption of all data (performing full data re-encryption may be slow and temporarily decrease the performance of the cluster)

To generate new keys, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **Generate New Encryption Keys** button that corresponds to the **Data Services** section.

The Generate New Encryption Keys pop-up screen will appear.

- The default, unchecking "Also encrypt all data on the storage using the new keys" initiates the shallow rekey process
- Checking "Also encrypt all data on the storage using the new keys" initiates the deep rekey process


*Note: It is not possible to specify a different KMS server when selecting to generate new keys during a deep rekey; however, this option is available during a shallow rekey.*

For more information on key rotation, see: https://core.vmware.com/blog/key-rotation-options-vsan-esa-vmware-cloud-foundation-51-and-vsan-8-u2

## vSAN Data-in-Transit Encryption

Data-in-Transit Encryption can be enabled independently or together with Data-at-Rest encryption to fully protect vSAN data. Data-in-Transit encryption uses FIPS 140-2 validated VMware VMkernel Cryptographic module. Both Data and metadata are encrypted. Unlike Data-at-Rest encryption, Data-in-Transit encryption does not require an external KMS. Keys are managed internally.

When designing Data-in-Transit encryption services for your environment, be aware that:

- vSAN uses AES-256-bit encryption on data in transit
- vSAN data-in-transit encryption is not related to data-at-rest-encryption. You can enable or disable each one separately
- Forward secrecy is enforced for vSAN data-in-transit encryption
- Traffic between data hosts and witness hosts is encrypted
- File service data traffic between the VDFS proxy and VDFS server is encrypted
- vSAN file services inter-host connections are encrypted

### Enabling Data-in-Transit Encryption - vSAN ESA

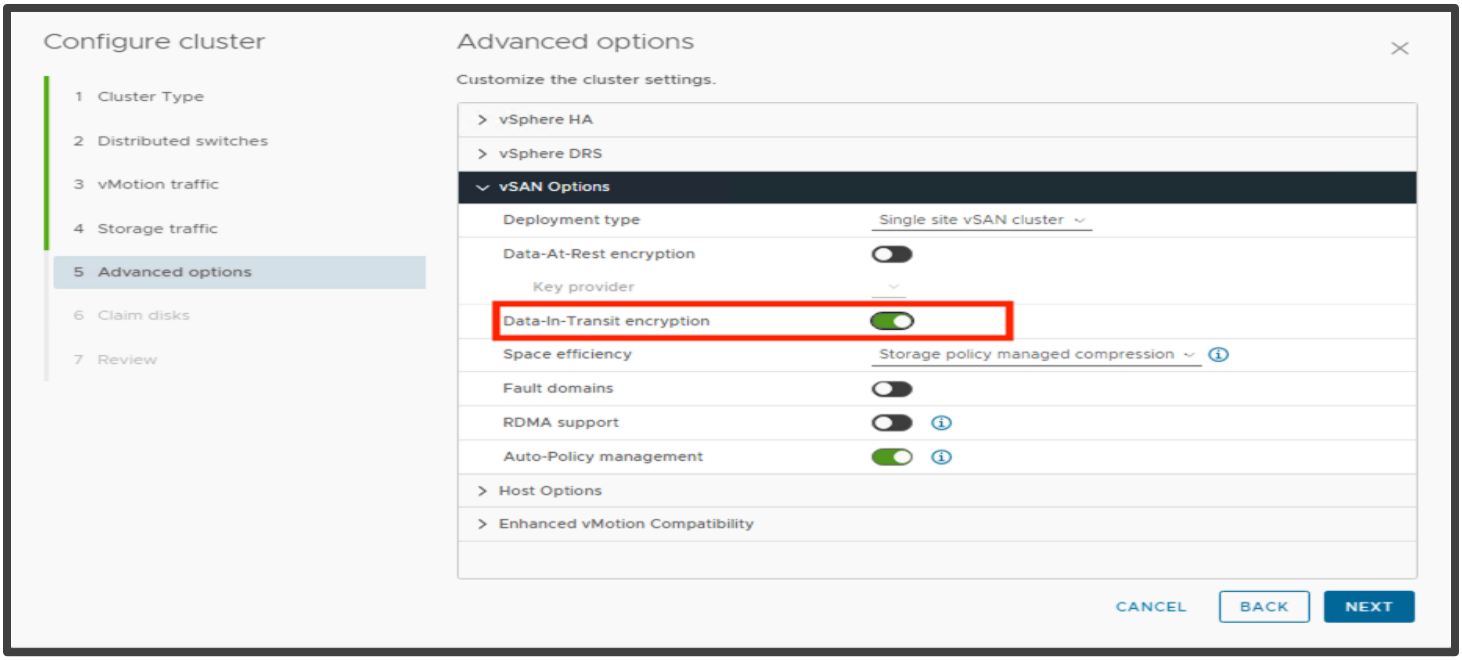
There are two options for enabling Data-in-Transit encryption:

- Day 0 - During vSAN ESA cluster creation
- Day N - Any time after vSAN ESA cluster creation

#### *Enabling Data-in-Transit Encryption Day 0 - During vSAN ESA Cluster Creation*

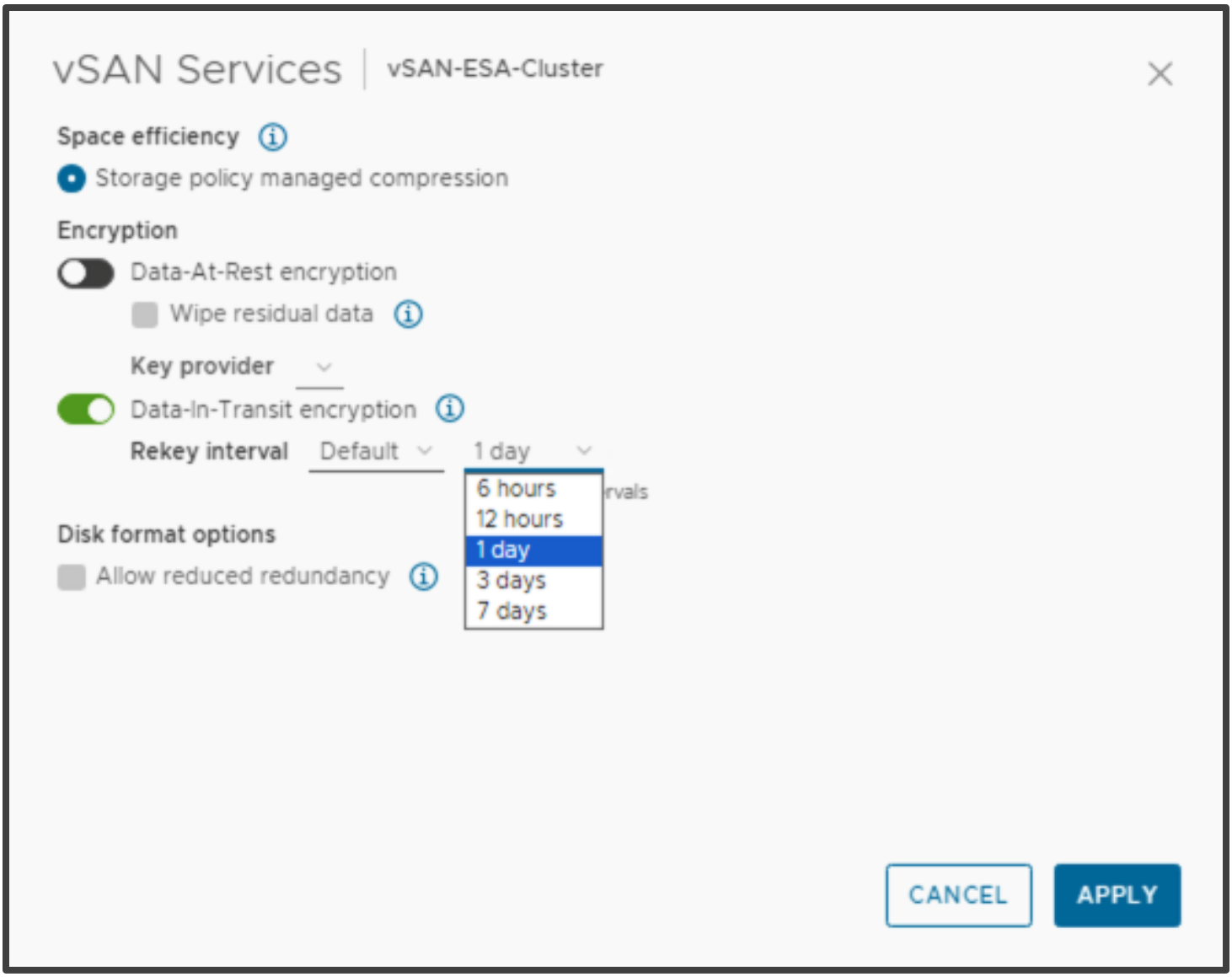
Data-in-Transit encryption can be enabled during the vSAN ESA cluster creation process. Specifically, during the Advanced Options step. A complete discussion of vSAN ESA cluster creation is available in the "vSAN Proof of Concept: vSAN Architecture Overview & Setup" guide.

Example of the Configure Cluster: Advanced Options screen with Data-in-Transit encryption enabled (Note that with vSAN ESA, Data-at-Rest encryption can be deployed in parallel during cluster creation):



*Enabling Data-in-Transit Encryption Day N - Post vSAN ESA Cluster Creation*

To enable vSAN ESA Data-at-Rest encryption on an existing vSAN ESA cluster, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **EDIT** button that corresponds to the **Data Services** section. Tick **Data-in-Transit encryption** and select your Rekey Interval. The default for rekey interval is one (1) day. Click **Apply:**




Enabling Data-in-Transit Encryption - vSAN OSA

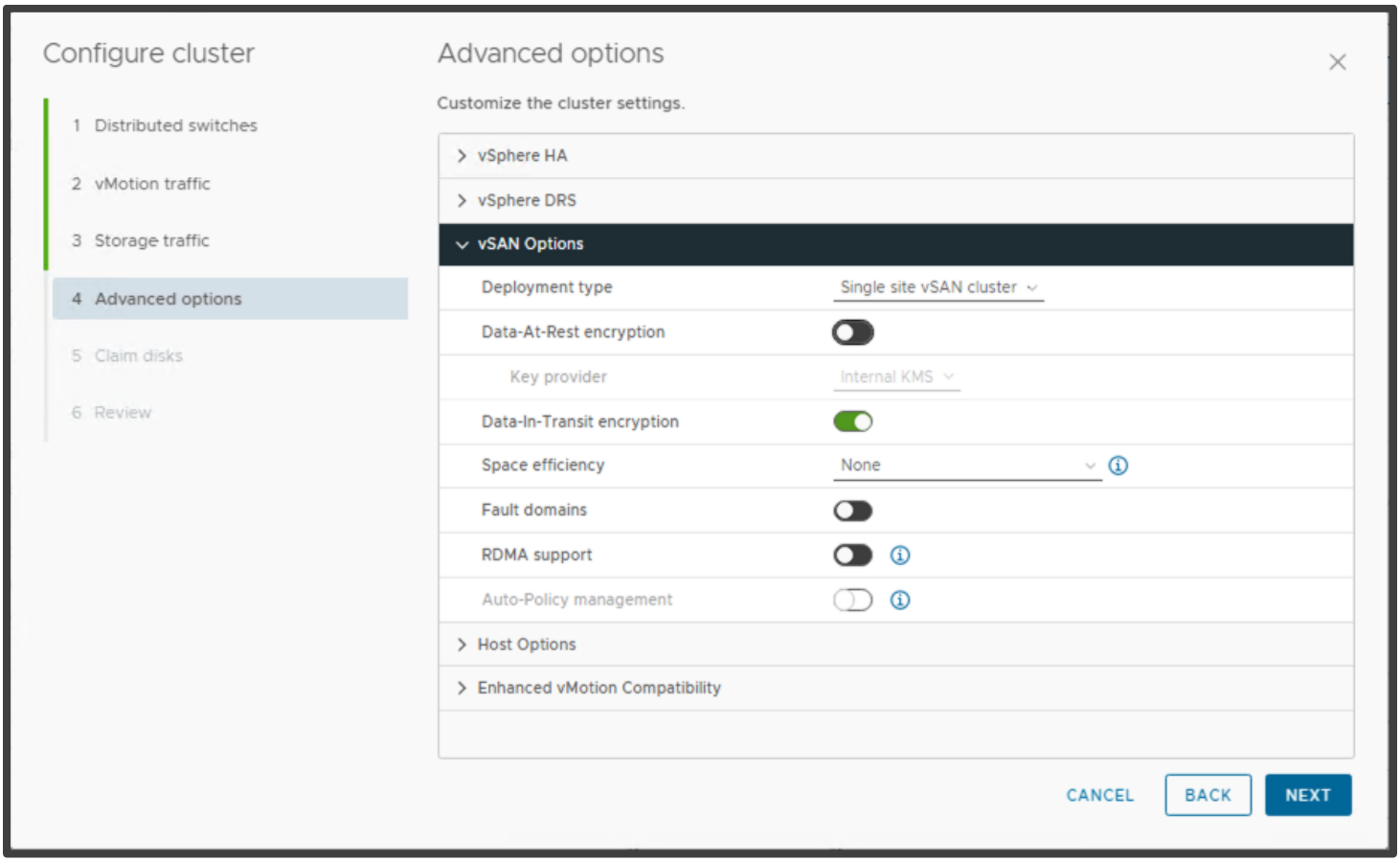
There are two options for enabling Data-in-Transit encryption:

- Day 0 - During vSAN OSA cluster creation
- Day N – Any time after vSAN OSA cluster creation

*Enabling Data-in-Transit Encryption Day 0 - During vSAN OSA Cluster Creation*

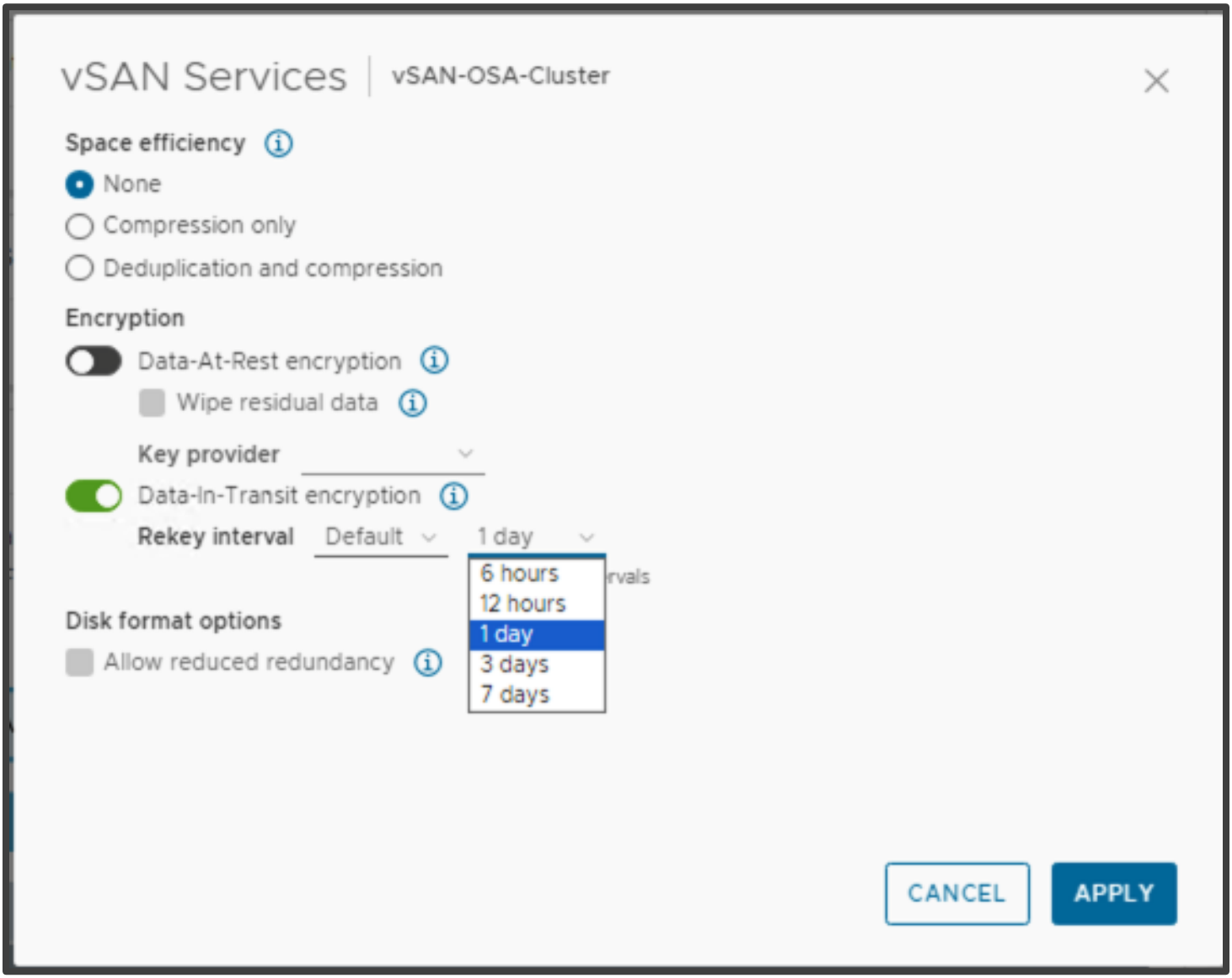
Data-in-Transit encryption can be enabled during the vSAN OSA cluster creation process. Specifically, during the Advanced Options step. A complete discussion of vSAN OSA cluster creation is available in the "vSAN Proof of Concept: vSAN Architecture Overview & Setup" guide.

Example of the Configure Cluster: Advanced Options screen with Data-in-Transit encryption enabled (Note that with vSAN OSA, Data-at-Rest encryption can be deployed in parallel both during and post cluster creation):



*Enabling Data-in-Transit Encryption Day N - Post vSAN OSA Cluster Creation*

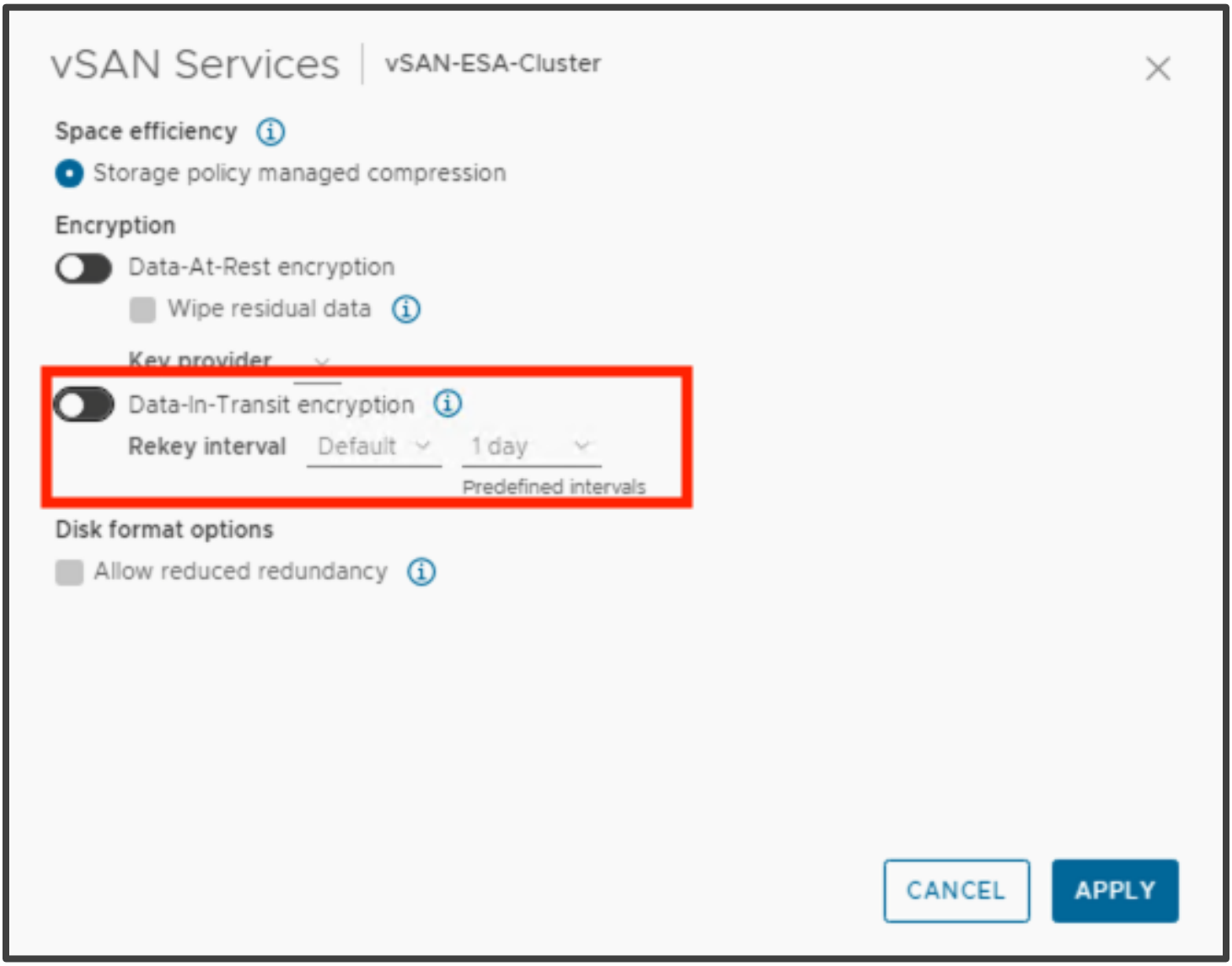
To enable vSAN OSA Data-at-Rest encryption on an existing vSAN OSA cluster, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **EDIT** button that corresponds to the **Data Services** section. Tick **Data-in-Transit encryption** and select your Rekey Interval. The default for rekey interval is one day. Click **Apply:**



### Disabling Data-in-Transit Encryption - vSAN ESA

Disabling vSAN ESA Data-in-Transit encryption follows a similar procedure as its post cluster creation enablement.

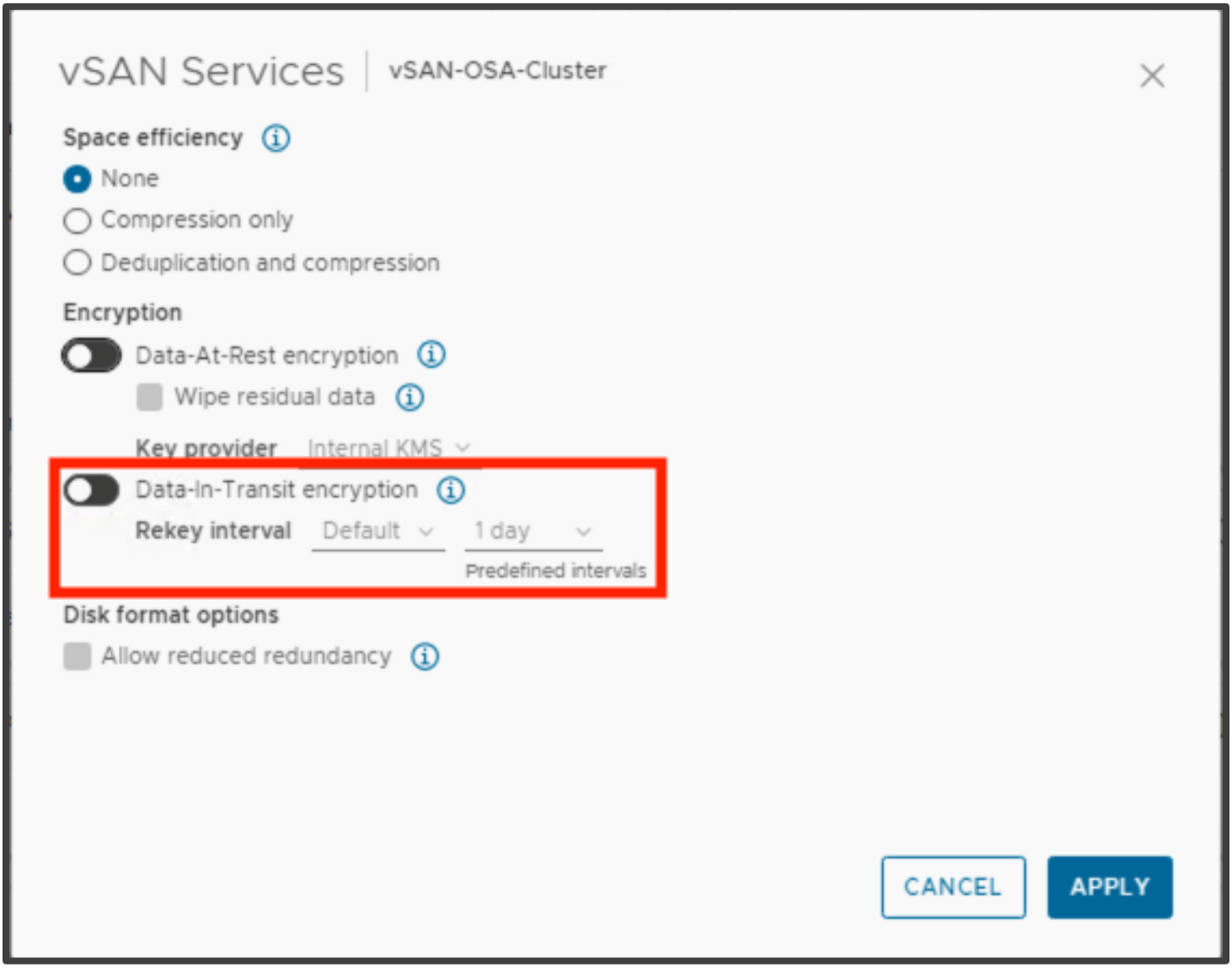
To disable vSAN encryption on a vSAN OSA cluster, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **EDIT** button that corresponds to the **Data Services** section. Once in the configuration pop-up screen, simply toggle **Data-in-Transit encryption** to off, then click **Apply**.



### Disabling Data-in-Transit Encryption - vSAN OSA

Disabling vSAN OSA Data-in-Transit encryption follows a similar procedure as its enablement.

To disable vSAN encryption on a vSAN OSA cluster, navigate to **[vSAN Cluster] > Configure > vSAN > Services > Data Services**, then click the **EDIT** button that corresponds to the **Data Services** section. Once in the configuration pop-up screen, simply toggle **Data-in-Transit encryption** to off, then click **Apply**:



## vSAN File Services (vSAN ESA and OSA)

The addition of vSAN File Service quickly enables NFS and SMB shares on vSAN without the need to install or manage a dedicated file service appliance. File shares can be presented to both VMs and containers. Moreover, the entire life cycle of provisioning and managing file services can be seamlessly performed through vCenter.

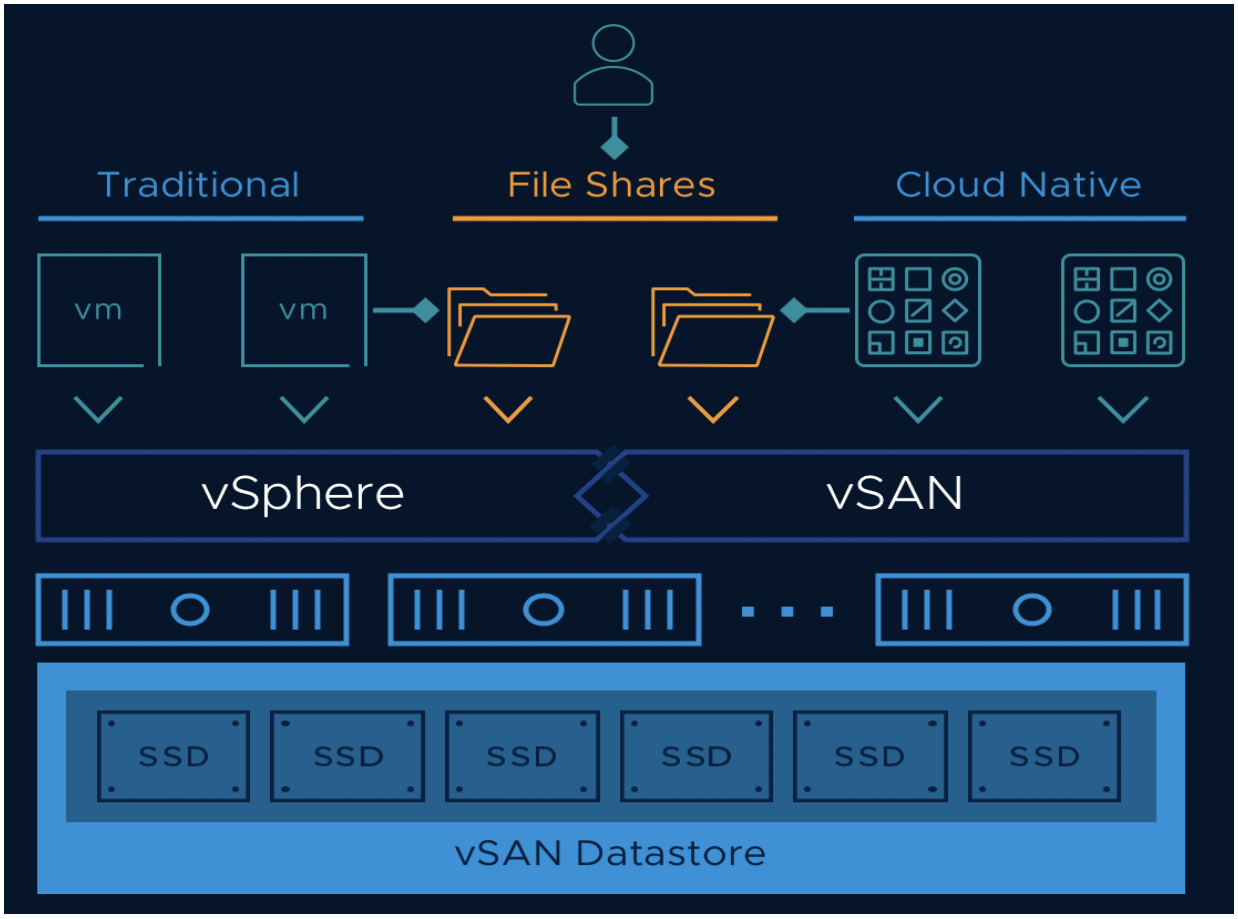
The data stored in a file share can be accessed from any device that has access rights. vSAN File Service is a layer that sits on top of vSAN to provide file shares. It currently supports:

- SMBv2.1 & SMBv3
- NFSv3 & NFSv4.1

The vSAN Distributed File System (vDFS) which provides the underlying scalable filesystem by aggregating:

- vSAN objects
- A Storage Services Platform that provides:
    - Resilient file server endpoints
    - A control plane for deployment, management, and monitoring

File shares are integrated into the existing vSAN Storage Policy Based Management on a per-share basis. vSAN file service brings in capability to host the file shares directly on the vSAN cluster.



In this section we will focus on enabling vSAN File Service, creating and mounting shares, viewing file share properties, and failure scenarios.

## Cloud Native Use Cases

File services in its first instance was designed to support Cloud-Native workloads. Cloud-Native workloads built on micro-services architecture require data access to be concurrent. Multiple micro-services read and update the same data repository at the same time from different nodes. Updates should be serialized, with no blocking, locking, or exclusivity. This approach differs from the current offering for Cloud-Native storage on vSAN. In the current model, vSAN backed VMDKs are presented to VMs and thus mounted to a single container.

For instance, web services applications like Apache, Nginx, and Tomcat require shared file access to support distributed operations. Rather than replicating this data to every instance, a single NFS share can be mounted into all containers running these workloads. Hence file storage is critical for Cloud-Native Applications.

## Considerations

- vSAN 8.0 supports two-node configurations and stretched clusters
- vSAN 8.0 supports 64 file servers in a 64-host setup
- vSAN 8.0 supports 100 file shares
- vSAN File Services does not support the following
    - Read-Only Domain Controllers (RODC) for joining domains because the RODC cannot create machine accounts (as a security best practice, a dedicated org unit should be pre-created in the Active Directory and the username mentioned here should be controlling this organization)
    - Disjoint namespace
    - Spaces in organizational units (OUs) names
    - Multi domain and Single Active Directory Forest environments
- When a host enters maintenance mode
    - The file server moves to another FSVM
    - The FSVM on the host that entered maintenance mode is powered off
    - After the host exits maintenance mode, the FSVM is powered on
- vSAN File Services VM (FSVM) docker internal network may overlap with the customer network without warning or reconfiguration
    - There is a known conflict issue if the specified file service network overlaps with the docker internal network (172.17.0.0/16) causings routing problems for the traffic to the correct endpoint
    - As a workaround, specify a different file service network so that it does not overlap with the docker internal network (172.17.0.0/16)

## Pre-Requisites

Before enabling file services, you will need the following:

- An existing vSAN HCI cluster, vSAN stretched cluster, or a vSAN ROBO cluster
- vSAN ESA and OSA is supported
- A unique IP address for each file service agent (as per best practice, this will be equal to the number of hosts in the cluster)
- DNS entries (forward and reverse lookup should be working correctly)
- Network details (subnet mask, gateway, etc.)
- Dedicated distributed switch port group (for DVS)
- MAC Learning, MAC change, unknown unicast flooding and forged frames enabled on the segment (for NSX)

In addition, you will need the following information for the cluster:

- File Services Domain – A unique namespace for the cluster that will be used across shares
- DNS Servers – Multiple DNS entries can be added for redundancy
- DNS Suffix
- Active Directory domain information (for SMB shares or Kerberos authentication)

Below we show an example on how to enable and configure vSAN file services.

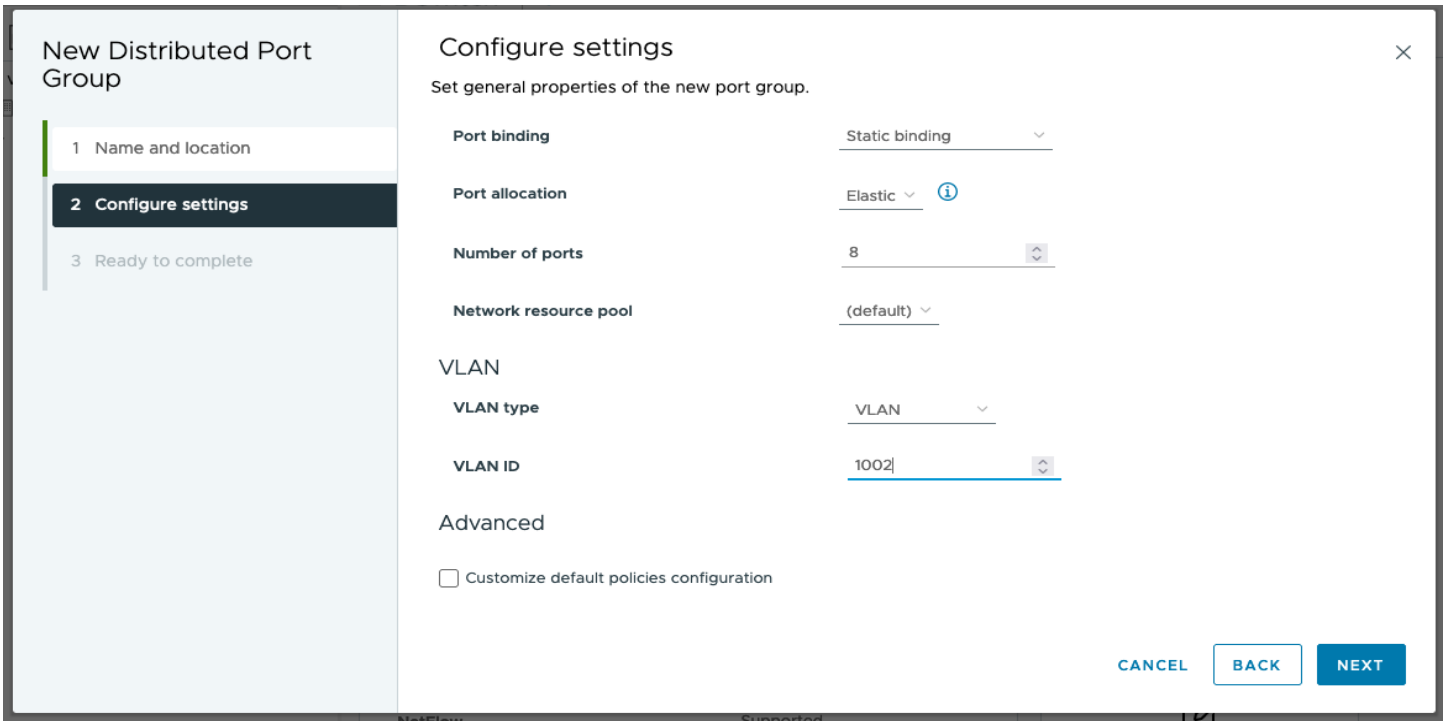
For full details, visit:
https://docs.vmware.com/en/VMware-vSphere/8.0/vsan-administration/GUID-82565B82-C911-42F7-85B1-E9EF973EE90C.html

## Enabling File Services - vSAN ESA and OSA

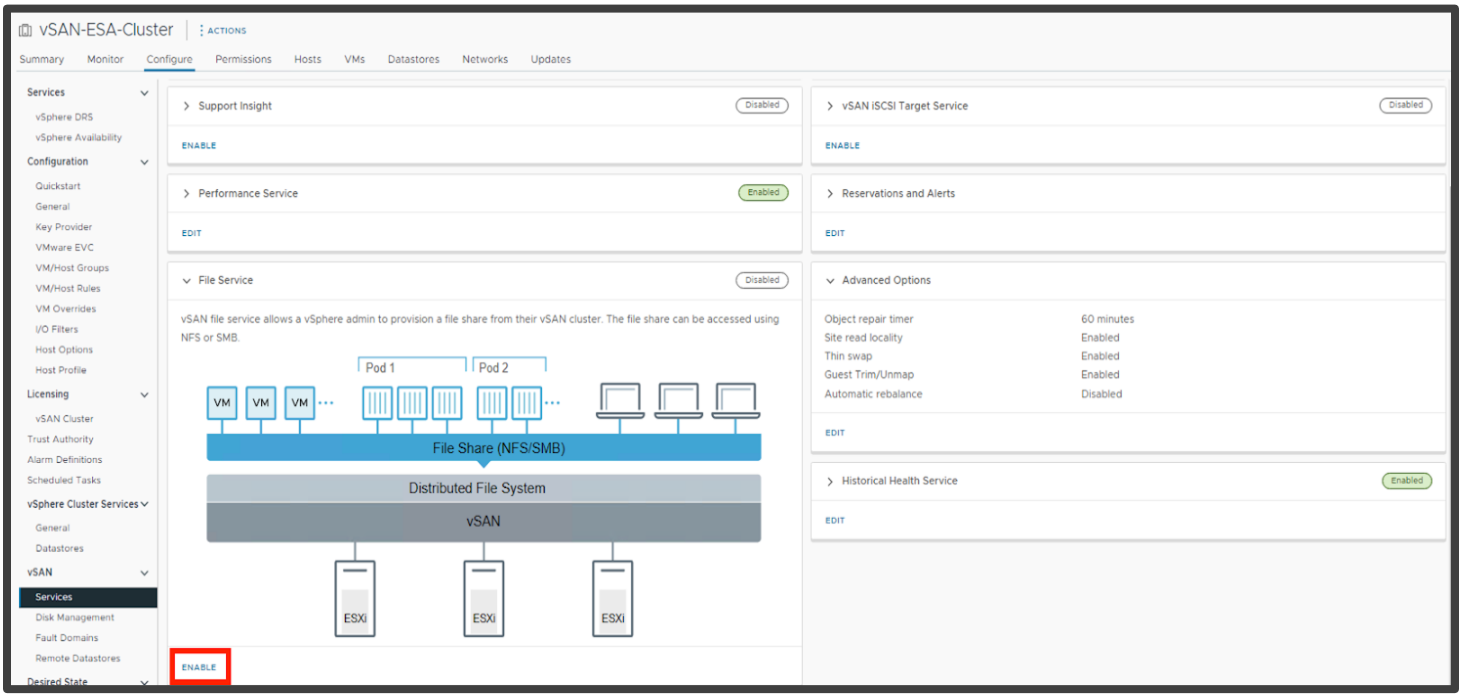
vSAN file services are implemented as a set of file server agent VMs (managed by the vSphere ESX Agent Manager). Each agent is a lightweight virtual appliance running Photon OS with a containerized file server. The agent VM is deployed using an OVF file stored in vCenter.

The process to enable vSAN file services is identical across both vSAN ESA and vSAN OSA. Although the walkthrough describes the process using a vSAN ESA cluster, the steps apply equally to vSAN OSA.

For clusters using DVS, it is recommended that a new network port group be created on the distributed switch that the hosts are connected to, as vSAN file services will enable both forged transmits and promiscuous mode. In the example below, we are creating a new port group on VLAN 1002:



Next, navigate to **[vSAN Cluster] > Configure > vSAN >Services.** In the list of services, we see that **File Services** is currently disabled. Begin by clicking **Enable**.

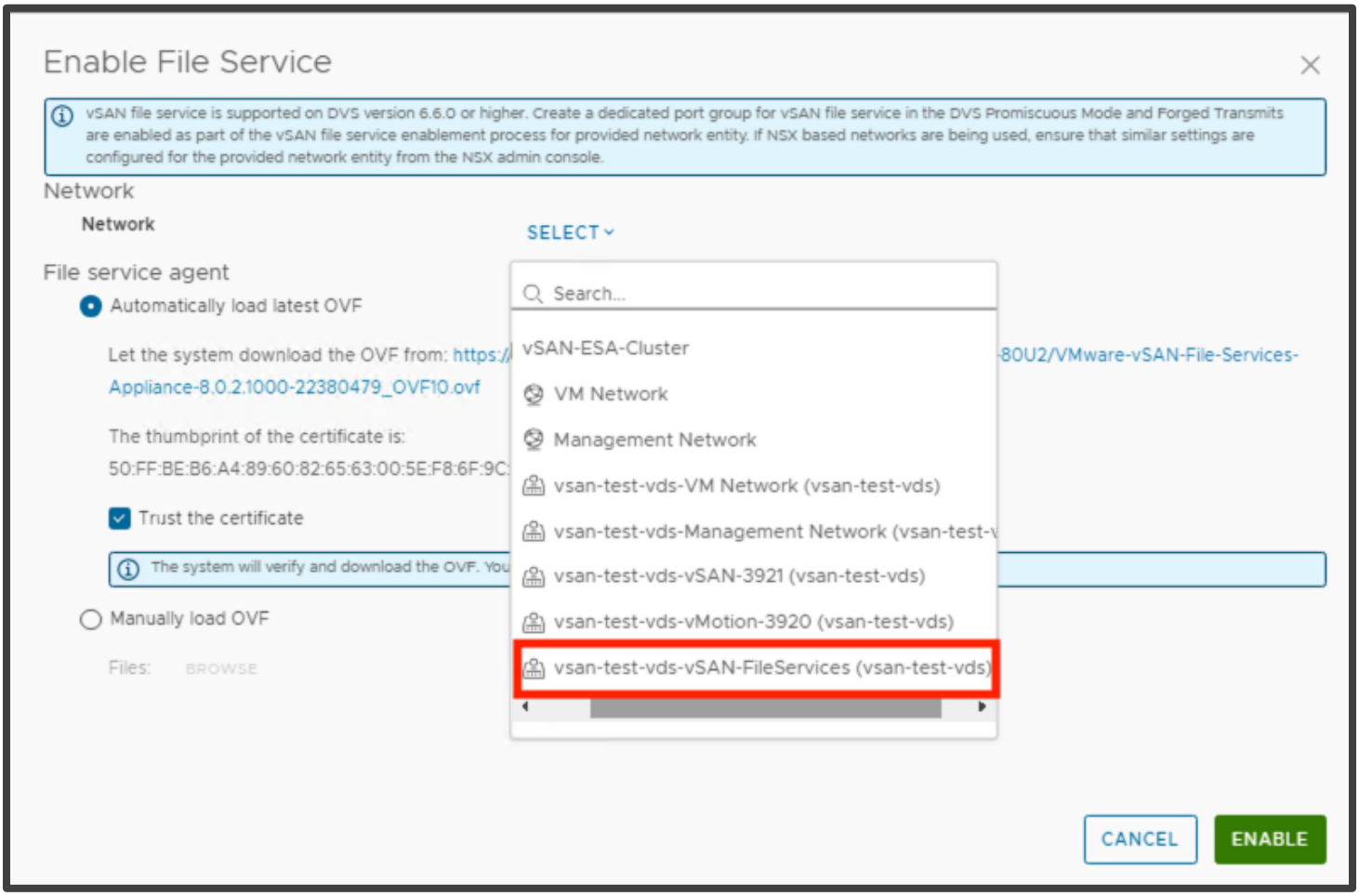



Next, select whether to download the File Service Agent OVF automatically or manually. In this example, "Automatically load latest OVF" as well as 'Trust the certificate' are selected.

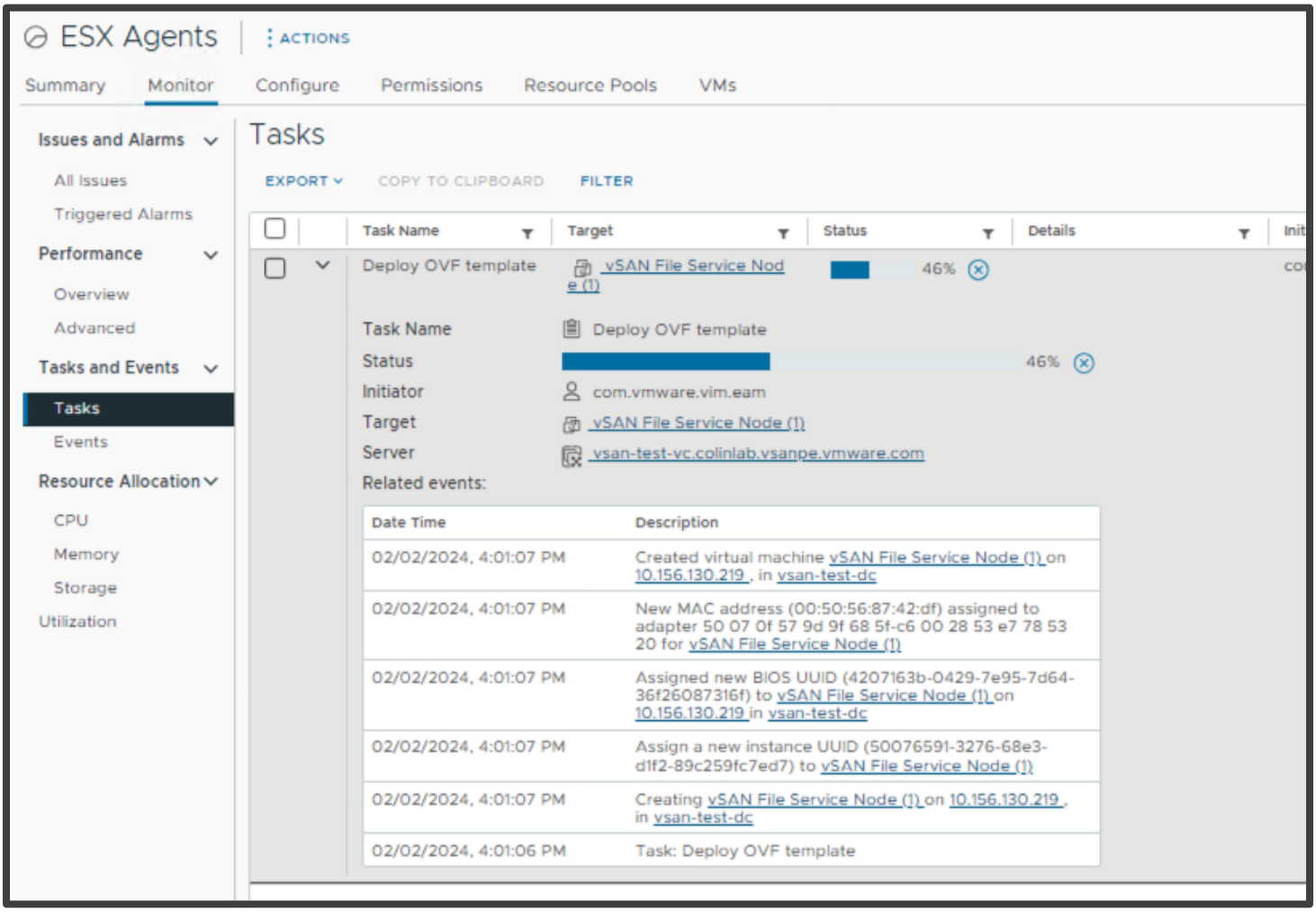
For air-gapped environments, download the agent OVF file from the link below (ensure it corresponds to the versions of vSphere/vSAN in your environment). Then select the 'Manually load OVF option to upload the file.

Direct Download:
https://customerconnect.vmware.com/en/downloads/info/slug/datacenter_cloud_infrastructure/vmware_vsan/8_0#drivers_tools

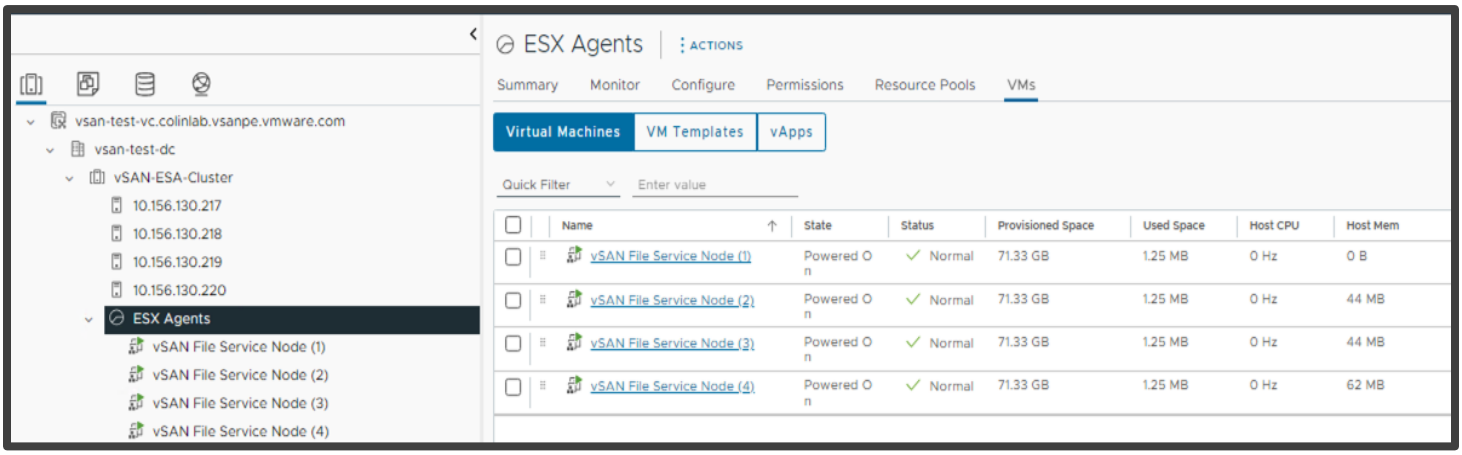
Next, select the port group that was created in the first step:



On clicking enable, vCenter will download the File Services OVF then deploy the agent VMs. A new resource pool named 'ESX Agents' will be created, and the stored OVF deployed and cloned:

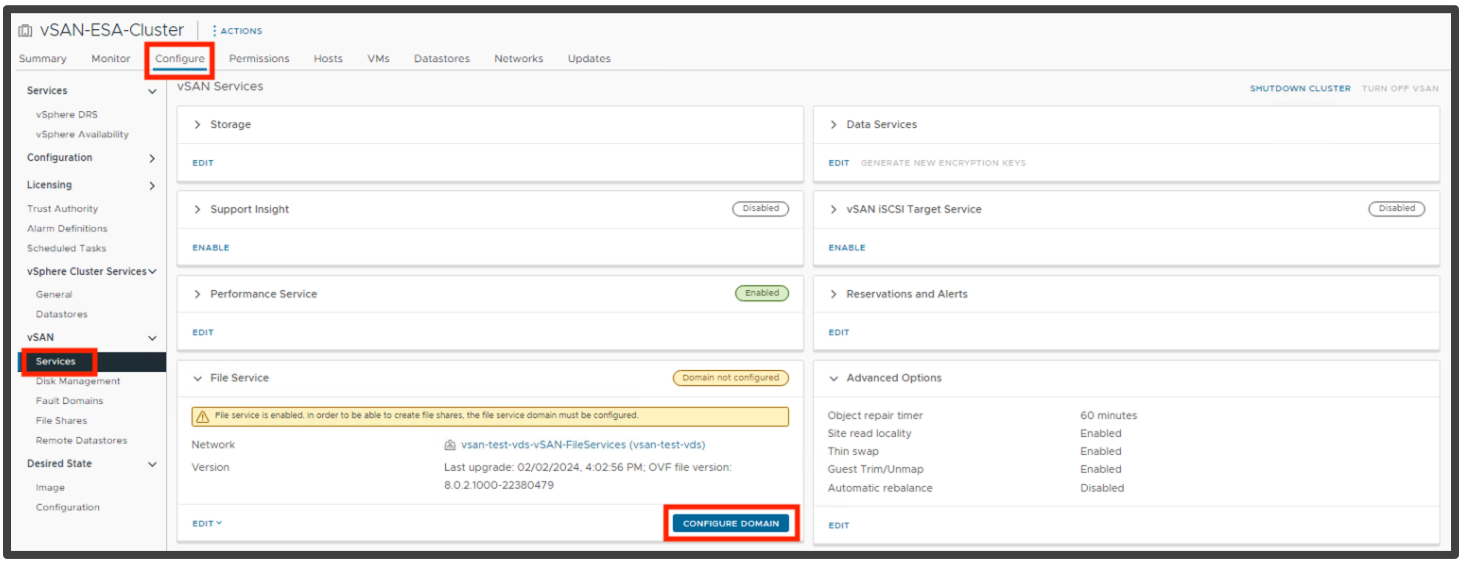



Once this process has finished, the service VMs will be visible in the inventory view:

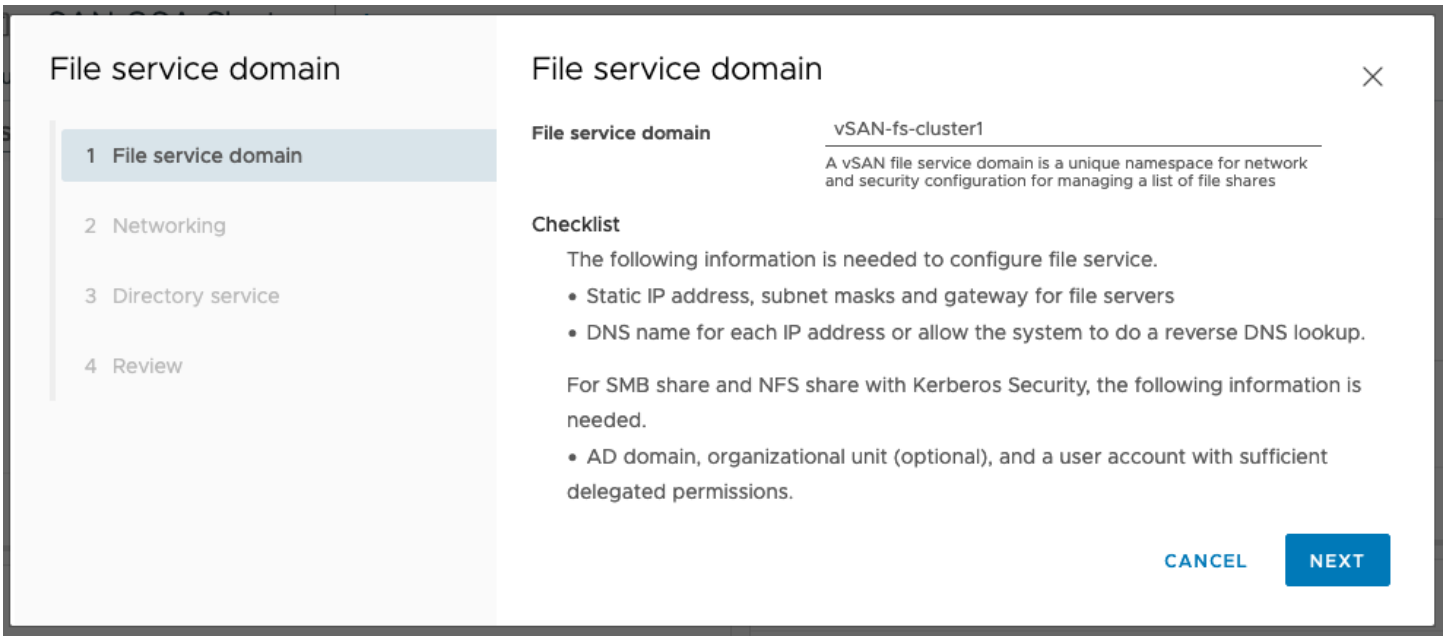


The file service agent VMs will use a customized storage policy "FSVM_Profile_DO_NOT_MODIFY". As the name suggests, do not modify this policy (or assign a different policy to the file service VM).

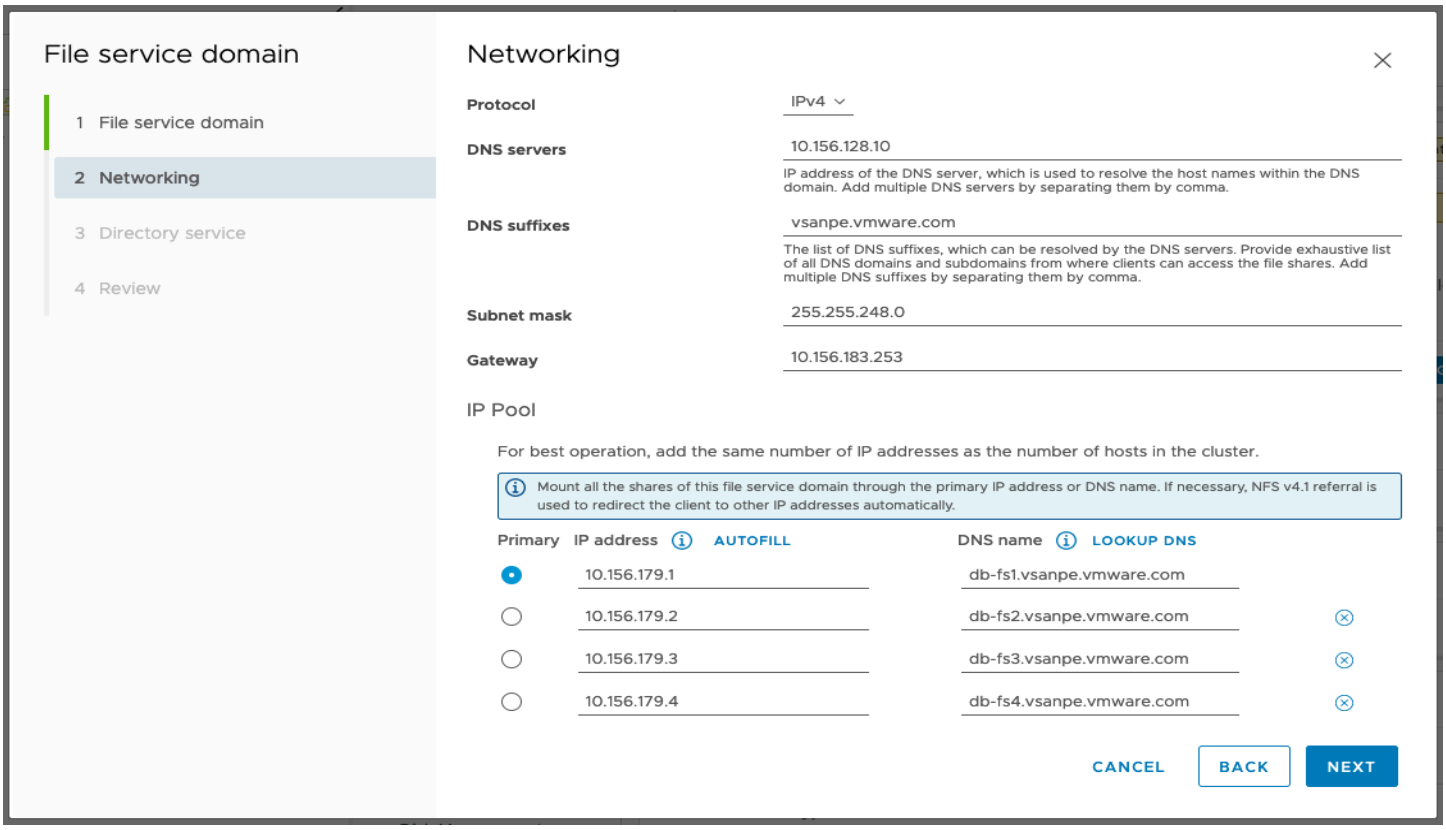
The next step is to create a file service domain. Navigate again to **[vSAN cluster] > Configure > vSAN > Services > File Service** and click on **CONFIGURE DOMAIN**:



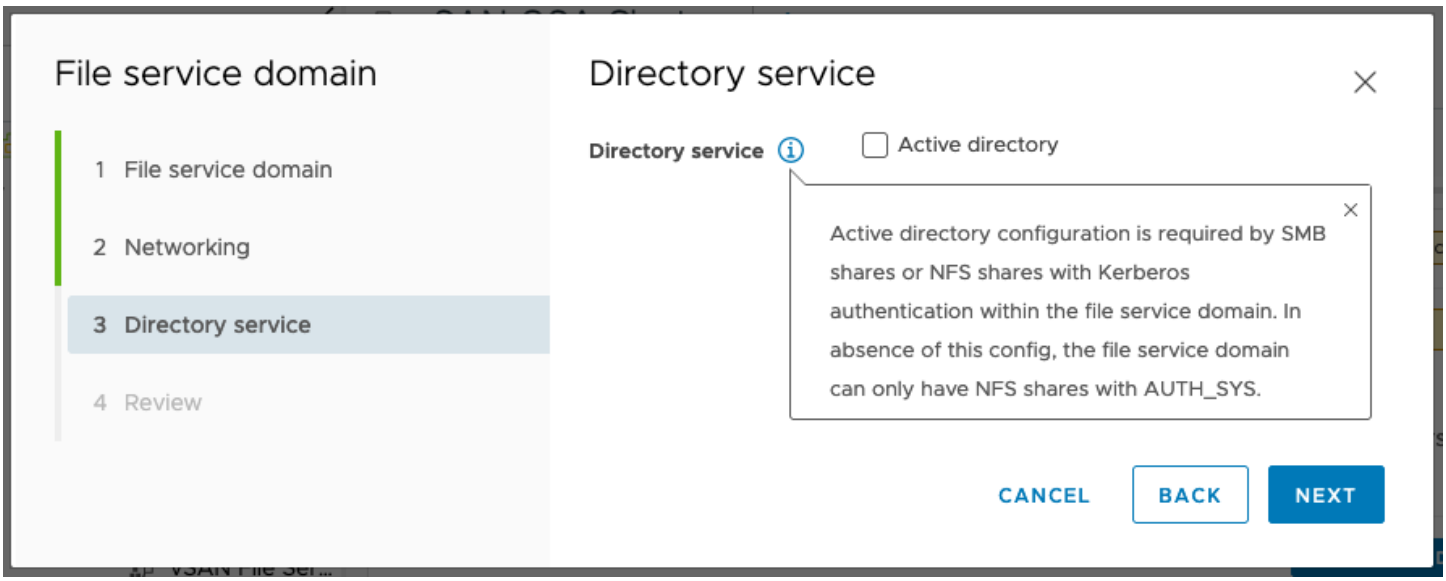

Name the domain. In this example, we have chosen the name 'vSAN-fs-cluster1':



On the next screen, enter the networking details (DNS server, suffix, gateway, etc.) and the IP addresses that will be used by the vSAN file service. For consecutive addresses, use **AUTOFILL** option to save on typing. Once the IP addresses have been specified, click on **LOOKUP DNS** to ensure that vCenter can resolve the addresses:



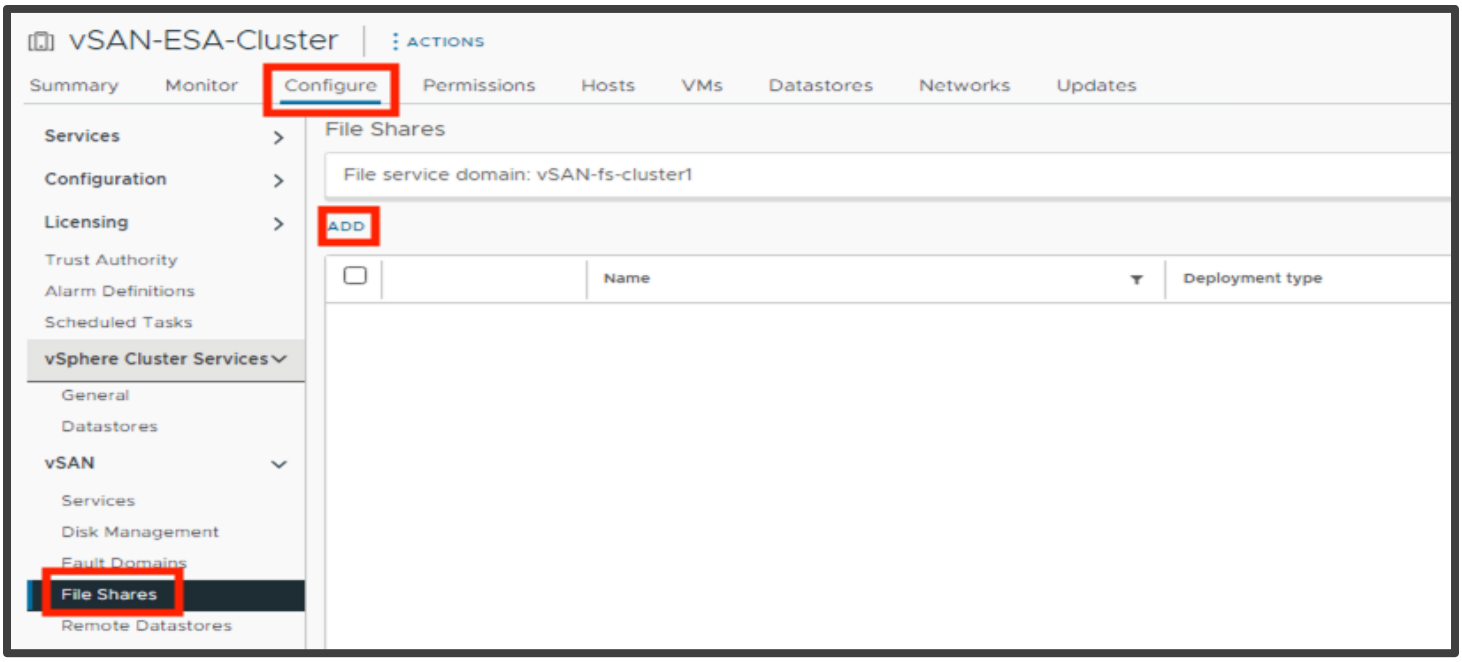

On the next screen, click to enable Active Directory services, if required. Again, this is needed for SMB shares or Kerberos authentication with NFS:



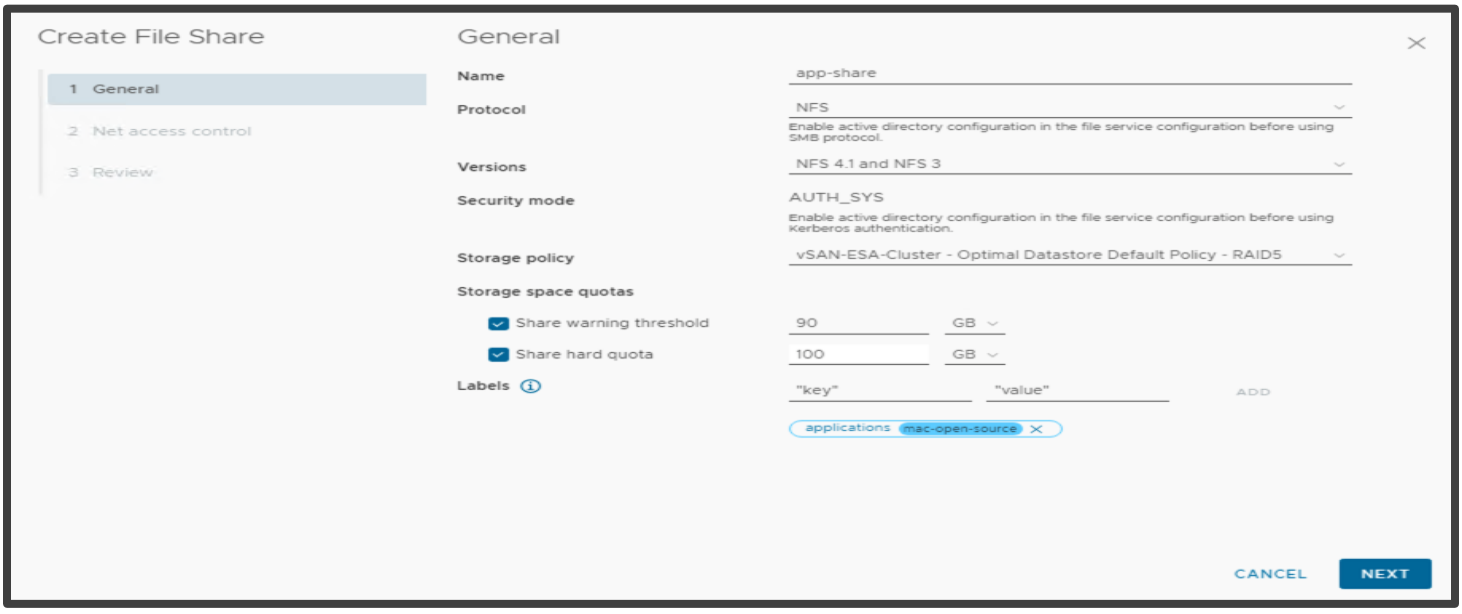
Click **Review** and finally **Finish** on the next screens. If all is well, vCenter will then proceed to enable the vSAN file service.

## Creating a File Share

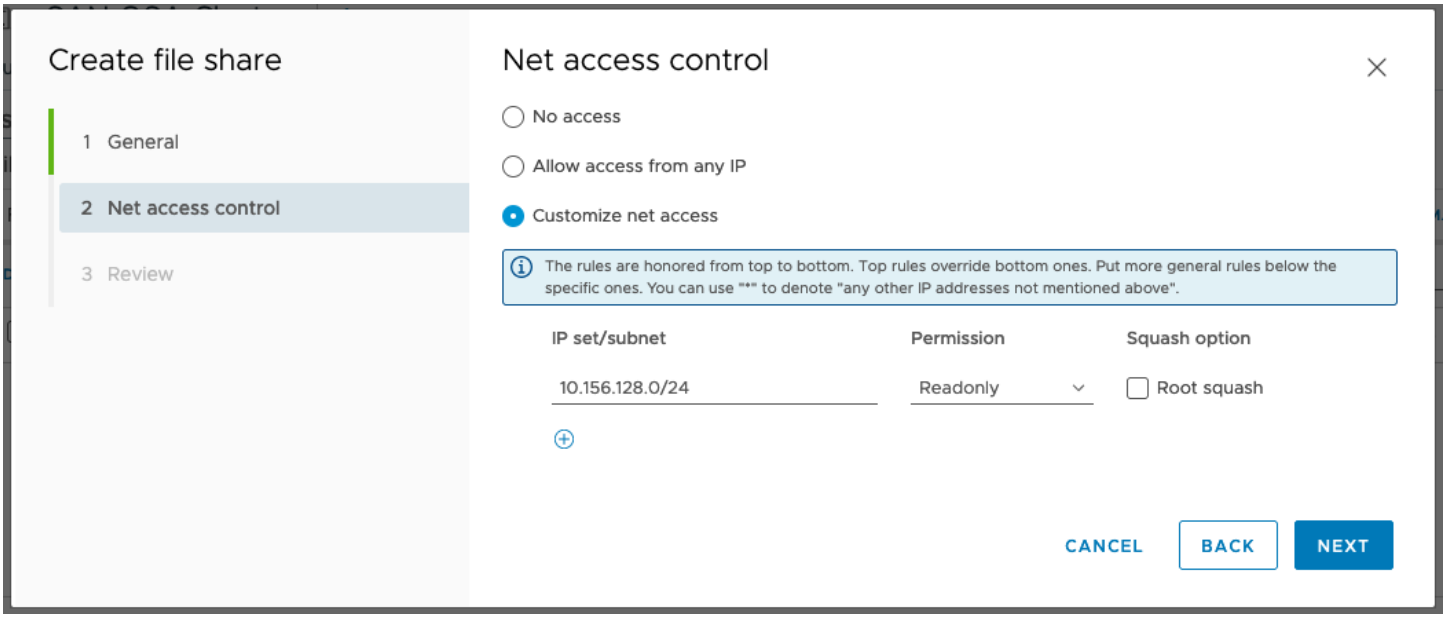
Once file services are enabled navigate to **[vSAN Cluster] > Configuration > vSAN > File Shares** and click on **ADD**:



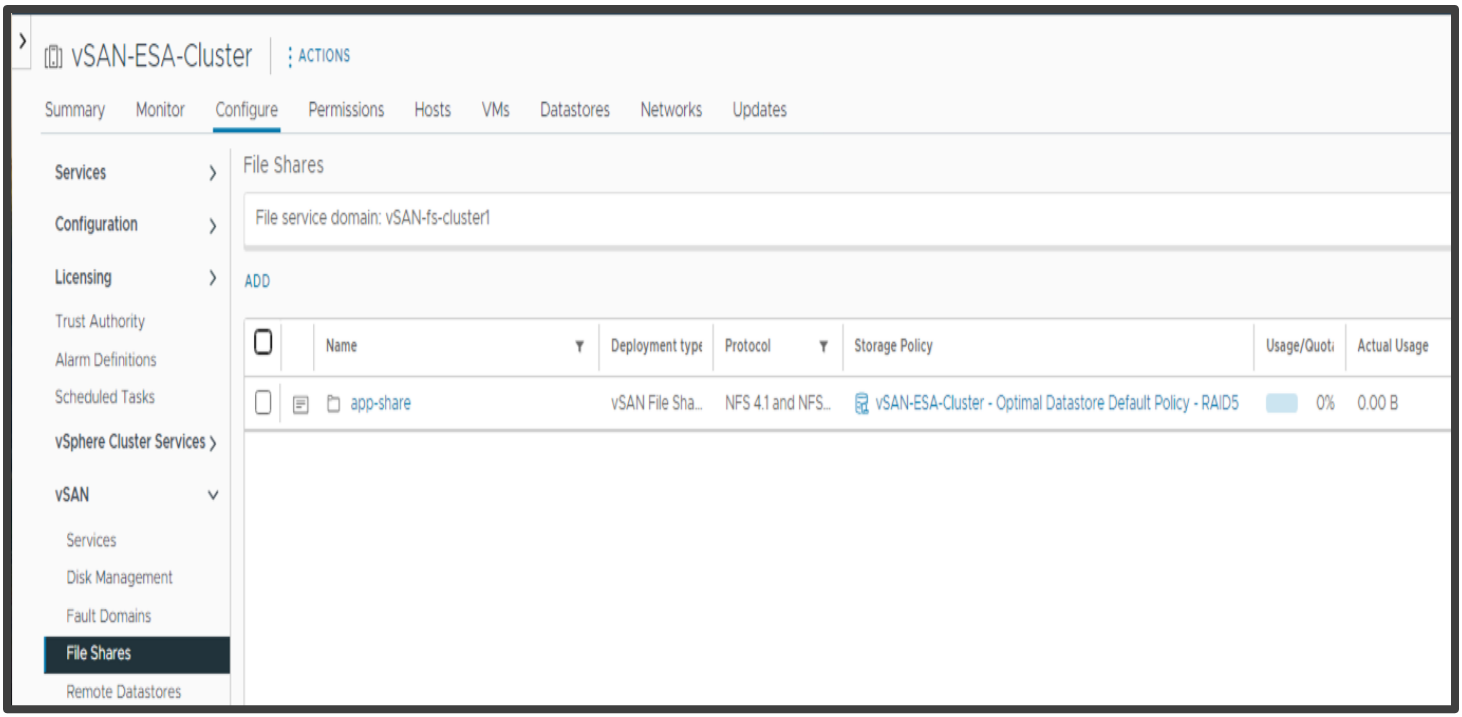

In this example, we are creating an NFS share called 'app-share' with the vSAN default storage policy. We have set a 90GB warning for space usage, with a hard quota of 100GB. Additionally, we have created a label 'mac-open-source' with the key of 'applications':



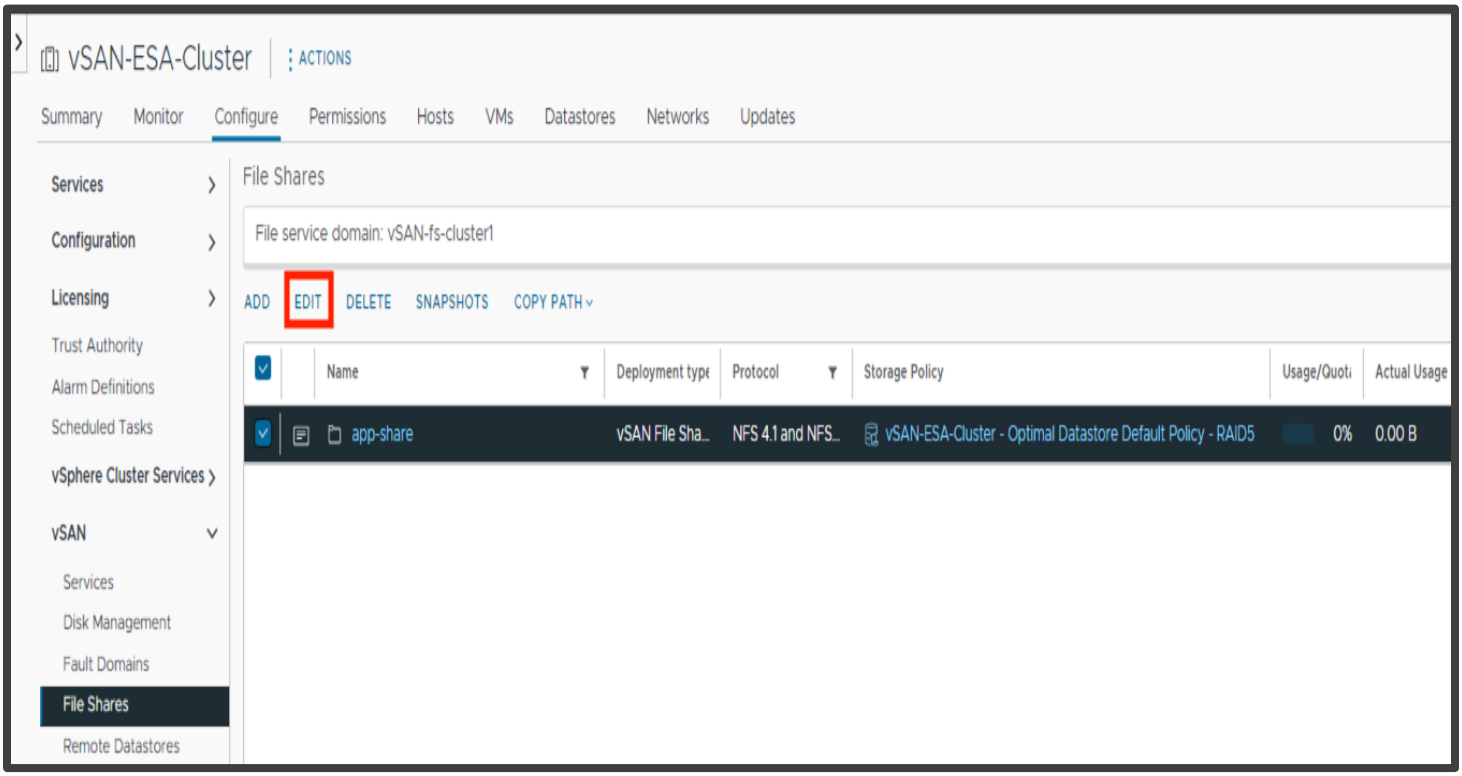
On the next screen we have a defined specific range of read-only clients (here we have defined a subnet, but a range of IP addresses could also be used):




Click **Review** and **Finish** to create the share. Once vCenter has created to file share, it will be displayed in the File Services Shares:

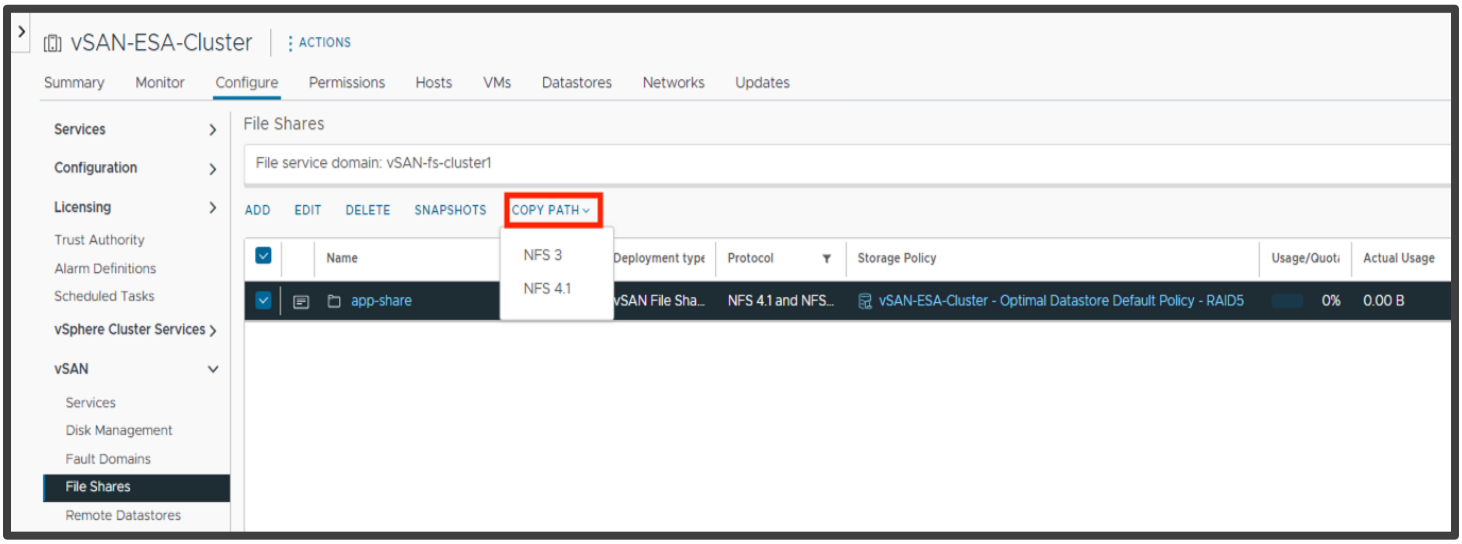


After the share has been created any quota, labels or network permissions can be made by selecting the file share and selecting **EDIT**.




## Mounting a File Share

First, obtain the path of the share by navigating to **[vSAN Cluster] > Configure > vSAN > File Shares**. Select the appropriate file share and click on **COPY PATH**:

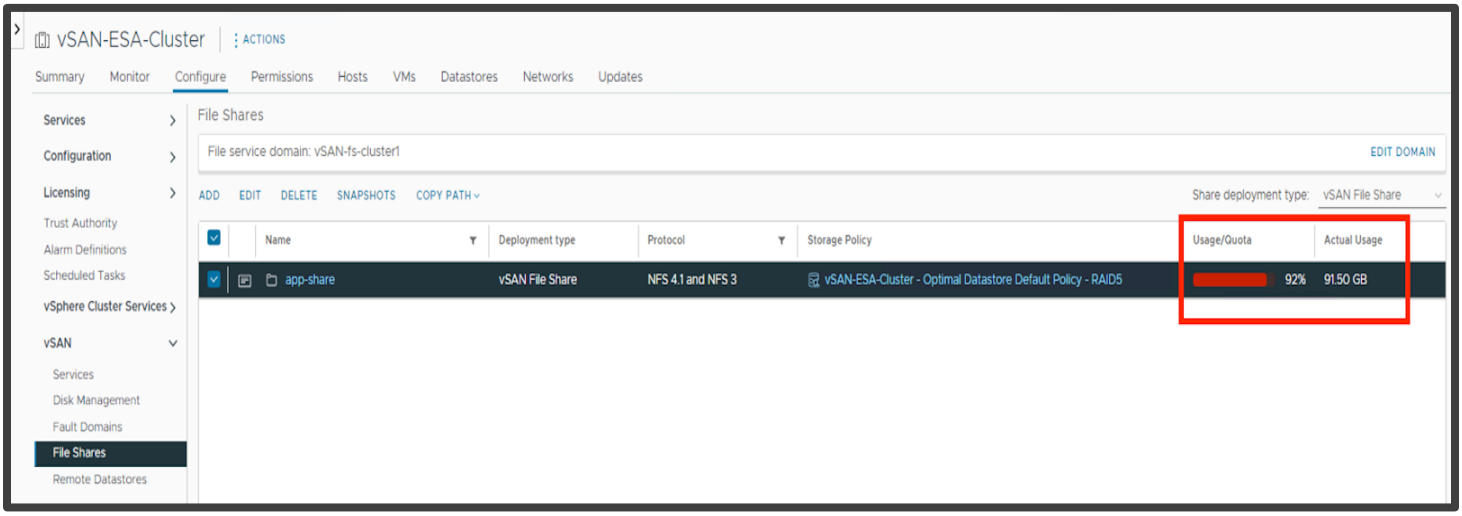


We can then mount the share as desired, for instance:

```
$ mount db-fs1.vsanpe.vmware.com:/vsanfs/app-share /mnt/app-share
```

## Quotas and Health Events

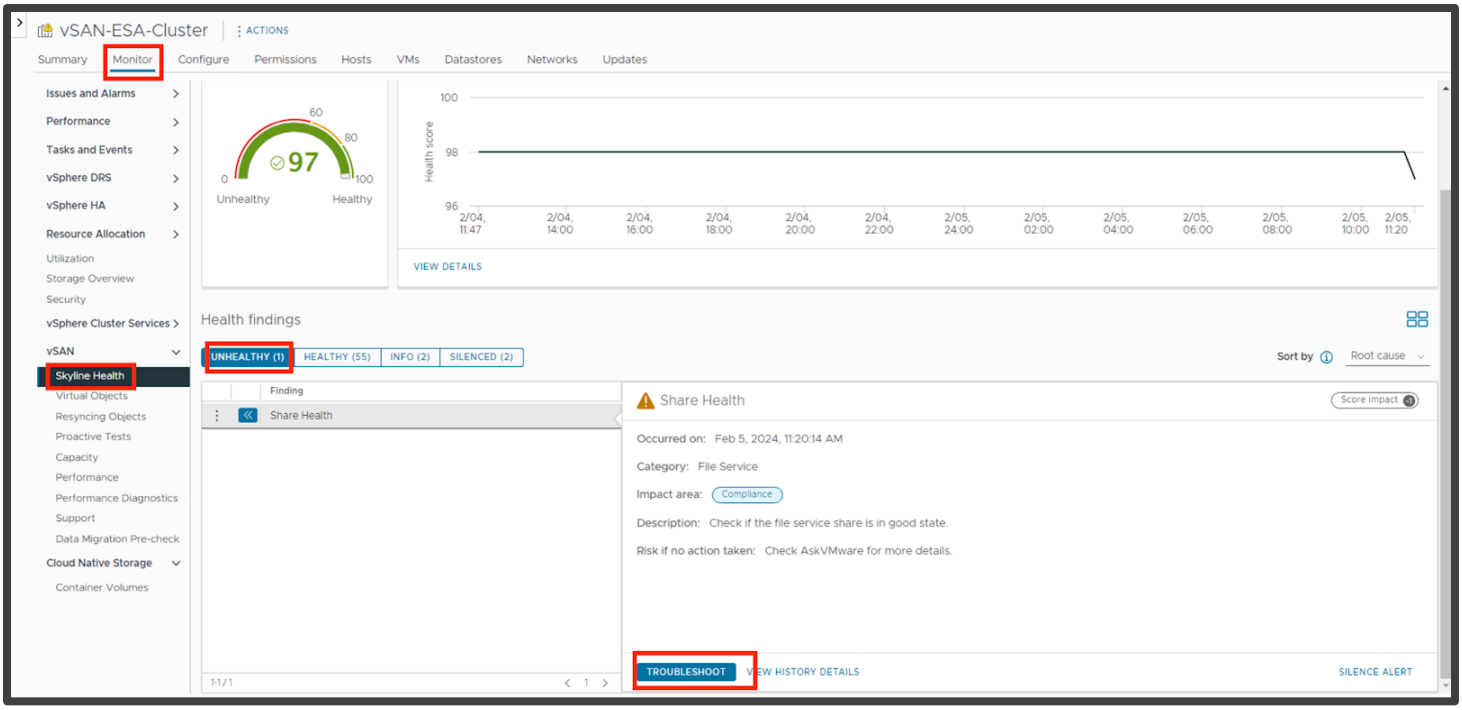
During file share creation, quotas and hard limits can be set. In this sample file share, a warning threshold of 90 GB was specified and a hard limit of 100GB was set. As part of this test copy some data to the file share to fill the space required to trigger the quota. Once the warning threshold is exceeded the Usage over the Quota field in the UI will turn red.

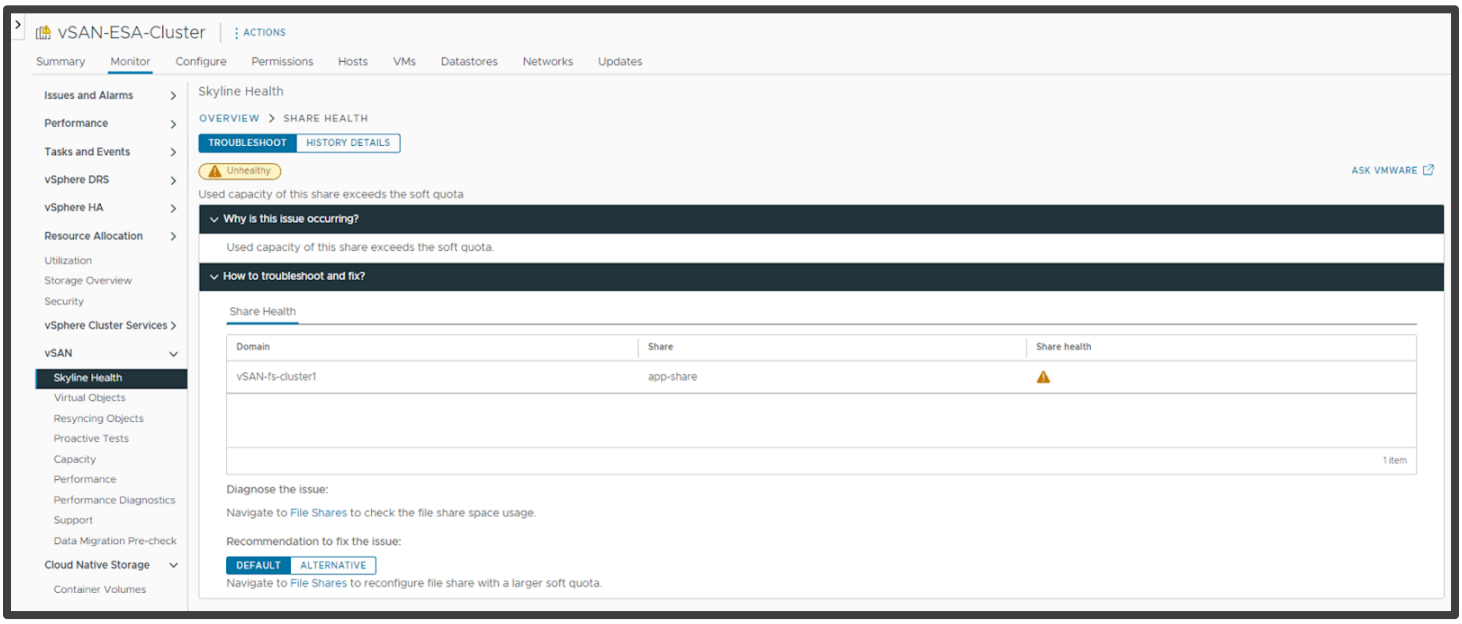



Once the hard quota is reached writes to the share will fail with a disk quota exceeded error as shown below.

```
cp: error writing 'file13.txt': Disk quota exceeded
```

If the quota is reached an alarm in Skyline Health is also triggered. The details of the alarm can be viewed by expanding the Share Health finding.

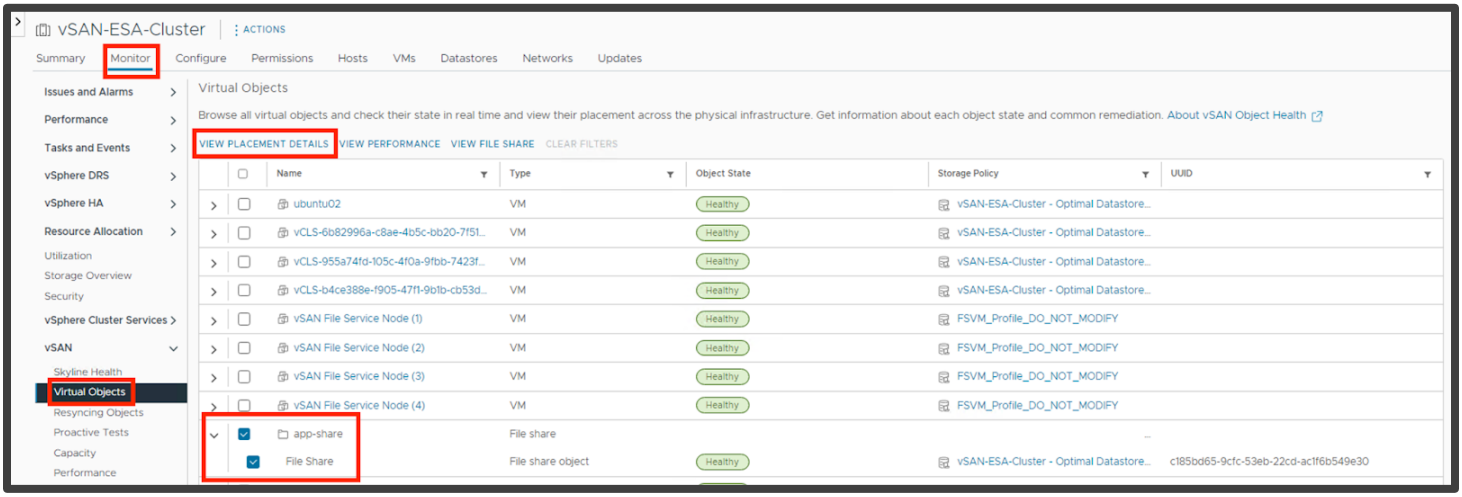


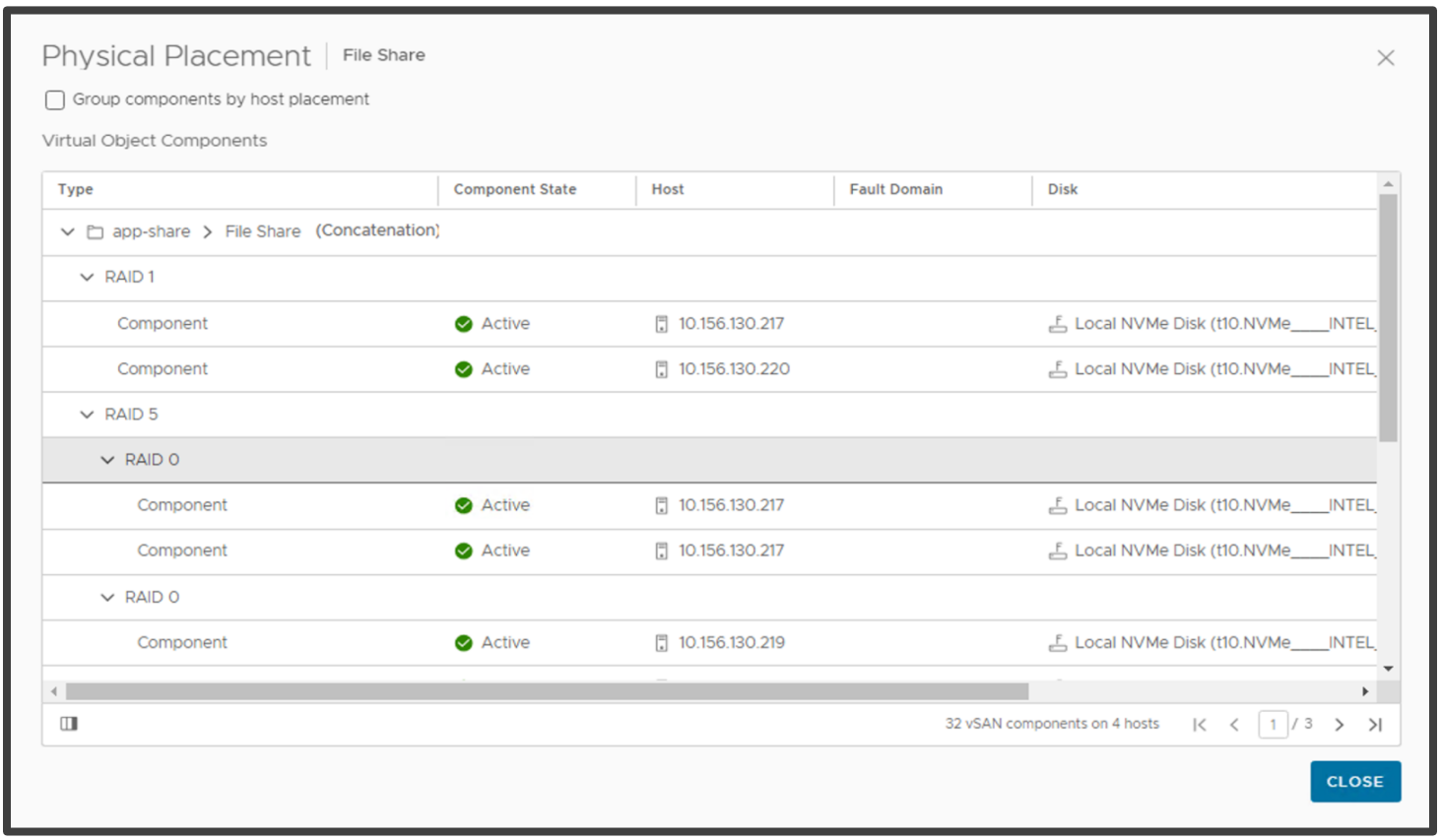

Once data has been removed from the Share Health alert is cleared and the File Service health reports as normal.

## Failure Scenarios

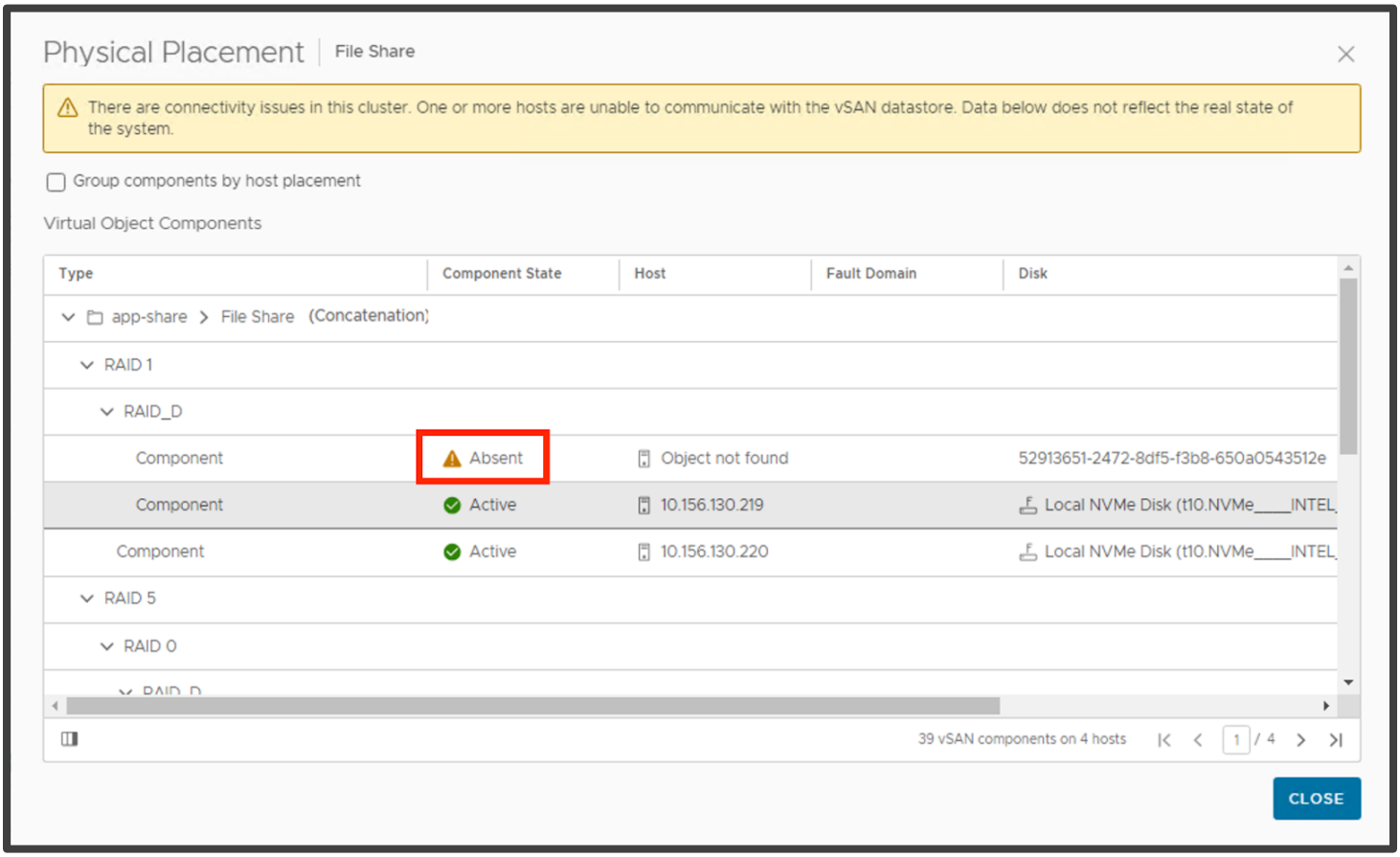
Storage policies apply to file service objects just as they do other virtual disk objects. Health and placement details of file shares are shown in the **Virtual Objects** view.




By clicking **VIEW PLACEMENT DETAILS** the layout of the underlying vSAN object can be viewed. This view shows component status, and on which hosts components of the share reside.



To test host failure, we will power off one of the hosts containing an active copy of the file share data. Once the host is powered off, we see that the component of the corresponding host is displayed as absent.

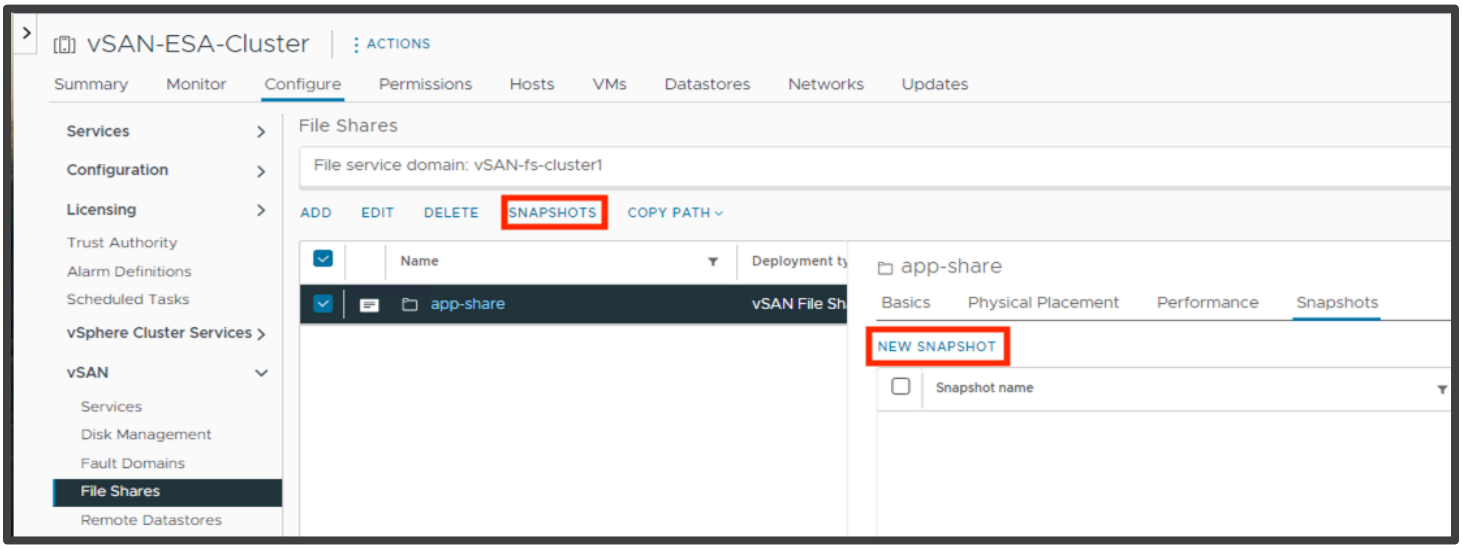



Now that the host has been shut down, you can validate from any of the client virtual machines through a file browser or logs to verify that file share is still accessible.

When ready power the host back on.

## File Services Snapshots

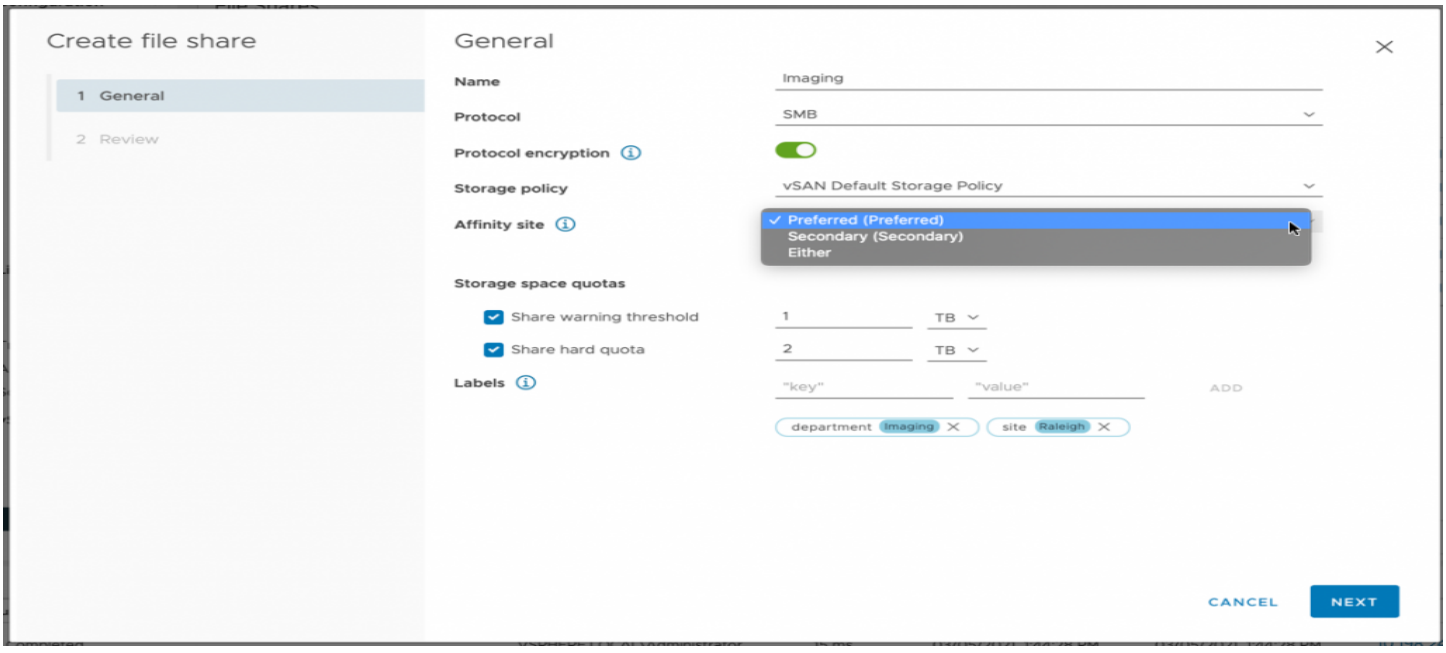
vSAN includes a new snapshotting mechanism allowing for point-in-time recovery of files. Snapshots for file shares can be created through the UI. Recovery of files is available through API allowing backup partners to extend current backup platforms to protect vSAN file shares.




## File Services Support for Stretched Clusters and Two Node Topologies

File services can now be used in vSAN stretched clusters and two node topologies. The site affinity setting for file shares defines where the presentation layer resides. Site affinity for file shares is defined by the storage policy associated with the individual file shares. The storage policy and site affinity settings to be applied to the file share are defined as part of the creation process.

The image below is an example of the site affinity setting available when creating a file share in a stretched cluster.



## vSAN Support for Kubernetes

vSAN fully supports native VMware Tanzu® as well as 'vanilla' Kubernetes clusters.

In VMware Tanzu deployments, vSAN natively provides both block and file storage services to persistent volumes (PV).

For more information on leveraging vSAN within VMware Tanzu, please refer to:

- vSphere with Tanzu Planning Guide- https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-with-tanzu-concepts-planning/GUID-E297DD43-AEEB-4B6D-8C93-4212CA62309A.html#GUID-E297DD43-AEEB-4B6D-8C93-4212CA62309A
- vSphere with Tanzu Installation Guide - https://docs.vmware.com/en/VMware-vSphere/8.0/vsphere-with-tanzu-installation-configuration/GUID-B09BFD99-DF1F-4831-BBA5-BAA78580AB47.html
- Tanzu Proof of Concept Guide - https://core.vmware.com/resource/tanzu-proof-concept-guide

For native Kubernetes deployments (non-Tanzu), Cloud-Native Storage (CNS), offers a platform for stateful cloud-native applications to persist state on vSphere backed storage. The platform allows users to deploy and manage containerized applications using cloud-native constructs such as Kubernetes persistent volume claims and maps these to native vSphere constructs such as storage policies. CNS integrates with vSphere workflows and offers the ability for administrators to perform tasks such as defining storage policies that could be mapped to storage classes, list/search and monitor health and capacity for PVs.

For more information on CNS, please refer to:

- vSphere Container Storage Plug-in Documentation Main Page - https://docs.vmware.com/en/VMware-vSphere-Container-Storage-Plug-in/index.html
- Getting Started with VMware vSphere Container Storage Plug - https://docs.vmware.com/en/VMware-vSphere-Container-Storage-Plug-in/3.0/vmware-vsphere-csp-getting-started/GUID-74AF02D7-1562-48BD-A9FE-C81A53342AC3.html
- vSphere CSI Driver GitHub - https://github.com/kubernetes-sigs/vsphere-csi-driver

# APPENDIX A: Creating Test VMs

Here we demonstrate how to quickly create a set of identical VMs for testing.

## Requirements:

- FreeBSD, Linux or MacOS VM/host environment
- Latest version of govc (download instructions below)

## Download govc:

Govc is a lightweight, open-source CLI tool written in Go (and part of the Govmomi/Go library for the vSphere API). Project page: https://github.com/vmware/govmomi/tree/master/govc

**To download the latest release, use the command below, or visit the release page:**
https://github.com/vmware/govmomi/releases

As with the majority of Go projects, it is packaged as a single binary (note that the tar command requires root privileges to copy the binary to the correct location):

```
curl -L -o - "https://github.com/vmware/govmomi/releases/latest/download/govc_$(uname -s)_$(uname -m).tar.gz" | tar -C /usr/local/bin -xvzf - govc
```

## Connecting to vCenter

To authenticate with vCenter, we need to define the username, password and URL, as per the example below:

```
export GOVC_USERNAME=administrator@vsphere.local
export GOVC_PASSWORD=P@ssw0rd
export GOVC_INSECURE=1
export GOVC_URL=10.156.163.1
```

Additionally, we will need to specify the default datastore and resource pool (we can define this as the default/top-level cluster, as per blow) for deploying our VMs:

```
export GOVC_DATASTORE=ESA-vsanDatastore
export GOVC_RESOURCE_POOL='vSAN ESA Cluster/Resources'
```

Finally test the connection to vCenter by issuing the command below, it should return with details:

```
govc about
FullName:     VMware vCenter Server 8.0.0 build-20519528
Name:         VMware vCenter Server
Vendor:       VMware, Inc.
Version:      8.0.0
Build:        20519528
...
```

### Configure Test VM

First, specify a location of an OVA file to use. In the example below, we use an Ubuntu 22.04 cloud image:

```
export vmLocation=https://cloud-images.ubuntu.com/releases/22.04/release/ubuntu-22.04-server-cloudimg-amd64.ova
```

We can then add our customizations, etc. by extracting the JSON from the OVA:

```
govc import.spec $vmLocation > ubuntu-vm.json
```

Ubuntu uses cloud-init to setup the OS environment. As we will be cloning the deployed VM, we need to define specific user-data (which will be encoded in base-64 and added to the customization JSON). Here we ensure that vSphere specific configuration is not disabled, and we modify the default netplan configuration file to ensure DHCP addresses are assigned by mac address (rather than machine-id).

To simplify the process, the user-data file can be downloaded from the link below:
https://raw.githubusercontent.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/main/VCF/test_vms/user-data

```
#cloud-config
runcmd:
  - 'echo "disable_vmware_customization: false" >> /etc/cloud/cloud.cfg'
  - echo -n > /etc/machine-id
  - |
    sed -i '' -e 's/match.*/dhcp-identifier: mac/g' -e '/mac/q' /etc/netplan/50-cloud-init.yaml
final_message: "The system is prepped, after $UPTIME seconds"
power_state:
  timeout: 30
  mode: poweroff
```

If available, use the cloud-init CLI to check the user-data file:

```
$ cloud-init schema --config-file user-data
```

Next, we encode the user-data to base64:

```
base64 -i user-data
```

Now we can edit the JSON file we extracted earlier. Change the file with the following:

- Disk provisioning set to 'thin'
- Add the public key of the machine we are connecting from
- Remove the hostname and password data
- Set the network for the VM (the name of the relevant portgroup in vCenter)

- Set the name of the VM
- In the 'user-data' section, paste in the base64 encoded data

Note we can avoid hand-editing the JSON by using jq. For example, we can update the user-data field directly in the JSON file:

```
jq 'select(.Key=="user-data").Value="$(base64 -i user-data)"' ubuntu-vm.json
```

Similarly, adding a public key stored in a user's GitHub profile:

```
jq 'select(.Key=="public-keys").Value="$(curl -sk https://api.github.com/users/[github user]/keys |
jq -r '.[].key')"' ubuntu-v.json
```

An example of this file can be seen here:
https://raw.githubusercontent.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/main/VCF/test_vms/ubuntu-vm.json

```
{
  "DiskProvisioning": "thin",
  "IPAllocationPolicy": "dhcpPolicy",
  "IPProtocol": "IPv4",
  "PropertyMapping": [
    {
      "Key": "instance-id",
      "Value": "id-ovf"
    },
    {
      "Key": "hostname",
      "Value": ""
    },
    {
      "Key": "seedfrom",
      "Value": ""
    },
    {
      "Key": "public-keys",
      "Value": "ssh-rsa AAAAB3NzaC1yc2EAAAAD..."
    },
    {
      "Key": "user-data",
      "Value": "I2Nsb3VkLWNvbmZpZwpy..."
    },
    {
      "Key": "password",
      "Value": ""
    }
  ],
  "NetworkMapping": [
    {
      "Name": "VM Network",
      "Network": "DSwitch-DHCP"
    }
  ],
```

```
  "MarkAsTemplate": false,
  "PowerOn": false,
  "InjectOvfEnv": false,
  "WaitForIP": false,
  "Name": "ubuntu-vm"
}
```

Once this JSON file has been defined, we can double-check our user-data encoding is still correct:

```
awk -F '"' '/user-data/{getline; print $4}' ubuntu-vm.json | base64 -d
```

This should return the user-data as we defined above.

### Import OVA to vCenter and Clone

We can then import the OVA into vCenter, specifying our JSON customization file:

```
govc import.ova -options=ubuntu-vm.json -name=ubuntu-template $vmLocation
```

After this has imported, we can update the virtual disk size. Here we set it to 100G:

```
govc vm.disk.change -vm ubuntu-template -disk.label "Hard disk 1" -size 100G
```

Power on the VM to allow it to run cloud-init (and thus our previously defined commands). Once complete, the VM will shutdown:

```
govc vm.power -on ubuntu-template
```

Once the VM has shutdown, mark it as a template:

```
govc vm.markastemplate ubuntu-template
```

Finally, we can clone our template VM as we need to. In the example below, we clone it ten times:

```
for x in {1..10};do govc vm.clone -vm ubuntu-vm ubuntu-vm$x;done
```

To do this for a large number of VMs, in parallel (and output to a log file) we could run:

```
for x in {1..250};do (govc vm.clone -vm ubuntu-template ubuntu-vm$x >> $(date +%d%m-%H%M)_clone.log
2>&1 &);done
```

We can monitor progress by probing the vCenter task-list:

```
govc tasks -f -l
```

After cloning, we can batch-execute commands on all the VMs. For example, the 'ls' command:

```
govc find -type m -name 'ubuntu-vm*' | xargs -P0 -I '{}' bash -c 'ssh -o "StrictHostKeyChecking=no"
ubuntu@$(govc vm.ip {}) ls'
```

# APPENDIX B: Cleanly Removing vSAN Configuration

## vCLS Retreat Mode

On occasion, it may become necessary to remove a vSAN cluster and reset hosts to a 'clean' state.

To expedite the process, it is advisable to first put vCLS into retreat mode. This will delete the vCLS VMs and make it easier to remove the vSAN datastore and put hosts into maintenance mode, etc.

To achieve this, an vCenter advanced setting, '`config.vcls.clusters.[domain].enabled`' needs to be set.

The procedure to do this is detailed in the documentation here: https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.resmgmt.doc/GUID-F98C3C93-875D-4570-852B-37A38878CE0F.html

To make this easier a script is available here to use (download to a Linux or Mac host, uses govc):
https://github.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/blob/main/VCF/vCLS.sh

## Remove vSAN Partitions and Clear Data

The next step is to turn off vSAN from vCenter, under [cluster] > Configure > Services > vSAN. If for some reason this step encounters errors, the method below may be useful.

First, open an SSH session to all hosts in the cluster and list the disks used by vSAN by using the command:

```
vdq -iH
```

The next step depends on the type of cluster

### OSA Clusters

Remove the cache device from each disk group, using the command:

```
esxcli vsan storage remove -s [cache device]
```

### ESA Clusters

Remove disks from the storage pool, using the command:

```
esxcli vsan storagepool remove -d [device]
```

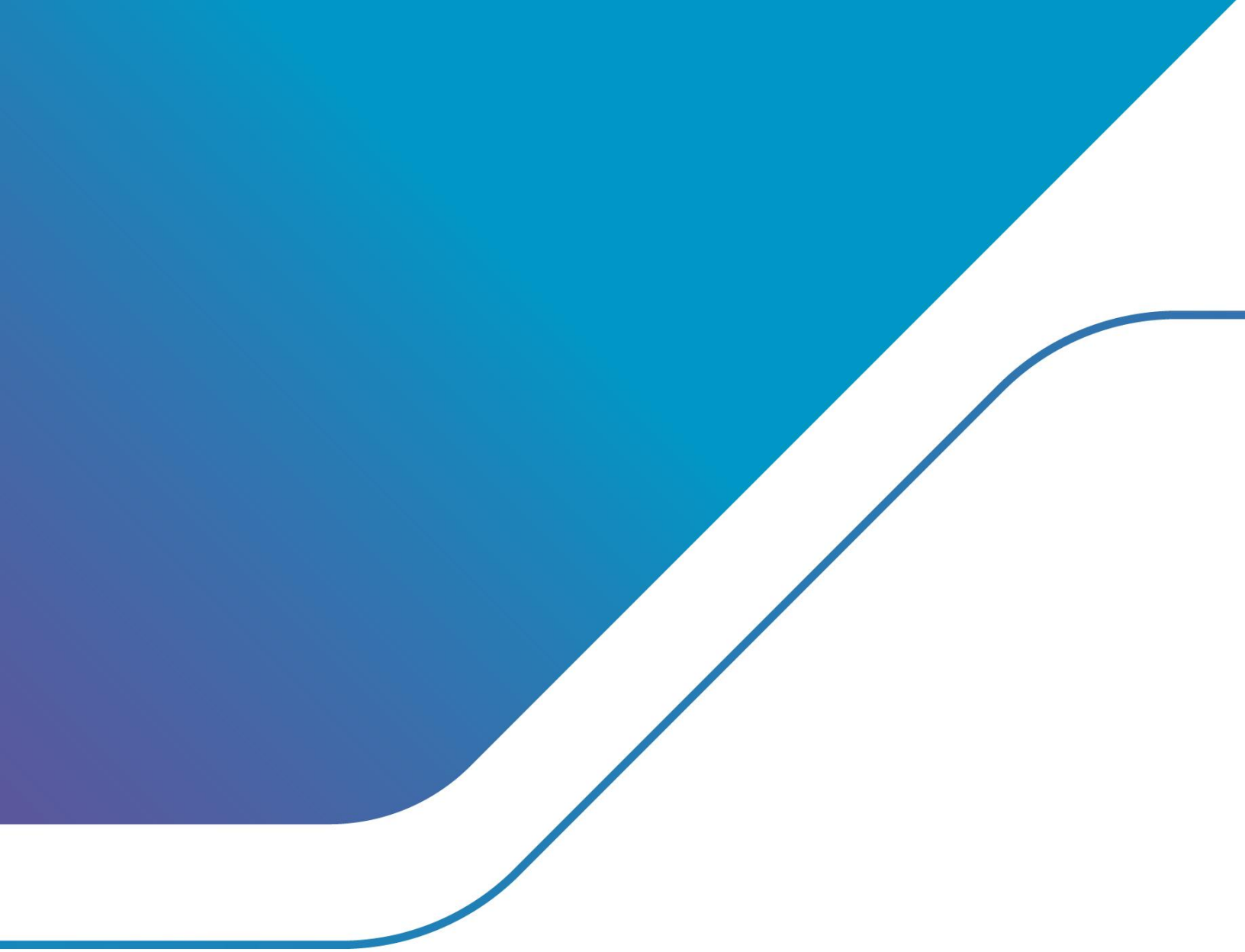
Next, relabel the disks:

```
partedUtil mklabel /vmfs/devices/disks/[disk] gpt
```

Again, to make this easier, a script is available to help with this:

OSA: https://github.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/blob/main/VCF/vsan-remove-esa.sh

ESA: https://github.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/blob/main/VCF/vsan-remove-esa.sh



vmware®
by Broadcom


Item No: vmw-bc-wp-tech-temp-uslet-word-2024 1/24