Table of Contents

Lab Overview - HOL-2108-01-HCI - vSAN 7 Features 3
- Lab Guidance 4

Module 1 - vSAN SPBM and Availability 7
- Introduction 8
- What's new in vSAN 7 9
- Storage Policy Based Management 11
- Scaling out the vSAN Environment 24
- Advanced Storage Based Policy Management 40
- Reserved Capacity 61
- Conclusion 65

Module 2 - vSAN File Services 66
- vSAN File Services Overview 67
- Enabling File Shares 69
- Creating NFS File Shares 73
- Mounting vSAN NFS File Shares to other Systems 77
- Creating SMB File Shares 85
- Client File Share Access 92
- vSAN File Services Monitoring 99
- Conclusion 106

Module 3 - vSAN Data Persistence Platform 107
- vSAN Data Persistence Platform 108

Module 4 - vSAN Cloud Native Storage (CNS) 112
- Kubernetes Overview 113
- Create a Block Persistent Volume 119
- Create a File Persistent Volume 131
- Conclusion 151

Module 5 - Day 2: Monitoring Health, Capacity and Performance 152
- Introduction 153
- vSAN Health Check Validation 154
- Monitoring vSAN Capacity 162
- Monitoring vSAN Performance 166
- Enabling vROPS for integration with vCenter 170
- vRealize Operations within vCenter Monitoring 179
- Conclusion 199
- Enabling vRealize Operations within vCenter 200

Module 6 - Day 2 Continue: Maintenance Mode and Lifecycle Management 207
- Introduction 208
- vSAN Maintenance 209
- vSphere Lifecycle Management (vLCM) 221
- Hands-on Labs Interactive Simulation: vSAN vLCM - Setup 223
- Hands-on Labs Interactive Simulation: vSAN vLCM - Update 224
- Hands-on Labs Interactive Simulation: vSAN vLCM - Rollback 225

Conclusion 226
Module 7 - vSAN HCI Mesh 227
vSAN HCI Mesh Overview 228
vSAN HCI Mesh Configure 230
Create Tag 239
Apply VM for HCI Mesh 255
Monitor HCI Mesh 260
Module 8 - vSAN Encryption and Security 263
Module 8 - vSAN Encryption 264
Data-In-Transit Encryption 266
DISA STIG (FIPS 140-2) Validated 268
vSAN Encryption 269
Conclusion 292
Appendix 294
Hands-on Labs Interface 295

Lab Overview - HOL-2108-01-HCI - vSAN 7 Features

Lab Guidance

Note: It may take more than 90 minutes to complete this lab. You should expect to only finish 2-3 of the modules during your time. The modules are independent of each other so you can start at the beginning of any module and proceed from there. You can use the Table of Contents to access any module of your choosing.

The Table of Contents can be accessed in the upper right-hand corner of the Lab Manual.

vSAN delivers flash-optimized, secure shared storage with the simplicity of a VMware vSphere-native experience for all of your critical virtualized workloads. Learn What's new with vSAN 7 such as file services and cloud native storage (CNS) and explore the vSAN environment including monitoring the health, capacity and performance of vSAN within vCenter and also via, built-in, vRealize Operations for vCenter Dashboards. Explore the all new, intuitive vSAN HTML5 user interface used to perform Day-2 Operations, maintain virtual machine availability, enable vSAN Encryption.

Lab Module List:

- **Module 1 - SPBM and Availability** (30 minutes) (Basic)

- Introduction to VMware vSAN. We will cover the power of Storage Based Policy Management (SPBM) and show you the availability of vSAN.

- **Module 2 - File Services** (15 minutes) (Basic)

- See how file services work and get created with vSAN.

- **Module 3 - vSAN Data Persistence Platform (DPP)** (30 minutes) (Advanced)

- Create a platform for stateful cloud native applications to persist state on vSAN. We will deploy and manage containerized applications using cloud native constructs such as Kubernetes persistent volume claims and map these to native vSphere constructs such as storage policies.

- **Module 4 - vSAN Cloud Native Storage (CNS)** (45 minutes) (Advanced)

- Learn how to transform modern stateful services will be deployed and managed in the private cloud.

- **Module 5 - Day 2 : Monitoring, Health, Capacity and Performance** (15 minutes) (Basic)

- Show you how to enable vRealize Operations within vCenter Server. We will cover the vSAN Health Check and how you can monitor your vSAN environment.

- **Module 6 - Day 2 Continue: Maintenance Mode, Lifecycle Management, and Adding Capacity** (30 minutes) (Advanced)

- Show you how to maintain your vSAN environment. We will also demonstrate how to upgrade your environment and expand the capacity of the vSAN Datastore.

- **Module 7 - vSAN HCI Mesh** (30 minutes) (Advanced)

- Learn how to bring together multiple independent vSAN clusters for a native, cross-cluster architecture that disaggregates resources and enables utilization of stranded capacity.

- **Module 8 - vSAN Encryption and Security** (30 minutes) (Advanced)

- Introduction to vSAN Encryption. We will enable a Key Management Server and demonstrate how to configure vSAN Encryption.

Lab Captains:

- **John Browne, Staff Technical Support Specialist, Cork, Ireland**
- **Aleksey Lib, Sr. Integration Architect, USA**
- **Diane Scott, Sr. Solutions Engineer, USA**

Principle:

- **Jim LaFollette, Staff Solutions Engineer, USA**

Content Architect:

- **Clive Wenman, Senior Content Architect, UK**

This lab manual can be downloaded from the Hands-on Labs Document site found here:

http://docs.hol.vmware.com

This lab may be available in other languages. To set your language preference and have a localized manual deployed with your lab, you may utilize this document to help guide you through the process:

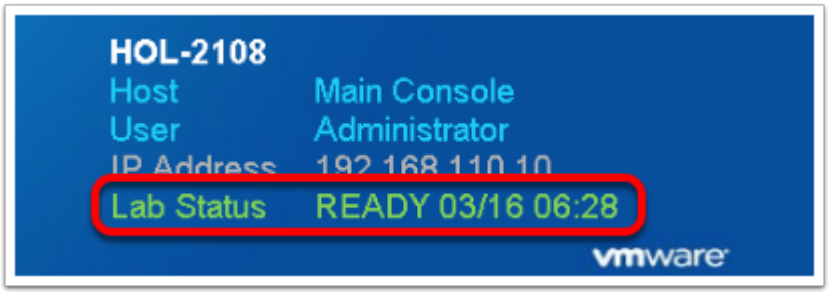
http://docs.hol.vmware.com/announcements/nee-default-language.pdf

First time using Hands-on Labs?

Welcome! If this is your first time taking a lab CLICK HERE to review the interface and features before proceeding.

For returning users, feel free to start your lab by clicking next in the manual.

You are ready....is your lab?



Please verify that your lab has finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait a few minutes. If after 5 minutes your lab has not changed to "Ready", please ask for assistance.

Module 1 - vSAN SPBM and Availability

Introduction

vSAN delivers flash-optimized, secure shared storage with the simplicity of a VMware vSphere-native experience for all your critical virtualized workloads. vSAN runs on industry-standard x86 servers and components that help lower TCO by up to 50% versus traditional storage. It delivers the agility to easily scale IT with a comprehensive suite of software solutions and offers the first native software-based, FIPS 140-2 validated HCI encryption. A VMware vSAN SSD storage solution powered by Intel® Xeon® Scalable processors with Intel® Optane™ technology can help companies optimize their storage solution in order to gain fast access to large data sets. View video to learn more.

vSAN 7 delivers a new HCI experience architected for the hybrid cloud with operational efficiencies that reduce time to value through a new, intuitive user interface, and provides consistent application performance and availability through advanced self-healing and proactive support insights. Seamless integration with VMware's complete software-defined data center (SDDC) stack and leading hybrid cloud offerings makes it the most complete platform for virtual machines, whether running business-critical databases, virtual desktops or next-generation applications.

What's new in vSAN 7

Before we jump in the Lab, let's take a moment to review What's New with vSAN 7. If you want to learn how to enable vSAN, please go to the vSAN Lightning Lab.

With vSAN 7, we are continuing to build on the robust features that make vSAN a high performing general-purpose infrastructure. vSAN 7 makes it easy for you to standardize on a single storage operational model with three new capabilities: integrated file services, enhanced cloud-native storage, and simpler lifecycle management. You can now unify block and file storage on hyperconverged infrastructure with a single control pane, which reduces costs and simplifies storage management. Cloud-native applications also benefit from these updates, which include integrated file services, vSphere with Kubernetes support, and increased data services. Finally, vSAN 7 also simplifies HCI lifecycle management by reducing the number of tools required for Day 2 operations, while simultaneously increasing update reliability.

Product Enhancements

The most significant new capabilities and updates of vSAN 7 include:

- **Enhanced Cloud-Native Storage**

vSAN supports file-based persistent volumes for Kubernetes on vSAN datastore. Developers can dynamically create file shares for their applications and have multiple pods share data.

- **Integrated File Services**

In vSAN 7, integrated file services make it easier to provision and share files. Users can now provision a file share from their vSAN cluster, which can be accessed via NFS 4.1 and NFS 3 and SMB. A simple workflow reduces the amount of time it takes to stand up a file share.

- **Simpler Lifecycle Management**

Consistent operations with a unified Lifecycle Mangement tool. vSAN 7 provides a unified VMware lifecycle management tool (vLCM) for Day 2 operations for software and server hardware. vLCM delivers a single lifecycle workflow for the full HCI server stack: vSphere, vSAN, drivers and OEM server firmware. vLCM constantly monitors and automatically remediates compliance drift.

- **Increased Visibility into vSAN Used Capacity**

Replication objects are now visible in vSAN monitoring for customers using VMware Site Recovery Manager and vSphere Replication. The objects are labeled “vSphere Replicas” in the “Replication” category.

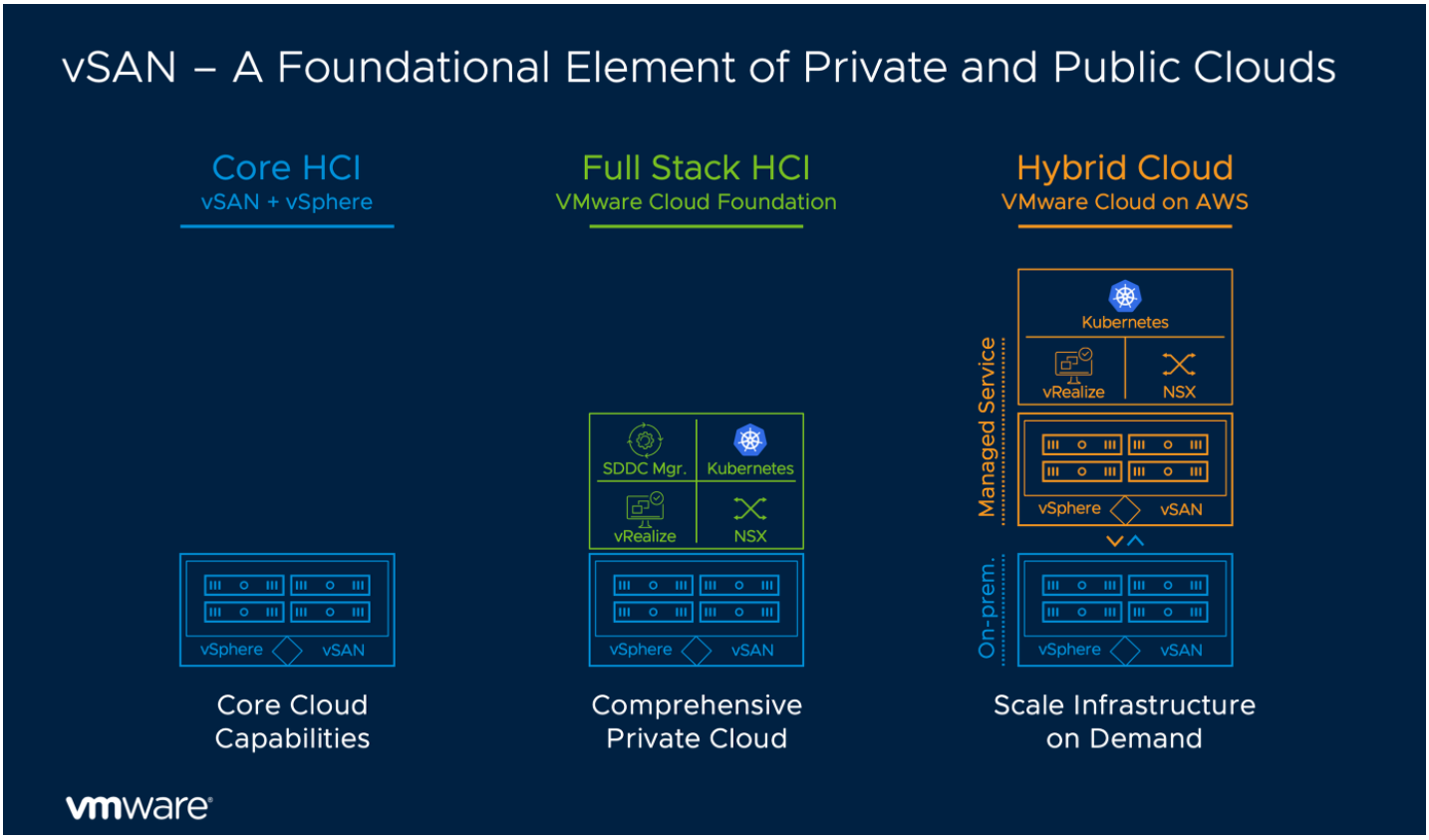
- **Uninterrupted Application Run Time**

vSAN 7 enhances uptime in Stretched Clusters by introducing the ability to redirect VM I/O from one site to another in the event of a capacity imbalance. Once the disks at the first site have freed up capacity, customers can redirect I/O back to the original site without disruption.

- **Automatic Policy Compliance for 2-Node vSAN Deployments**

vSAN 7 keeps 2-node deployments in policy compliance by automating repair objects operations during witness replacement.

Why vSAN?



VMware's solution stack offers the levels of flexibility that is needed for today's rapidly changing needs. It's built off of a foundation of VMware vSphere, paired with vSAN. This provides the basis for a fully software defined storage and virtualization platform that removes dependencies from legacy solutions using physical hardware. Next is VMware Cloud Foundation, the integrated solution that provides the full stack of tools for an automated private cloud. And finally, there is VMware Cloud on AWS. The same software that you already know, sitting in Amazon Web Services, and provide consistent operations to all of the existing workflows used in your private clouds. The result is a complete solution regardless of where the topology sit on-prem or on the cloud.

Storage Policy Based Management

As an abstraction layer, Storage Policy Based Management (SPBM) abstracts storage services delivered by Virtual Volumes, vSAN, I/O filters, or other storage entities. Multiple partners and vendors can provide Virtual Volumes, vSAN, or I/O filters support. Rather than integrating with each individual vendor or type of storage and data service, SPBM provides a universal framework for many types of storage entities.

SPBM offers the following mechanisms:

• Advertisement of storage capabilities and data services that storage arrays and other entities, such as I/O filters, offer.

• Bi-directional communications between ESXi and vCenter Server on one side, and storage arrays and entities on the other.

• Virtual machine provisioning based on VM storage policies.

Examine the Default Storage Policy

vSAN requires that the virtual machines deployed on the vSAN Datastore are assigned at least one storage policy. When provisioning a virtual machine, if you do not explicitly assign a storage policy to the virtual machine the vSAN Default Storage Policy is assigned to the virtual machine.

The default policy contains vSAN rule sets and a set of basic storage capabilities, typically used for the placement of virtual machines deployed on vSAN Datastore.

vSAN Default Storage Policy Specifications

The following characteristics apply to the **vSAN Default Storage Policy**.

• The vSAN default storage policy is assigned to all virtual machine objects if you do not assign any other vSAN policy when you provision a virtual machine.

• The vSAN default policy only applies to vSAN datastores. You cannot apply the default storage policy to non-vSAN datastores, such as NFS or a VMFS datastore.

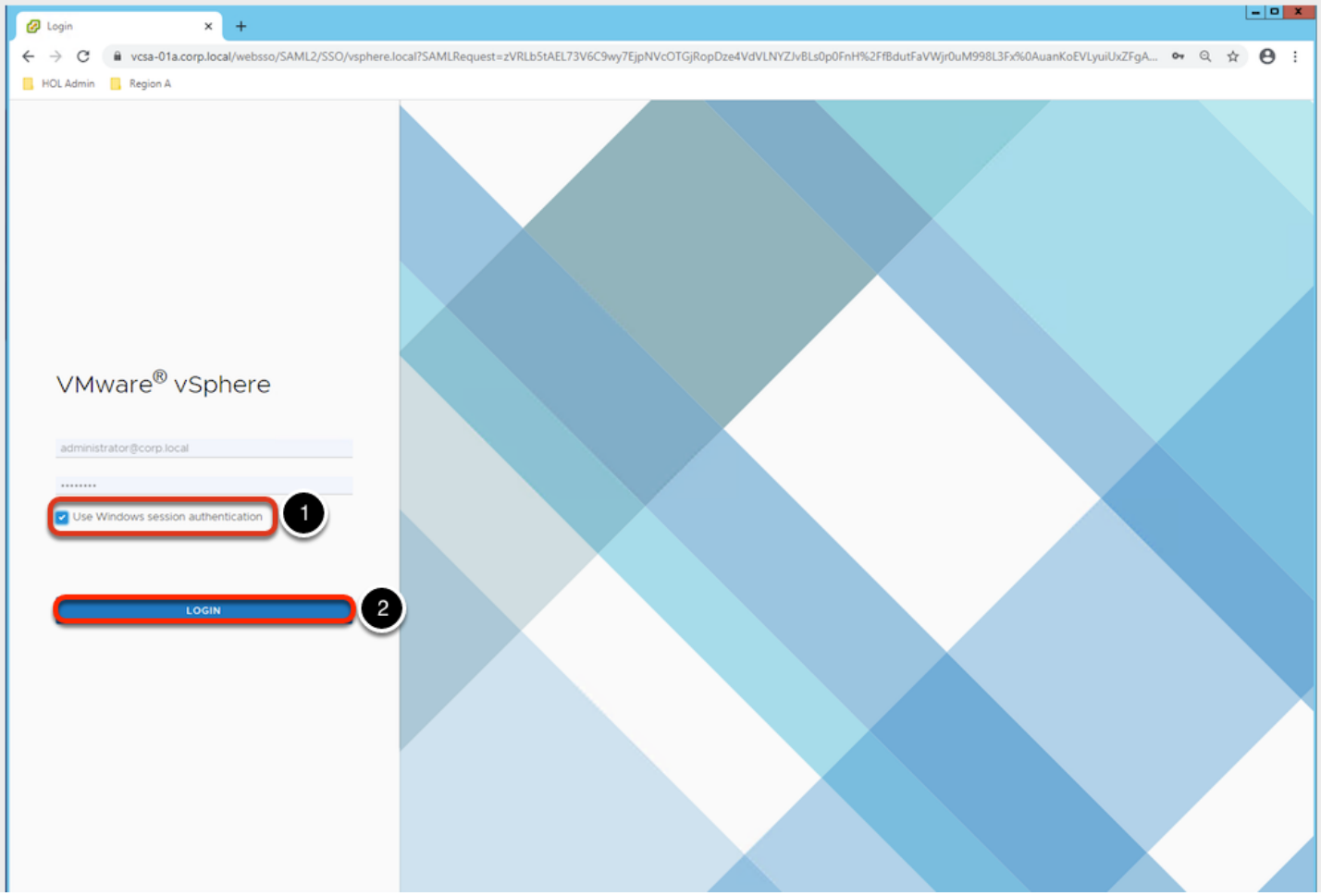
• You can clone the default policy and use it as a template to create a user-defined storage policy.

• You cannot delete the default policy.

Open Chrome Browser from Windows Quick Launch Task Bar

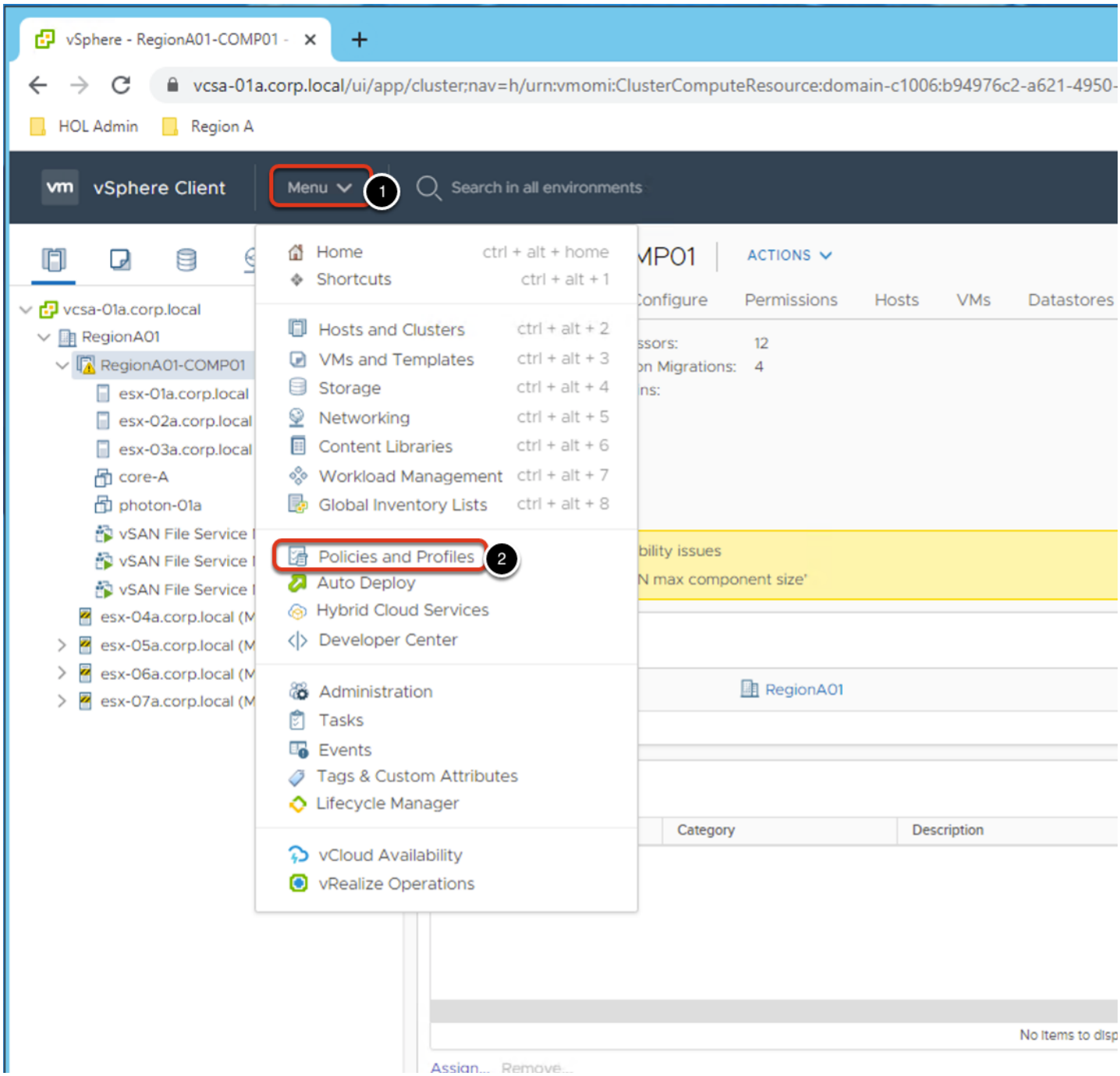
1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.

Login to vSphere Client



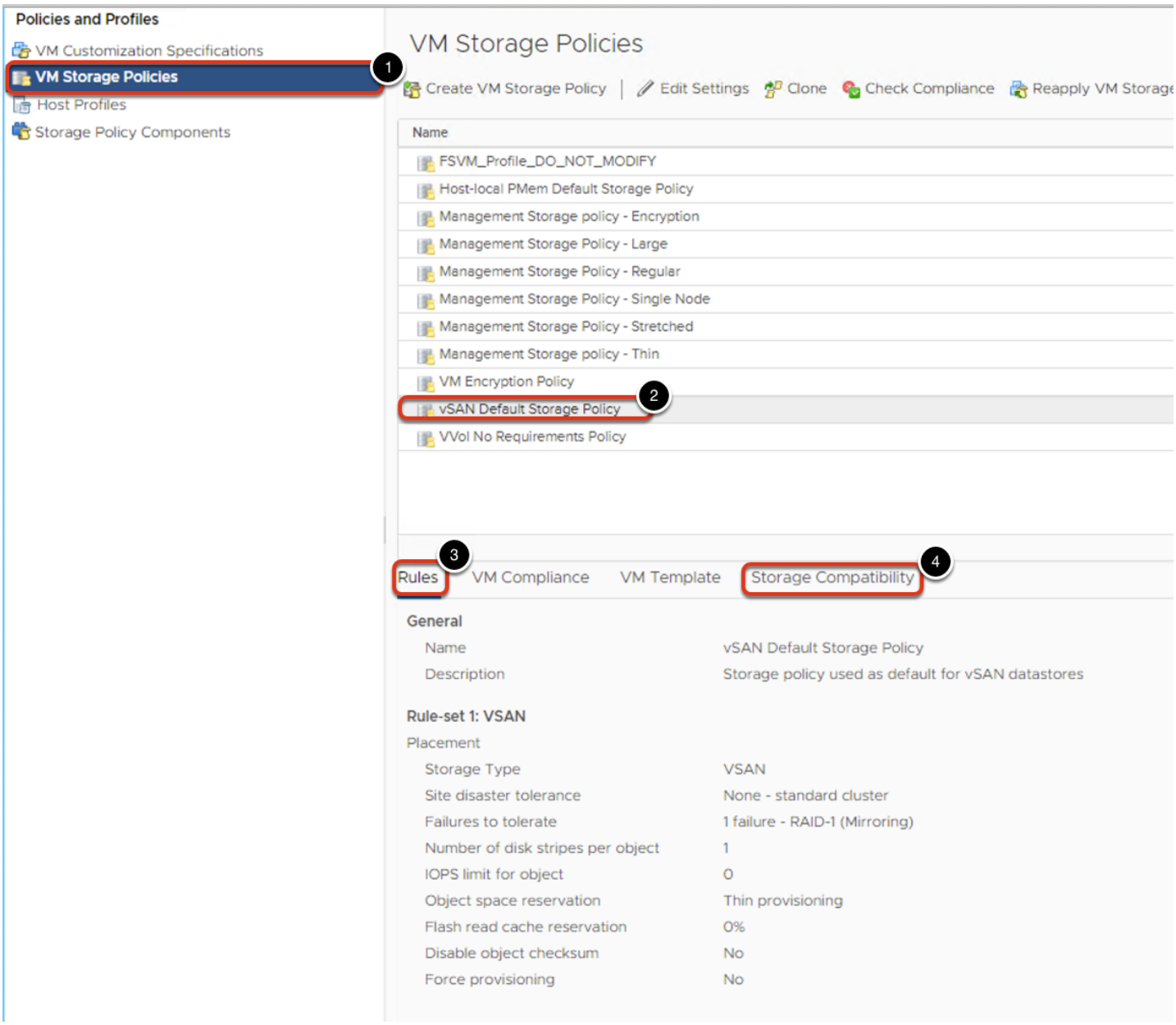
1. On the vSphere Client login screen, select "**Use Windows session authentication**"
2. Click **LOGIN**

Examine the Default Storage Policy



1. From the **Menu** page of the vSphere Client
2. Select **Policies and Profiles**

Examine the Default Storage Policy



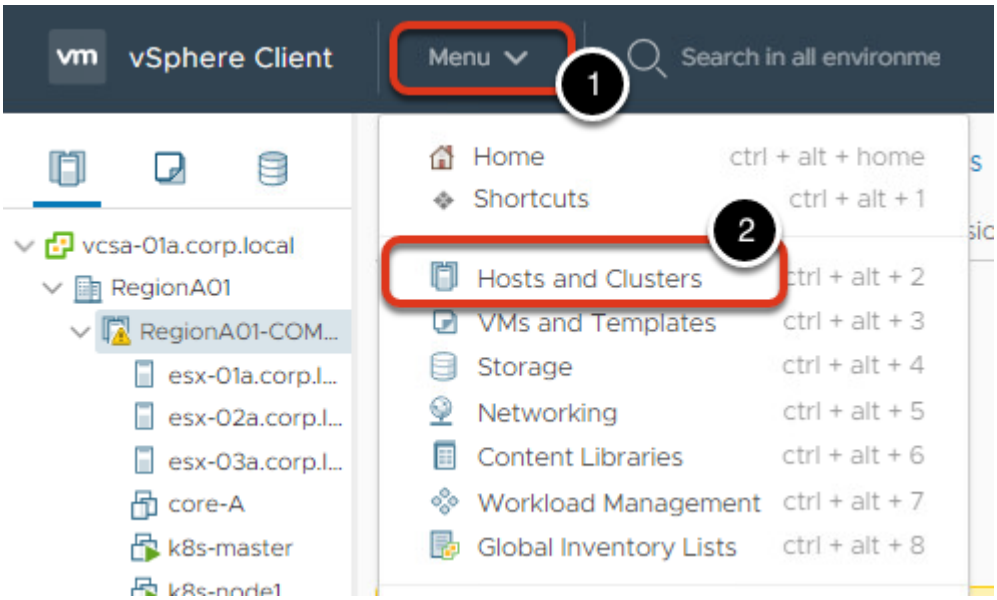
1. Select **VM Storage Policies**
2. Select the VM Storage Policy called **vSAN Default Storage Policy**
3. Select **Rules**

The default rules for the Storage Policy are displayed.

4. Select **Storage Compatibility**

Here we can see that the **vsanDatastore** is compatible with this storage policy. (not picture)

Deploy VM with Default Policy

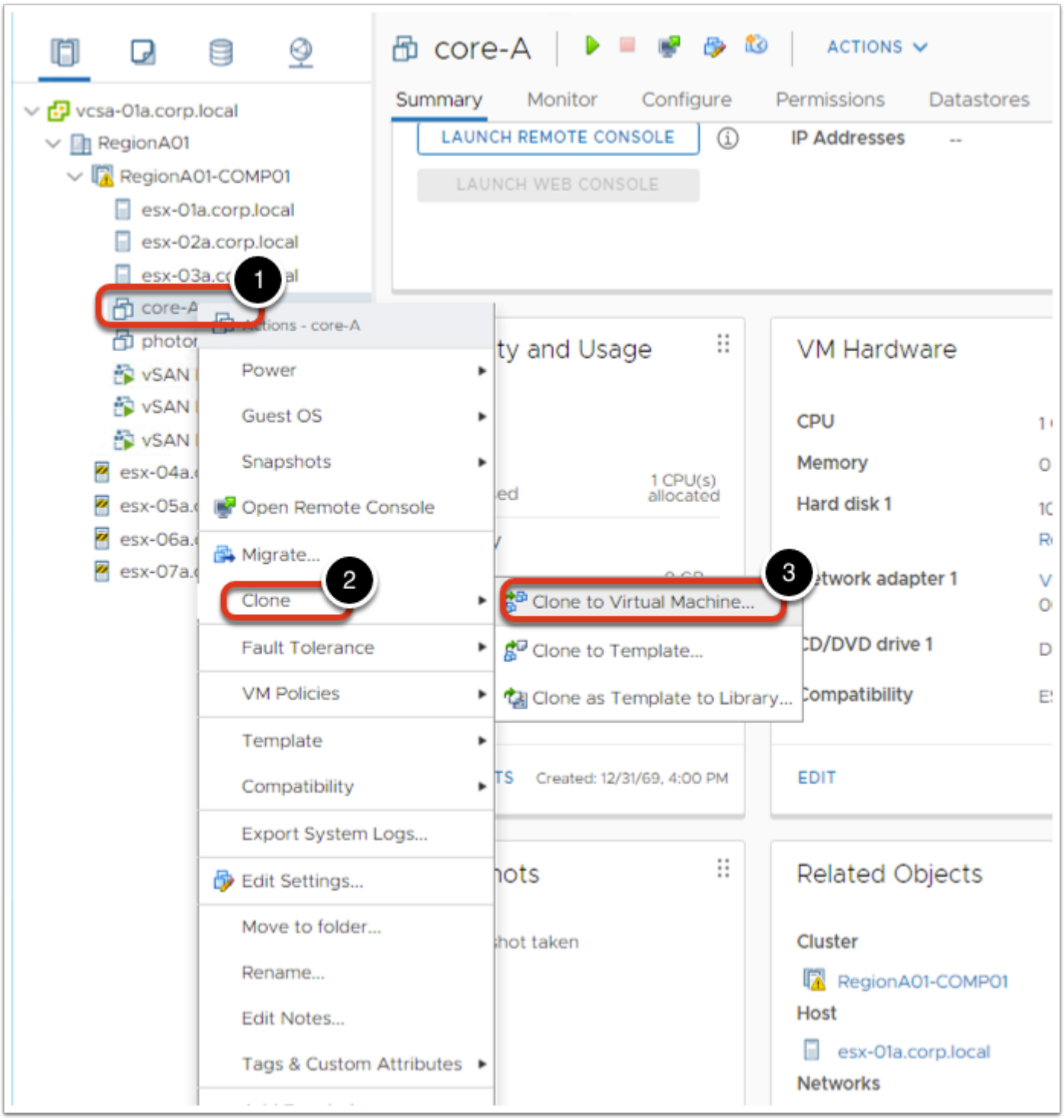


We will now clone a VM and apply the Default Storage Policy

1. Select **Menu**
2. Select **Hosts and Clusters**

Deploy VM with Default Policy

We will clone the VM called **core-A** (which currently resides on an iSCSI VMFS datastore) to the vSAN Datastore and apply the Default Storage Policy.

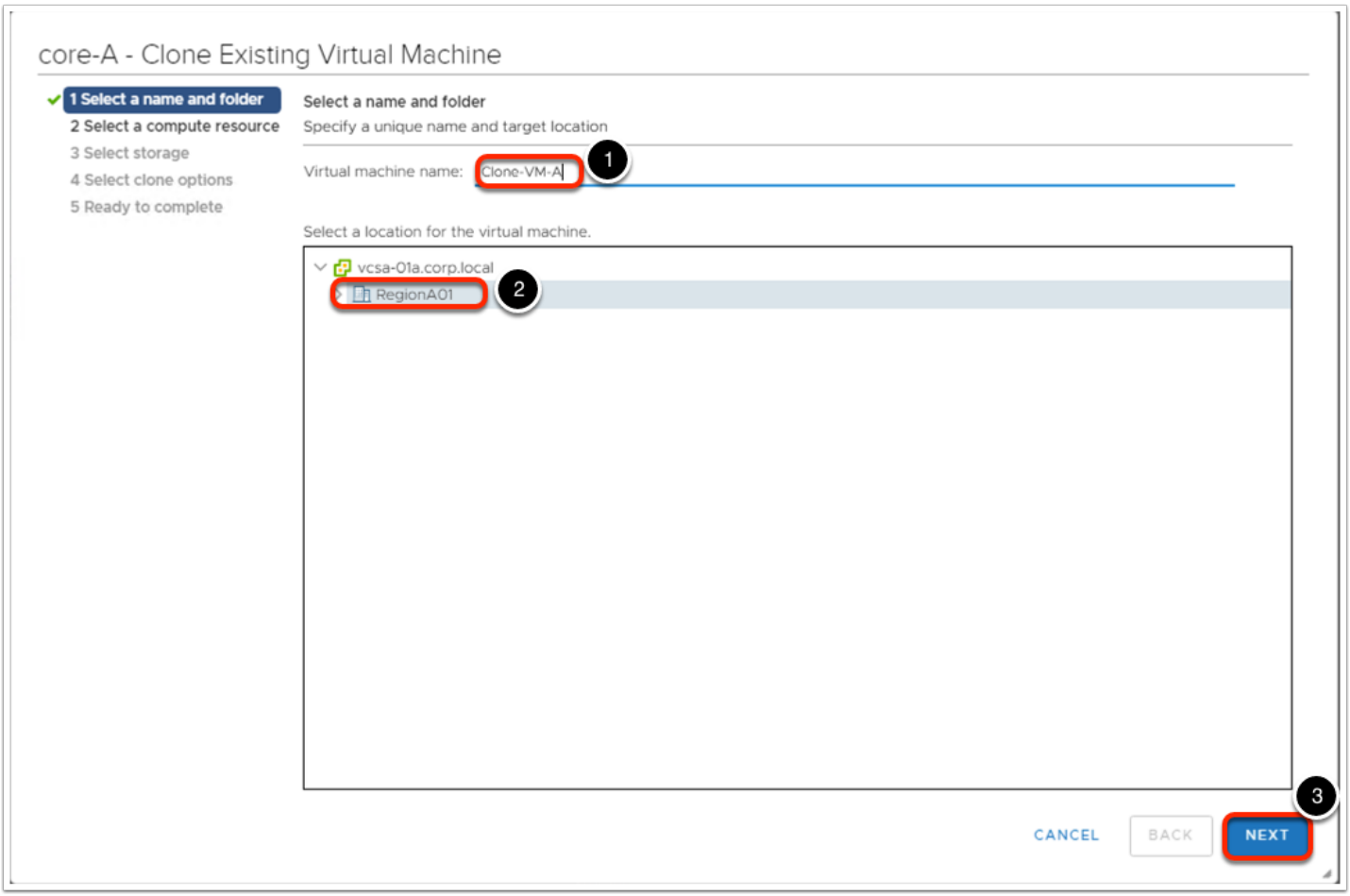


1. Expand the ESXi Cluster called **RegionA01-COMP01** and right click the VM called

core-A

2. Select **Clone**
3. Select **Clone to Virtual Machine**

Deploy VM with Default Policy

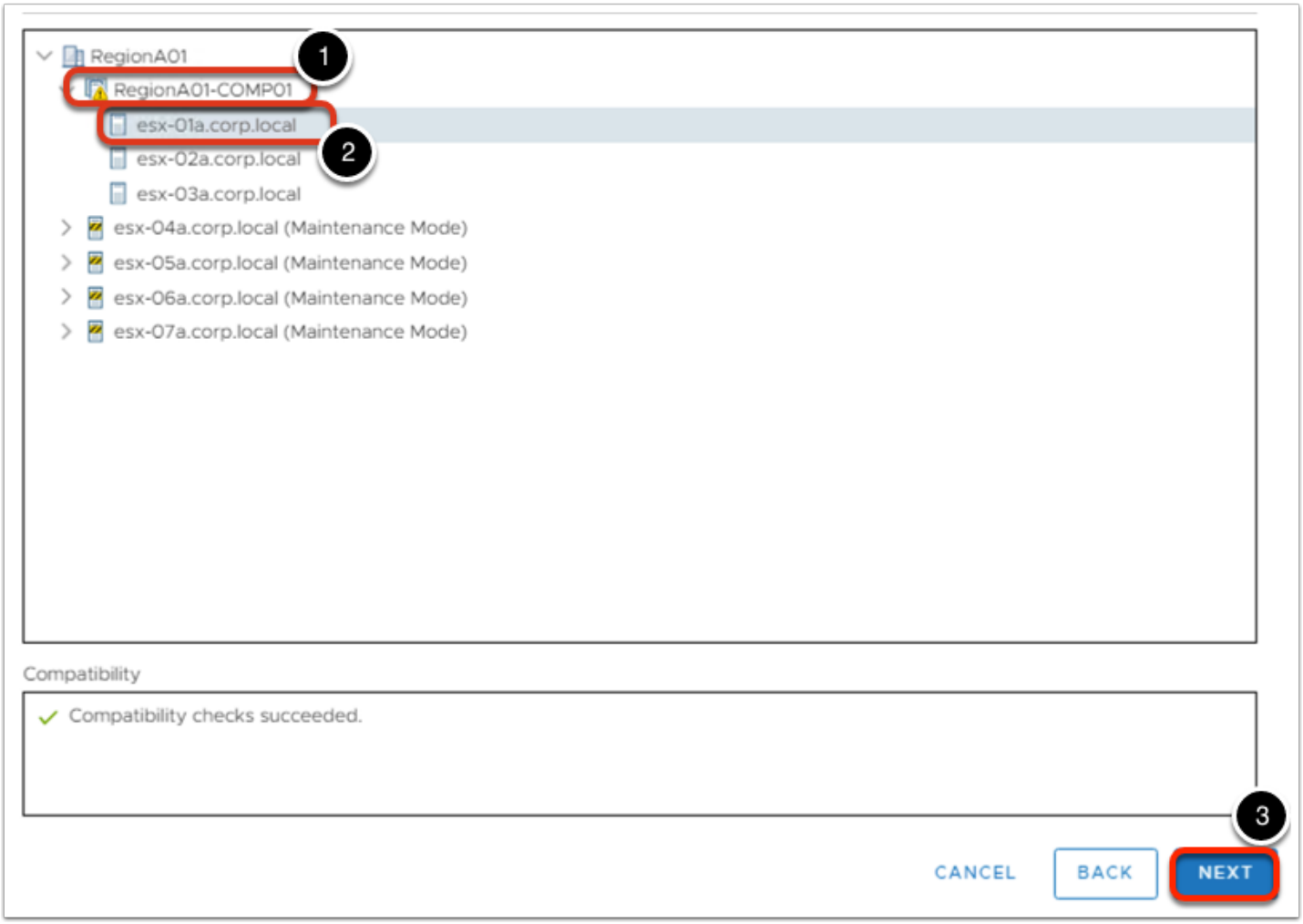


1. Give the Virtual Machine a name:

```
Clone-VM-A
```

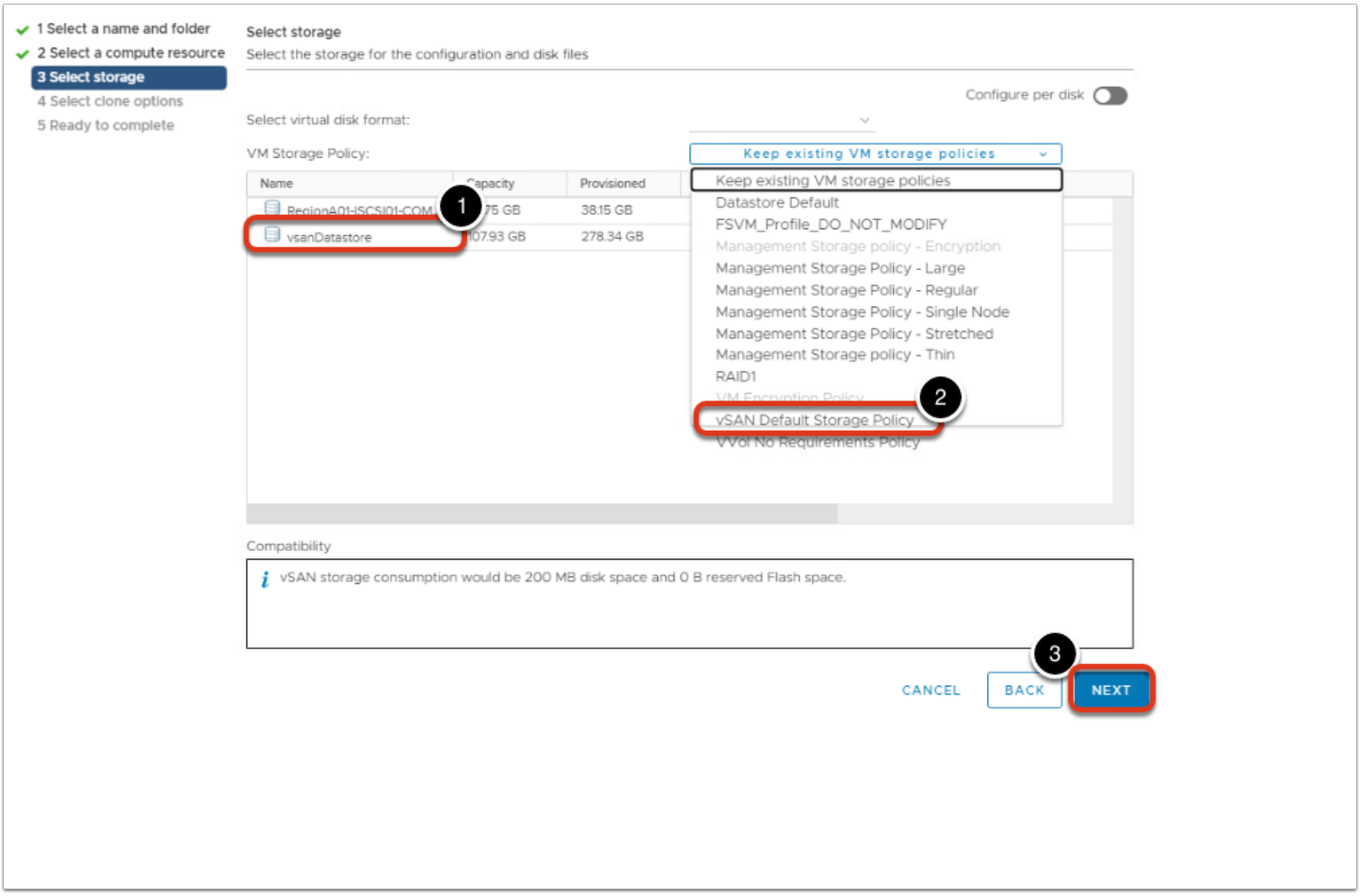
2. Select **vcsa-01a.corp.local** > **RegionA01**
3. then Click **Next**

Deploy VM with Default Policy



1. Expand the Compute resource called **RegionA01-COMP01**
2. Select the ESXi host called **esx-01a.corp.local**
3. Click **NEXT**

Deploy VM with Default Policy



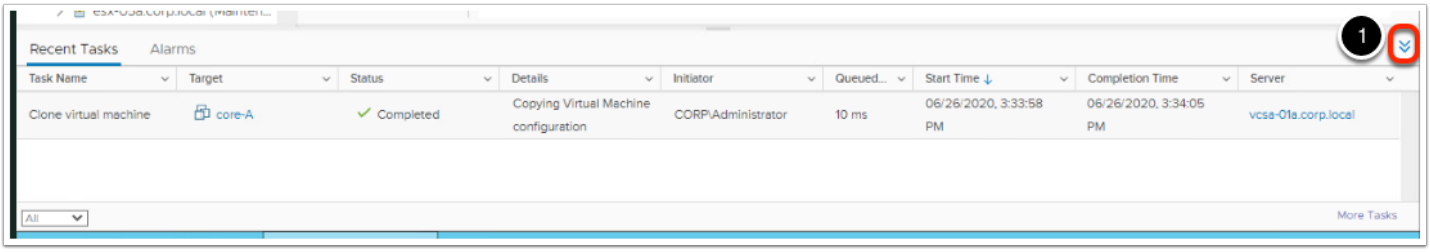
1. Click on **vsanDatastore**
2. For the VM Storage Policy dropdown, select **vSAN Default Storage Policy**

The resulting list of compatible datastores will be presented, in our case the vsanDatastore. In the lower section of the screen we can see the vSAN storage consumption would be 200.00 MB disk space and 0.00 B reserved Flash space.

Since we have a VM with 100 MB disk and the Default Storage Policy, the VSAN disk consumption will be 200.00 MB disk.

3. Click **NEXT** then, Click **NEXT** on the **Select clone options** (not picture), then Click **FINISH** (not picture)

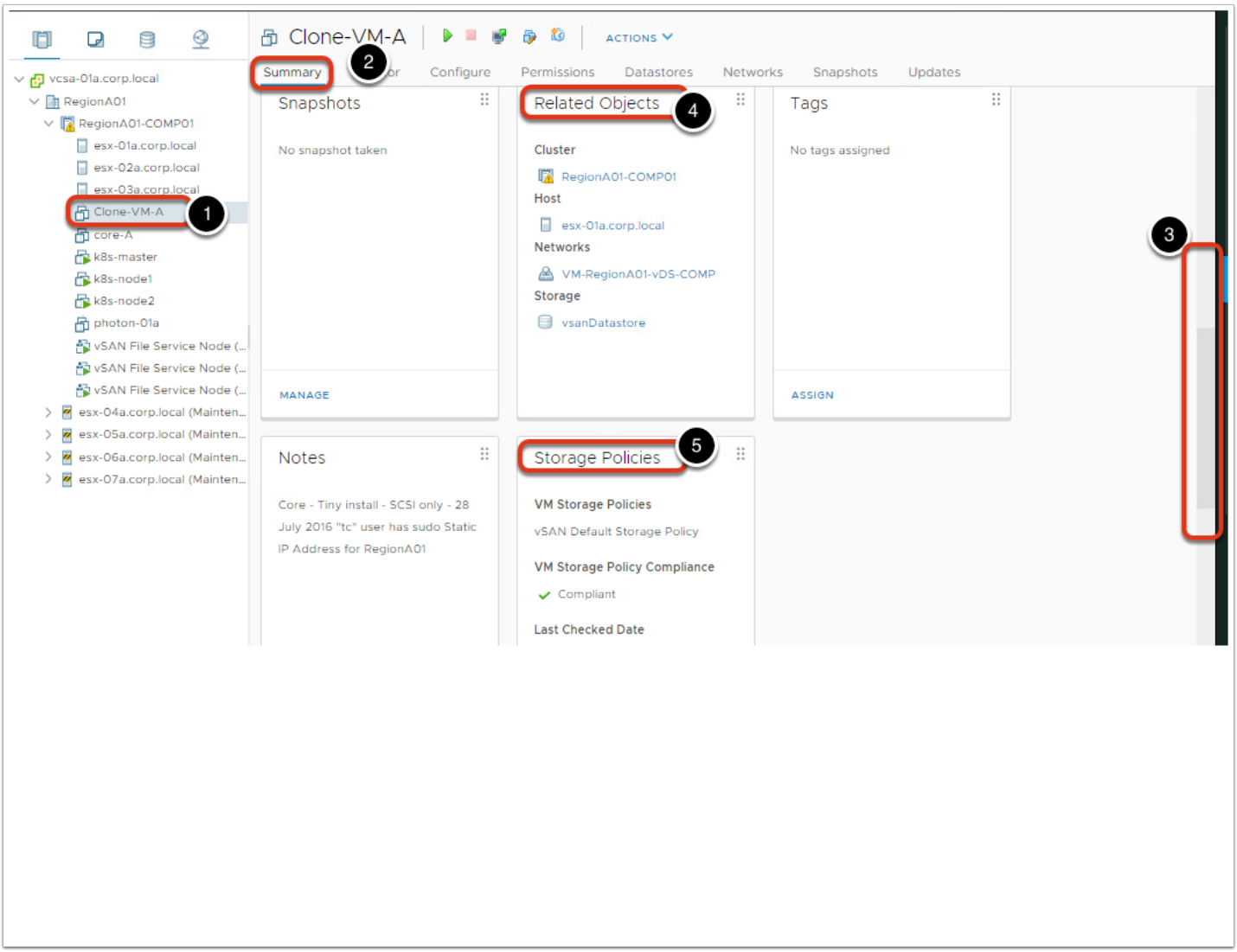
Deploy VM with Default Policy



Wait for the Clone operation to complete.

1. Check the **Recent Tasks** for a status update on the Clone virtual machine task.

Verify VM has Default Storage Policy



Once the clone operation has completed,

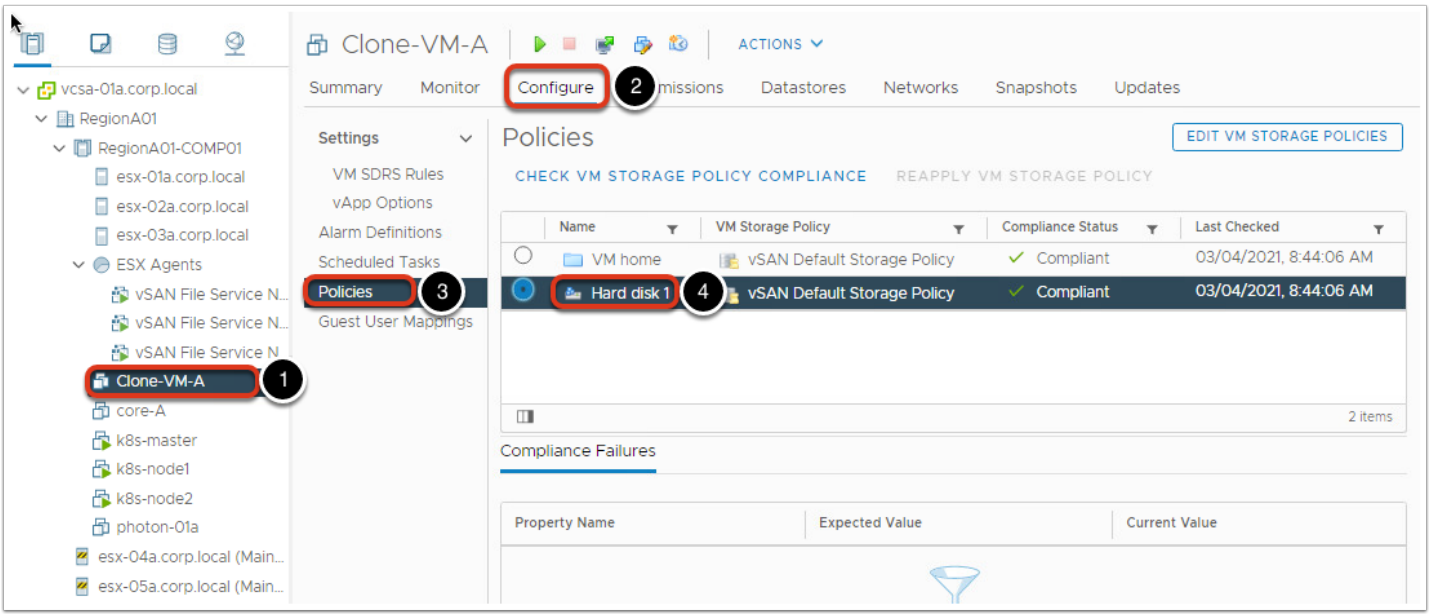
1. Select the VM called **Clone-VM-A**
2. Select **Summary**
3. Scroll down
4. View **Related Objects**

The VM is now residing on the **vsanDatastore**

5. Scroll down and View **Storage Policies**

Here we can see that the VM Storage Policy for this VM is set to **vSAN Default Storage Policy** and the policy is **compliant**.

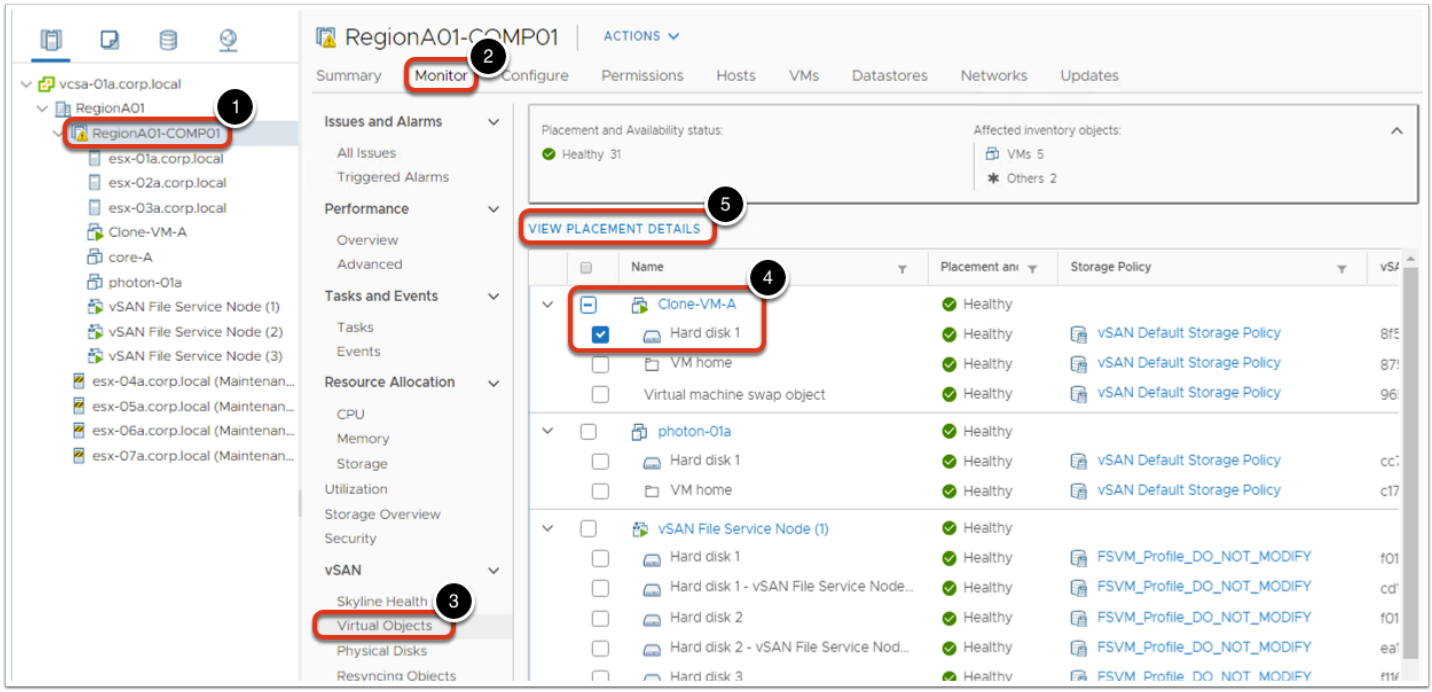
VM Disk Policies



1. Select the VM called **Clone-VM-A**
2. Select **Configure**
3. Select **Policies**
4. Select **Hard Disk 1**

Here we can see the VM Storage Policy that is applied to **VM Home Object** and the **Hard Disk Object.**

VM Disk Policies

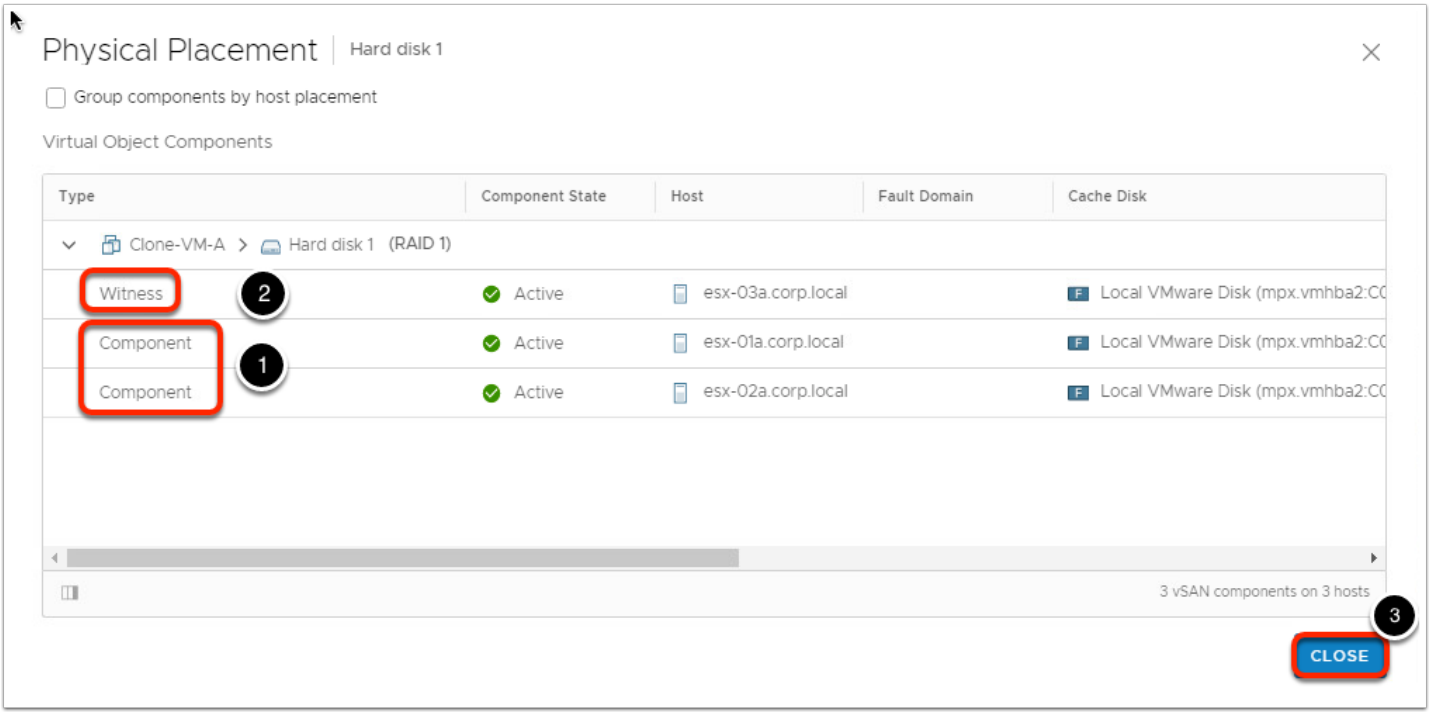


1. Select the **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Virtual Objects**
4. Select **Clone-VM-A > Hard disk 1**

Verify that the Placement and Availability is **Healthy** and **vSAN Default Storage Policy** is applied.

5. Click **View Placement Details**

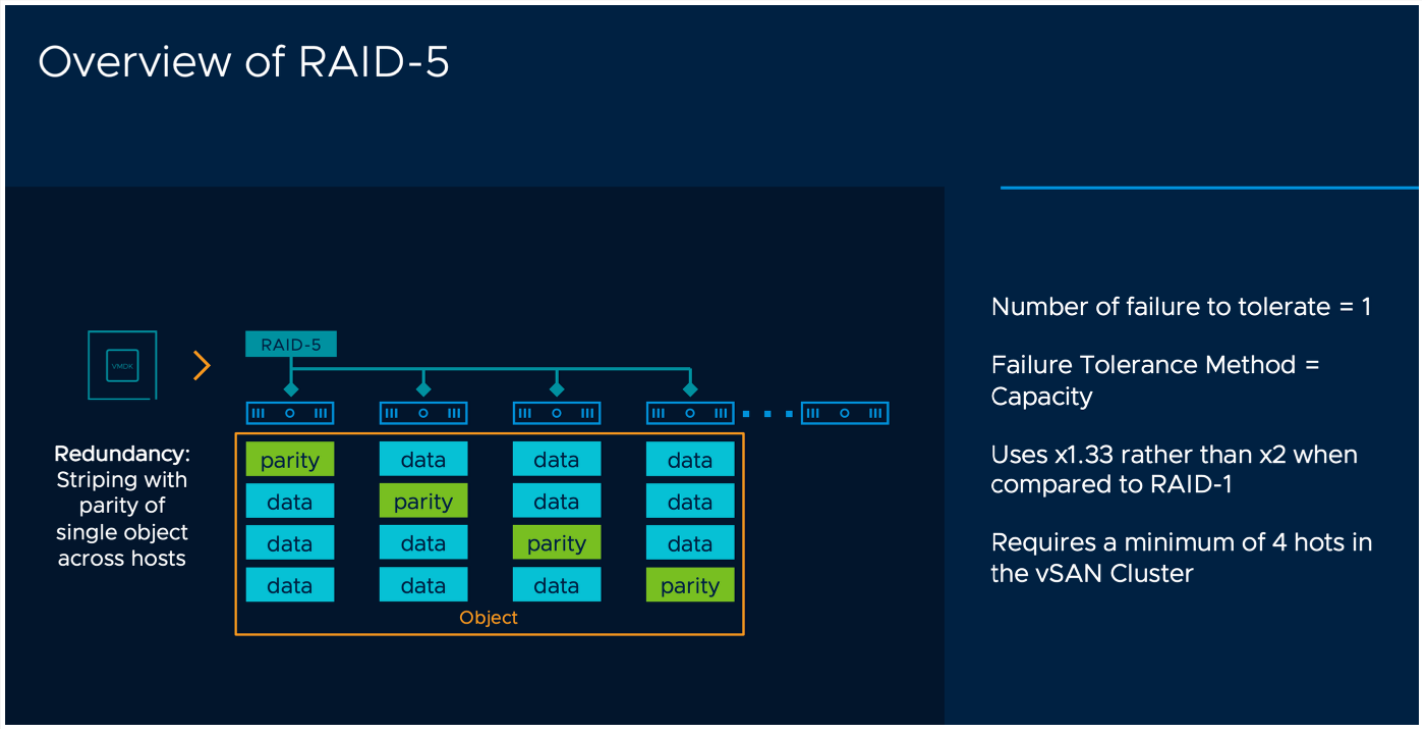
VM Disk Policies



Here we can see the Component layout for the Hard Disk.

1. There are 2 Components spread across 2 different ESXi Hosts
2. The Witness component on another ESXi host.
3. Click **CLOSE**

Scaling out the vSAN Environment



Note that there is a requirement on the number of hosts needed to implement RAID-5 or RAID-6 configurations on vSAN.

For RAID-5, a minimum of 4 hosts are required; for RAID-6, a minimum of 6 hosts are required.

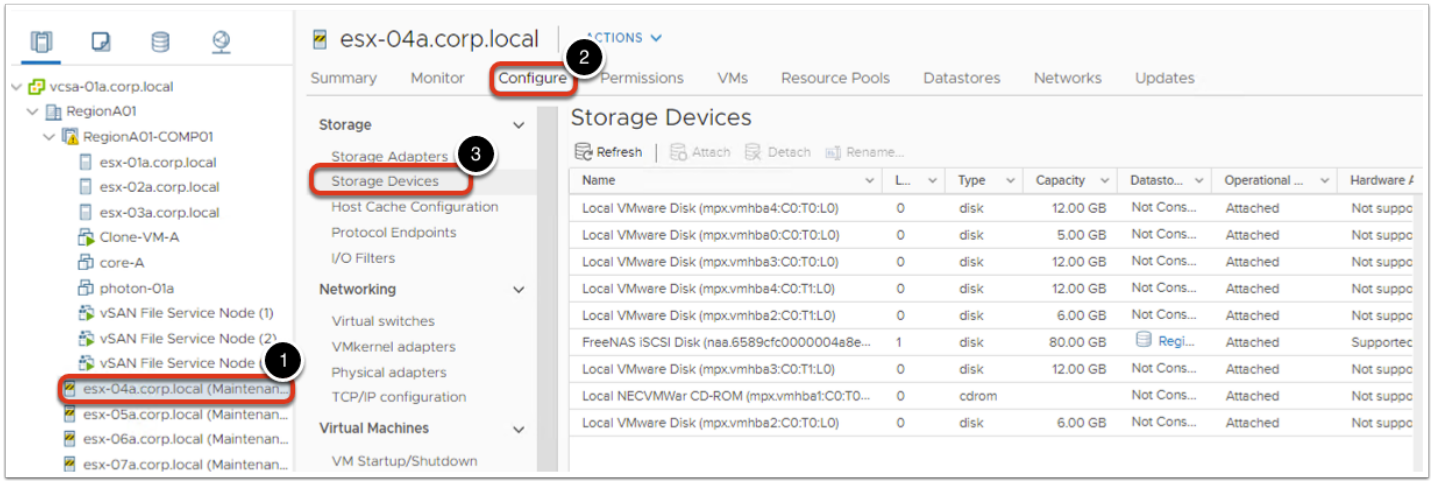
The objects are then deployed across the storage on each of the hosts, along with a parity calculation. The configuration uses distributed parity, so there is no dedicated parity disk. When a failure occurs in the cluster, and it impacts the objects that were deployed using RAID-5 or RAID-6, the data is still available and can be calculated using the remaining data and parity if necessary.

A new policy setting has been introduced to accommodate the new RAID-5/RAID-6 configurations.

This new policy setting is called Failure Tolerance Method. This policy setting takes two values: performance and capacity. When it is left at the default value of performance, objects continue to be deployed with a RAID-1/mirror configuration for the best performance. When the setting is changed to capacity, objects are now deployed with either a RAID-5 or RAID-6 configuration.

The RAID-5 or RAID-6 configuration is determined by the number of failures to tolerate setting. If this is set to 1, the configuration is RAID-5. If this is set to 2, then the configuration is a RAID-6.

Lab Environment Review - Compute



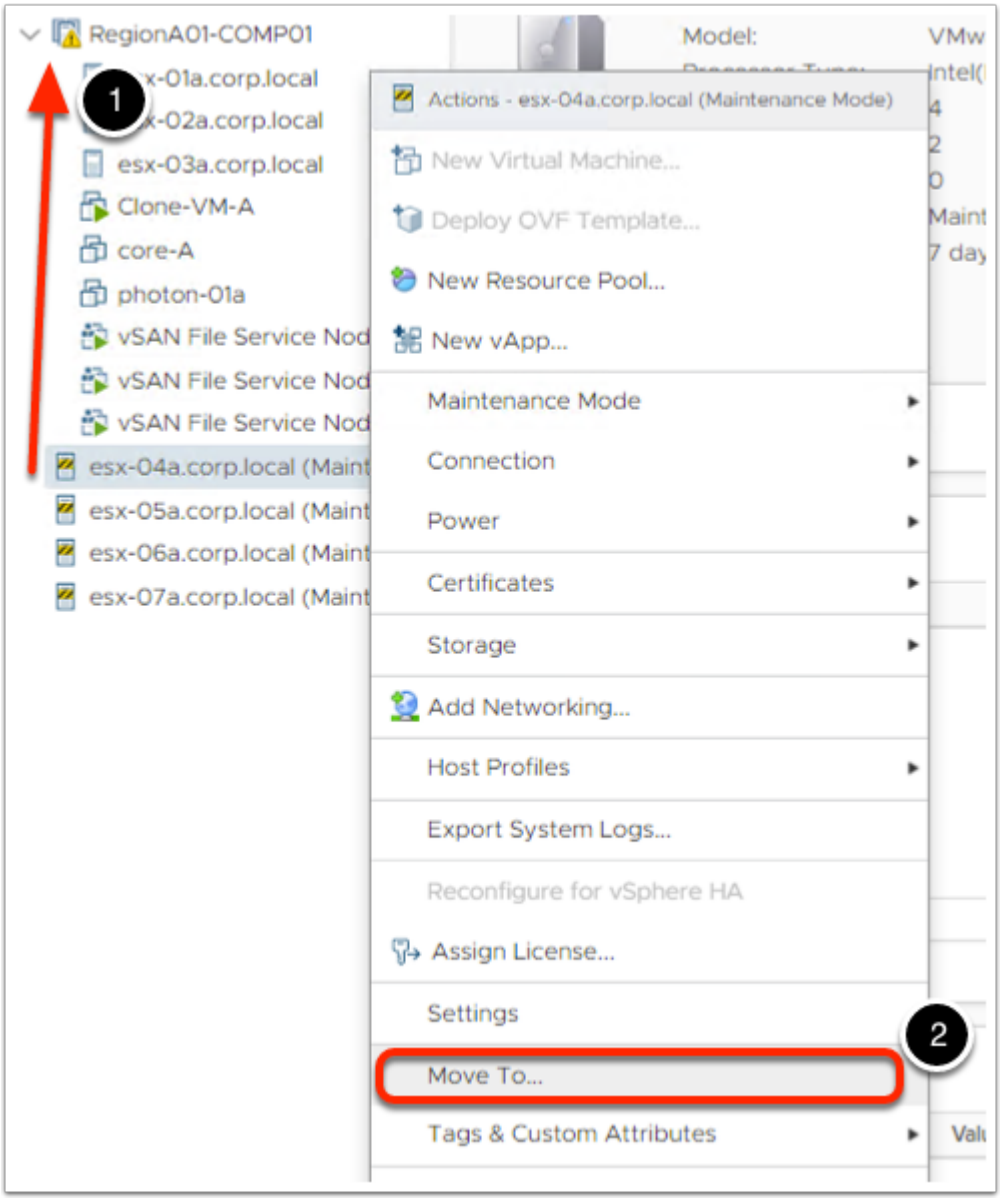
Lets have a look at how our cluster currently looks.

There are currently three hosts in the cluster, and there are additional hosts not in the cluster.

1. Select the ESXi host called **esx-04a.corp.local**
2. Select **Configure**
3. Select **Storage > Storage Devices**

On the ESXi host you can see that we have some devices that we can use to expand our vSAN Datastore (there are multiple 6 GB Flash and the 12 GB Flash devices).

Add Additional nodes to cluster



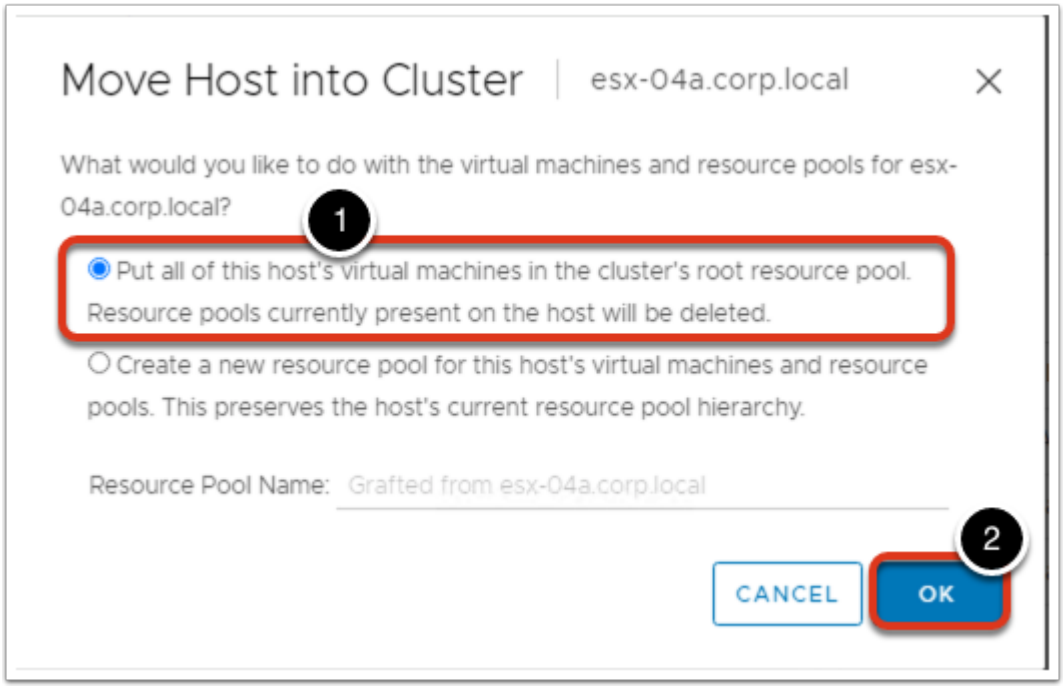
We are now going to add the **esx-04a.corp.local** to the vSAN Cluster.

1. Drag and drop **esx-04a.corp.local** into **RegionA01-COMP01** cluster

If the Drag and Drop does not seem to be working for you,

2. Right click the ESXi host called **esx-04a.corp.local** and select Move to .... [new dialogue box] Select the cluster called **RegionA01-COMP01**.

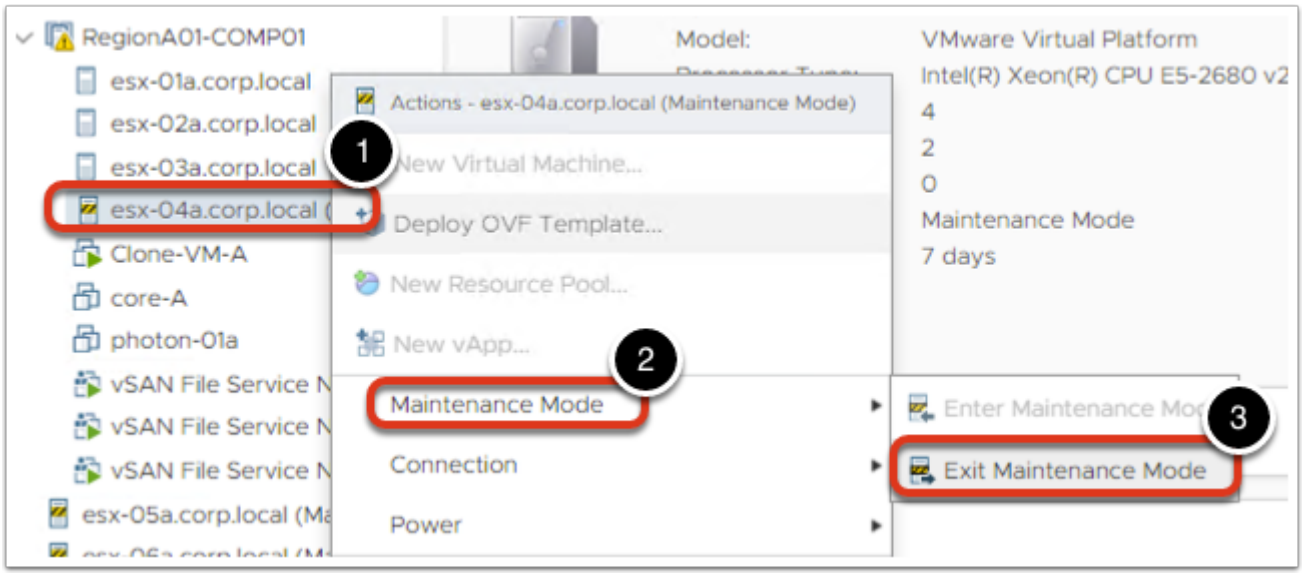
Move Host into Cluster



Confirming that we want to move host into the vSAN Cluster and resource pool.

1. Select "**Put all of this host's virtual machines in the cluster's root resource pool. Resource pools currently present on the host will be deleted**."
2. Click **OK**

Take Host out of Maintenance Mode

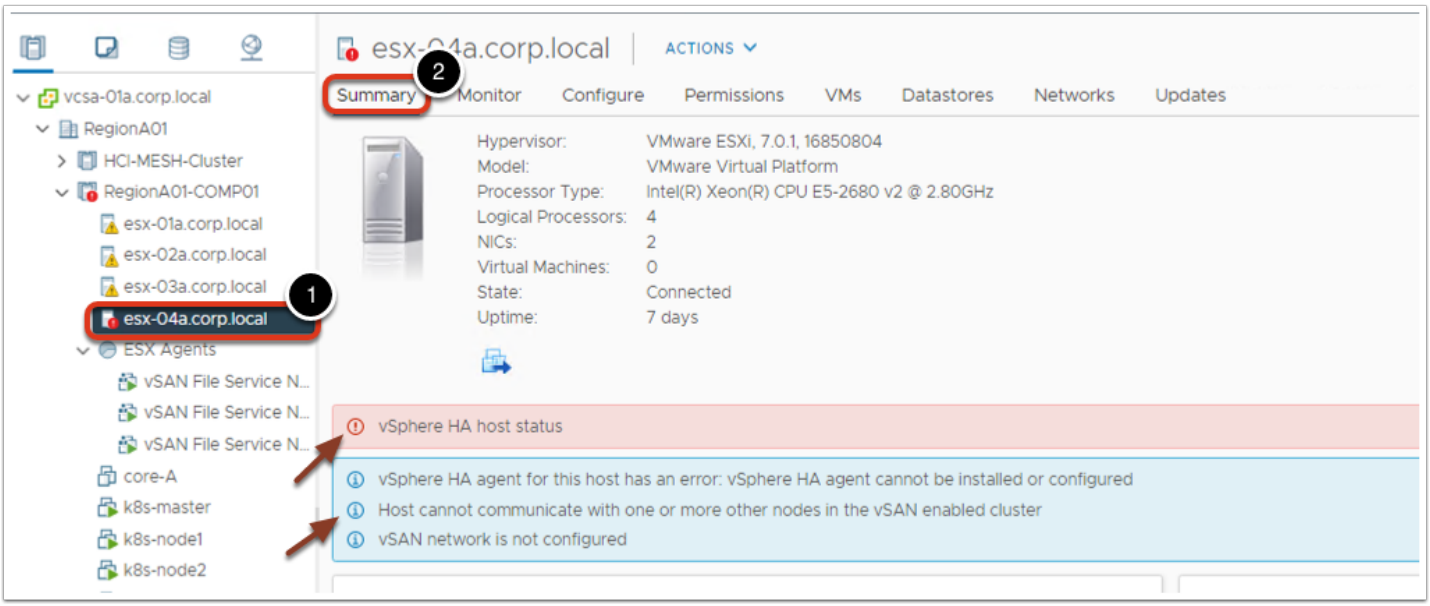


The ESXi host is still in Maintenance Mode.

1. Right click the ESXi host called **esx-04a.corp.local**
2. Select **Maintenance Mode**
3. Select **Exit Maintenance Mode**

If the **Exit Maintenance Mode** option is not available, refresh the vSphere Client and try the operation again.

Configure vSAN Networking

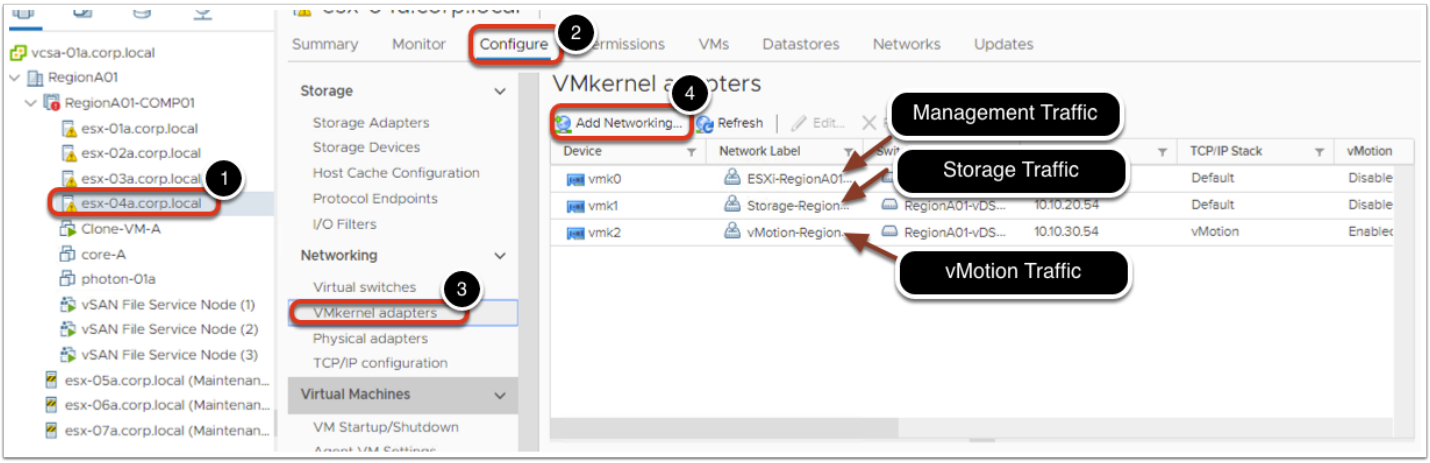


Now that we have taken the host out of maintenance mode, we can see a few informational messages on the Summary screen.

1. Select the ESXI host called **esx-04a.corp.local**
2. Select **Summary**

These messages are telling us that we have hosts in the vSAN Cluster that cannot communicate with each other over the vSAN Network. The other message is saying the host is not HA but we will remediate it later.

Configure vSAN Networking



Lets review the current state of the Networking on the ESXi host.

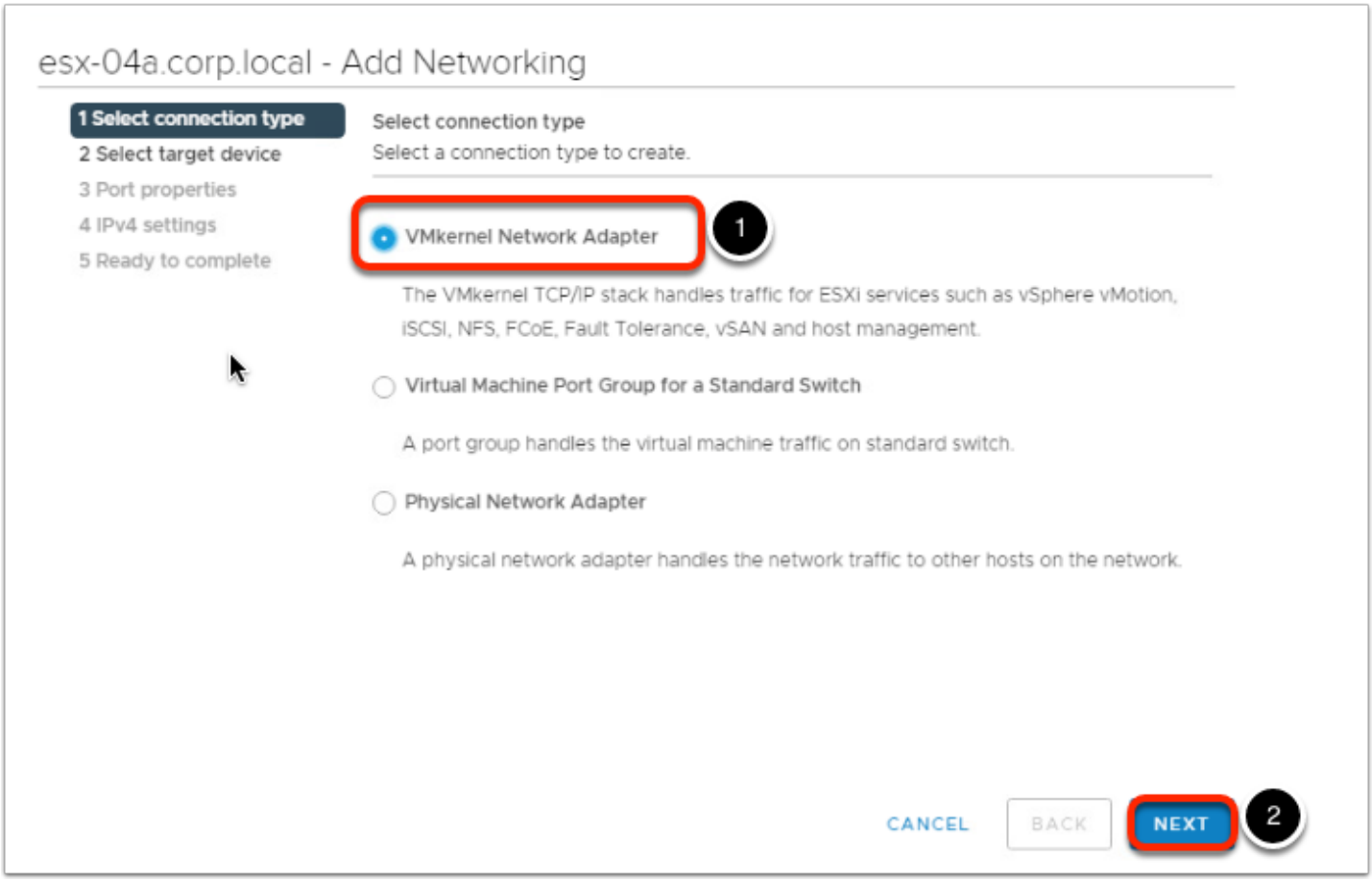
1. Select the ESXi host called **esx-04a.corp.local**
2. Select **Configure**
3. Select **Networking > VMkernel adapters**

There are 3 VMkernel adapters configured, one for Management traffic, one for traditional Storage traffic and one for vMotion traffic.

We will now configure a VMkernel adapter for the vSAN Network traffic for this host.

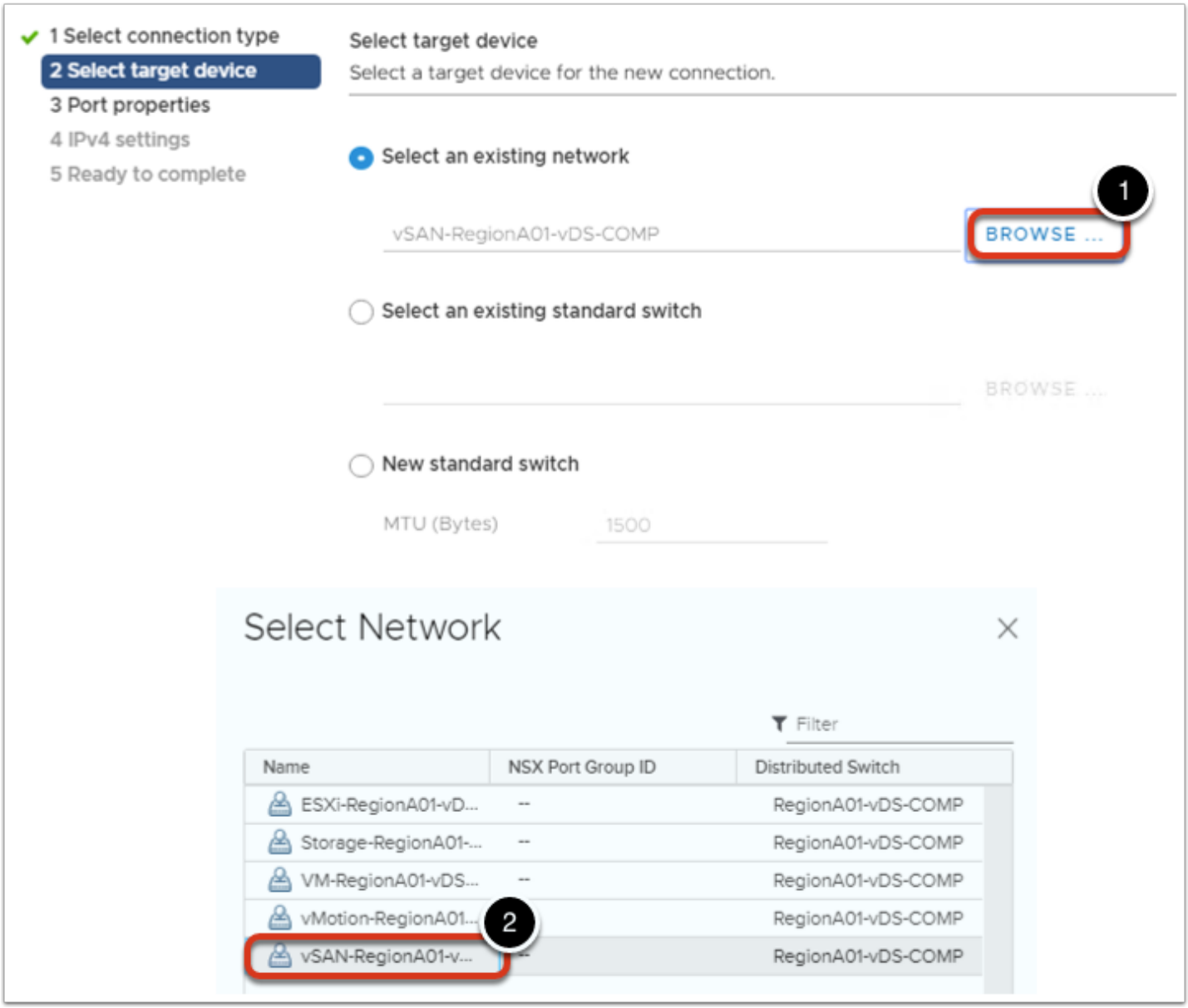
4. Select **Add Networking**

Configure vSAN Networking



1. Select **VMkernel Network Adapter**
2. Click **NEXT**

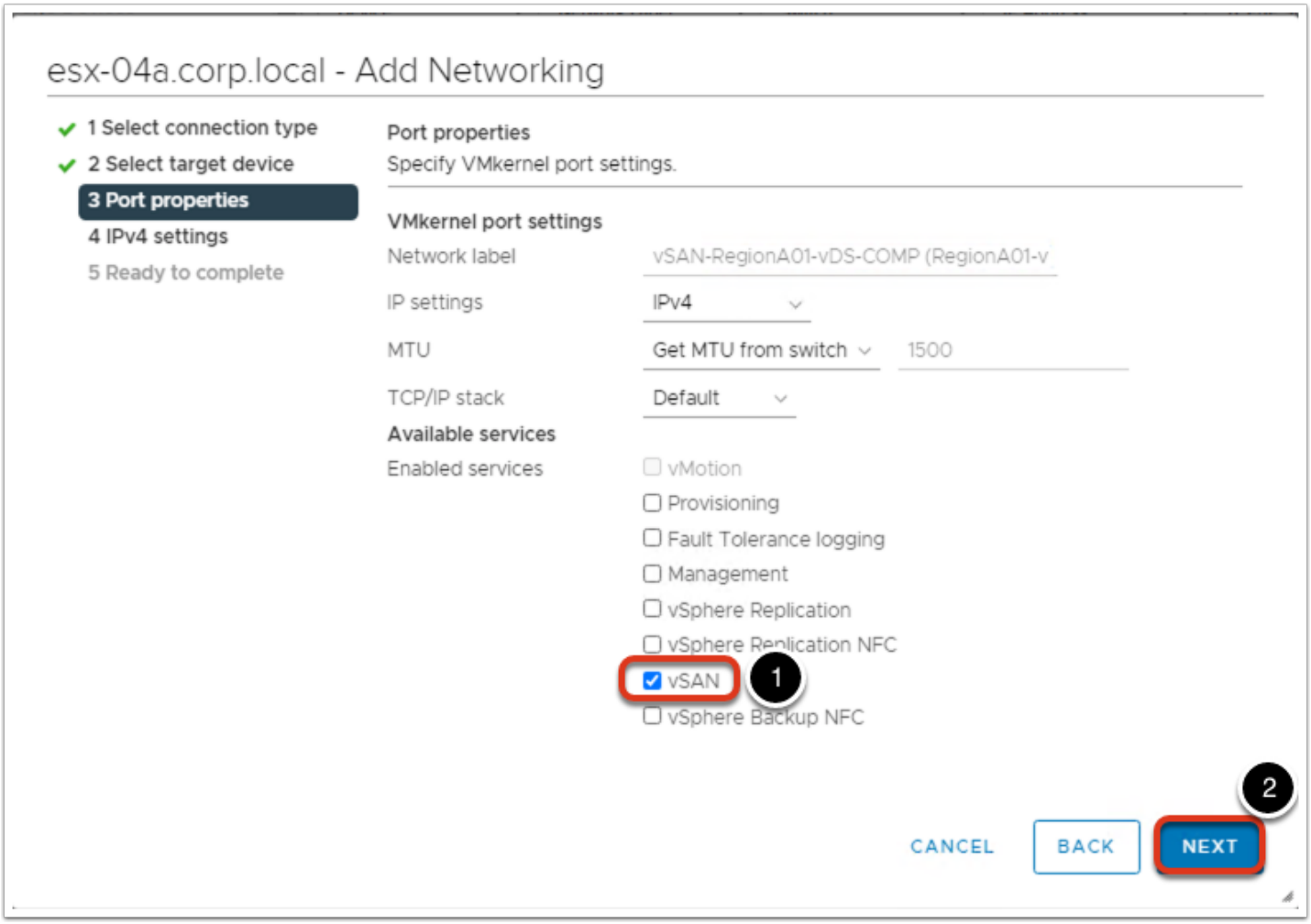
Configure vSAN Networking



1. Click the **Browse** button
2. Select the VMkernel adapter called **vSAN-RegionA01-vDS-COMP**

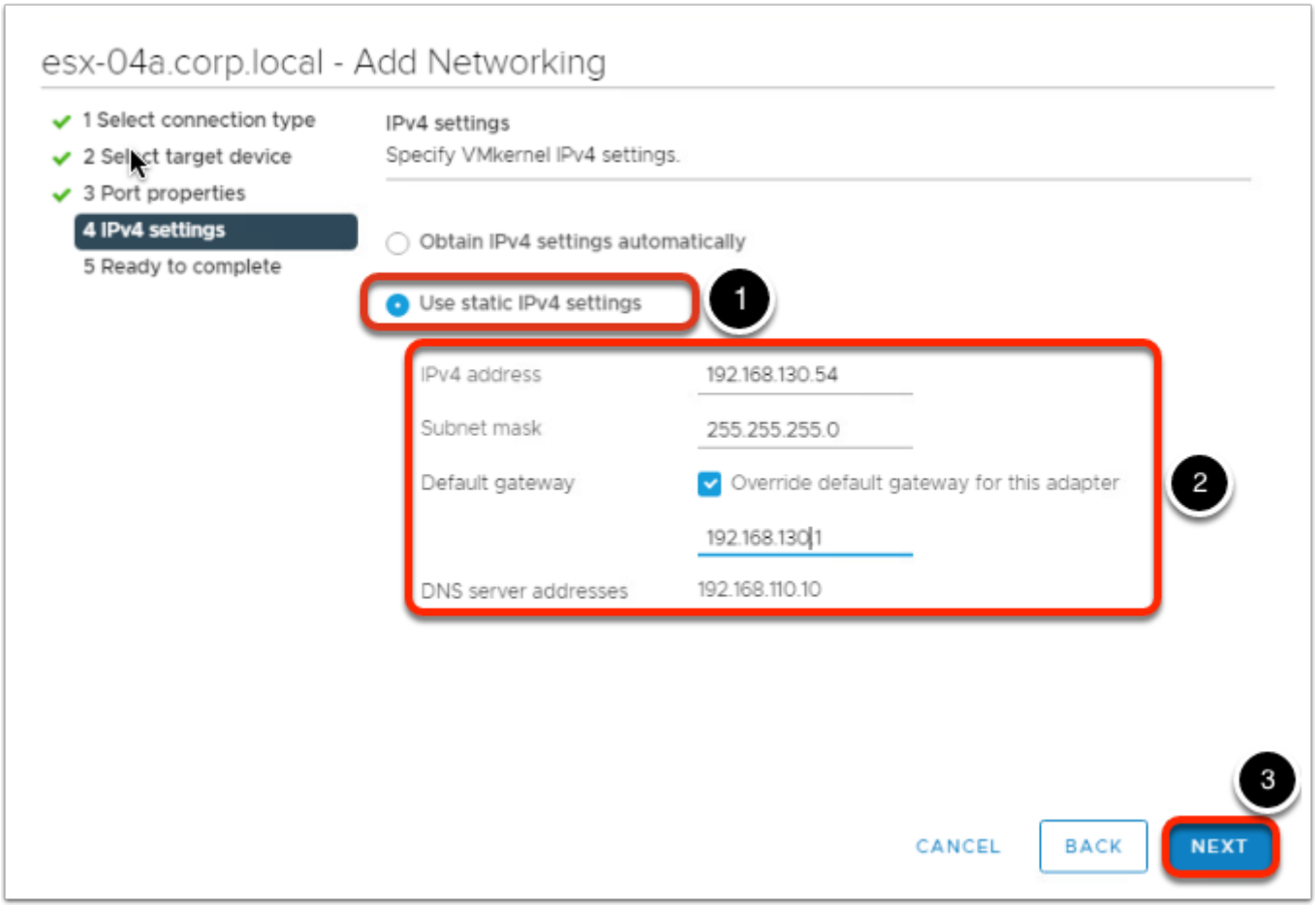
- Click **OK** then Click **NEXT**

Configure vSAN Networking



1. Enable the **vSAN** service
2. Click **NEXT**

Configure vSAN Networking

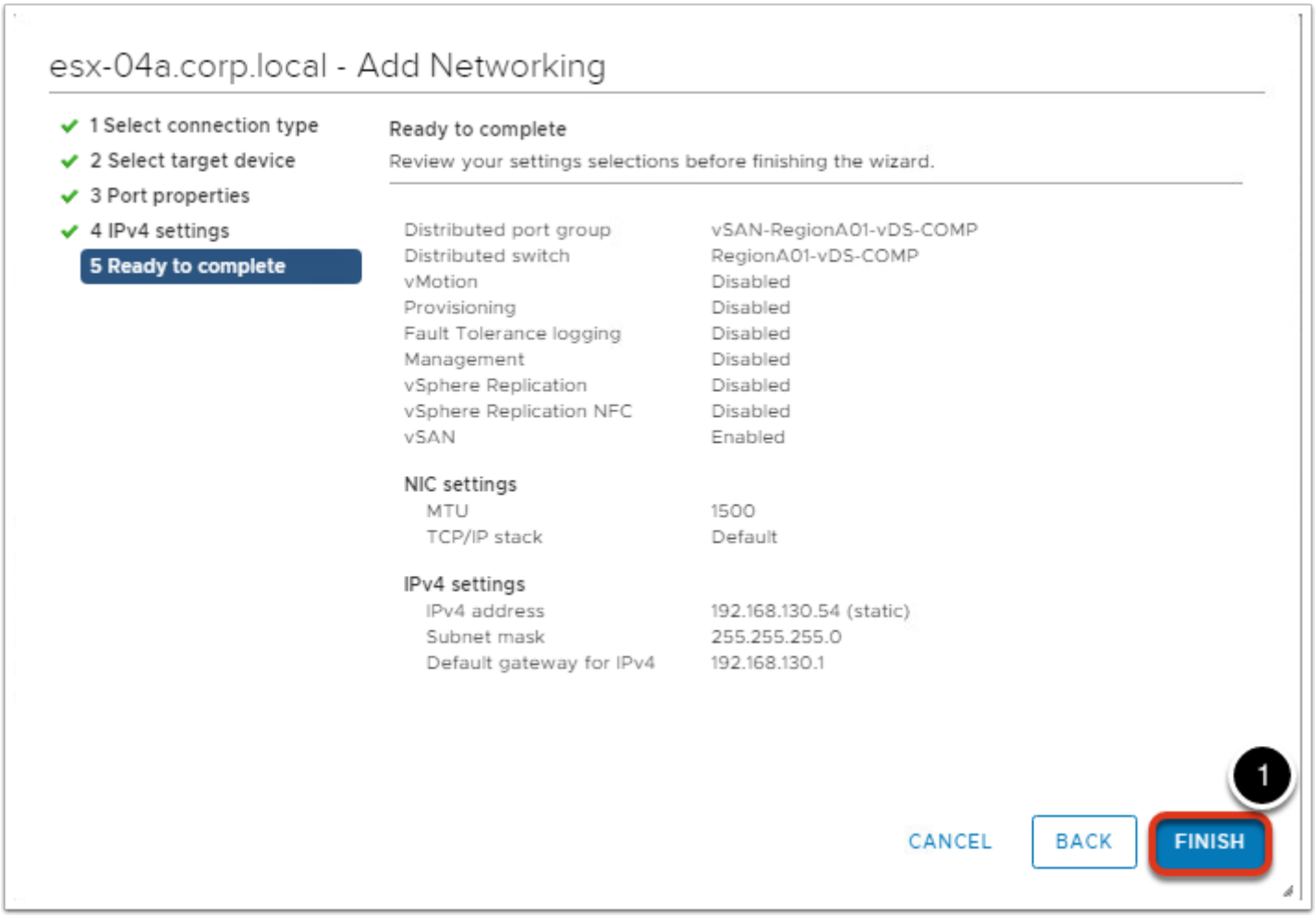


1. Select **Use static IPv4 settings**
2. Enter the following information for the Network configuration :

```
IPv4 address : 192.168.130.54
Subnet mask : 255.255.255.0
Override default gateway for this adapter : Enabled
Default gateway : 192.168.130.1
DNS server addresses : 192.168.110.10
```

3. Click **NEXT**

Configure vSAN Networking



Review the configuration settings

1. Click **FINISH**

Verify vSAN Networking

1. Select the VMkernel adapter called **vSAN-RegionA01-vDS-COMP**
2. Review the properties of the VMkernel adapter. Verify that the **vSAN Service** is enabled

After a while the Alarms should disappear from the host.

Create a Disk Group on a New Host

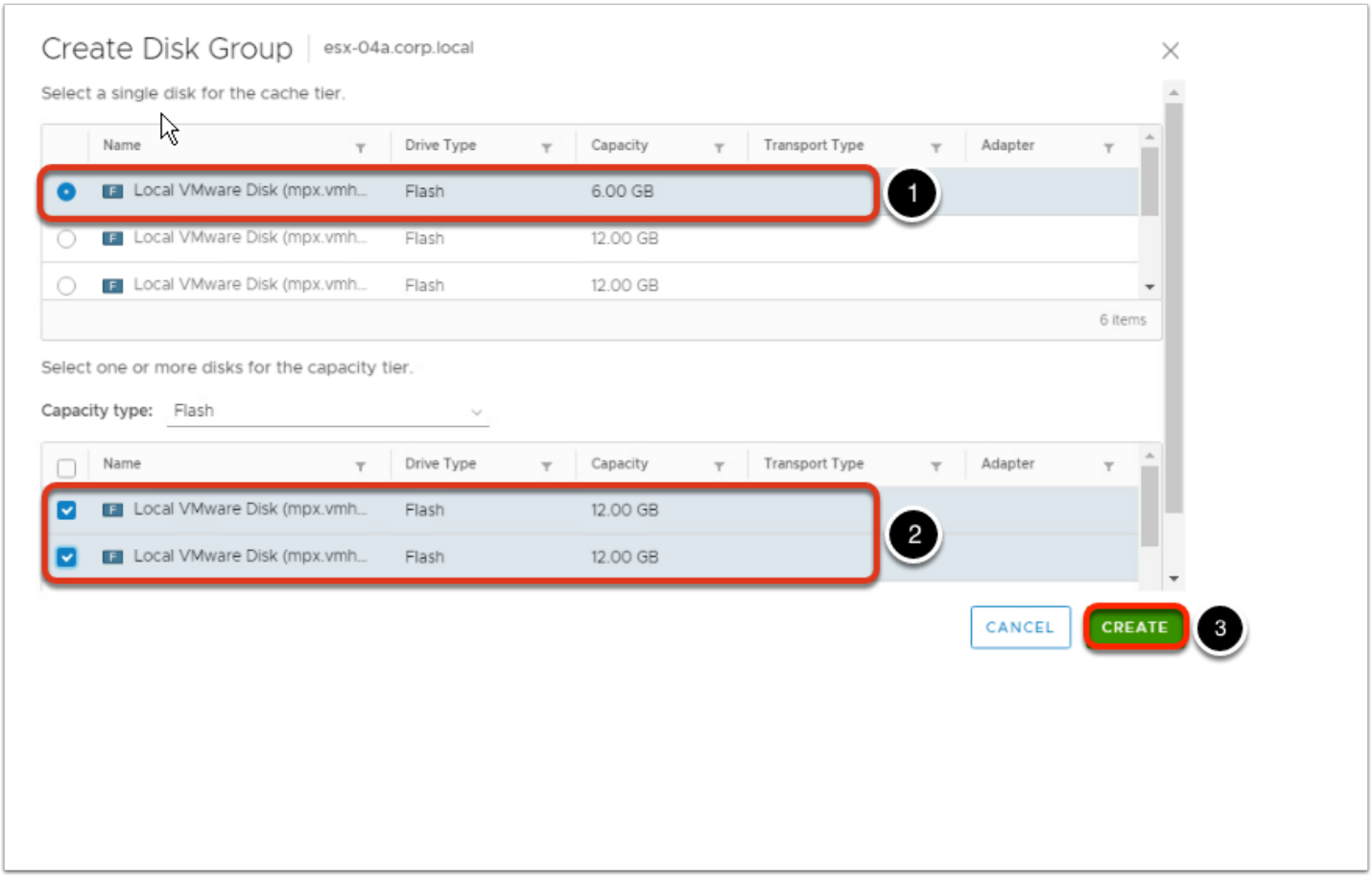
Now that we have configured the Networking, we will grow our vSAN Datastore by using the local storage on the ESXi host.

1. Select vSAN Cluster called **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > Disk Management**
4. Select **esx-04a.corp.local** (do not click host name hyperlink directly, click server icon beside the name)

The ESXi host called **esx-04a.corp.local** is now part of the vSAN Cluster, but it is not contributing any storage to the Disk Groups yet.

5. Click **CREATE DISK GROUP**

Create a Disk Group on a New Host



As before, we select a flash device as cache disk and two flash devices as capacity disks. This is so that all hosts in the cluster maintain a uniform configuration.

1. Select one **6 GB flash** drive for the **Cache** tier
2. Select **2x 12 GB** flash drives for the **Capacity** tier
3. Click **CREATE**

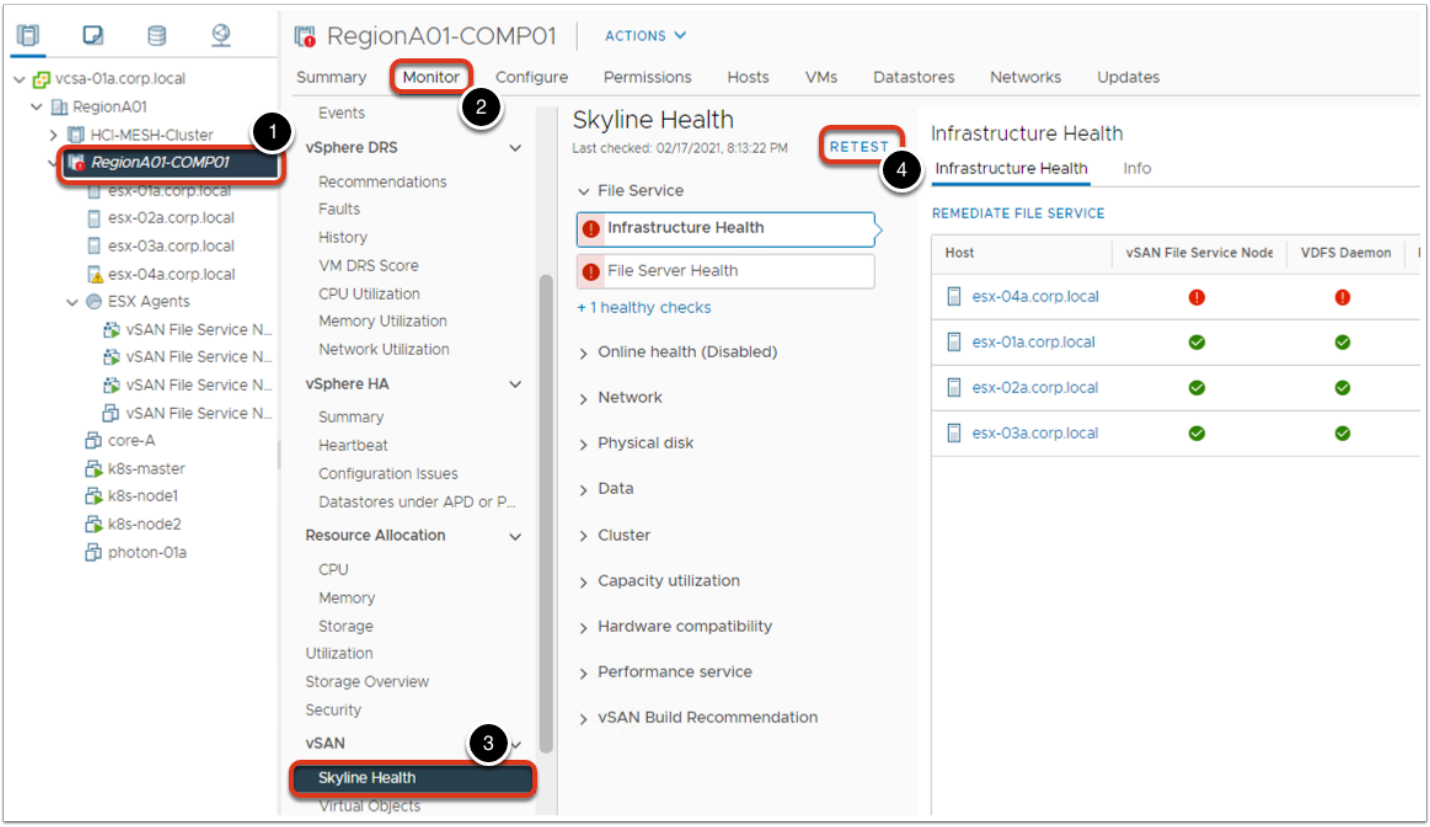
Verify a Disk Group on a New Host

Once the disk group has been created, the disk management view should be revisited.

Verify that the:

1. That all the Disk Groups are in the same Network Partition Group
2. The Disk Format Version is the same on all Disk Groups
3. vSAN Health Status is Healthy

vSAN Health Status



vSAN is not healthy, let's remediate it.

1. Select vSAN Cluster, **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Skyline Health**
4. Click on **RETEST**

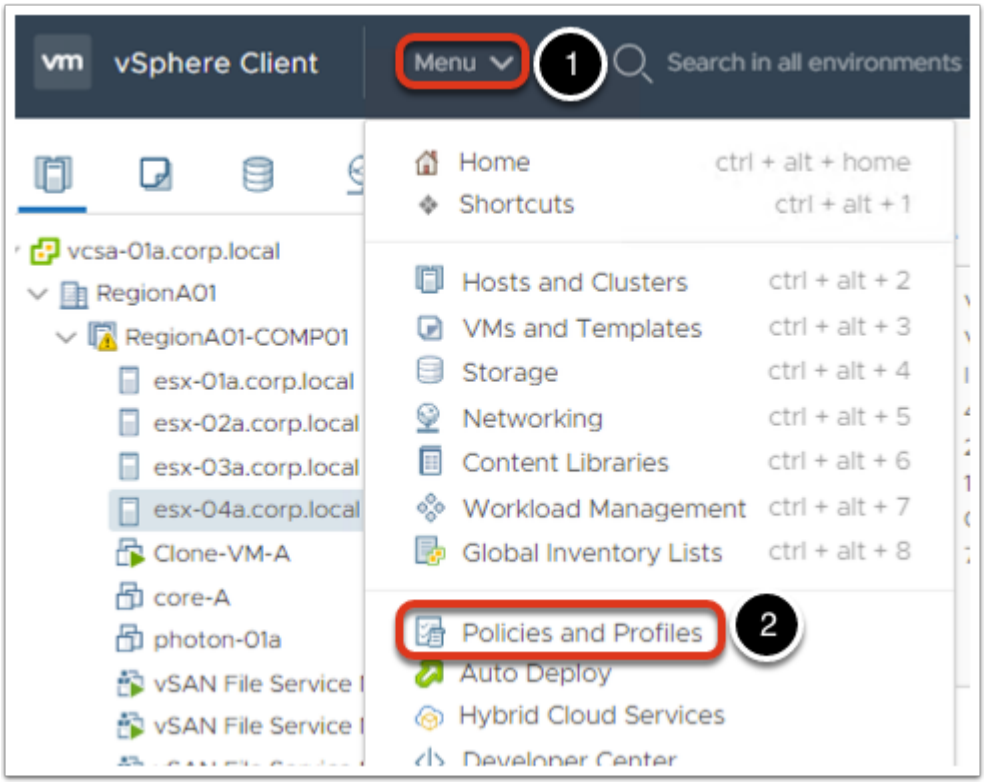
It will take a moment for the retest. The 4th node is now HA within the vSAN cluster. There are other errors but we will work on those errors in other modules. There is a File Server Health error and Infrastructure Health error but in Module 2 when we discuss vSAN File Services, it will be fix there.

Advanced Storage Based Policy Management

Consider these guidelines when you configure RAID 5 or RAID 6 erasure coding in a vSAN cluster.

- RAID 5 or RAID 6 erasure coding is available only on all-flash disk groups.
- On-disk format version 3.0 or later is required to support RAID 5 or RAID 6.
- You must have a valid license to enable RAID 5/6 on a cluster.
- You can achieve additional space savings by enabling deduplication and compression on the vSAN cluster.

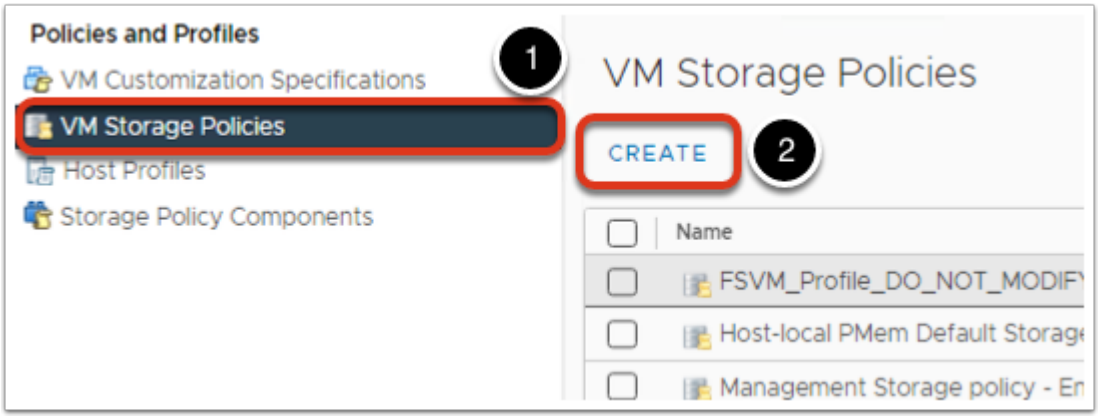
New VM Storage Policy (Raid 5/6 - Erasure coding)



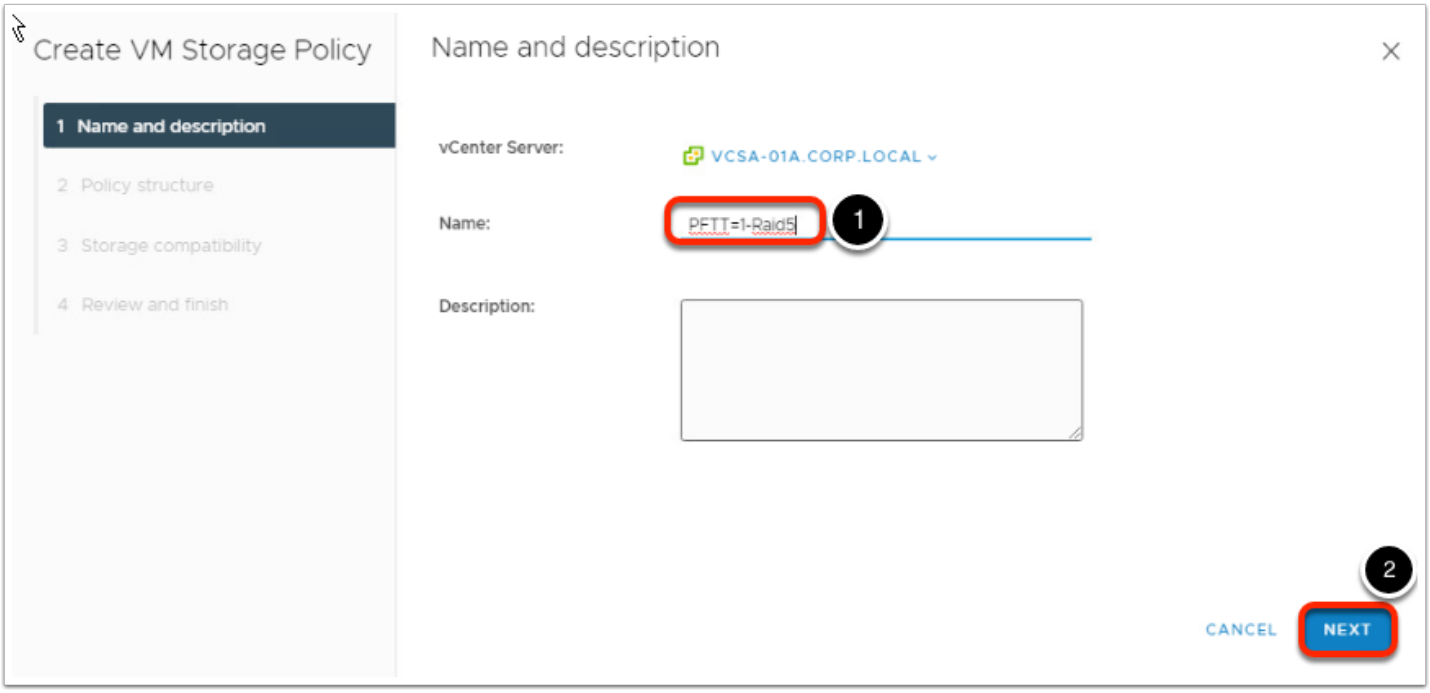
First we need to create a VM Storage Policy that will define the Failure Tolerance method of Raid5/6.

1. From the **Menu** page of the vSphere Client
2. Select **Policies and Profiles**

New VM Storage Policy (Raid 5/6 - Erasure coding)



1. Select **VM Storage Policies**
2. Click on **CREATE**

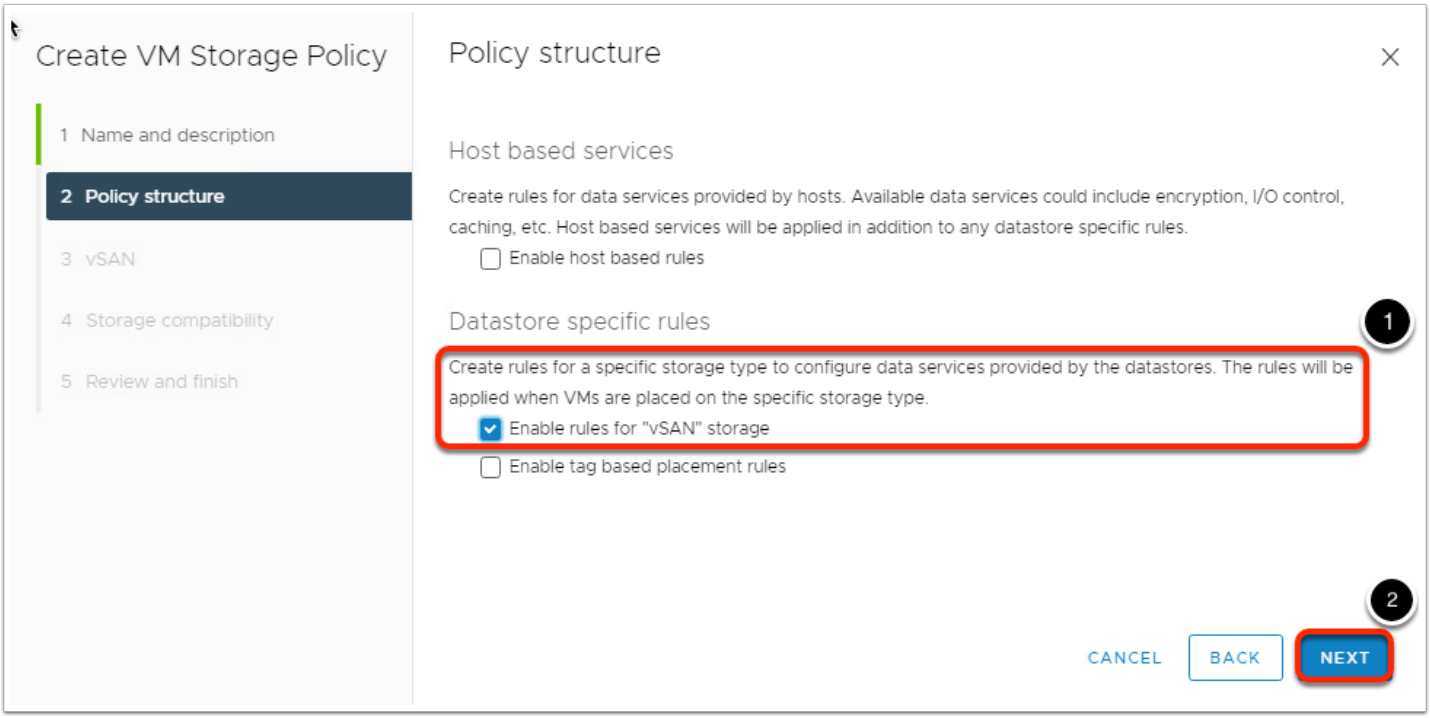


1. Create a new VM Storage Policy using the following name :

```
PFTT=1-Raid5
```

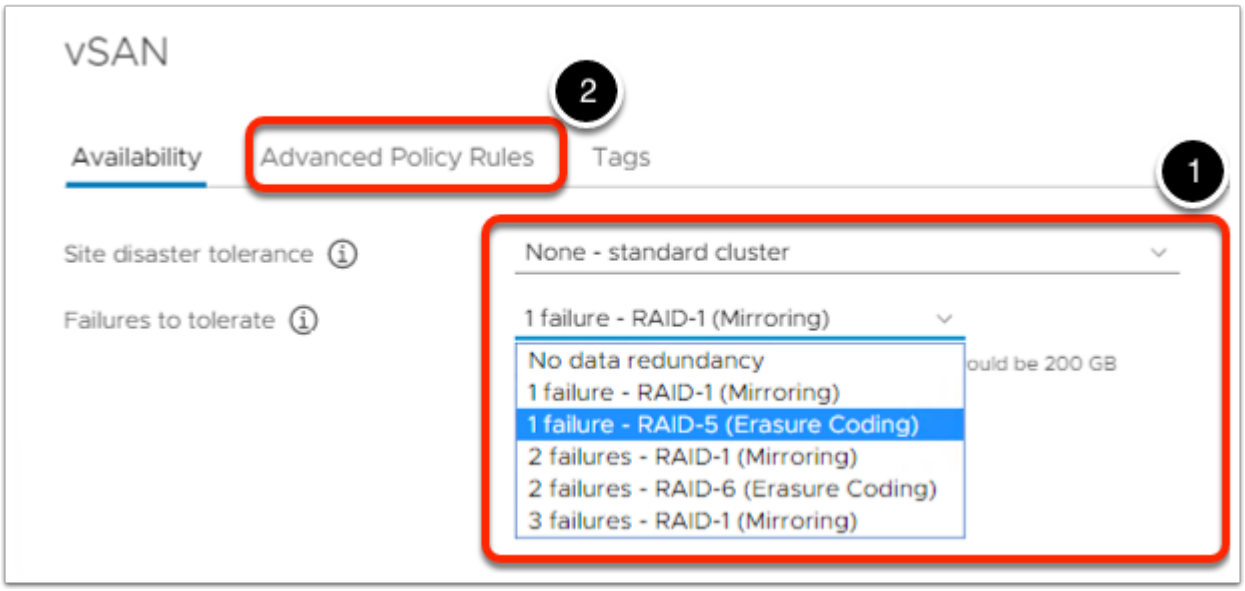
2. Click **NEXT**

New VM Storage Policy (Raid 5/6 - Erasure coding)



1. Select **Enable rules for "vSAN" storage**
2. Click **NEXT**

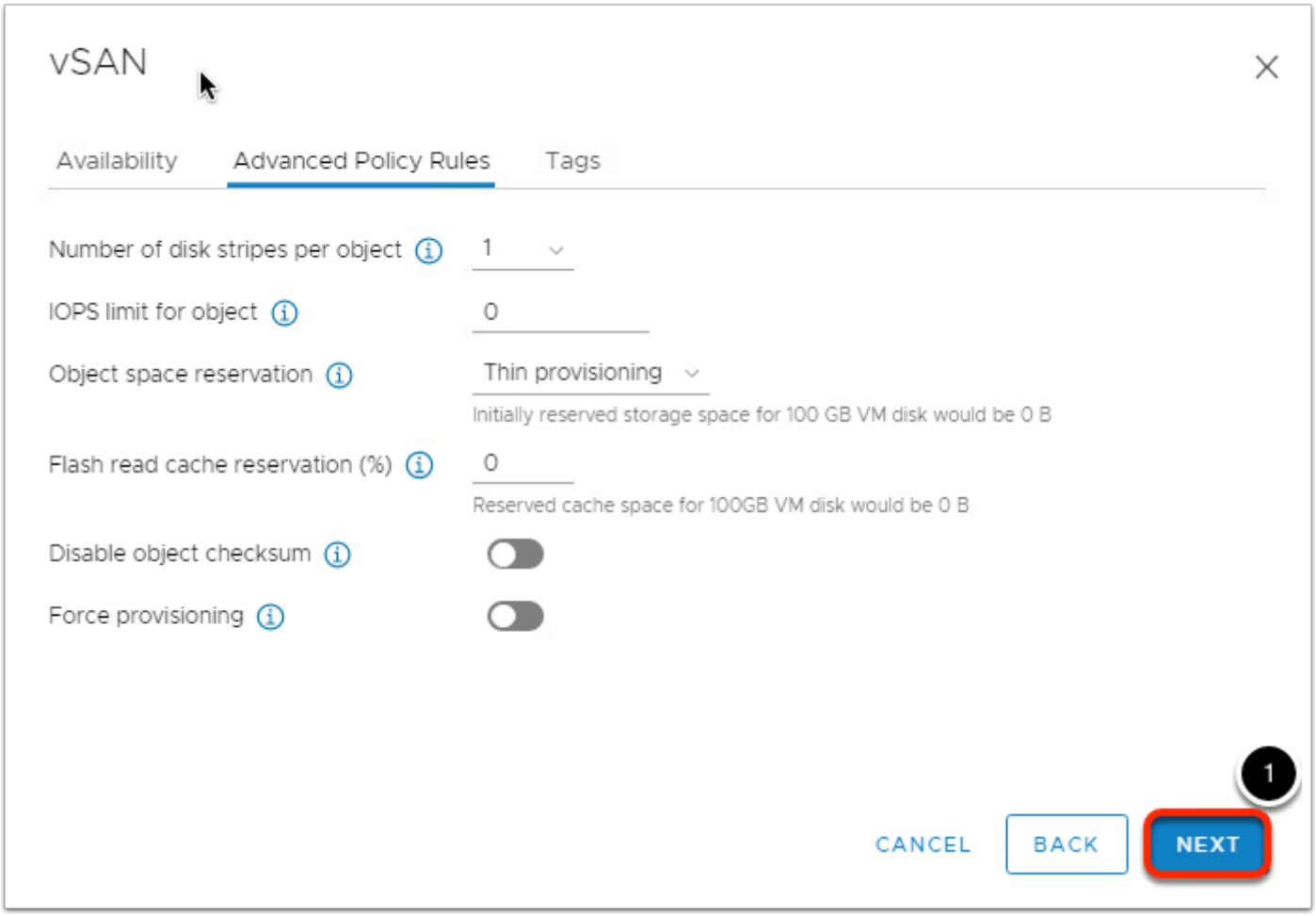
Create New VM Storage Policy (Raid 5/6 - Erasure coding)



1. Select the following options :

```
Site disaster tolerance : None - standard cluster
Failures to Tolerate: 1 failure - Raid-5 (Erasure Coding)
```

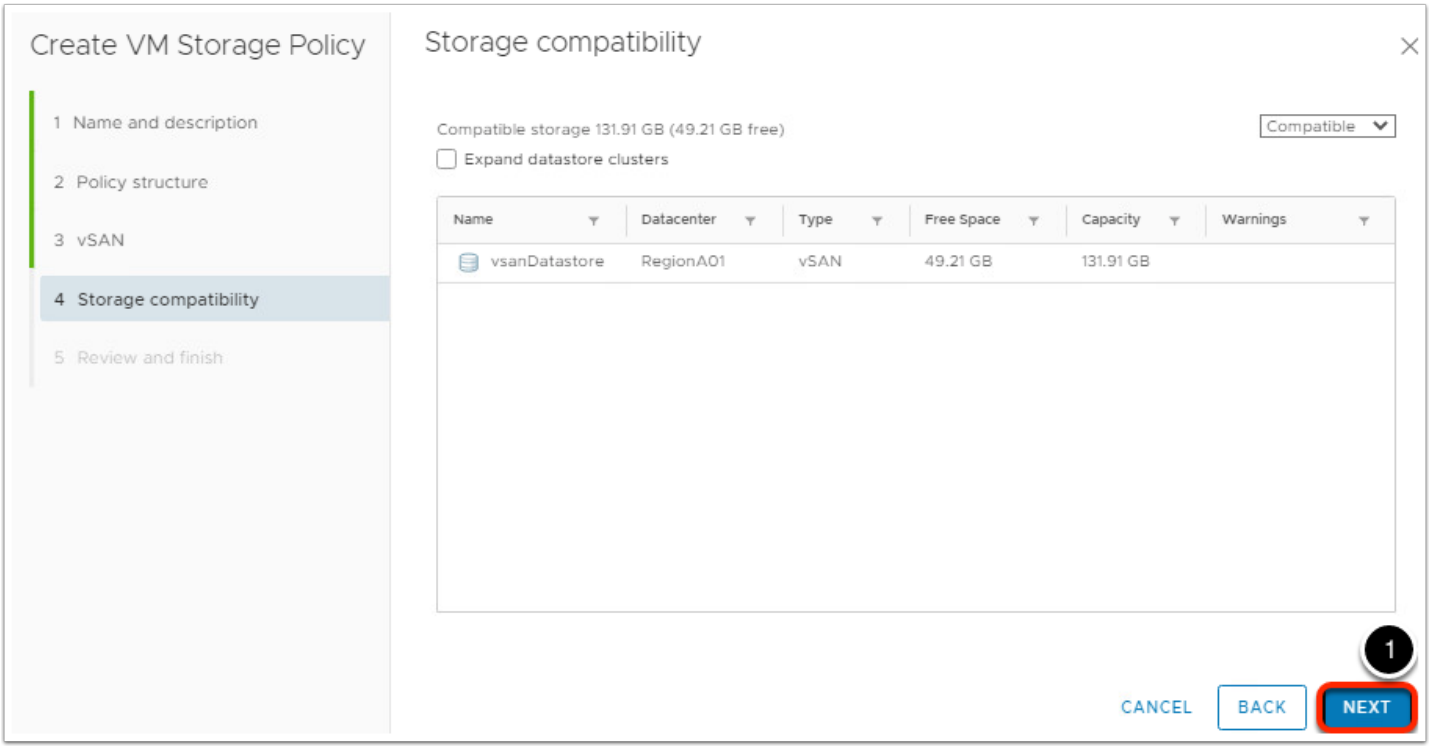
2. Click **Advanced Policy Rules**



Review the options that are available here, but leave at the default settings.

3. Click **NEXT**

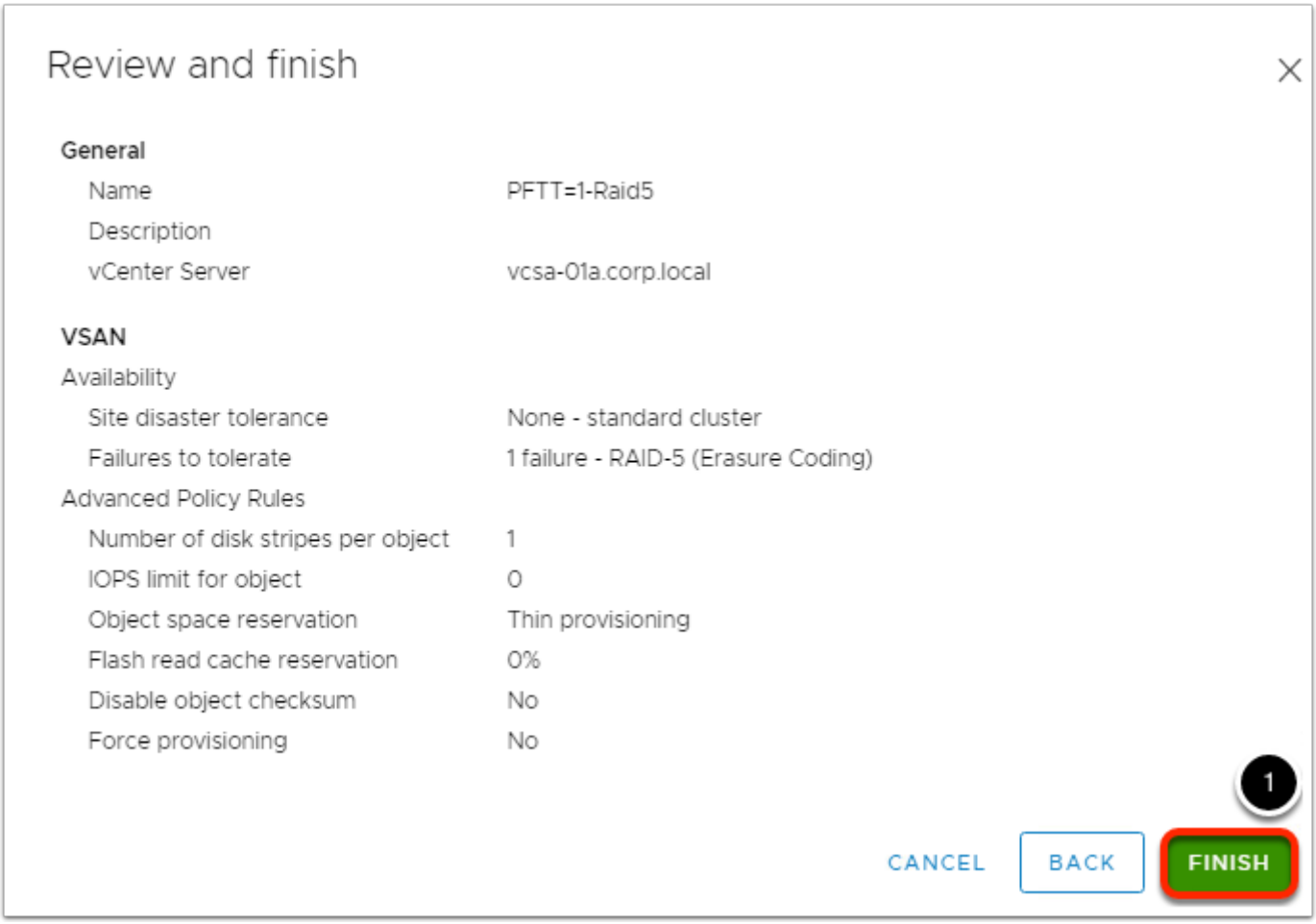
Create New VM Storage Policy (Raid 5/6 - Erasure coding)



Verify that the **vsanDatastore** is compatible against the VM Storage Policy.

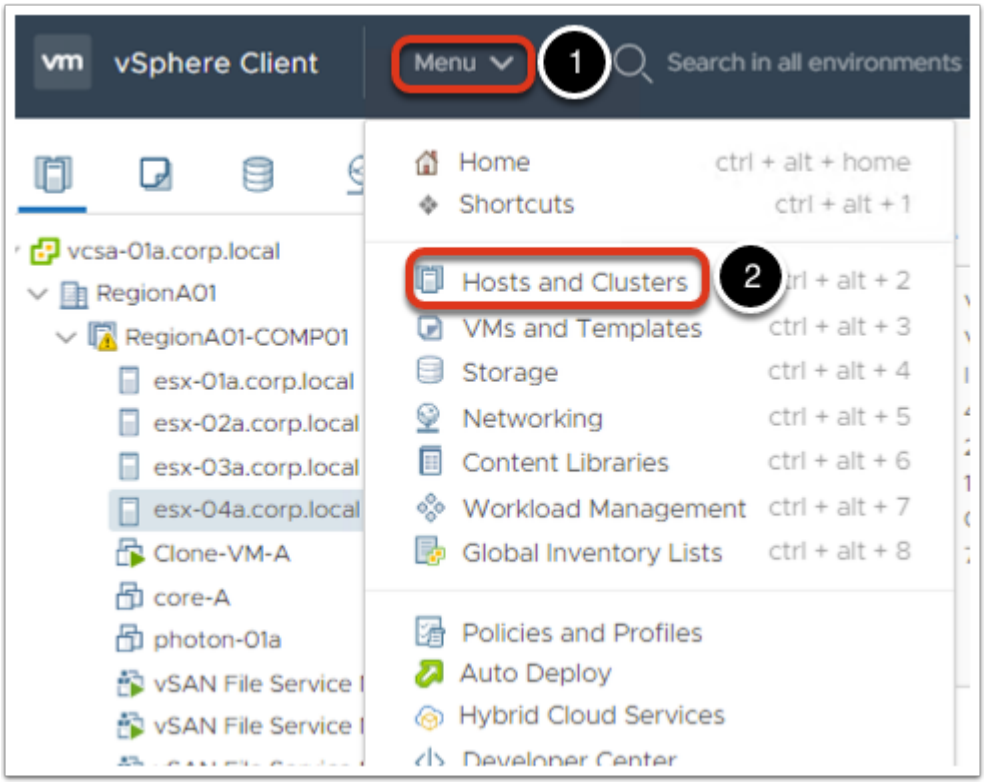
1. Click **NEXT**

New VM Storage Policy (Raid 5/6 - Erasure coding)



1. Review the settings and click **FINISH**

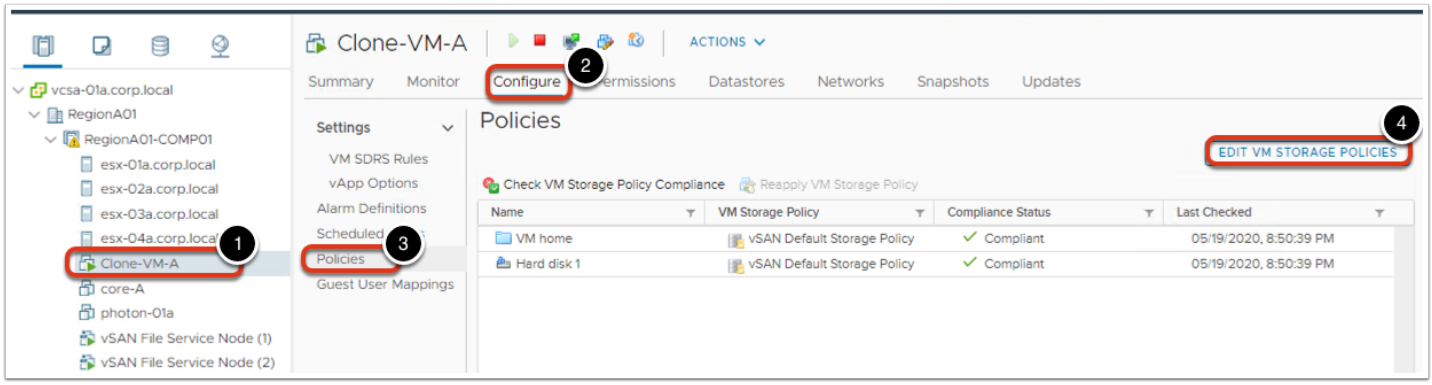
Assign VM Storage Policy to an existing VM



Now that we have created a new VM Storage Policy , lets assign that policy to an existing VM on the vSAN Datastore.

1. Select **Menu** on the vSphere Client
2. Select **Hosts and Clusters**

Assign VM Storage Policy to an existing VM

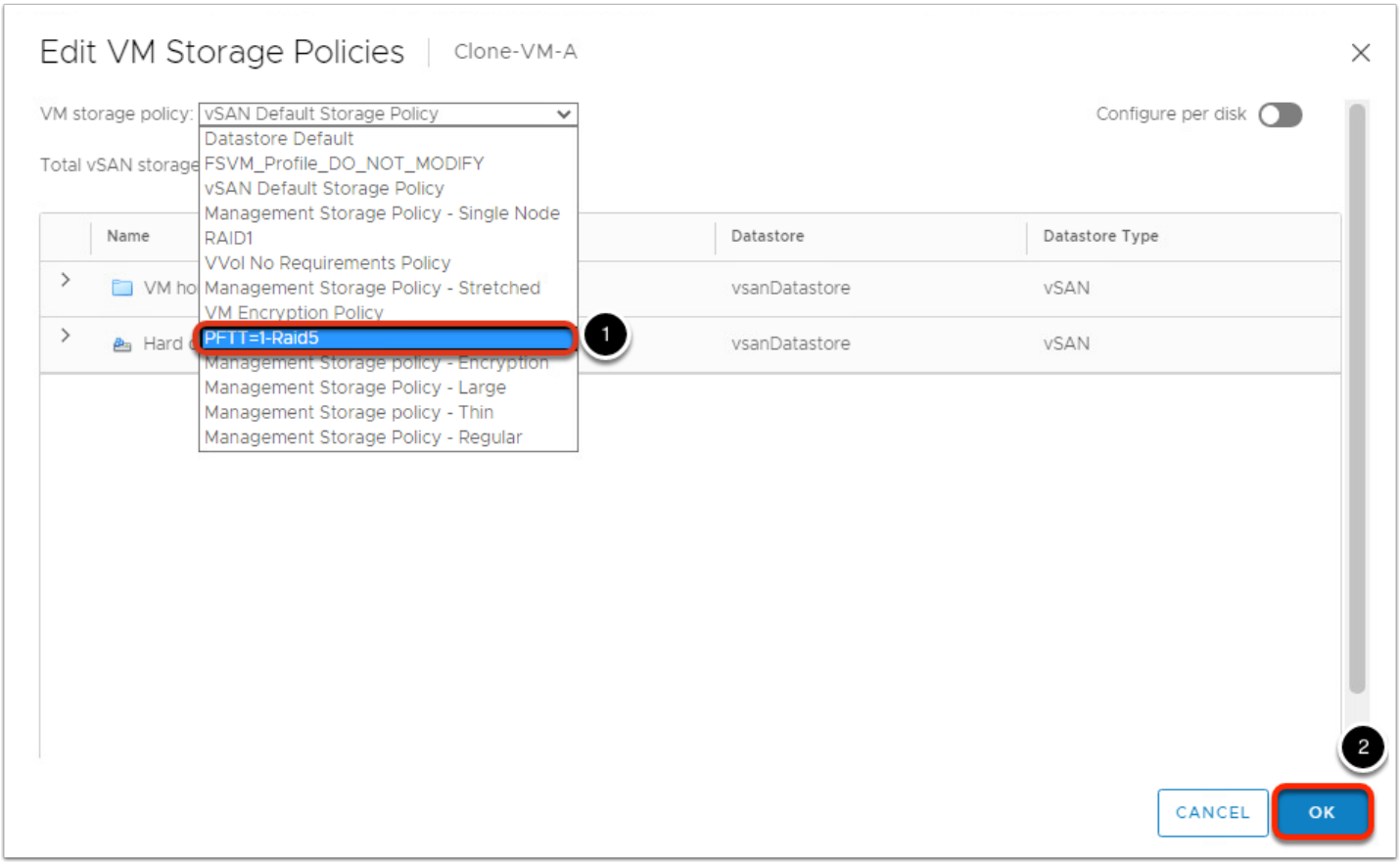


1. Select the VM called **Clone-VM-A**
2. Select **Configure**
3. Select **Policies**

Here we can see that the **vSAN Default Storage Policy** is assigned to this VM.

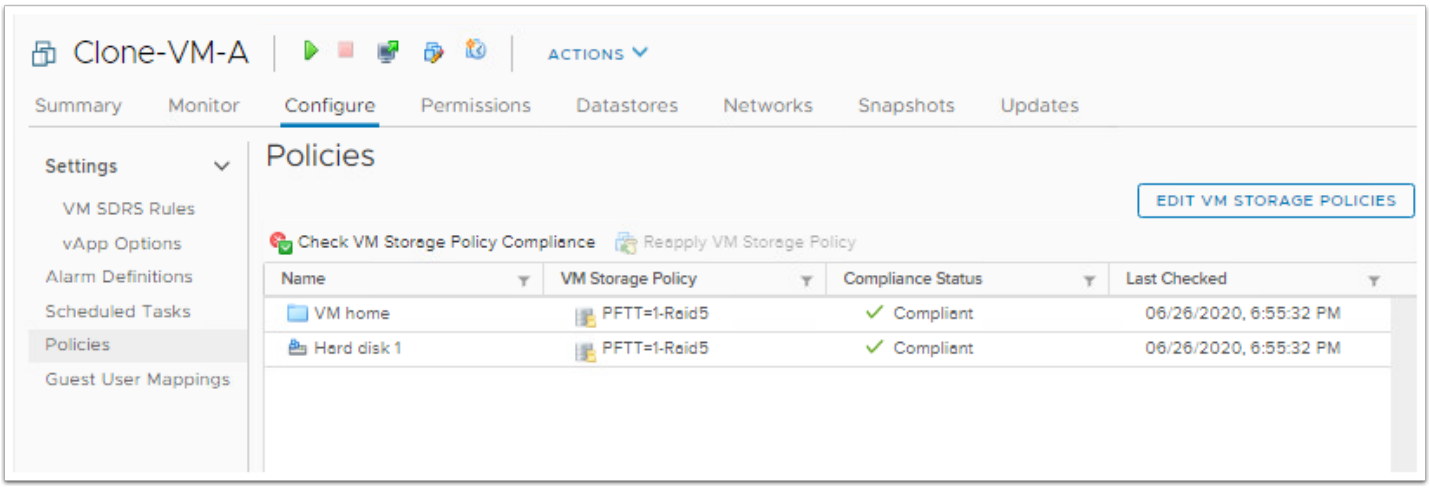
4. Select **EDIT VM STORAGE POLICIES**

Assign VM Storage Policy to an existing VM



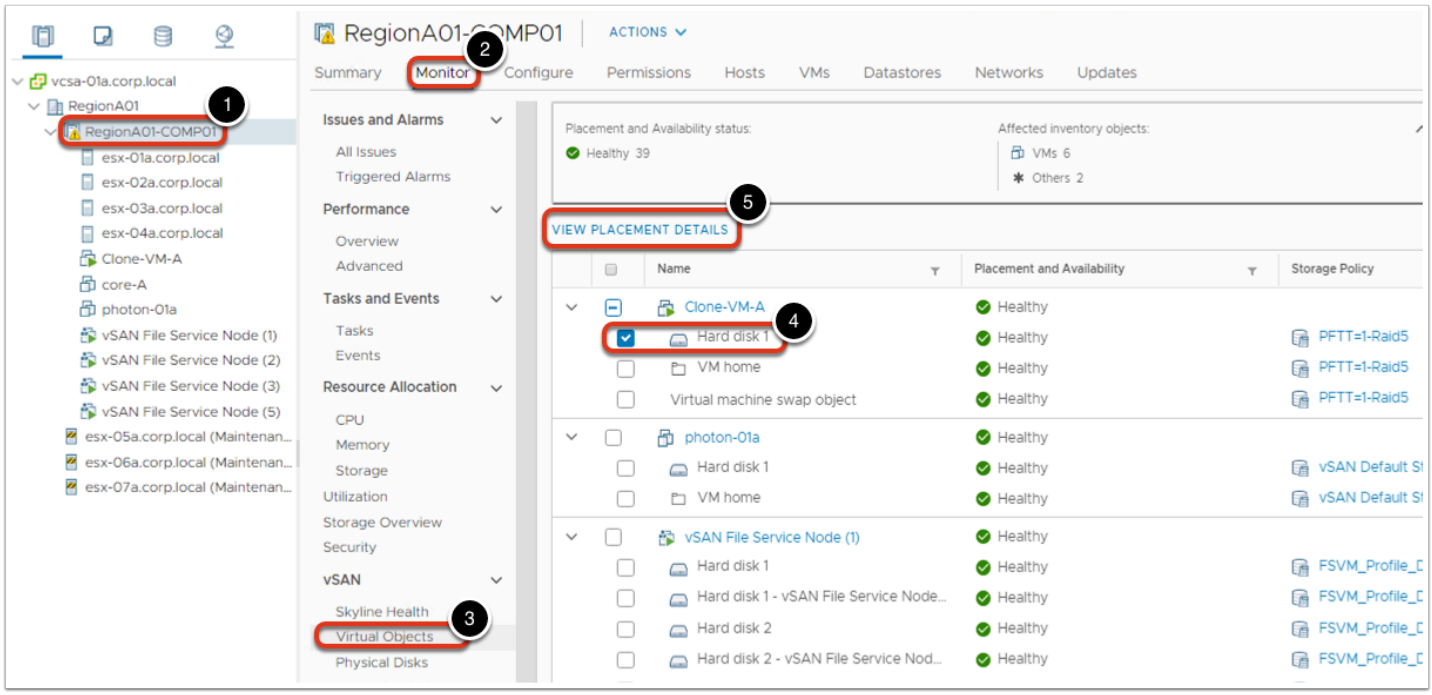
1. Change the VM storage Policy from the dropdown list to **PFTT=1-Raid5**
2. Click **OK**

Assign VM Storage Policy to an existing VM



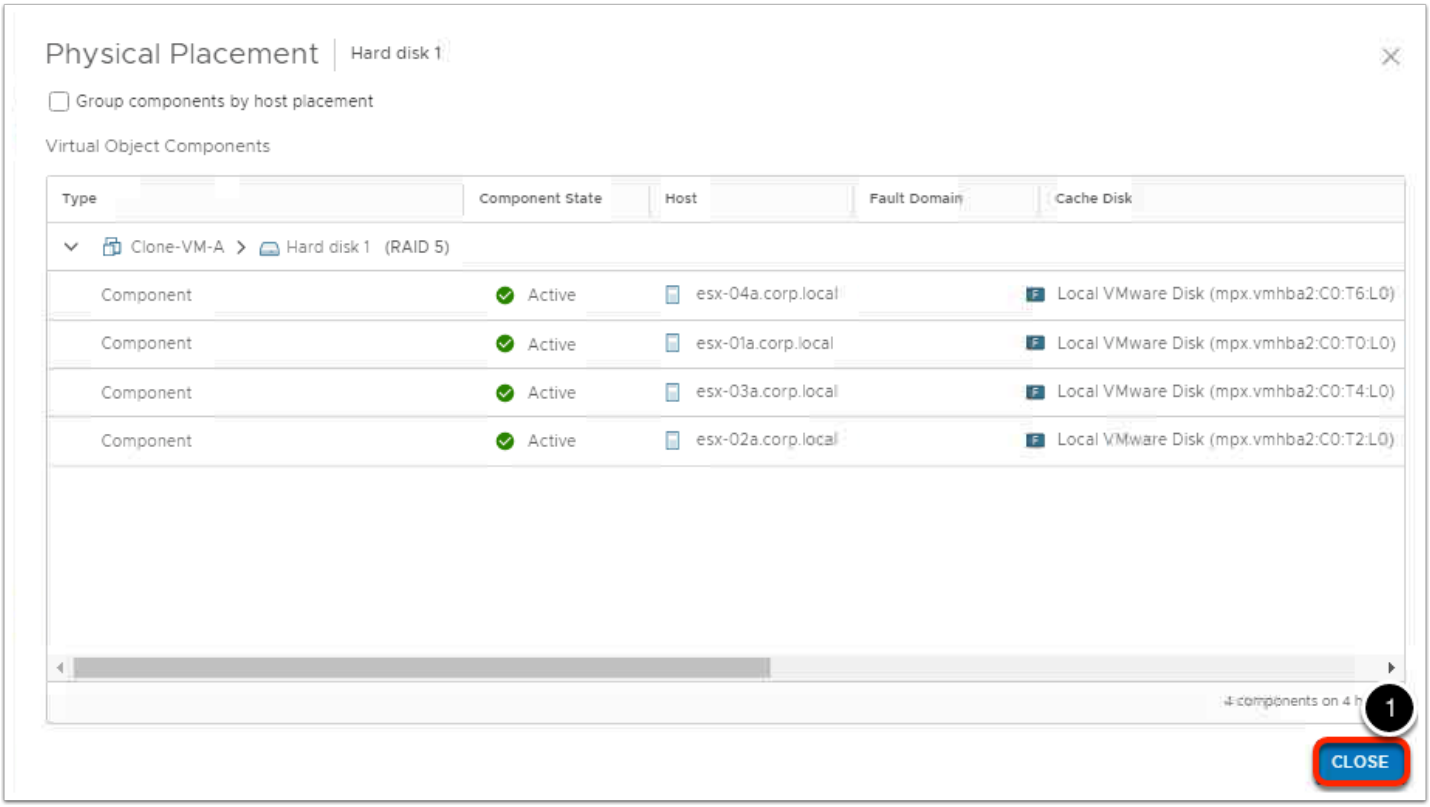
Verify that the VM Storage Policy has been changed and that the VM is compliant against the new storage Policy. You might have to hit the refresh button to see the change.

Assign VM Storage Policy to an existing VM



1. Select the cluster called **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Virtual Objects**
4. For the **Clone-VM-A**, select **Hard disk 1**
5. Select **VIEW PLACEMENT DETAILS**

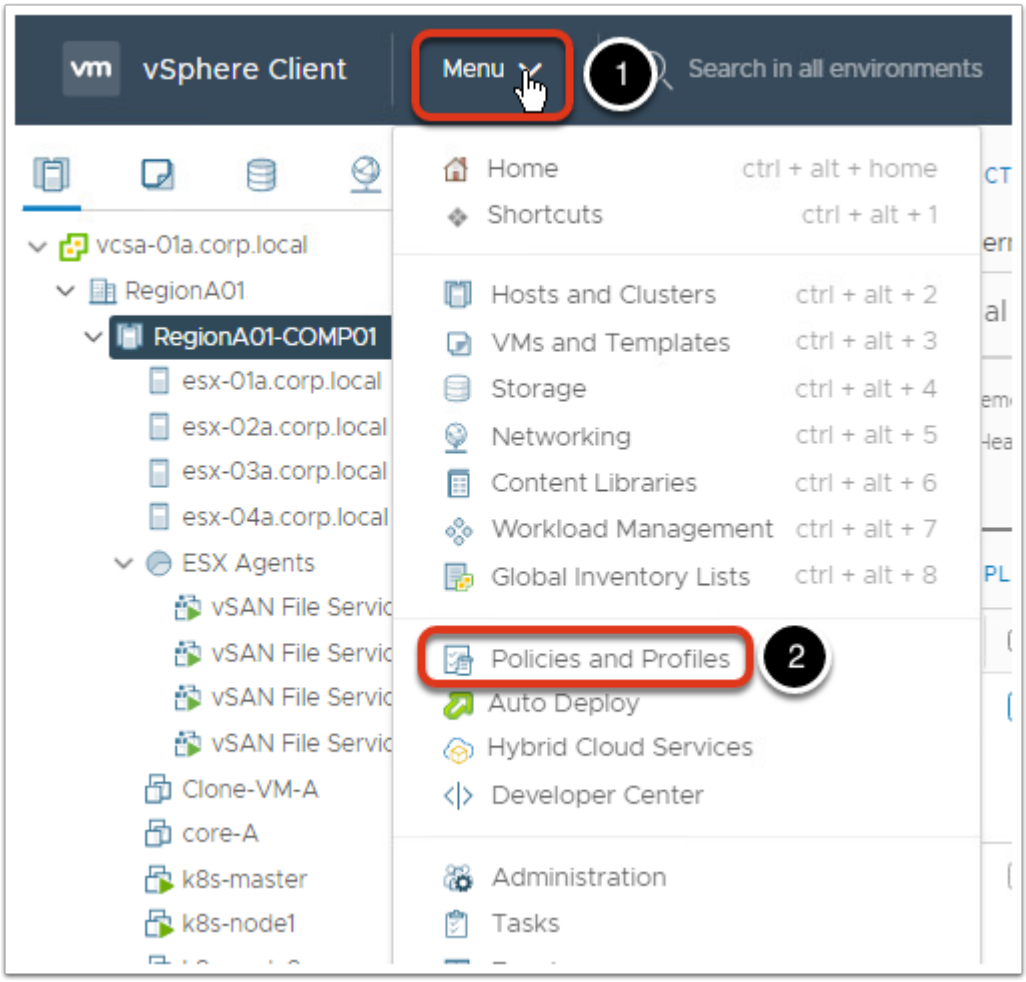
Assign VM Storage Policy to an existing VM



Here we can see the new revised Component layout for the VM with the **Raid-5 Storage Policy (**4 components on 4 hosts).

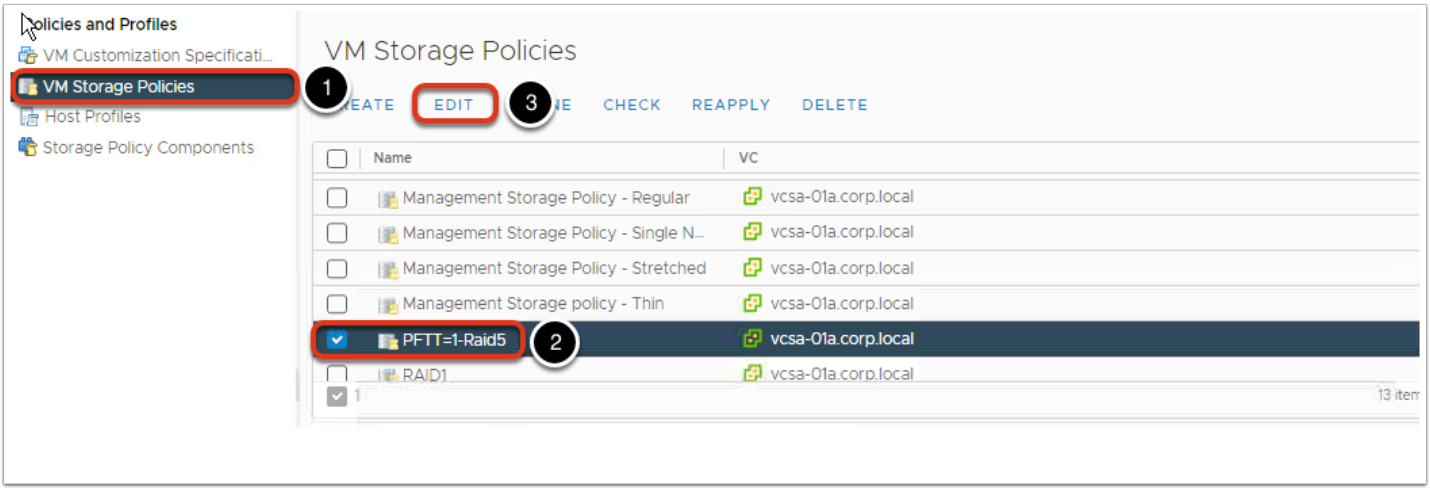
1. Click **CLOSE**

Edit VM Storage Policy



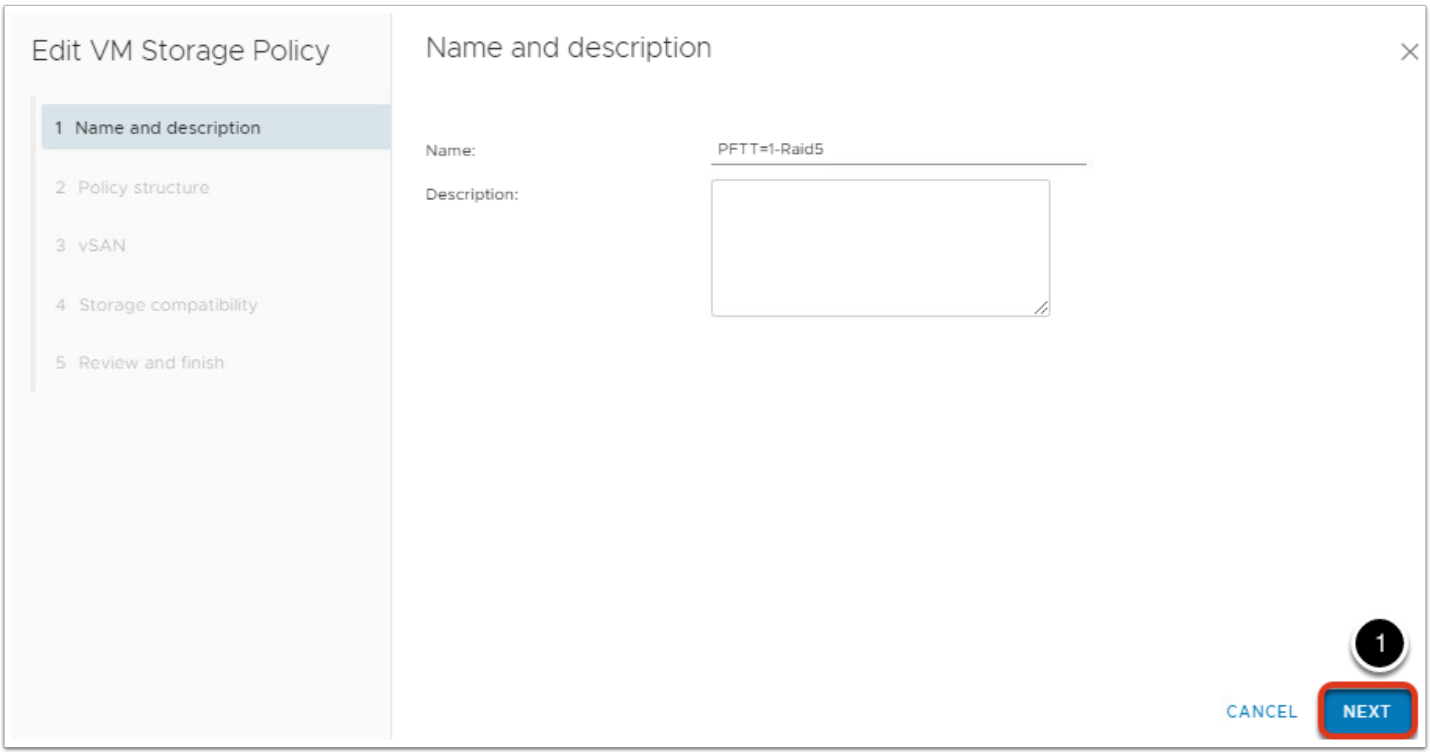
1. Select **Menu** in the vSphere Client
2. Select **Policies and Profiles**

Edit VM Storage Policy



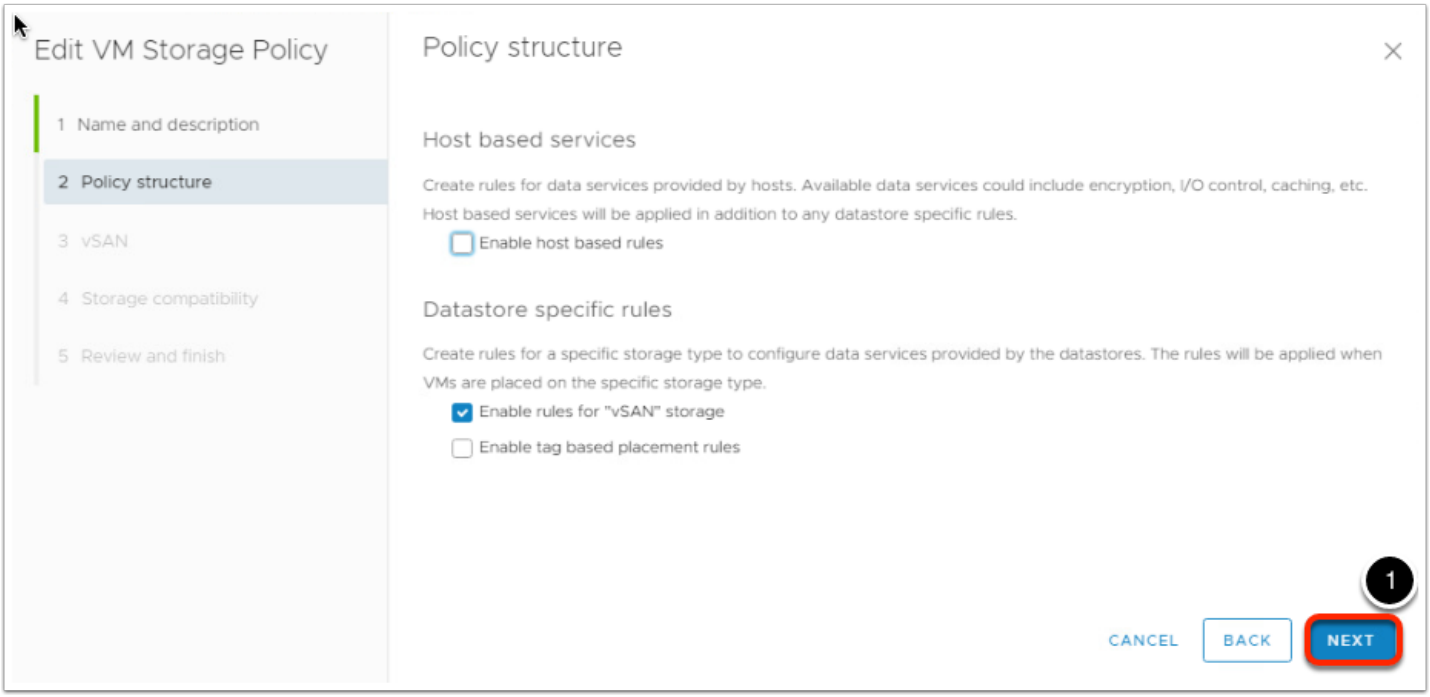
1. Select **VM Storage Policies**
2. Select the VM Storage Policy called **PFTT=1-Raid5**
3. Select **EDIT**

Edit VM Storage Policy



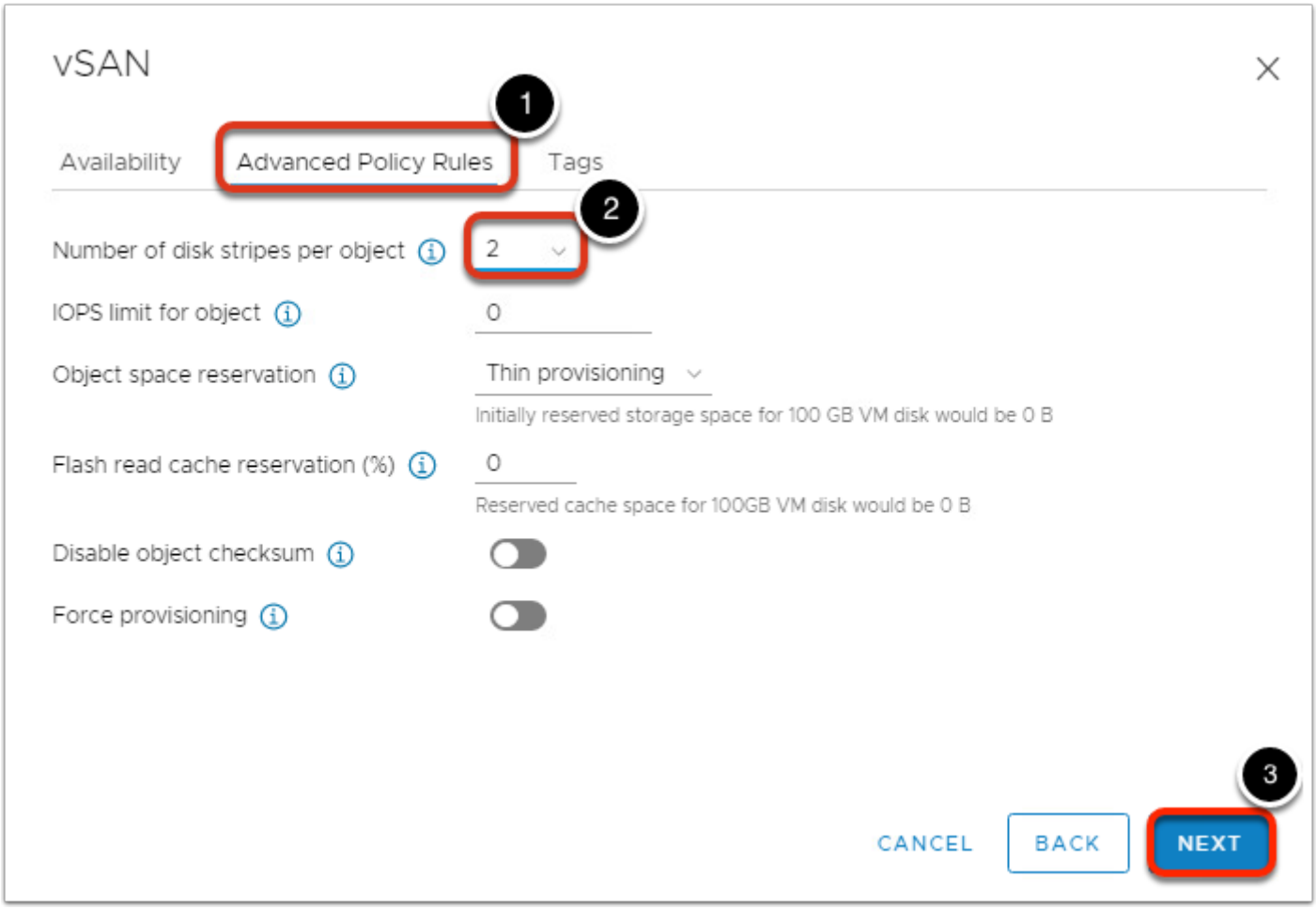
1. On the **Name and description** dialog, Click **NEXT**

Edit VM Storage Policy



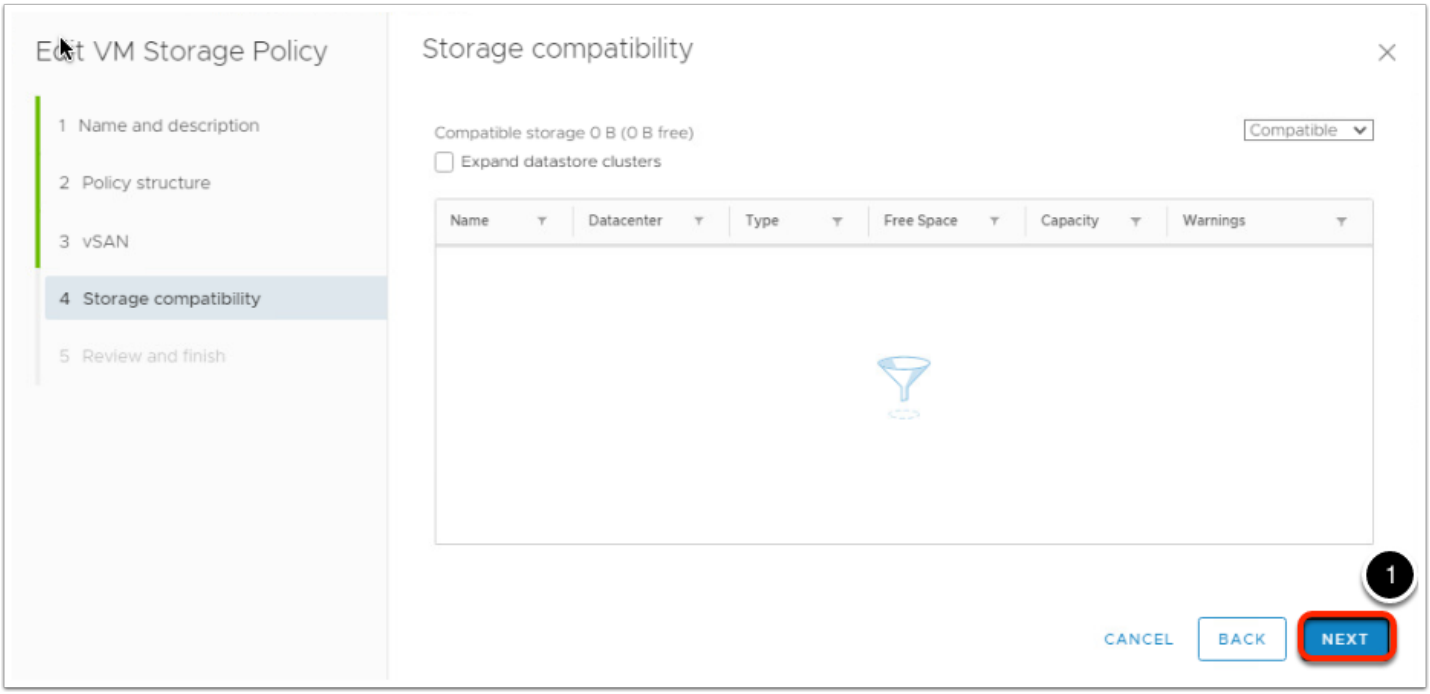
1. On the **Policy structure** dialog, Click **NEXT**

Edit VM Storage Policy



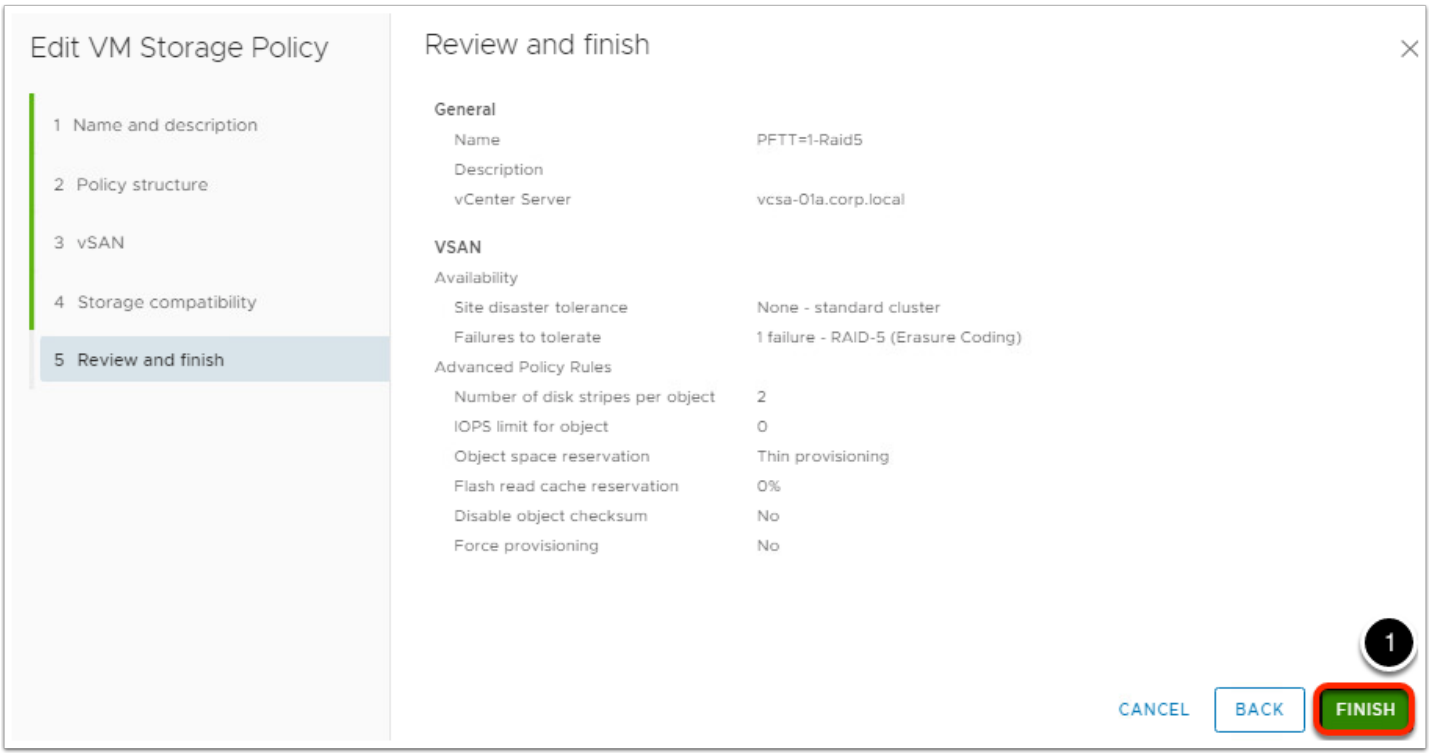
1. On the vSAN dialog, select **Advanced Policy Rules**
2. Modify the **Number of disk stripes per object** to **2**
3. Click **NEXT**

Edit VM Storage Policy



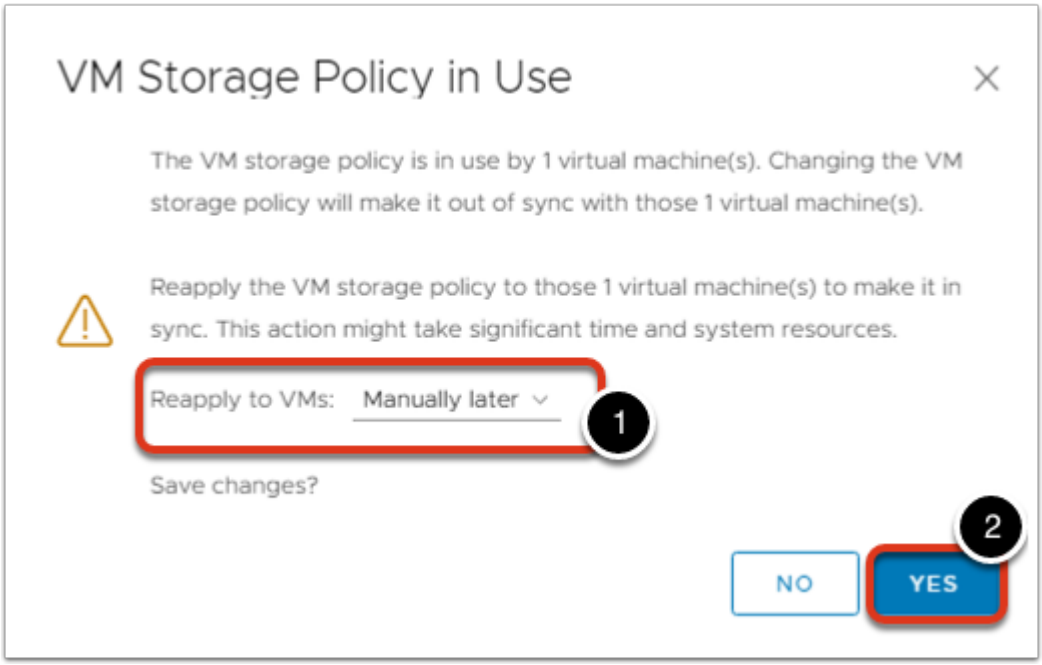
1. On the **Storage compatibility** dialog, click **NEXT**

Edit VM Storage Policy



1. On the **Review and Finish** dialog, click **FINISH**

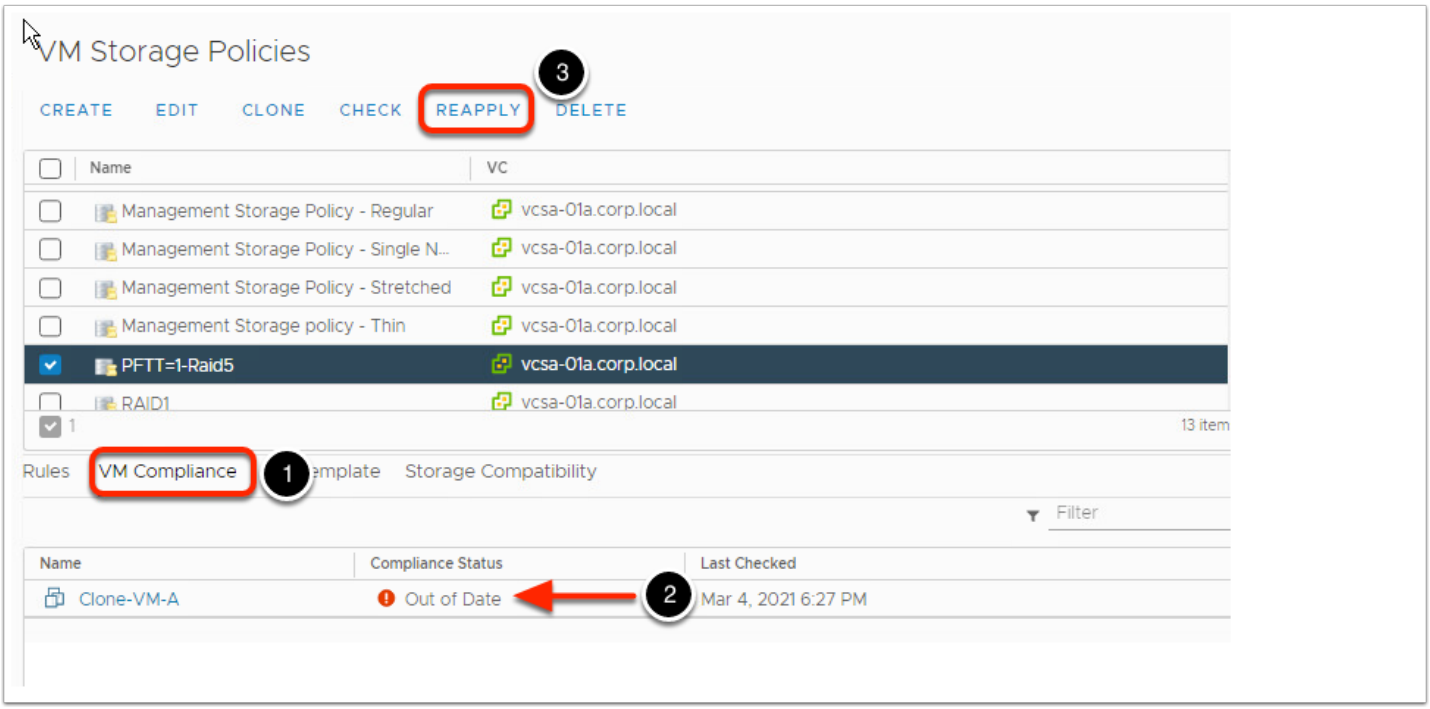
Edit VM Storage Policy



The VM storage policy is in use by 1 virtual machine(s). Changing the VM storage policy will make it out of sync with those 1 virtual machine(s).

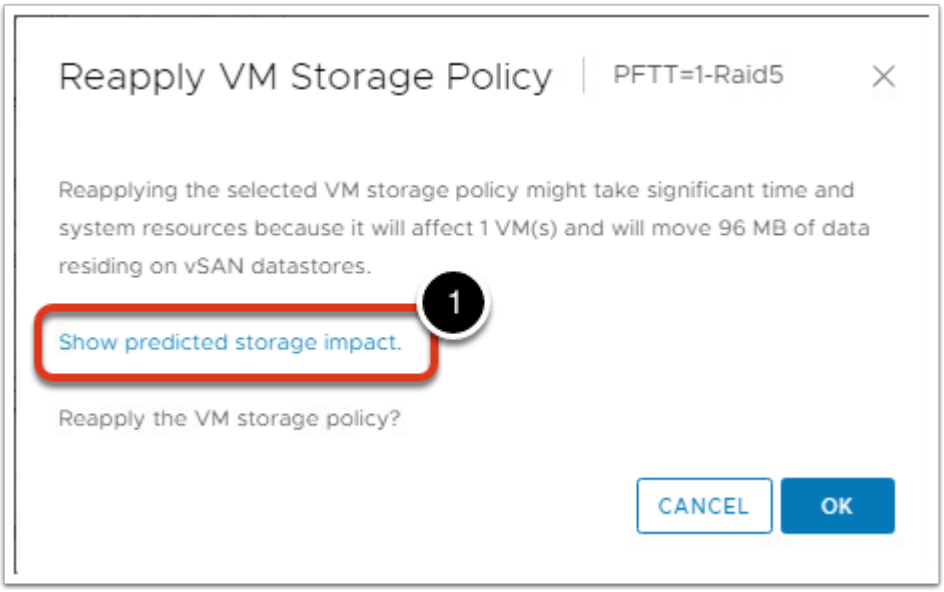
1. Select **Manually later**
2. Select **Yes**

Edit VM Storage Policy



1. Select **VM Compliance**
2. You will see that the Compliance Status of the VM (Clone-VM-A) has now changed to **Out of Date** since we have changed the VM Storage Policy that this VM has been using.
3. Click **REAPPLY**

Edit VM Storage Policy



Reapplying the selected VM storage policy might take significant time and system resources because it will affect 1 VM(s) and will move 96 MB of data residing on vSAN datastores.

1. Click **Show predicted storage impact**

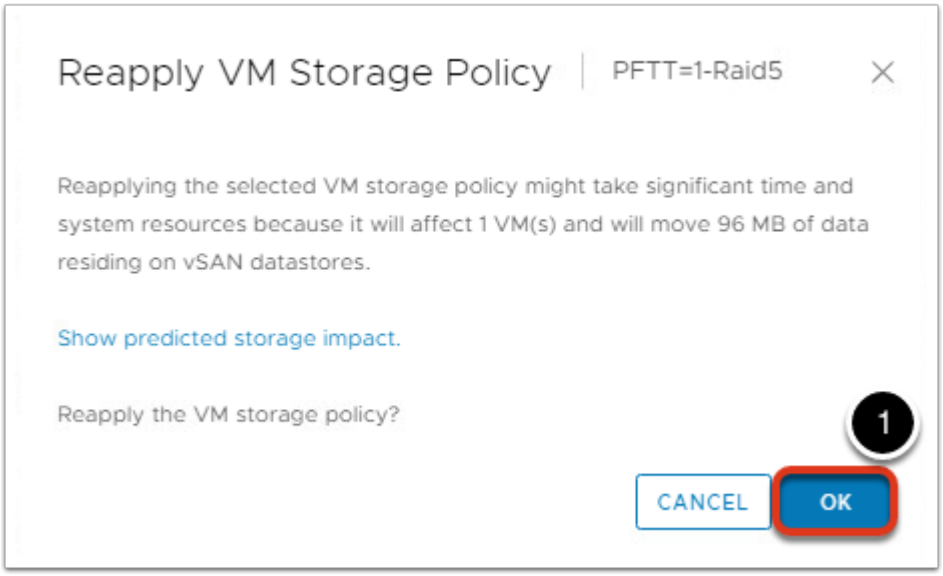
Modify an existing VM Storage Policy

The changes in the VM storage policies will lead to changes in the storage consumption on some datastores. The storage impact can be predicted only for vSAN datastores, but datastores of other types could also be affected.

After you reapply the VM storage policies, the storage consumption of the affected datastores is shown.

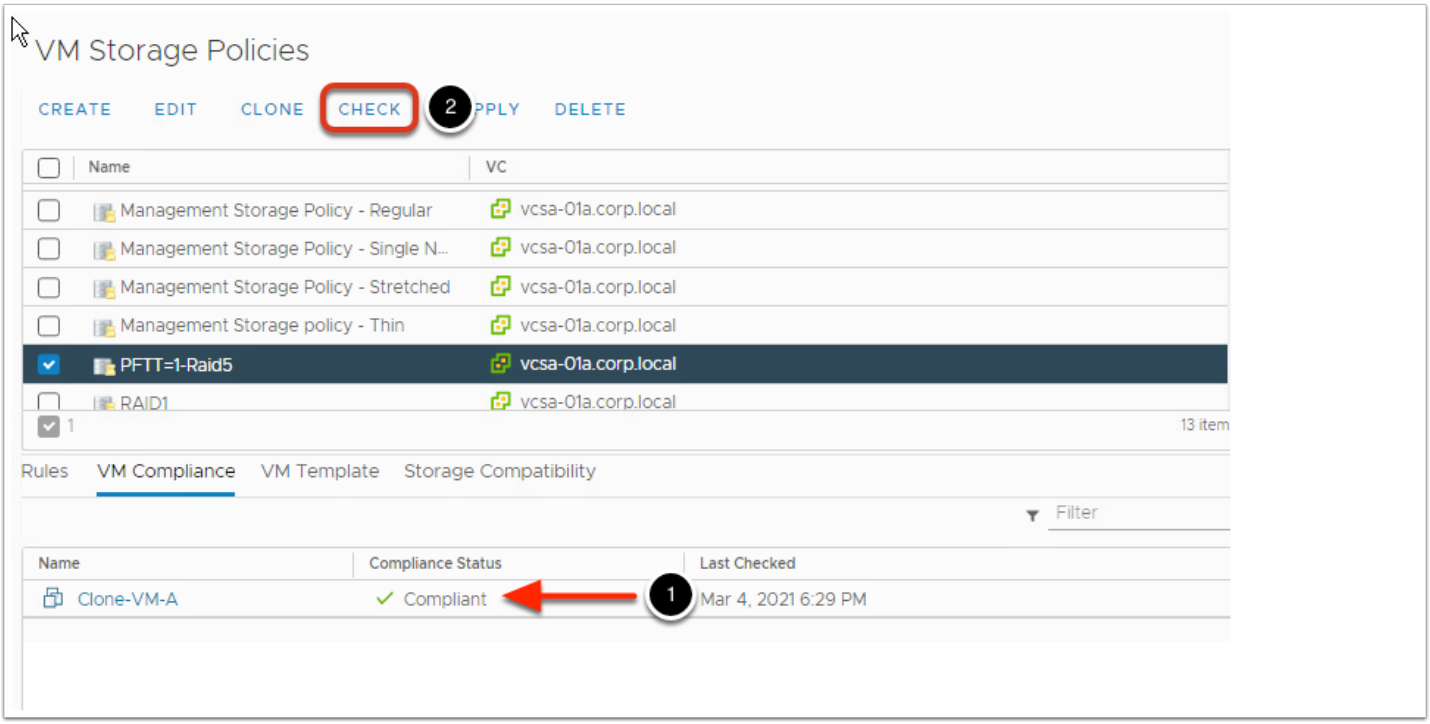
1. Click **CLOSE**

Edit VM Storage Policy



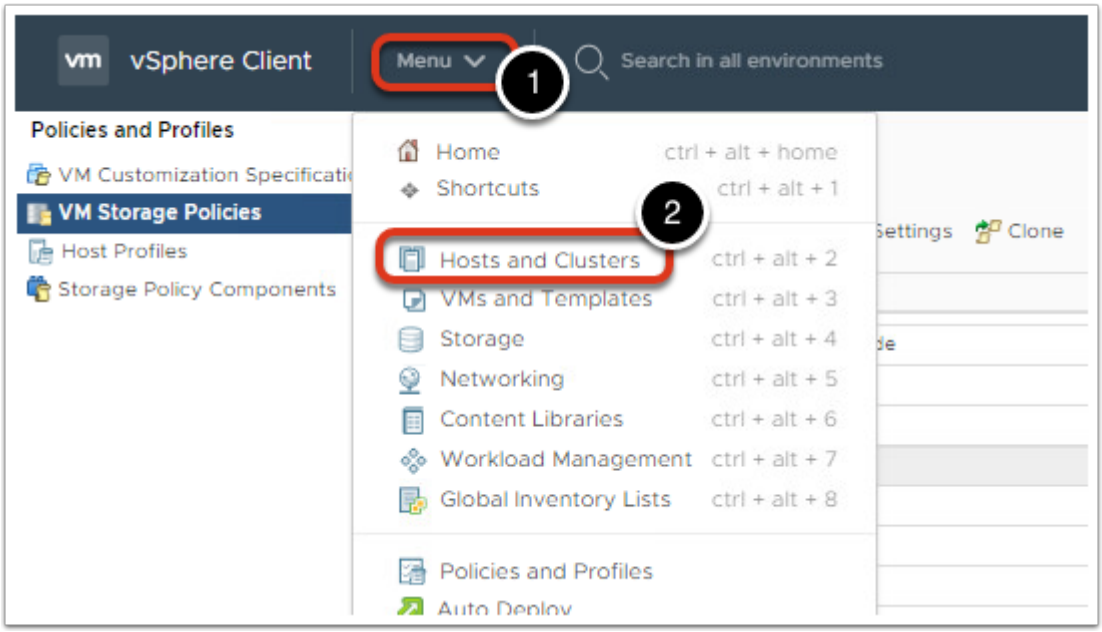
1. Click **OK** to reapply the VM Storage Policy.

Edit VM Storage Policy



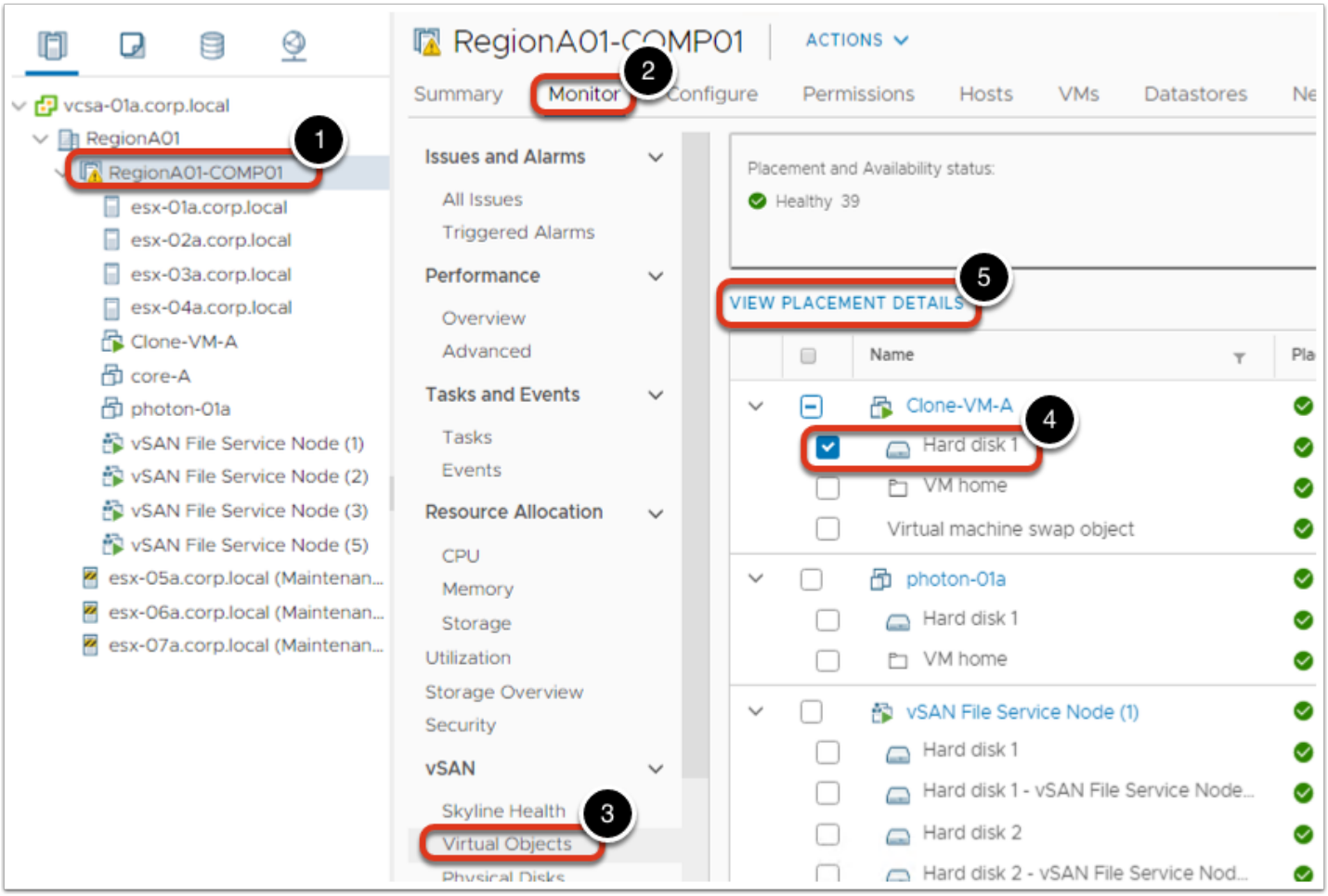
1. Once the VM Storage Policy has been reapplied, verify that the VM is in a Compliant state again with the VM Storage Policy.
2. If the VM does not show **Compliant**, click **CHECK**

Edit VM Storage Policy



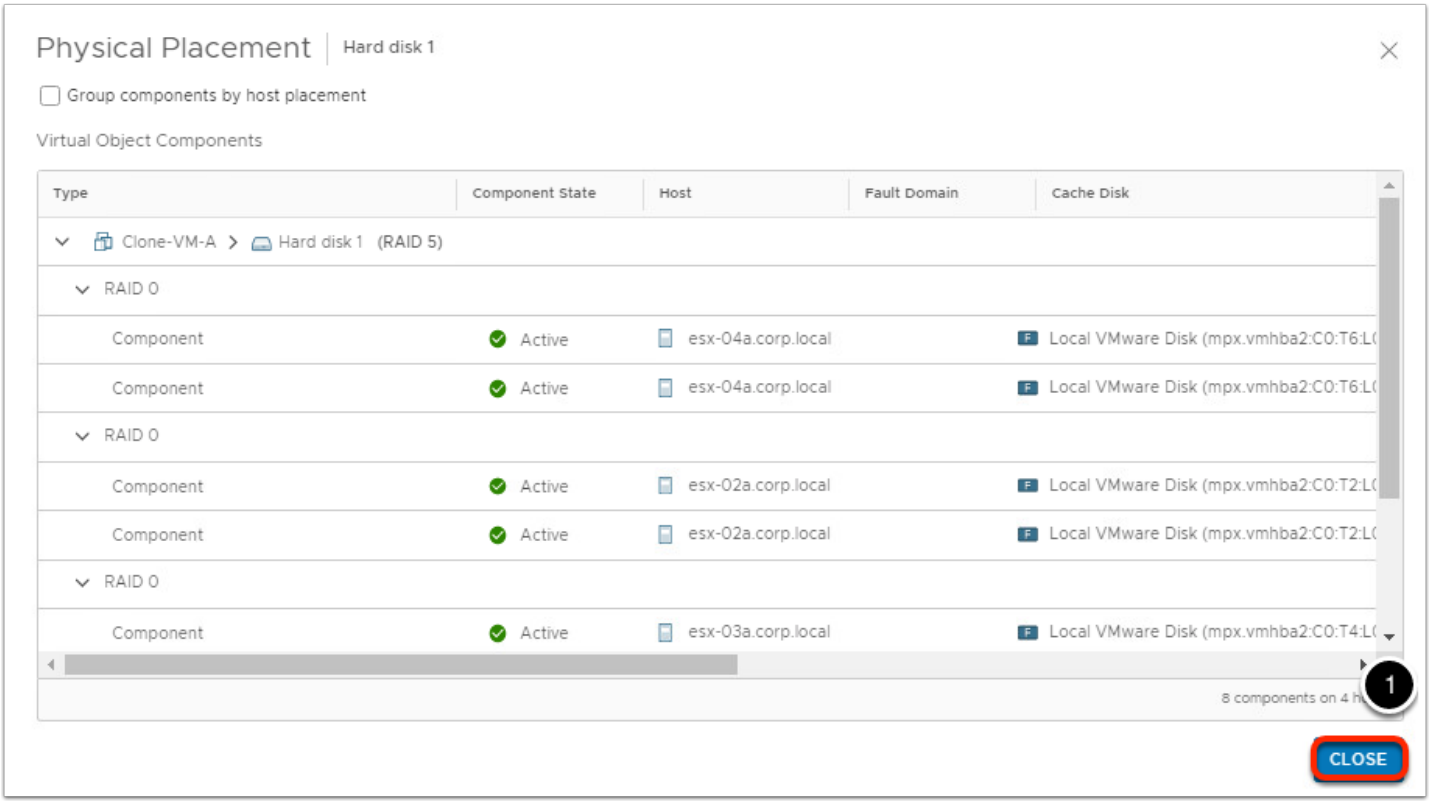
1. From vSphere Client, select **Menu**
2. Select **Hosts and Clusters**

Modify an existing VM Storage Policy



1. Select the cluster called **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Virtual Objects**
4. For the **Clone-VM-A** , select **Hard disk 1**
5. Select **VIEW PLACEMENT DETAILS**

Physical Placement



Here we can see the new revised Components layout for the VM with the **Raid-5 Storage Policy (**8 components on 4 hosts).

We now have components spread across 4 ESXi hosts with Raid 5 but object is RAID 0 due to the number of disk stripes per object is set to 2.

1. Click **CLOSE**

Reserved Capacity

We now have the ability to control the amount of capacity that is reserved for both rebuild operations and transient operations such as temporary capacity need to do policy change on an object.

By default, the Capacity Reserve feature is disabled, meaning all vSAN capacity is available for workloads.You can enable capacity reservations for internal cluster operations and host failure rebuilds. Reservations are soft-thresholds designed to prevent user-driven provisioning activity from interfering with internal operations, such as data rebuilds, rebalancing activity, or policy re-configurations. The capacity required to restore a host failure matches the total capacity of the largest host in the cluster and minimum of 4 hosts are required.

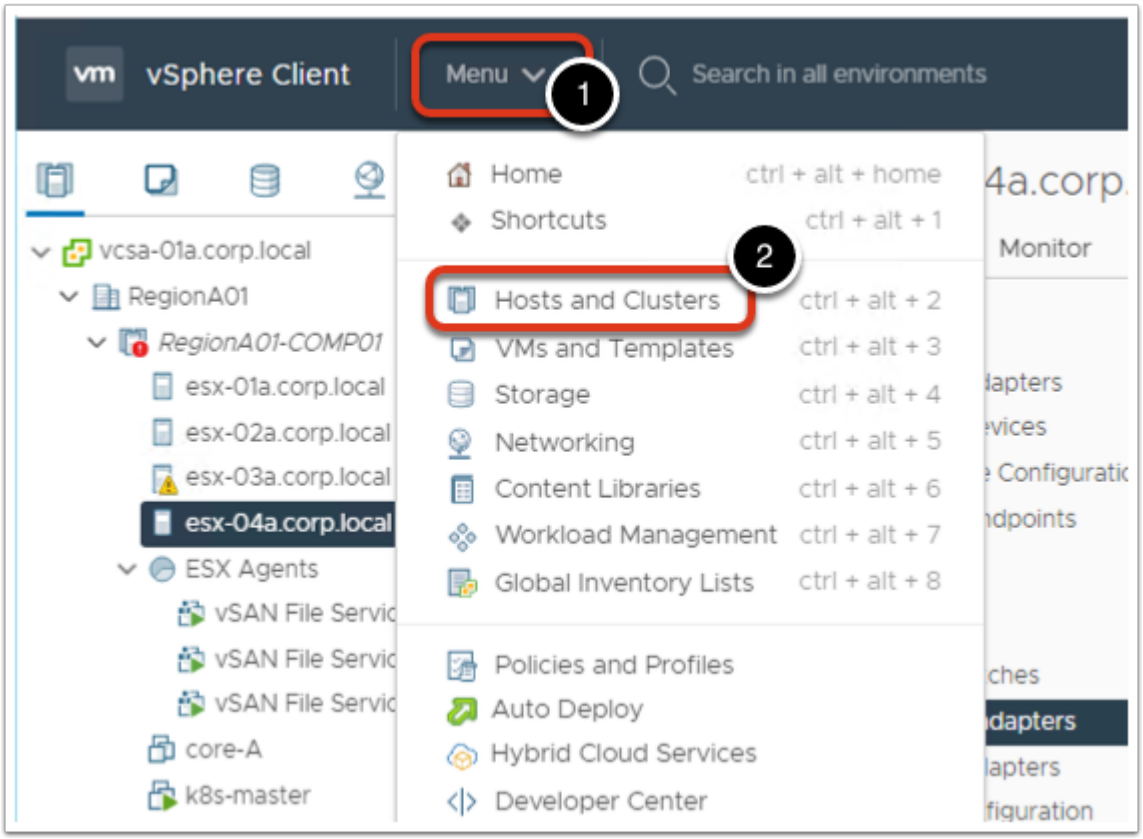
By enabling reserve capacity in advanced, vSAN prevents you from using the space to create workloads and intends to save the capacity available in a cluster.

If there is enough free space in the vSAN cluster, you can enable the operations reserve and/or the host rebuild reserve.

- Operations Reserve - Reserved space in the cluster for vSAN internal operations.
- Host Rebuild Reserve - Reserved space for vSAN to be able to repair in case of a single host failure.

The reserved capacity is not supported on a stretched cluster, cluster with fault domains and nested fault domains, ROBO cluster, or the number of hosts in the cluster is less than four.

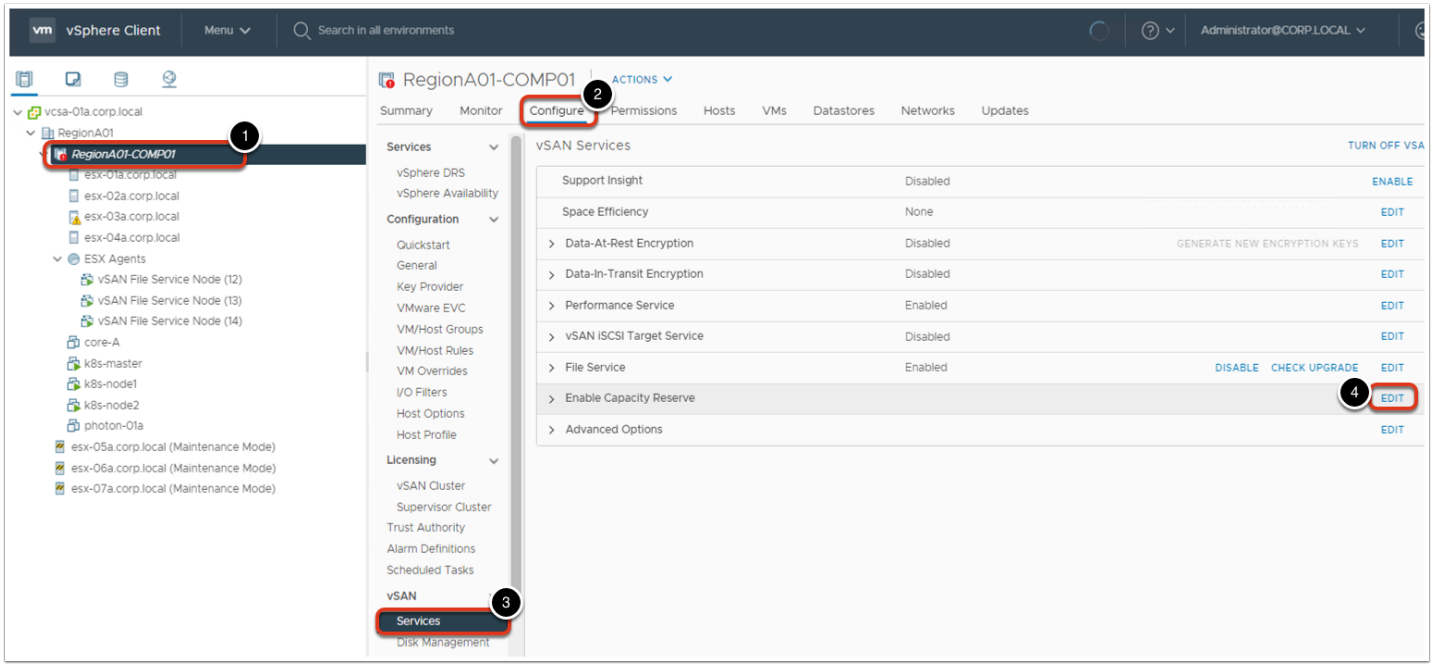
Configure Reserved Capacity



Now that we have 4 hosts in the cluster, we can start configuring the Reserve Capacity.

1. Select **Menu**
2. Select **Hosts and Clusters**

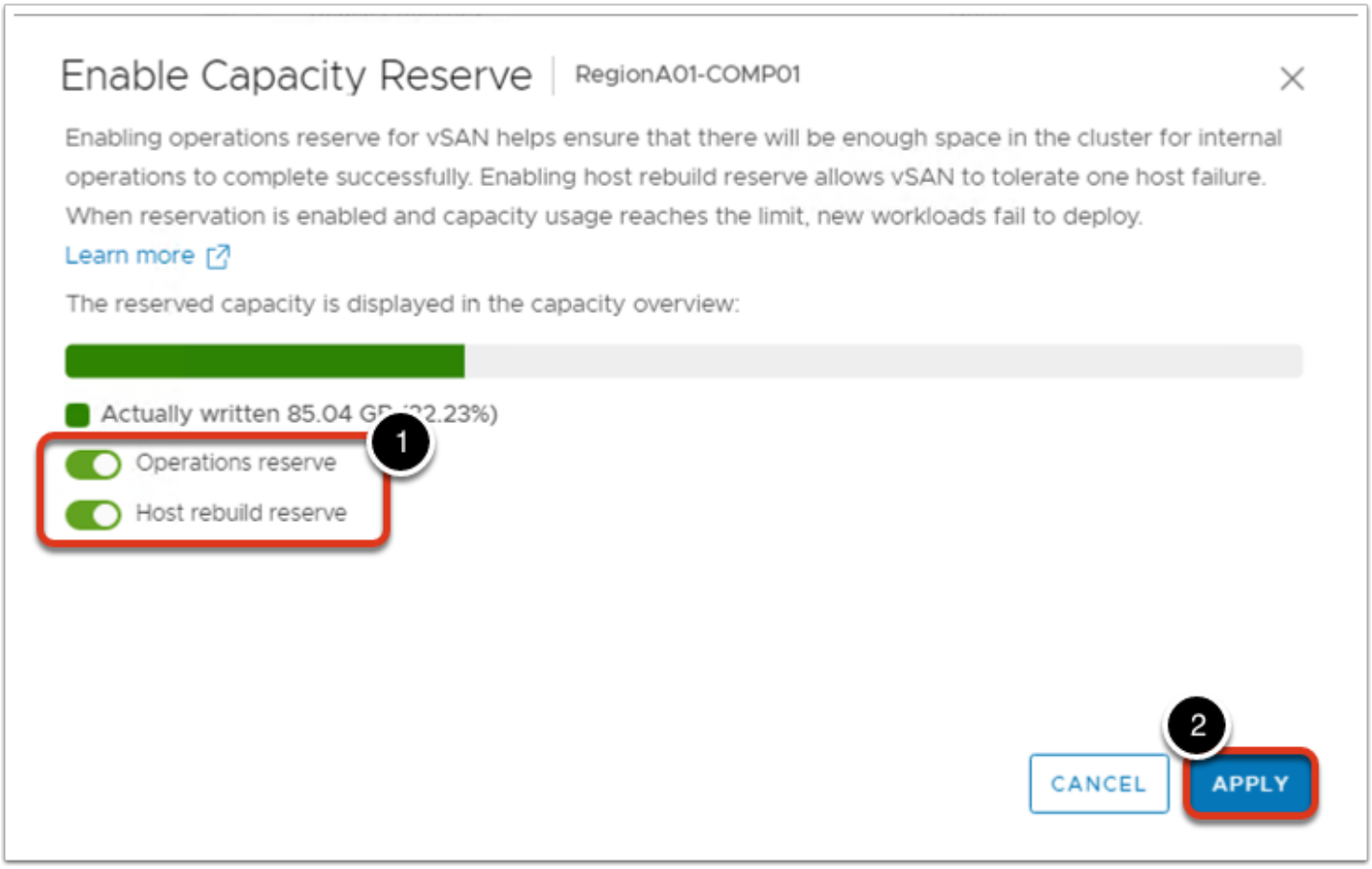
Configure Reserve Capacity



Once we edit the Enable Capacity reserve, you will be shown how much of the total vSAN datastore capacity (by default) is allocated to each reserve.

1. Navigate to the vSAN Cluster, **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > Services**
4. Under the **Enable Capacity Reserve**, click on **EDIT**

Enable Capacity Reserve



1. Toggle **Operations reserve** and **Host rebuild reserve** to be on
2. Click on **APPLY**

Conclusion

Storage Policy Based Management (SPBM) is a major element of your software-defined storage environment. It is a storage policy framework that provides a single unified control panel across a broad range of data services and storage solutions.

The framework helps to align storage with application demands of your virtual machines.

You Finished Module 1

Congratulations on completing Module 1.

How to End Lab

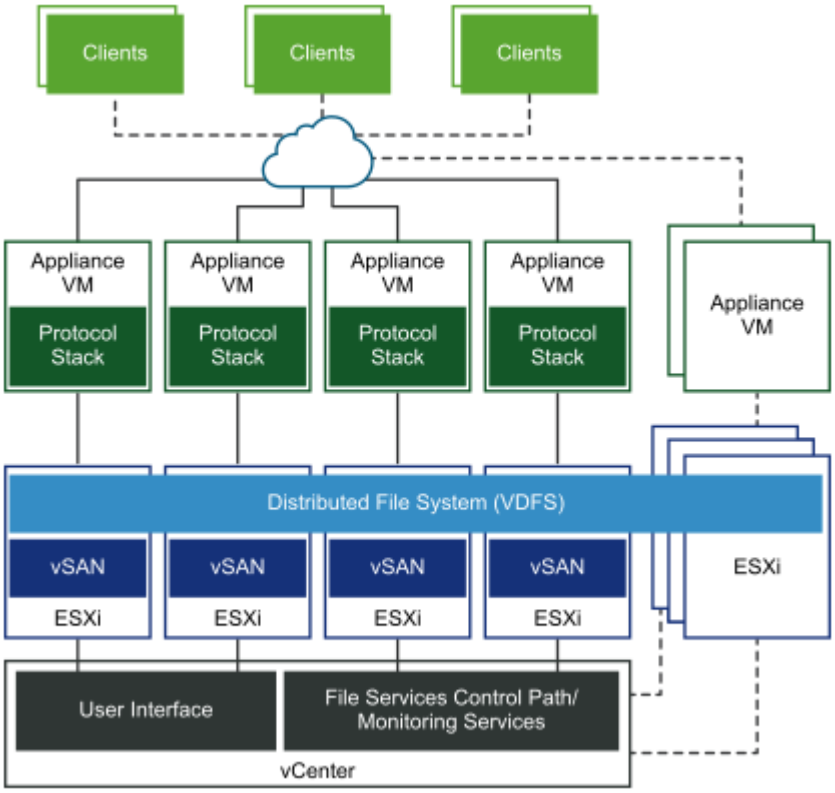
To end your lab, click on the **END** button.

Module 2 - vSAN File Services

vSAN File Services Overview

The vSAN File Services are used to create file shares in the vSAN datastore that client workstations or VMs can access. 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 SMB, NFSv3 and NFSv4.1 file shares. vSAN File Service comprises of vSAN Distributed File System (vDFS) which provides the underlying scalable filesystem by aggregating vSAN objects, a Storage Services Platform which provides resilient file server end points and a control plane for deployment, management, and monitoring. File shares are integrated into the existing vSAN Storage Policy Based Management, and on a per-share basis. vSAN File Service brings in capability to host the file shares directly on the vSAN cluster.



When you configure the vSAN File Service, vSAN creates a single vDFS distributed file system for the cluster which will be used internally for management purposes. vSAN File Services is powered and managed by the vSphere platform that deploys a set of containers on each of the hosts. These containers act as the primary delivery vehicle to provision file services and are tightly integrated with the hypervisor.

A static IP address pool should be provided as an input while enabling the file service workflow. One of the IP addresses is designated as the primary IP address. The primary IP address can be used for accessing all the shares in the File Services cluster with the

help of SMB and NFSv4.1 referrals. A file server is started for every IP address provided in the IP pool. However, the file shares are evenly distributed across all the file servers. To provide computing resources that help manage access requests, the number of IP addresses must be equal to the number of hosts in the vSAN cluster. These IP addresses must have DNS forward and reverse lookup entries.

- **Configure File Services**
  You can configure the File Service, which enables you to create file shares on your vSAN datastore. You can enable vSAN Files Service only on a regular vSAN cluster. Currently the File Service is not supported on a vSAN Stretched Cluster.
- **Create a File Share**
  When the vSAN File Service is enabled, you can create one or more file shares on the vSAN datastore. vSAN File Service does not support using these file shares as NFS datastores on ESXi.
- **View File Shares**
  You can view the list of vSAN file shares.
- **Access File Shares**
  You can access a file share from a host client, using an operating system that communicates with NFS file systems. For RHEL-based Linux distributions, NFS 4.1 support is available in RHEL 7.3 and CentOS 7.3-1611 running kernel 3.10.0-514 or later. For Debian based Linux distributions, NFS 4.1 support is available in Linux kernel version 4.0.0 or later. All NFS clients must have unique hostnames for NFSv4.1 to work. You can use the Linux mount command with the Primary IP to mount a vSAN file share to the client.
  - **Access NFS Kerberos File Share (**A linux client accessing an NFS Kerberos share should have a valid Kerberos ticket.)
  - **Access SMB File Share (**You can access an SMB file share from a Window client.)

- **Edit a File Share**
  You can edit the settings of a vSAN file share.
- **Delete a File Share**
  You can delete a file share when you no longer need it.
- **Upgrade File Share**
  When you upgrade the File Service, the upgrade is performed on a rolling basis. During the upgrade, the file server containers running on the virtual machines which are undergoing upgrade fails over to other virtual machines. The file shares remain accessible during the upgrade. During the upgrade, you might experience some interruptions while accessing the file shares.
- **Monitor Performance**
  You can monitor the performance of vSAN File Services.
- **Monitor Capacity**
  You can monitor the capacity for both native file shares and Cloud Native Storage managed file shares.
- **Monitor Health**
  You can monitor the health of both vSAN File Service and file share objects.

Enabling File Shares

To enable vSAN Files Services, we need to make sure we meet the network requirements:

- A static IP address to use as the single point of access to vSAN file shares. For best performance, the number of IP addresses must be equal to the number of hosts in the vSAN cluster.
- The static IP addresses should be part of the Forward lookup and Reverse lookup zones in the DNS server.
- All the static IP addresses should be from the same subnet.
- vSAN File services is supported on DVS version 6.6.0 or higher. Create a dedicated port group for vSAN File Service in the DVS.
- Promiscuous Mode and Forged Transmits are enabled as part of the vSAN File Services enablement process for provided network entity. If NSX based networks are being used, ensure that similar settings are configured for the provided network entity from the NSX admin console.

We have already ENABLED File Services in this lab.

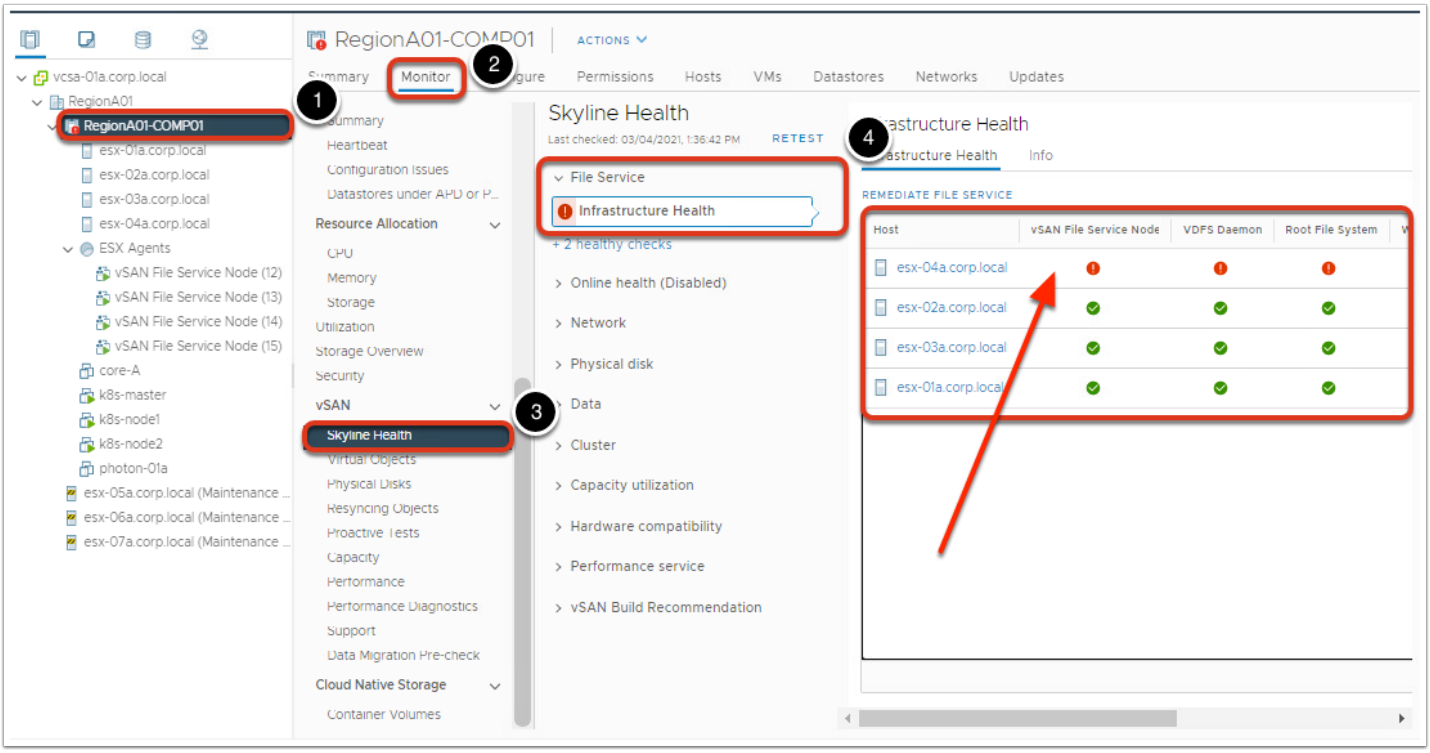
Enable File Services

As we said, vSAN File Services is already ENABLED. We can checked that it was created.

1. Select vSAN Cluster (**RegionA01-COMP01**)
2. Select **Configure**
3. Select **vSAN > Services**
4. vSAN Services : **File Service = Enabled**
5. Expand the **Files Services**, you can see the network details of the file services

You can see the vSAN File Service Nodes were created. One for each vSAN node enabled and up to 32 file shares and 8 file servers (vSAN 7.0) and 32 file servers (vSAN 7.0 update 1) can be configure for vSAN File Services.

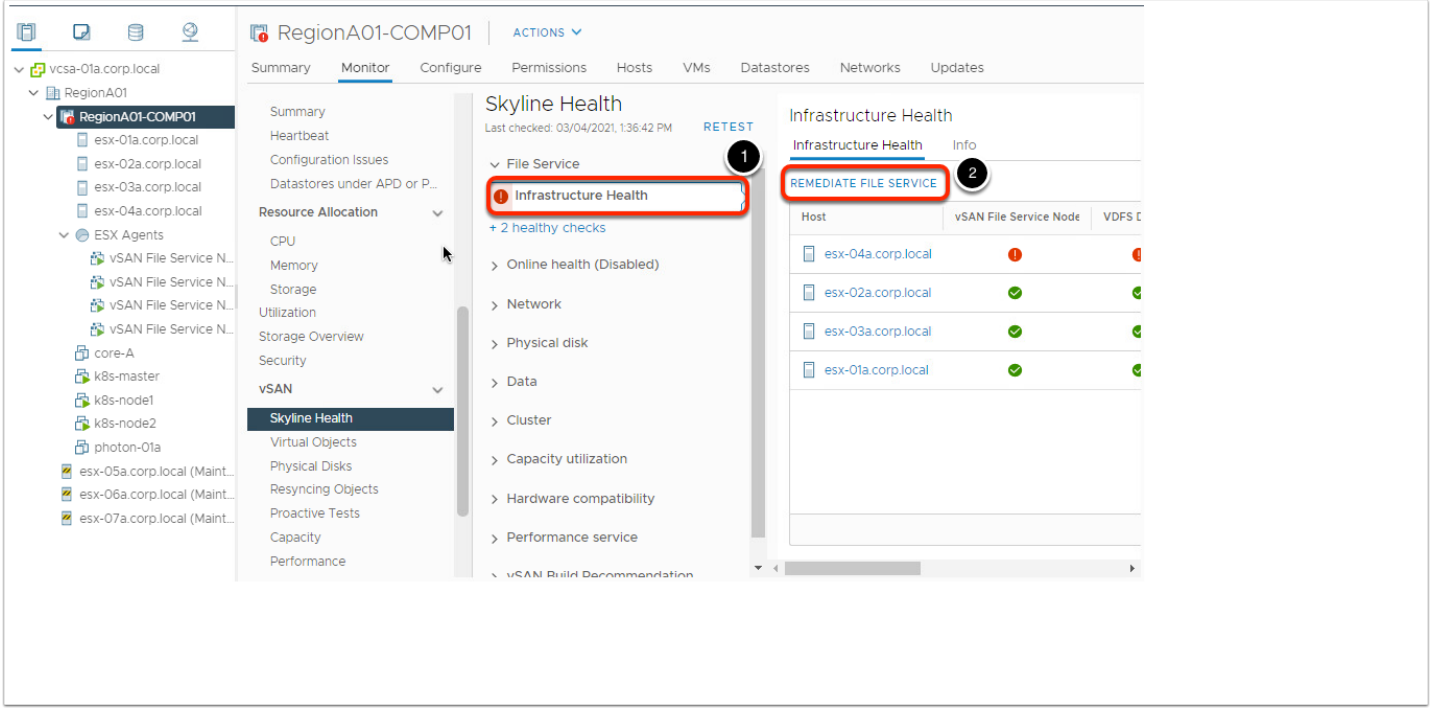
Enable File Server



1. Select vSAN Cluster, **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Skyline Health**
4. Select **File Server Health**

Notice there are only 3 hosts that are contributing into the File Servers even though we add the 4th node into the vSAN cluster, we see there are errors.

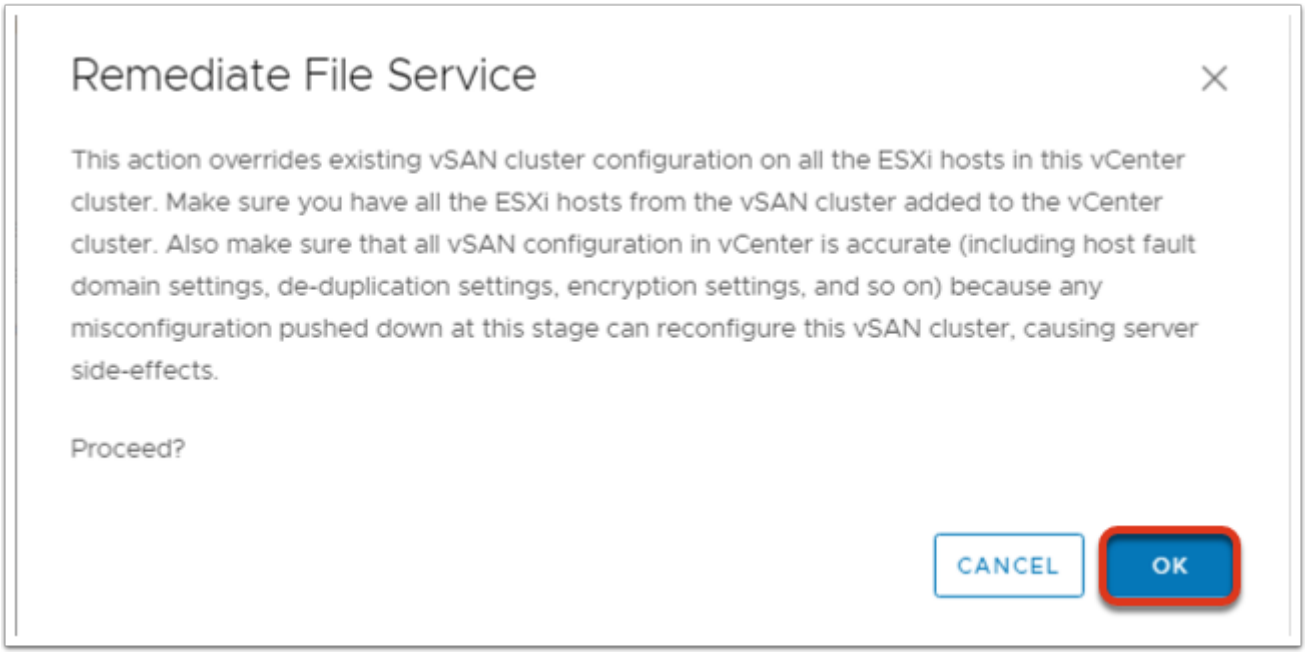
Remediate File Server



Since we add the 4th node in the vSAN cluster in Module 1, you notice that there are 4 vSAN File Service Node however it is not part of the File Services even though it was created.

1. Select **Infrastructure Health**
2. Click on **REMEDIATE FILE SERVICE**

Remediate File Service Continue



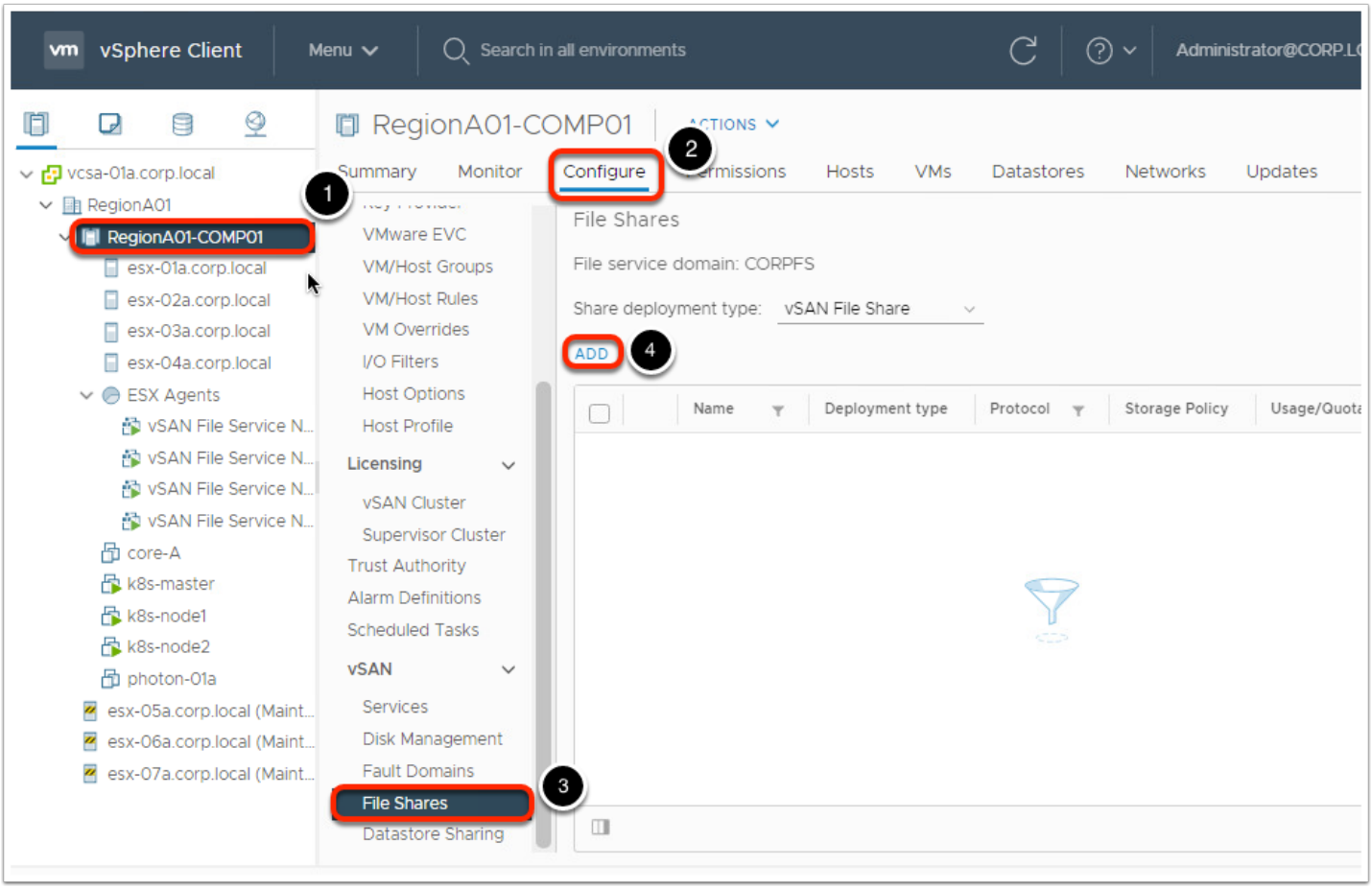
Click **OK**

Creating NFS File Shares

To create vSAN Files Services for NFS, consider the followings for share names and usages:

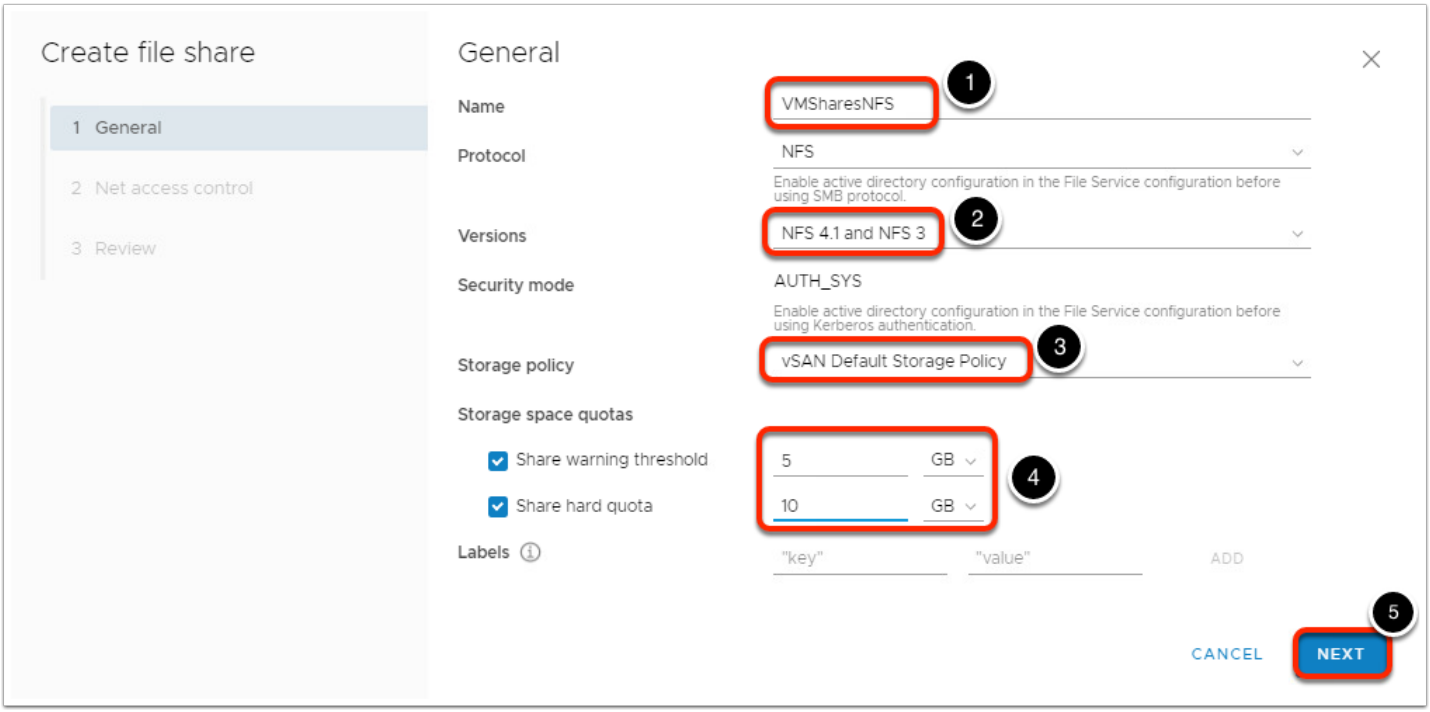
- Usernames with non-ascii characters can be used to access share data.
- Share names can contain only English characters.
- For Pure NFSv4 type shares, the file and directories can only contain any UTF-8 compatible.
- For Pure NFSv3 and NFSv3+NFSv4 shares file and directories can contain only ASCII compatible strings.

Create a File Share



1. Select the cluster **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > File Shares**
4. Click **ADD**

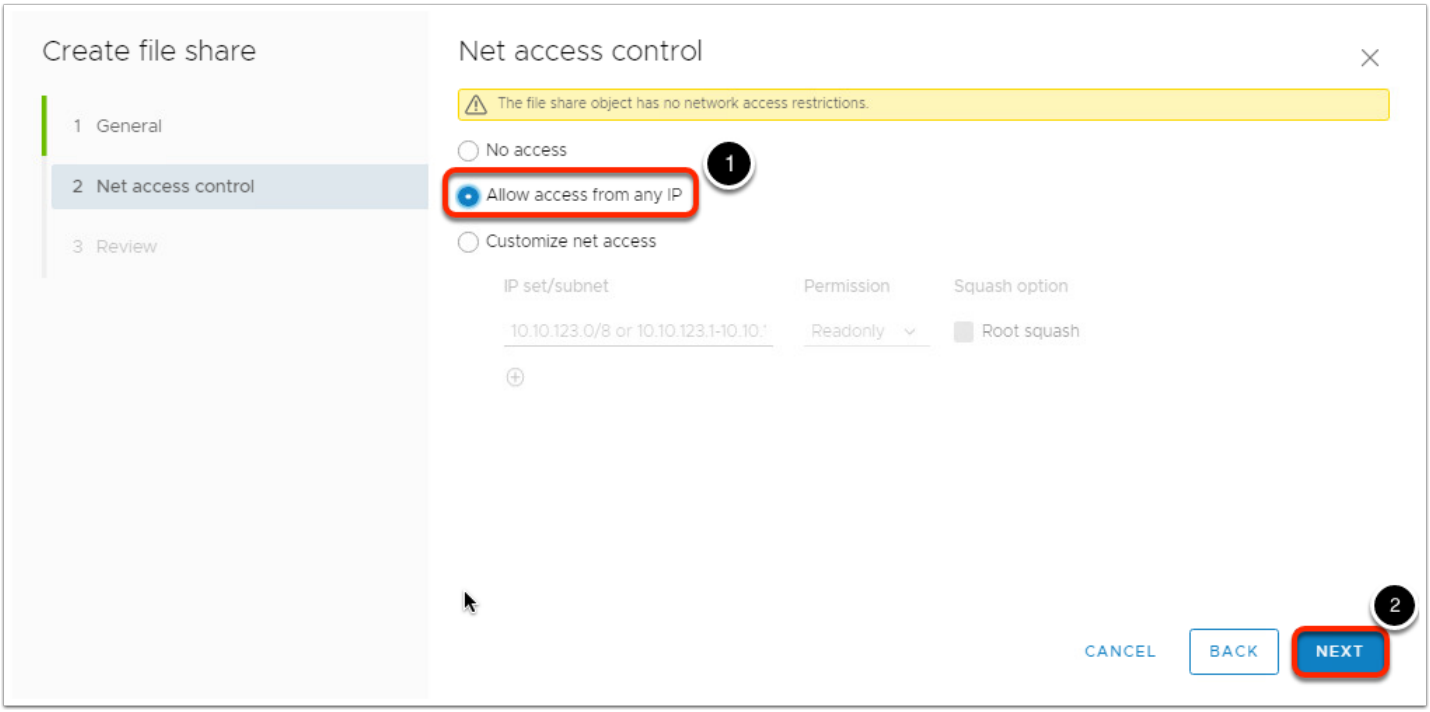
Create File Share Cont'



In the General page,

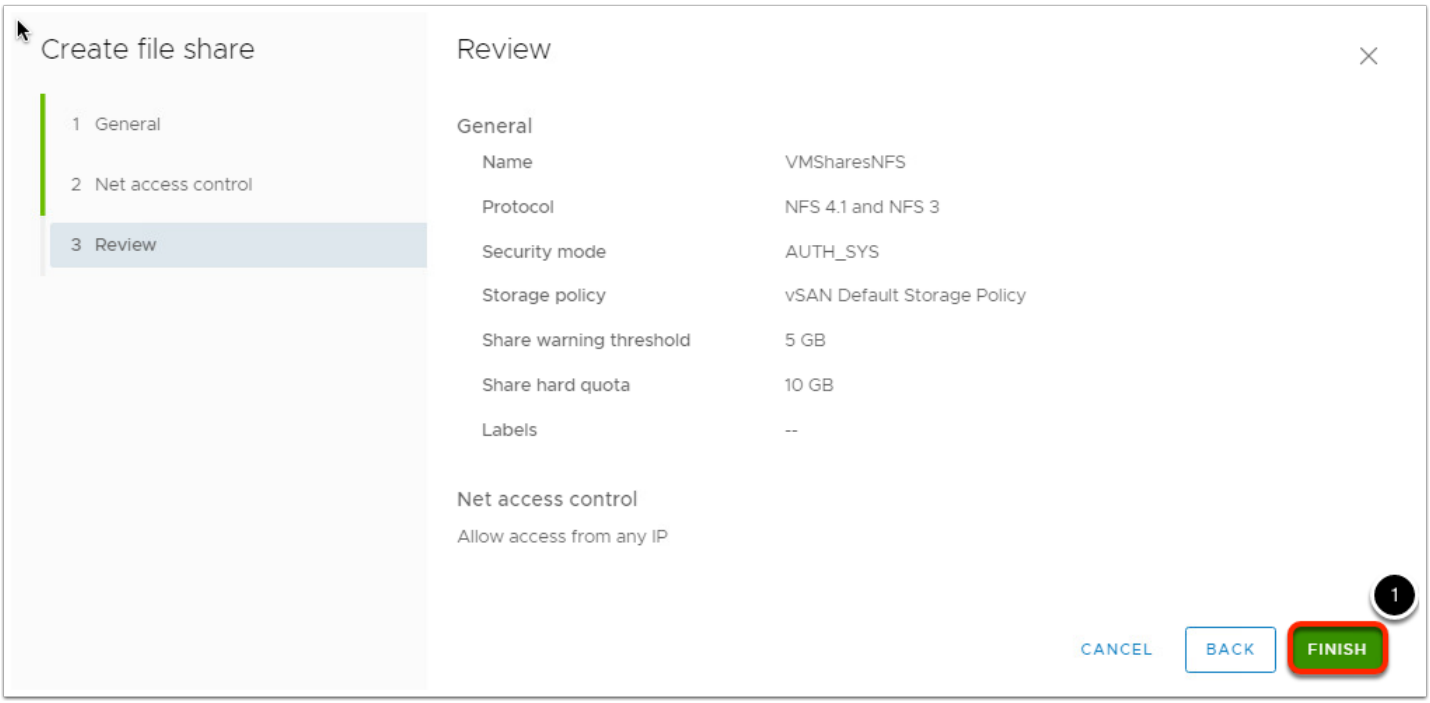
1. Enter the **Name** of the Shares: VMSharesNFS
2. Select **Versions**: NFS 4.1 and NFS 3
3. Select **Storage Policy**: vSAN Default Storage Policy
4. Enter the threshold and hard quota for the storage: **5 GB** threshold and **10 GB** for hard quota
5. Click **NEXT**

Create File Share



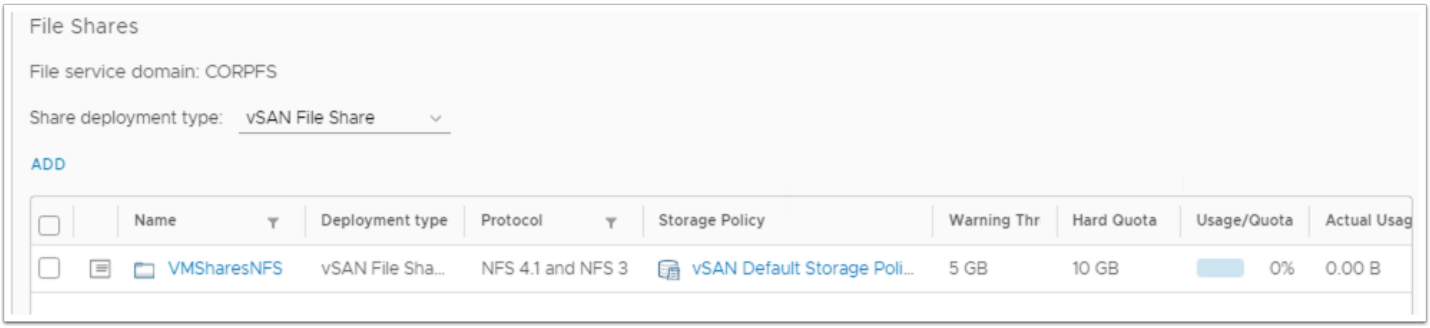
In the Net access control page,

1. Select **Allow access from any IP**
2. Click **NEXT**



1. You can review what you enter then Click **FINISH**

Create File Share



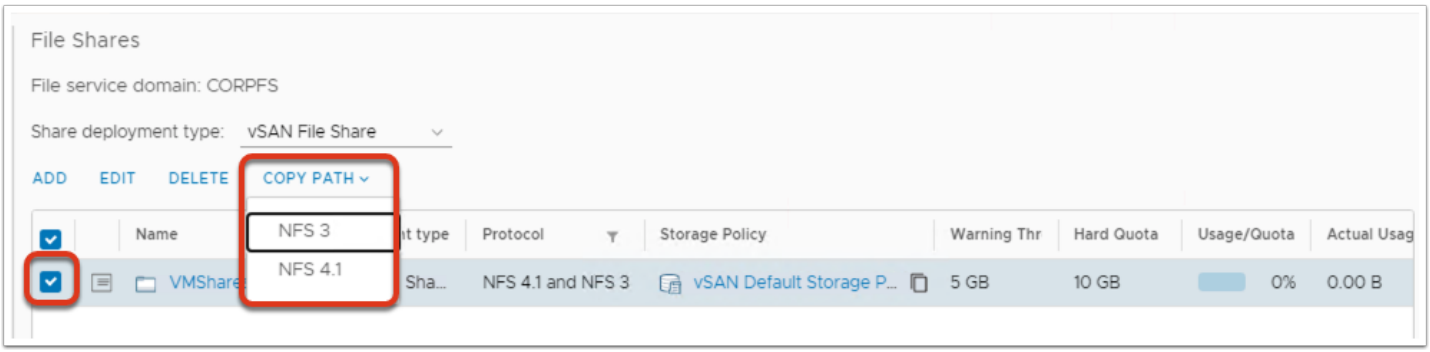
To view the list of vSAN file shares, navigate to the vSAN cluster and click **Configure** > **vSAN** > **File Shares**.

You can view the list of vSAN file shares we just created called **VMSharesNFS** with the appropriate storage policy, hard quota, usage over quota, actual usage, and so on.

Mounting vSAN NFS File Shares to other Systems

You can access a file share from a host client, using an operating system that communicates with NFS file systems. For RHEL-based Linux distributions, NFS 4.1 support is available in RHEL 7.3 and CentOS 7.3-1611 running kernel 3.10.0-514 or later. For Debian based Linux distributions, NFS 4.1 support is available in Linux kernel version 4.0.0 or later. All NFS clients must have unique hostnames for NFSv4.1 to work. You can use the Linux mount command with the Primary IP to mount a vSAN file share to the client.

Mount File Service Shares



1. Check the box next to **VMShareNFS**
2. Select **COPY URL,** you can chose from **NFSv3 or NFSv4.1**

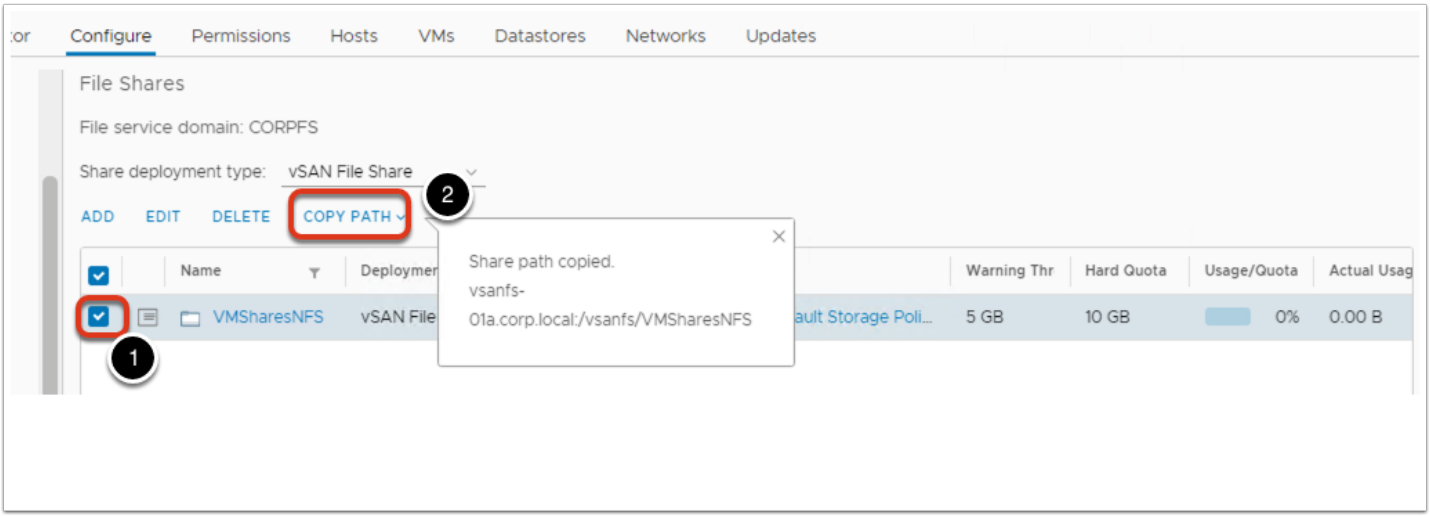
Connection string copied.

NFSv3: vsanfs-02a.corp.local:/VMSharesNFS

NFSv4.1: vsanfs-01a.corp.local:/vsanfs/VMSharesNFS

To mount this file share as NFS v4.1 and the NFSv4.1 referral mechanism, you need to include the root share (/vsanfs) in the mount path

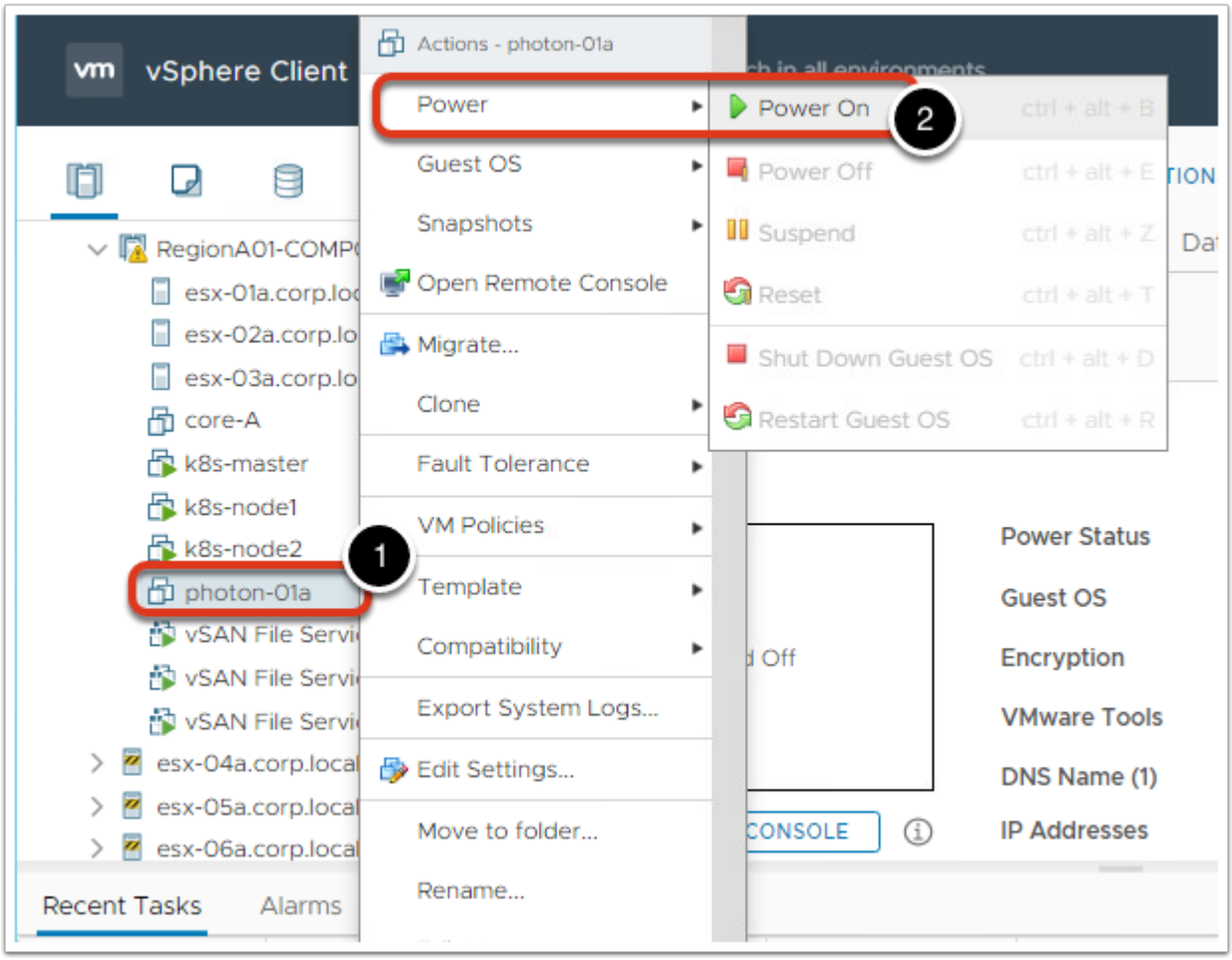
Mount File Service Shares



1. Check the box next to **VMShareNFS**
2. Select **COPY PATH,** then select **NFSv4.1**

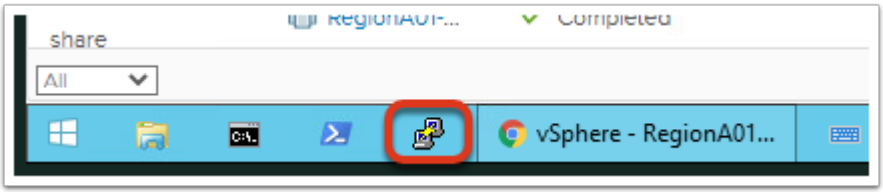
We will use NFSv4.1 to mount the shares.

PowerOn VM, photon-01a



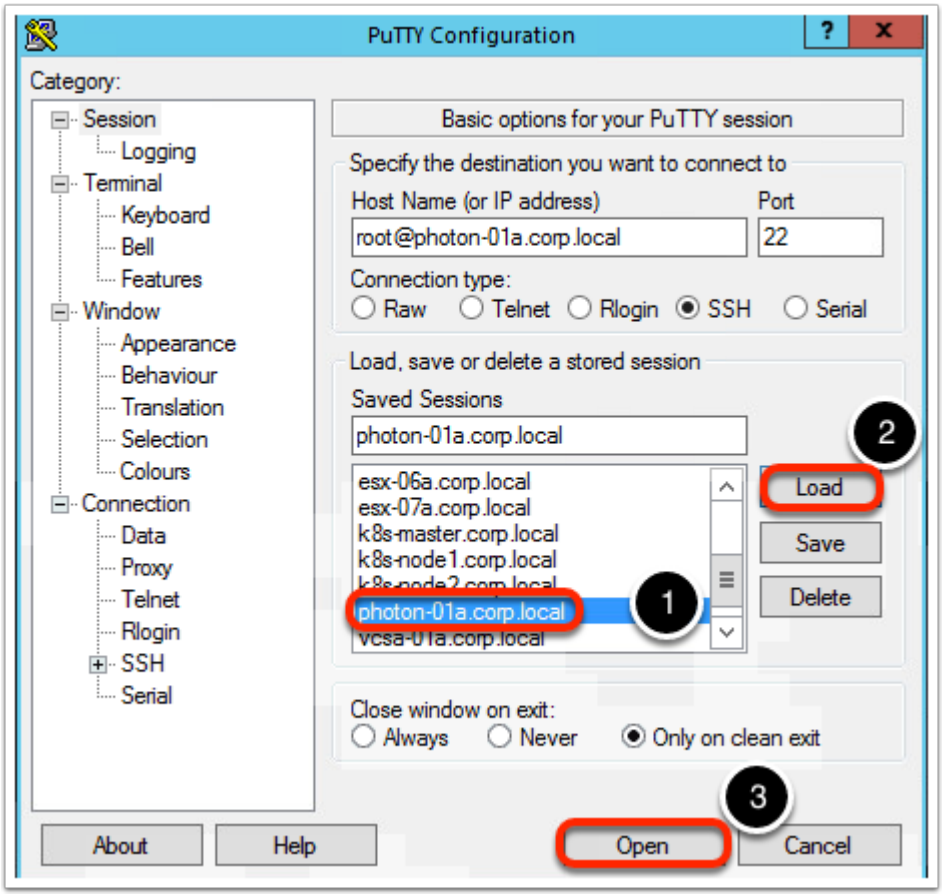
1. Right click **photon-01a**
2. Select **Power > Power On**

Launch PuTTy



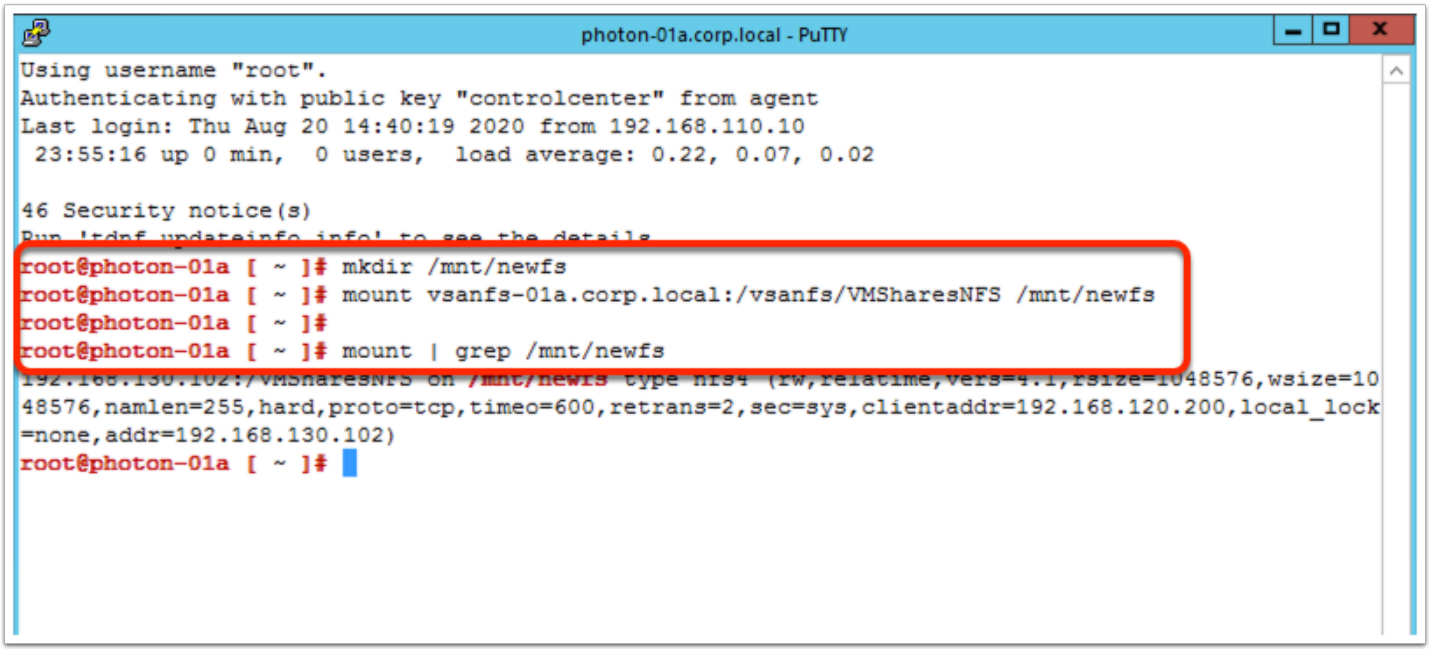
1. Launch the **PuTTy** application from the Windows Taskbar.

Choose photon-01a



1. Select the linux vm called **photon-01a.corp.local** (you might need to scroll down)
2. Select **Load**
3. Select **Open**

Mount vSAN File Shares

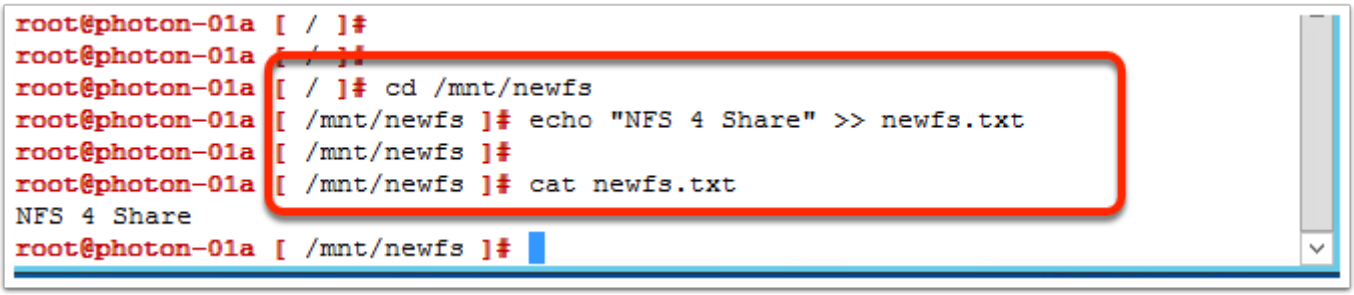


Below is the command to mount the file share using NFS v4.1. NFS v4.1 is also the default mount version if no protocol if specified. In that case, the client will negotiate the mount protocol with the server and mount with the highest matching protocol, which for vSAN 7 Native File Services is NFS v4.1.

By typing:

```
root@photon-01a[ ~ ]# mkdir /mnt/newfs
root@photon-01a[ ~ ]# mount vsanfs-01a.corp.local:/vsanfs/VMSharesNFS /mnt/newfs
root@photon-01a[ ~ ]# mount | grep /mnt/newfs
```

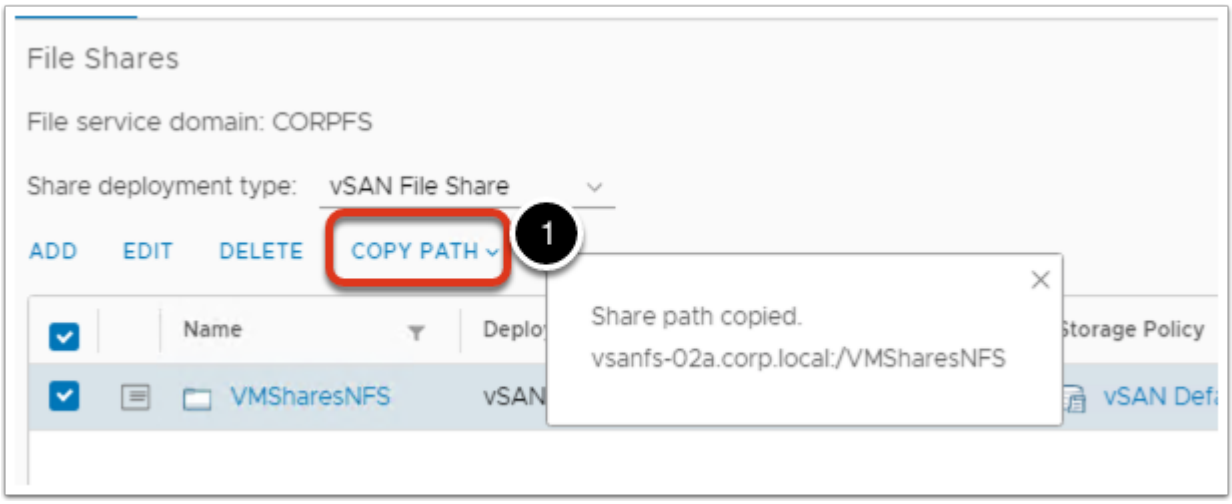
Mount vSAN FIle Shares



Let's write a file to the file share and confirmed it worked.

```
root@photon-01a[ ~ ]# cd /mnt/newfs
root@photon-01a[ ~ ]# echo "NFS 4 Share" >> newfs.txt
root@photon-01a[ ~ ]# cat newfs.txt
```

Mount vSAN FIle Shares



Let us mount to the NFSv3:

1. Select **COPY PATH**, then select **NFSv3**

Take notice of the new connection string. It might be different from what is picture since it is on a pool of IP

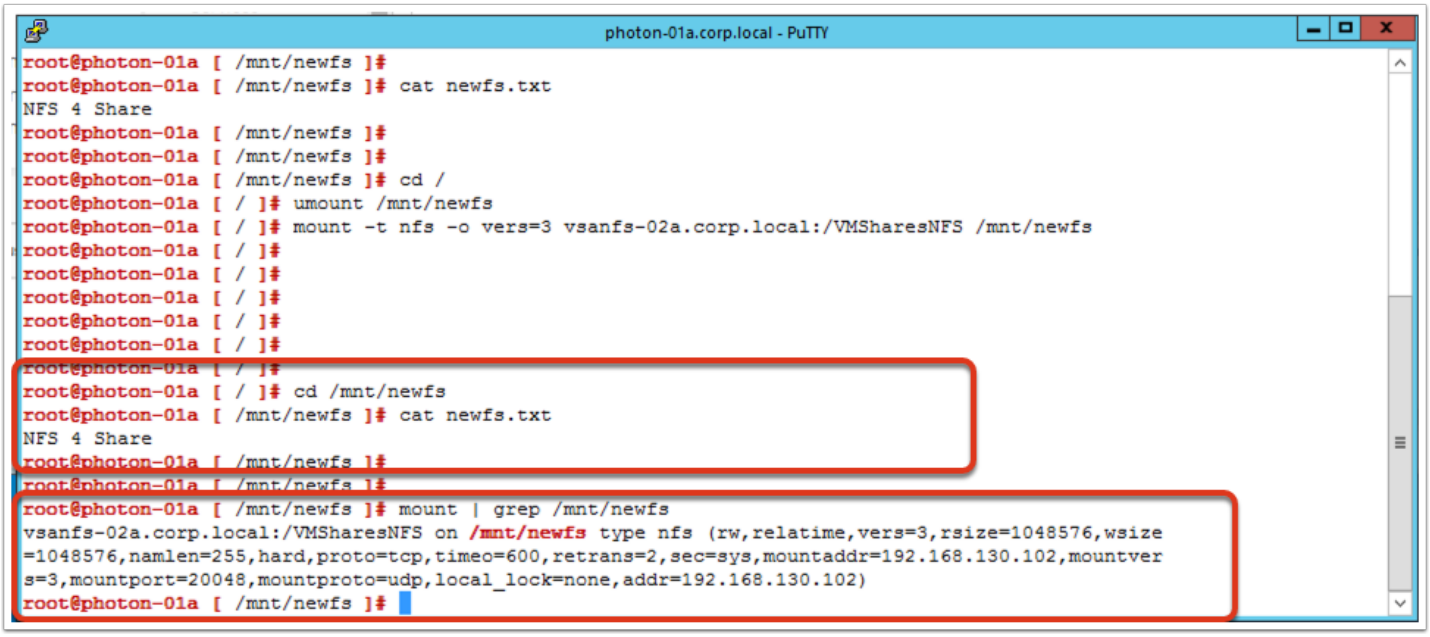
Mount vSAN File Shares

```
photon-01a.corp.local - PuTTY
 23:55:16 up 0 min,  0 users,  load average: 0.22, 0.07, 0.02

46 Security notice(s)
Run 'tdnf updateinfo info' to see the details.
root@photon-01a [ ~ ]# mkdir /mnt/newfs
root@photon-01a [ ~ ]# mount vsanfs-01a.corp.local:/vsanfs/VMSharesNFS /mnt/newfs
root@photon-01a [ ~ ]#
root@photon-01a [ ~ ]# mount | grep /mnt/newfs
192.168.130.102:/VMSharesNFS on /mnt/newfs type nfs4 (rw,relatime,vers=4.1,rsize=1048576,wsize=10
48576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=192.168.120.200,local_lock
=none,addr=192.168.130.102)
root@photon-01a [ ~ ]#
root@photon-01a [ ~ ]#
root@photon-01a [ ~ ]# cd /mnt/newfs
root@photon-01a [ /mnt/newfs ]# echo "NFS 4 Share" >> newfs.txt
root@photon-01a [ /mnt/newfs ]#
root@photon-01a [ /mnt/newfs ]# cat newfs.txt
NFS 4 Share
root@photon-01a [ /mnt/newfs ]#
root@photon-01a [ /mnt/newfs ]#
root@photon-01a [ /mnt/newfs ]# cd /
root@photon-01a [ / ]# umount /mnt/newfs
root@photon-01a [ / ]# mount -t nfs -o vers=3 vsanfs-02a.corp.local:/VMSharesNFS /mnt/newfs
root@photon-01a [ / ]#
```

Switch back to the putty session for 'Photon-01a' and run:

NOTE - Use the Connection Strings for NFSv3 that is display in your lab. Since it is on a pool of IP, it might be different from what is shown above in the manual.

```
root@photon-01a[ ~ ]# cd /
root@photon-01a[ ~ ]# umount /mnt/newfs
root@photon-01a[ ~ ]# mount -t nfs -o vers=3 vsanfs-02a.corp.local:/VMSharesNFS /mnt/
newfs
```




```
root@photon-01a[ ~ ]# cd /mnt/newfs
root@photon-01a[ ~ ]# cat newfs.txt
```

You should see 'NFS 4 Share' as a return value

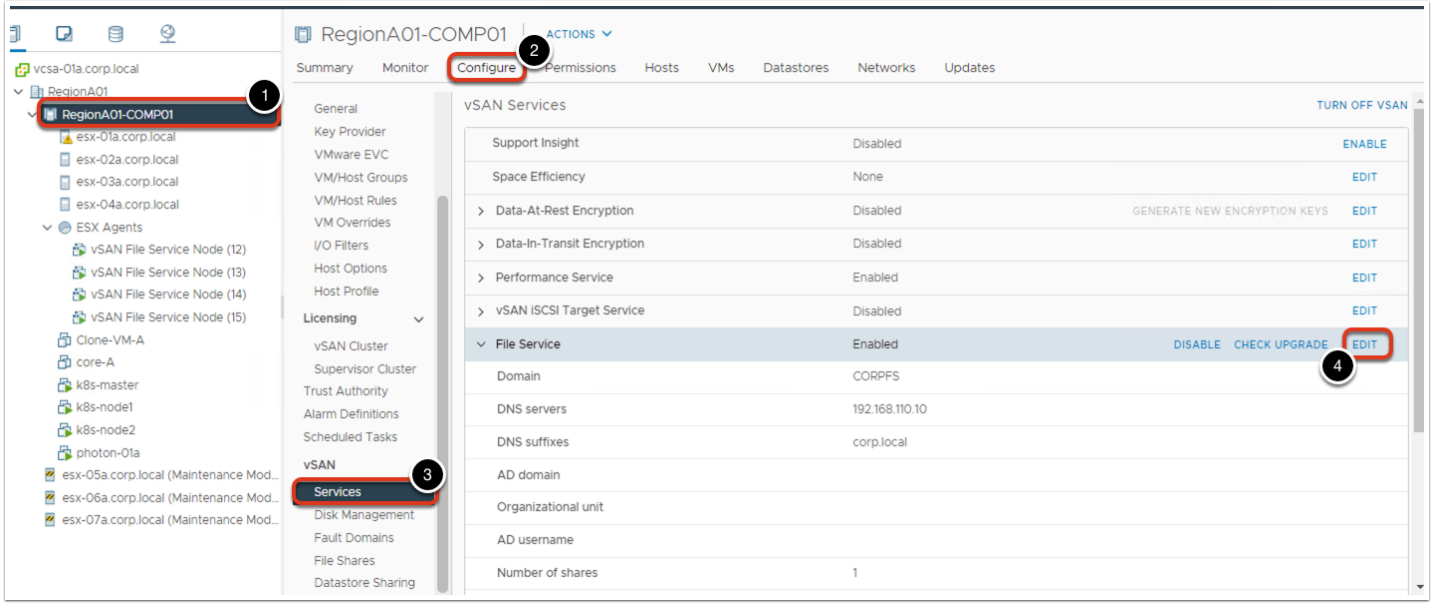
```
root@photon-01a[ ~ ]# mount | grep /mnt/newfs
```

You can see the /mnt/newfs is now mounted via NFSv3

Creating SMB File Shares

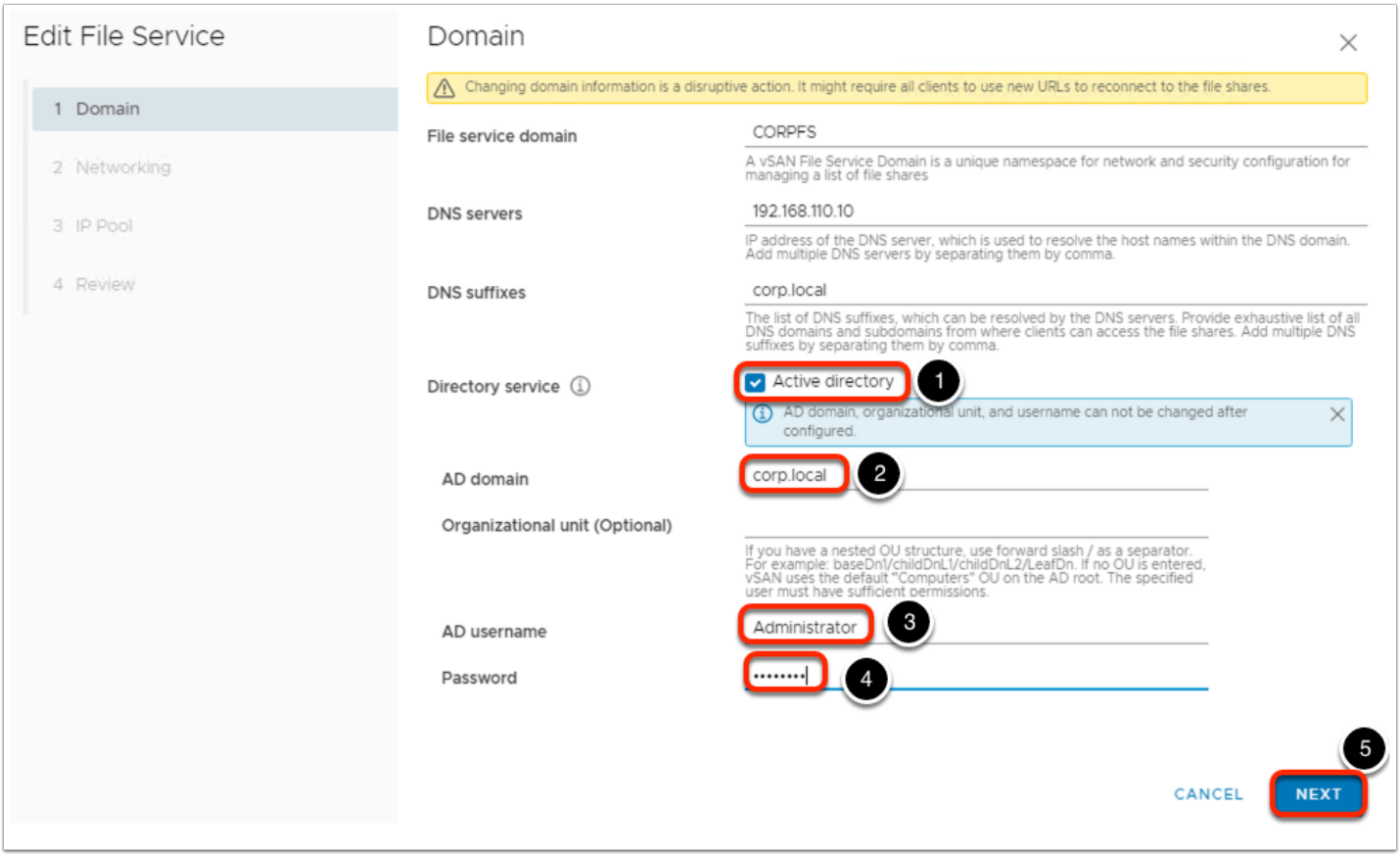
If you are creating a SMB file share or an NFSv4.1 file share with Kerberos security, then ensure that you have configured vSAN File Service with AD domain credentials.

Enable AD Domain for File Share



1. Select vSAN Cluster, **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > Services**
4. Click on **EDIT** in the File Service

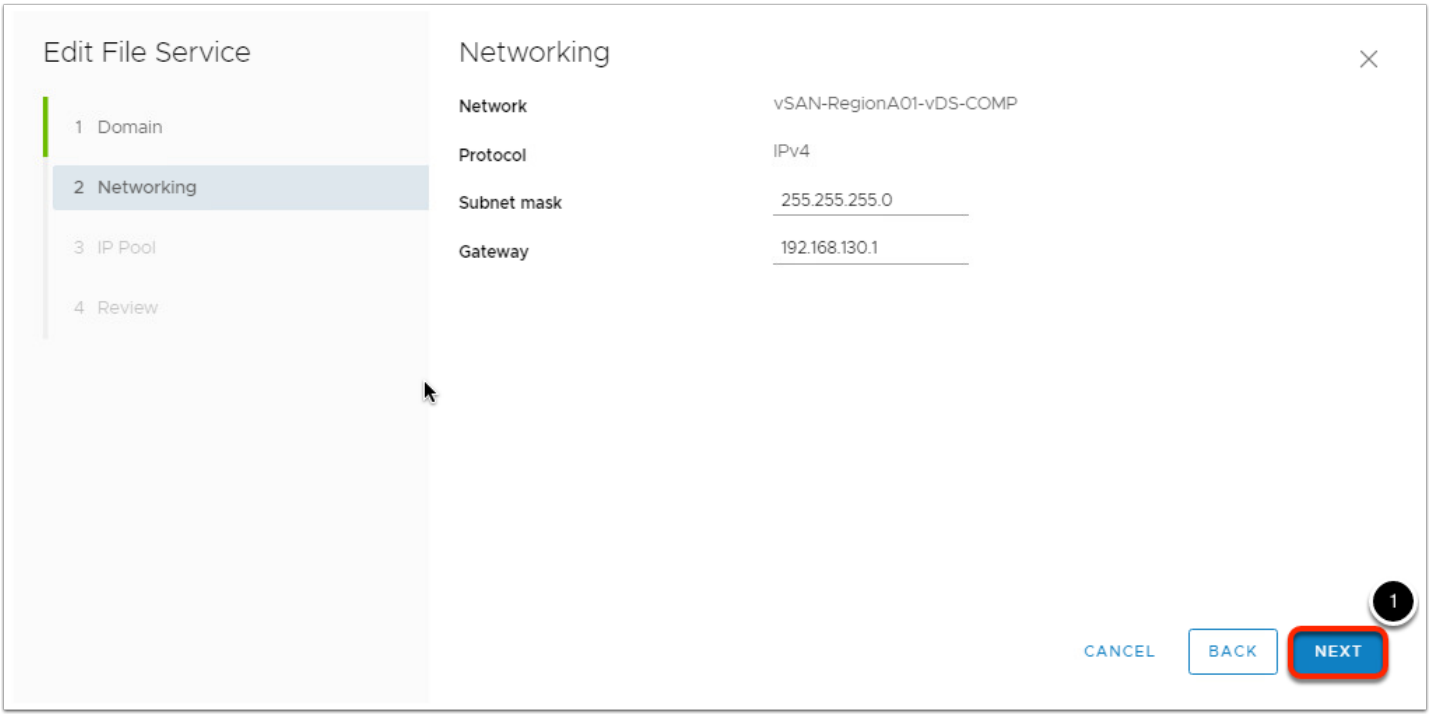
Edit File Service - Domain



Since we already enabled vSAN File Services and add the appropriate DNS services for the NFS file shares, we need to add the Active Directory domain information.

1. Click on **Active Directory** to enable
2. Enter AD domain: **corp.local**
3. Enter AD username: **Administrator**
4. Enter password: **VMware1!**
5. Click on **NEXT**

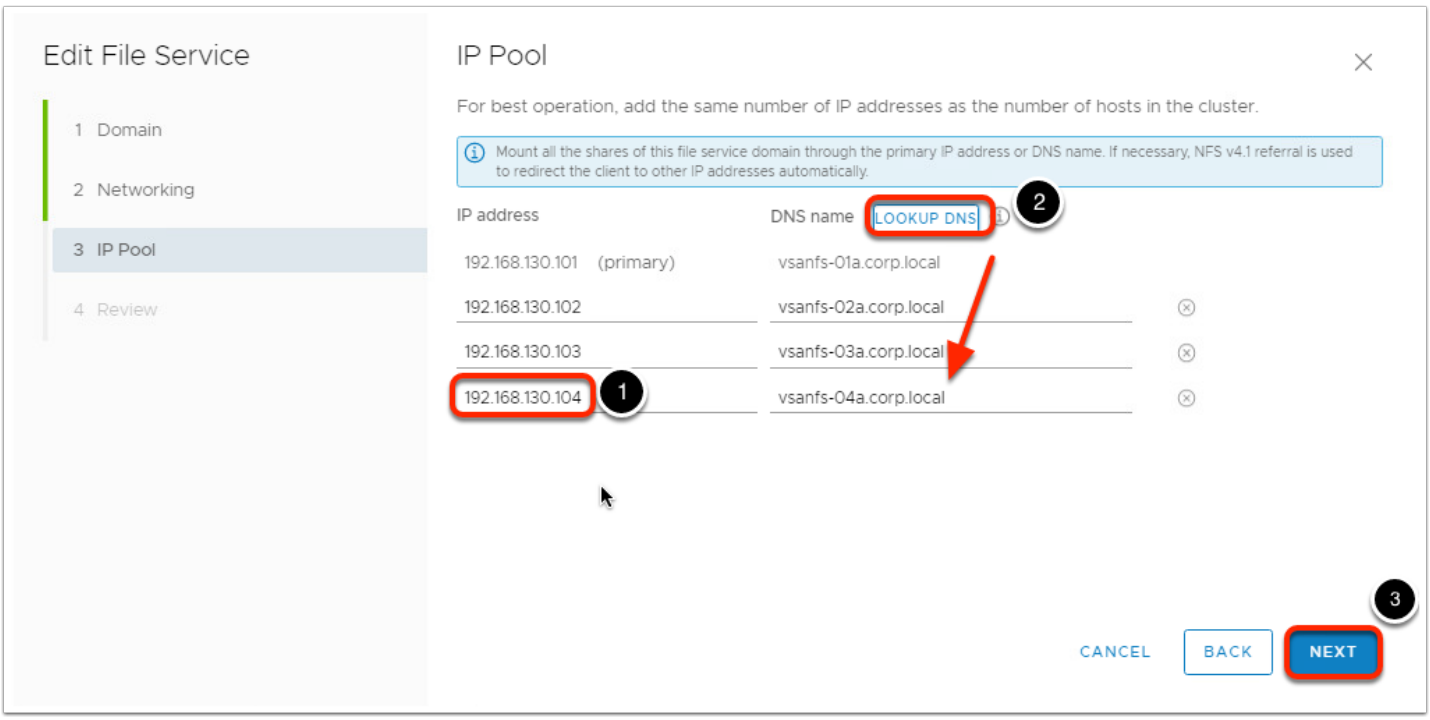
Edit File Service - Networking



Networking information is automatically filled when we enabled the vSAN File Service

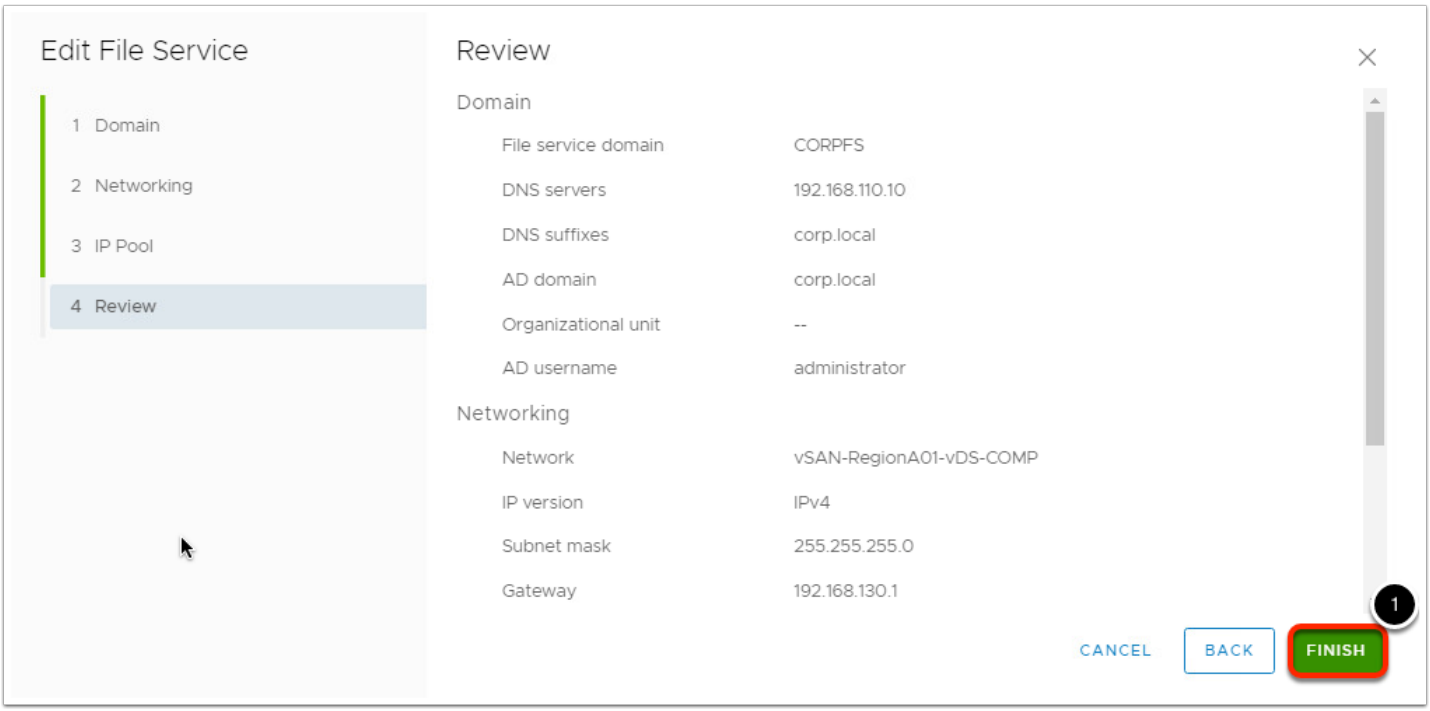
1. Click **NEXT**

Edit File Service - IP Pool



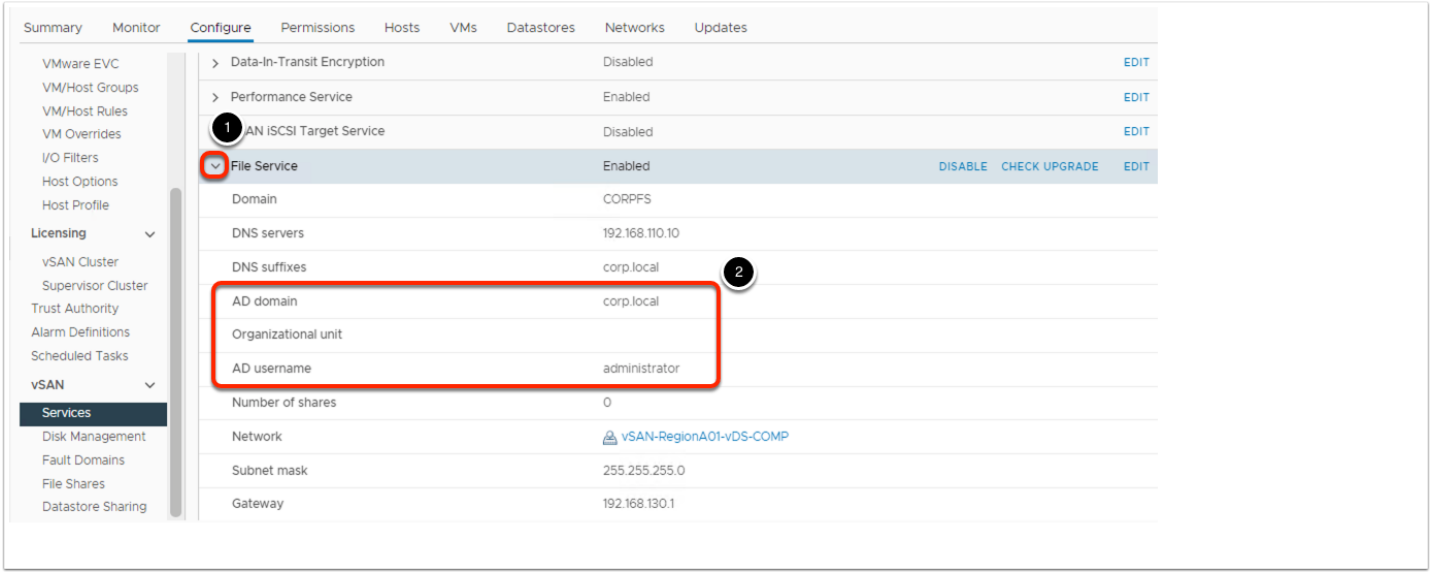
Since we added the 4th node in the vSAN Cluster, we can add the 4th file service server.

1. Enter IP Address: **192.168.130.104**
2. Click on **LOOKUP DNS.** It will automatically fill in the DNS name for the IP address
3. Click **NEXT**



1. Click **FINISH** on the Review page

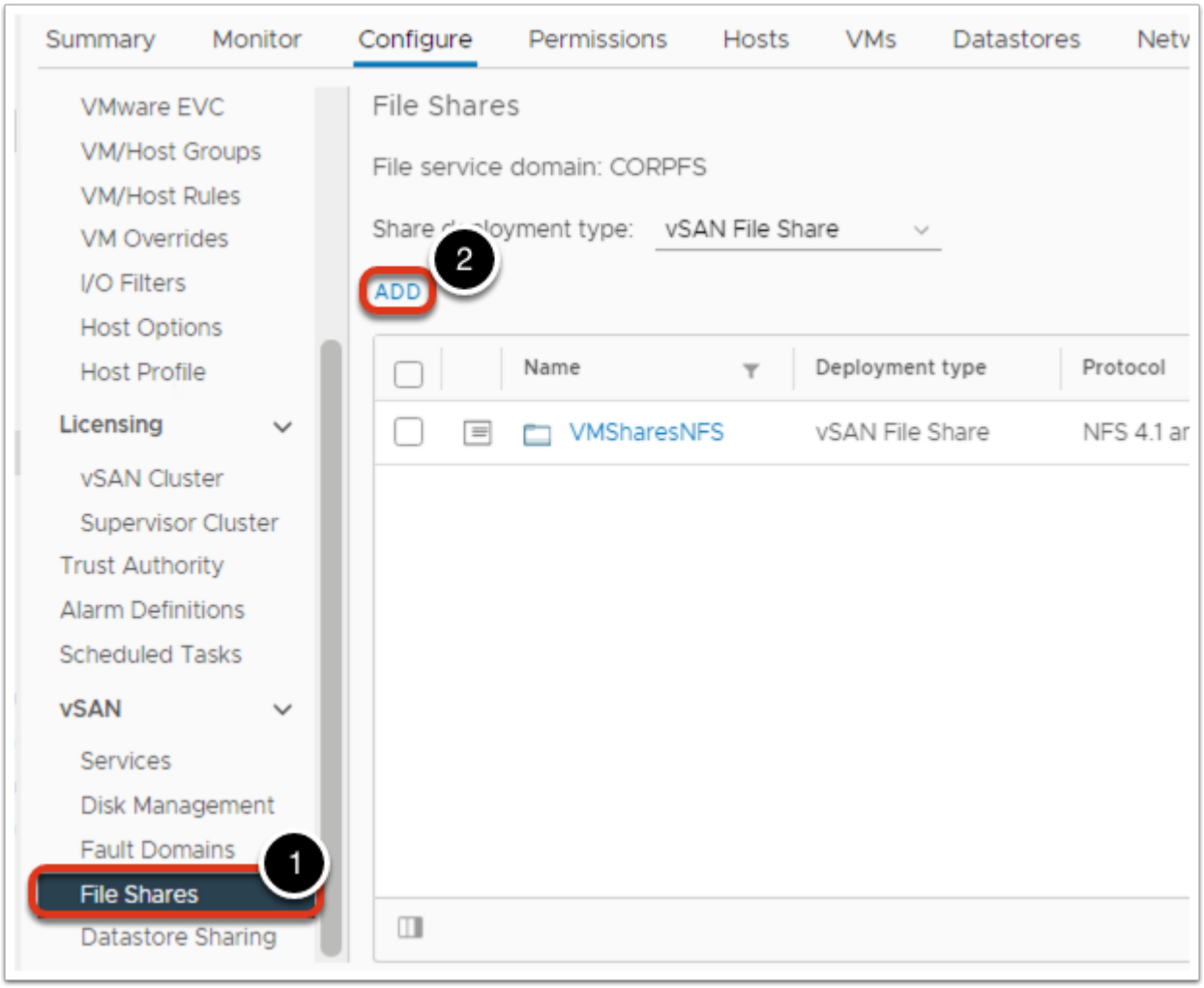
File Service - AD Domain enabled



1. Expand the **File Services** section

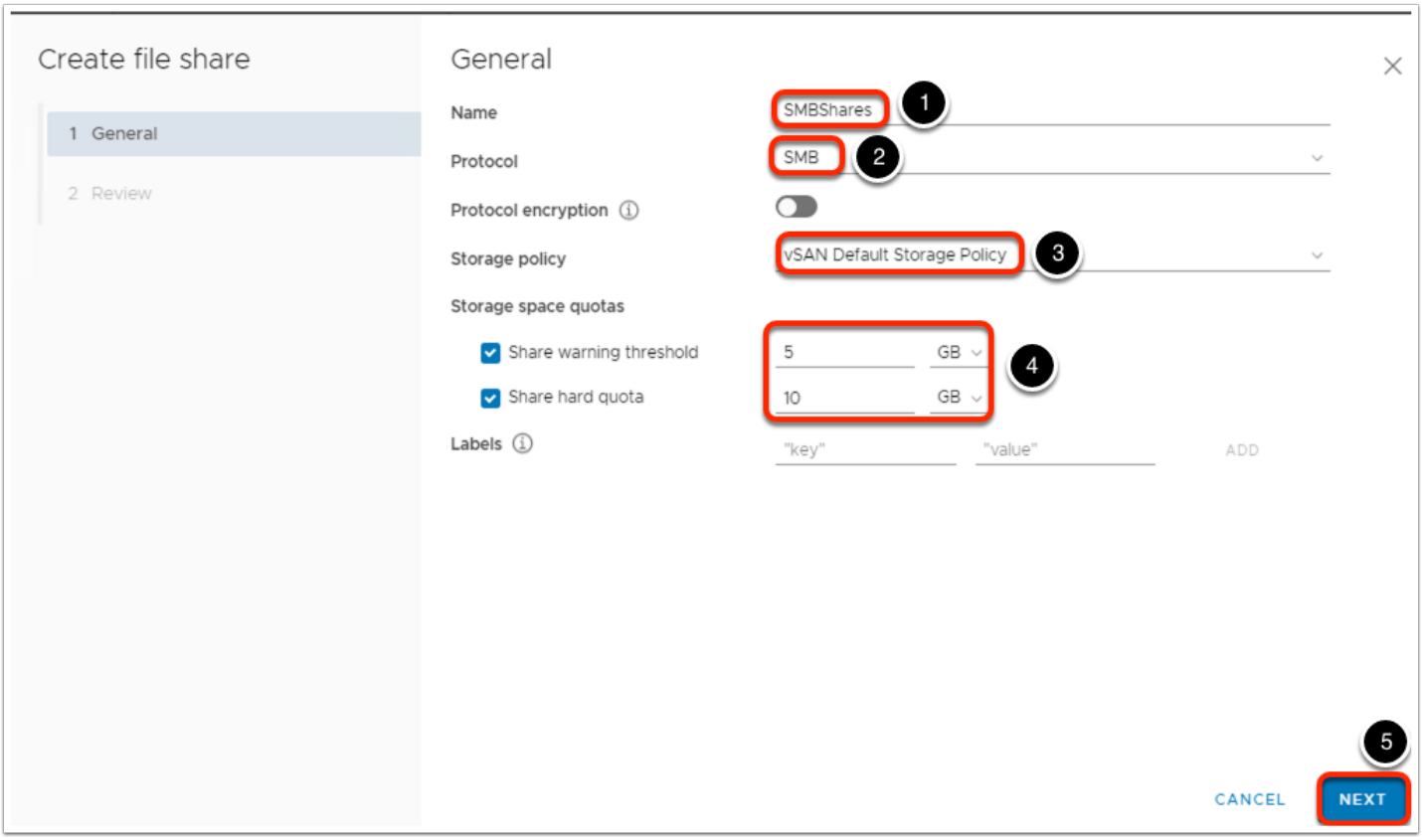
2. You can see that the AD domain and username are part of the File Service

Create SMB File Share



1. Select **vSAN > File Shares**
2. Click **ADD**

Create File Share - General

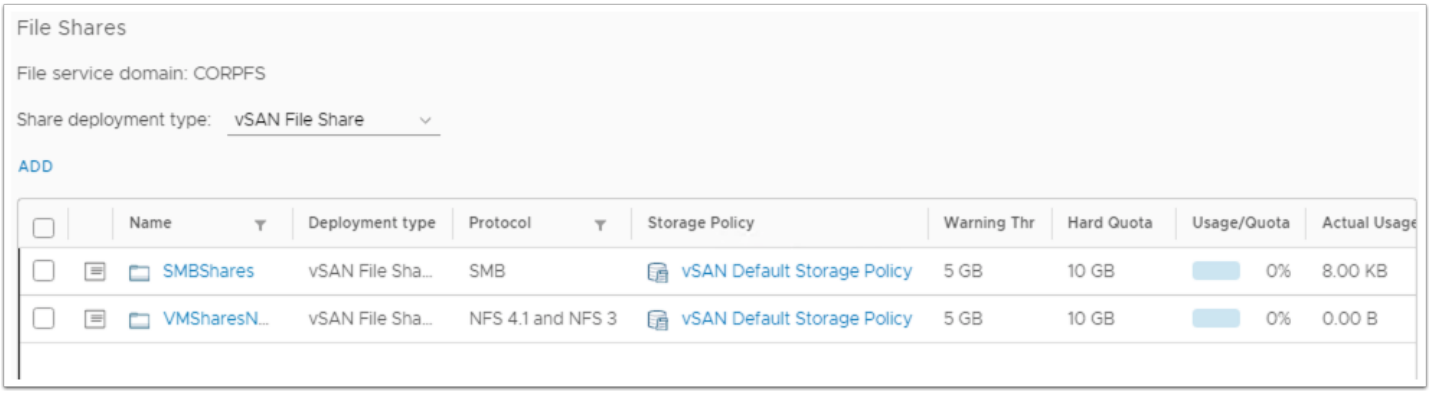


In the General page,

1. Enter the **Name** of the Shares: SMBShares
2. Select Protocol: **SMB**
3. Select **Storage policy**: vSAN Default Storage Policy
4. Enter the threshold and hard quota for the storage: **5 GB** threshold and **10 GB** for hard quota
5. Click **NEXT**

Click **FINISH** on the Review page

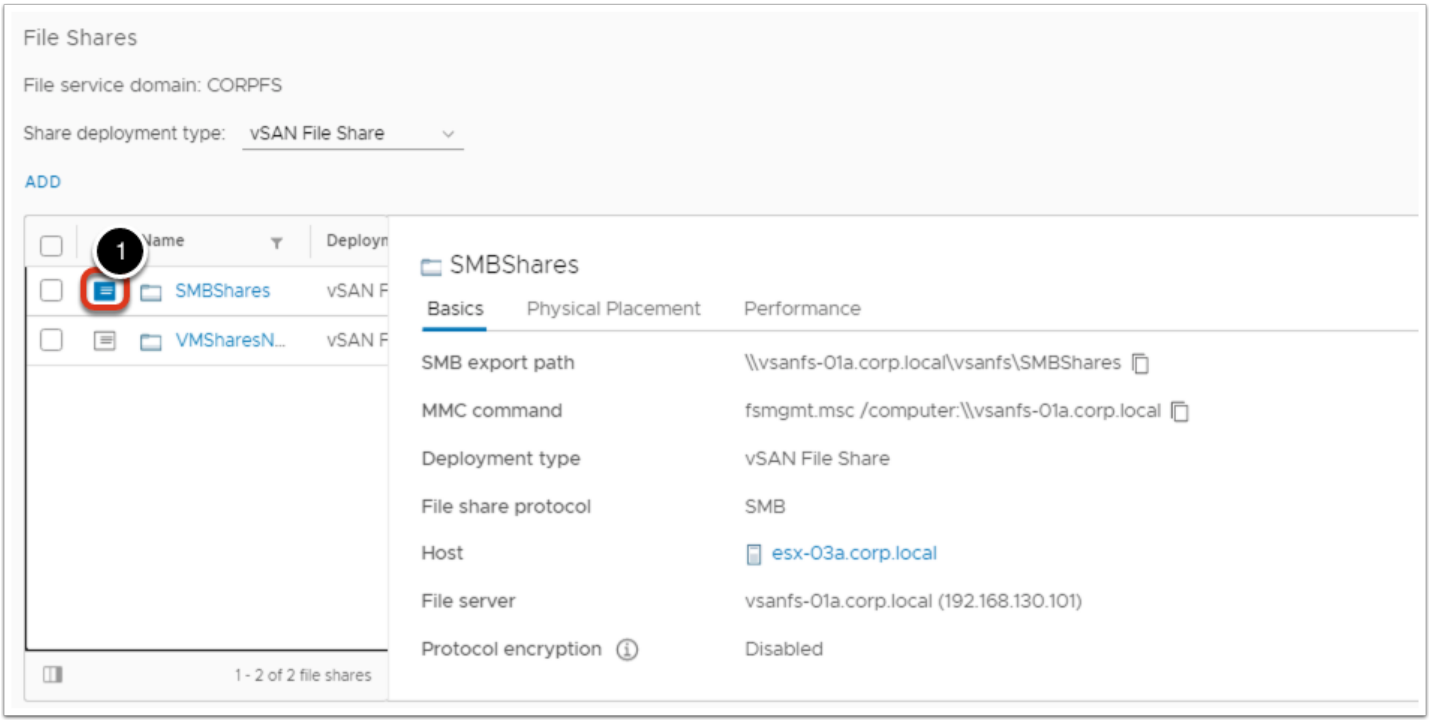
Create File Share



To view the list of vSAN file shares, navigate to the vSAN cluster and click **Configure** > **vSAN** > **File Shares**.

You can view the list of vSAN file shares we just created called **VMShares** for NFS and **SMBShares** for SMB with the appropriate storage policy, hard quota, usage over quota, actual usage, and so on.

View File Shares



1. Click on the button next to the **SMBShares**

You have the ability to review the physical placement of the SMB file share across the hosts and disks in the vSAN Cluster. Also can view the SMB export path and the MMC command.

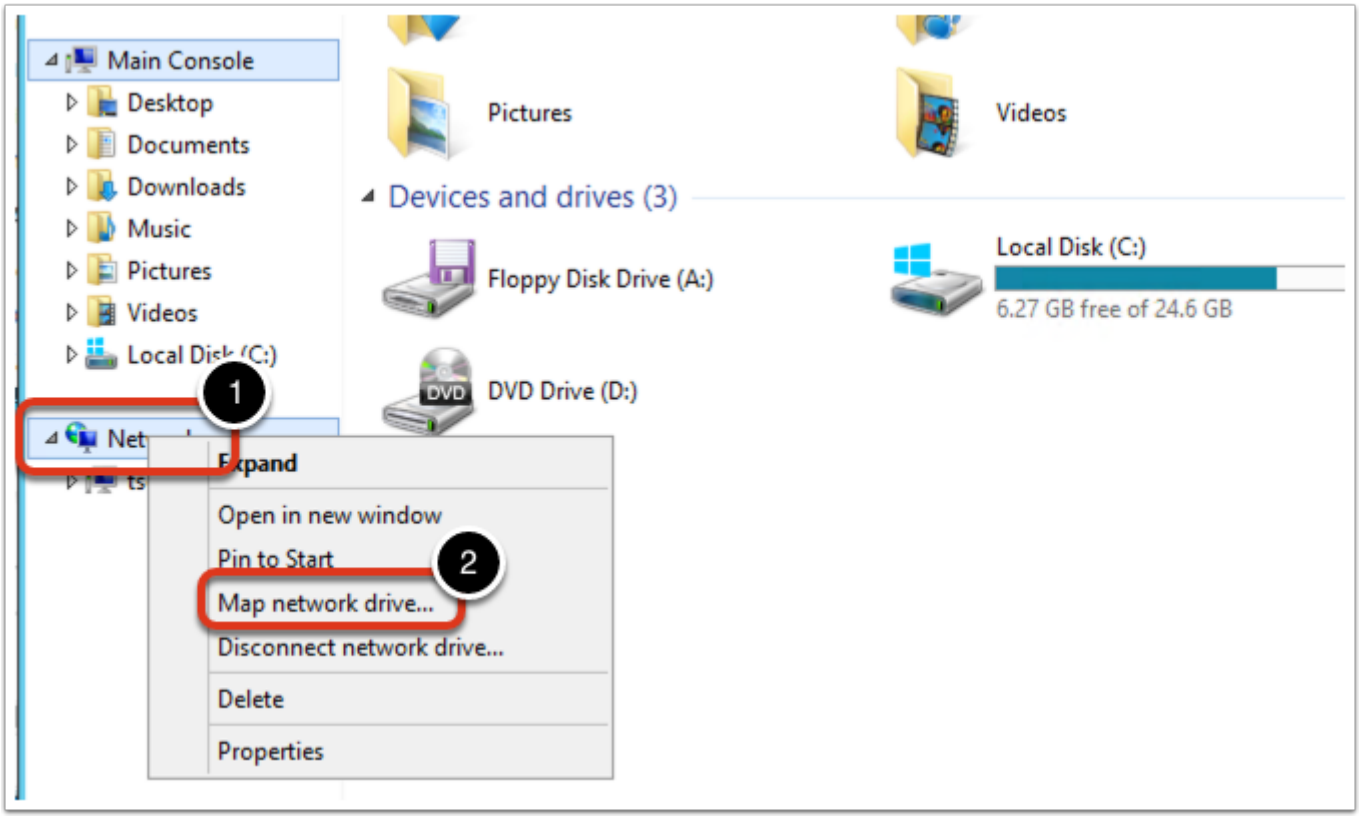
Client File Share Access

vSAN File Service supports SMB export path and shared folders snap-in for Microsoft Management Console (MMC) for managing SMB Shares on the vSAN Cluster.

Client File Share Access

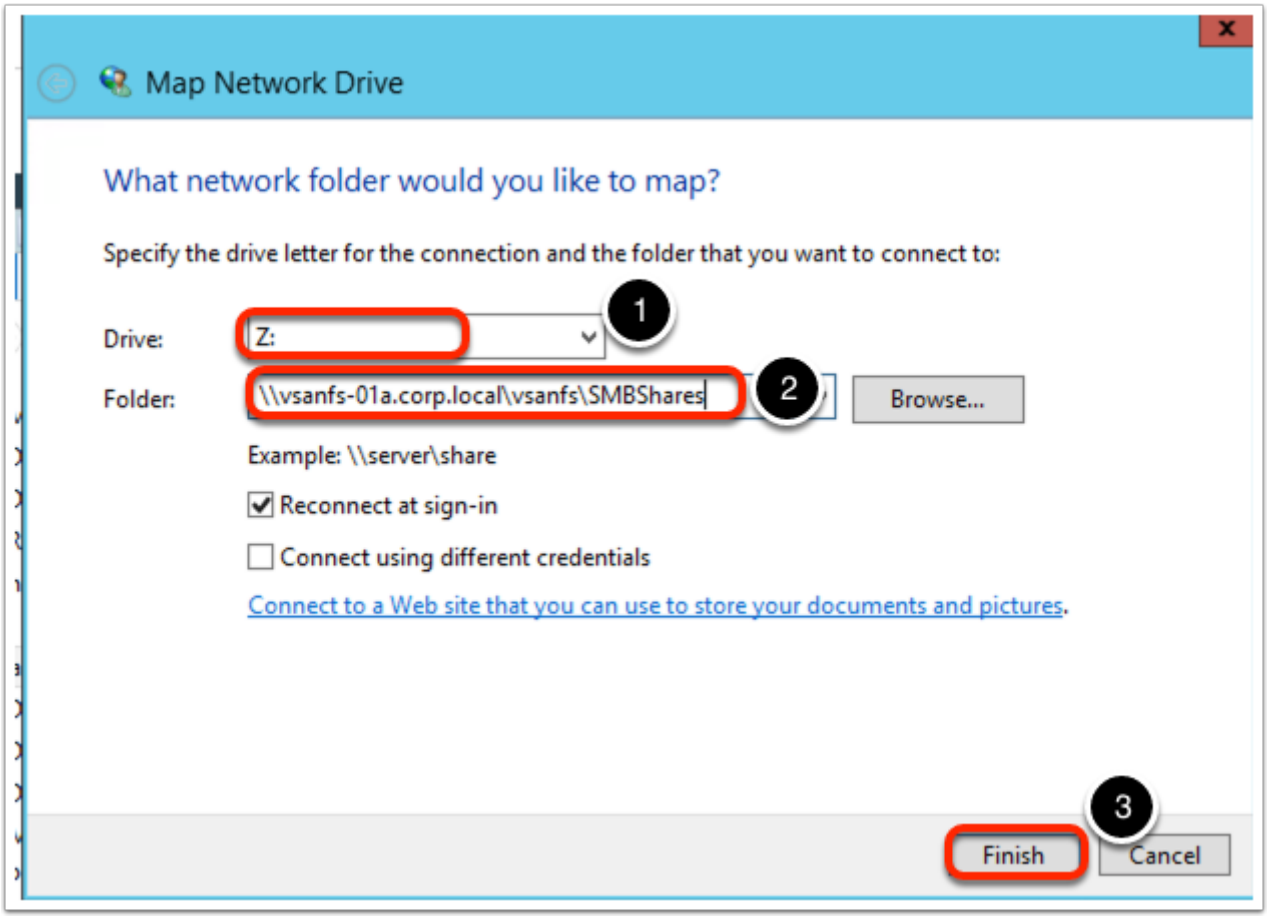
1. Launch **File Explorer** on the taskbar

Client File Shares Access via Network



1. Right click **Network**
2. Select **Map network drive...**

Mapping Network Drive

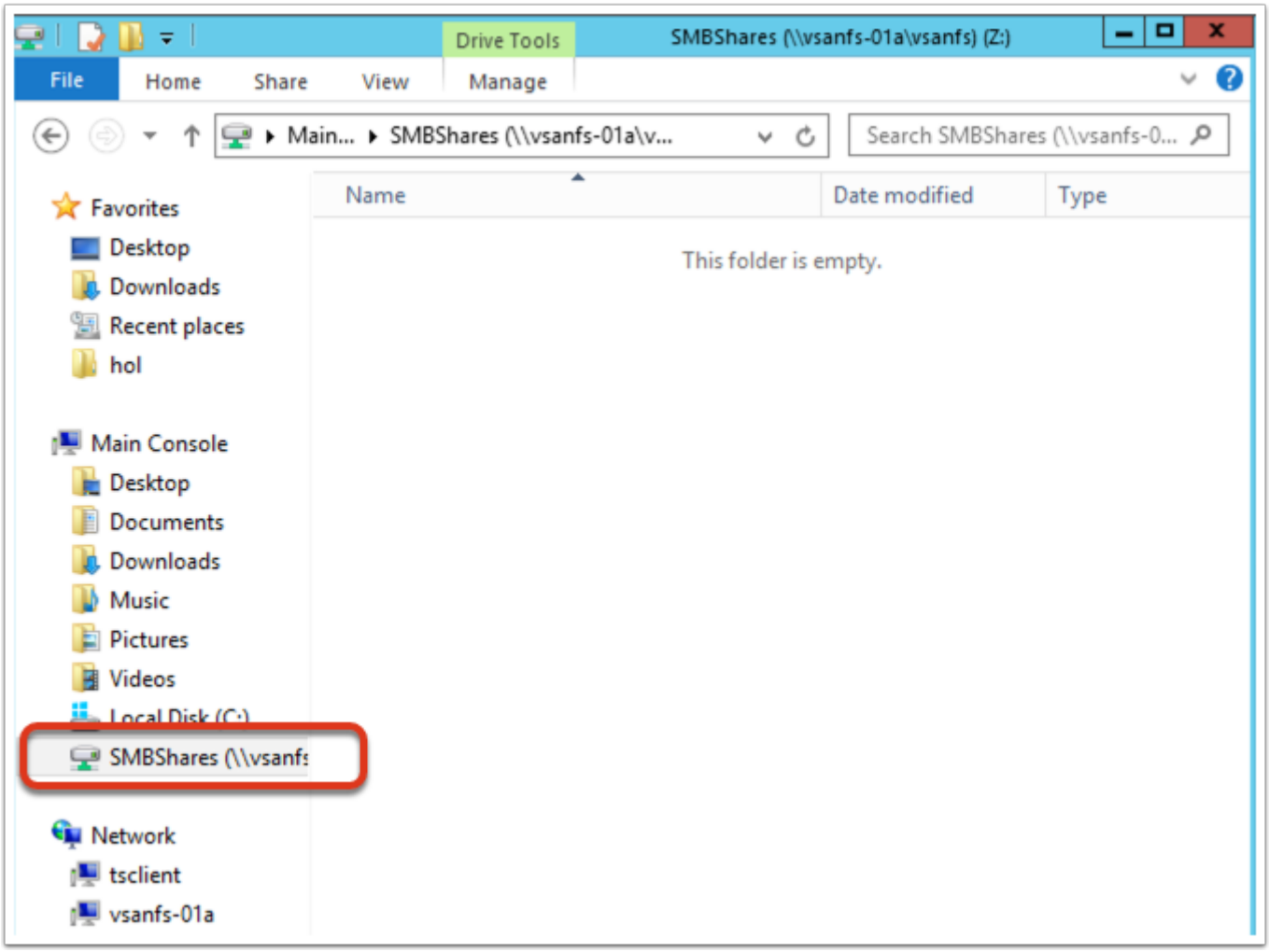


1. Enter the Drive : Z:
2. Enter the folder:

```
\\vsanfs-01a.corp.local\vsanfs\SMBShares
```

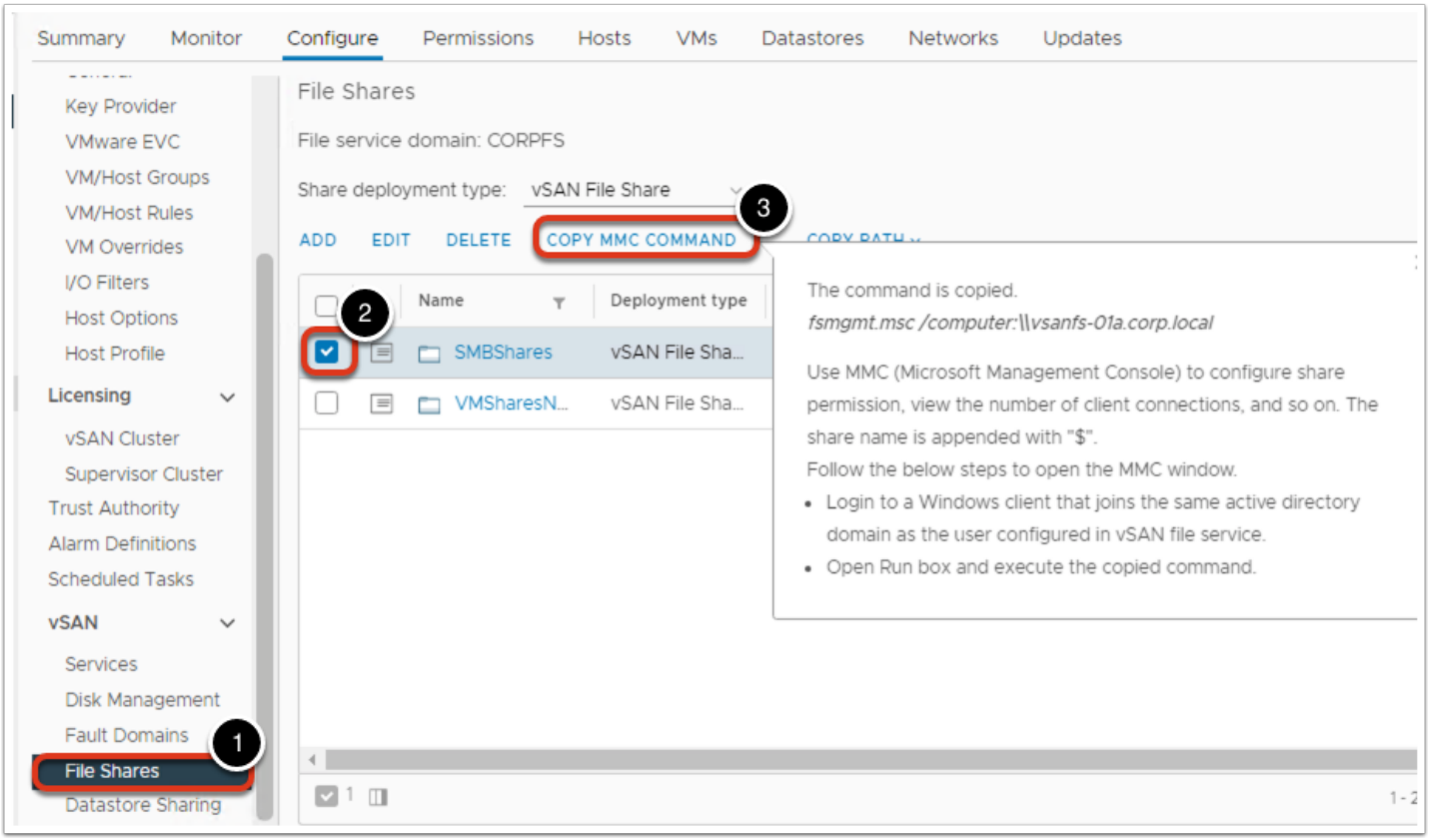
3. Click **Finish**

Mapping Network Drive



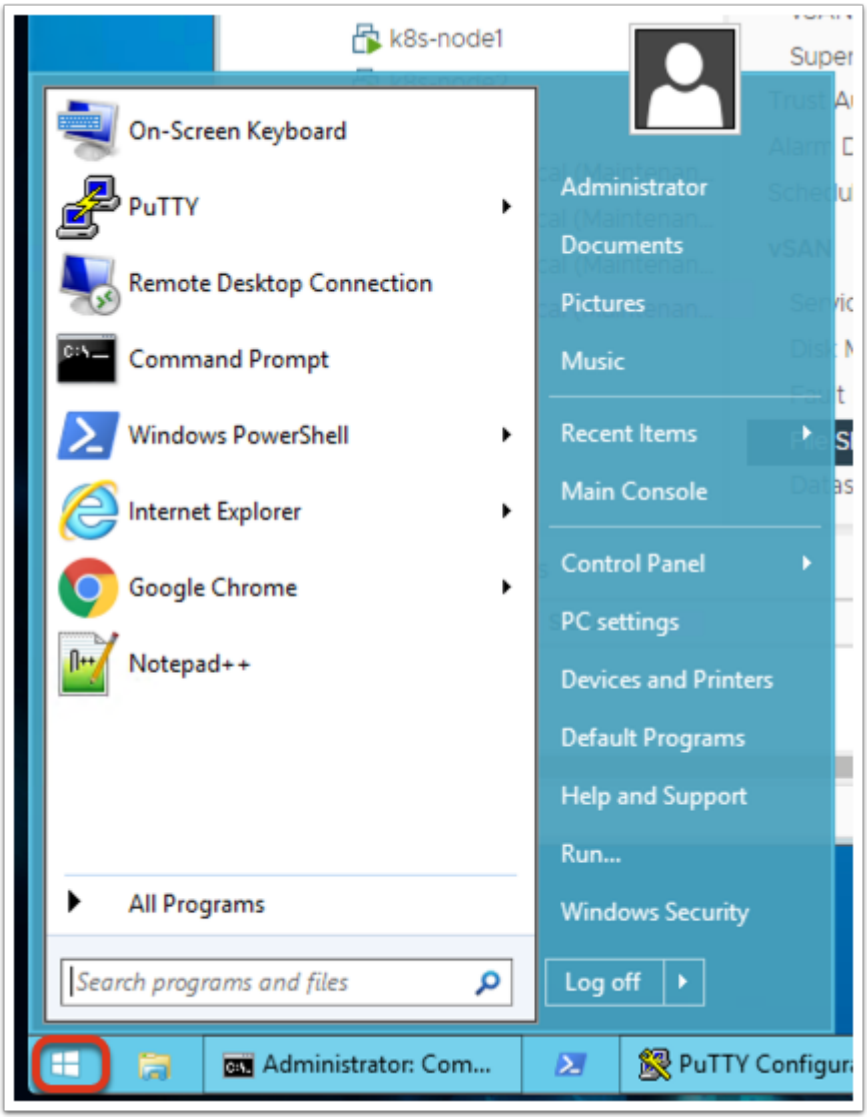
You can see **SMBShares** is mapped from the file explorer.

Create File Shares via Microsoft Management Console

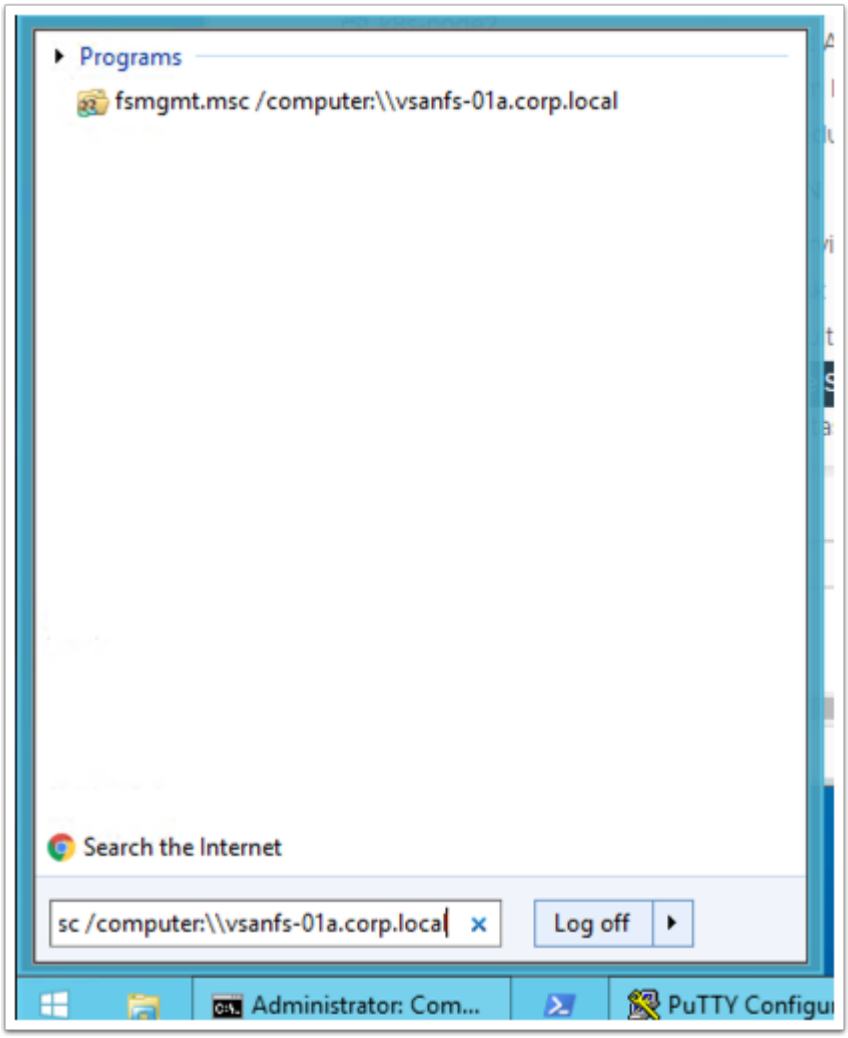


We will find the MMC command so we can use it to launch MMC.

1. Select **vSAN > File Shares**
2. Check mark on **SMBShares**
3. Click on **COPY MMC Command** to view the command string and details

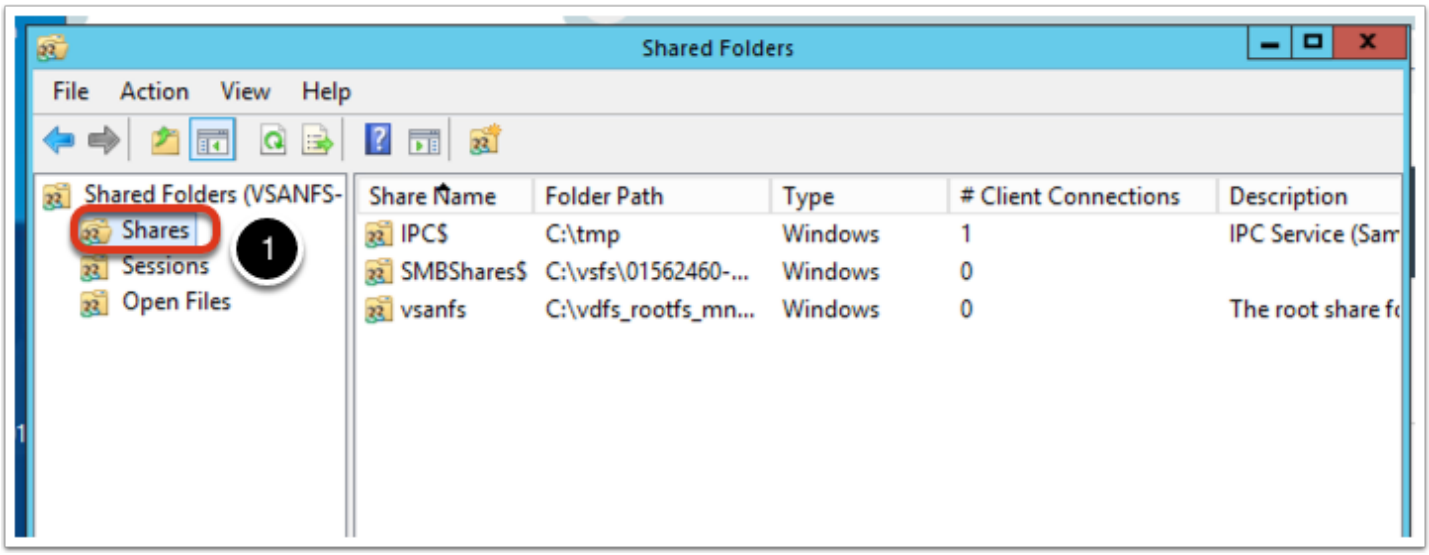


Launch **Start** from the task bar



Enter the command below at the Search bar then hit Return

```
fsmgmt.msc /computer:\\vsanfs-01a.corp.local
```



1. Select **Shares**

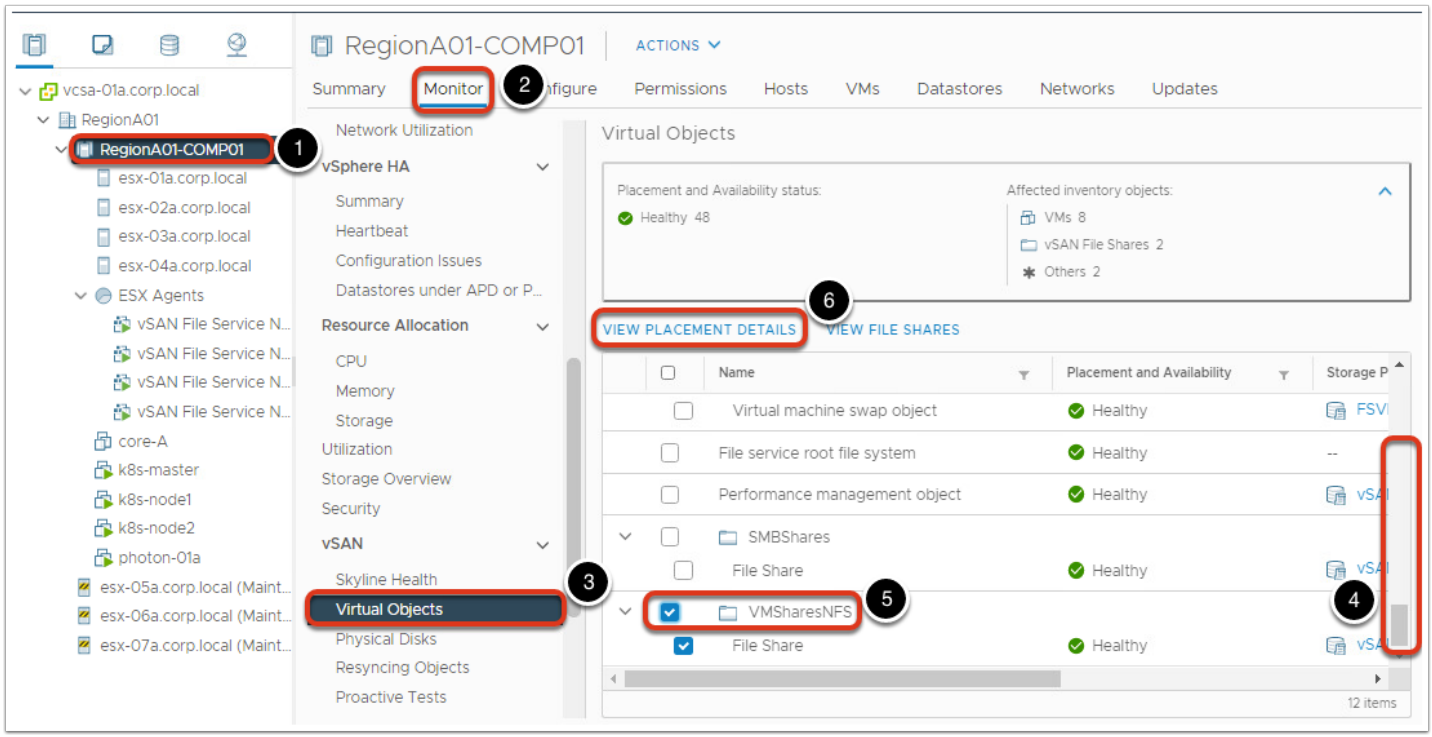
You can see the SMB shares called **SMBShares**

vSAN File Services Monitoring

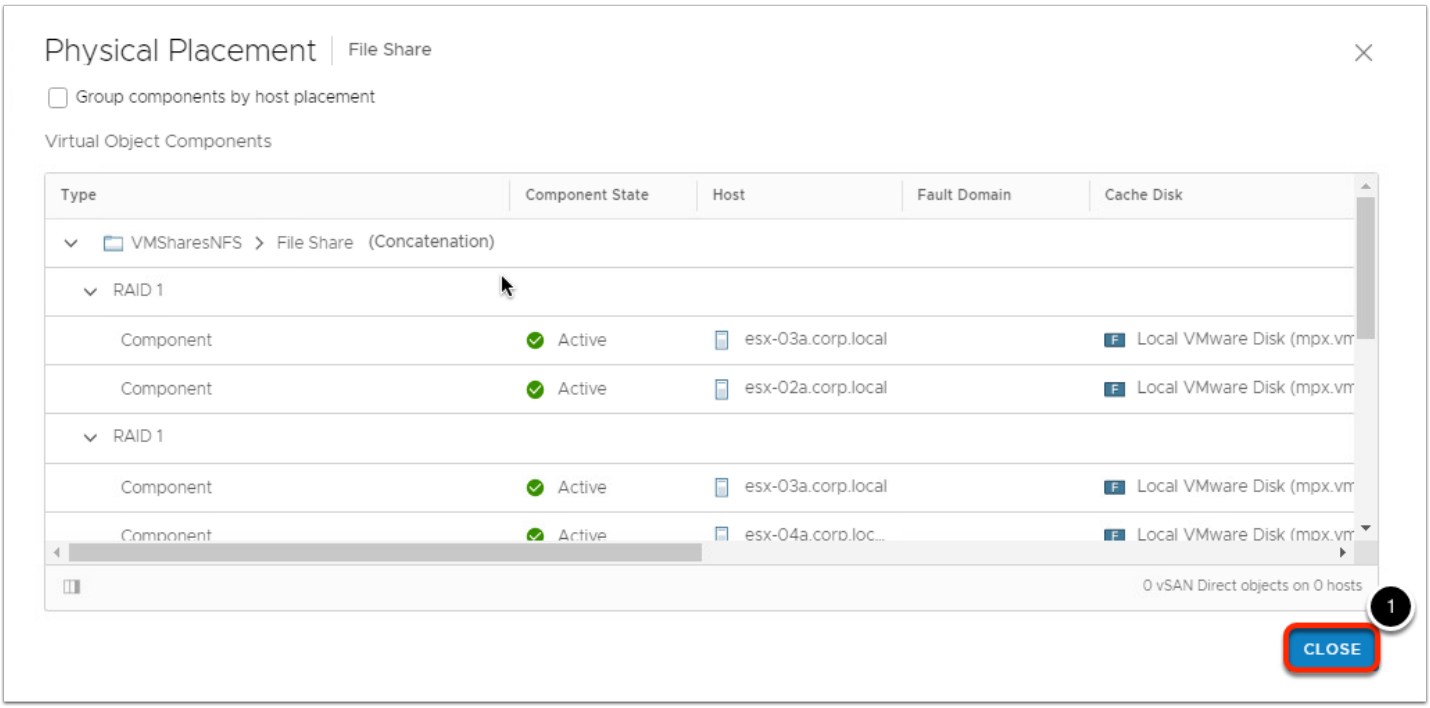
You can monitor the performance and capacity of vSAN file shares.

vSAN Object View and Health Integration

Since the file share is instantiated on vSAN, you can see the file share from the file shares view and see the file share details in the Virtual Object View



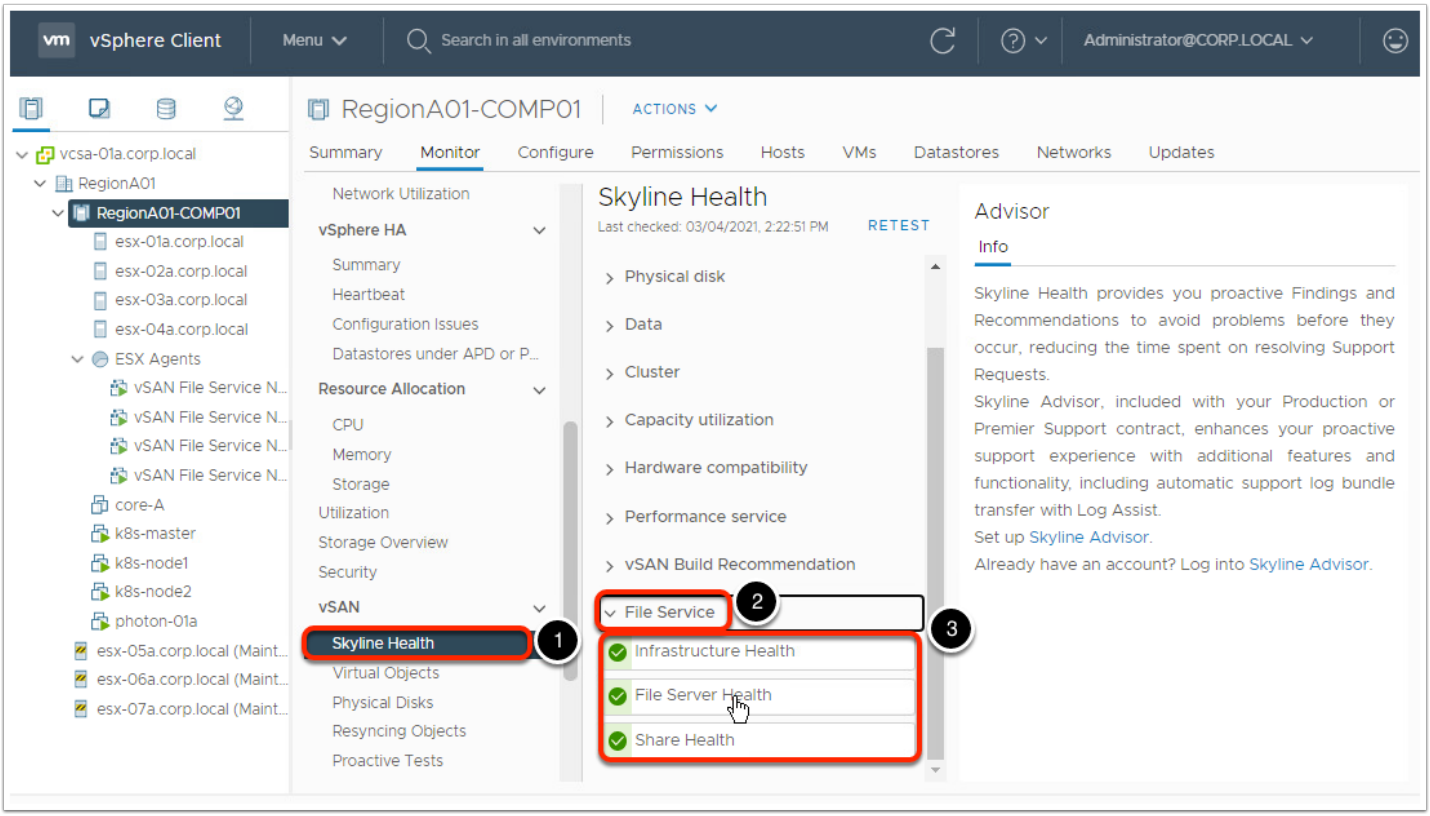
1. Select **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Virtual Objects**
4. Scroll down
5. Check mark **VMSharesNFS**
6. Select **VIEW PLACEMENT DETAILS**



You can see layout of the underlying vSAN object to see which hosts and which physical storage devices are used for placing the components of the file share object.

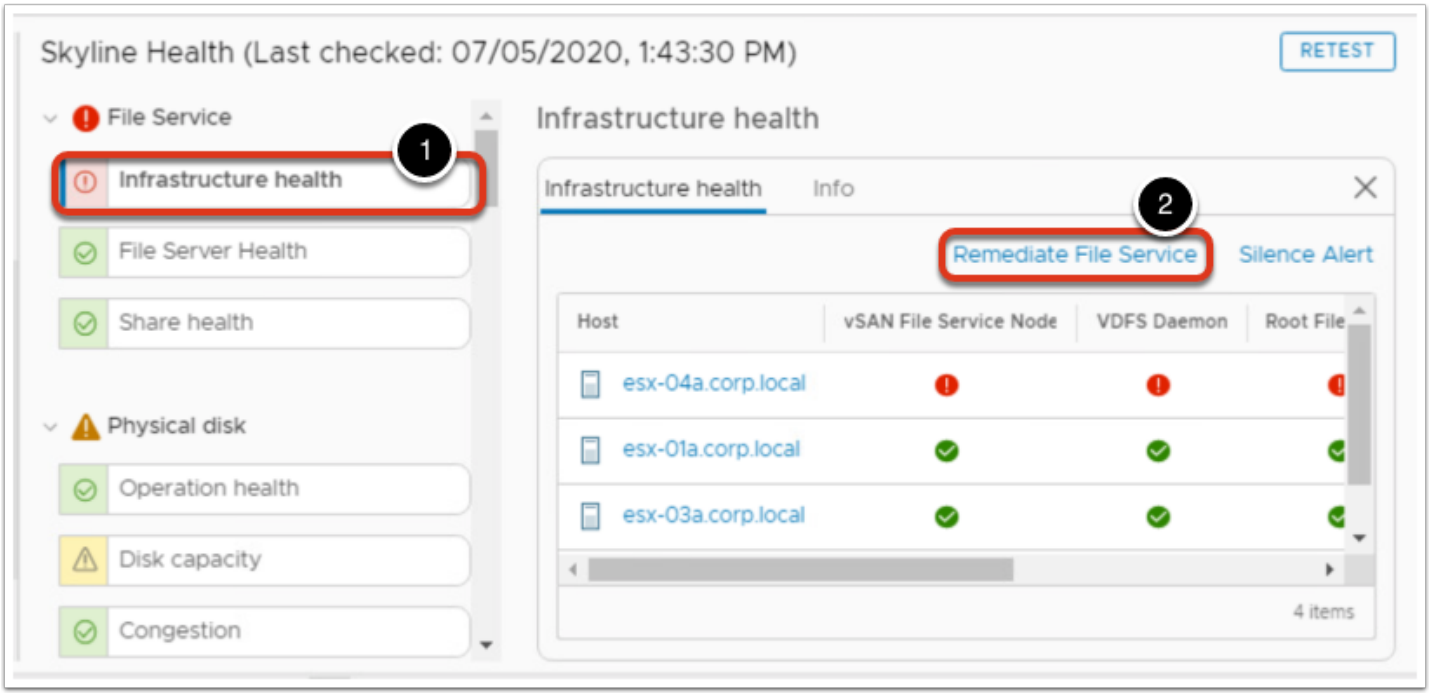
1. Click **CLOSE**

Skyline Health - Infrastructure



1. Select **vSAN > Skyline Health**
2. Expand **File Service** section
3. You can see the File Service Health has been extended to include a number of additional health checks specific to the vSAN File Service.

Skyline Health - Infrastructure



You will notice the File Service is red.

1. Click **Infrastructure health**

ESXi host, esx-04a.corp.local, is not compliant with vSAN File Services. In previous module, we added the 4th host to the vSAN cluster and have not add the proper network configuration for the File Services.

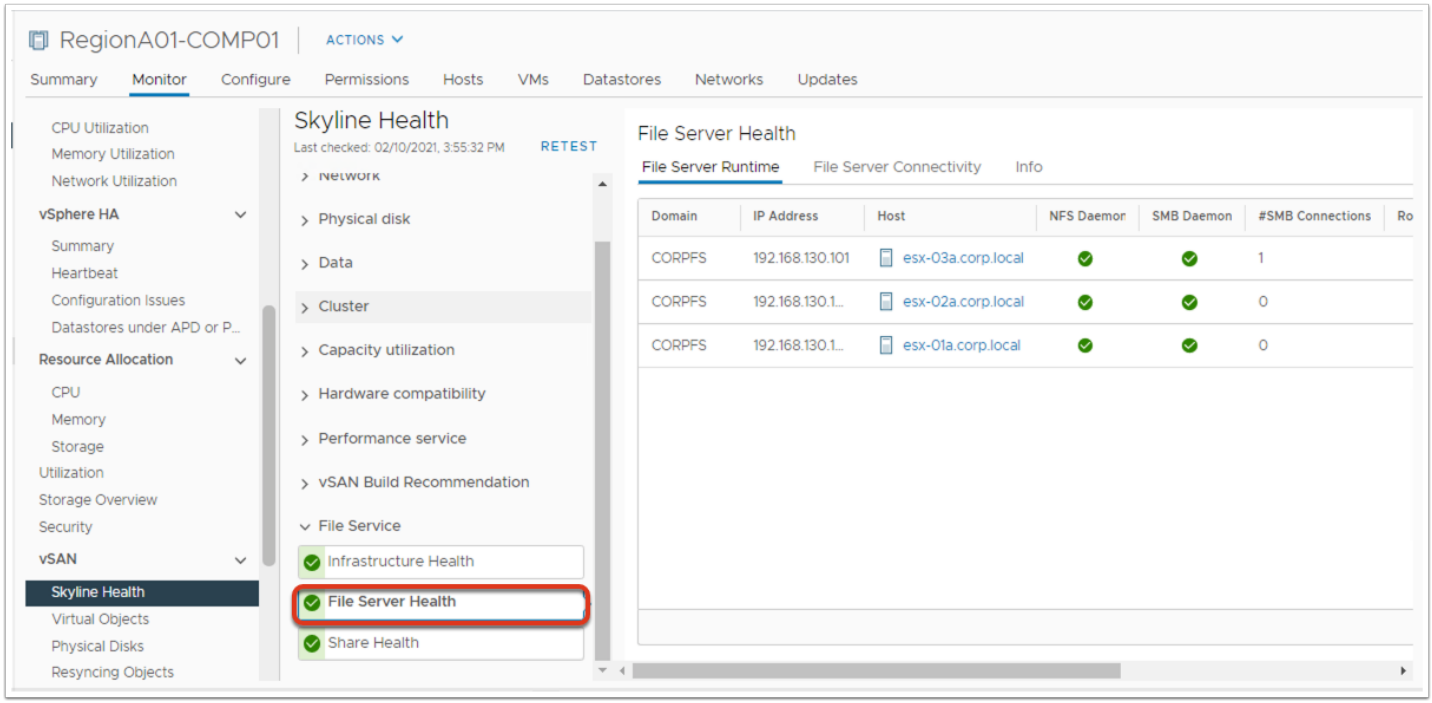
2. Click on **Remediate File Service**

Remediate File Service dialogue box will pop-up.

3. Click **OK** to proceed (not shown)

Everything should be healthy now.

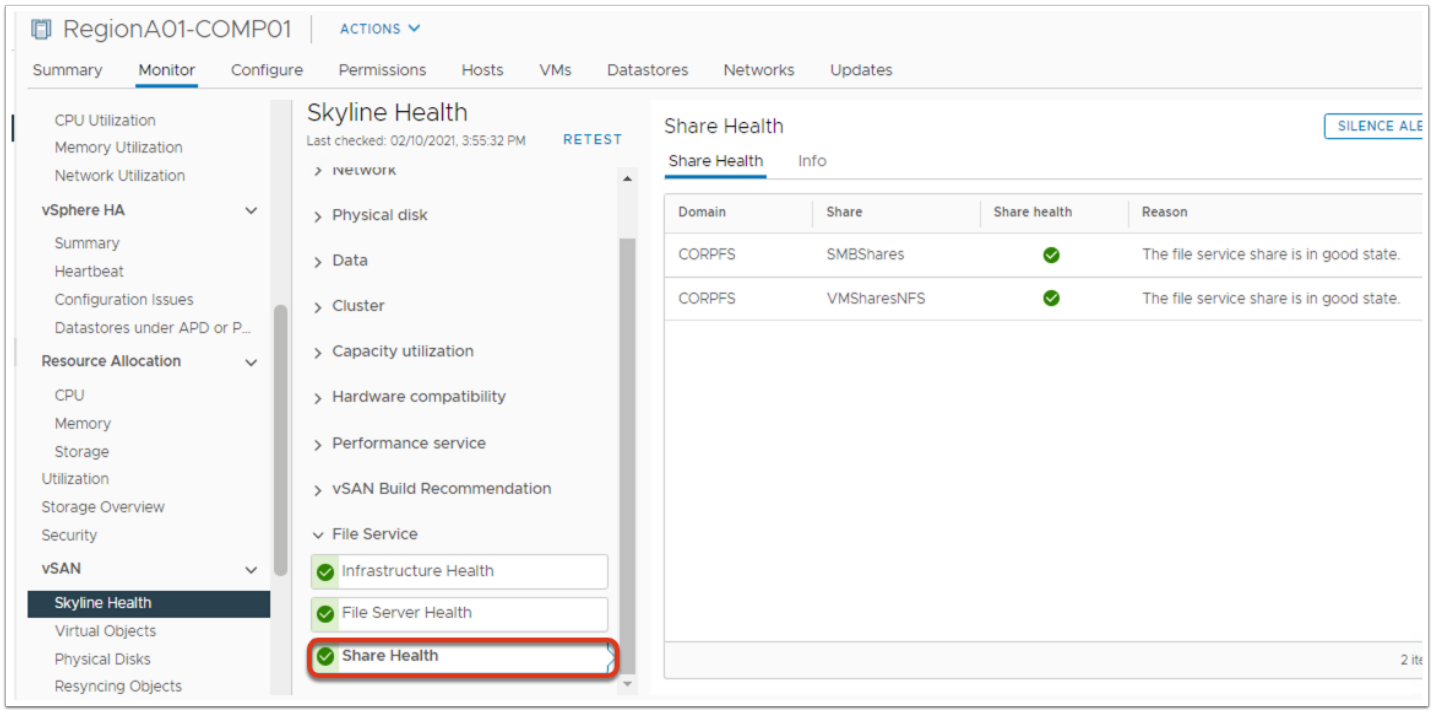
Skyline Health - File Server Health



There is a complete section for File Services.

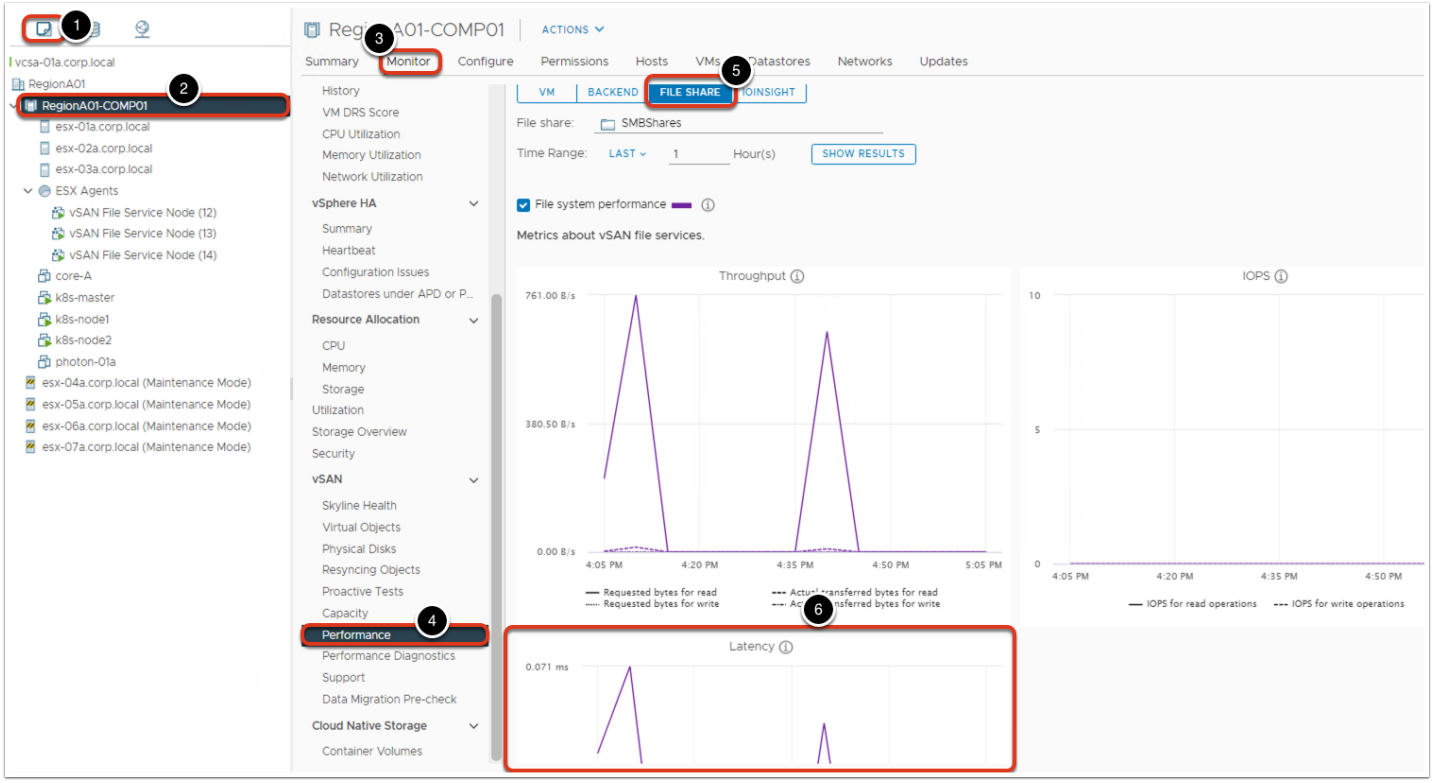
Click on **File Server Health.** You can see the protocol service/file server containers.

Skyline Health - File Server Health



Click on **Share Health.** You can see the file shares we have created.

Monitoring Performance

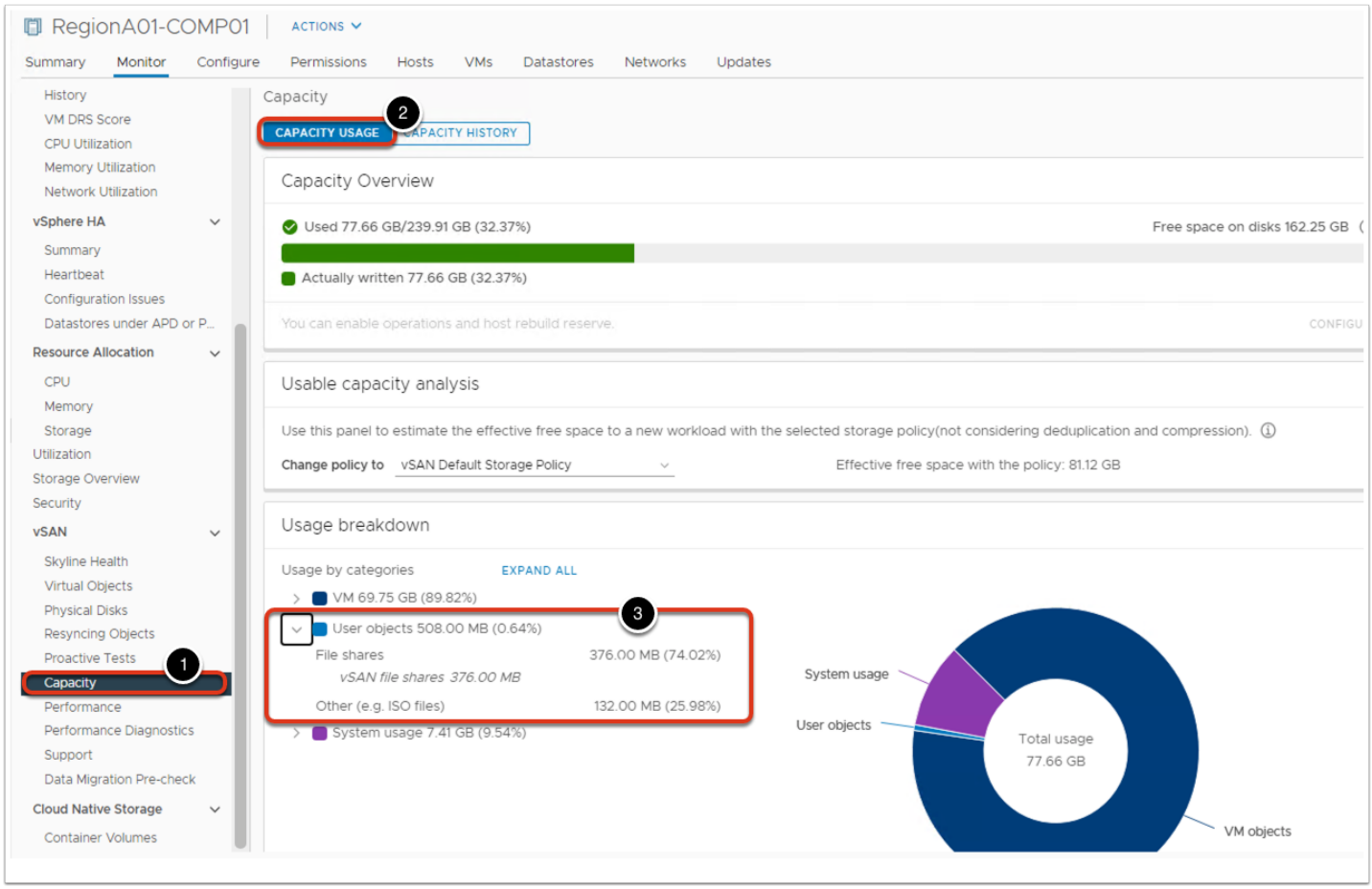


1. Select **Host and Cluster** Icon
2. Select **RegionA01-COMP01**
3. Select **Monitor**
4. Select **vSAN > Performance**
5. Select **FILE SHARE**
6. View **Latency**.

You can see the latency and throughput spikes when we mount the shares and created a file.

Monitor Capacity

You can monitor the capacity for both native file shares and Cloud Native Storage (CNS)-managed file shares.



1. Select **vSAN > Capacity**
2. Select **CAPACITY USAGE**
3. Expand **User Objects**

You can see the FIle Shares usage

Conclusion

In this module, you were provided an overview of vSAN FIle Services and how to mount the shares to another system, access and monitor the vSAN File Shares

You Finished Module 2

Congratulations on completing Module 2.

How to End Lab

To end your lab, click on the **END** button.

Module 3 - vSAN Data Persistence Platform

vSAN Data Persistence Platform

vSphere with Tanzu offers the vSAN Data Persistence platform. The platform provides a framework for modern stateful service partners to interoperate with the underlying VMware infrastructure, so that third-party software can run on vSphere with Tanzu optimally.

The benefits of using vSAN Data Persistence include the following:

Automatic Service Deployment and Scaling

Using the vSphere Client, administrators can install and deploy a modern stateful service on a Supervisor Cluster and grant access to the service namespace to DevOps engineers. The DevOps engineers can provision and scale instances of the stateful service dynamically in a self-service manner through Kubernetes APIs.

Service Monitoring Integrated with vCenter Server

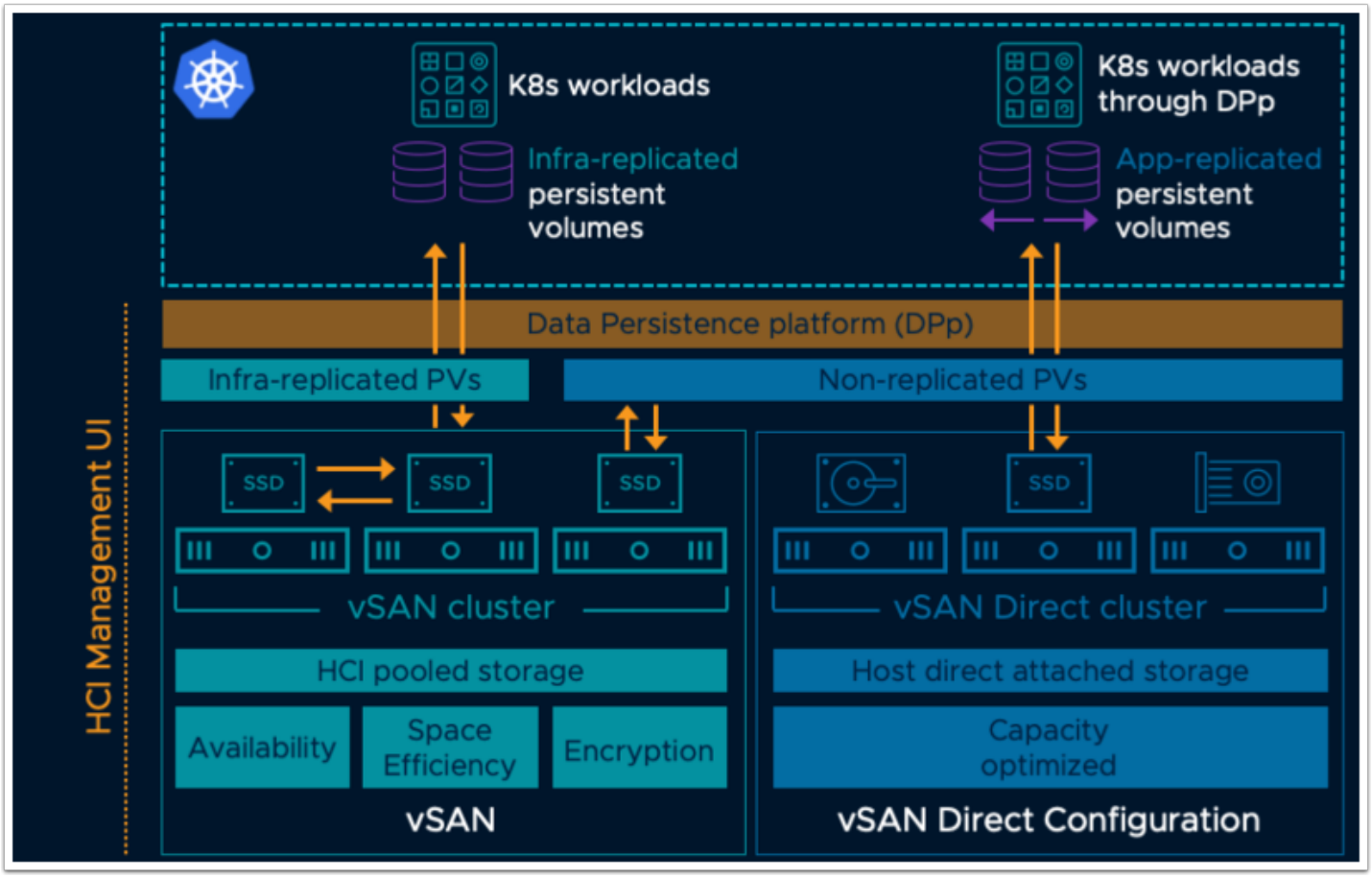
Modern stateful service partners can build dashboard plugins that integrate with vCenter Server. Using the UI plugins, the vSphere administrators can manage and monitor the stateful services. In addition, vSAN offers health and capacity monitoring capabilities for these integrated third-party services.

Optimized Storage Configuration with vSAN Direct

vSAN Direct enables modern stateful services to interface directly with the underlying direct-attached storage for optimized I/O and storage efficiency.

The platform supports the following types of services:

- Object storage, such as MinIO.
- NoSQL databases, also called non-relational databases.
- Traditional databases.



Demo of vSAN Data Persistence Platform (DPp)

Demo of vSAN Data Persistence Platform (DPp) Maintenance and Failures

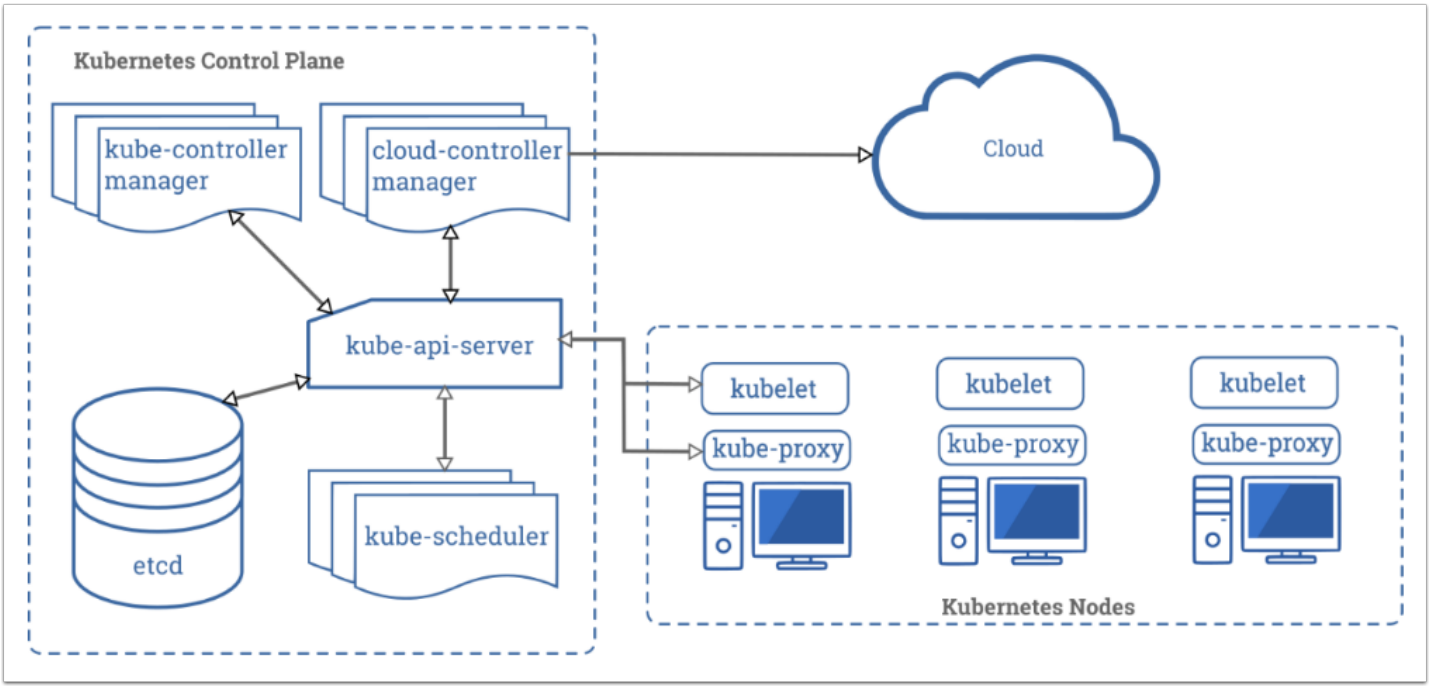
Module 4 - vSAN Cloud Native Storage (CNS)

Kubernetes Overview

Kubernetes is an open-source container orchestration platform that enables the operation of an elastic web server framework for cloud applications. Kubernetes can support data center outsourcing to public cloud service providers or can be used for web hosting at scale. Website and mobile applications with complex custom code can deploy using Kubernetes on commodity hardware to lower the costs on web server provisioning with public cloud hosts and to optimize software development processes.

Kubernetes features the ability to automate web server provisioning according to the level of web traffic in production. Web server hardware can be located in different data centers, on different hardware, or through different hosting providers. Kubernetes scales up web servers according to the demand for the software applications, then degrades web server instances during downtime. Kubernetes also has advanced load balancing capabilities for web traffic routing to web servers in operations.

Kubernetes Components



Here’s the diagram of a Kubernetes cluster with all the components tied together.

Kubernetes Definitions

Kubernetes (often abbreviated to "K8s") is part of the Cloud Native Computing Foundation, which supports the development of shared networking standards in cloud

data center management software. Docker is the most popular container virtualization standard used by Kubernetes. Docker offers integrated software lifecycle development tools for programming teams. RancherOS, CoreOS, and Alpine Linux are popular operating systems specifically designed for container usage. Container virtualization is different than VM or VPS tools using hypervisors and generally requires a smaller operating system footprint in production.

Kubernetes Master

- Legacy term, used as synonym for nodes hosting the control plane
- The term is still being used by some provisioning tools, such as kubeadm, and managed services, to label nodes with kubernetes.io/role and control placement of control plane pods

Kubernetes Nodes

- A node is a worker machine in Kubernetes
- A worker node may be a VM or physical machine, depending on the cluster. It has local daemons or services necessary to run Pods and is managed by the control plane. The daemons on a node include kubelet, kube-proxy and a container runtime implementing the CRI such as Docker

Kubernetes Pods

- The smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster
- A Pod is typically set up to run a single primary container. It can also run optional sidecar containers that add supplementary features like logging. Pods are commonly managed by a Deployment

Kubernetes ReplicaSet

- A ReplicaSet (aims to) maintain a set of replica Pods running at any given time
- Workload objects such as Deployment make use of ReplicaSets to ensure that the configured number of Pods are running in your cluster, based on the spec of that ReplicaSet.

Kubernetes Storage:

Kubernetes Container Storage Interface (CSI)

- The Container Storage Interface (CSI) defines a standard interface to expose storage systems to containers
- CSI allows vendors to create custom storage plugins for Kubernetes without adding them to the Kubernetes repository (out-of-tree plugins). To use a CSI driver from a storage provider, you must first deploy it to your cluster. You will then be able to create a Storage Class that uses that CSI driver.

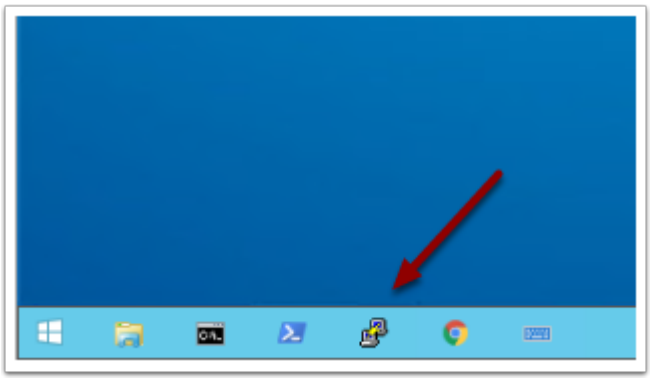
Kubernetes Persistent Volumes

- An API object that represents a piece of storage in the cluster. Available as a general, pluggable resource that persists beyond the lifecycle of any individual Pod
- PersistentVolumes (PVs) provide an API that abstracts details of how storage is provided from how it is consumed. PVs are used directly in scenarios where storage can be created ahead of time (static provisioning). For scenarios that require on-demand storage (dynamic provisioning), PersistentVolumeClaims (PVCs) are used instead

Persistent Volume Claims

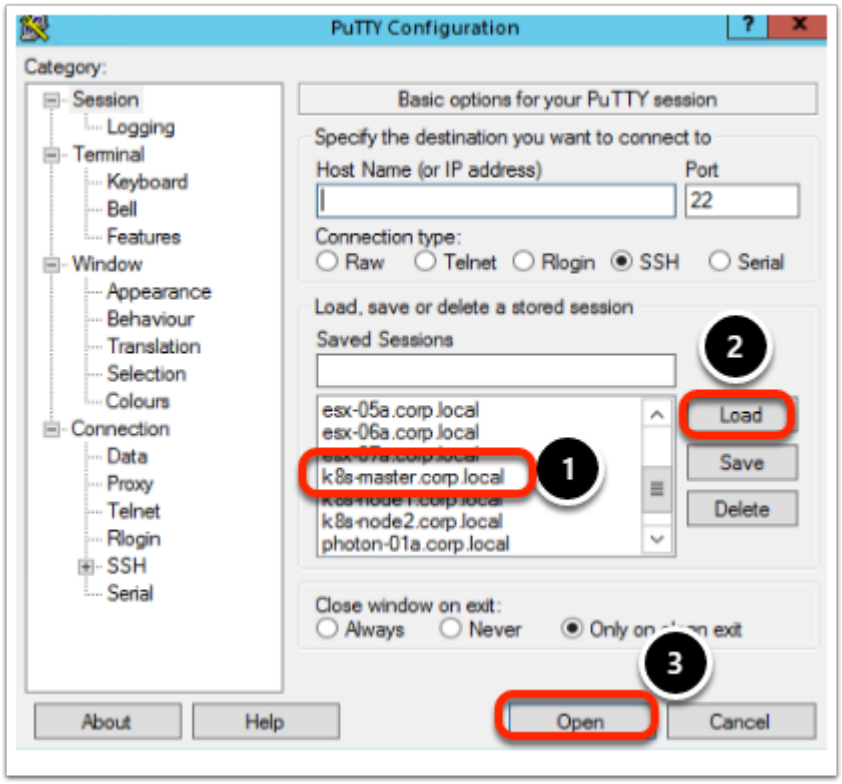
- Claims storage resources defined in a PersistentVolume so that it can be mounted as a volume in a container
- Specifies the amount of storage, how the storage will be accessed (read-only, read-write and/or exclusive) and how it is reclaimed (retained, recycled or deleted). Details of the storage itself are described in the PersistentVolume object.

Launch Putty Application



- From the Main Console launch the PuTTY Application

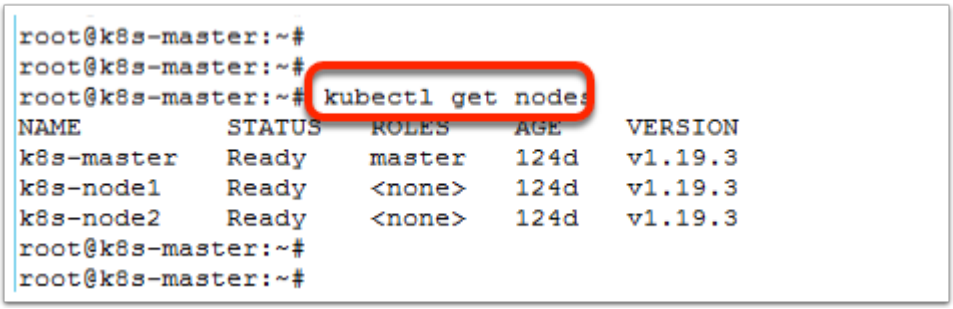
Connect to the Kubernetes Master Node



1. Scroll down the list and select **k8s-master.corp.local**
2. Click **Load**
3. Click **Open**

You will be automatically logged into the Kubernetes Master Node

Verify Kubernetes Master and Nodes are running



```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get nodes
NAME          STATUS   ROLES    AGE    VERSION
k8s-master    Ready    master   124d   v1.19.3
k8s-node1     Ready    <none>   124d   v1.19.3
k8s-node2     Ready    <none>   124d   v1.19.3
root@k8s-master:~#
root@k8s-master:~#
```

- Run the following command :

```
kubectl get nodes
```

Verify that you can see one Kubernetes Master and two Kubernetes Nodes and their status shows are ready

Verify Kubernetes Master and Nodes are running(2)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get nodes -o wide
NAME          STATUS    ROLES     AGE     VERSION    INTERNAL-IP       EXTERNAL-IP   OS-IMAGE
      KERNEL-VERSION       CONTAINER-RUNTIME
k8s-master    Ready     master    124d    v1.19.3    192.168.120.100   <none>        Ubuntu 18.04.4
LTS   4.15.0-101-generic   docker://18.6.0
k8s-node1     Ready     <none>    124d    v1.19.3    192.168.120.101   <none>        Ubuntu 18.04.4
LTS   4.15.0-101-generic   docker://18.6.0
k8s-node2     Ready     <none>    124d    v1.19.3    192.168.120.102   <none>        Ubuntu 18.04.4
LTS   4.15.0-101-generic   docker://18.6.0
root@k8s-master:~#
root@k8s-master:~#
```

- Run the following command :

```
kubectl get nodes -o wide
```

The "-o wide" shows additional information about then Master Nodes such as the Internal and External IP, the OS-Image and the version of docker running

Verify Kubernetes Pods are running

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pods --namespace=kube-system
NAME                                       READY   STATUS    RESTARTS   AGE
coredns-f9fd979d6-78tnm                    1/1     Running   7          124d
coredns-f9fd979d6-lqxfn                    1/1     Running   7          124d
etcd-k8s-master                            1/1     Running   8          124d
kube-apiserver-k8s-master                  1/1     Running   8          124d
kube-controller-manager-k8s-master         1/1     Running   23         124d
kube-flannel-ds-h2gn8                      1/1     Running   7          124d
kube-flannel-ds-z4fzh                      1/1     Running   7          124d
kube-flannel-ds-zlcrx                      1/1     Running   8          124d
kube-proxy-4cb87                           1/1     Running   7          124d
kube-proxy-5cr6x                           1/1     Running   7          124d
kube-proxy-zggpn                           1/1     Running   7          124d
kube-scheduler-k8s-master                  1/1     Running   23         124d
vsphere-cpi-wjz95                          1/1     Running   2          31d
vsphere-csi-controller-75455477dc-lm2r5    6/6     Running   10         31d
vsphere-csi-node-ljq8q                     3/3     Running   21         124d
vsphere-csi-node-np49k                     3/3     Running   21         124d
vsphere-csi-node-th7gb                     3/3     Running   21         124d
root@k8s-master:~#
root@k8s-master:~#
```

- Run the following command :

```
kubectl get pods --namespace=kube-system
```

This command will the Pods that belong in the kube-system namespace.

These are the Pods that make up your Kubernetes environment. There are additional Pods running here as well. We are using Flannel for the Pod Networking. You will also see the vsphere-csi-contoller and vsphere-csi-node which integrate with Storage Policy Based Management and allow us to consume vSphere Storage for Kubernetes such as persistent volumes that inherit the capabilities of the underlying infrastructure such as RAID levels, encryption and deduplication and compression.

Create a Block Persistent Volume

We are positioning vSAN as a platform for both traditional virtual machine workloads and newer containerized workloads.

The vSphere Container Storage Interface (CSI) driver only provisions **block** based Persistent Volumes which were **Read-Write-Once**, meaning that only one Pod could consume the volume at a time.

In this module we will look at how to deploy an application that uses Block Persistent Volumes. We will show you some of the features that we have build into the vSphere 7 UI for managing and monitoring containers.

In this module we will demonstrate:

1. Create a StorageClass which is using the vSAN Default Storage Policy, showing that the Persistent Volume will be created on vSAN Datastore
2. Create a READWRITEONCE Persistent Volume Claim (PVC) to manually create a Persistent Volume (PV)
3. Create a Pod that uses that PVC which gets the Persistent Volume associated with the Persistent Volume Claim
4. Demonstrate the new features in the vSphere 7 UI for managing and monitoring Persistent Volumes

Kubernetes uses configuration files written in YAML (Yet another Markup Language) to specify configuration-type information by using a combination of Maps of name-value pairs of lists of items.

We have pre-created all of these configuration files for you in these labs.

Create a Storage Claim (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat block-sc.yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: block-sc
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.vsphere.vmware.com
parameters:
  storagepolicyname: "vSAN Default Storage Policy"
root@k8s-master:~#
root@k8s-master:~#
```

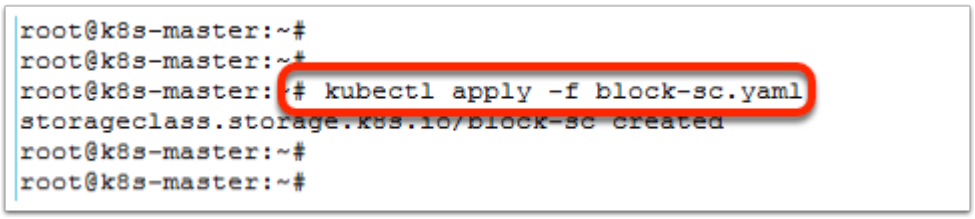
First, we need to create a Storage Claim.

- Run the following command to display the YAML manifest that we will use to create the Storage Claim:

```
cat block-sc.yaml
```

1. The **kind** field describes what each object is. The first manifest is the StorageClass, which selects a vSphere datastore to place any provisioned Persistent Volumes
2. The **provisioner** field is a reference to the VMware CSI driver
3. The **storagepolicyname** field refers to an SPBM policy is vSphere. The claim will select the vSAN datastore

Create a Storage Claim (2)

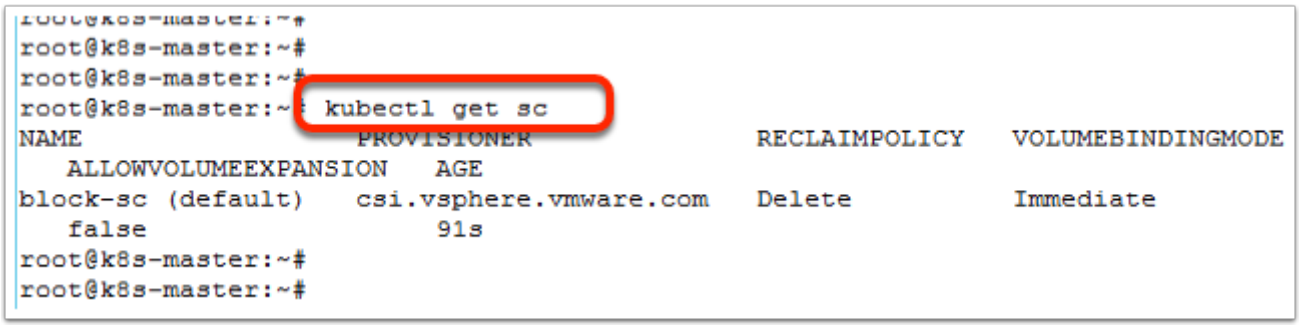


- To create the Storage Claim, run the following command:

```
kubectl apply -f block-sc.yaml
```

The storage claim will be created

Verify Storage Claim was created



- To verify the storage claim was created, run the following command :

```
kubectl get sc
```

Create Persistent Volume Claim (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat block-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: block-pvc
  labels:
    type: Block
    size: 2Gi
spec:
  storageClassName: block-sc
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
root@k8s-master:~#
root@k8s-master:~#
```

First, we need to create a Persistent Volume Claim.

- Run the following command to display the YAML manifest that we will use to create the Storage Claim:

```
cat block-pvc.yaml
```

This is the Persistent Volume Claim manifest.

It results in the creation of a 2Gi Persistent Volume (VMDK) on vSphere storage reference by the **storageClassName**.

Since this storageClassName refers to the StorageClass above, this PV will be created on the vSAN datastore.

Create Persistent Volume Claim (2)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl apply -f block-pvc.yaml
persistentvolumeclaim/block-pvc created
root@k8s-master:~#
root@k8s-master:~#
```

- To create the Storage Claim, run the following command:

```
kubectl apply -f block-pvc.yaml
```

Verify Persistent Volume Claim was created

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pvc
NAME        STATUS   VOLUME                                     CAPACITY
   ACCESS MODES   STORAGECLASS   AGE
block-pvc   Bound    pvc-d0320d24-6c12-4bb3-b964-1e678d7cff86   2Gi
   RWO            block-sc       2m58s
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the Persistent Volume Claim was created, run the following command :

```
kubectl get pvc
```

If the STATUS reports Pending, run the same command again.

Verify Persistent Volume was created

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY
   STATUS   CLAIM                  STORAGECLASS   REASON   AGE
pvc-d0320d24-6c12-4bb3-b964-1e678d7cff86   2Gi        RWO            Delete
   Bound    default/block-pvc   block-sc                  4m13s
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the Persistent Volume was created, run the following command :

```
kubectl get pv
```

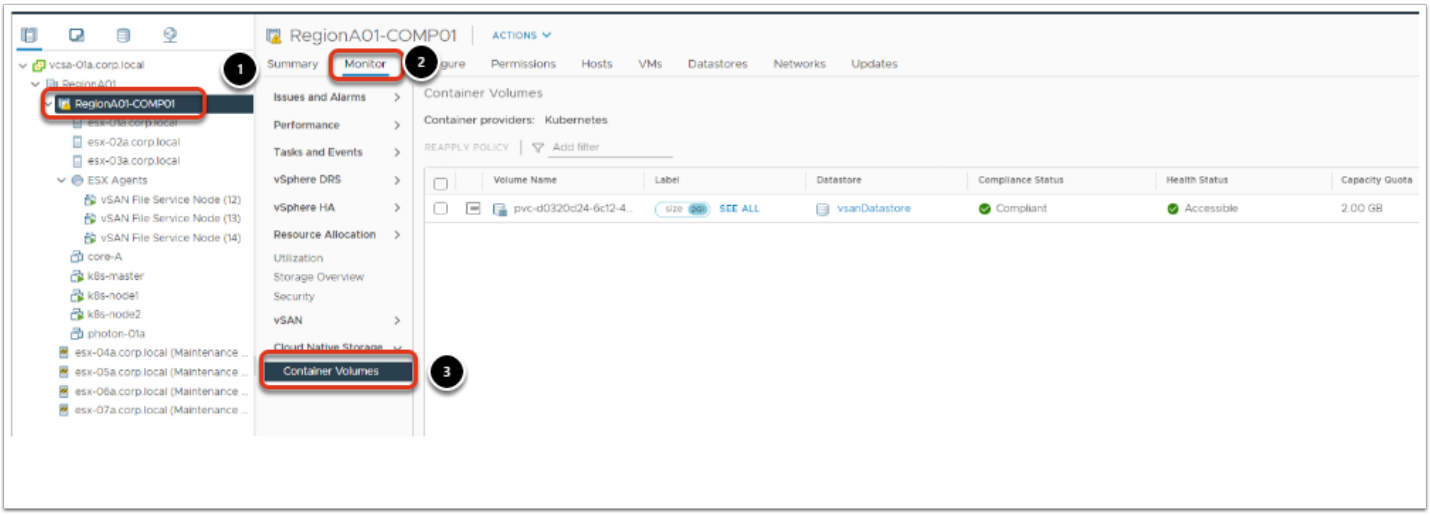
Persistent Volume created in vSphere (1)

Recent Tasks Alarms

Task Name	Target	Status	Details	Initiator	Queued For	Start Time
Update container volume	vcse-01a.corp.local	Completed		com.vmware.cns	12 ms	06/09/2020, 11:11:26 AM
Update container volume	vcse-01a.corp.local	Completed		com.vmware.cns	18 ms	06/09/2020, 11:11:21 AM
Create a virtual disk object	esx-03a.corp.local	Completed		VirtualCenter	7 ms	06/09/2020, 11:10:24 AM
Create container volume	vcse-01a.corp.local	Completed		com.vmware.cns	339 ms	06/09/2020, 11:10:17 AM

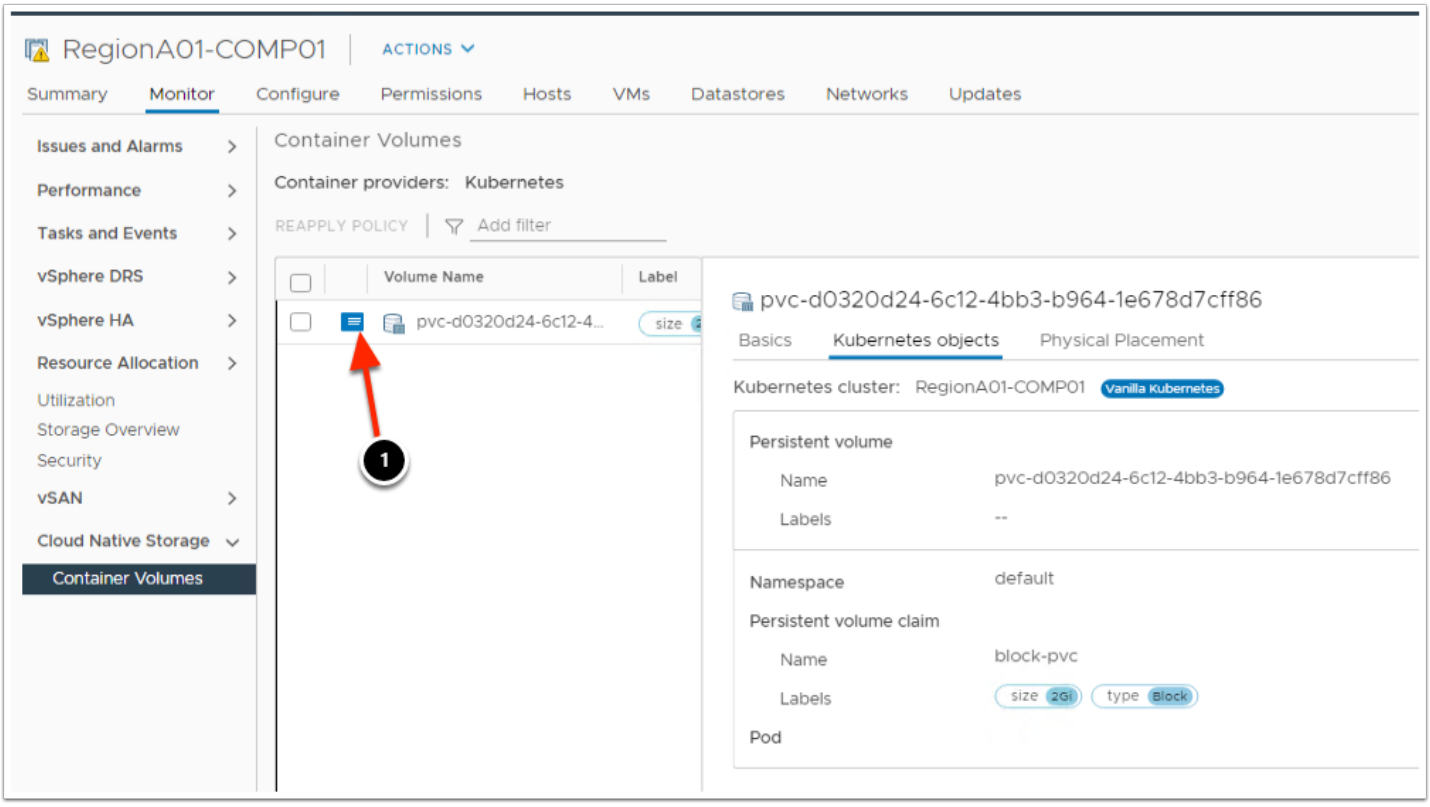
- Review the Recent Tasks in vSphere UI

Persistent Volume created in vSphere (2)



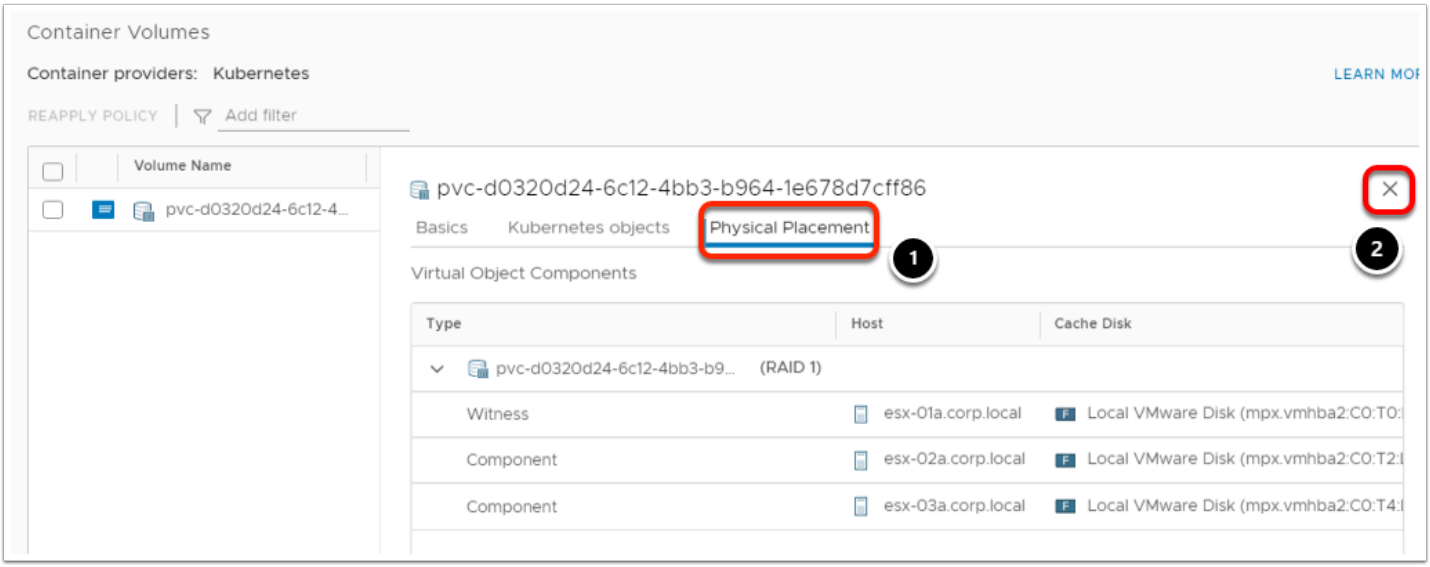
1. Select **RegionA01-COMP01**
2. Select **Monitor**
3. Select **Cloud Native Storage > Container Volumes**

Persistent Volume created in vSphere (2)



1. Click the **Details** to get more information about the Persistent Volume - here we can see some details on the virtual objects, like the Persistent Volume Name, Persistent Volume Claim and any Label that have been applied to these.

View Placement Details (2)



We can see that it has been instantiated as a RAID-1 (Mirror) on the vSAN Datastore.

1. Click Physical Placement, here we can see the Components and Witness spread across the ESxi hosts
2. Click X to Close the window

Create a POD to mount the Volume (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat block-pod-a.yaml
apiVersion: v1
kind: Pod
metadata:
  name: block-pod-a
spec:
  containers:
  - name: block-pod-a
    image: "k8s.gcr.io/busybox"
    volumeMounts:
    - name: block-vol
      mountPath: "/mnt/volume1"
    command: [ "sleep", "1000000" ]
  volumes:
    - name: block-vol
      persistentVolumeClaim:
        claimName: block-pvc
root@k8s-master:~#
root@k8s-master:~#
```

Finally, we need to create a Pod to mount the Volume.

- Return to Putty and run the following command to display the YAML manifest that we will use to create the POD:

```
cat block-pod-a.yaml
```

Create a POD to mount the Volume (2)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl apply -f block-pod-a.yaml
pod/block-pod-a created
root@k8s-master:~#
root@k8s-master:~#
```

- Run the following command:

```
kubectl apply -f block-pod-a.yaml
```

Verify the POD is running

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
block-pod-a   1/1     Running   0          51s
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the Persistent Volume Claim was created, run the following command :

```
kubectl get pod
```

The Pod is ready and the Status is Running

Examine POD Events

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get event --field-selector involvedObject.name=block-pod-a
LAST SEEN   TYPE      REASON                  OBJECT              MESSAGE
2m40s       Normal    Scheduled               pod/block-pod-a     Successfully assigned default/block-po
d-a to k8s-node2
2m33s       Normal    SuccessfulAttachVolume  pod/block-pod-a     AttachVolume.Attach succeeded for volu
me "pvc-d0320d24-6c12-4bb3-b964-1e[illegible]8d7cff86"
2m20s       Normal    Pulling                 pod/block-pod-a     Pulling image "k8s.gcr.io/busybox"
2m18s       Normal    Pulled                  pod/block-pod-a     Successfully pulled image "k8s.gcr.io/
busybox" in 2.160314353s
2m18s       Normal    Created                 pod/block-pod-a     Created container block-pod-a
2m17s       Normal    Started                 pod/block-pod-a     Started container block-pod-a
root@k8s-master:~#
root@k8s-master:~#
```

- We can use the following command to examine the events associated with the creation of the Pod.

```
kubectl get event --field-selector involvedObject.name=block-pod-a
```

1. We can see that it Successfully Attached the Volume to the Pod

Check Volume was mounted in the POD

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl exec -it block-pod-a -- /bin/sh
/ #
/ #
/ # mount | grep /mnt/volume1
/dev/sdb on /mnt/volume1 type ext4 (rw,relatime,data=ordered)
/ #
/ #
/ #
/ # df /mnt/volume1
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sdb               1998672      6144   1871288   0% /mnt/volume1
/ #
/ #
```

1. First lets login to the POD by running the following command:

```
kubectl exec -it block-pod-a -- /bin/sh
```

2. Check is the volume mounted by running the following command:

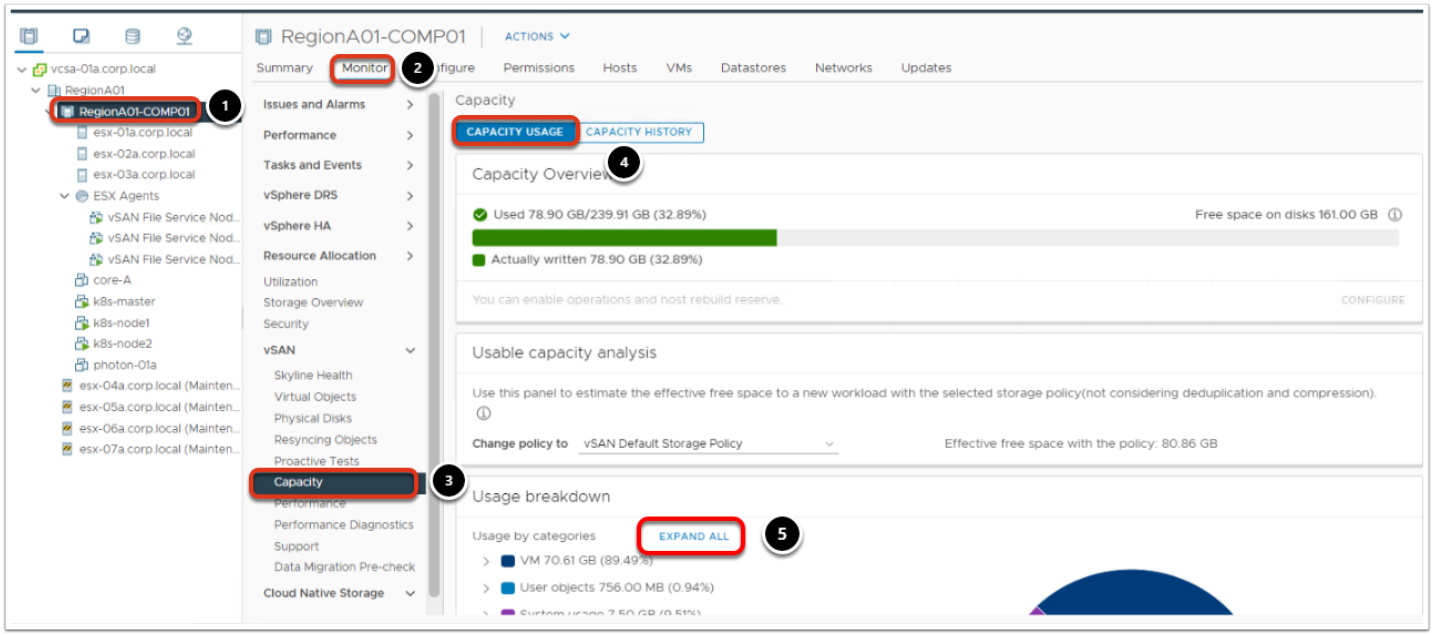
```
mount | grep /mnt/volume1
```

3. Finally, lets check the size of the Volume

```
df /mnt/volume1
```

We can see the 2 GB Volume is mounted in the POD

vSAN Capacity View (1)



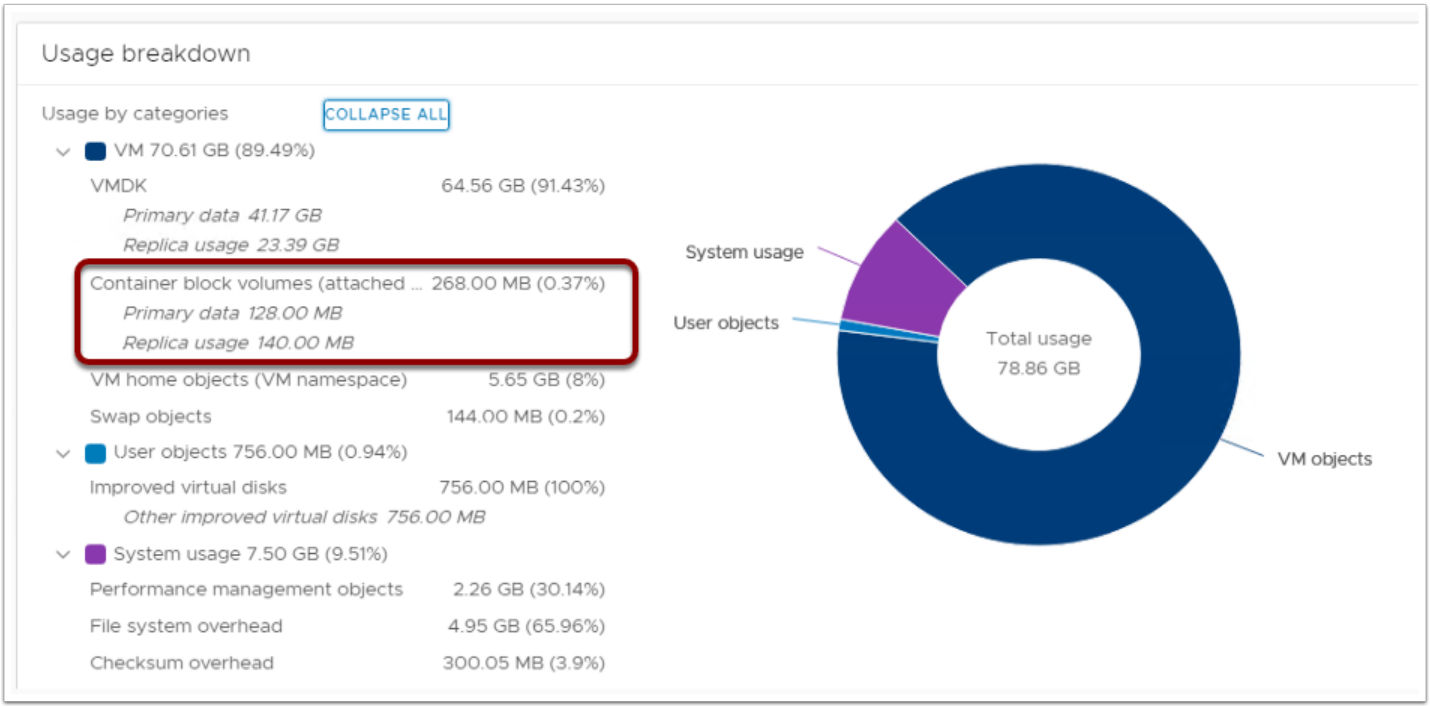
Let now have a look at the capacity consumed by the Persistent Volumes

1. Select the Cluster called **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Capacity**
4. Select **CAPACITY USAGE**

What we are interested is the **Usage breakdown before dedupe and compression**, so scroll down the screen

5. Click the EXPAND ALL link

vSAN Capacity View (2)



And now that the volume has been attached to a VM (the K8s worker node where the Pod was scheduled), the Usage breakdown now changes how it is reported. It now appears in the VM category, under **Container block volumes (attached to a VM).** So it is not longer *stranded*.

Delete Pod

Return back to the K8S-Master Putty Session

1. Enter the following command to exit the shell of the pod

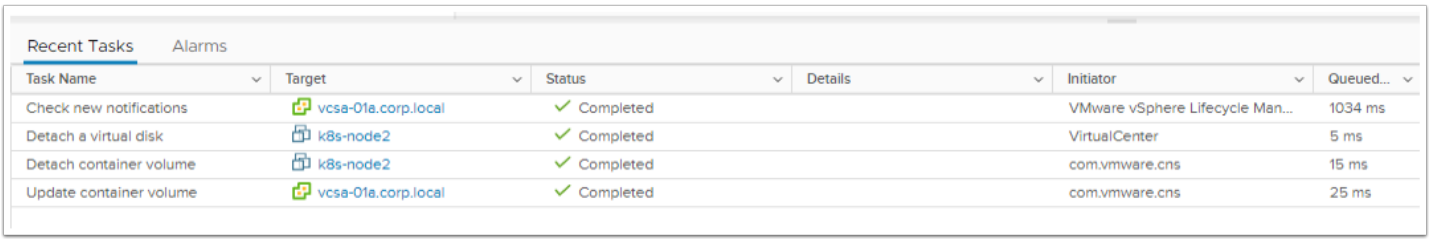
```
exit
```

2. Enter the following command to delete the pod

```
kubectl delete -f block-pod-a.yaml
```

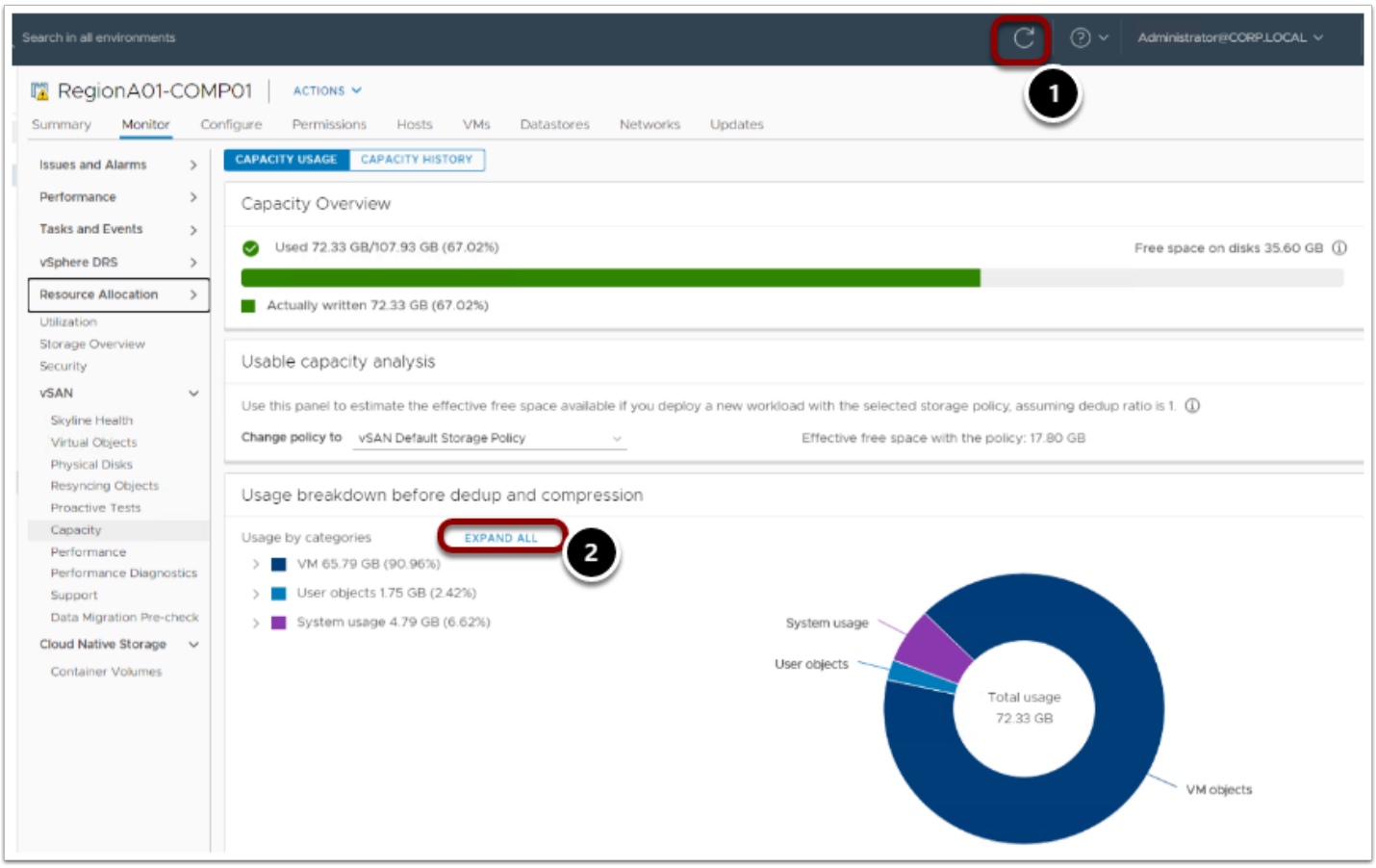
Verify that the pod was deleted. It may take a few seconds for the pod to be deleted.

Delete Pod



- Check in the vSphere Client and verify in the Recent Tasks that the volume has been detached from the Pod.

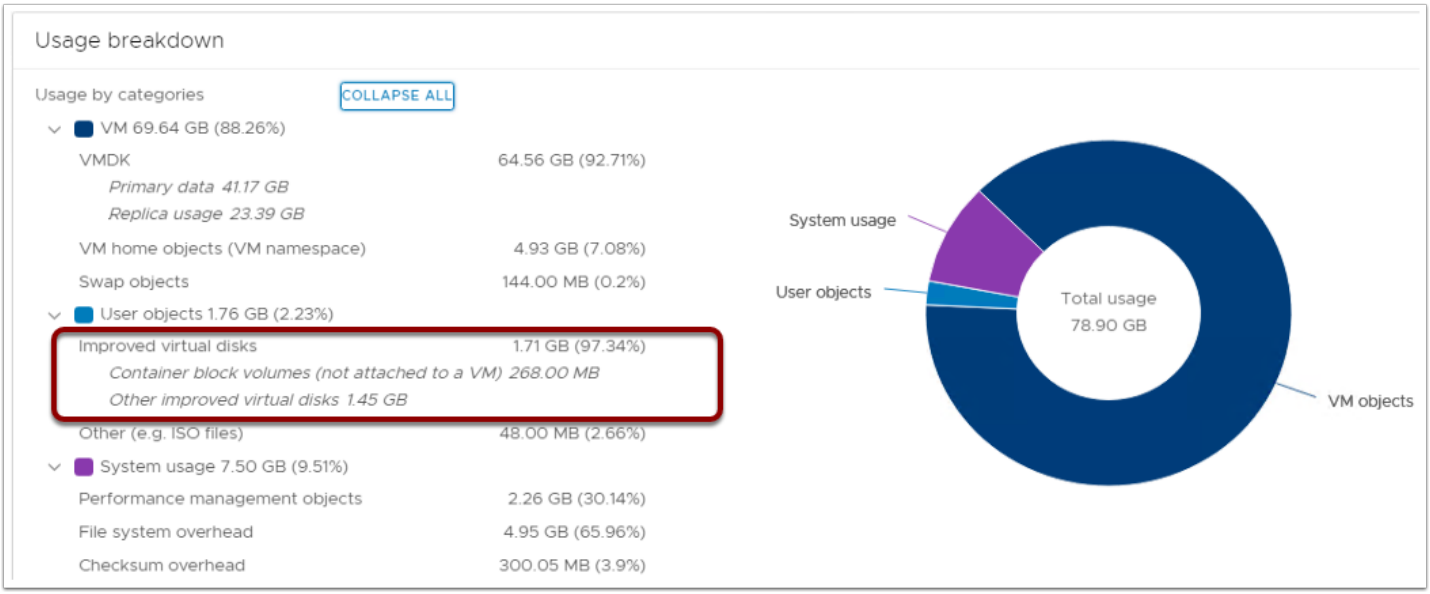
Delete Pod



Still in the vSphere Client UI

1. Refresh the vSphere Client
2. Click the **EXPAND ALL** link

Delete Pod



Since we have deleted the POD, the Volume is no longer attached to a POD and appears as a **Container block volumes (not attached to a VM)**

Create a File Persistent Volume

We are positioning vSAN as a platform for both traditional virtual machine workloads and newer containerized workloads.

The vSphere CSI driver provisions **File** based Persistent Volumes which were **Read-Write-Many**, meaning that multiple Pods could consume the volume at a time.

In this module we will look at how to deploy an application that uses File Persistent Volumes. We will show you some of the features that we have build into the vSphere 7 UI for managing and monitoring containers.

In this module we will demonstrate:

1. Create a StorageClass which is using the RAID1 Storage Policy, showing that the Persistent Volume will be created on vSAN Datastore as an NFS Share.
2. Create a ReadWriteMany Persistent Volume Claim (PVC) to manually create a Persistent Volume (PV)
3. Create a Pod that uses that Persistent Volume Claim which in turn means that it gets the Persistent Volume associated with the Persistent Volume Claim
4. Launch another Pod with the same PVC which will demonstrate that these Pods **can** share the same ReadWriteMany volume
5. Demonstrate the new features in the vSphere 7 UI for managing and monitoring Persistent Volumes

Kubernetes uses configuration files written in YAML (Yet another Markup Language) to specify configuration-type information by using a combination of Maps of name-value pairs of lists of items.

We have pre-created all of these configuration files for you in these labs.

Create a Storage Claim (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat file-sc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: file-sc
provisioner: csi.vsphere.vmware.com
allowVolumeExpansion: true
parameters:
  storagepolicyname: "RAID1"
  csi.storage.k8s.io/fstype: nfs4
root@k8s-master:~#
root@k8s-master:~#
```

First, we need to create a Storage Claim.

- Run the following command to display the YAML manifest that we will use to create the Storage Claim:

```
cat file-sc.yaml
```

1. The **kind** field describes what each object is. This first manifest is the StorageClass, which put simply, select a vSphere datastore in which to place any provisioned Persistent Volumes.
2. The **provisioner** field is a reference to the VMware CSI driver.
3. The **storagepolicyname** field refers to an SPBM policy is vSphere. In this case, that policy will result in selecting the vSAN Datastore.

Create a Storage Claim (2)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl apply -f file-sc.yaml
storageclass.storage.k8s.io/file-sc created
root@k8s-master:~#
root@k8s-master:~#
```

- To create the Storage Claim, run the following command:

```
kubectl apply -f file-sc.yaml
```

The storage claim will be created

Verify Storage Claim was created

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get sc
NAME                 PROVISIONER              RECLAIMPOLICY   VOLUMEBINDINGMODE   ALLOWVOLUMEEXPANSION
   AGE
block-sc (default)   csi.vsphere.vmware.com   Delete          Immediate           false
   127m
file-sc              csi.vsphere.vmware.com   Delete          Immediate           true
   17s
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the storage claim was created, run the following command :

```
kubectl get sc
```

A new storage claim called **file-sc** has been created.

Create Persistent Volume Claim (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat file-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: file-pvc
  labels:
    type: file
    size: 3Gi
spec:
  storageClassName: file-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 3Gi
root@k8s-master:~#
root@k8s-master:~#
```

First, we need to create a Persistent Volume Claim.

- Run the following command to display the YAML manifest that we will use to create the Storage Claim:

```
cat file-pvc.yaml
```

This is the Persistent Volume Claim manifest.

It results in the creation of a 3Gi Persistent Volume (VMDK) on vSphere storage reference by the **StorageClassName**.

Since this StorageClassName refers to the StorageClass above, this PV will be created on the vSAN datastore.

Create Persistent Volume Claim (2)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl apply -f file-pvc.yaml
persistentvolumeclaim/file-pvc created
root@k8s-master:~#
root@k8s-master:~#
```

- To create the Storage Claim, run the following command:

```
kubectl apply -f file-pvc.yaml
```

Verify Persistent Volume Claim was created

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pvc
NAME         STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
   AGE
block-pvc    Bound     pvc-d0320d24-6c12-4bb3-b964-1e678d7cff86   2Gi        RWO            block-sc
   126m
file-pvc     Bound     pvc-86ce5ffd-6e81-4c66-9638-7eda01211ceb   3Gi        RWX            file-sc
   84s
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the Persistent Volume Claim was created, run the following command :

```
kubectl get pvc
```

If the STATUS reports pending, run the same command again until it reports Bound.

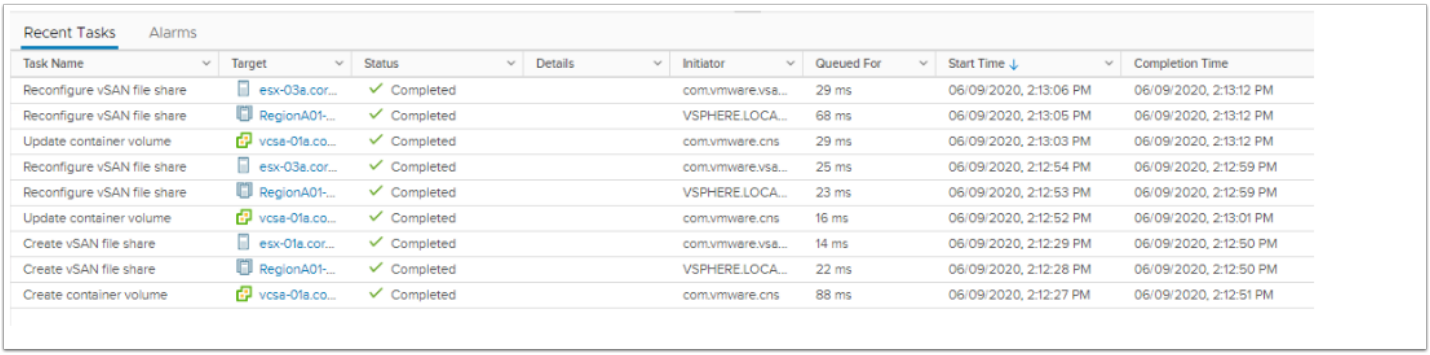
Verify Persistent Volume was created

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
             STORAGECLASS   REASON   AGE
pvc-86ce5ffd-6e81-4c66-9638-7eda01211ceb   3Gi        RWX            Delete           Bound    default
/file-pvc    file-sc                  2m59s
pvc-d0320d24-6c12-4bb3-b964-1e678d7cff86   2Gi        RWO            Delete           Bound    default
/block-pvc   block-sc                 127m
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the Persistent Volume was created, run the following command :

```
kubectl get pv
```

Persistent Volume created in vSphere (1)

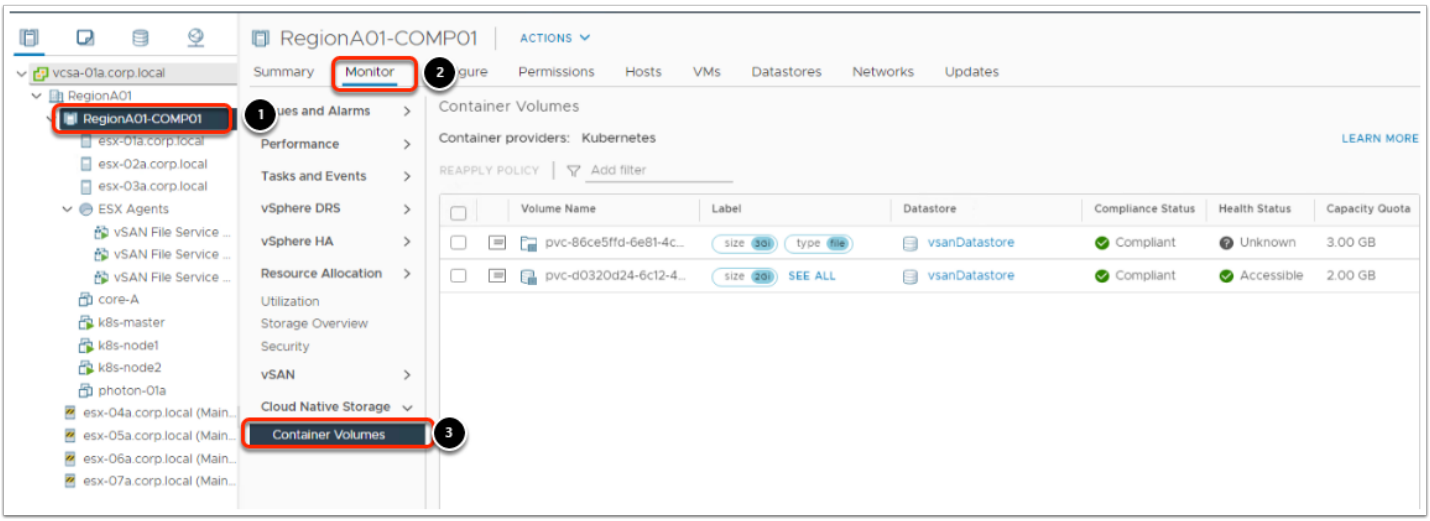


Recent Tasks | Alarms

Task Name	Target	Status	Details	Initiator	Queued For	Start Time	Completion Time
Reconfigure vSAN file share	esx-03a.cor...	Completed		com.vmware.vsa...	29 ms	06/09/2020, 2:13:06 PM	06/09/2020, 2:13:12 PM
Reconfigure vSAN file share	RegionA01-...	Completed		VSPHERE.LOCA...	68 ms	06/09/2020, 2:13:05 PM	06/09/2020, 2:13:12 PM
Update container volume	vcsa-01a.co...	Completed		com.vmware.cns	29 ms	06/09/2020, 2:13:03 PM	06/09/2020, 2:13:12 PM
Reconfigure vSAN file share	esx-03a.cor...	Completed		com.vmware.vsa...	25 ms	06/09/2020, 2:12:54 PM	06/09/2020, 2:12:59 PM
Reconfigure vSAN file share	RegionA01-...	Completed		VSPHERE.LOCA...	23 ms	06/09/2020, 2:12:53 PM	06/09/2020, 2:12:59 PM
Update container volume	vcsa-01a.co...	Completed		com.vmware.cns	16 ms	06/09/2020, 2:12:52 PM	06/09/2020, 2:13:01 PM
Create vSAN file share	esx-01a.cor...	Completed		com.vmware.vsa...	14 ms	06/09/2020, 2:12:29 PM	06/09/2020, 2:12:50 PM
Create vSAN file share	RegionA01-...	Completed		VSPHERE.LOCA...	22 ms	06/09/2020, 2:12:28 PM	06/09/2020, 2:12:50 PM
Create container volume	vcsa-01a.co...	Completed		com.vmware.cns	88 ms	06/09/2020, 2:12:27 PM	06/09/2020, 2:12:51 PM

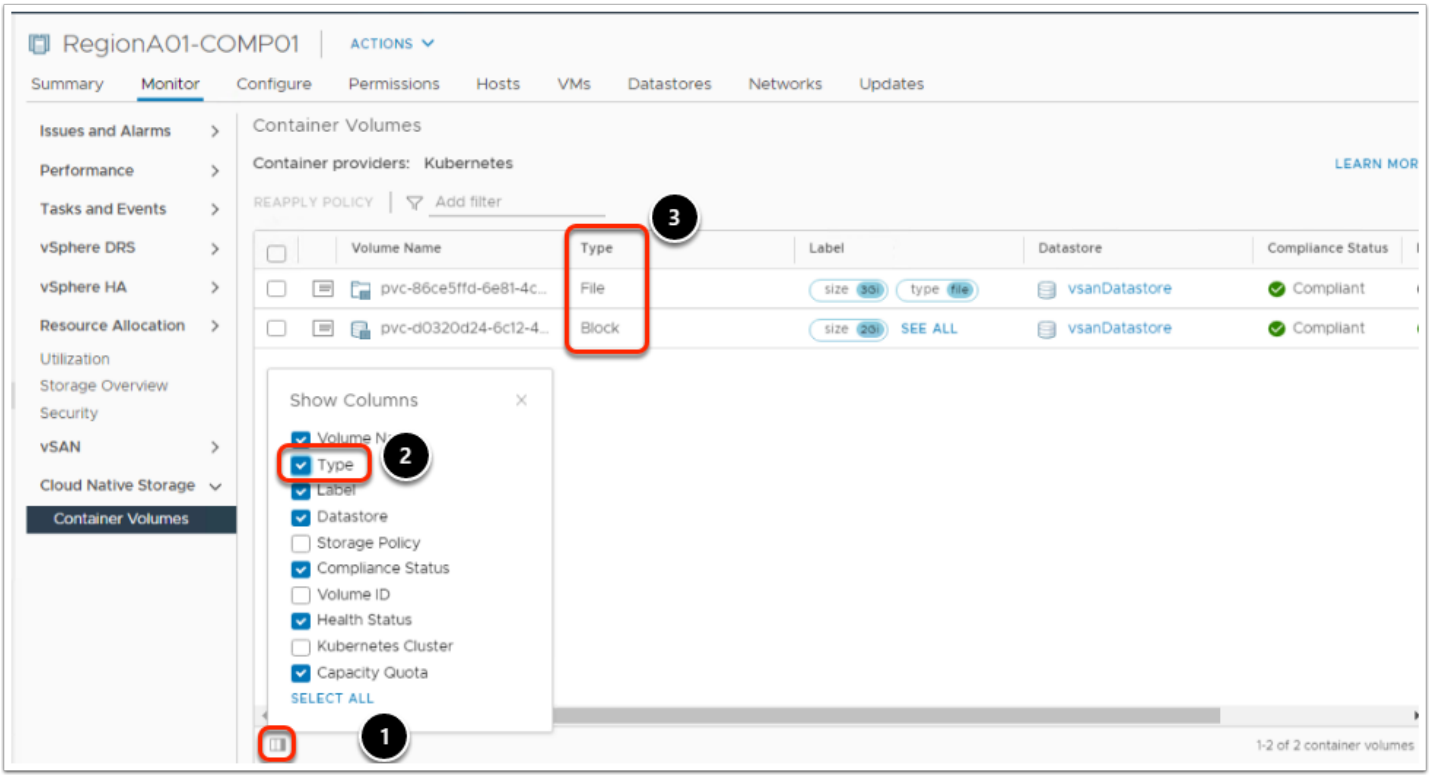
- Review the Recent Tasks in vSphere UI

Persistent Volume created in vSphere (2)



1. Select the Cluster called RegionA01-COMP01
2. Select Monitor
3. Then, Cloud Native Storage > Container Volumes

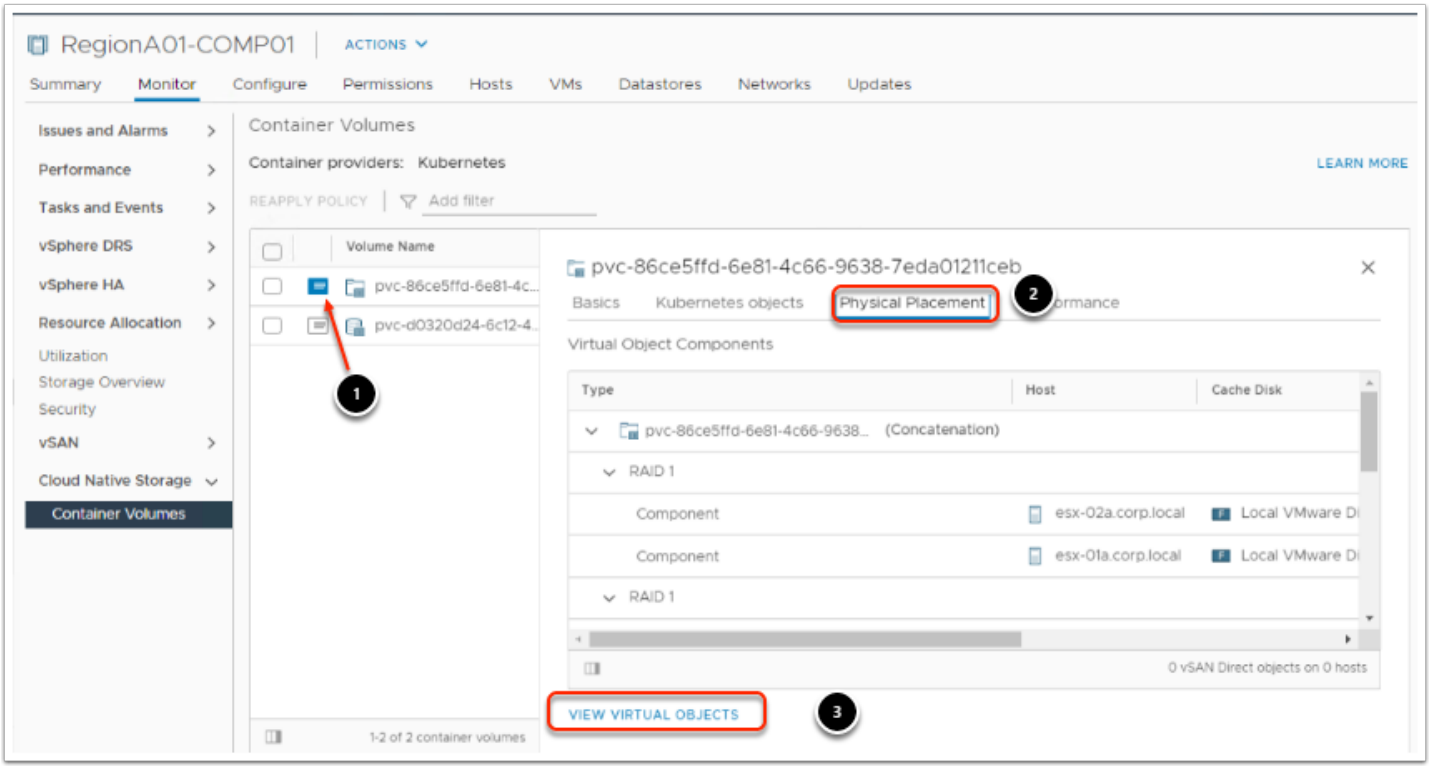
Persistent Volume created in vSphere (3)



To differentiate between Block and File Persistent Volumes, we can add another column to the display

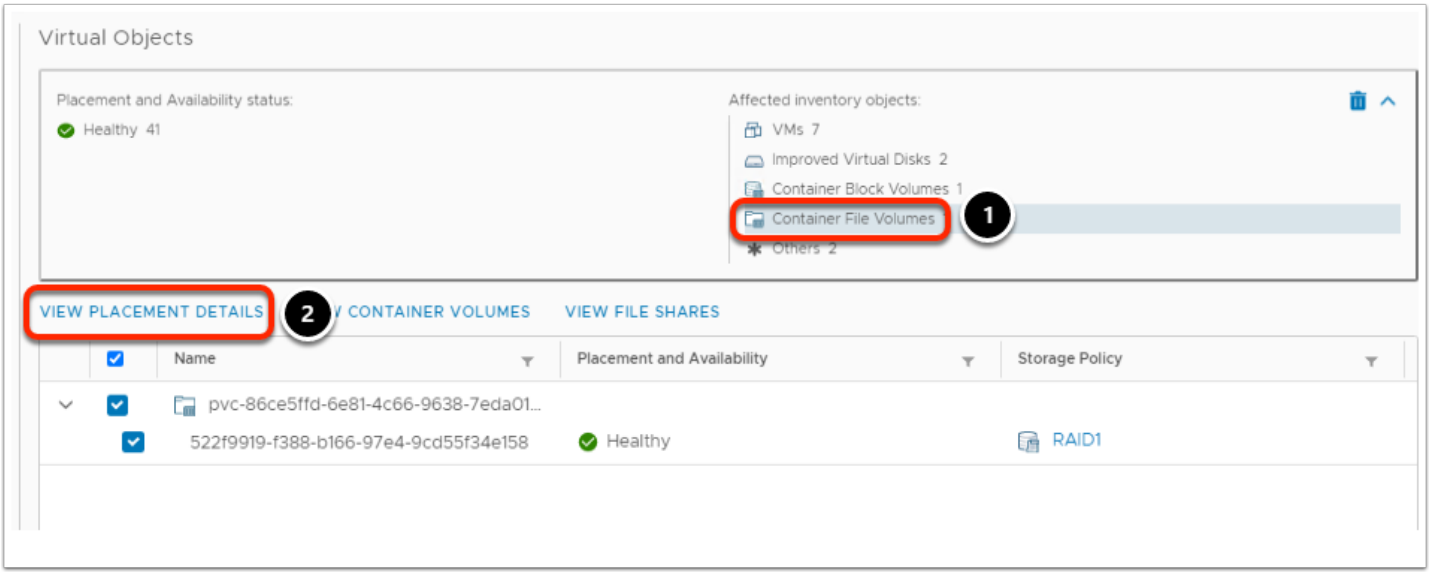
1. Click the Show Columns menu
2. Select Type
3. The Persistent Volume Type will be shown

Persistent Volume created in vSphere (4)



1. Click the Details for the File Persistent Volume to get more information about the Persistent Volume
   We can see the Persistent Volume Name, Persistent Volume Claim and any Labels that we have applied.
2. Click on **Physical Placement**
3. Click on **VIEW VIRTUAL OBJECTS**

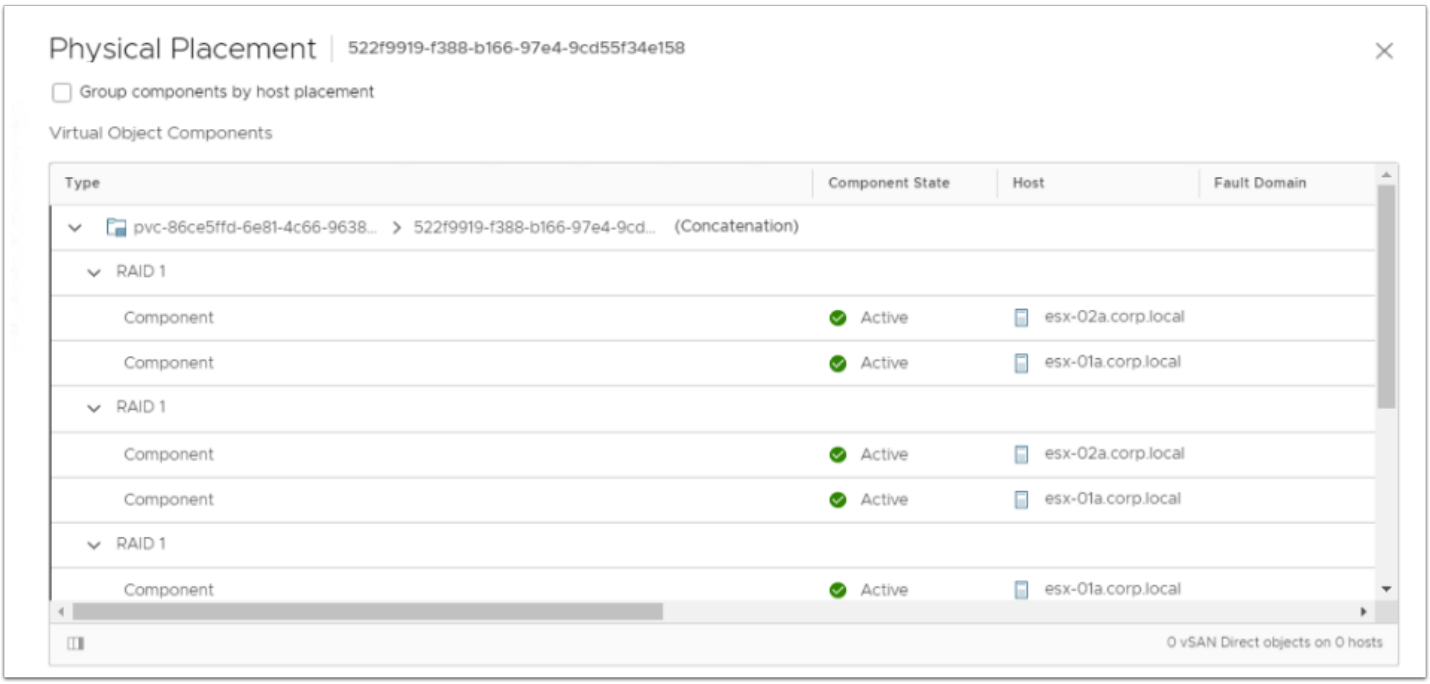
View Placement Details (1)



If we click on the volume name, and then **View Placement Details**, we can see that the actual layout of the Persistent Volume across vSAN hosts and vSAN disks, allowing vSphere admins to track right down and see exactly which infrastructure components are backing a volume.

1. Verify that **Container File Volumes** is selected
2. Click **View Placement Details**

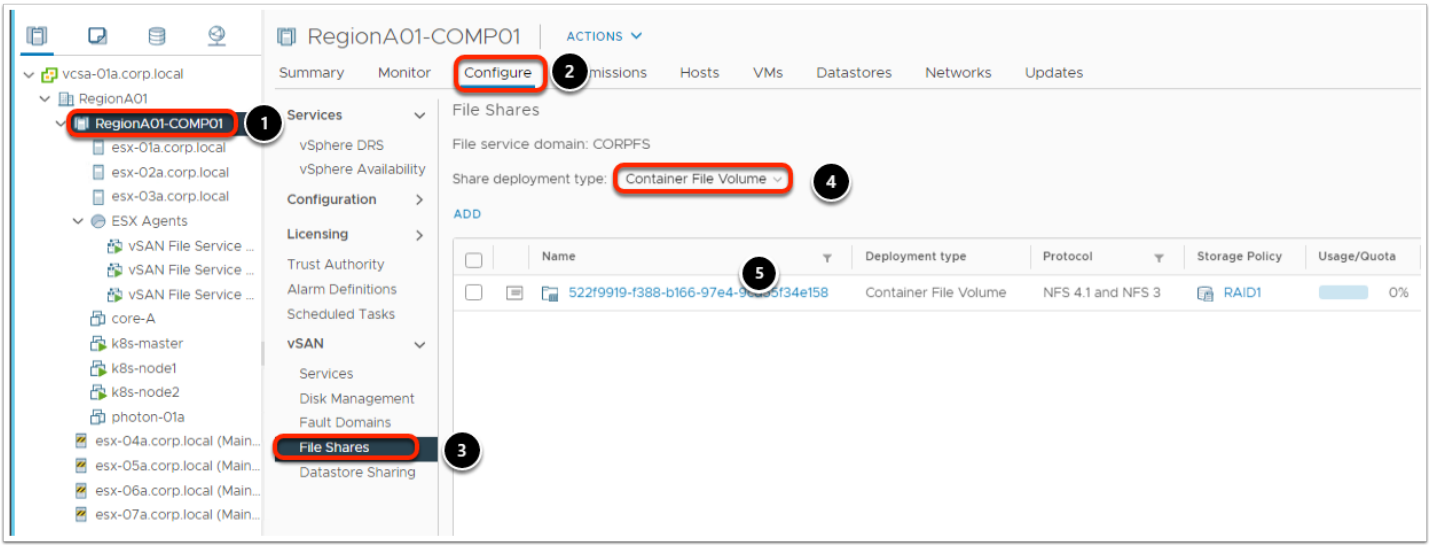
View Placement Details (2)



We can see that it has been instantiated as a RAID-1 (Mirror) on the vSAN Datastore.

1. Click CLOSE

File Service Shares



If we navigate to **Configure** > **vSAN** > **File Service Shares**, we observe that a new dynamically create file share now exists, of type Container Volume.

1. Select the Cluster called **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > File Shares**
4. From the 'Share deployment type' dropdown selector, select **Container File Volume**
5. We observe that a new dynamically create file share now exists, of type Container File Volume

Create a POD to mount the File Volume (1)

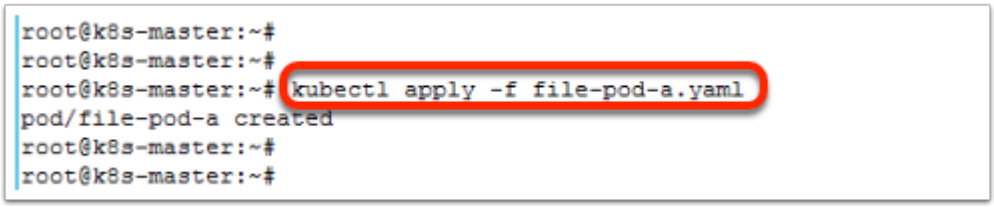
```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat file-pod-a.yaml
apiVersion: v1
kind: Pod
metadata:
  name: file-pod-a
spec:
  containers:
  - name: file-pod-a
    image: "k8s.gcr.io/busybox"
    volumeMounts:
    - name: file-vol
      mountPath: "/mnt/volume1"
    command: [ "sleep", "1000000" ]
  volumes:
    - name: file-vol
      persistentVolumeClaim:
        claimName: file-pvc
root@k8s-master:~#
root@k8s-master:~#
```

Finally, we need to create a Pod to mount the Volume. Return to Putty session.

- Run the following command to display the YAML manifest that we will use to create the POD:

```
cat file-pod-a.yaml
```

Create a POD to mount the Volume (2)



- Run the following command

```
kubectl apply -f file-pod-a.yaml
```

Verify the POD is running

- To verify the POD was created, run the following command :

```
kubectl get pod
```

The Pod is ready and the Status is Running.

IF the Pod reports a STATUS of ContainerCreating, run the same command again.

Examine POD Events

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get event --field-selector involvedObject.name=file-pod-a
LAST SEEN   TYPE      REASON                  OBJECT            MESSAGE
2m3s        Normal    Scheduled               pod/file-pod-a    Successfully assigned default/file-pod-
a to k8s-node1
2m3s        Normal    SuccessfulAttachVolume  pod/file-pod-a    AttachVolume.Attach succeeded for volum
e "pvc-86ce5ffd-6e81-4c66-9638-7eda01211ceb"
114s        Normal    Pulling                 pod/file-pod-a    Pulling image "k8s.gcr.io/busybox"
111s        Normal    Pulled                  pod/file-pod-a    Successfully pulled image "k8s.gcr.io/b
usybox" in 2.225603229s
111s        Normal    Created                 pod/file-pod-a    Created container file-pod-a
110s        Normal    Started                 pod/file-pod-a    Started container file-pod-a
root@k8s-master:~#
root@k8s-master:~#
```

- We can use the following command to examine the events associated with the creation of the Pod.

```
kubectl get event --field-selector involvedObject.name=file-pod-a
```

We can see that it Successfully Attached the Volume to the POD

Check Volume was mounted in the POD

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl exec -it file-pod-a -- /bin/sh        1
/ #
/ #
/ # mount | grep /mnt/volume1        2
192.168.130.103:/52f44abc-24f1-ca15-3aba-85dc50f356cf on /mnt/volume1 type nfs4 (rw,relatime,vers=4
.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=10.
244.1.2,local_lock=none,addr=192.168.130.103)
/ #
/ #        3
/ # df /mnt/volume1
Filesystem           1K-blocks      Used Available Use% Mounted on
192.168.130.103:/52f44abc-24f1-ca15-3aba-85dc50f356cf
                      38796288         0  38693888   0% /mnt/volume1
/ #
/ #
```

1. First lets login to the POD by running the following command:

```
kubectl exec -it file-pod-a -- /bin/sh
```

2. Check is the volume mounted by running the following command:

```
mount | grep /mnt/volume1
```

3. Finally, lets check the size of the Volume

```
df /mnt/volume1
```

We can see the File Container Volume is mounted in the POD

Write something to the Volume

1. Change to the /mnt/volume1

```
cd /mnt/volume1
```

2. Make a folder called CreatedByPodA

```
mkdir CreatedByPodA
```

3. Change to the folder called CreatedByPodA

```
cd CreatedByPodA
```

4. Write some text to a file

```
echo "Pod A was here" >> sharedfile
```

5. View the contents of the file

```
cat sharedfile
```

- Exit from the Pod with the following command

```
exit
```

Create a second POD to mount the same Volume (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# cat file-pod-b.yaml
apiVersion: v1
kind: Pod
metadata:
  name: file-pod-b
spec:
  containers:
  - name: file-pod-b
    image: "k8s.gcr.io/busybox"
    volumeMounts:
    - name: file-vol
      mountPath: "/mnt/volume1"
    command: [ "sleep", "1000000" ]
  volumes:
    - name: file-vol
      persistentVolumeClaim:
        claimName: file-pvc
root@k8s-master:~#
root@k8s-master:~#
```

Create a Second Pod to mount the Volume.

- Run the following command to display the YAML manifest that we will use to create the POD:

```
cat file-pod-b.yaml
```

Create a second POD to mount the same Volume (1)

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl apply -f file-pod-b.yaml
pod/file-pod-b created
root@k8s-master:~#
root@k8s-master:~#
```

- Run the following command :

```
kubectl apply -f file-pod-b.yaml
```

Verify the POD is running

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get pod
NAME          READY   STATUS    RESTARTS   AGE
file-pod-a    1/1     Running   0          6m13s
file-pod-b    1/1     Running   0          58s
root@k8s-master:~#
root@k8s-master:~#
```

- To verify the POD was created, run the following command :

```
kubectl get pod
```

If the STATUS reports ContainerCreating, wait for a few seconds and run the command again.

Examine POD Events

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl get event --field-selector involvedObject.name=file-pod-b
LAST SEEN   TYPE     REASON                  OBJECT            MESSAGE
112s        Normal   Scheduled               pod/file-pod-b    Successfully assigned default/file-pod-
b to k8s-node2
111s        Normal   SuccessfulAttachVolume  pod/file-pod-b    AttachVolume.Attach succeeded for volum
e "pvc-86ce5ffd-6e81-4c66-9638-7eda01211ceb"
93s         Normal   Pulling                 pod/file-pod-b    Pulling image "k8s.gcr.io/busybox"
92s         Normal   Pulled                  pod/file-pod-b    Successfully pulled image "k8s.gcr.io/b
usybox" in 972.809505ms
91s         Normal   Created                 pod/file-pod-b    Created container file-pod-b
91s         Normal   Started                 pod/file-pod-b    Started container file-pod-b
root@k8s-master:~#
root@k8s-master:~#
```

- We can use the following command to examine the events associated with the creation of the Pod.

```
kubectl get event --field-selector involvedObject.name=file-pod-b
```

We can see that it Successfully Attached the Volume to the POD

Check Volume was mounted in the second POD

```
root@k8s-master:~#
root@k8s-master:~#
root@k8s-master:~# kubectl exec -it file-pod-b -- /bin/sh
/ #
/ #
/ # mount | grep /mnt/volume1
192.168.130.103:/52f44abc-24f1-ca15-3aba-85dc50f356cf on /mnt/volume1 type nfs4 (rw,relatime,vers=4
.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=10.
244.1.2,local_lock=none,addr=192.168.130.103)
/ #
/ #
/ #
/ # df /mnt/volume1
Filesystem           1K-blocks      Used Available Use% Mounted on
192.168.130.103:/52f44abc-24f1-ca15-3aba-85dc50f356cf
                      38779904         0  38677504   0% /mnt/volume1
/ #
/ #
```

1. First lets login to the POD by running the following command:

```
kubectl exec -it file-pod-b -- /bin/sh
```

2. Check is the volume mounted by running the following command:

```
mount | grep /mnt/volume1
```

3. Finally, lets check the size of the Volume

```
df /mnt/volume1
```

We can see the File Container Volume is mounted in the POD

Verify sharedfile is present

```
/ #
/ #
/ # cd /mnt/volume1/CreatedByPodA
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA # cat sharedfile
Pod A was here
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA # echo "Pod B was here also" >> sharedfile
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA # cat sharedfile
Pod A was here
Pod B was here also
/mnt/volume1/CreatedByPodA #
/mnt/volume1/CreatedByPodA #
```

1. Change to the folder called CreatedByPodA

```
cd /mnt/volume1/CreatedByPodA
```

2. View the contents of the shared file

```
cat sharedfile
```

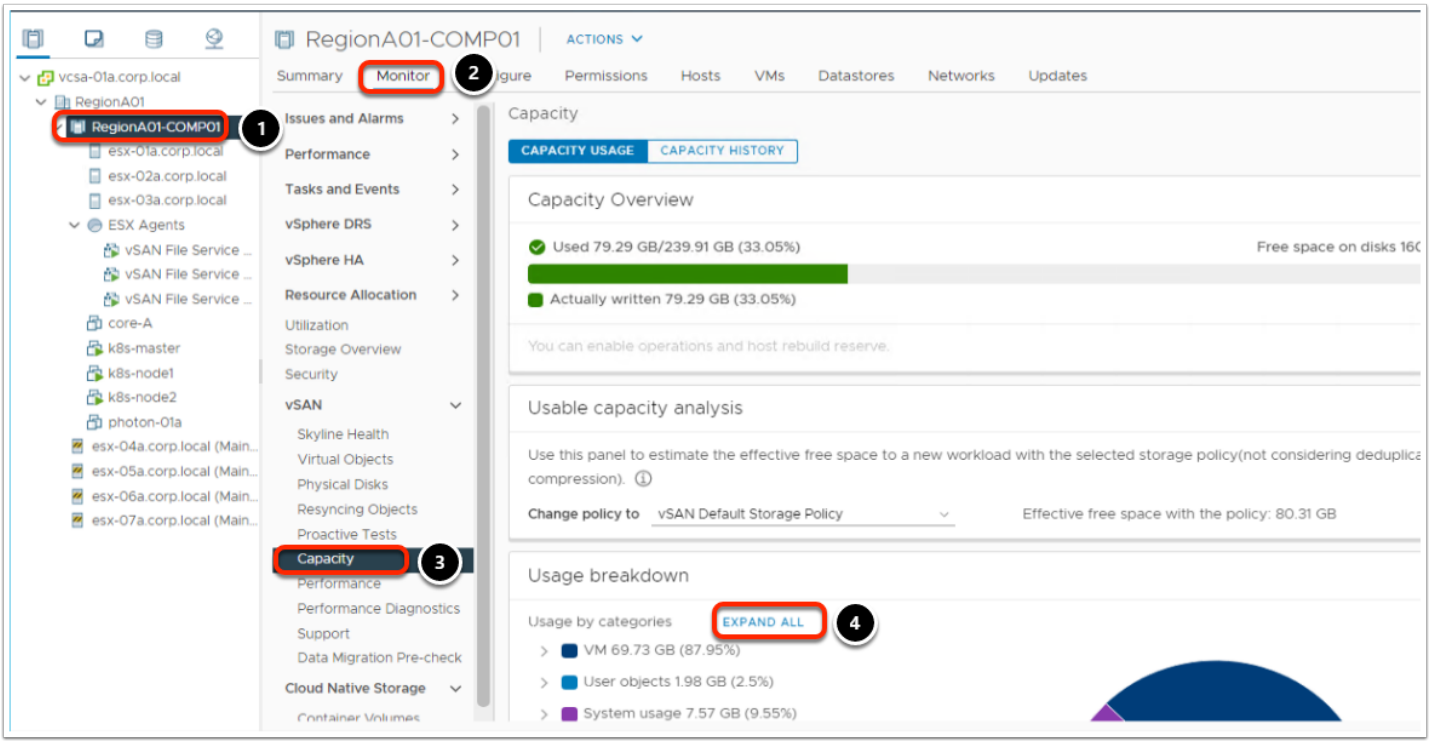
3. Write something to the sharedfile

```
echo "Pod B was here also" >> sharedfile
```

4. View the new contents of the sharedfile

```
cat sharedfile
```

vSAN Capacity View (1)



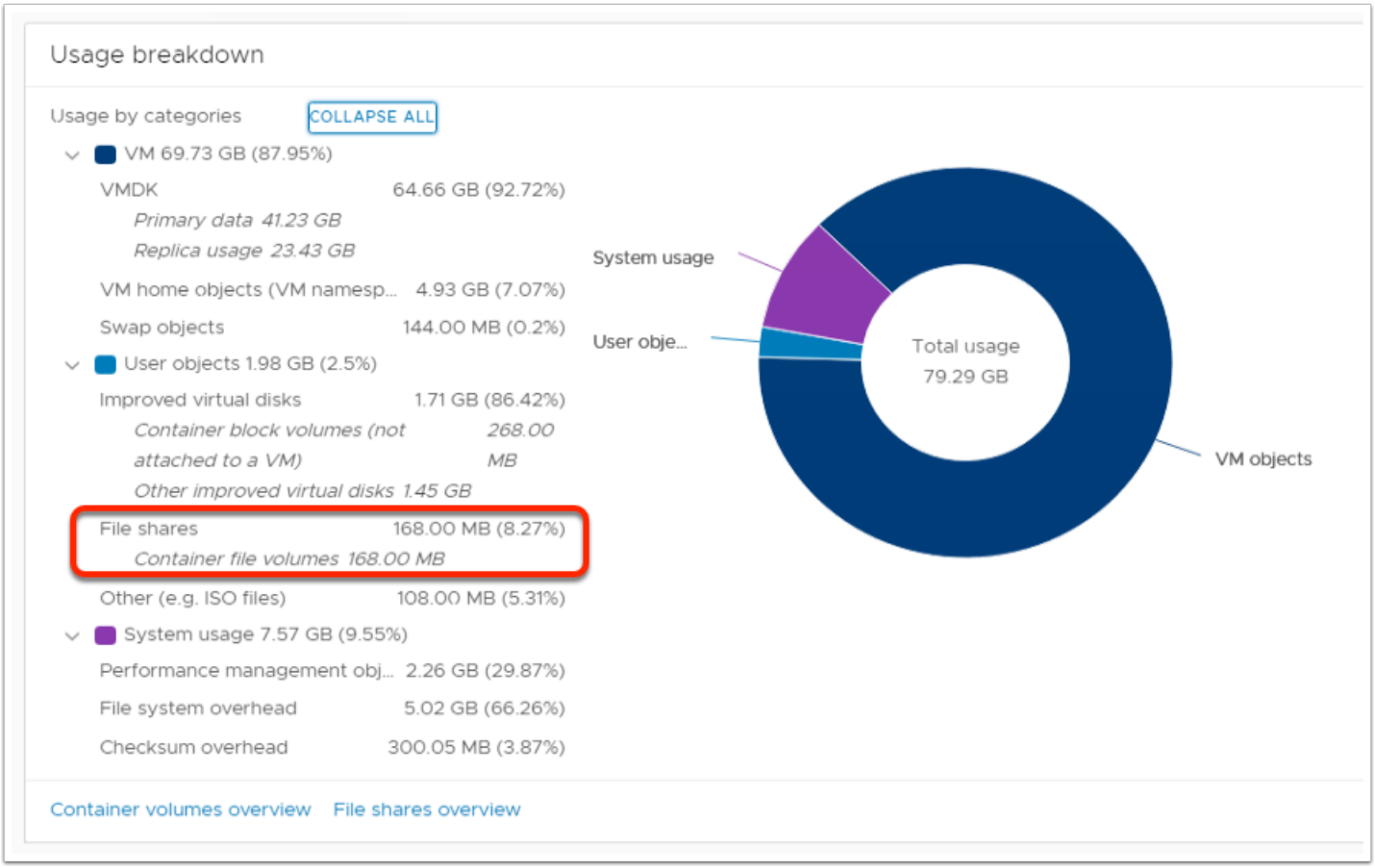
Let now have a look at the capacity consumed by the Persistent Volumes

1. Select the Cluster called **RegionA01-Comp01**
2. Select **Monitor**
3. Select **vSAN > Capacity**

What we are interested is the **Usage breakdown before dedupe and compression**, so scroll down the screen

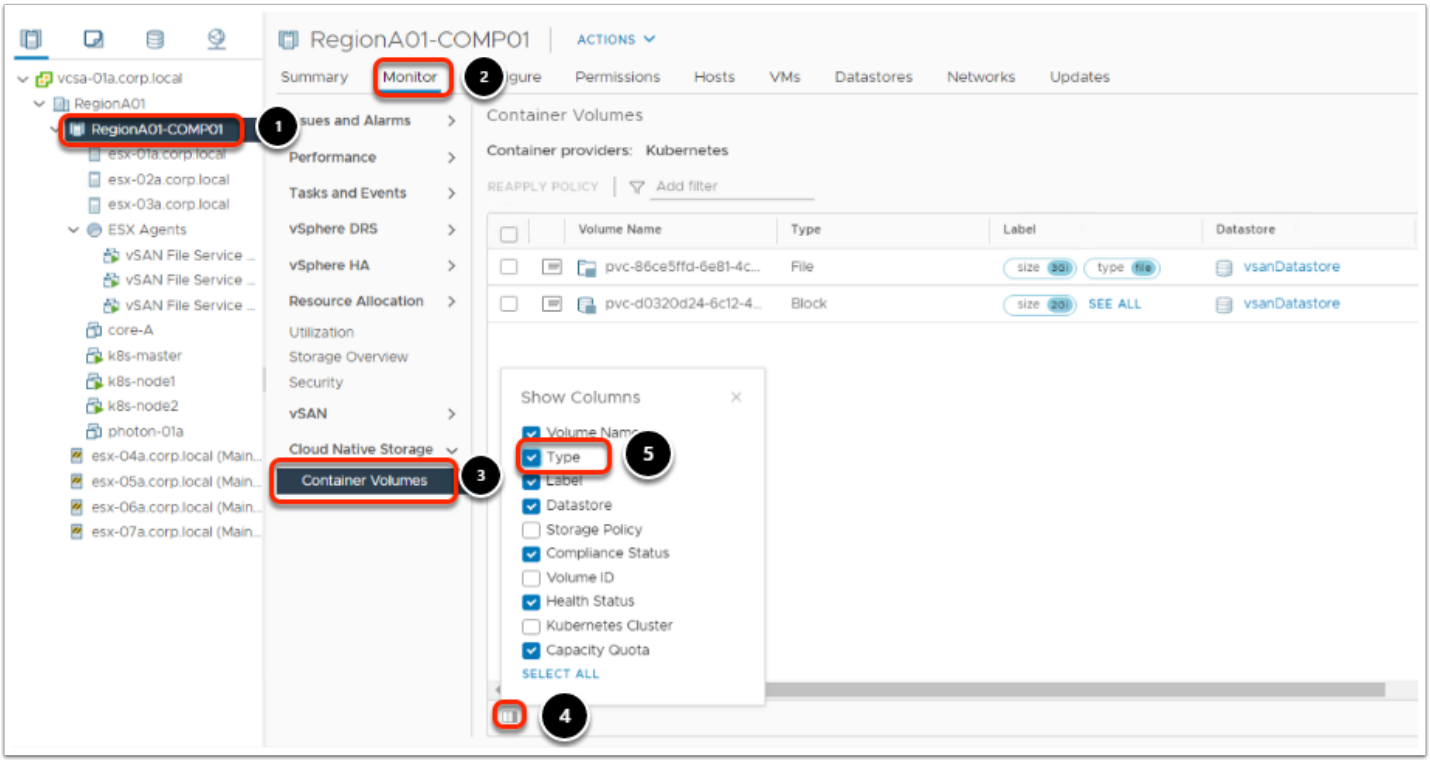
4. Click **EXPAND ALL** link

vSAN Capacity View (2)



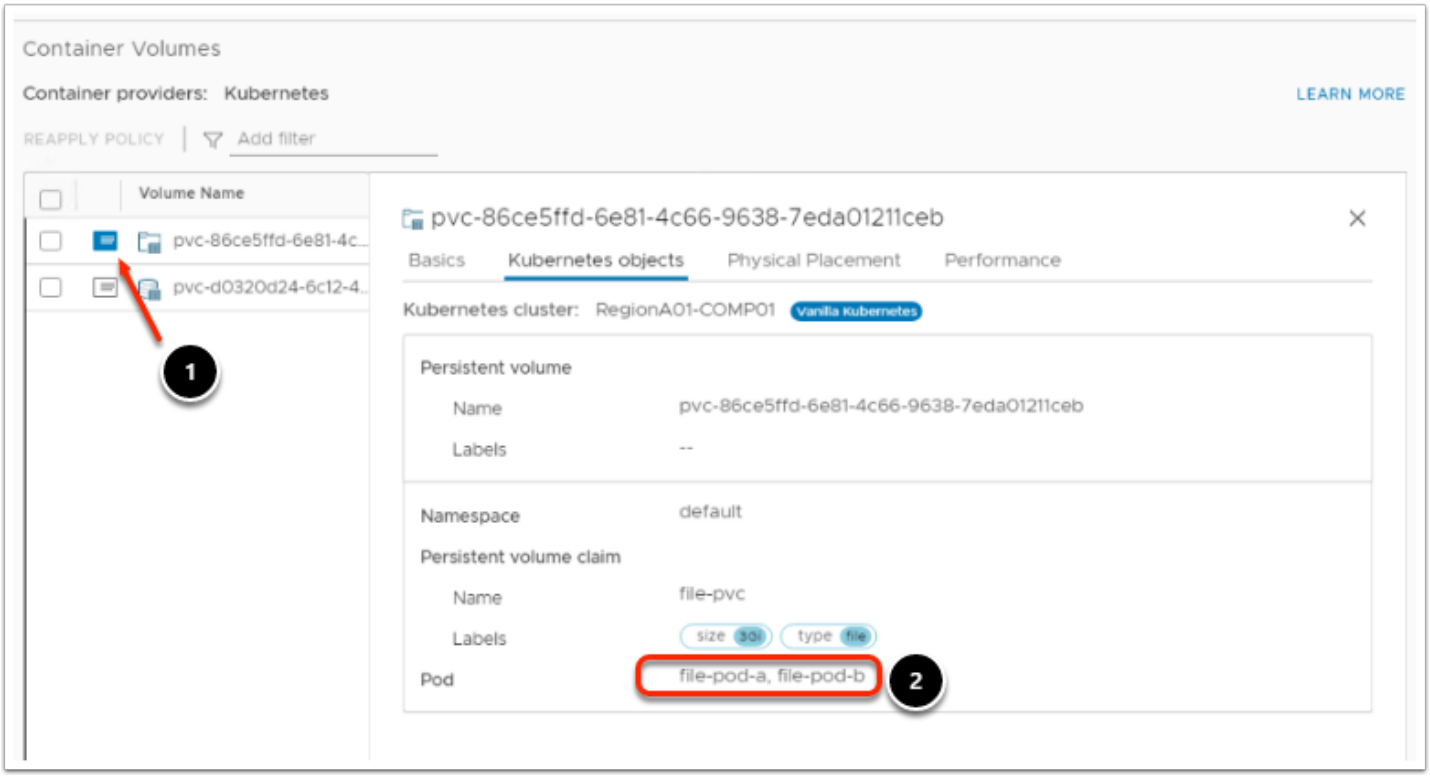
And now that the volume has been attached to a VM (the K8s worker node where the Pod was scheduled), the Usage breakdown now changes how it is reported. It now appears in the VM category, under File Shares.

Container Volumes View



1. Select the cluster called **RegionA01-COMP01**
2. Select **Monitor**
3. Select **Container Volumes**
4. Select **Show Columns**
5. Select **Type**

Container Volumes View



1. Click the Details for the Volume Name that has a type of File
2. We can see that the volume has been mounted by the 2 Pods that we created, namely file-pod-a and file-pod-b.

Conclusion

In this module, you were provided an overview of Kubernetes and how to create a block volume and file volume in vSAN Cloud Native Storage. We showed you how to mount these volumes into pods. We demonstrated that in the case of a POD with a file share volume we were able to mount that volume on multiple pods.

You Finished Module 4

Congratulations on completing Module 4.

How to End Lab

To end your lab, click on the **END** button.

Module 5 - Day 2: Monitoring Health, Capacity and Performance

Introduction

A critical aspect of enabling a vSAN Datastore is validating the Health of the environment. vSAN has over a hundred out of the box Health Checks to not only validate initial Health but also report ongoing runtime Health. vSAN 7 introduces exciting new ways to monitor the Health, Capacity and Performance of your Cluster via vRealize Operations within vCenter, all within the same User Interface that VI Administrators use today.

vSAN Health Check Validation

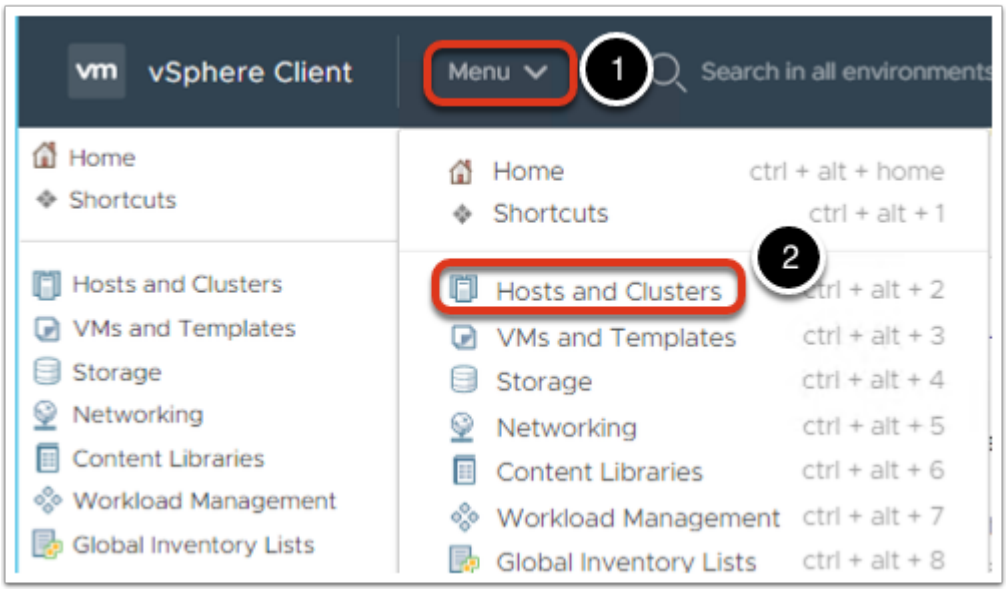
One of the ways to monitor your vSAN environment is to use the vSAN Health Check.

The vSAN Health runs a comprehensive health check on your vSAN environment to verify that it is running correctly and will alert you if it finds some inconsistencies and options on how to fix these.

vSAN Health Check

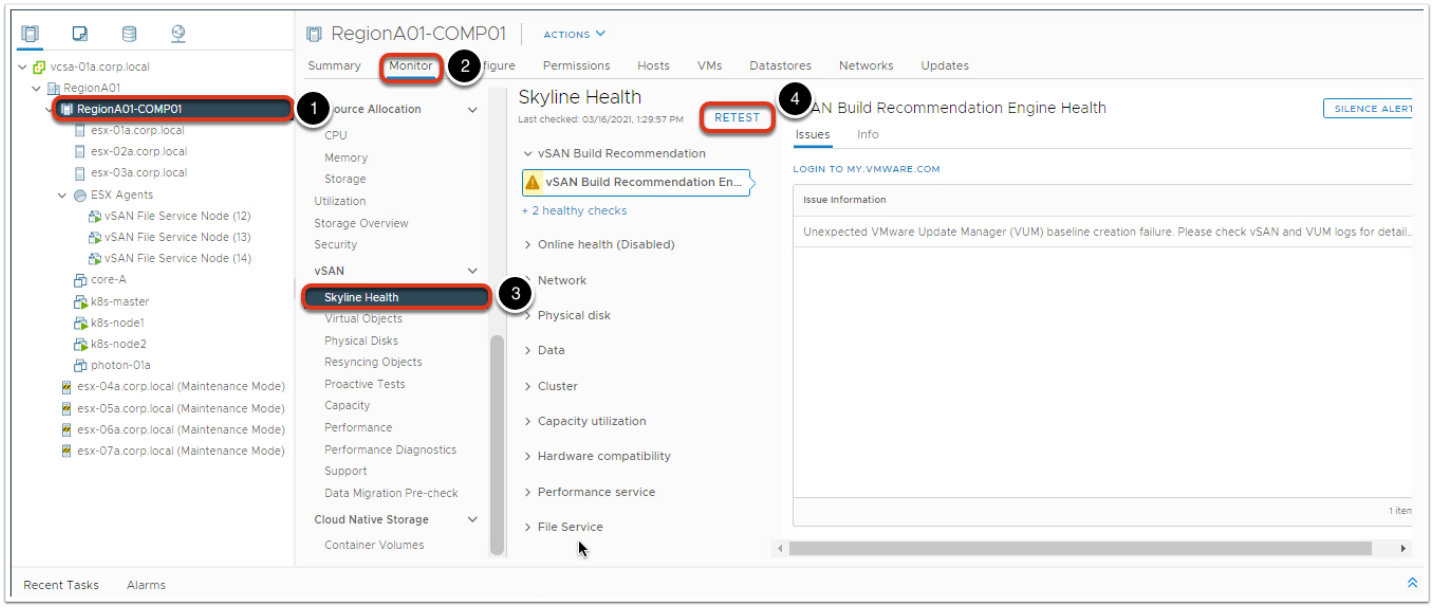
Running individual commands from one host to all other hosts in the cluster can be tedious and time consuming. Fortunately, since vSAN 6.0, vSAN has a health check system, part of which tests the network connectivity between all hosts in the cluster. One of the first tasks to do after setting up any vSAN cluster is to perform a vSAN Health Check. This will reduce the time to detect and resolve any networking issue, or any other vSAN issues in the cluster.

Use Health Check to Verify vSAN Functionality



1. Select **Menu**
2. Select **Hosts and Clusters**

Use Health Check to Verify vSAN Functionality



To run a vSAN Health Check,

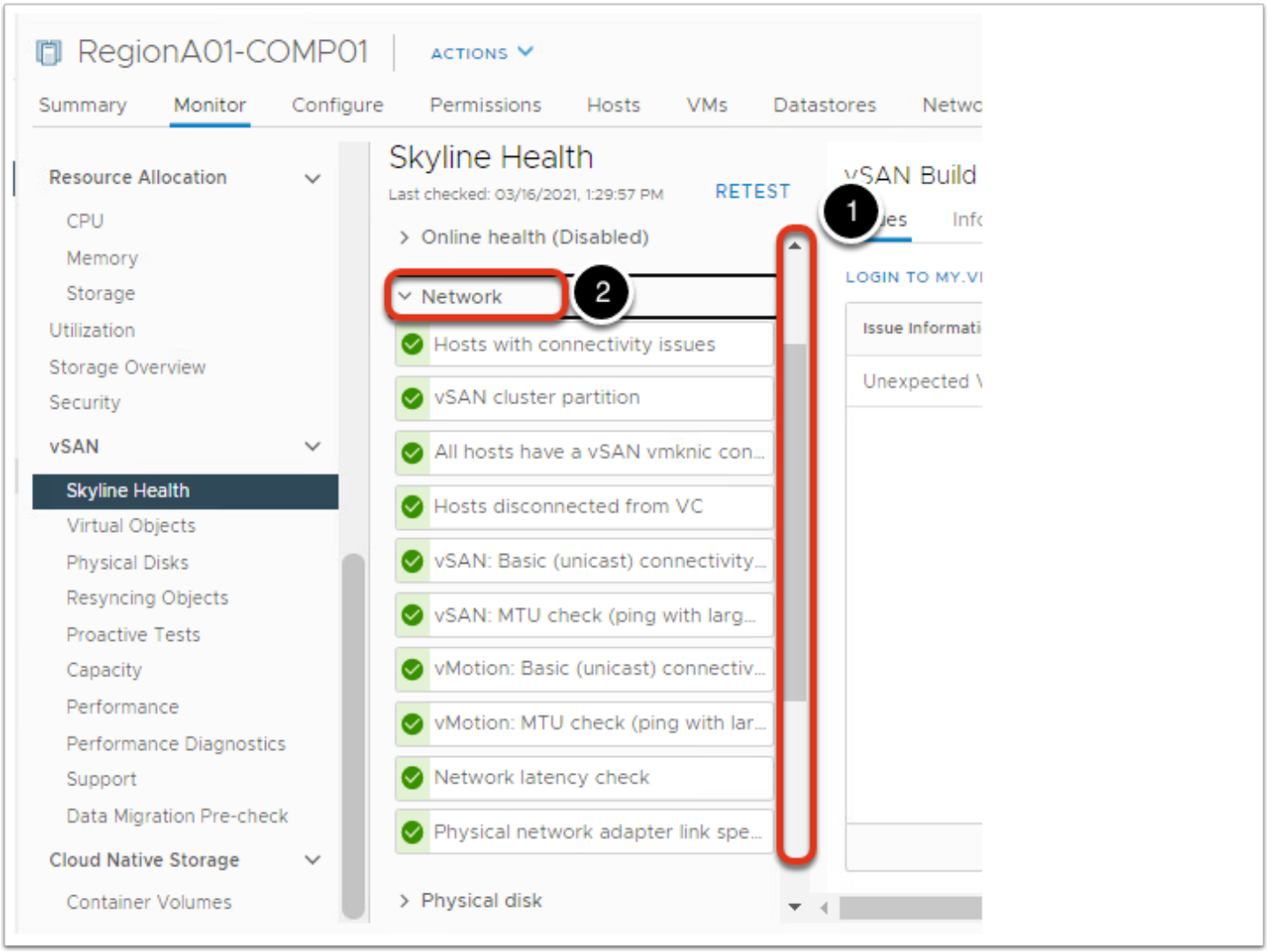
1. Select the vSAN Cluster, **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Skyline Health**

You will see the categories of Health Check that can be run and their status

4. To run the tests at any time click the **RETEST** button

Note that some of the Health Checks are in a Warning State. This is due to the fact that we are running a vSAN cluster in a nested virtualized environment.

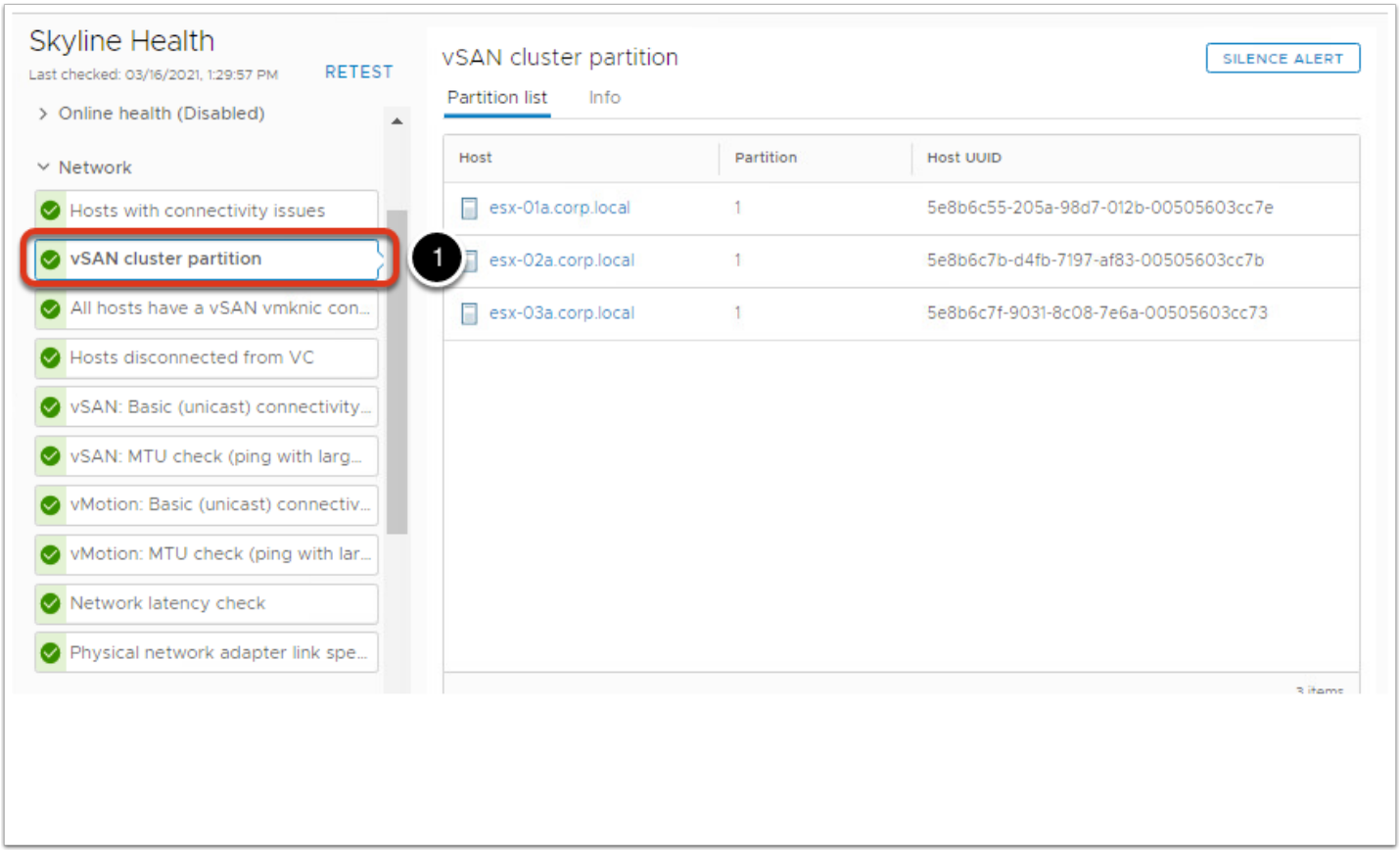
Network Health Check



To see the individual tests that cen be run from within a vSAN Health category.

1. Scroll **down** to see Network
2. Expand the **Network** health category

Getting Detail on a Network Health Check

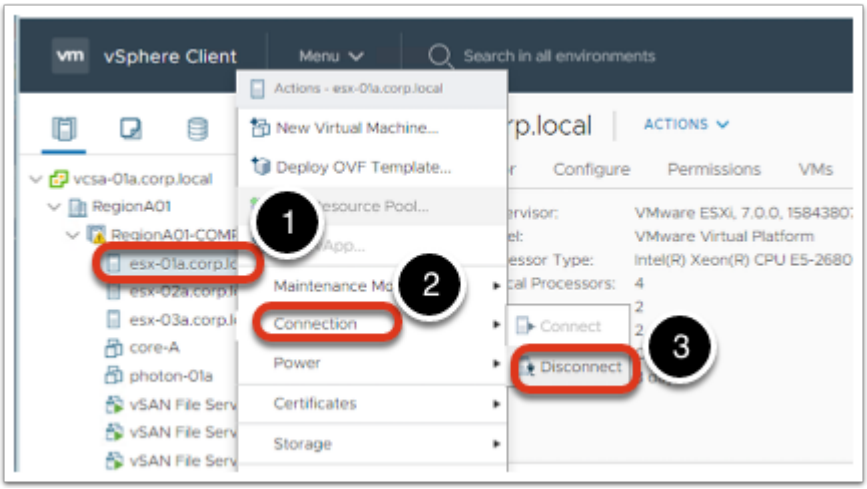


To get additional information on a Health Check item, select the appropriate check and examine the details pane on the right for information on how to resolve the issue.

1. Select **vSAN cluster partition**

Here we have the details and the results of the Health Check that was performed, in this case we can see that all the ESXi host in the vSAN Cluster have the same partition number.

Inducing a vSAN Health Check Failure

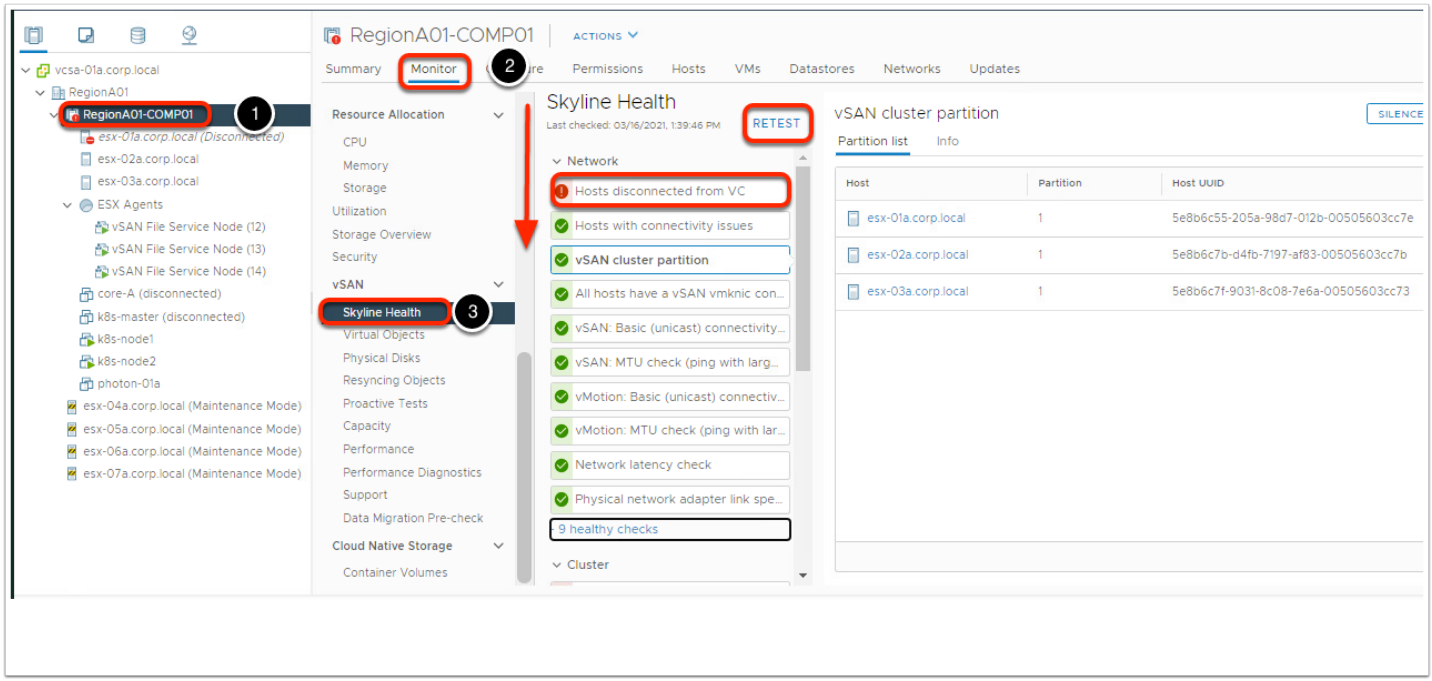


Lets induce a vSAN Health Check failure to test the health Check.

1. Right click the ESXi host called **esx-01a.corp.local**
2. Select **Connection**
3. Select **Disconnect**

- Answer **OK** to disconnect the selected host.

Inducing a vSAN Health Check Failure



Lets return to the vSAN Health Check

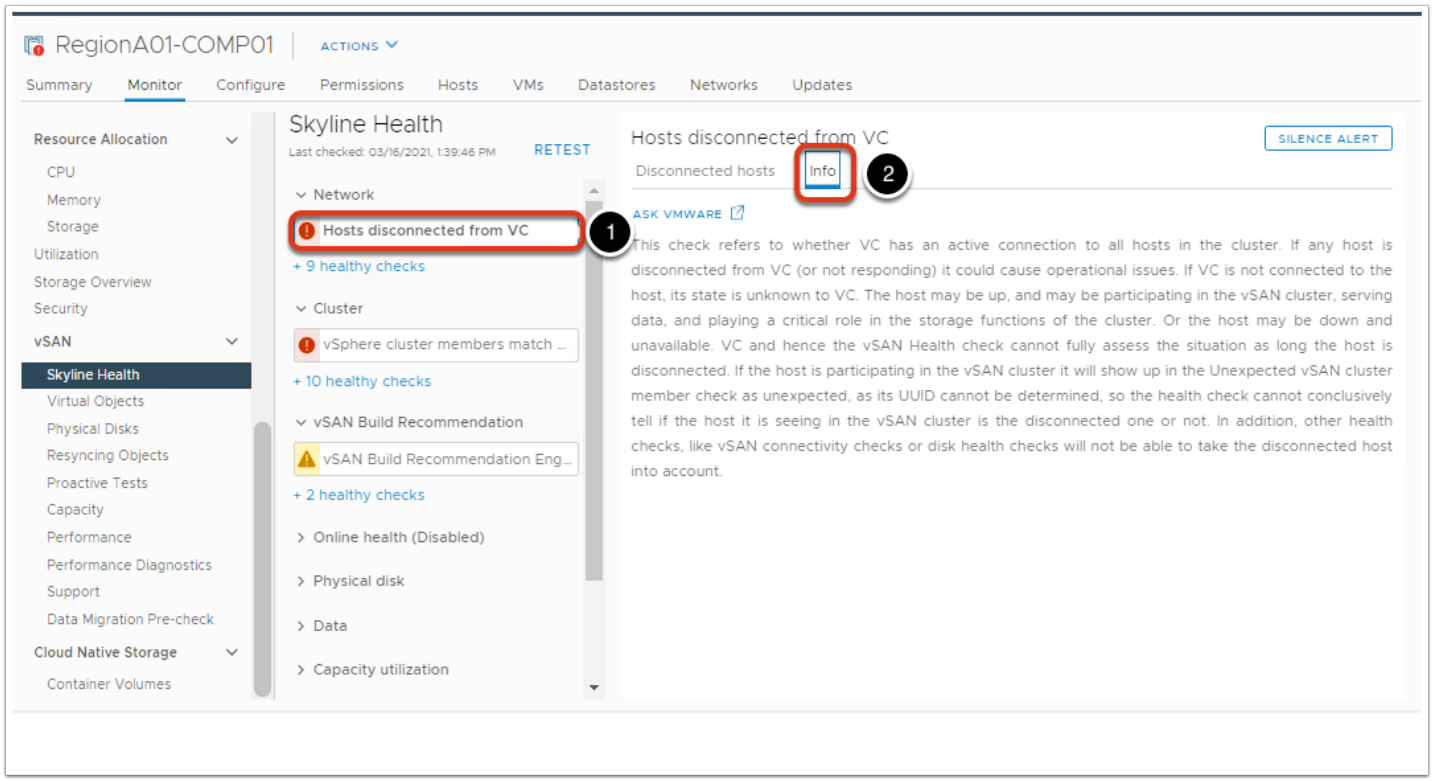
1. Select the vSAN Cluster, **RegionA01-COMP01**

2. Select **Monitor**
3. Select **vSAN Health**

Click the **RETEST** button if the Health Check for**Hosts disconnected from VC** does not show as red.

Here we will see a vSAN Network Health Check that has failed.

Inducing a vSAN Health Check Failure



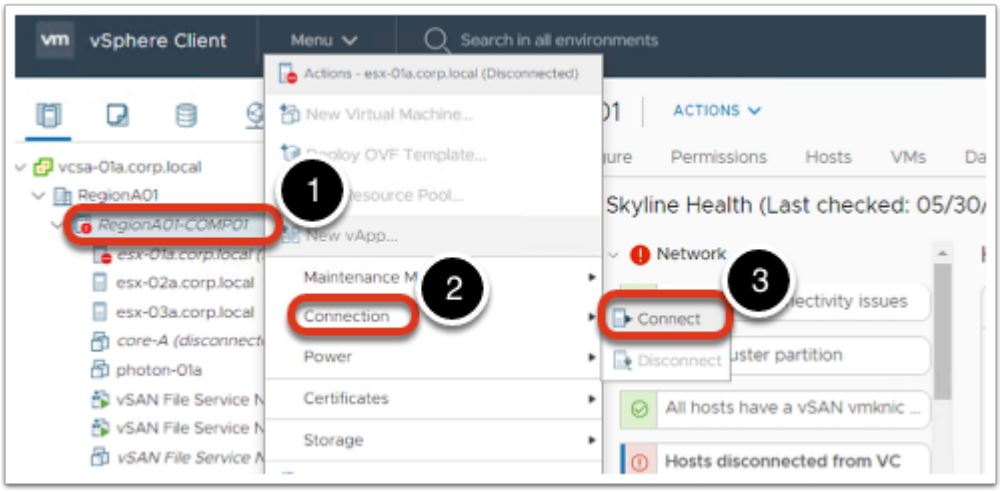
1. Click the **Hosts Disconnected from VC** to get additional information

Here we can see that the ESXi host called **esxi-01a.corp.local** is showing as **Disconnected**.

Each details view under the Info tab also contains an **Ask VMware** button where appropriate, which will take you to a VMware Knowledge Base article detailing the issue, and how to troubleshoot and resolve it.

2. Select **Info**

Resolving a vSAN Health Check Failure

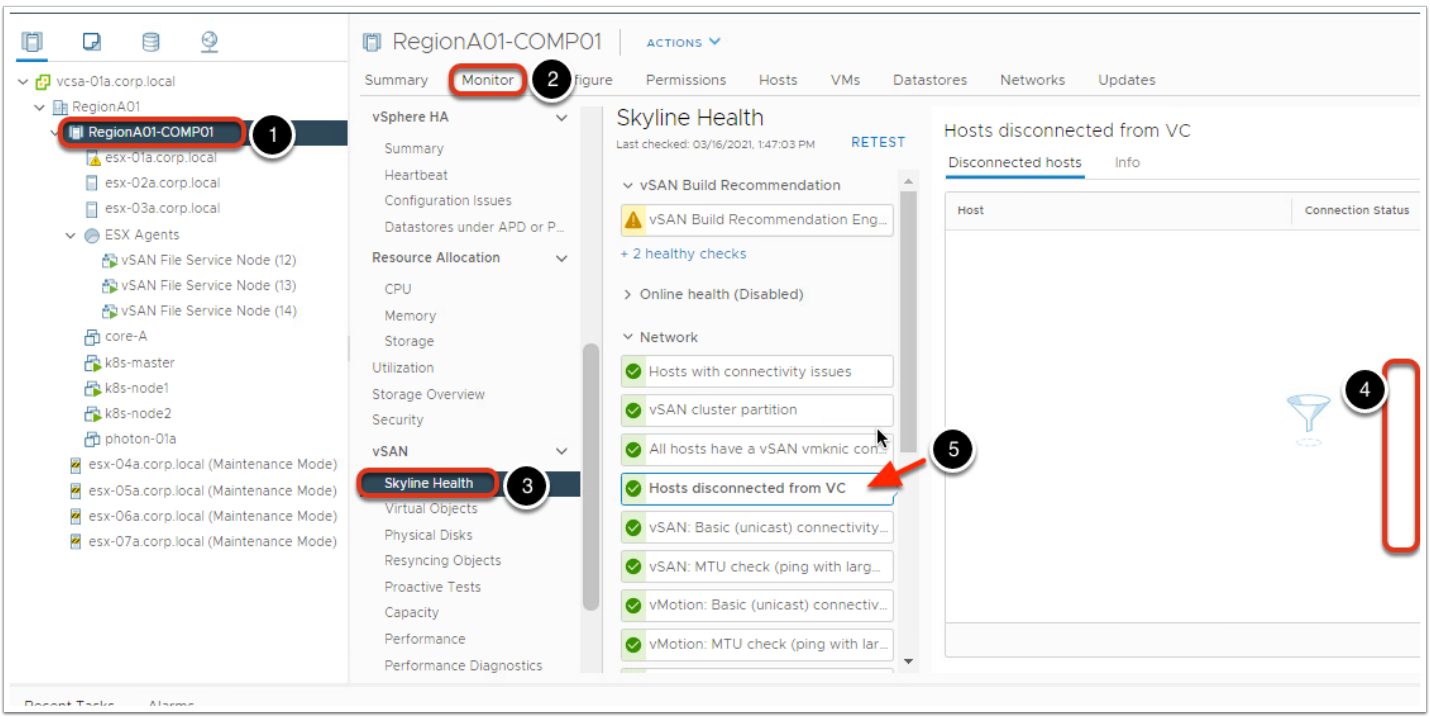


Lets resolve the vSAN Health Check failure.

1. Right click the ESXi host called **esx-01a.corp.local**
2. Select **Connection**
3. Select **Connect**

Answer **OK** to reconnect the selected host.

Resolving a vSAN Health Check Failure



Lets return to the vSAN Health Check

1. Select the vSAN Cluster, **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Health**
4. Scroll **down** to until you see Network
5. The **Hosts disconnect from VC** test has passed again as all the ESXi host in the vSAN Cluster are connected.

Click the **RETEST** button if the Health Checks does not appear green.

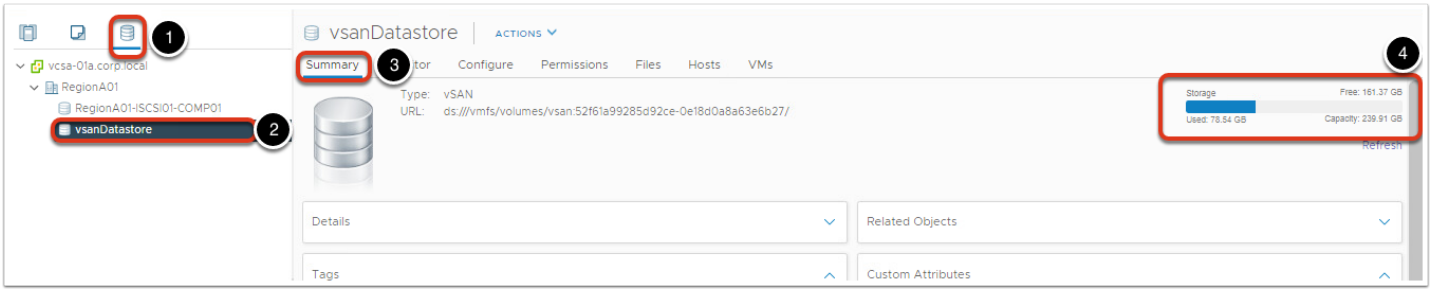
Conclusion

You can use the vSAN health checks to monitor the status of cluster components, diagnose issues, and troubleshoot problems. The health checks cover hardware compatibility, network configuration and operation, advanced vSAN configuration options, storage device health, and virtual machine objects.

Monitoring vSAN Capacity

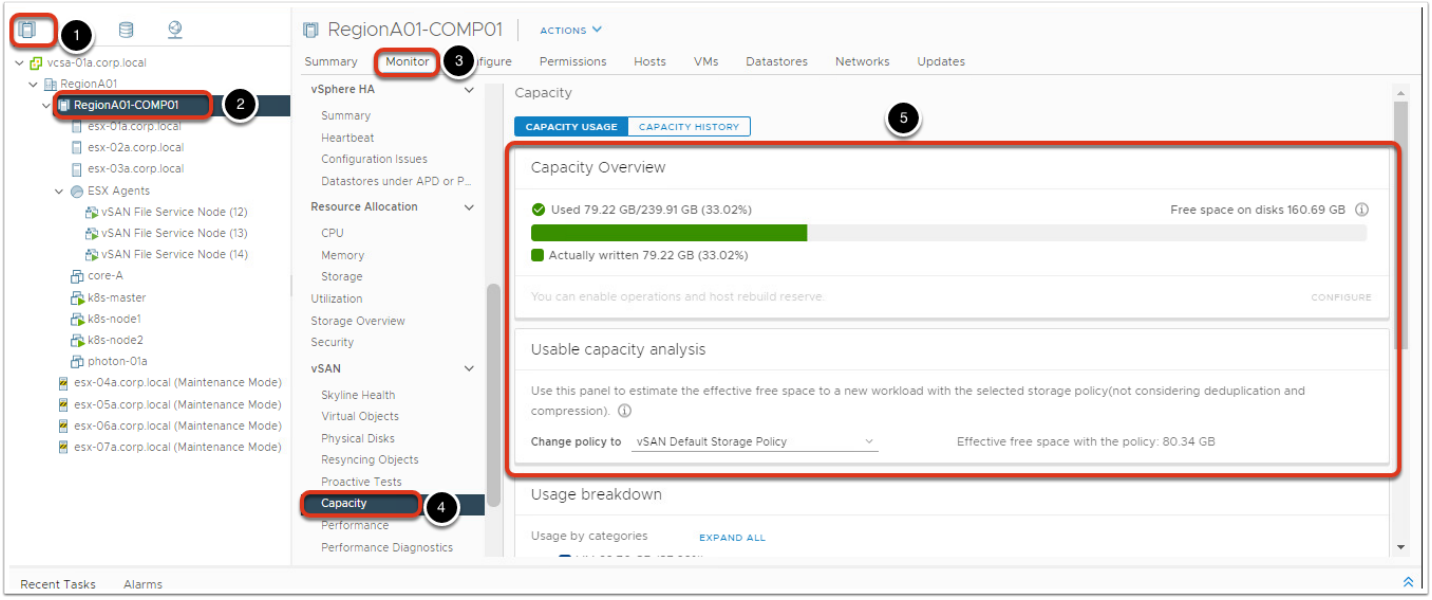
The capacity of the vSAN Datastore can be monitored from a number of locations within the vSphere Client. First, one can select the Datastore view, and view the summary tab for the vSAN Datastore. This will show you the capacity, used and free space.

Datastore View



1. Select **Storage** Icon
2. Select **RegionA01-VSAN-COMP01**
3. Click **Summary**
4. Note the amount of **Used** and **Free Capacity** Information

Capacity Overview

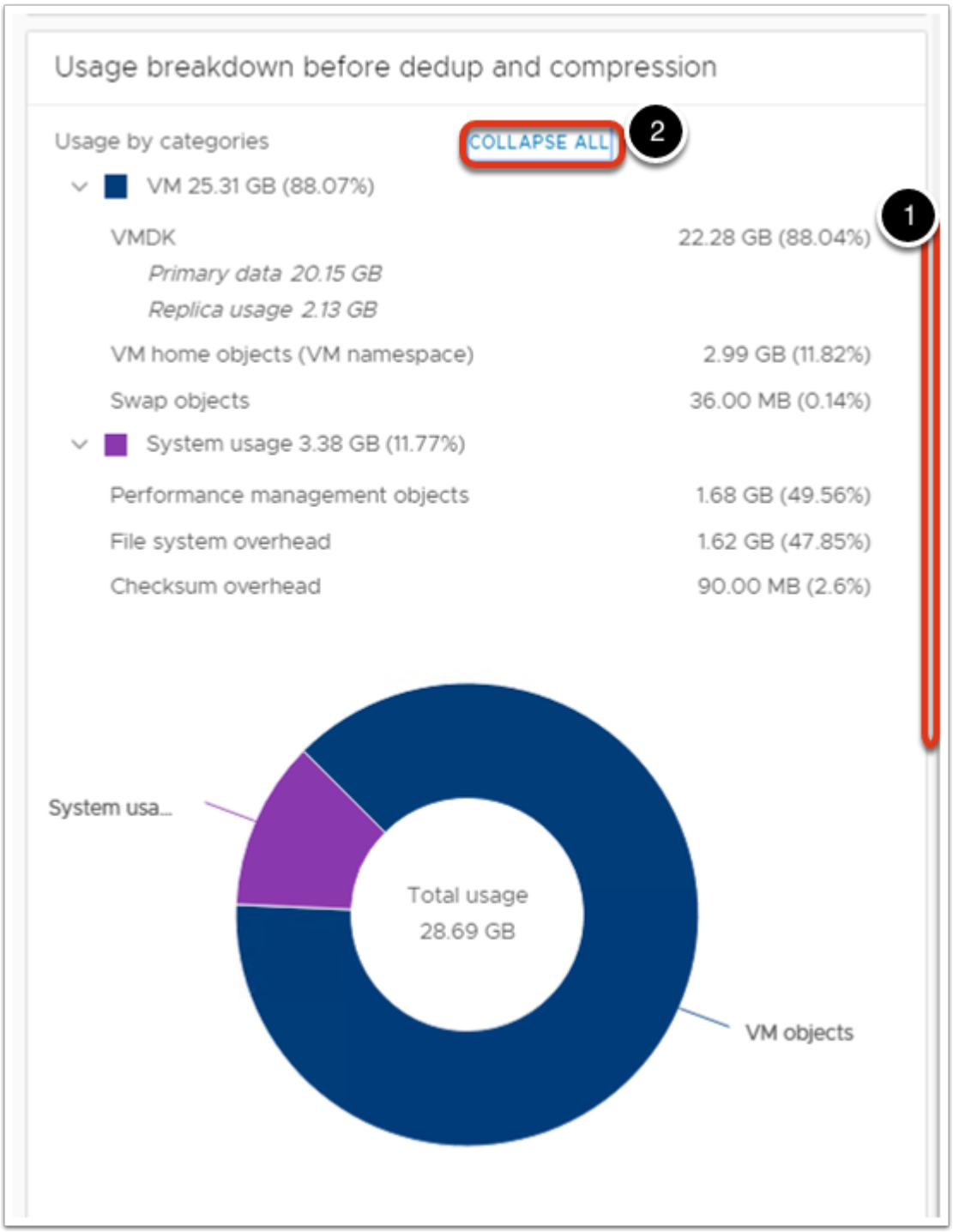


1. Select **Hosts and Clusters** Icon
2. Select **RegionA01-COMP01**
3. Select **Monitor**
4. Scroll down and Click **vSAN > Capacity**

5. Note the **Capacity Overview** and **Usable Capacity Analysis** Information

The Capacity Overview displays the storage capacity of the vSAN Datastore, including used space and free space. The Deduplication and Compression Overview indicates storage usage before and after space savings are applied, including a Ratio indicator.

Usage breakdown before dedup and compression



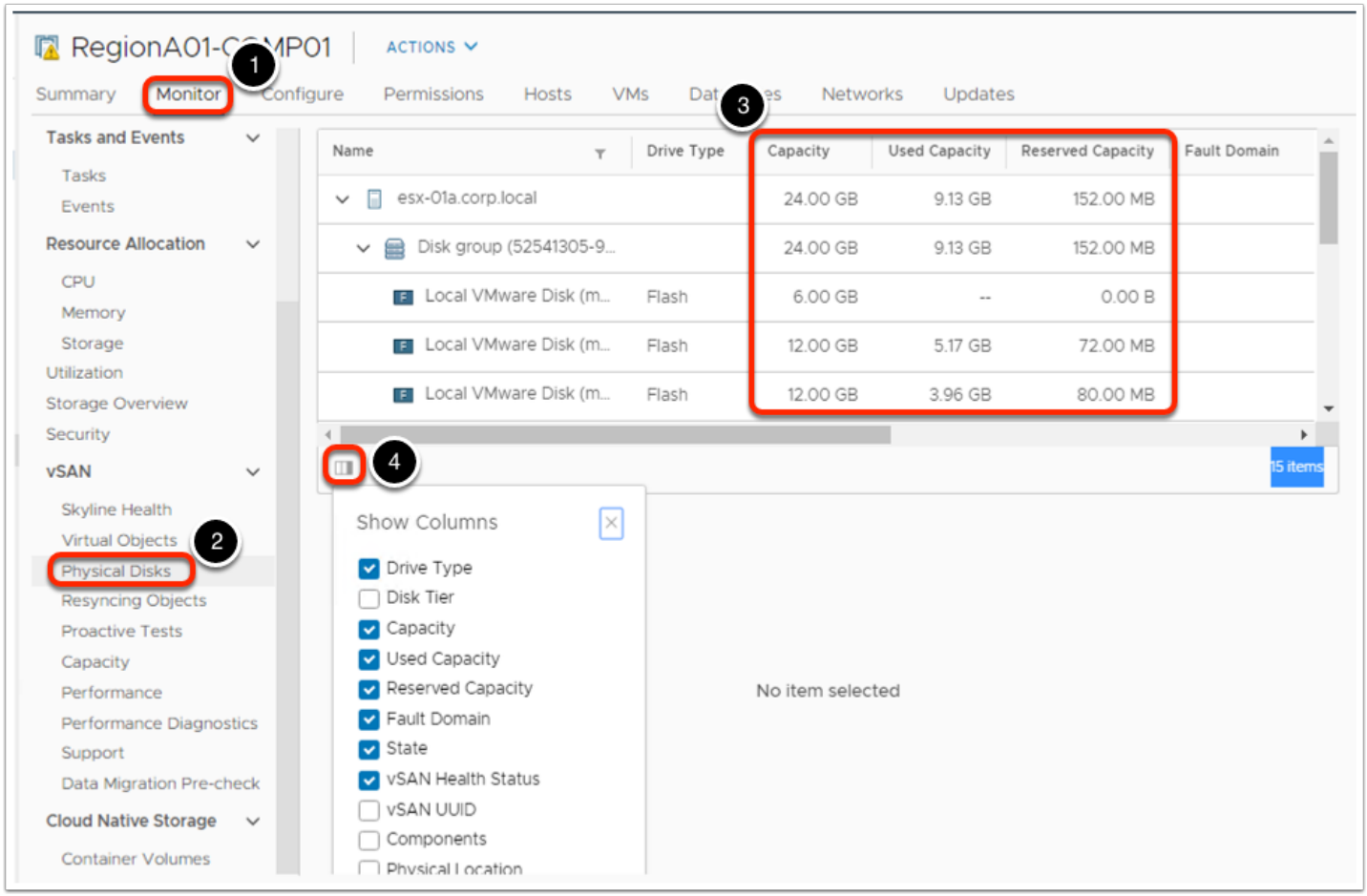
1. Scroll Down to view the **Usage breakdown before dedup and compression**

2. Click on **EXPAND ALL** (note - screenshot showing expanded)

These are all the different object types one might find on the vSAN Datastore. We have VMDKs, VM Home namespaces, and swap objects for virtual machines. We also have performance management objects when the vSAN performance logging service is enabled. There are also the overheads associated with on-disk format file system, and checksum overhead. Other (not shown) refers to objects such as templates and ISO images, and anything else that doesn't fit into a category above.

It's important to note that the percentages shown are based on the current amount of **Used** vSAN Datastore space. These percentages will change as more Virtual Machines are stored within vSAN (e.g. the **File system overhead %** will decrease, as one example).

Physical Disks Capacity



1. Make sure you are in the **Monitor** Tab
2. Select **vSAN > Physical Disks**
3. Note the **Capacity, Used Category**, and **Reserved Capacity** amounts
4. Click on the **Column** Icon, you can see see other options to view the details on those disks.

Here we can see the amount of Used Capacity per Physical Disk.

Monitoring vSAN Performance

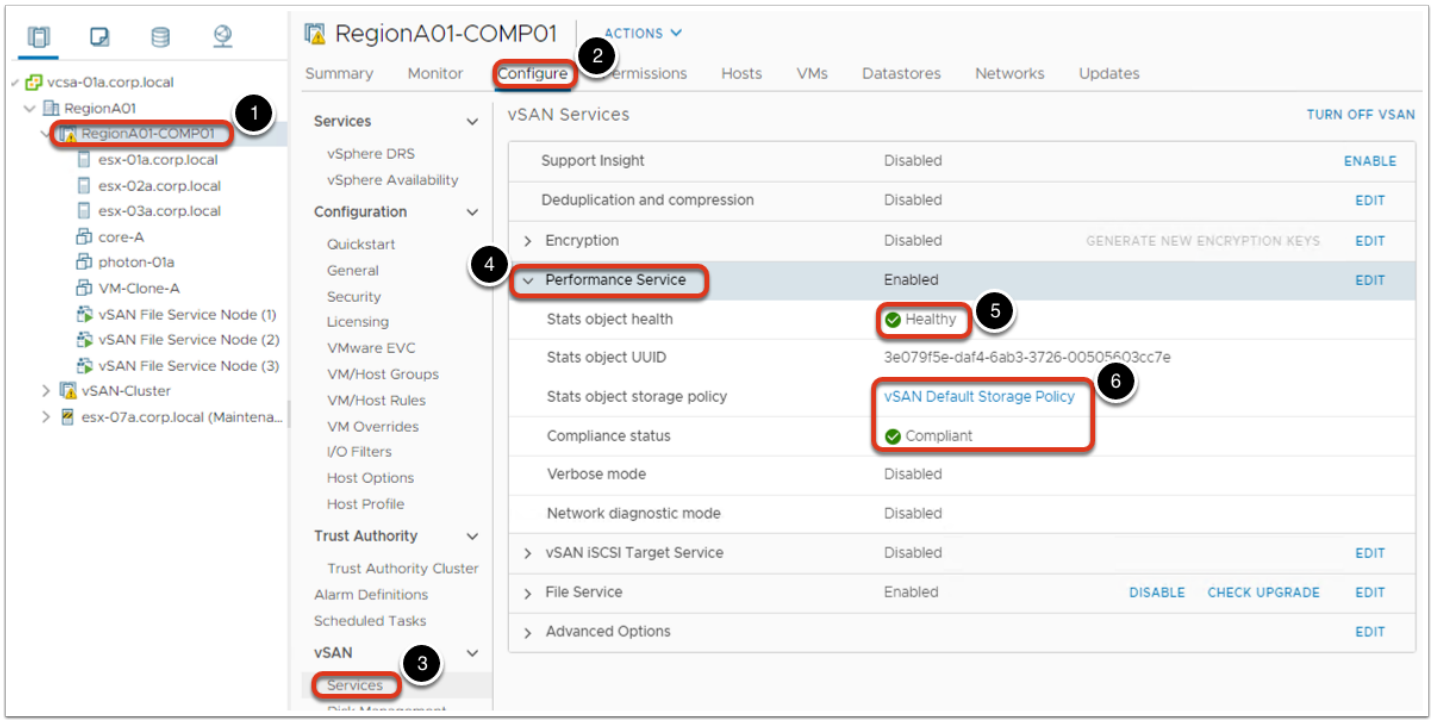
A healthy vSAN environment is one that is performing well. vSAN includes many graphs that provide performance information at the cluster, host, network adapter, virtual machine, and virtual disk levels. There are many data points that can be viewed such as IOPS, throughput, latency, packet loss rate, write buffer free percentage, cache de-stage rate, and congestion. Time range can be modified to show information from the last 1-24 hours or a custom date and time range. It is also possible to save performance data for later viewing.

Performance Service

With vSAN 7, the performance service is automatically enabled at the cluster level. The performance service is responsible for collecting and presenting Cluster, Host and Virtual Machine performance related metrics for vSAN powered environments. The performance service is integrated into ESXi, running on each host, and collects the data in a database, as an object on a vSAN Datastore. The performance service database is stored as a vSAN object independent of vCenter Server. A storage policy is assigned to the object to control space consumption and availability of that object. If it becomes unavailable, performance history for the cluster cannot be viewed until access to the object is restored.

Performance Metrics are stored for 90 days and are captured at 5 minute intervals.

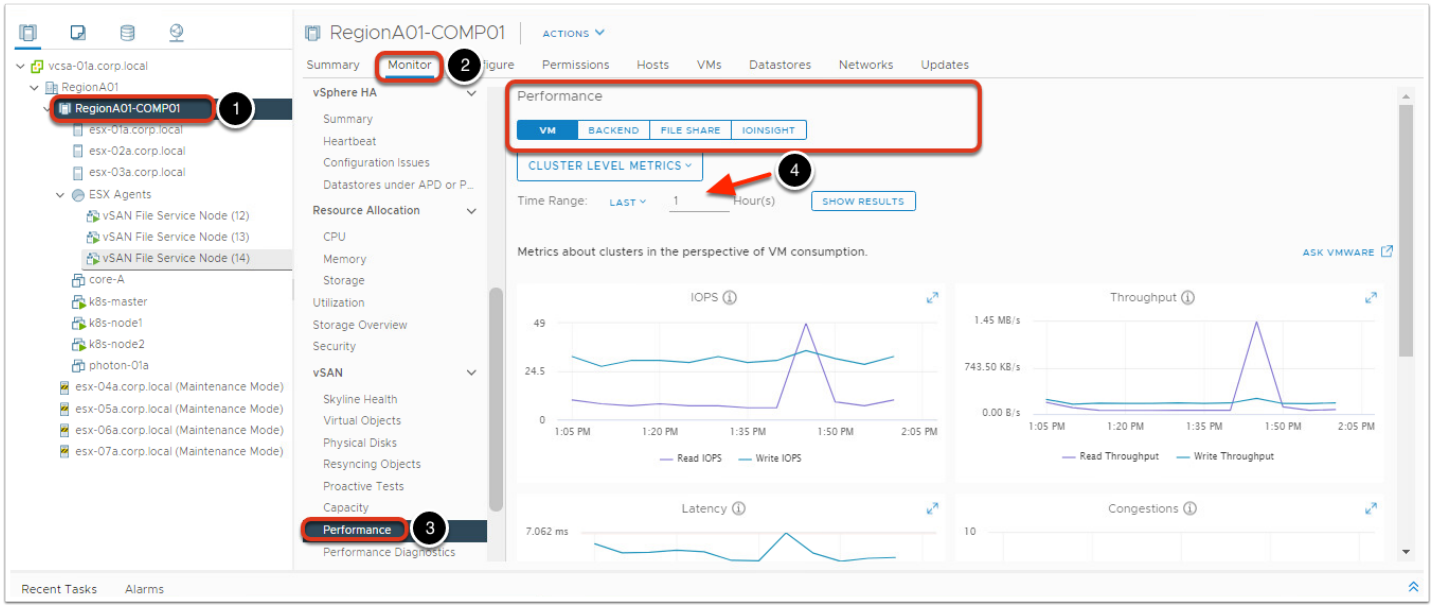
Validate Performance Service



1. Select **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > Services**
4. Expand **Performance Service**
5. Note that the Performance Stats Database Object is reported as **Healthy**
6. Note that the Stats DB is using the **vSAN Default Storage Policy** (RAID-1, Failures to Tolerate = 1) and is reporting **Compliant** status

Let's examine the various Performance views next at a Cluster, Host and Virtual Machine level.

Cluster Performance



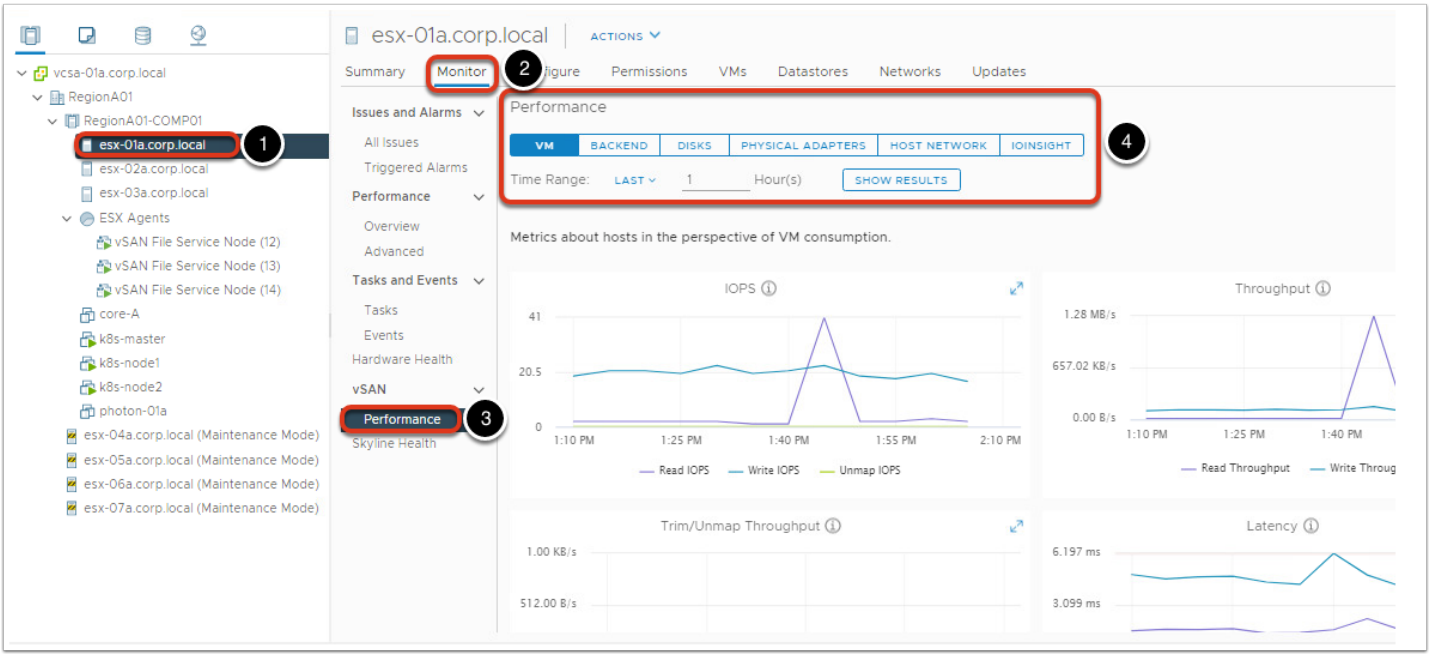
1. Select **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Performance**
4. Note that we can choose to view **VM**, **Backend, File Share** and **IOInsight** Performance views at the Cluster level (you can also customize the **Time Range** if desired).

Note: You might need to click on 'Show Results'

Scroll-down to view the various metrics that are collected (**IOPS**, **Throughput**, **Latency**, etc.)

“Front End” **VM** traffic is defined as the type of storage traffic being generated by the VMs themselves (the reads they are requesting, and the writes they are committing). “**Back End**” vSAN traffic accounts for replica traffic (I/Os in order to make the data redundant/highly available), and well as synchronization traffic. Both of these traffic types take place on the dedicated vSAN vmkernel interface(s) per vSphere Host.

Host Performance

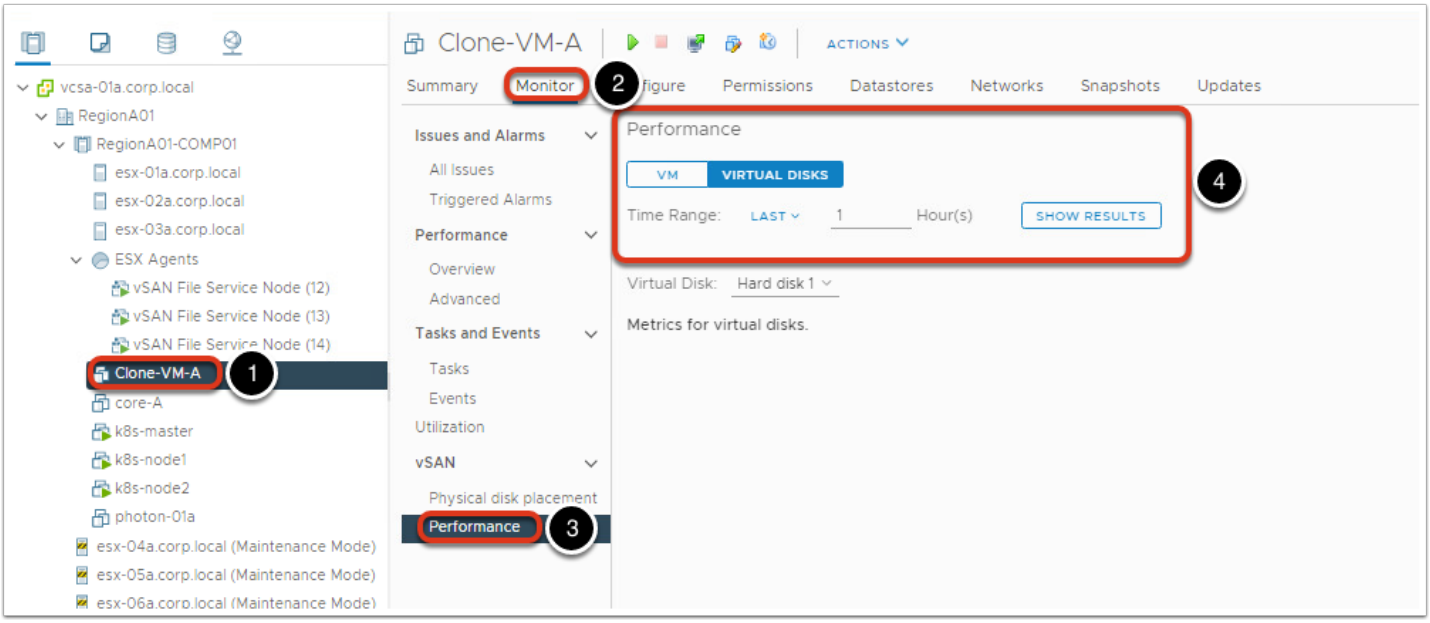


1. Select host, **esx-01a.corp.local**
2. Select **Monitor**
3. Select **vSAN > Performance**
4. Note that we can choose to view **VM**, **Backend, Disks, Physical Adapters, Host Network** and **IOInsight** Performance views at the Host level (you can also customize the **Time Range** if desired).

Scroll-down to view the various metrics that are collected (**IOPS**, **Throughput**, **Latency**, etc.)

In this view we can see more Performance related metrics at the Host level vs. Cluster. Feel free to examine the various categories indicated in Step 4 to get a feel for the information that is available.

Virtual Machine Performance



1. Select **Clone-VM-A**
2. Select **Monitor**
3. Select **vSAN > Performance**
4. Note that we can choose to view **VM** and **Virtual Disks** Performance views at the Virtual Machine level (you can also customize the **Time Range** if desired).

Scroll-down to view the various metrics that are collected (**IOPS**, **Throughput**, **Latency**, etc.)

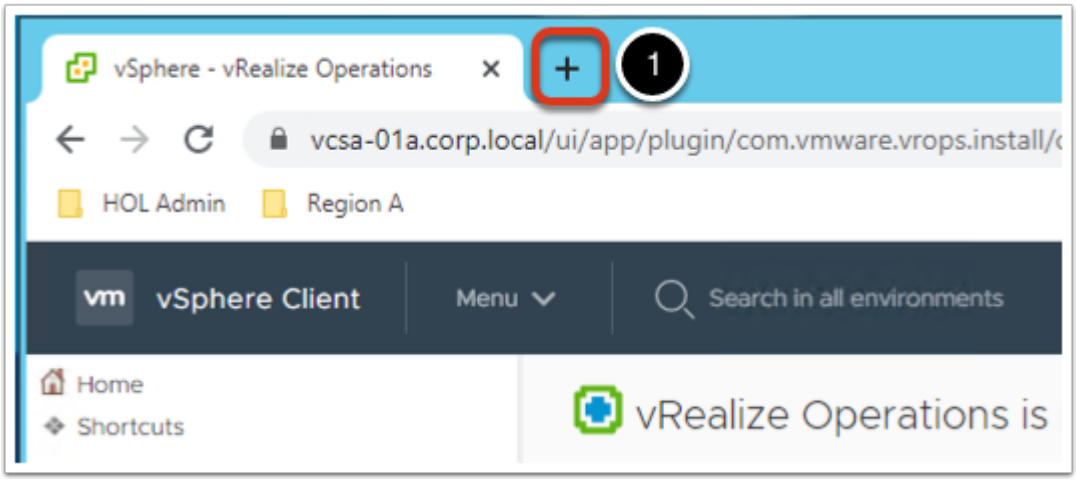
Next we will examine the vSAN information that is accessible via the new built-in vRealize Operations for vCenter Dashboards as well as within vRealize Operations itself.

Enabling vROPS for integration with vCenter

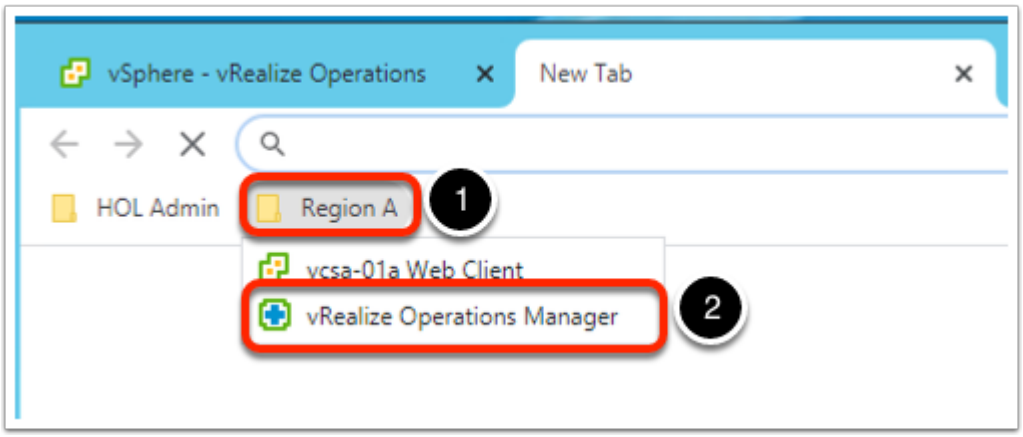
It takes approximately 30 minutes to enable vRealize Operations for vCenter within our Lab environment.

In these next steps, we will ask you to walk through the steps necessary to enable vRealize Operations for vCenter.

Login to vRealize Operations Client

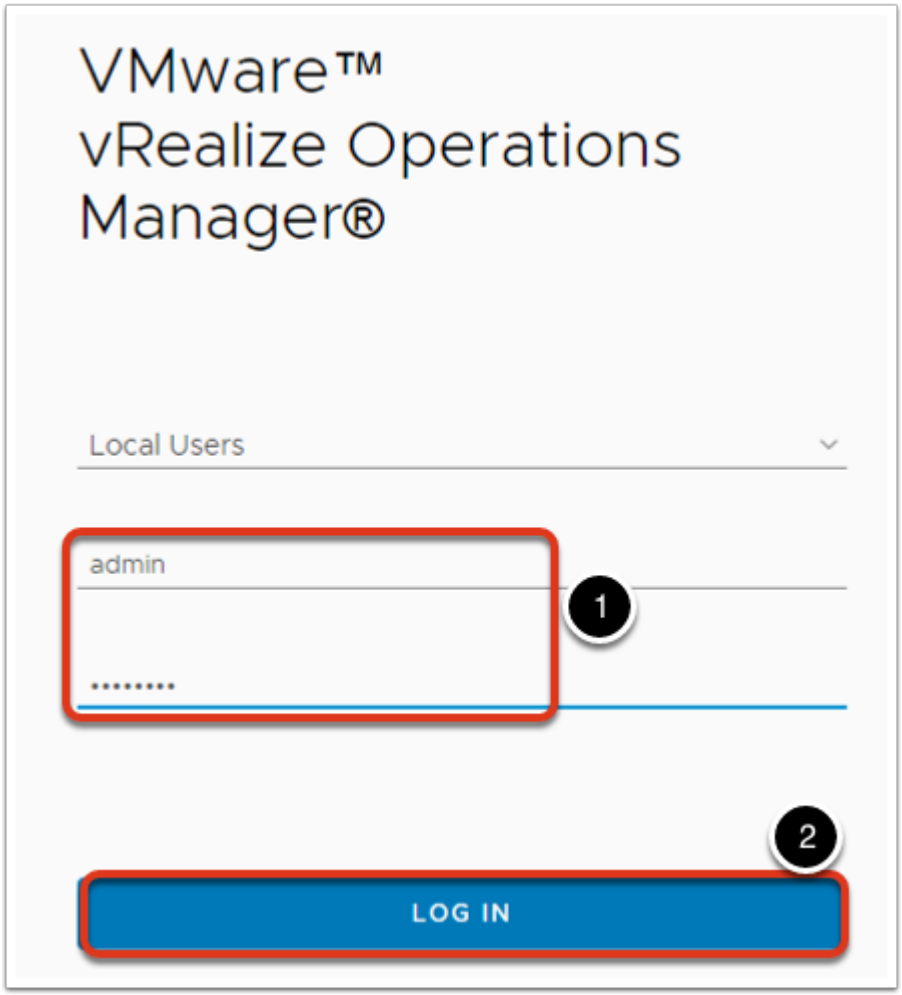


1. Open a new Tab from the Chrome Browser



1. Select **Region A**
2. Then choose **vRealize Operations Manager**

Login to vRealize Operations Manager

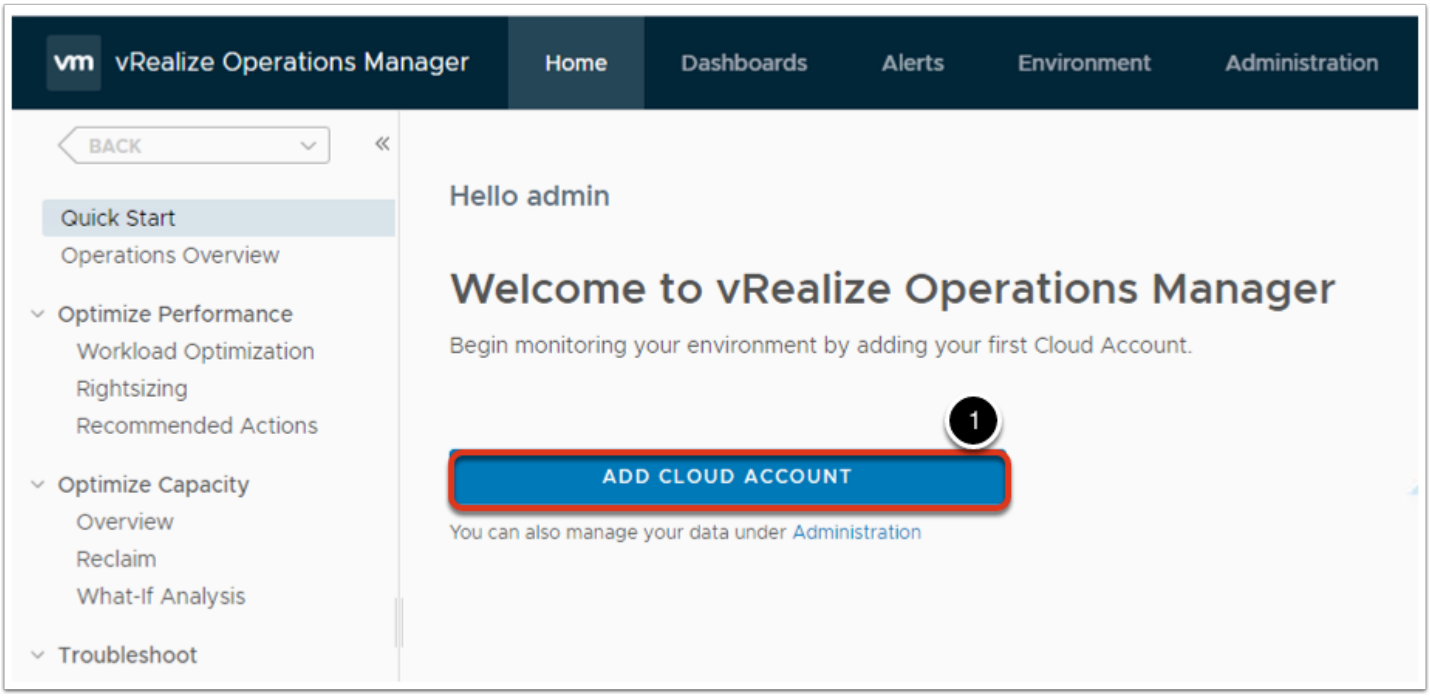


1. Log into **VMware vRealize Operations Manager**

```
User name: admin
Password: VMware1!
```

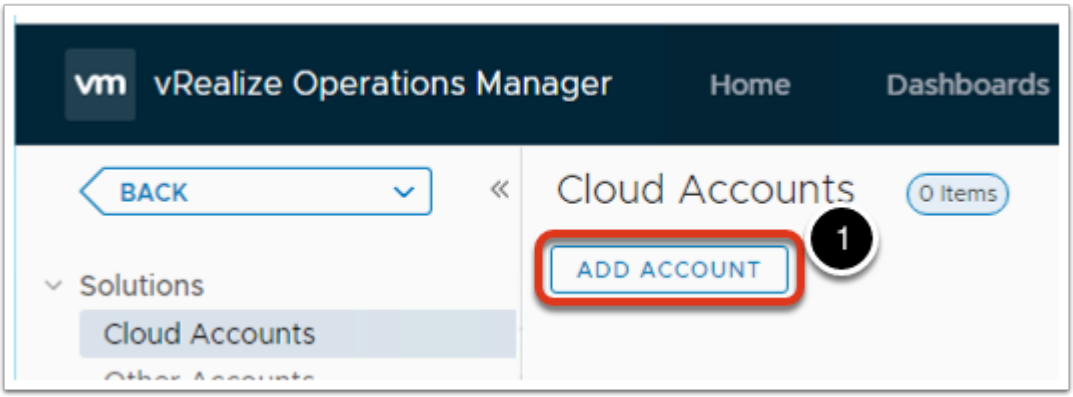
2. Click **LOG IN**

Configure vRealize Operations Cloud



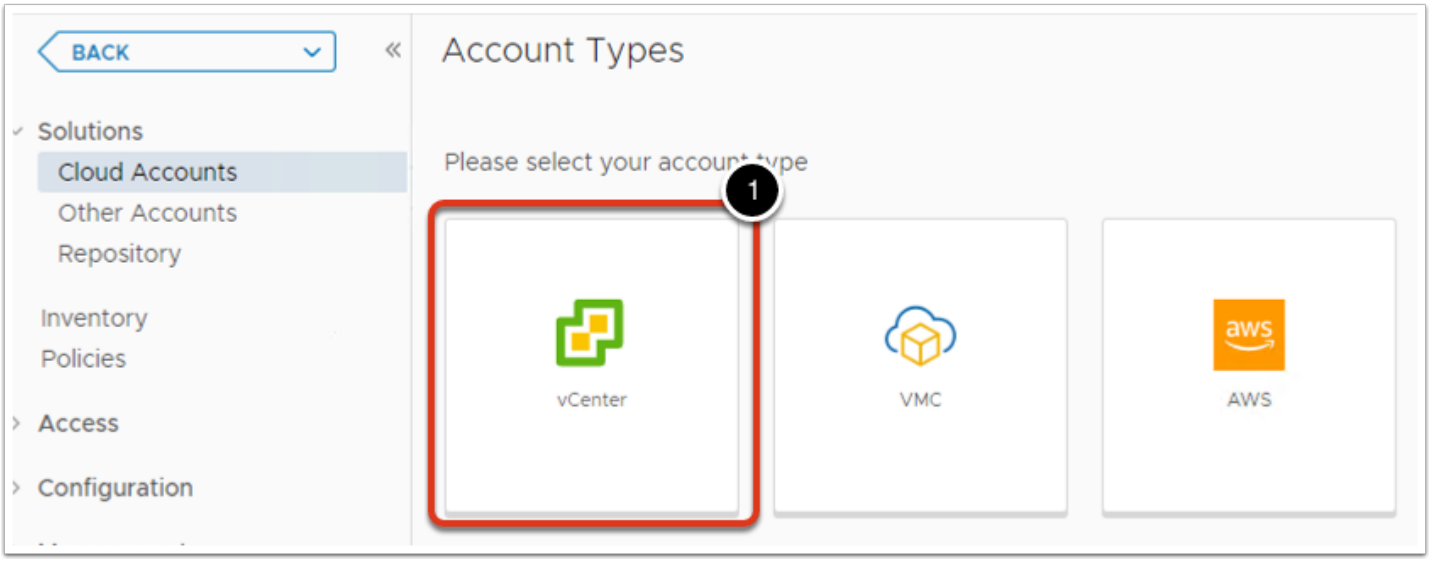
1. Select **ADD CLOUD ACCOUNT**

Configure vRealize Operations Cloud



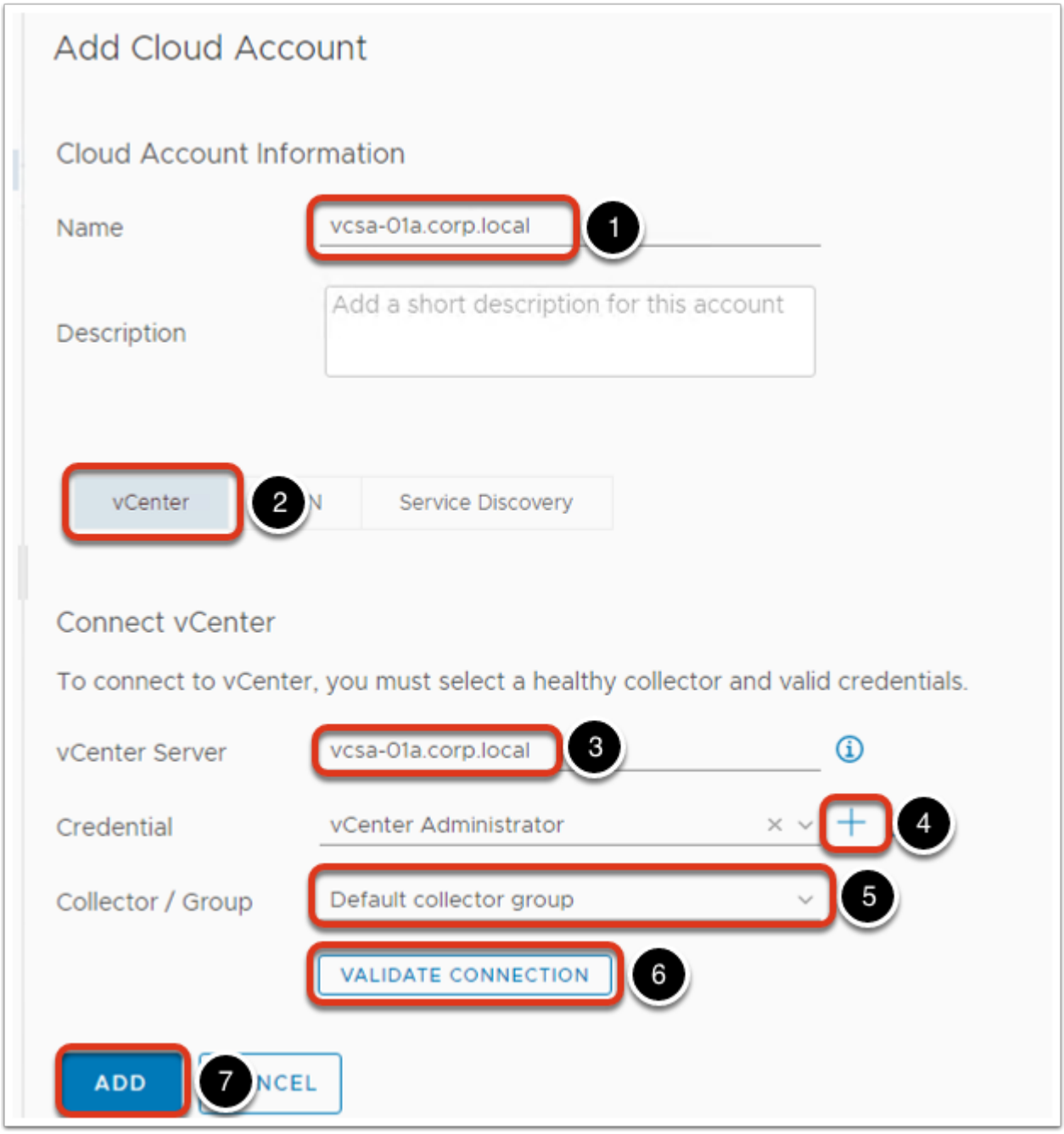
1. Select **ADD ACCOUNT**

Configure vRealize Operations Cloud



1. Select **vCENTER**

Add Cloud Account



1. Enter **Cloud Account Information**

```
Name: vcsa-01a.corp.local
```

2. Select **vCenter**
3. Enter Connect vCenter

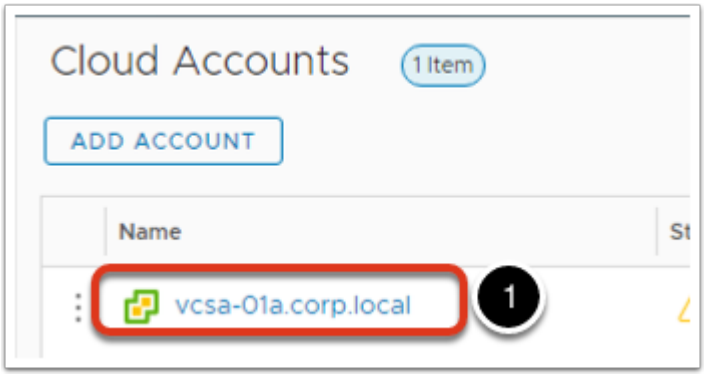
```
vCenter Server: vcsa-01a.corp.local
```

4. Click on **+** (plus) icon then enter the **Credential** (not picture)

```
Manage Credential

Credential Name: vCenter Administrator
User Name: administrator@corp.local
Password: VMware1!
Actions User Name: leave blank
Actions Password: leave blank
Then Click OK
```

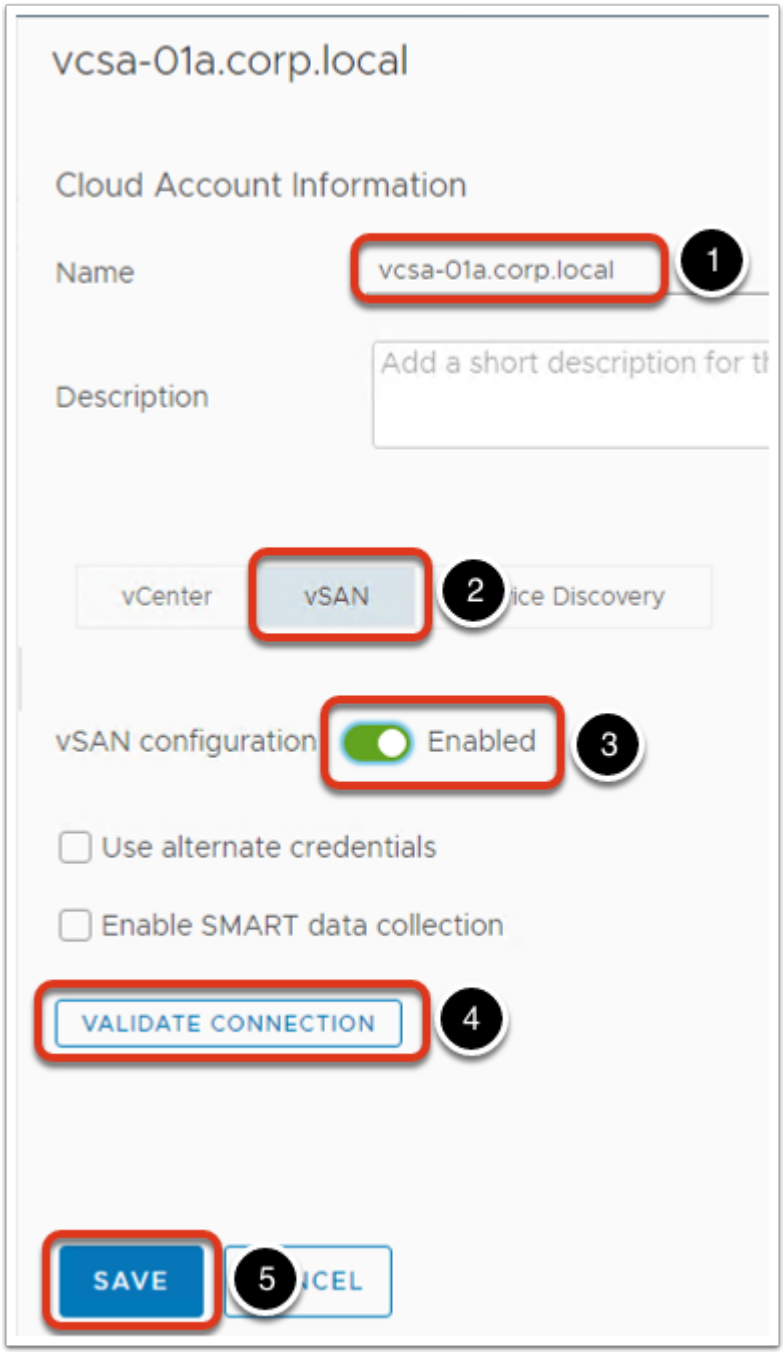
5. Select **Default collector group** from the drop-down menu of **Collector/Group**
6. Click **VALIDATE CONNECTION** and let it connect (not picture) if it does not connect make sure you enter the correct information
7. Click **ADD**

Configure vRealize Operations Cloud for vSAN



1. Select **vcsa-01a.corp.local**

Configure vRealize Operations Cloud for vSAN (vcsa-01a.corp.local)



1. Enter **Cloud Account Information**

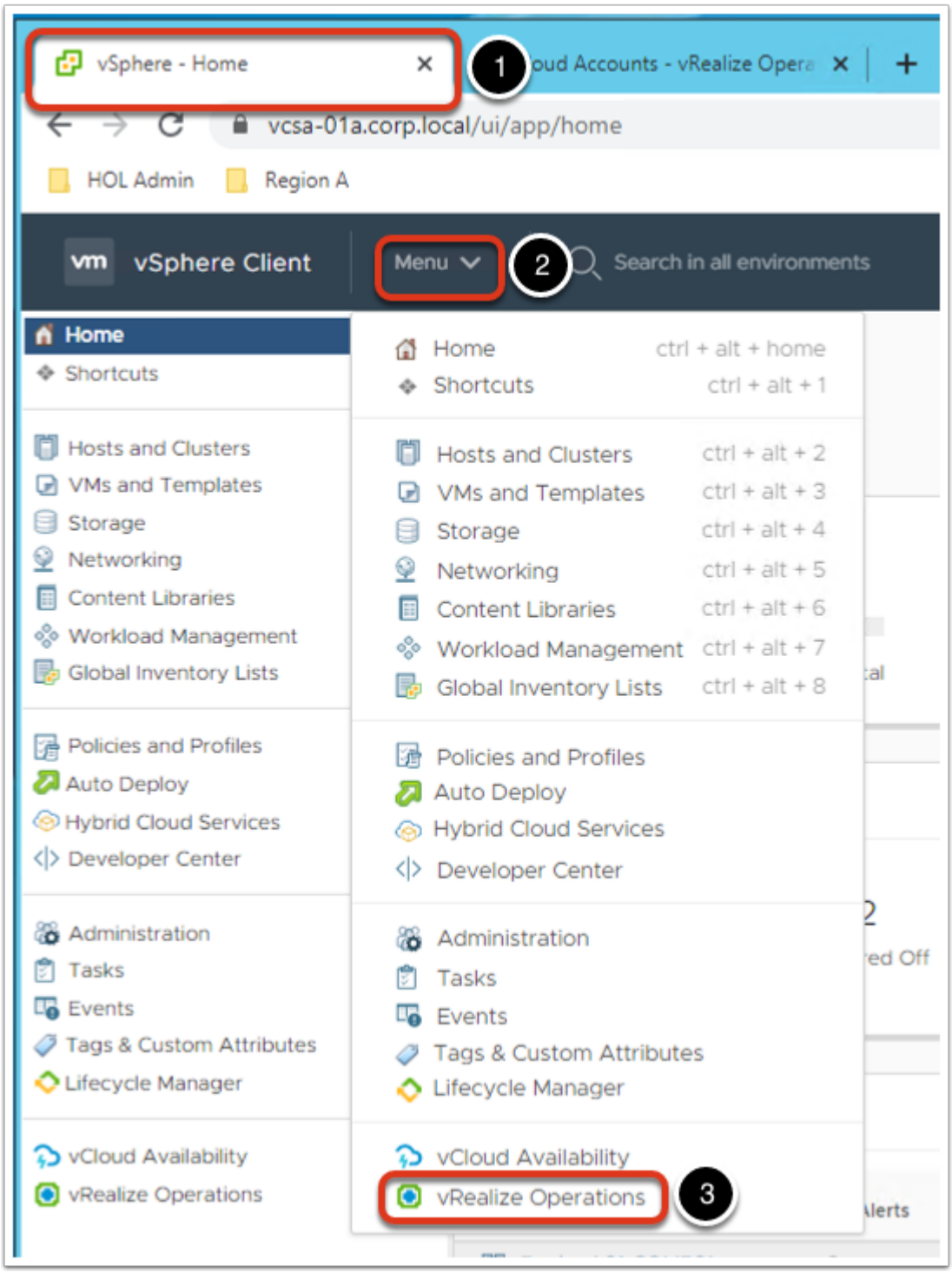
```
Name: vcsa-01a.corp.local
```

2. Select **vSAN**
3. Enable the vSAN configuration

4. Click **VALIDATE CONNECTION**
5. Click **SAVE**

- Click **SAVE** again (not picture)

vRealize Operation within vCENTER



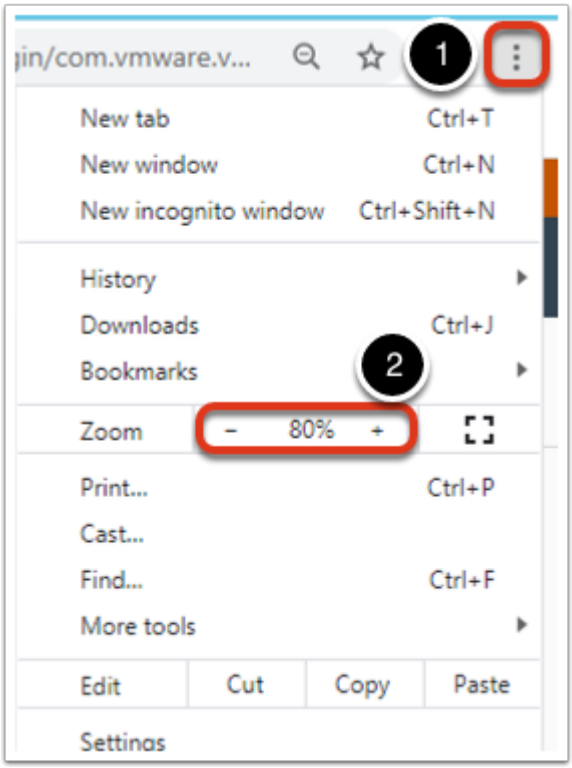
1. Switch back to the **vSphere - Home** tab on Chrome

2. Select **Menu**
3. Select **vRealize Operations**

vRealize Operations within vCenter Monitoring

vSphere and vSAN 7 now includes vRealize Operations within vCenter . This new feature allows vSphere customers to see a subset of intelligence offered up by vRealize Operations (vROps) through a single vCenter user interface. Light-weight purpose-built dashboards are included for both vSphere and vSAN. It is easy to deploy, provides multi-cluster visibility, and does not require any additional licensing.

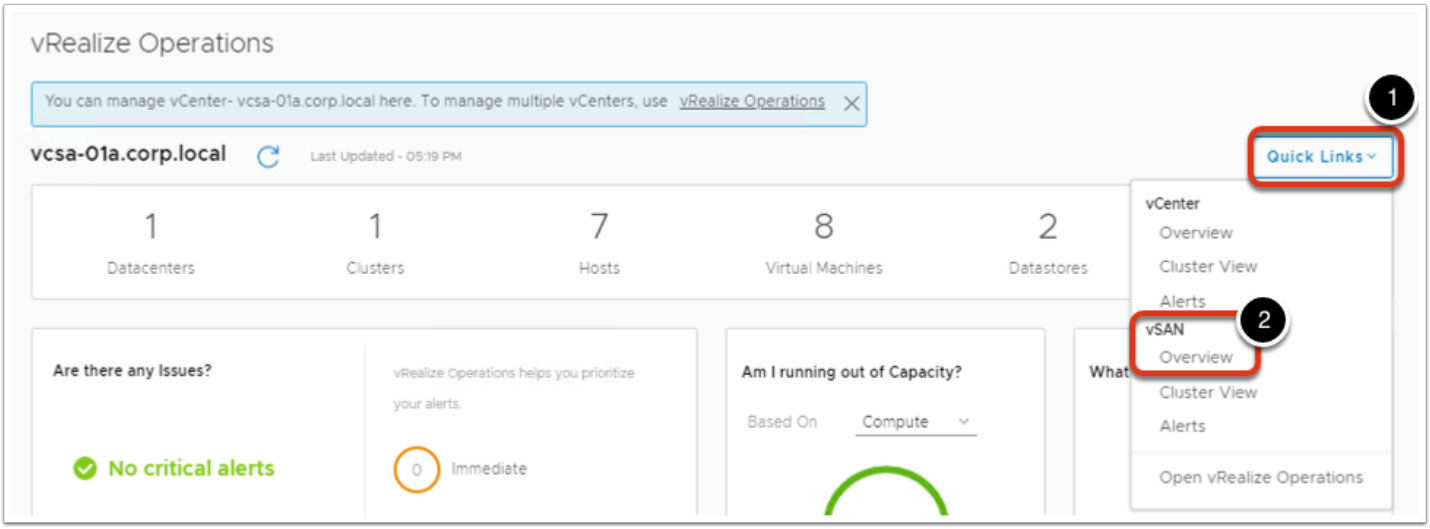
Chrome Browser Zoom



In our VMware Learning Platform (Lab) environment we have a limited amount of screen real estate (1024x768). Let's reduce our Chrome Browser Zoom so that we can view more on screen:

1. Select the **Vertical Ellipses** in the upper-right hand corner of your Chrome Browser
2. Click the **'-'** sign to reduce your **Zoom** to **80%**

Integrated Dashboards



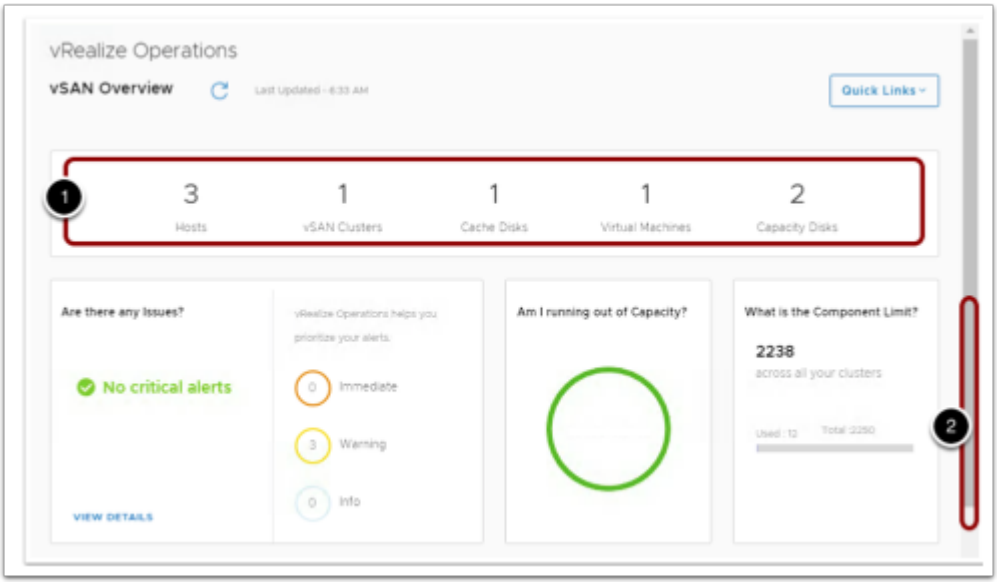
There are three dashboards for vSphere/vCenter, and three dashboards built specifically for vSAN. These dashboards do not replace the dashboards found in the full vR Ops product, but place a subset of the most important information directly within vCenter, for a single, unified pane of visibility. These dashboards contain widgets designed to maintain clarity and simplicity, and unlike the full vR Ops UI, will have a minimal amount of customization available. The vCenter Overview dashboard gives an aggregate view of the activity and status of your clusters managed across vCenter.

Let's examine the vSAN Dashboards:

1. Select **Quick Links**
2. Click **vSAN > Overview**

NOTE: If you receive messages like, "You do not have any vSAN Clusters" or "Unfortunately, you have no Clusters configured!" this is due to vROps for vCenter not being fully configured yet and you will need to wait longer for this to complete. Thank you for your patience.

vSAN Overview

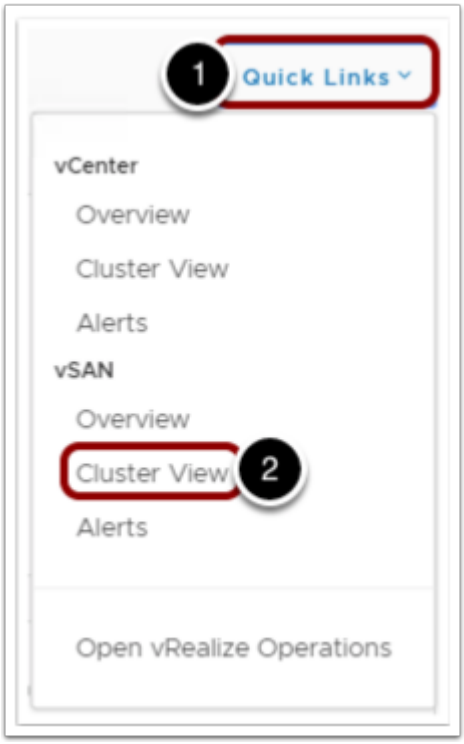


The vSAN Overview dashboard gives an aggregate view of the activity and status of your clusters, but only for those running vSAN. Administrators can view rollup statistics for hosts, VMs, alerts, capacity, performance metrics, and more.

1. Note that information for all vSAN Clusters is **aggregated** in the top display panel
2. Scroll-down to examine the additional Dashboard information presented

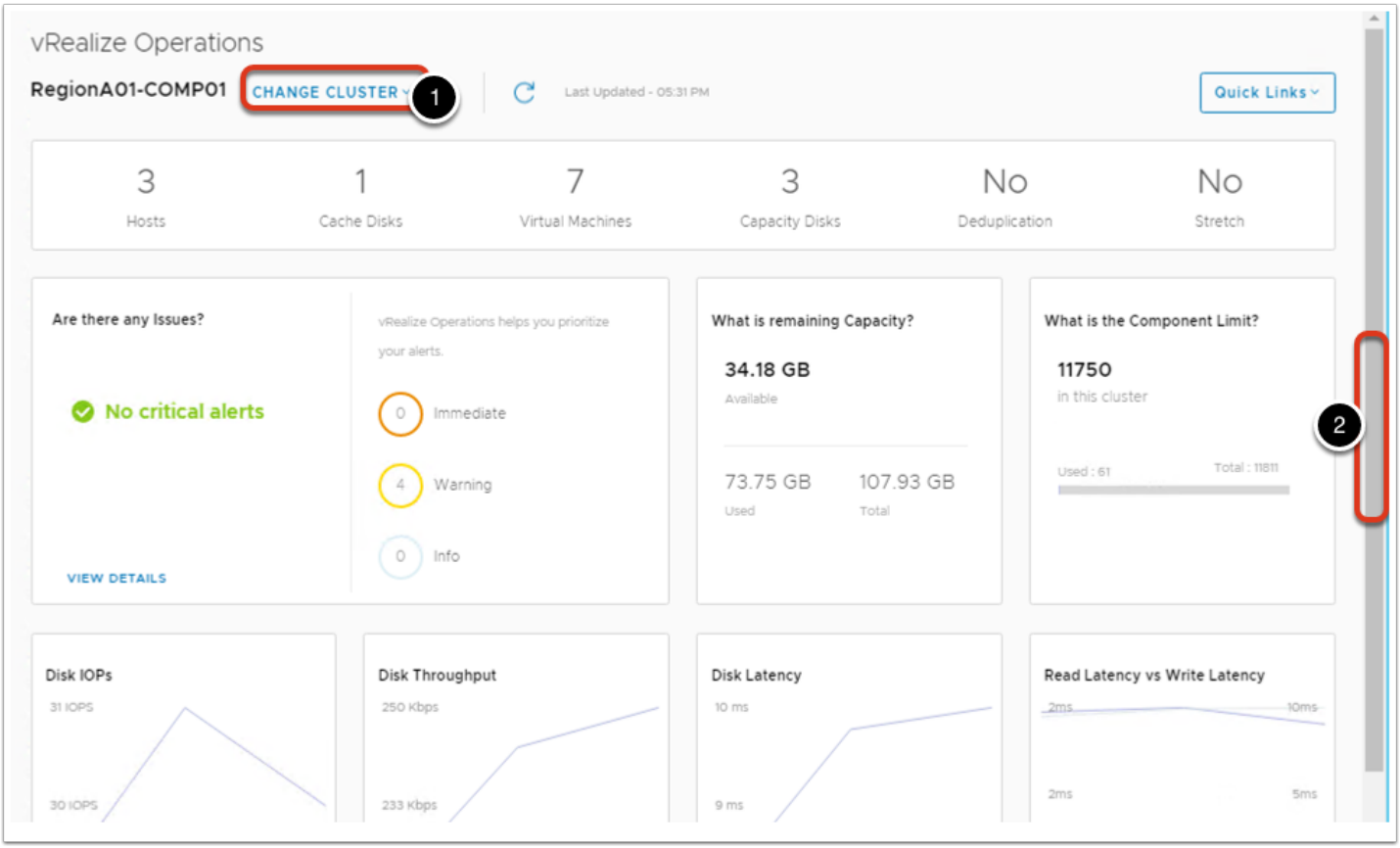
We will examine the **Cluster View** Dashboard next

Quick Links



1. Select **Quick Links**
2. Click **vSAN > Cluster View**

vSAN Cluster View

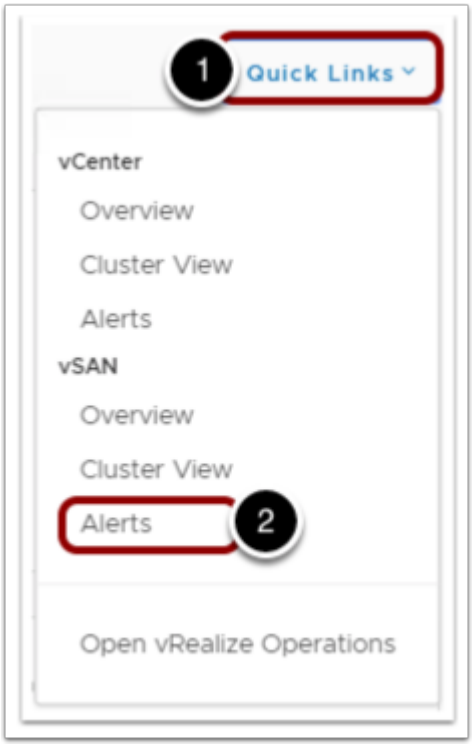


The vSAN Cluster View dashboard provides more details specific to the selected vSAN cluster.

1. Note that you choose additional vSAN Clusters via the **Change Cluster** drop-down menu (our Lab environment only contains a single vSAN Cluster)
2. Scroll-down to review vSAN related metrics such as **Remaining Capacity, Component Limits, Disk IOPS, Disk Throughput**, and **Read vs Write Latency** for the selected cluster

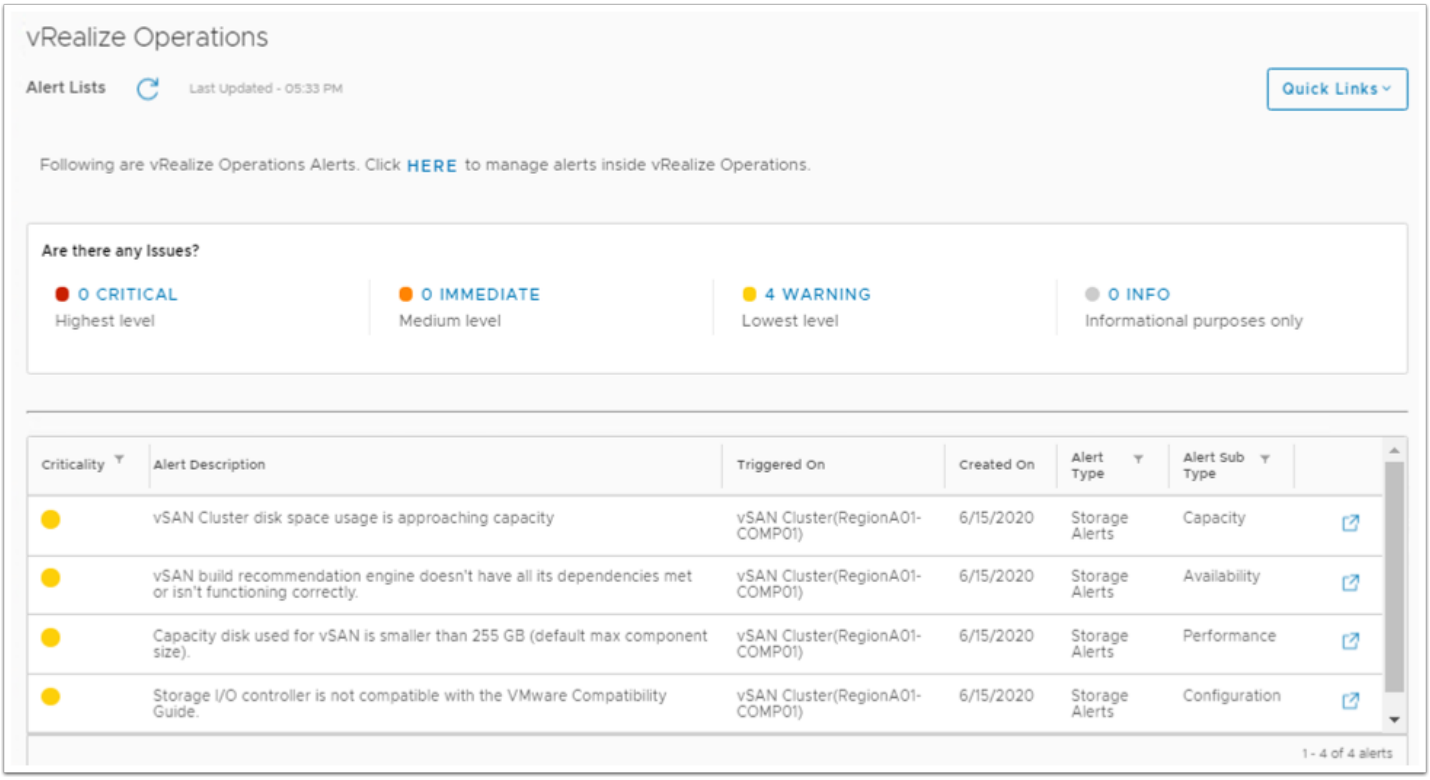
Let's examine our final vRealize Operations within vCenter vSAN related Dashboard.

Quick Links



1. Select **Quick Links**
2. Click **vSAN > Alerts**

Alert Lists

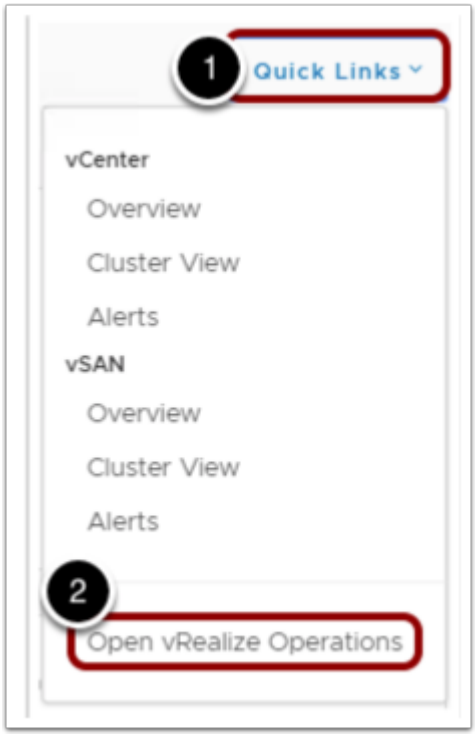


1. The Alerts List will surface Critical, Immediate, Warning and Info Alerts which can be examined in further detail if desired.

Note: Issues in your Lab may differ from those shown in the screenshot.

For our final lesson within this module, we will log directly into vRealize Operations to examine the vSAN related Dashboards that are available.

Quick Links



1. Select **Quick Links**
2. Click **Open vRealize Operations**

Login

1. Enter Parameters:

```
admin
VMware1!
```

2. Click **LOG IN**

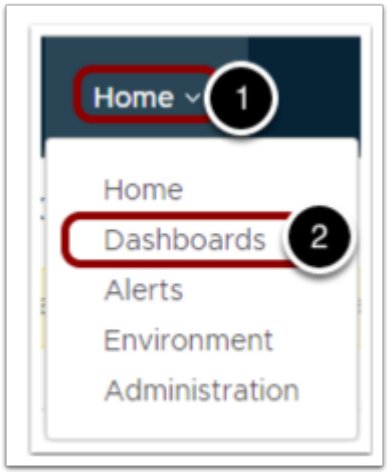
vRealize Operations Overview

vSAN integration is now fully built into vRealize Operations 6.6 and later, which means the same level of monitoring and analytics for vSphere is easily extended to vSAN. The APIs in vSAN were significantly enhanced to allow vROPs to fetch data directly from vSAN. This results in more detailed information for vR Ops to analyze, and make visible. Out of the box, vROPs provides the following:

- Four, prebuilt vSAN dashboards with multi-cluster visibility and analytics.
- Dashboards display vSAN and non-vSAN metrics together to show critical correlation across various resources.
- Native integration into vR Ops means that no additional Management Packs need to be installed.
- Dashboards can be cloned, and are fully customizable.

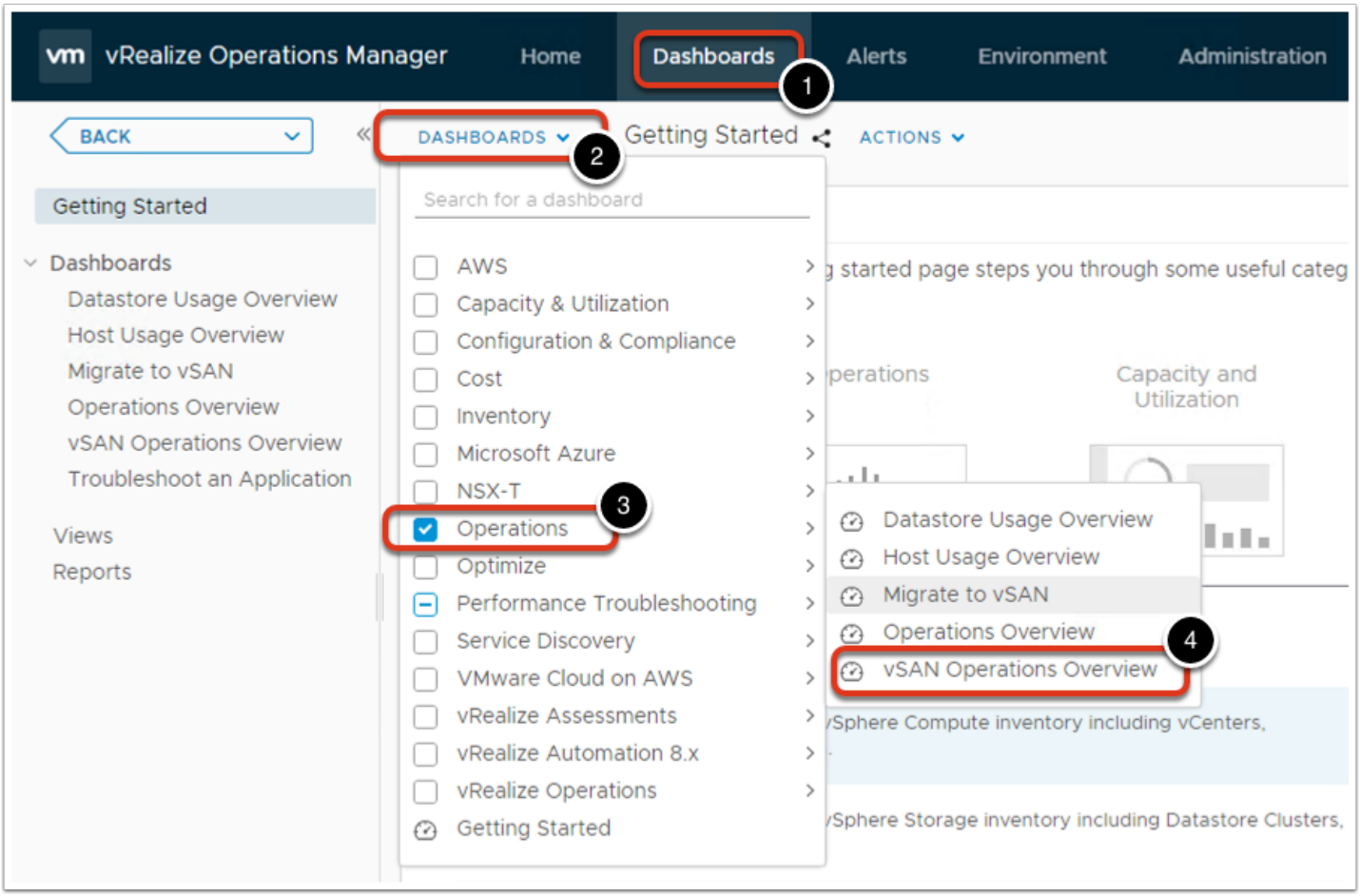
vRealize Operations uses vSAN's enhanced set of APIs to fetch data collected by the vSAN health and performance service. The vSAN health and performance service was introduced in vSAN 6.2, and provides a way for vSAN administrators to look at basic performance metrics of vSAN directly in vCenter. Unlike other metrics, vSAN performance metrics are not stored in vCenter. It is housed as an object that resides on the vSAN datastore. With each subsequent release of vSAN, additional metrics have been exposed in the performance service. However, the metrics in the performance services are not customizable, and have a limited window in which data can be viewed (1 hour to 24 hours), and a limited retention time (90 days). vR Ops fetches this vSAN performance data, and provides the user with much more flexibility in manipulation and retention of the data. vR Ops requires that the vSAN health and performance service be enabled to properly collect vSAN related metrics.

Dashboards



1. Select the **Home** drop-down menu
2. Click **Dashboards**

All Dashboards

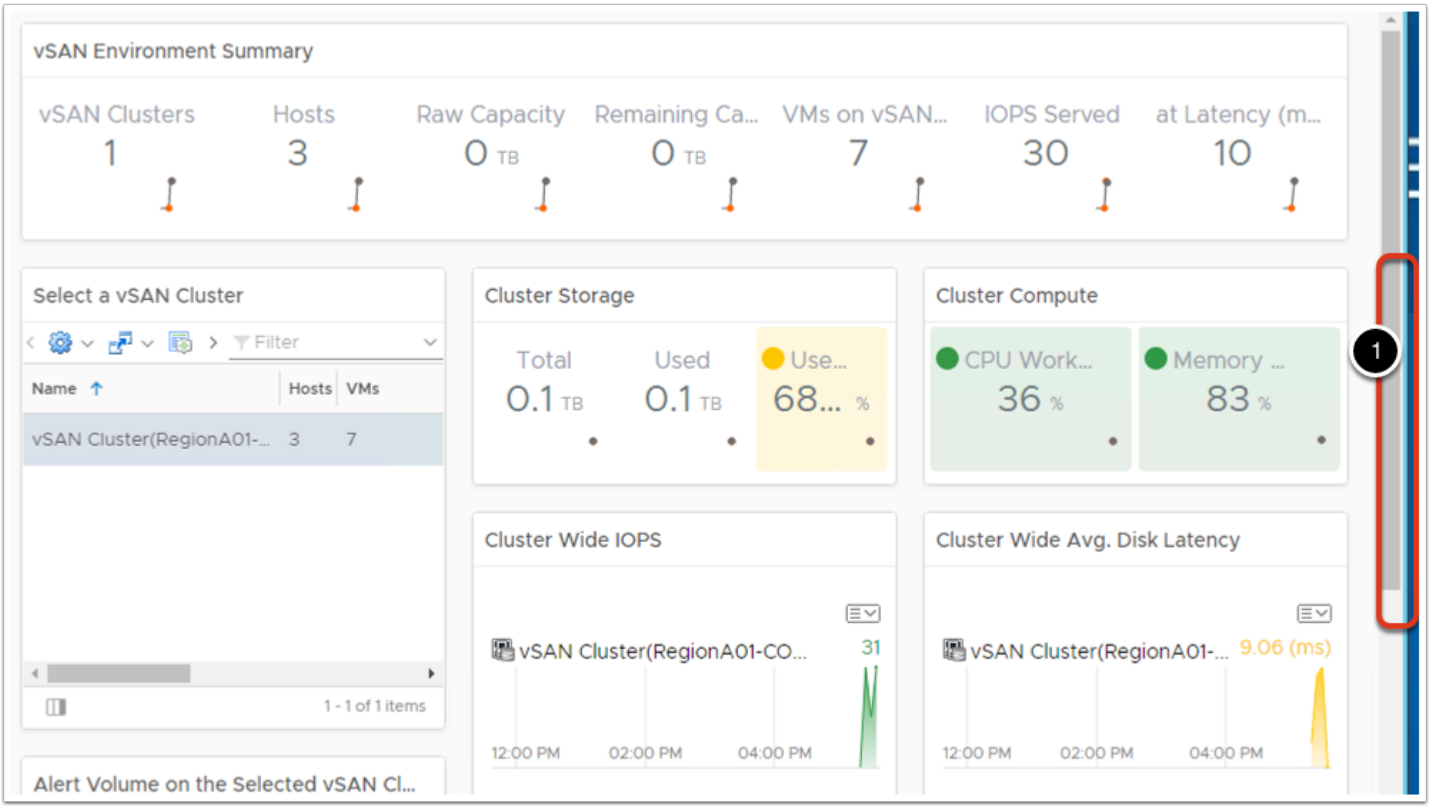


vRealize Operations handily groups out of box dashboards by activity type including **Operations**, **Capacity & Utilization** and **Performance Troubleshooting.**

We will examine vSAN Operations first:

1. Select **Dashboards**
2. Select the **All Dashboards** drop-down menu
3. Hover over **Operations**
4. Click **vSAN Operations Overview**

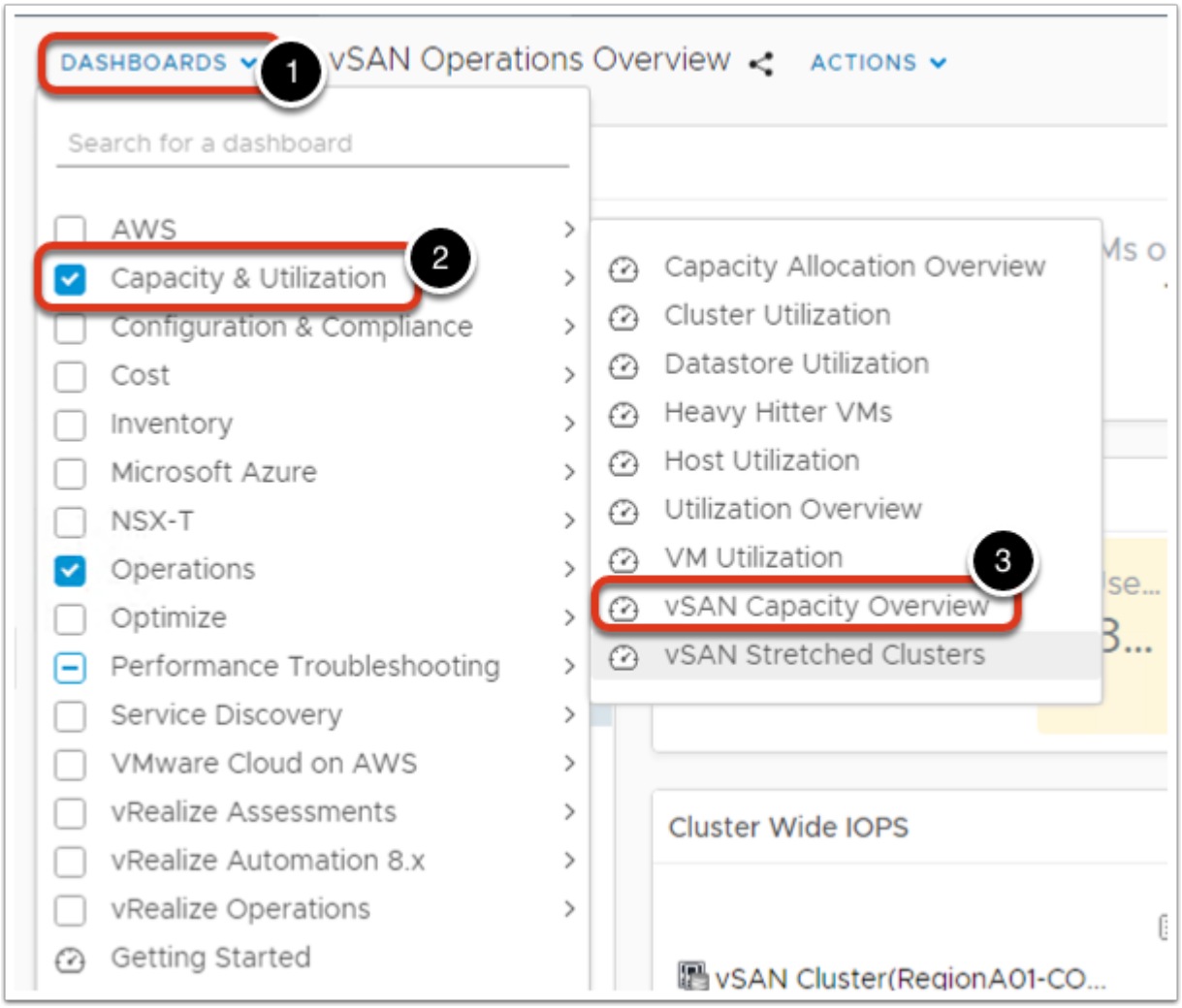
vSAN Operations Overview



The vSAN Operations Overview dashboard aims to provide a broad overview of the status of one or more vSAN powered clusters in an environment. This dashboard allows an administrator to see aggregate cluster statistics, along with cluster specific measurements. Not only does this dashboard touch on some of the key indicators of storage such as IOPS, throughput, and latency, it also provides other measurements that contribute to the health and well-being of the cluster, such as the host count, CPU and Memory utilization, and alert volume.

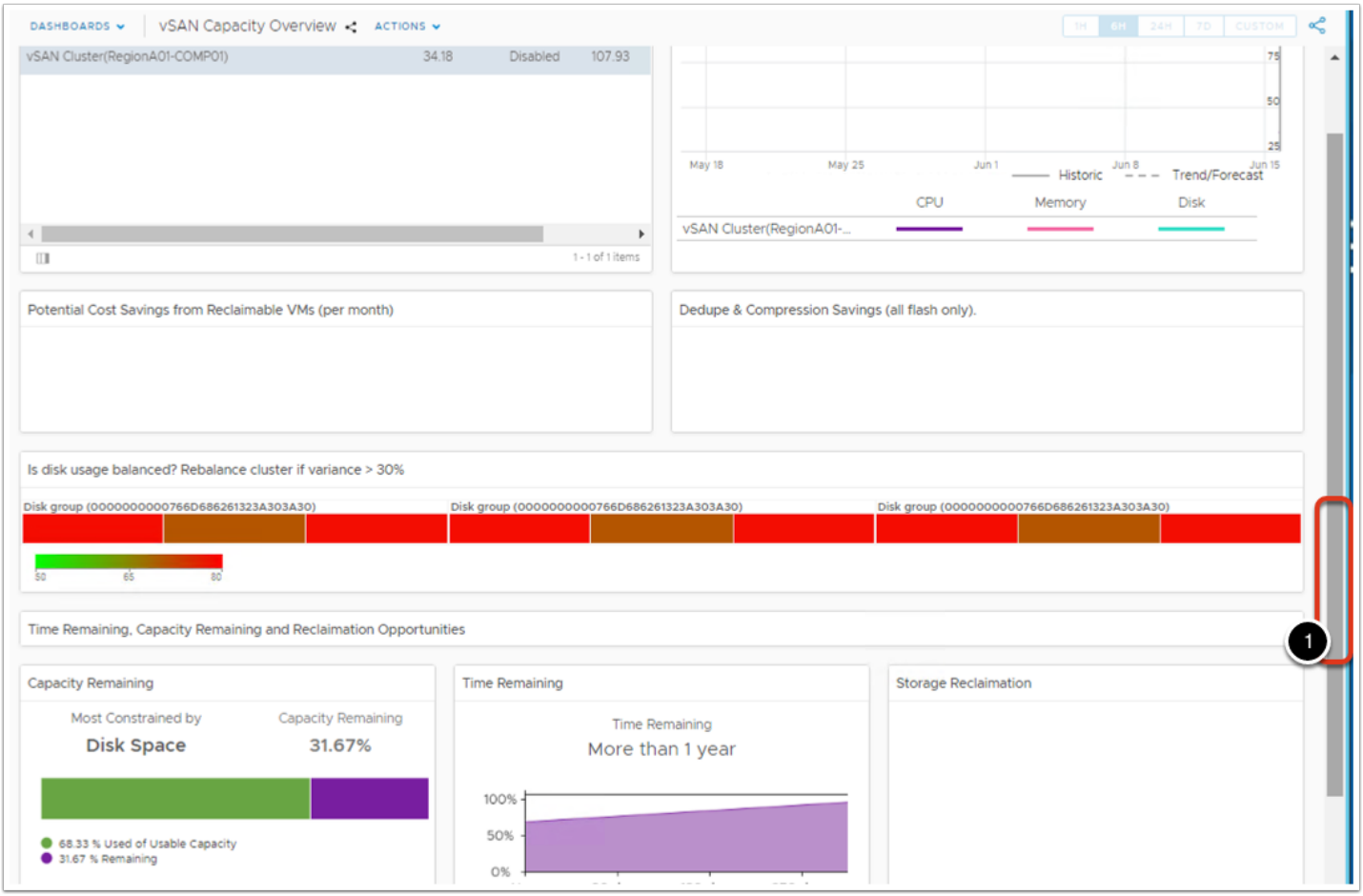
1. **Scroll-down** to view more information

All Dashboards



1. Select the **All Dashboards** drop-down menu
2. Hover over **Capacity & Utilization**
3. Click **vSAN Capacity Overview**

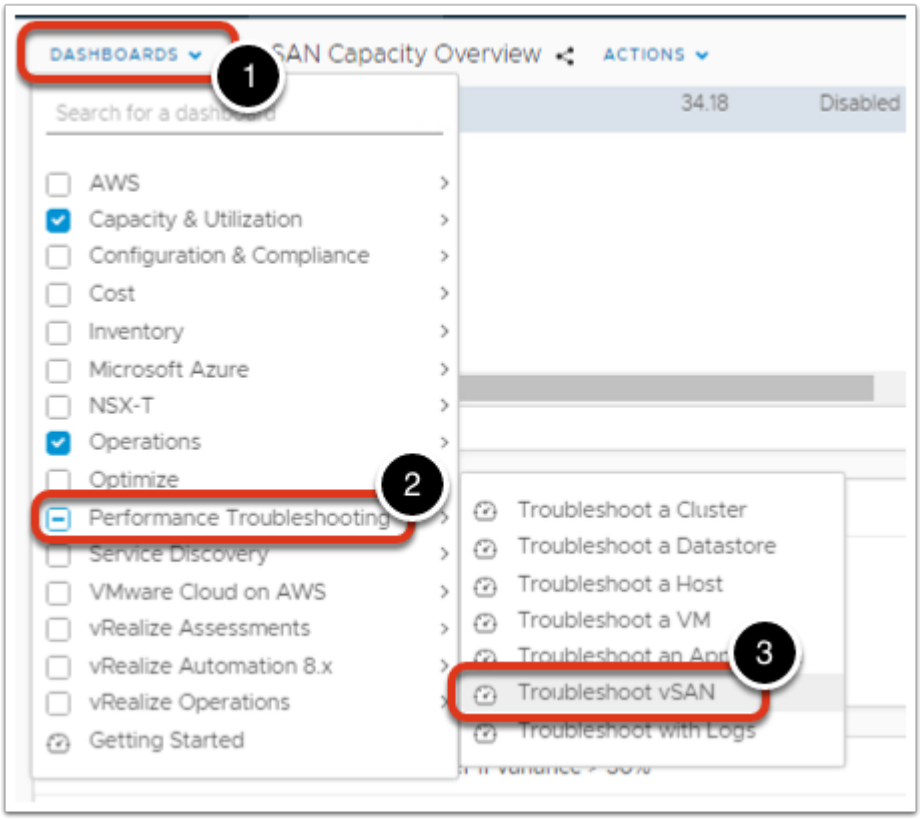
vSAN Capacity Overview



The vSAN Capacity Overview dashboard provides a wealth of vSAN capacity information not available in the point-in-time storage capacity statistics found in vCenter. This dashboard takes advantage of vROps' ability to capture capacity utilization over a period of time, which offers extensive insight into past trends of capacity usage. Capacity is about more than just storage resource usage. It is about CPU and memory capacity as well. This dashboard gives a window in to remaining CPU and memory capacity for a vSAN cluster. This data, paired with the storage utilization data will give an administrator a better understanding if scaling up (adding more storage to each host) or scaling out (adding more hosts) will be the best approach for an environment.

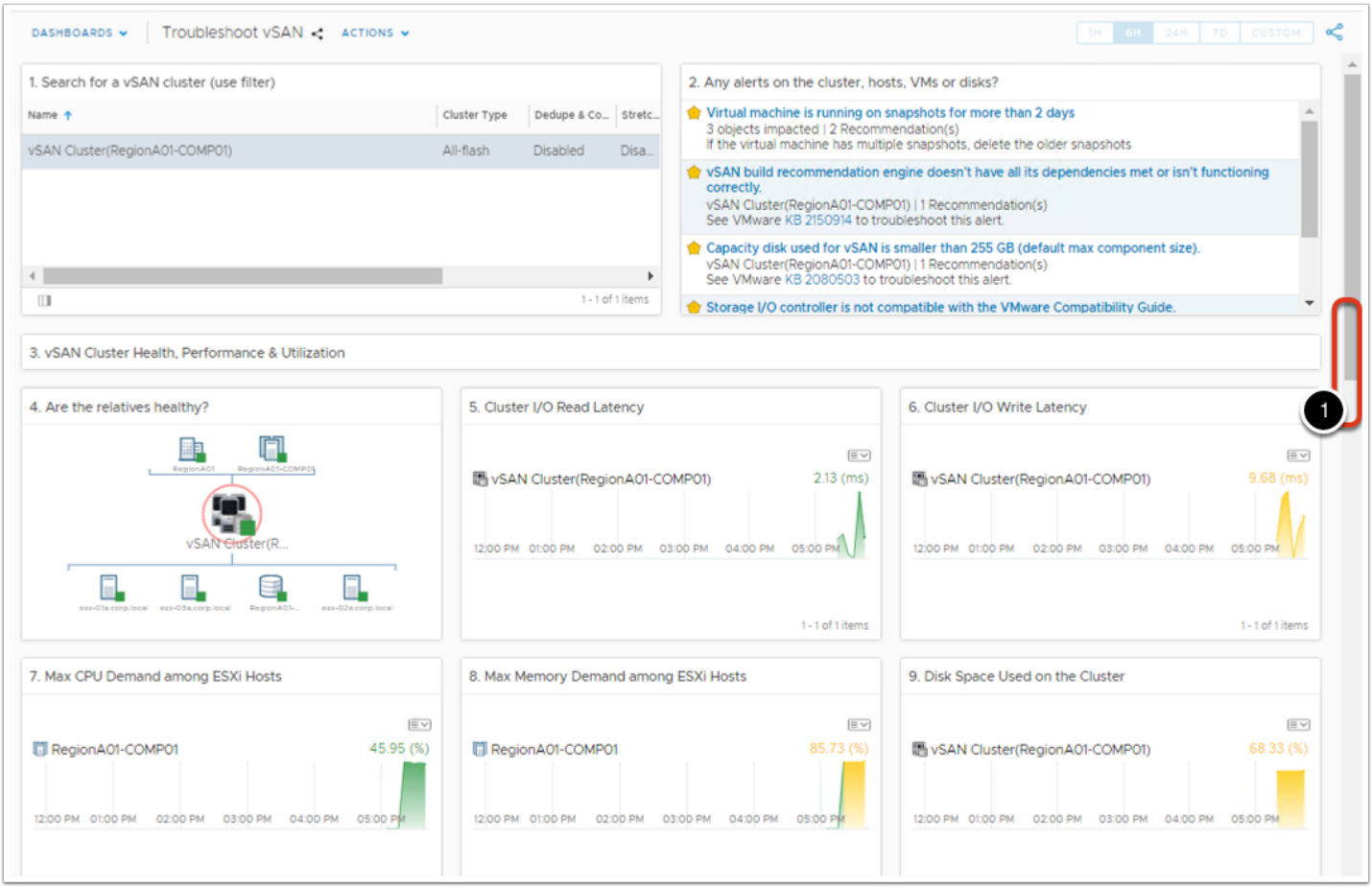
1. **Scroll-down** to view more information

All Dashboards



1. Select the **All Dashboards** drop-down menu
2. Hover over **Performance Troubleshooting**
3. Click **Troubleshoot vSAN**

Troubleshoot vSAN

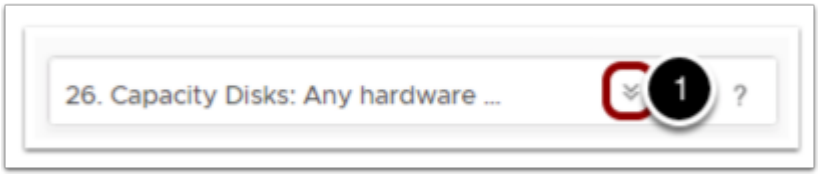


The Troubleshoot vSAN dashboard assembles a collection of alerts, metrics, and trending results to help determine the source of what changed in an environment, and when the change occurred. It assembles them in a systematic, layered approach to assist with troubleshooting and root cause analysis of an environment.

The dashboard begins with widgets showing any active alerts for the selected cluster, and identifies the hosts contributing to the alerts. Also displayed are key performance indicators at the cluster level. Highlighting the desired cluster will expose trending of cluster related resources (CPU Workload, Memory workload, Capacity remaining, etc.) over the past 12 hours. Widgets for VM read and write latency show a history of storage performance for the past 24 hours.

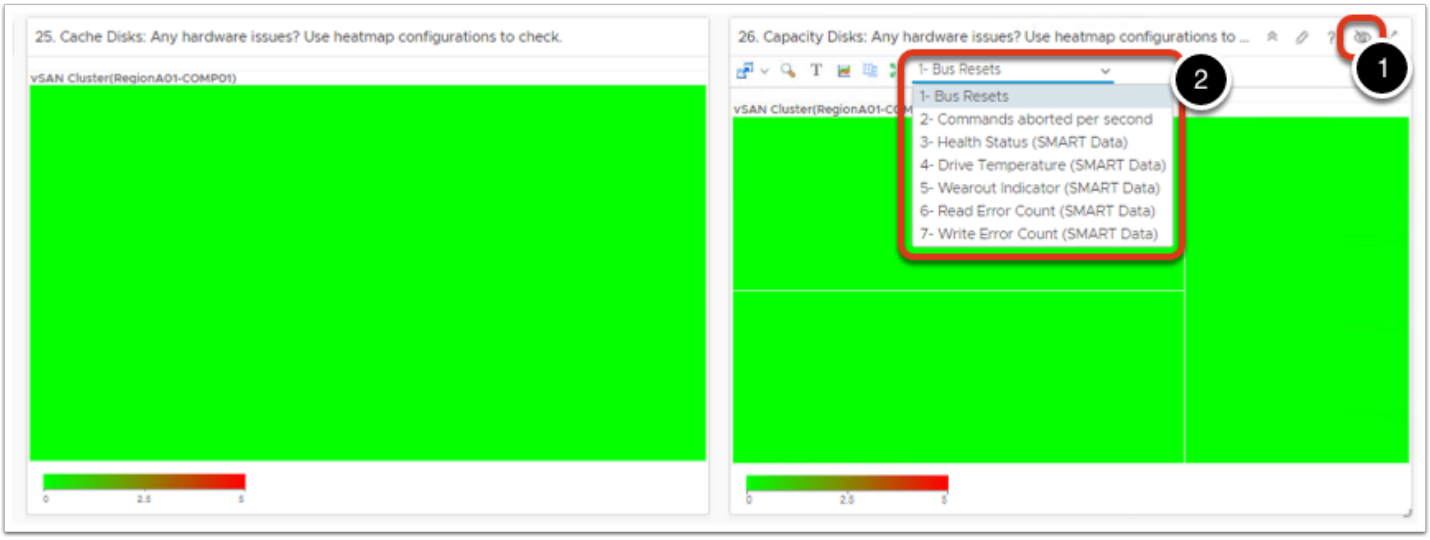
1. **Scroll-down** to view more information

Troubleshoot vSAN, cont.



1. Click the down **Chevrons** to expand **Capacity Disks**

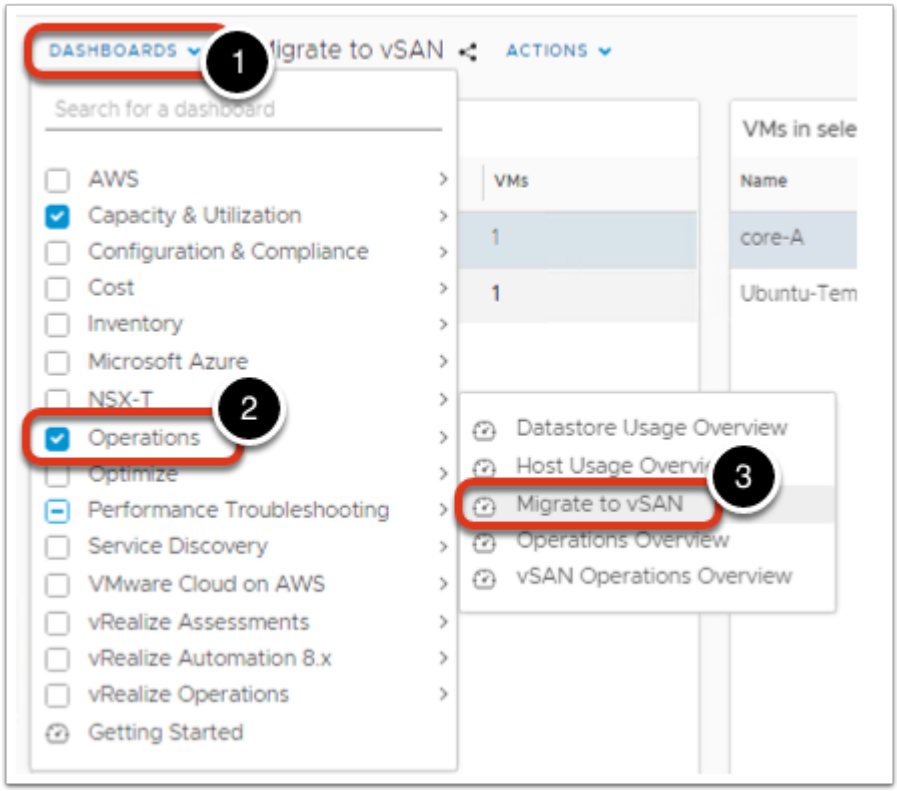
Troubleshoot vSAN, cont.



1. Hover over the **Capacity Disks** toolbar and click the **Show Toolbar** icon
2. Expand the **1-Bus Resets** pull-down menu

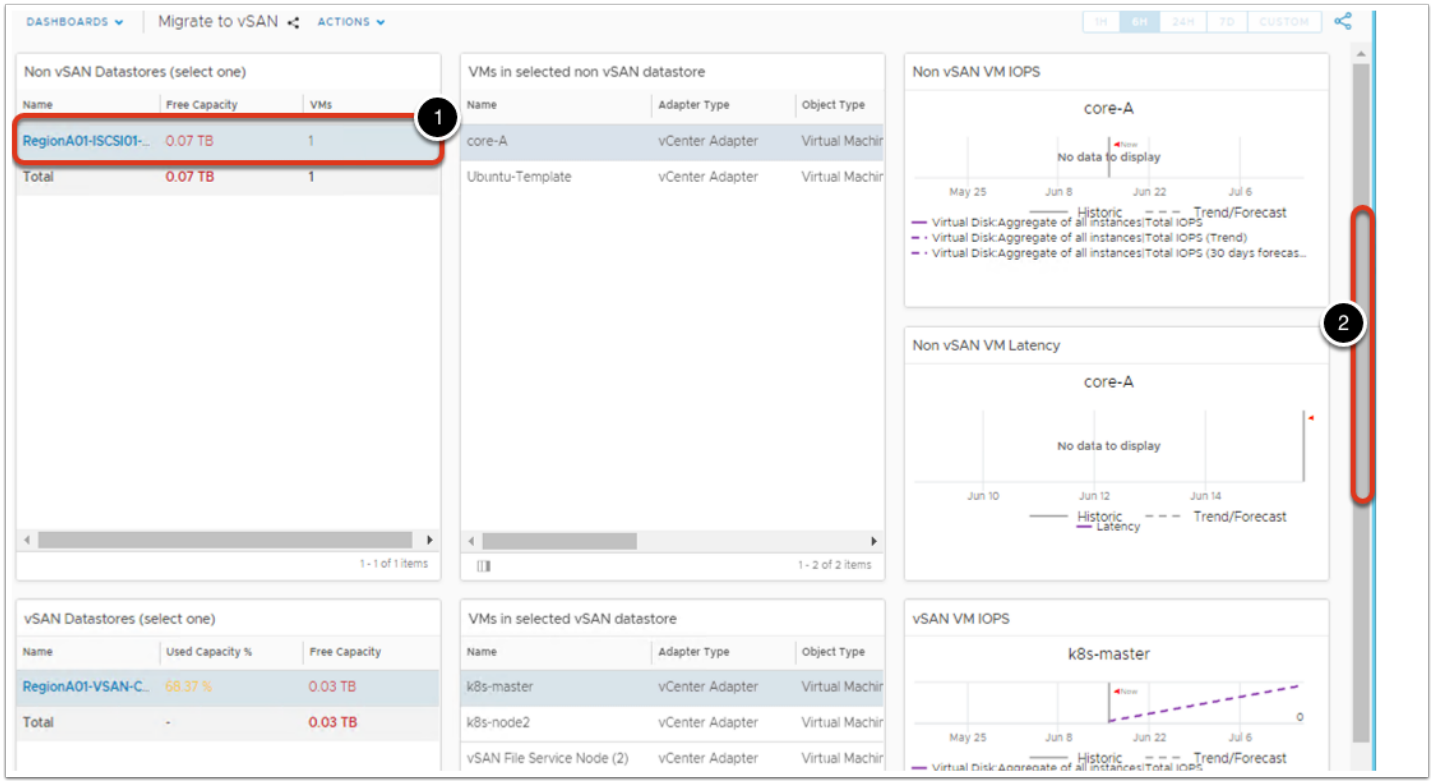
The Troubleshoot vSAN dashboard also looks at the health and performance of cache and capacity disks of the selected vSAN cluster. These widgets allow you to choose from one of the seven defined data types, and will then render the amount of activity in the heat map. The data types that can be viewed include bus resets, commands aborted per second, and five types of SMART data measurements.

All Dashboards



1. Select the **All Dashboards** drop-down menu
2. Hover over **Operations**
3. Click **Migrate to vSAN**

Migrate to vSAN



The Migrate to vSAN dashboard is designed to assist with migration efforts over to vSAN. The dashboard provides a comparison of critical storage metrics of a VM running on a data store running on traditional storage against a VM powered by vSAN. This dashboard recognizes the phased approach that can occur when transitioning to a new storage system, and is intended to monitor what matters most in the transition: the effective performance behavior between storage systems as seen by the application, or VM.

While each VM's workload will be unique, and not an exact moment by moment mirror of another VM's workload, one can compare similar systems effectively. For example, applications farms such as SQL clusters, ERP systems, SharePoint servers, or some other multi-tiered application use a cluster of VMs to provide back-end, middle-tier, or front-end services. Any of these examples would be an ideal scenario for comparison, as one of the systems in the application farm can be migrated over to vSAN, and compared to a similar system running on the legacy storage.

1. Note that we have a non-vSAN Datastore in our Lab (freeNAS appliance: **RegionA01-ISCSI01-...**)
2. **Scroll-down** to compare **Non vSAN VM IOPS and Latency** vs. vSAN VM IOPS and Latency

- The **Non vSAN VM widgets** will show the aggregate virtual disk IOPS, read latency, and write latency of the selected VM running on a legacy data store. The

vSAN VM widgets will show the same metrics of the selected VM running on a vSAN powered data store.

- Thanks to the power of customization, you may wish to simplify this dashboard. This might allow you to utilize a larger portion of the screen for key metrics, and simplify operation, or reduce the time window of observation to a smaller window.

Conclusion

In this module, we showed you how to validate vSAN Health, Monitor vSAN Capacity & Performance as well as utilize vRealize Operations for vCenter and vRealize Operations Dashboards.

You Finished Module 5

Congratulations on completing Module 5.

How to End Lab

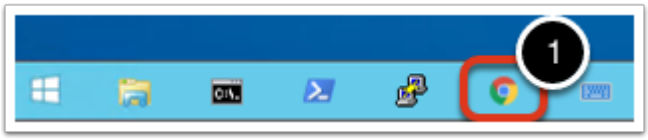
To end your lab, click on the **END** button.

Enabling vRealize Operations within vCenter

It takes approximately 30 minutes to enable vRealize Operations for vCenter within our Lab environment.

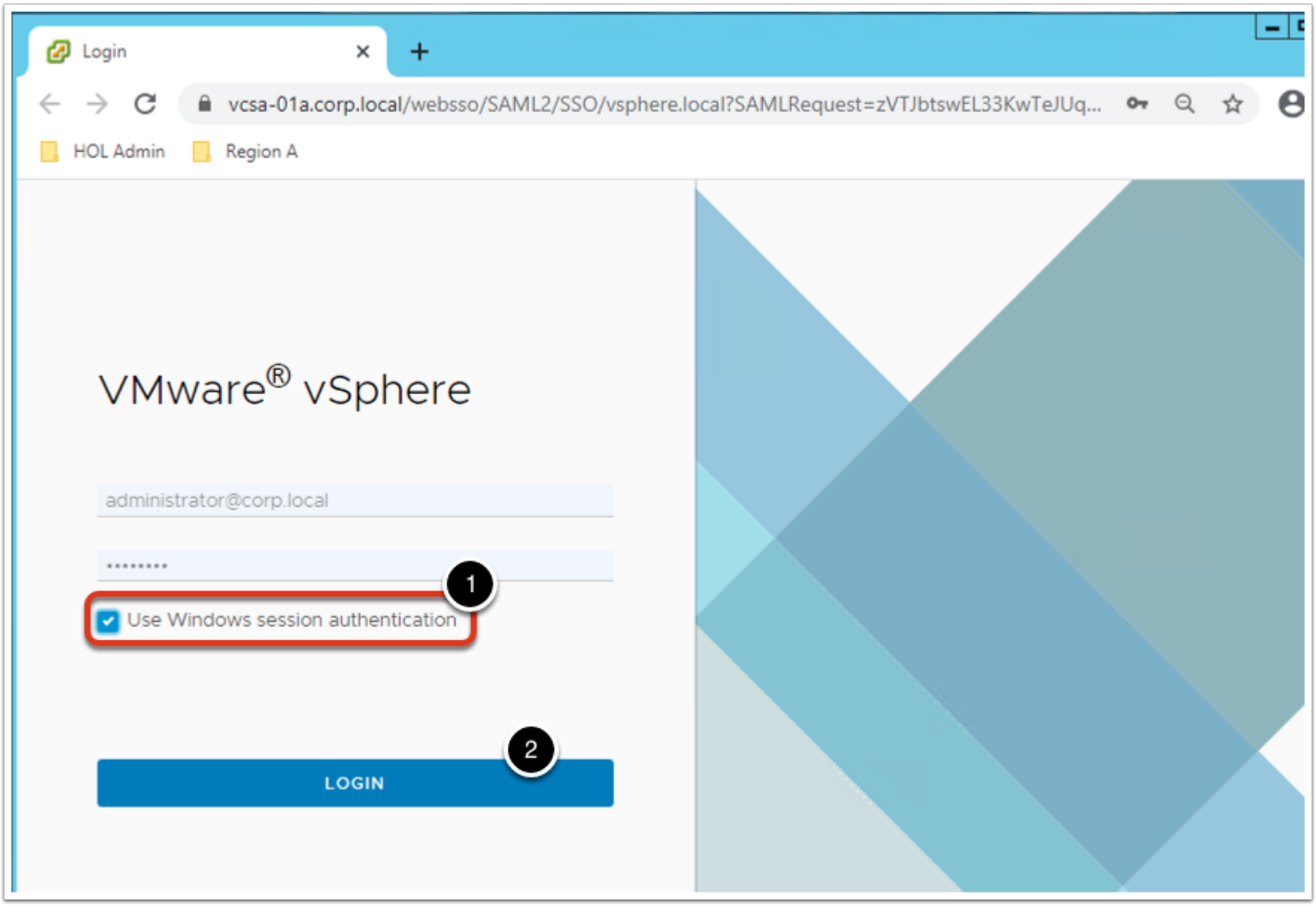
In these next steps, we will ask you to walk through the steps necessary to enable vRealize Operations for vCenter.

Open Chrome Browser from Windows Quick Launch TaskBar



1. Click on the **Chrome** Icon on the Windows Quick Launch task bar.

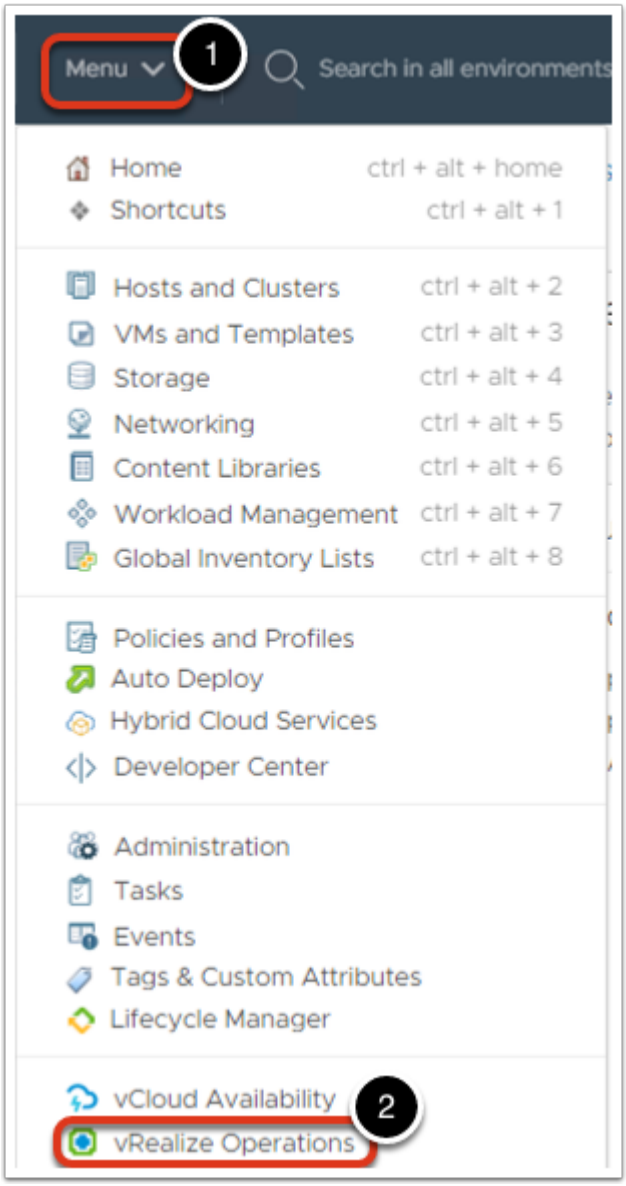
Login to vSphere Client



1. On the vSphere Client Login screen, select "**Use Windows session authentication**"

2. Click **Login**

vRealize Operations

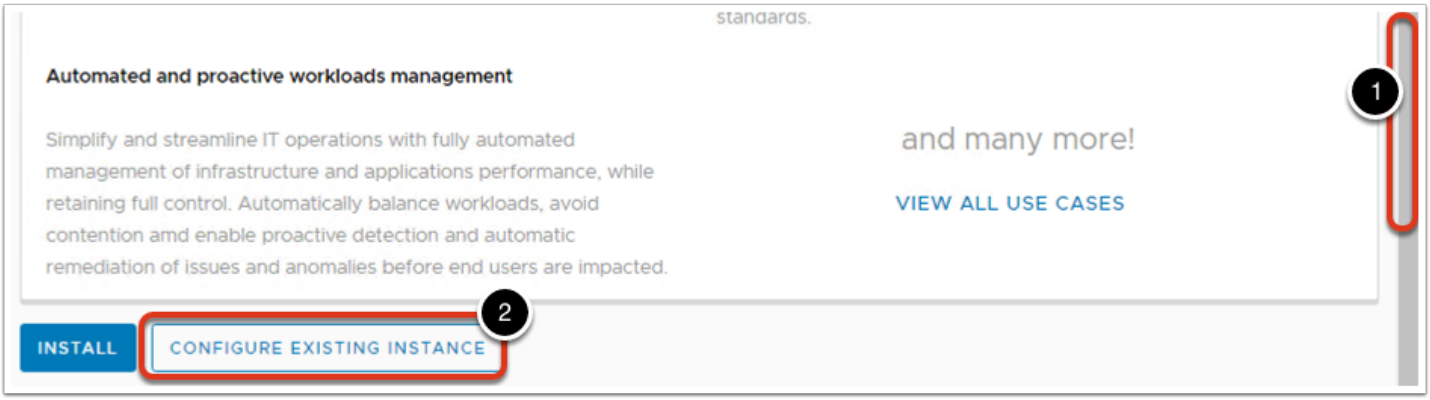


1. Select **Menu**

2. Click **vRealize Operations**

Note that you receive a message indicating that vRealize Operations is not present, We will correct that condition next.

Configure Existing Instance

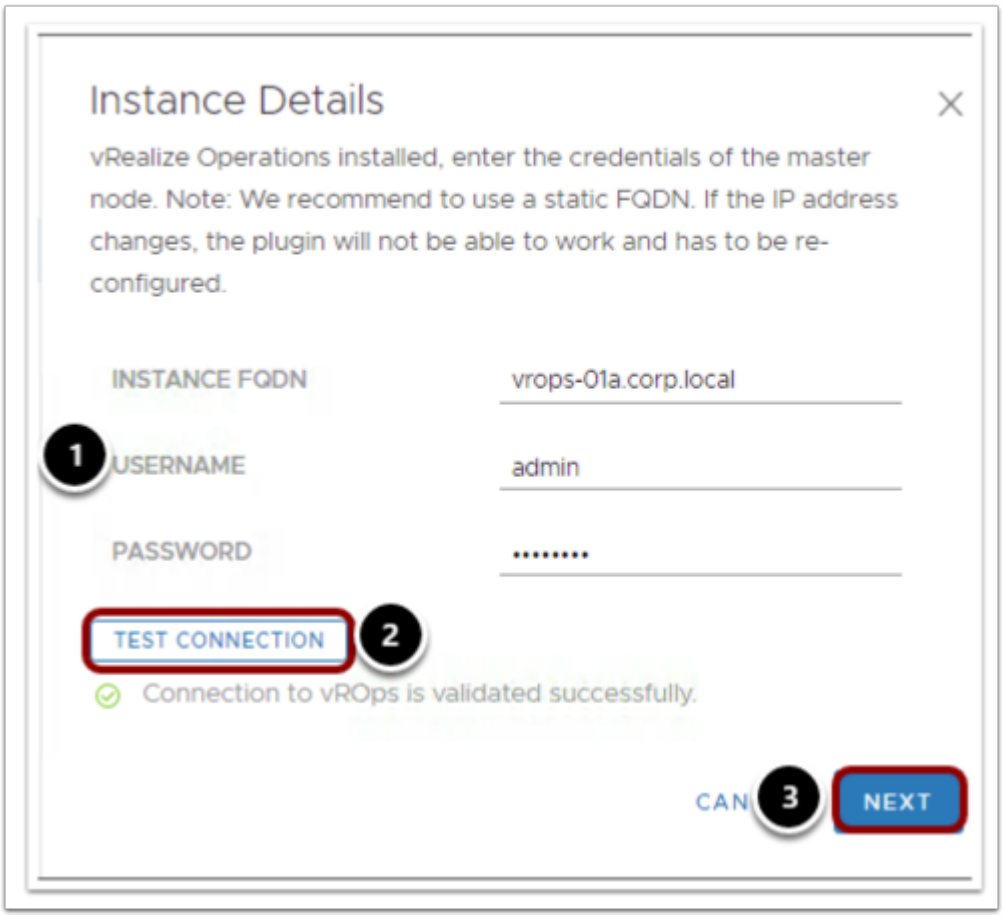


1. Scroll **Down**

2. Click **Configure Existing Instance**

Note that you have the options to install a new instance of vRealize Operations or Configure Existing Instance. In out Lab, vRealize Operations has already been installed and this is the instance that we will tie into.

Instance Details



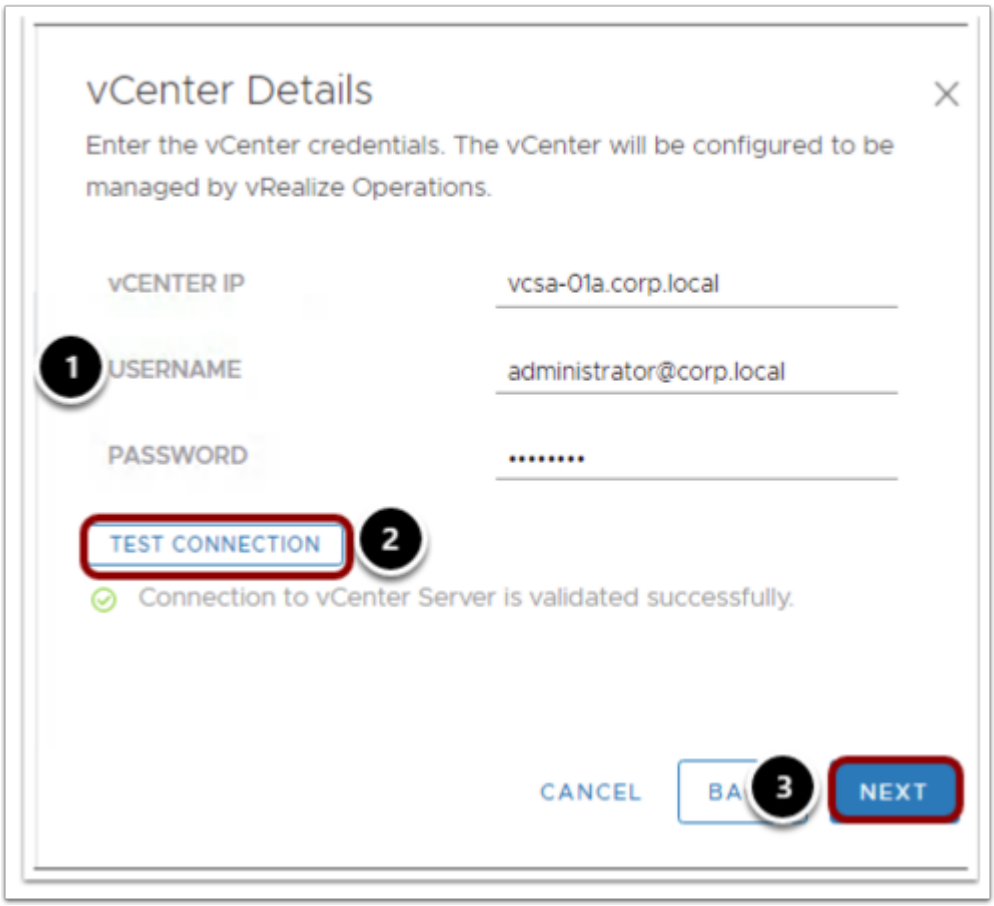
1. Enter parameters:

```
INSTANCE FQDN: vrops-01a.corp.local
USERNAME: admin
PASSWORD: VMware1!
```

2. Select **Test Connection** to validate credentials (re-enter if necessary)

3. Click **NEXT**

vCenter Details



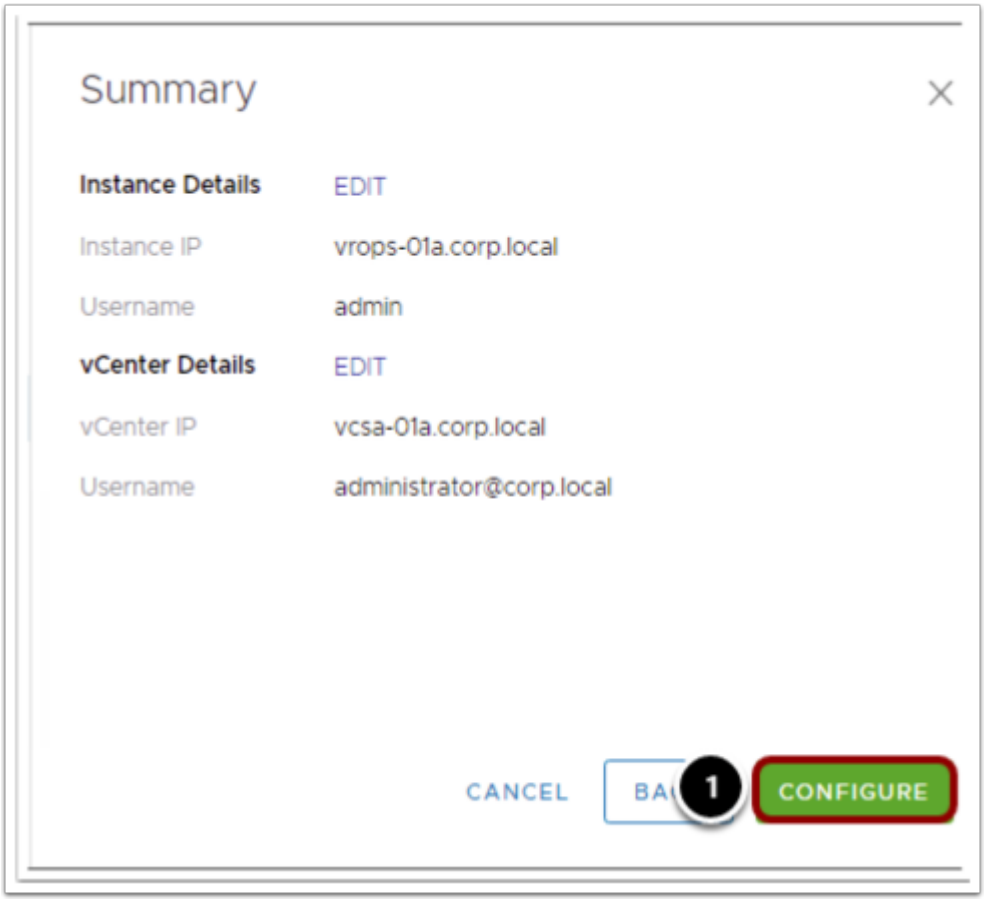
1. Enter parameters:

```
INSTANCE FQDN: vcsa-01a.corp.local
USERNAME: administrator@corp.local
PASSWORD: VMware1!
```

2. Select **Test Connection** to validate credentials (re-enter if necessary)

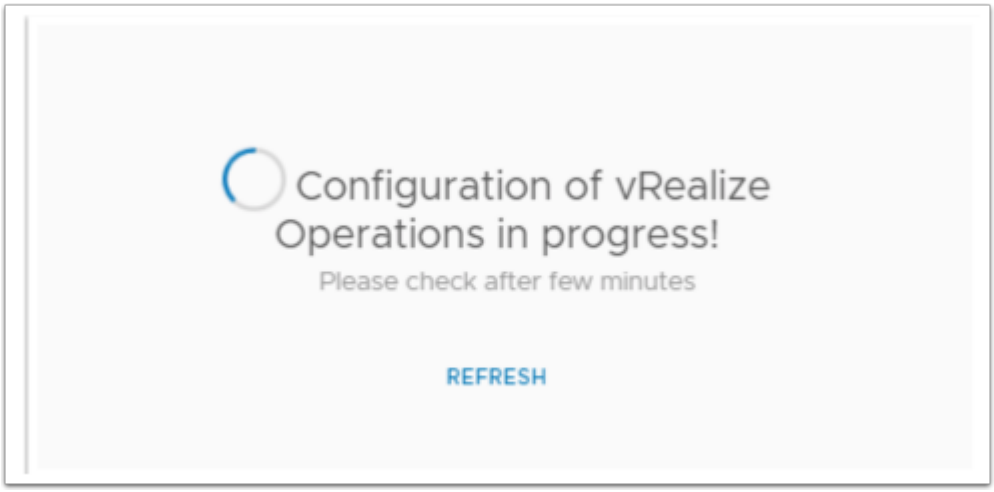
3. Click **Next**

Summary



1. Click **Configure**

vROPs Configuration



Configuration will take several minutes in our Lab.

Do not click REFRESH yet.

While we wait, let's examine vSAN Health and vCenter Capacity & Performance Monitoring before utilizing vRealize Operations for vCenter.

Module 6 - Day 2 Continue: Maintenance Mode and Lifecycle Management

Introduction

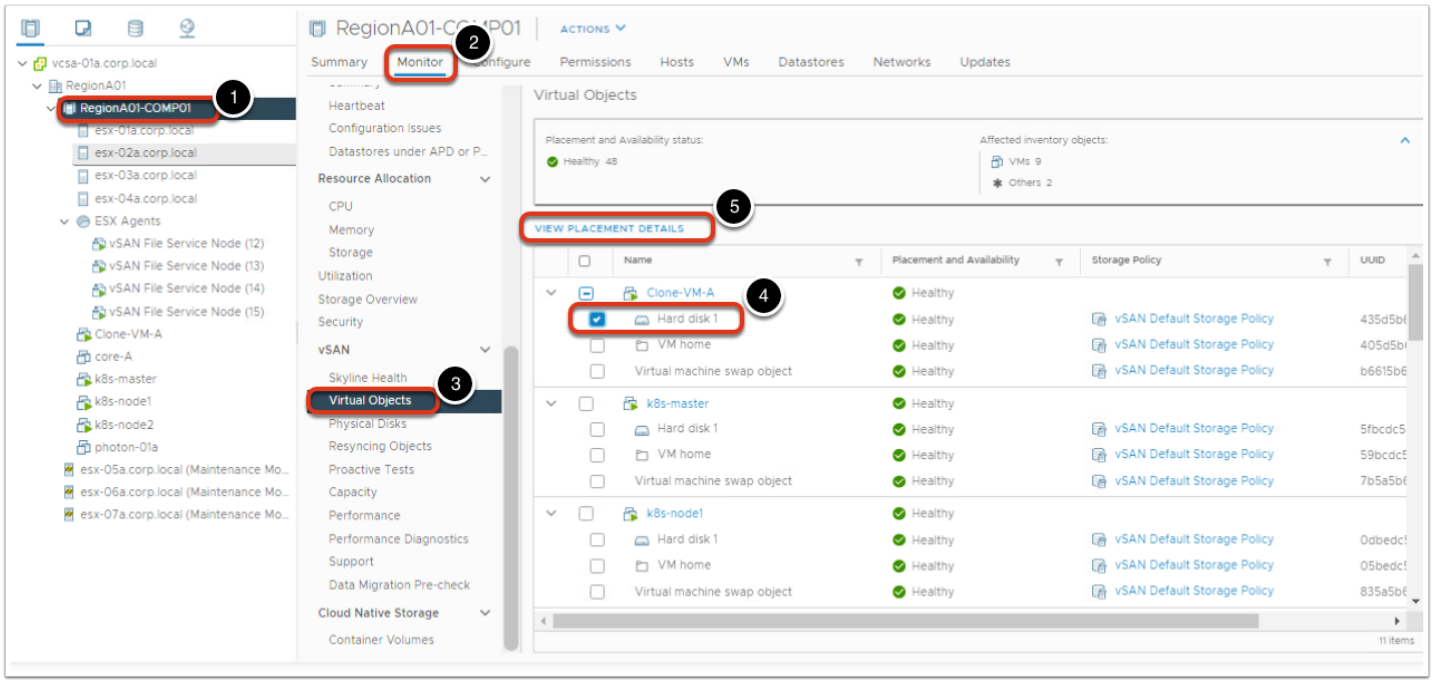
What happens after you've successfully enabled your vSAN Cluster and used all the great features of file services and cloud native storage?

Now it's time to learn how to utilize the Day 2 and what to expect when performing Maintenance activities, adding additional Capacity and updating vSAN via vSphere Lifecycle Manager (vLCM).

vSAN Maintenance

Before you shut down, reboot, or disconnect a host that is a member of a vSAN cluster, you must place the host in maintenance mode. When you place a host in maintenance mode, you must select a data evacuation mode, such as **Full data migration** to other hosts, **Ensure data accessibility** from other hosts or **No data migration**. In this Lesson, we will examine the various Maintenance Mode options as well as discuss when you might want to use one method vs. another.

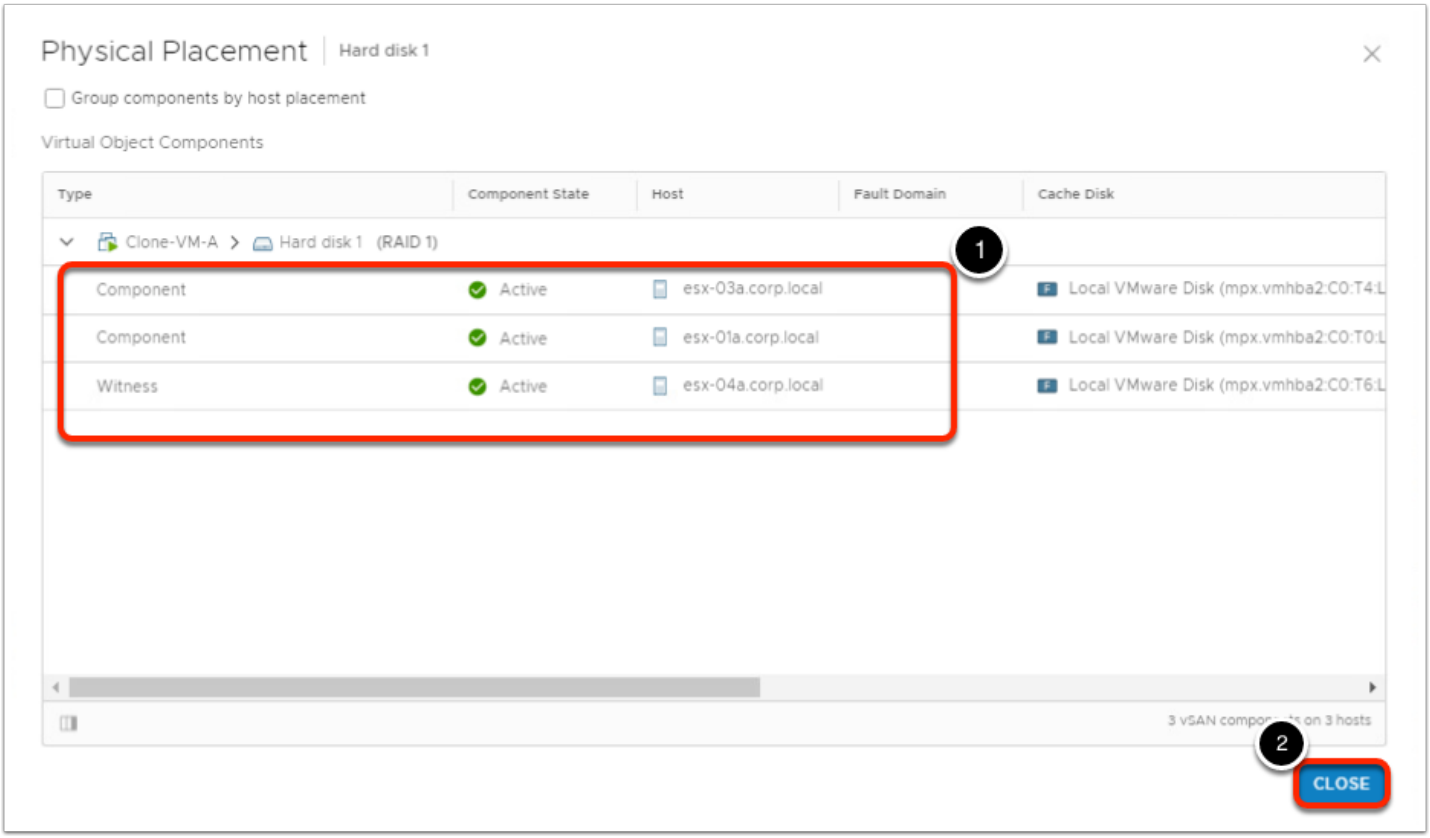
Virtual Objects



Let's examine the component layout across the vSAN Datastore for the Virtual Machine, **Clone-VM-A:**

1. Select **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Virtual Objects**
4. Enable the Checkbox for **Hard disk 1**
5. Click **View Placement Details**

Physical Placement



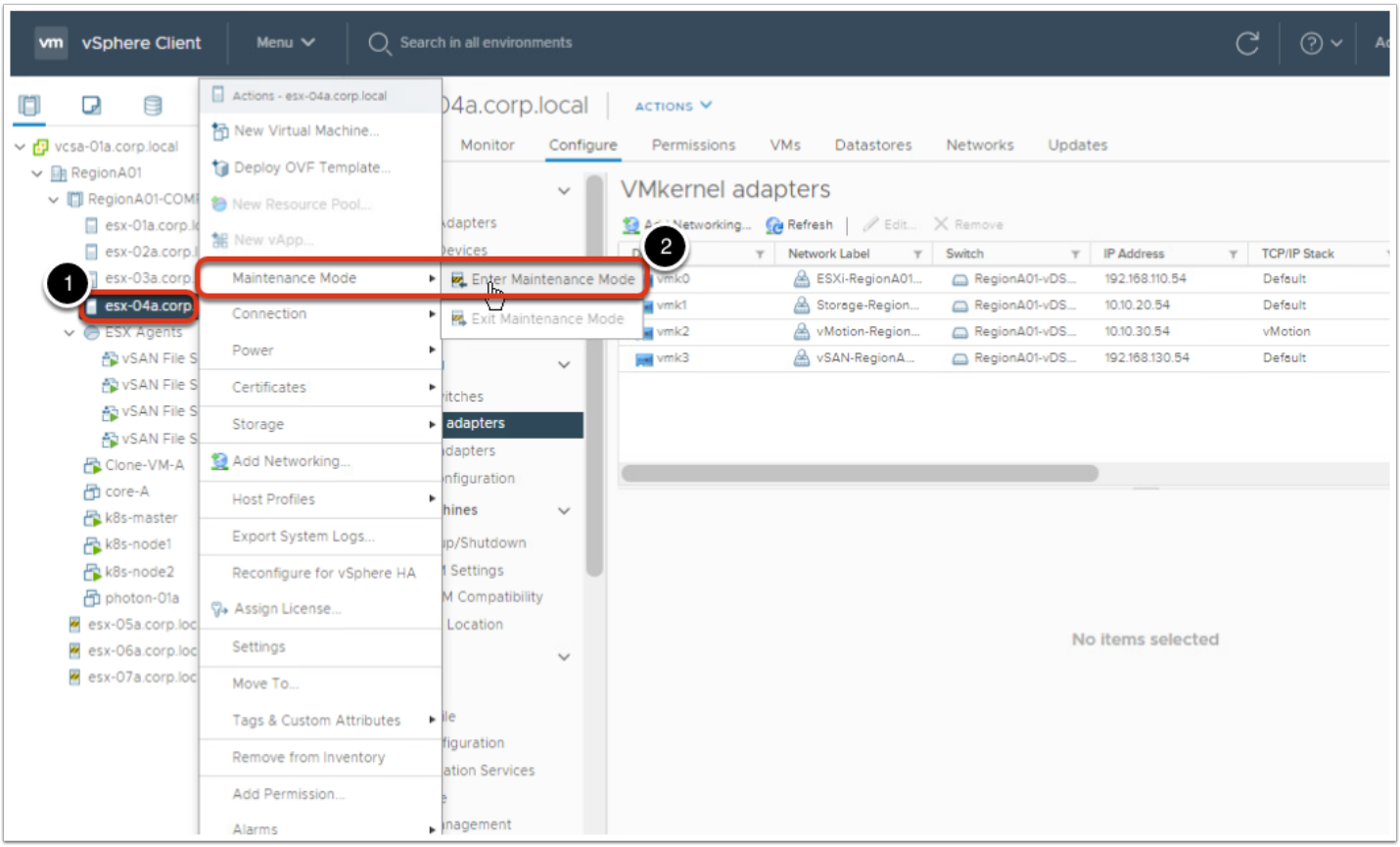
This Virtual Machine is using the Default vSAN Storage Policy (Primary Failures to Tolerate = 1)(PFTT).

1. PFTT=1 means that two of the VM's vSAN Components are in a RAID-1 Mirror across different vSphere Hosts as evidenced by the two **Component replicas** shown. A third **Witness** component is also present. Furthermore, we can see that all three components are reporting an **Active** (green) status.

Note: VM Object layout may be different in your lab vs. screenshot example

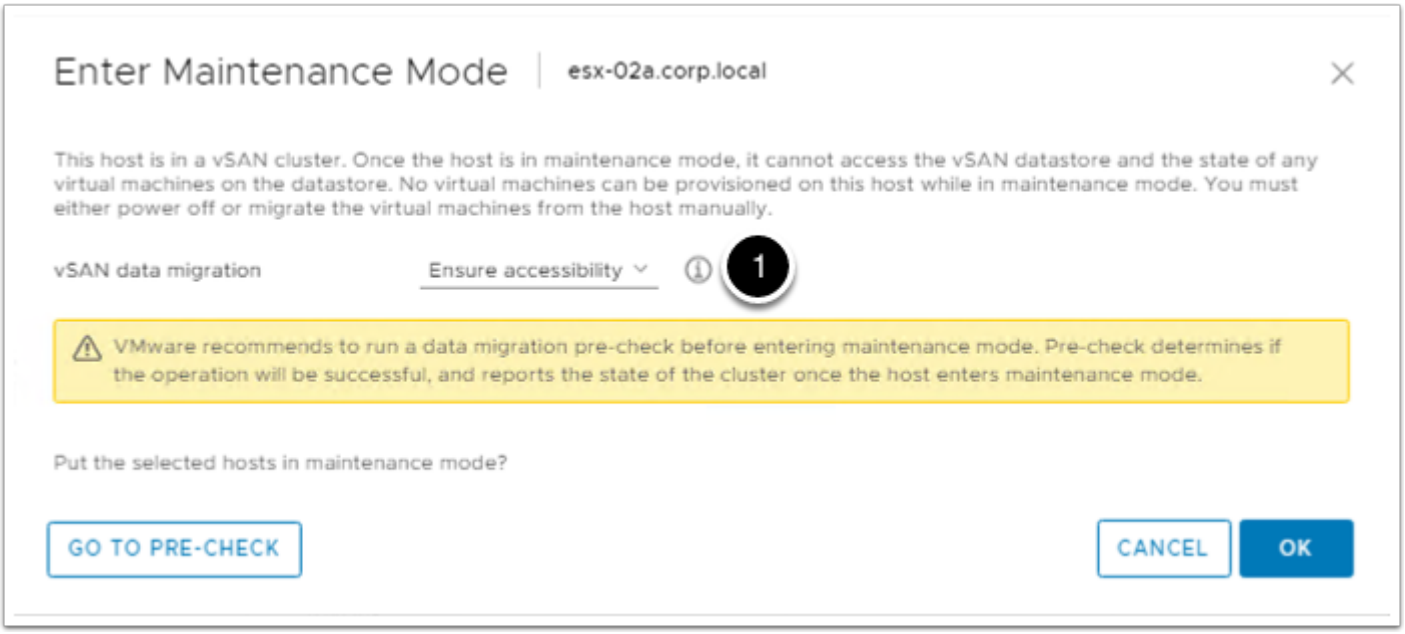
2. Click **Close**

Maintenance Mode



1. Right-click **esx-04a.corp.local**
2. Select **Maintenance Mode>Enter Maintenance Mode**

Maintenance Mode, cont.

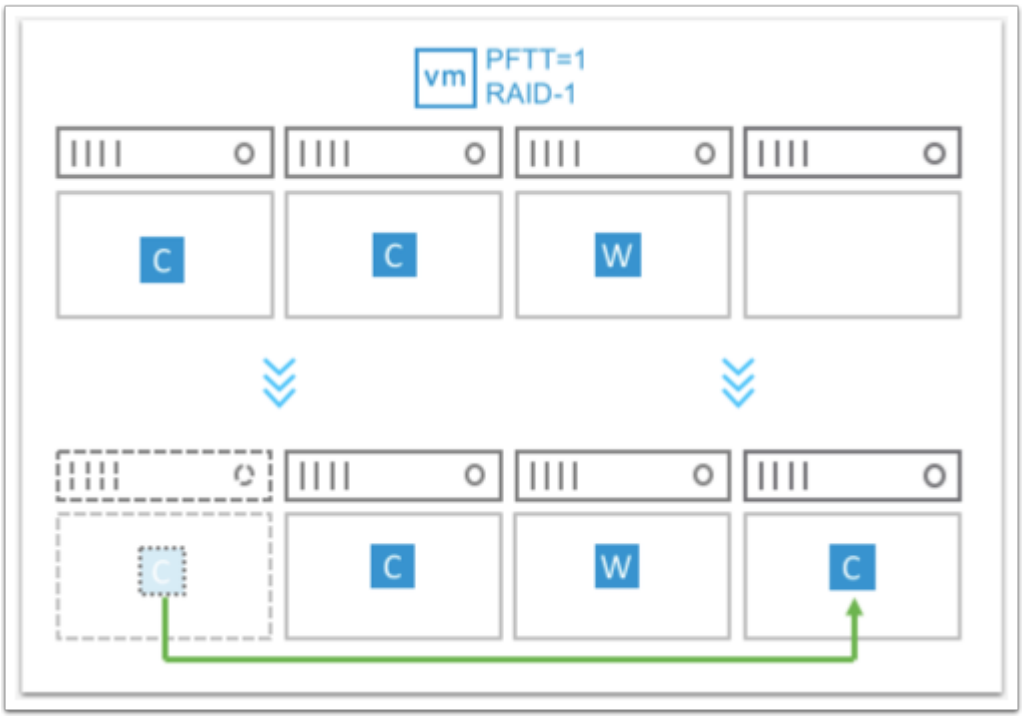


Note that there are three **vSAN data migration** options:

1. **Full data migration**
2. **Ensure accessibility**
3. **No data migration**

Let's examine these choices in more detail.

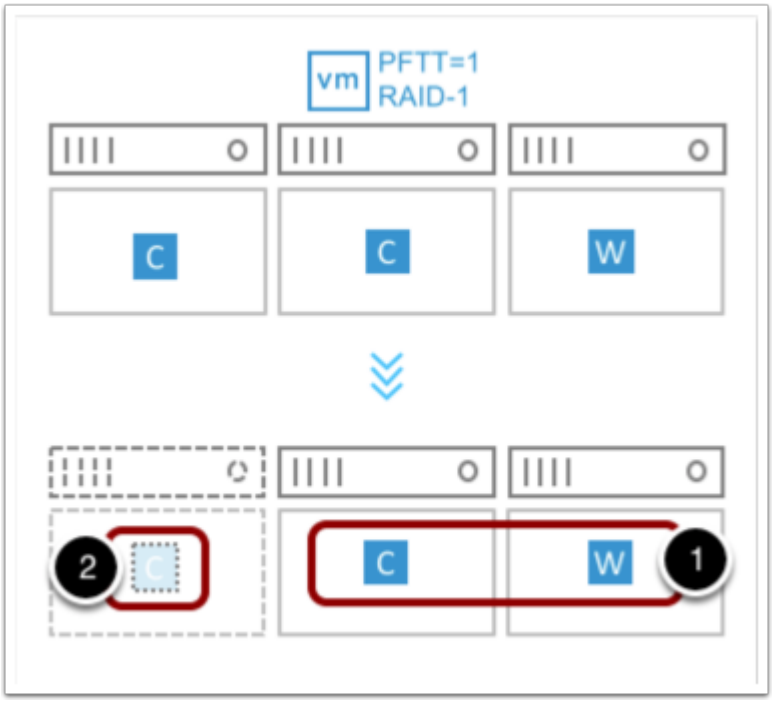
Full Data Migration



This option moves all of the vSAN components from the Host entering maintenance mode to other Hosts in the vSAN cluster. This option is commonly used when a Host will be offline for an extended period of time or permanently decommissioned.

Note: In order to maintain PFTT=1, we must have a 4th Host in our Cluster to Migrate any impacted Components to. In our current 3-Node Cluster, we do not have enough Hosts to satisfy this Maintenance Mode choice.

Ensure Accessibility

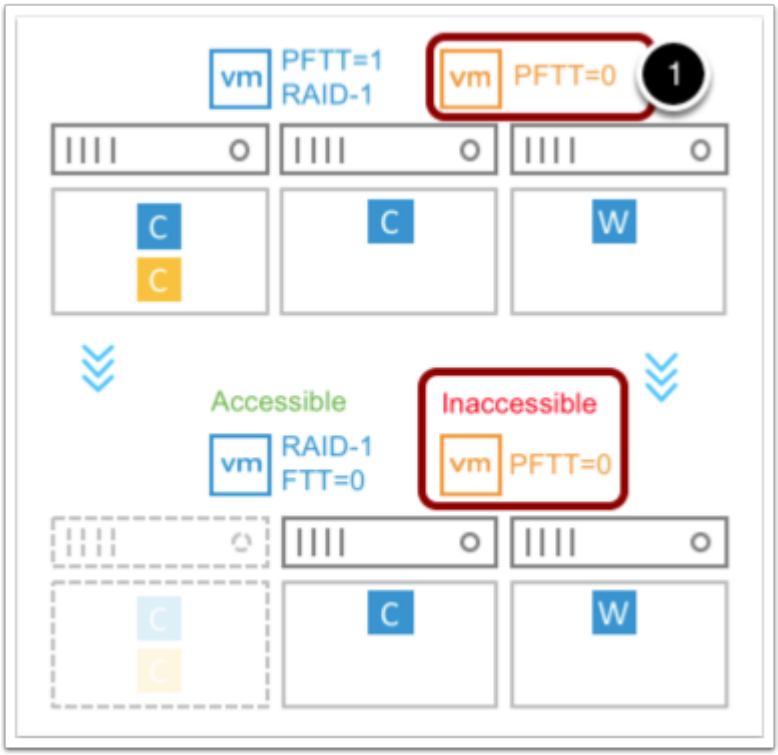


1. vSAN will verify whether the majority of a Virtual Machine's objects remain accessible even though one or more components will be absent due to the host entering maintenance mode.
2. If the majority of a Virtual Machine's objects will remain accessible, vSAN will not migrate impacted component(s).

If the VM's objects would become inaccessible, vSAN will migrate the necessary number of components to other hosts ensuring that the object accessibility. This option is the default and it is commonly used when the host will be offline for just a short amount of time, e.g., a host reboot. It minimizes the amount of data that is migrated while ensuring all objects remain accessible. However, the level of failure tolerance will likely be reduced for some objects until the host exits maintenance mode.

Note: This is the only **evacuation** mode available if you are working with a three-host cluster or a vSAN cluster configured with three fault domains.

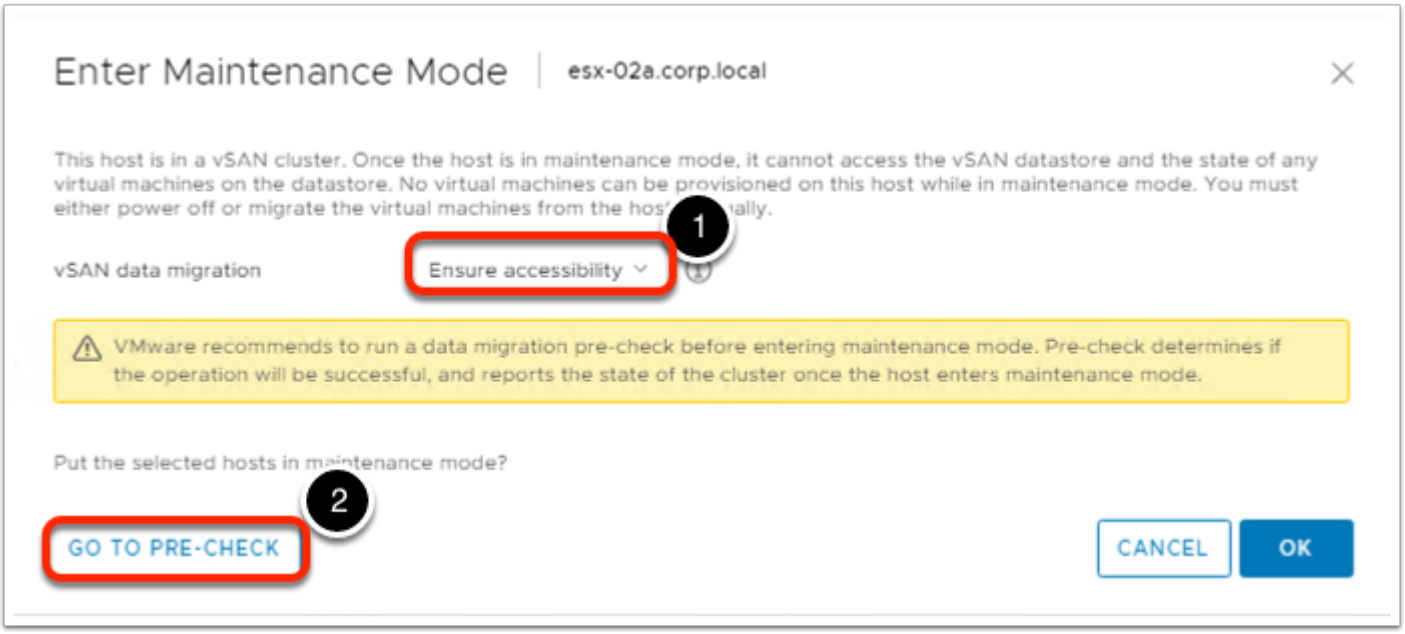
No Data Migration



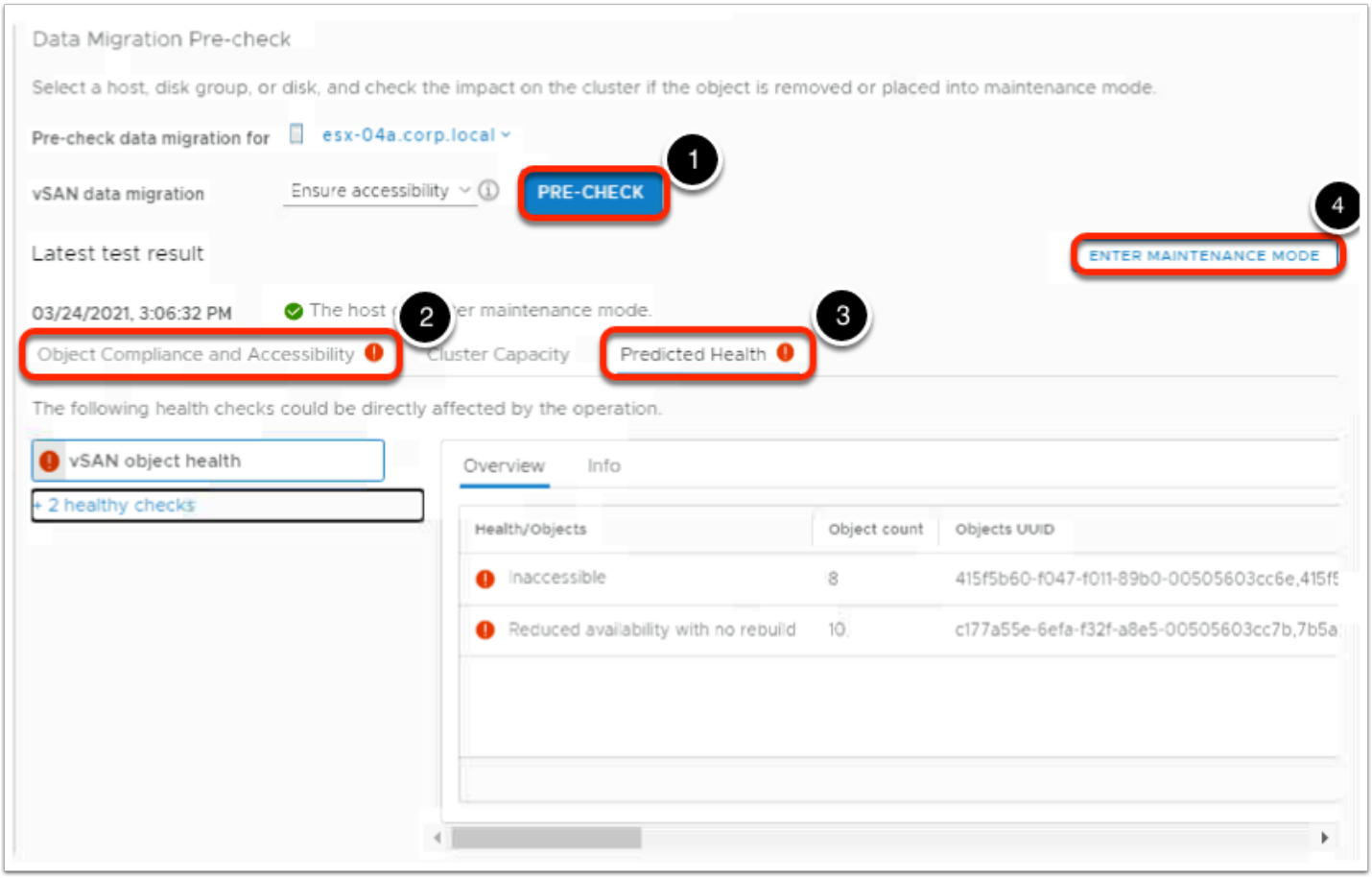
Data is not migrated from the host as it enters maintenance mode. This option can also be used when the host will be offline for a short period of time. All objects will remain accessible as long as they have a storage policy assigned where the Primary Level of Failures to Tolerate is set to one or higher.

1. This Virtual Machine (utilizing a PFTT=0 policy) will not be accessible when the Host enters Maintenance mode as it's storage component will not be migrated and will therefore be offline

Enter Maintenance Mode



1. Validate that **Ensure accessibility** is selected
2. Click **GO TO PRE-CHECK**



As you can see, we can enter maintenance mode but we will have come compliance and health issues within the cluster.

1. Click on **PRE-CHECK**
2. Select **Object Compliance and Accessibility**

You see that drives and objects will be inaccessible.

NOTE: The Virtual Machines objects will become **non-compliant** with their storage policy

As we learned earlier, the Virtual Machine storage will still be accessible as there are enough remaining components elsewhere on the Cluster. Non-compliant, in this case, does not mean that the VM is not available.

NOTE: A rebuild operation for any non-compliant objects will be triggered in 60 minutes, unless the host is taken out of maintenance mode. You can change this timer for the cluster in the vSAN advanced settings.

2. Select **Predicted Health**

You see how many incessible objects and reduced availability with no rebuild

3. Click on **Enter Maintenance Mode.** It will take you back to the previous screen.

- Click **OK** (not shown in picture)

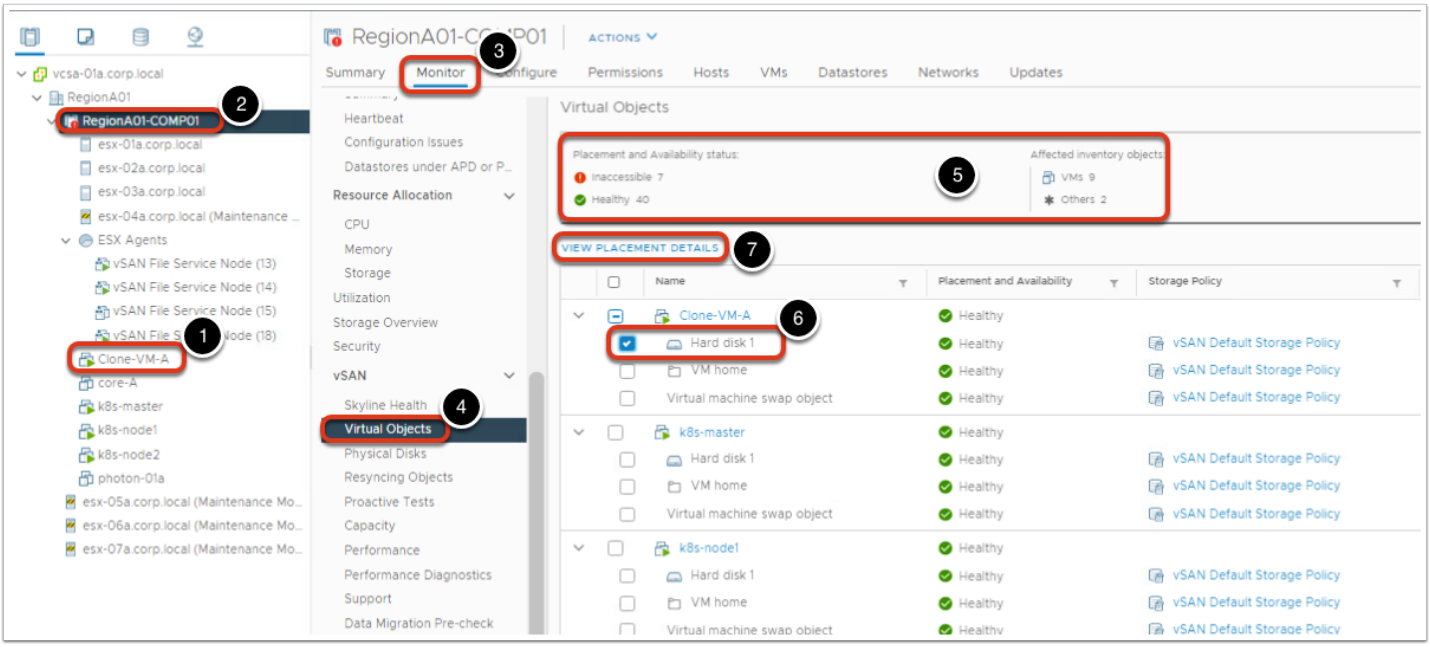
Enter Maintenance Mode, cont.

Recent Tasks Alarms

Task Name	Target	Status	Details	Initiator	Queued For	Start Time	Completion Time	Server
Update vSAN configuration	esx-03a.corp.local	Completed		com.vmware.vsan.health	19 ms	06/01/2020, 5:24:50 PM	06/01/2020, 5:24:56 PM	vcsa-01a.corp.local
Update vSAN configuration	esx-02a.corp.local	Completed		com.vmware.vsan.health	9 ms	06/01/2020, 5:24:50 PM	06/01/2020, 5:24:55 PM	vcsa-01a.corp.local
Update vSAN configuration	esx-01a.corp.local	Completed		com.vmware.vsan.health	34 ms	06/01/2020, 5:24:50 PM	06/01/2020, 5:24:56 PM	vcsa-01a.corp.local
Remediate vSAN cluster	RegionA01-COMP01	Completed		com.vmware.vsan.health	40 ms	06/01/2020, 5:24:48 PM	06/01/2020, 5:24:56 PM	vcsa-01a.corp.local
Delete virtual machine	vSAN File Service ...	Completed		com.vmware.vsan.health	13 ms	06/01/2020, 5:24:48 PM	06/01/2020, 5:24:48 PM	vcsa-01a.corp.local
Install agent	esx-02a.corp.local	Completed		com.vmware.vim.esm	21 ms	06/01/2020, 5:20:24 PM	06/01/2020, 5:20:24 PM	vcsa-01a.corp.local

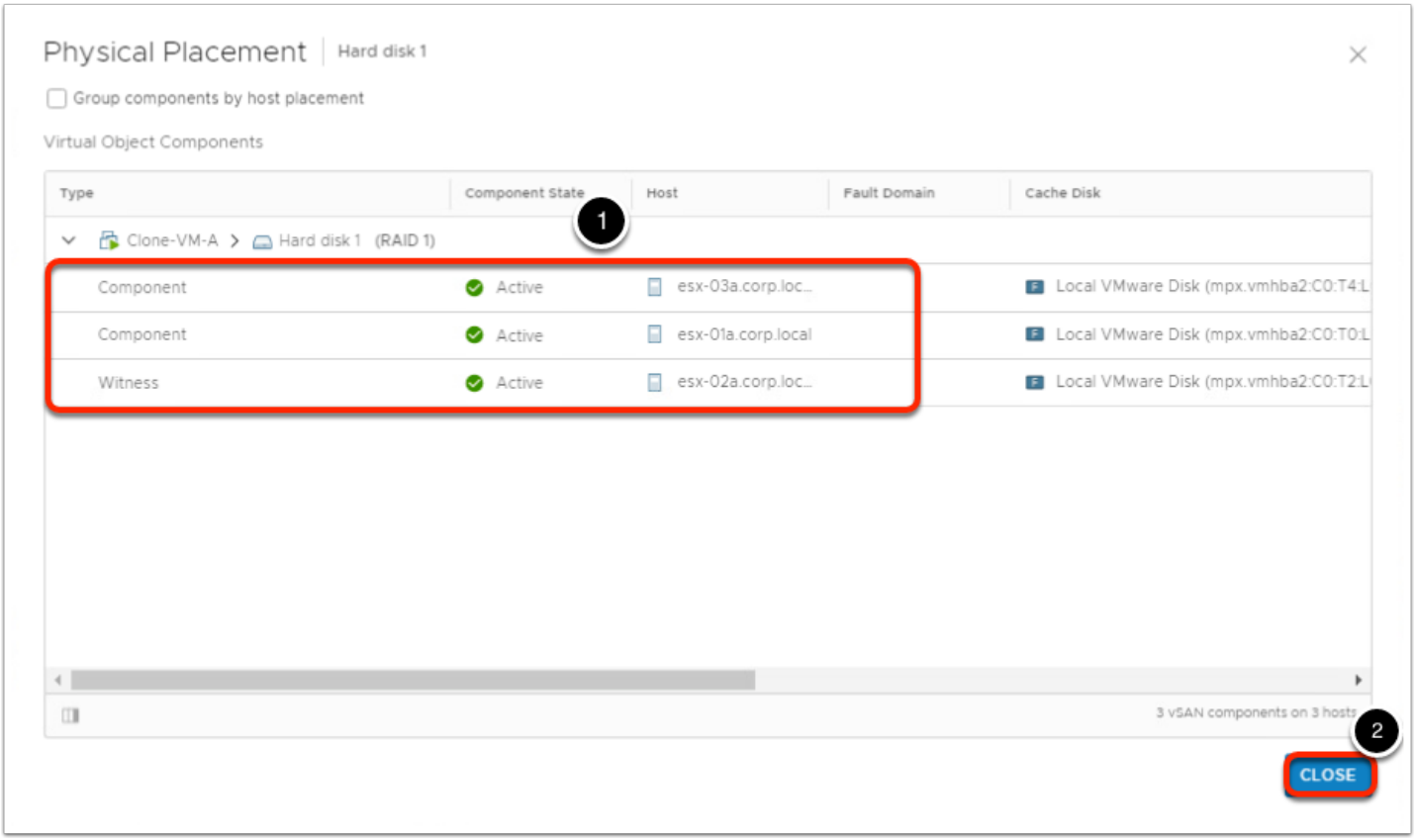
1. Select **Recent Tasks** (lower-right hand corner)
2. Monitor the maintenance mode and update on vSAN configuration

Virtual Objects



1. Notice that our **Clone-VM-A** is still online as evidenced by the Green Play Icon (you could perform a ping test as well but we will skip that for now)
2. Select **RegionA01-COMP01**
3. Select **Monitor**
4. Select **vSAN** > **Virtual Objects**
5. Note that the **VM-Clone-A** is currently in a **Reduced availability with no rebuild** status**.** Additionally, a delay timer is counting down (more on that in a moment)
6. Select **Hard disk 1**
7. Click **View Placement Details**

Physical Placement



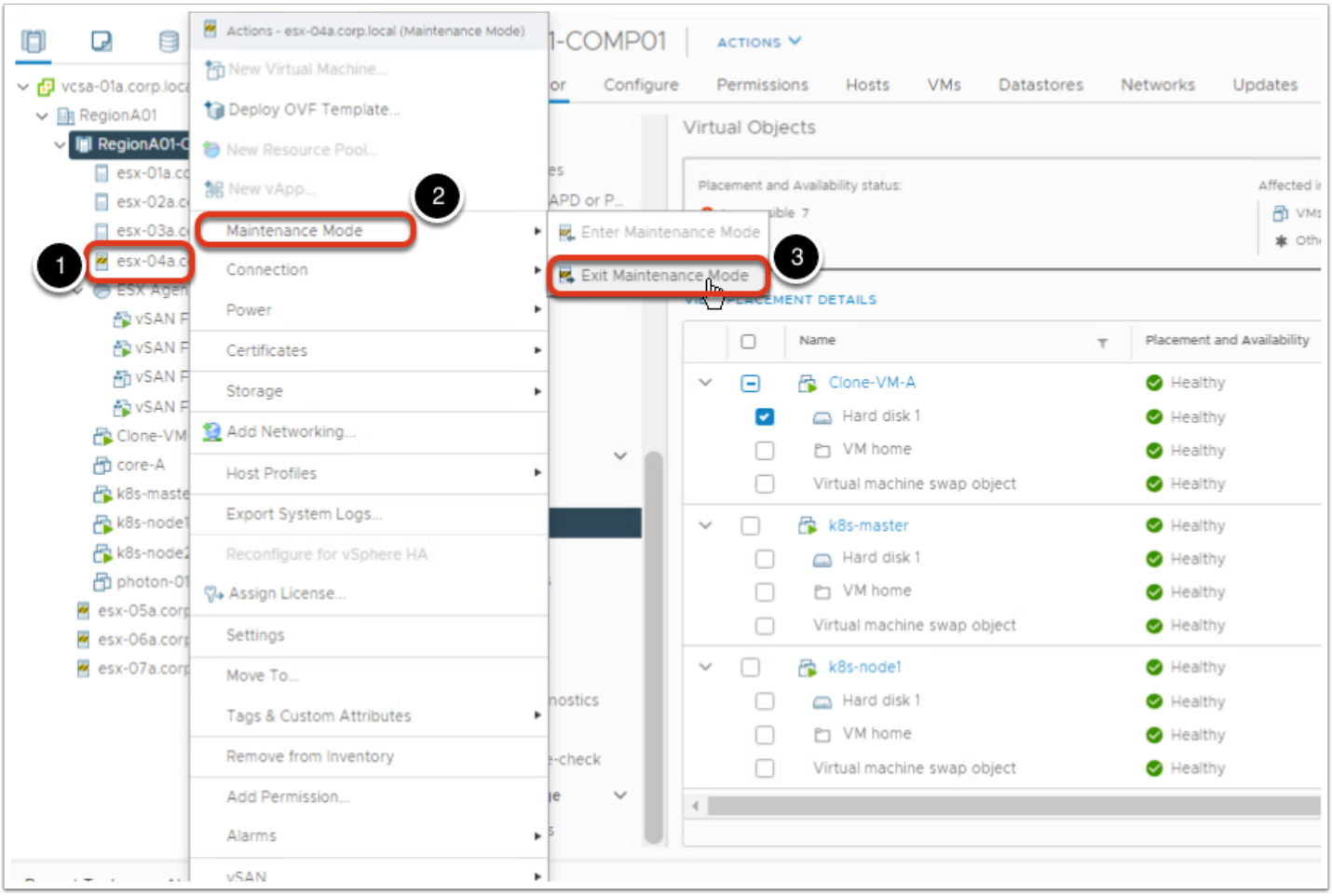
1. Here we can see that the VM's components are active as we still have 3 hosts available

vSAN waits 60 minutes before rebuilding any objects located on the unavailable host (hence the **rebuild timer** notification we saw previously). If the Host does not return within 60 minutes, vSAN marks the impacted Objects as **Degraded** and attempts to rebuild them on another Host in the cluster (assuming there is an available Host, which in our current 3-Node Cluster, there is not).

(The timer length is configurable and you can find a link to a KB article in the Conclusion section for this Module with more details).

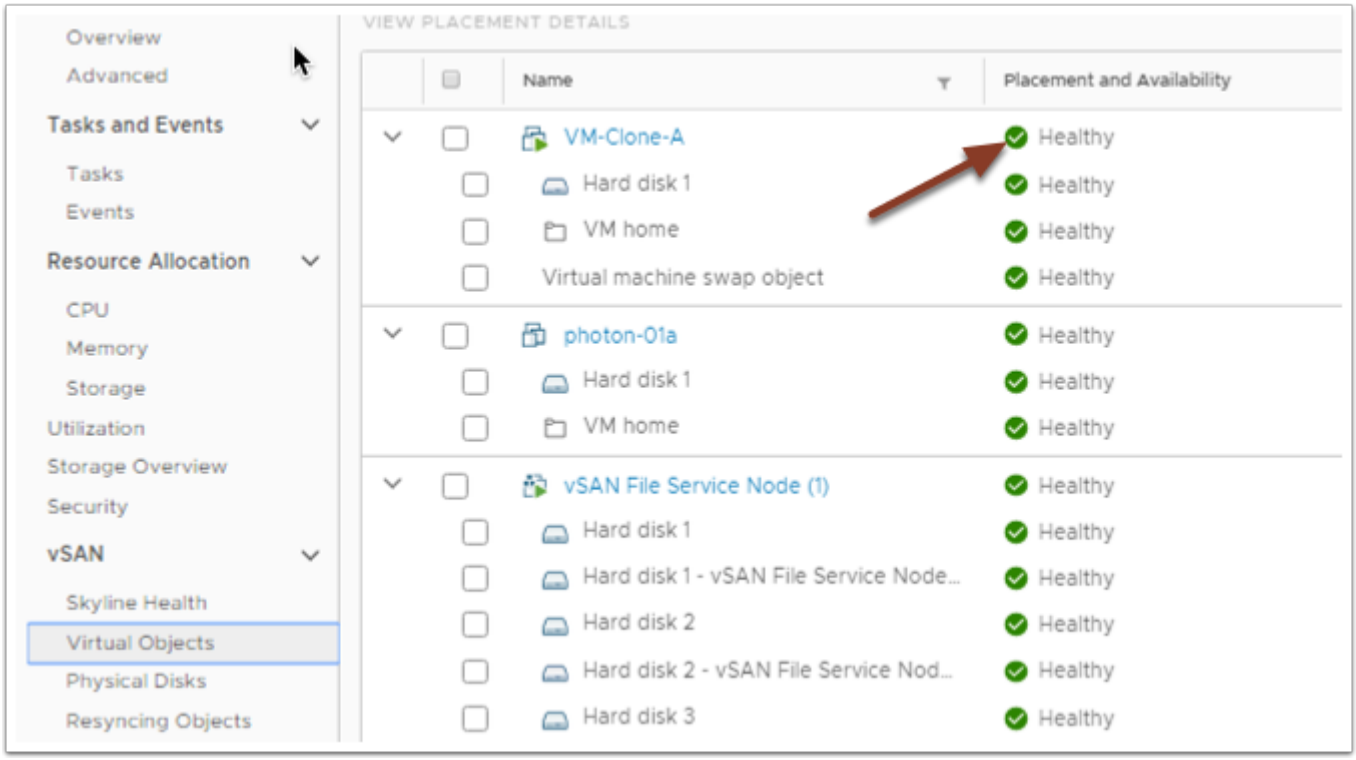
2. Click **Close**

Exit Maintenance Mode



1. Right-click **esx-02a.corp.local**
2. Hover over **Maintenance Mode**
3. Click **Exit Maintenance Mode**

Healthy

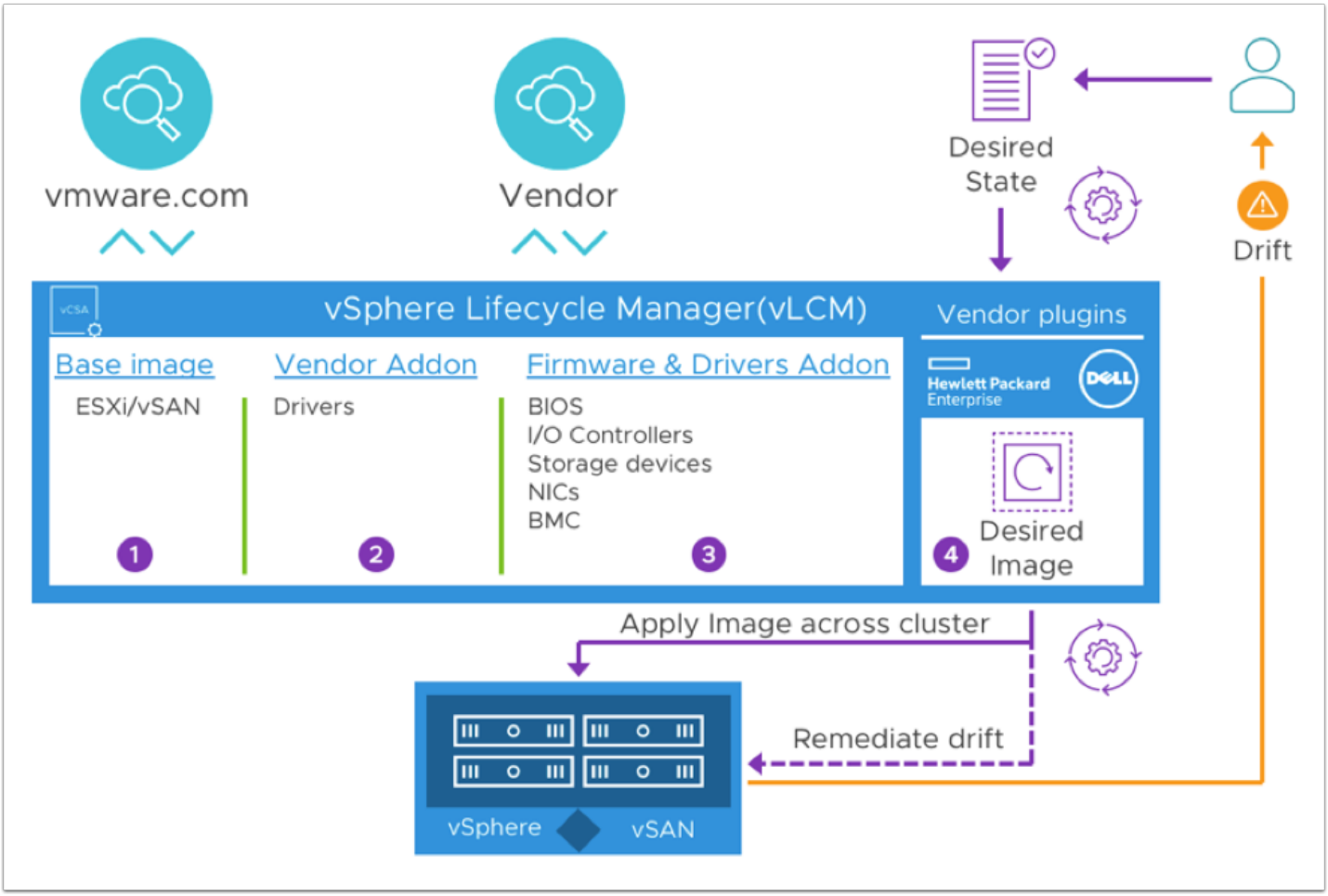


Note that the Virtual Machine is once again reporting a **Healthy** (green) status. vSAN has automatically 'caught up' the Object that was on the Maintenance Mode Host as part of putting the Host back into service.

(You might need to refresh the vSphere client in order to confirm Healthy status)

vSphere Lifecycle Management (vLCM)

vSphere 7 introduces an entirely new solution for unified software and firmware lifecycle management that is native to vSphere. vSphere Lifecycle Manager, or vLCM, is a powerful new framework based on a desired state model to deliver simple, reliable and consistent lifecycle operations for vSphere and HCI clusters.



Simplify Cluster Updates with vSphere Lifecycle Manager

Lifecycle management is a time-consuming task. Admins maintain their infrastructure with many tools that require specialized skills. VMware customers easily use two or more separate interfaces for day two operations: vSphere Update Manager (VUM) for software and drivers and one or more server vendor-provided utilities for firmware updates. Moreover, vSAN admins must ensure that driver and firmware versions are on the VMware Compatibility Guide (VCG) rather than simply applying the latest and greatest versions supplied by the server vendor.

vLCM is built off a desired-state model that provides lifecycle management for the hypervisor and the full stack of drivers and firmware for your hyperconverged infrastructure. vLCM can be used to apply a desired image at the cluster level, monitor

compliance, and remediate the cluster if there is a drift. This eliminates the need to monitor compliance for individual components and helps maintain a consistent state for the entire cluster in adherence to the VCG. Lets take a closer look.

vLCM Desired Image

vLCM is based on a desired-state or declarative model which allows the user to define a desired image (ESXi version, drivers, firmware) and apply to an entire vSphere or HCI cluster. Once defined and applied, all hosts in the cluster will be imaged with the desired state. This model is superior to managing individual hosts because the image is applied at the cluster level which provides consistency and simplicity.

A vLCM Desired Image consists of a base ESXi image (required), vendor addons, and firmware and driver addons.

- Base Image: The desired ESXi version that can be pulled from vSphere depot or manually uploaded.
- Vendor Addons: Packages of vendor specified components such as firmware and drivers.

While vendor addons are not required for an image, they are required for users wanting to take advantage of the overall full server-stack firmware management available with vLCM.

Hardware Compatibility Checks

HCI administrators understand using compatible hardware components, drivers, and firmware is critically important to the operation and performance of the cluster and the virtual machines running on it. Using the Hardware Capability tab in vLCM, administrators can validate that the desired ESXi version is compatible with the server hardware and that the storage IO controller firmware and drivers are compatible with the VCG. It is a recommended best practice to validate a configured desired image against the VCG before the user remediates the cluster.

Hands-on Labs Interactive Simulation: vSAN vLCM - Setup

This part of the lab is presented as a **Hands-on Labs Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as if you are interacting with a live environment.

1. Click here to open the interactive simulation. It will open in a new browser window or tab.
2. When finished, click the “Return to the lab” link to continue with this lab.

The lab continues to run in the background. If the lab goes into standby mode, you can resume it after completing the module.

Hands-on Labs Interactive Simulation: vSAN vLCM - Update

This part of the lab is presented as a **Hands-on Labs Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as if you are interacting with a live environment.

1. Click here to open the interactive simulation. It will open in a new browser window or tab.
2. When finished, click the “Return to the lab” link to continue with this lab.

The lab continues to run in the background. If the lab goes into standby mode, you can resume it after completing the module.

Hands-on Labs Interactive Simulation: vSAN vLCM - Rollback

This part of the lab is presented as a **Hands-on Labs Interactive Simulation**. This will allow you to experience steps which are too time-consuming or resource intensive to do live in the lab environment. In this simulation, you can use the software interface as if you are interacting with a live environment.

1. Click here to open the interactive simulation. It will open in a new browser window or tab.
2. When finished, click the “Return to the lab” link to continue with this lab.

The lab continues to run in the background. If the lab goes into standby mode, you can resume it after completing the module.

Conclusion

In this module, you learned what happen when a host enters maintenance mode. Planned events such as maintenance mode activities and unplanned events such as host outages may make the effective storage policy condition different than the assigned policy. vSAN constantly monitors this, and when resources become available to fulfill the rules of the policy, it will adjust the data accordingly. vSphere Lifecycle Manager reduces the complexity of monitoring and maintaining your hyper converged infrastructure by consolidating software, driver and firmware update tools and introducing a desired-state model of implementing a desired image.

You Finished Module 6

Congratulations on completing Module 6.

How to End Lab

To end your lab, click on the **END** button.

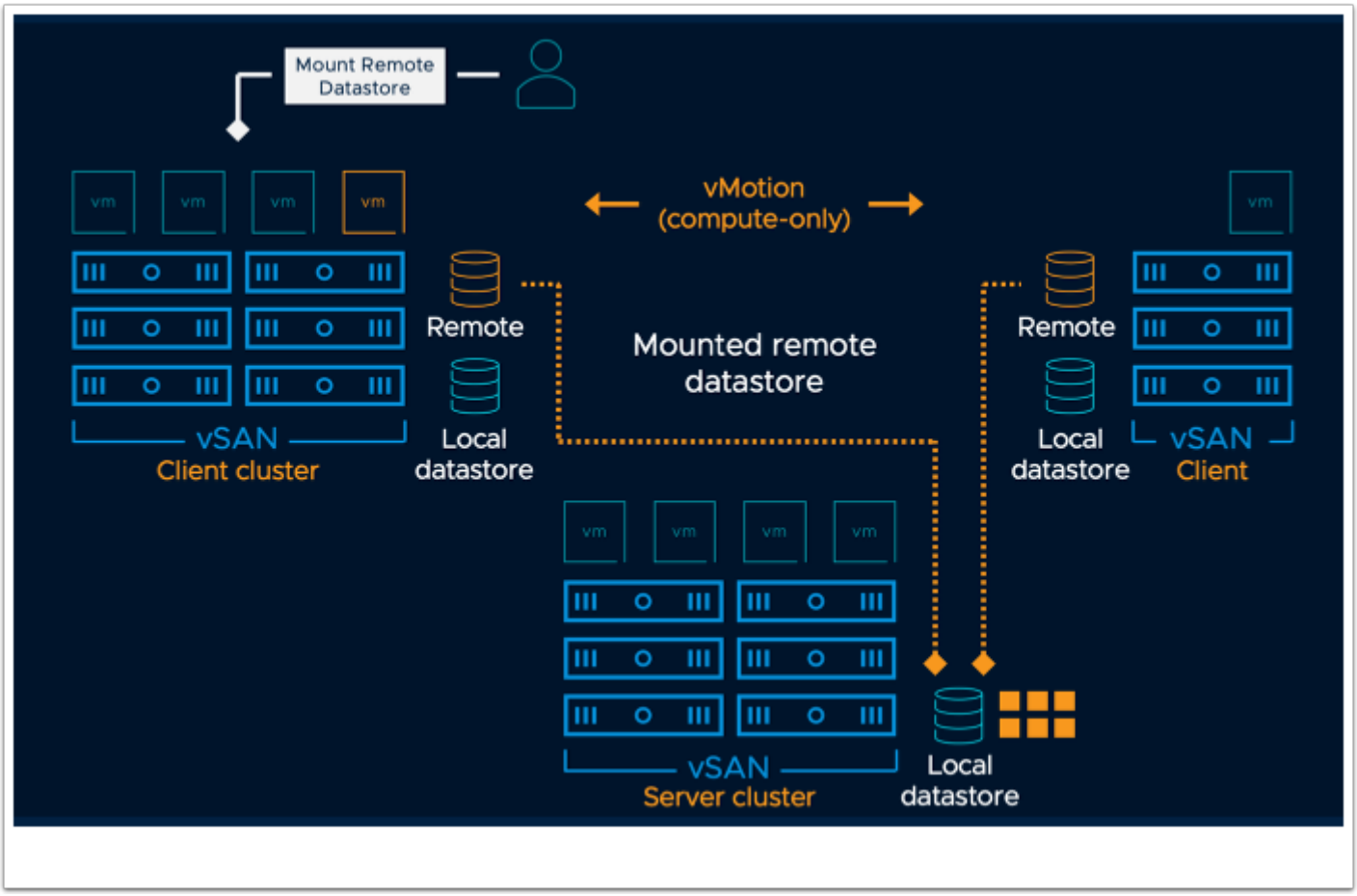
Module 7 - vSAN HCI Mesh

vSAN HCI Mesh Overview

Overview

HCI Mesh is a unique, software-based approach for disaggregation of compute and storage resources. HCI Mesh brings together multiple independent vSAN clusters for a native, cross-cluster architecture that disaggregates resources and enables utilization of stranded capacity. Simply, vSAN allows one or more vSAN clusters to remotely mount datastores from other vSAN clusters (servers) within vCenter inventory. This approach maintains the essence and simplicity of HCI by not fundamentally changing the existing HCI model or requiring specialized hardware. Now, a cluster with excess compute can mount excess storage from a remote vSAN cluster.

HCI Mesh works by using the existing vSAN VMkernel ports, and transport protocols end to end. Unique to VMware, HCI Mesh requires no specialized hardware. Customers can use any vSAN ReadyNode™ to share capacity across clusters. HCI Mesh is scalable. It can support up to 64 hosts across clusters in a mesh. A client cluster can mount up to five remote datastores.



NOTE: HCI Mesh does not support the following:

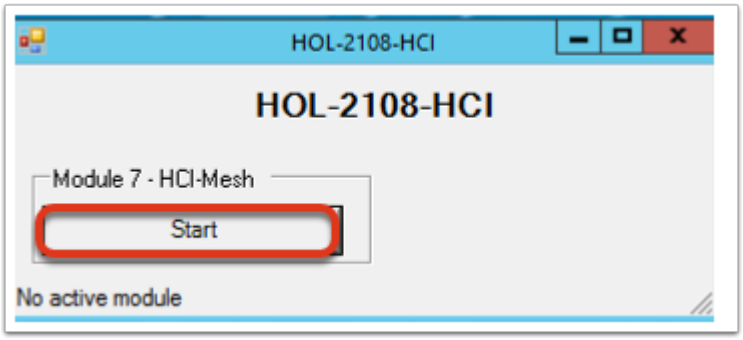
- vSAN data-in-transit encryption
- vSAN Stretched clusters
- vSAN 2-node clusters
- Remote provisioning of vSAN file share, iSCSI volumes, or CNS persistent volumes
- Air-gapped vSAN networks or clusters using multiple vSAN VMkernel ports

vSAN HCI Mesh Configure

Prep-work

We will use a Module Switcher for this lab, please double click on HOL-2008-01 HCI icon to launch the switcher.

Module Switcher



Click on **Start** to turn on the switcher.

The module switcher enables another vSAN Cluster besides the one we did in Module 1. Wait til it runs.

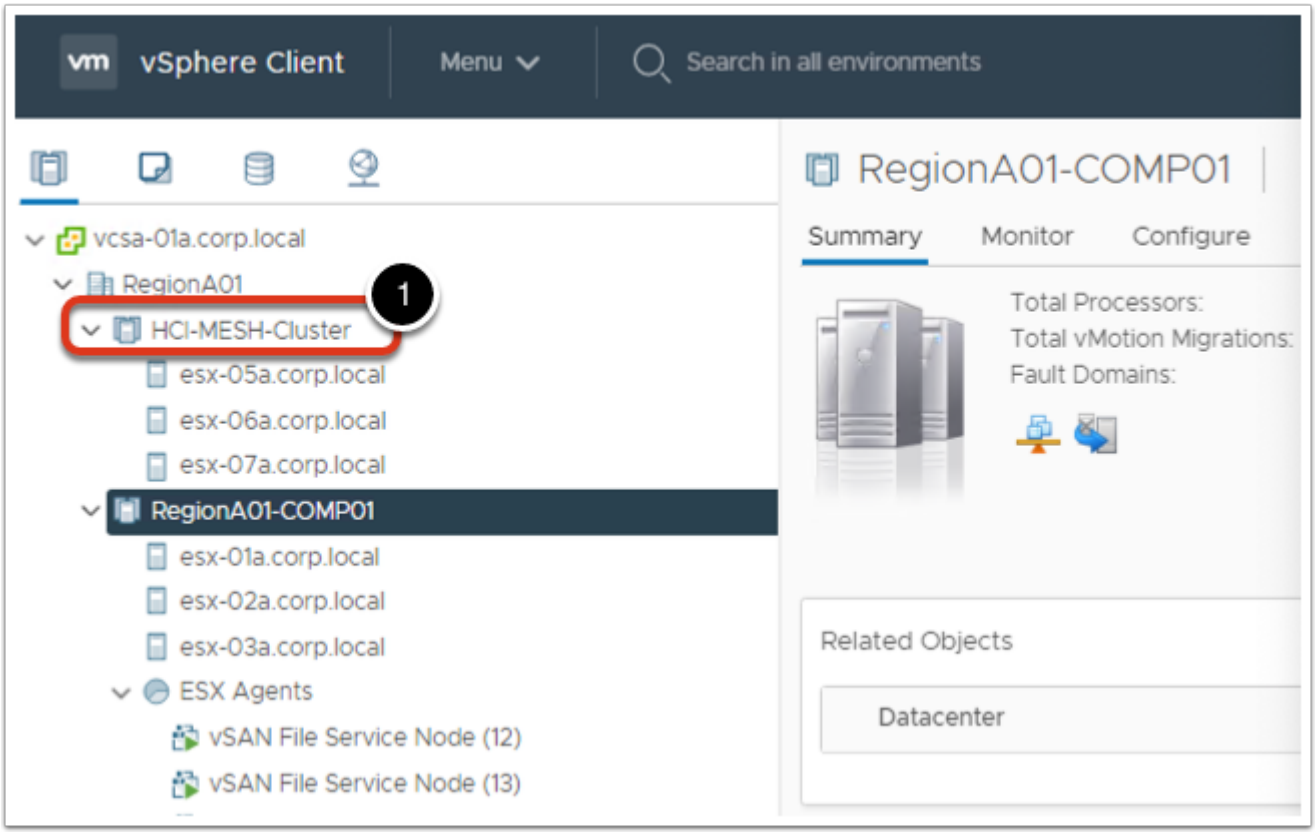
Module Switcher

```
Administrator: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
                    00000766d686261333a303a30/


VMHost             : esx-07a.corp.local
Uuid               : 00000000000766d686261323a303a30
DiskGroupType      : AllFlash
DiskFormatVersion  : 13
IsMounted          : True
ExtensionData      : VMware.Vim.VsanHostDiskMapping
Name               : Disk group (00000000000766d686261323a303a30)
Id                 : HostSystem-host-2003/00000000000766d686261323a303a30
Uid                : /VIServer=corp\administrator@vcsa-01a.corp.local:443/VsanDiskGroup=HostSystem-host-2003&slash;00000
                     00000766d686261323a303a30/

Wait-Task : 2/18/2021 4:57:40 PM        Wait-Task              The operation for the entity "vCLS (4)" failed with the
following message:
"The attempted operation cannot be performed in the current state (Powered on)."
At C:\HOL\ModuleSwitcher\HOL-2108-HCI\Module01.ps1:155 char:2
+     Wait-Task (Get-Task) | Out-Null
+     ~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (:) [Wait-Task], InvalidPowerState
    + FullyQualifiedErrorId : Client20_TaskServiceImpl_CheckServerSideTaskUpdates_OperationFailed,VMware.VimAutomation
   .Common.Commands.Cmdlets.WaitTask

True

DatacenterId                   : Datacenter-datacenter-1001
Datacenter                     : RegionA01
ParentFolderId                 : Folder-group-s1004
ParentFolder                   : datastore
DatastoreBrowserPath           : vmstores:\vcsa-01a.corp.local@443\RegionA01\HCI-Mesh-Datastore
FreeSpaceMB                    : 72223
CapacityMB                     : 73680
Accessible                     : True
Type                           : vsan
StorageIOControlEnabled        : False
CongestionThresholdMillisecond :
State                          : Available
ExtensionData                  : VMware.Vim.Datastore
CapacityGB                     : 71.953125
FreeSpaceGB                    : 70.5302734375
Name                           : HCI-Mesh-Datastore
Id                             : Datastore-datastore-16009
Uid                            : /VIServer=corp\administrator@vcsa-01a.corp.local:443/Datastore=Datastore-datastore-160
                                 09/

Disconnecting from vCenter(s)
Successfully Started Module 3
02/18/2021 16:57:45 Finished. Runtime was 3 minutes.
Press Enter to continue...:
```

You can close those this window.

You might need to refresh the vSphere Client to see the 2nd vSAN Cluster.

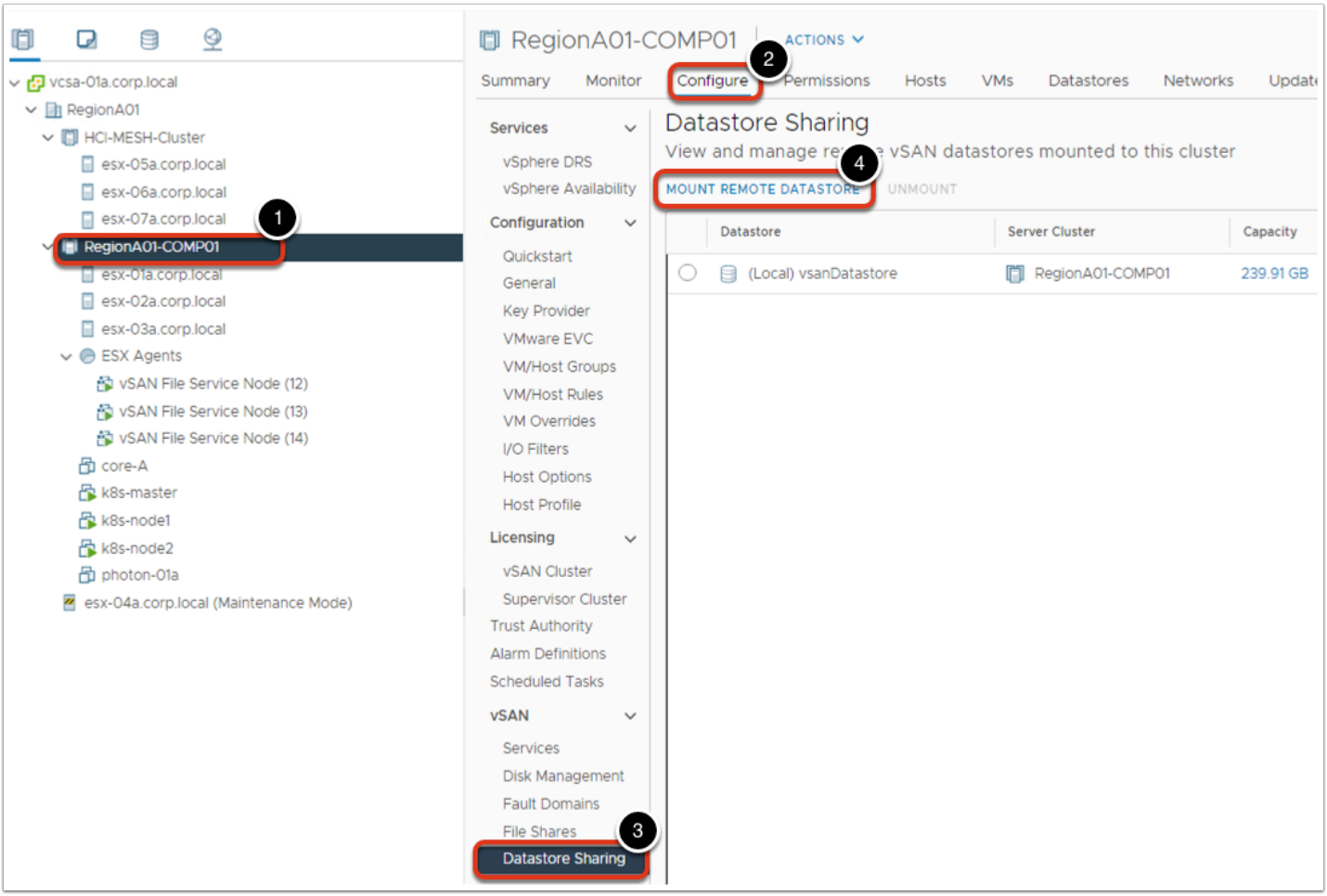
Configure HCI Mesh



1. Expand 2nd vSAN Cluster, **HCI-MESH-Cluster**.

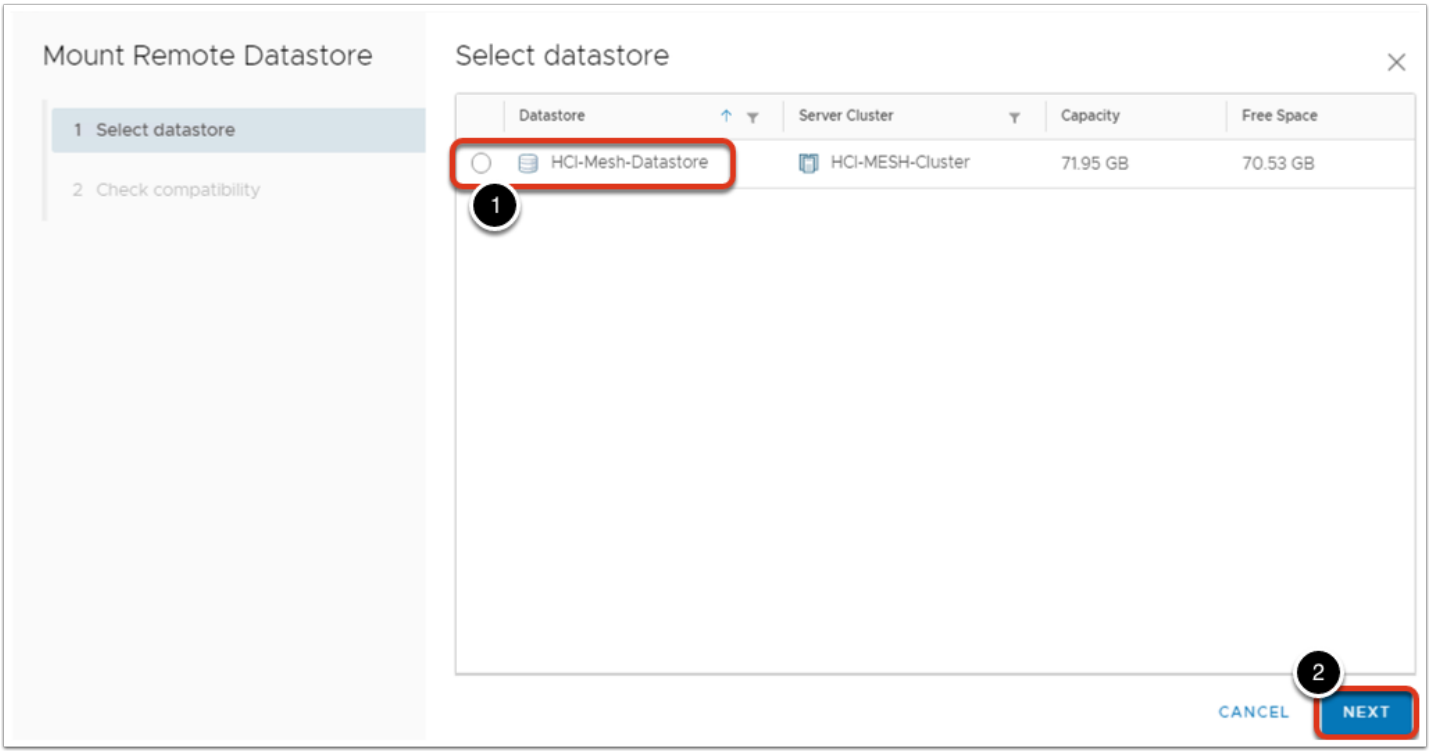
This will be the remote vSAN cluster datastore which contains 3 hosts, esx-05a.corp.local, esx-06a.corp.local and esx07a.corp.local.

Configure Remote Datastore



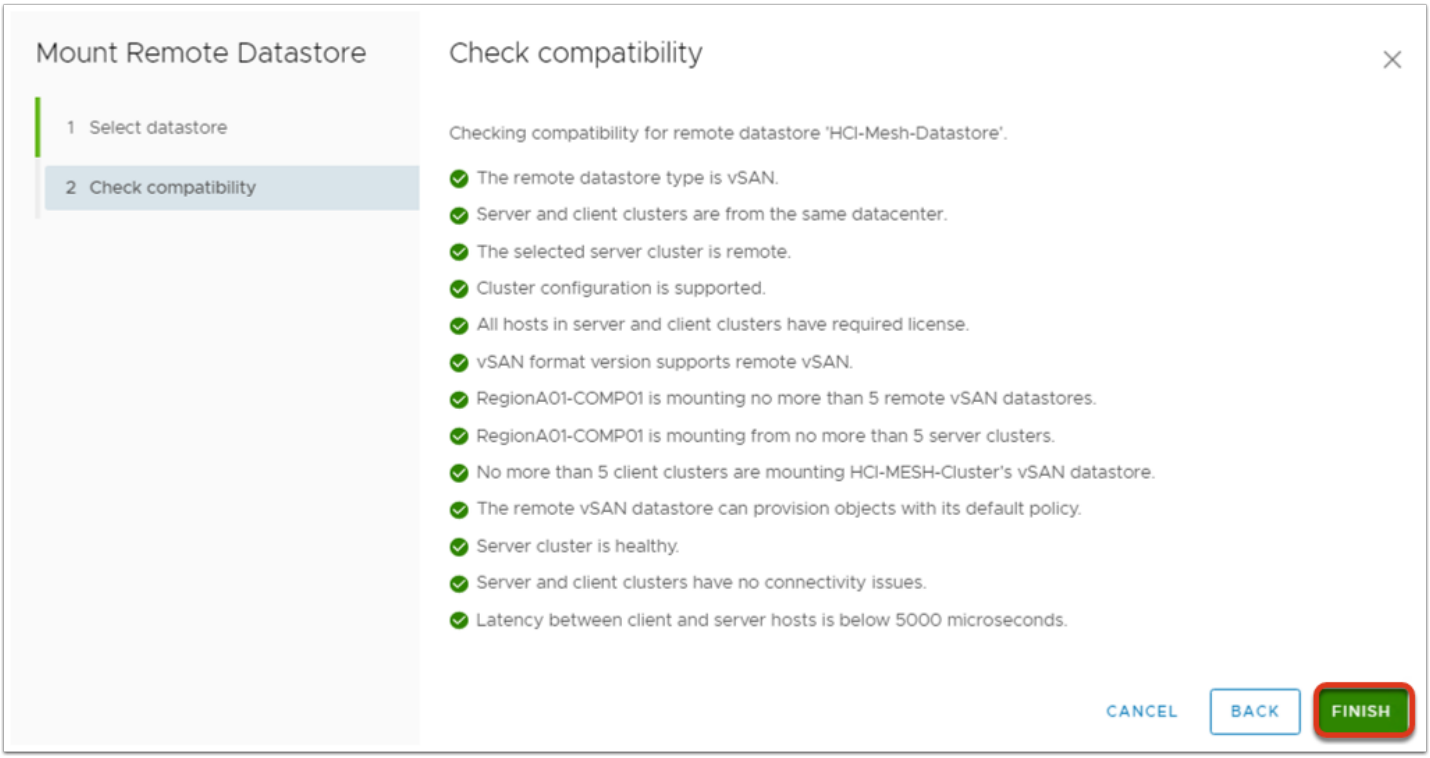
1. Select vSAN Cluster, **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > Datastore Sharing**
4. Click **Mount Remote Datastore**

Mount Remote Datastore



1. Check **HCI-Mesh-Datastore**
2. Click **Next**

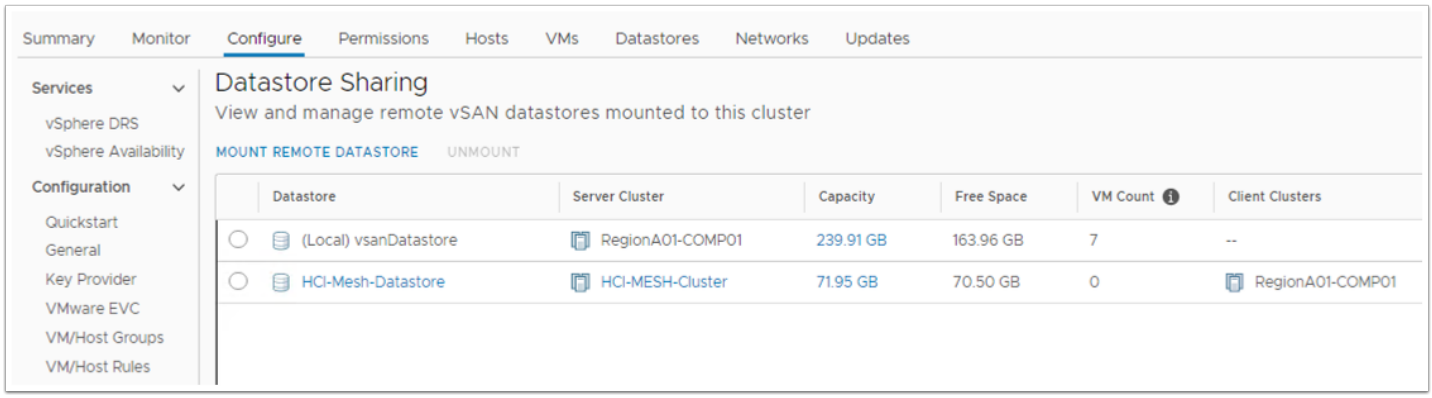
Mount Remote Datastore - Check compatibility



Click **Finish**

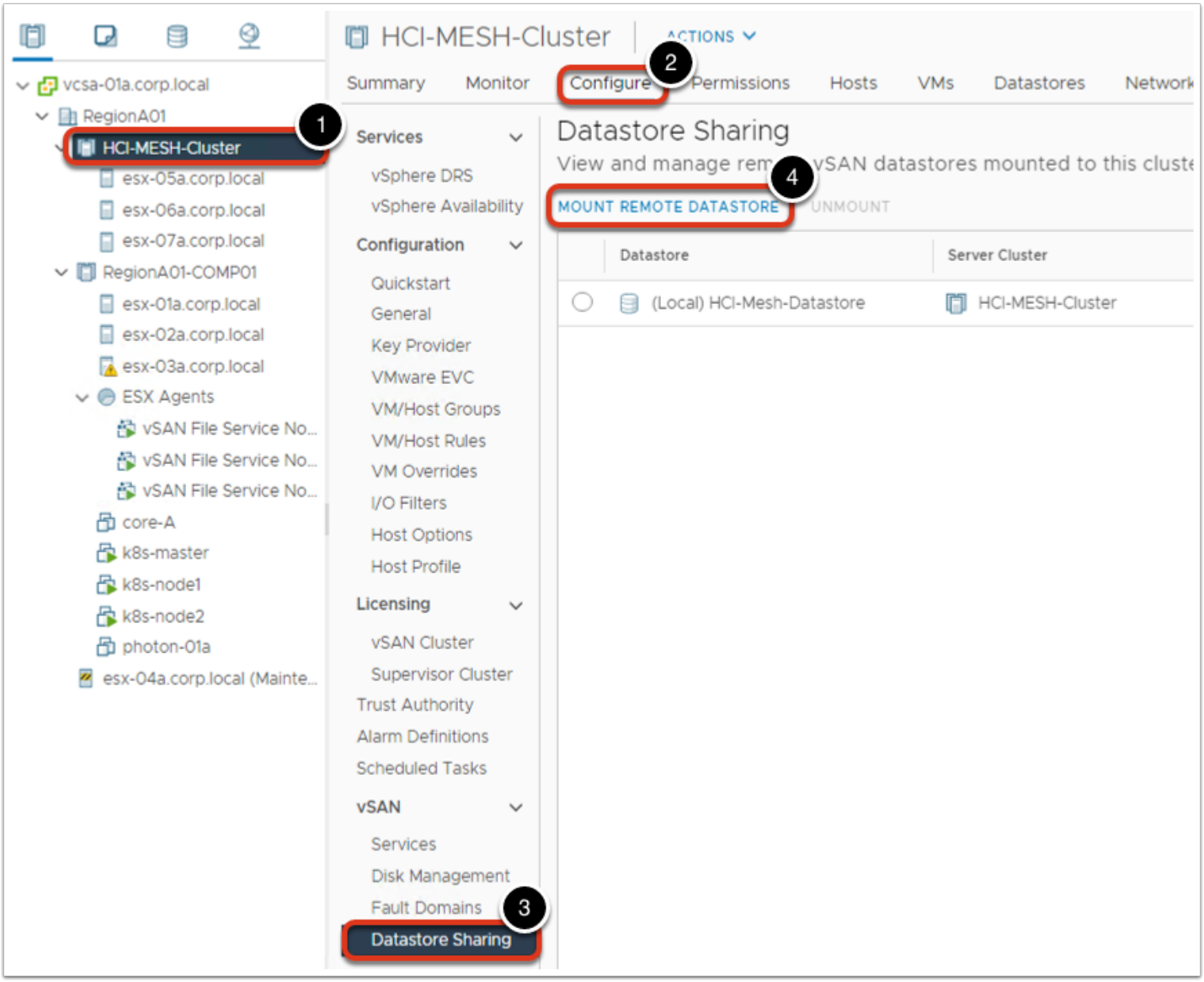
As you can see, it checks to make sure the remote vSAN datastore is compatible, healthy, and meets the requirements. It will take a moment to finish.

Mount Remote Datastore



You can see that there are two datastores, **vsanDatastore** and **HCI-Mesh-Datastore.** vSAN Cluster, **RegionA01-COMP01,** has a local vSAN datastore as well as access to the remote “HCI-MESH-Cluster” vSAN datastore.

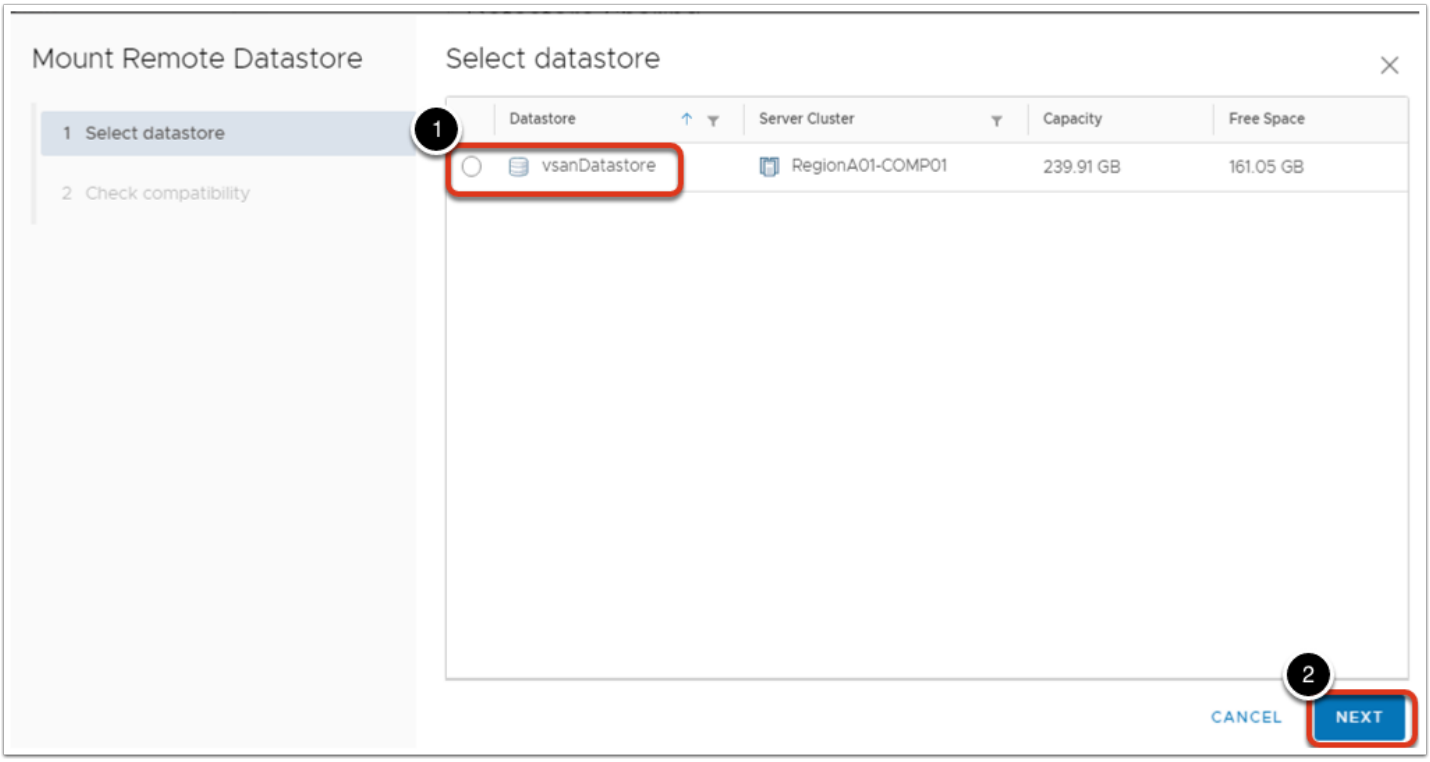
Mount Another Datastore



We are going to mount the other vSAN Cluster, **HCI-MESH-Cluster**, local datastore to the remote datastore of vSAN Cluster, **RegionA01-COMP01**

1. Select vSAN Cluster, **HCI-MESH-Cluster**
2. Select **Configure**
3. Select vSAN > **Datastore Sharing**
4. Click on **MOUNT REMOTE DATASTORE**

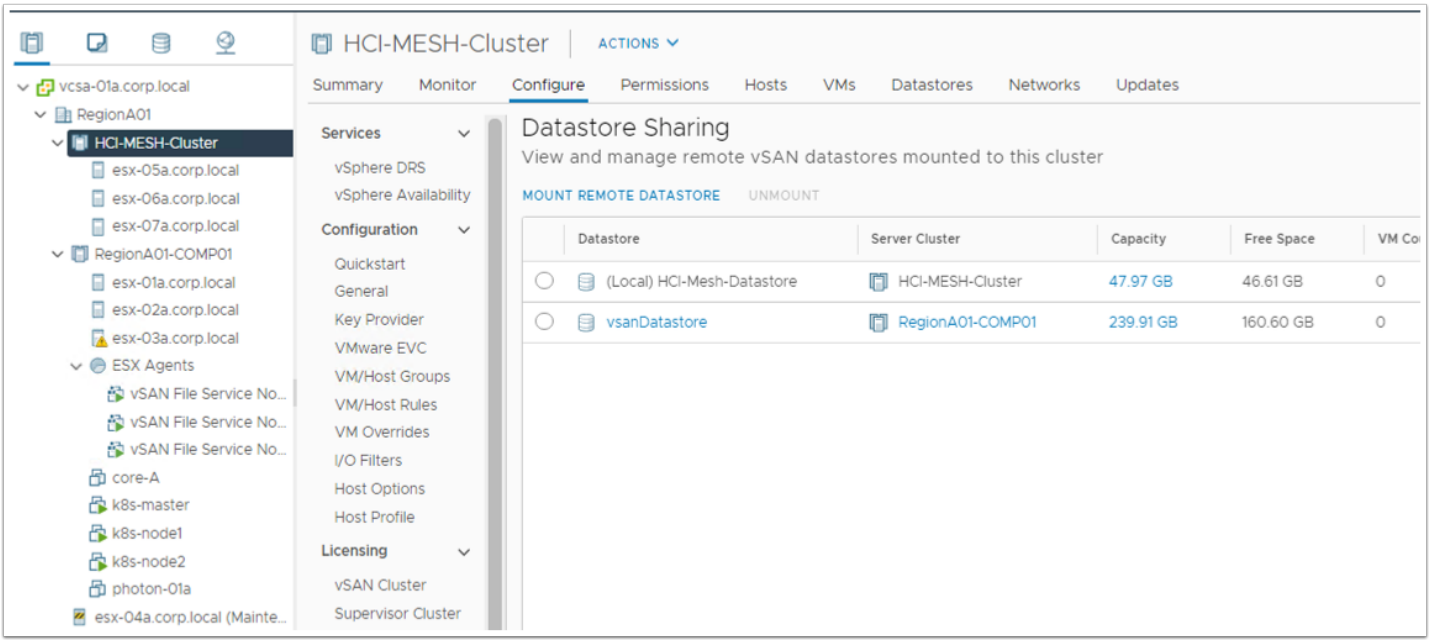
Mount Remote Datastore



1. Click **vsanDatastore**
2. Click **Next**

Click **Finish** on the Check compatibility page.

Remote Datastore

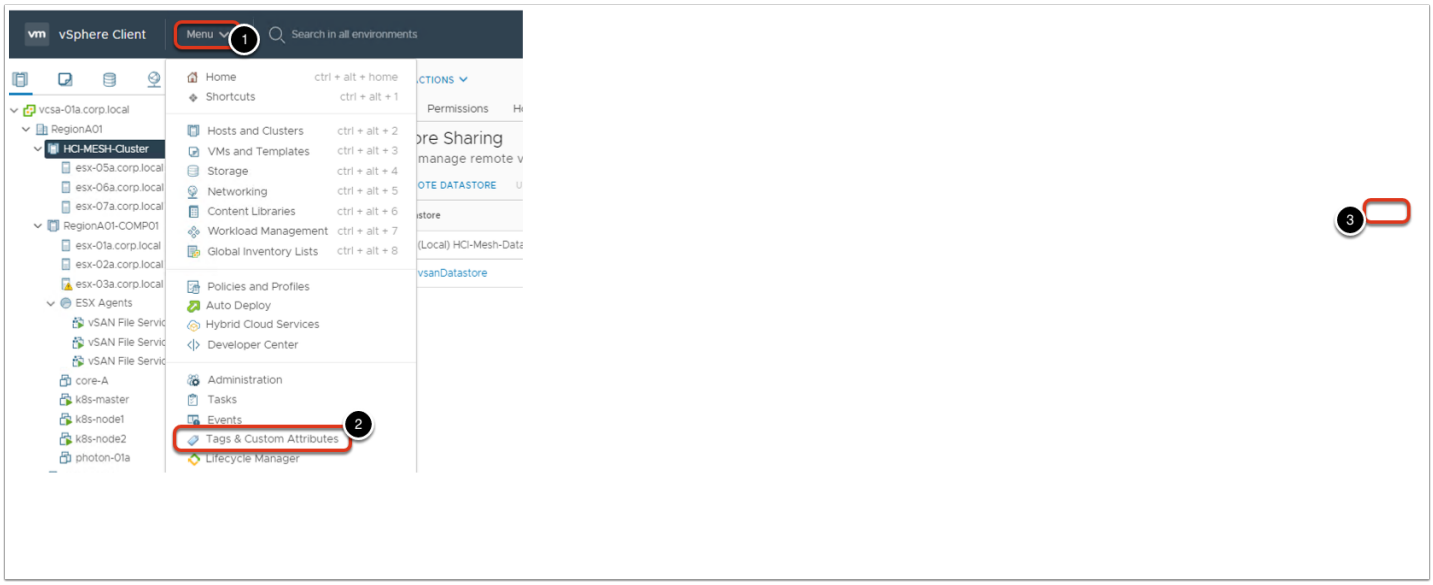


You can see that RegionA01-COMP01 datastore, **vsanDatastore** is the remote datastore for the HCI-MESH-Cluster.

Create Tag

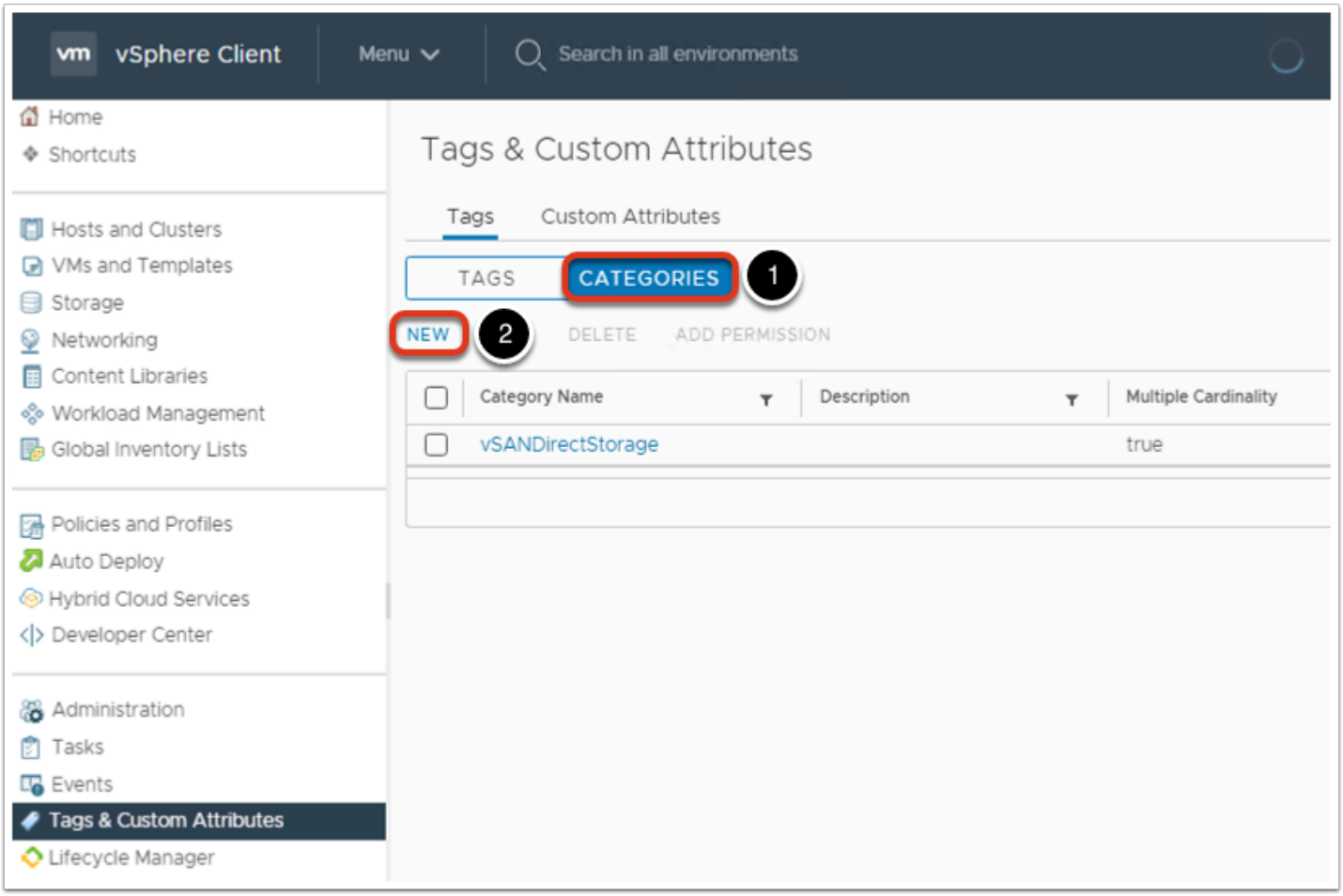
Creating tag is not special but since we have HCI Mesh enable, using tag is important.

Creating a vSAN Policy to include a Tag



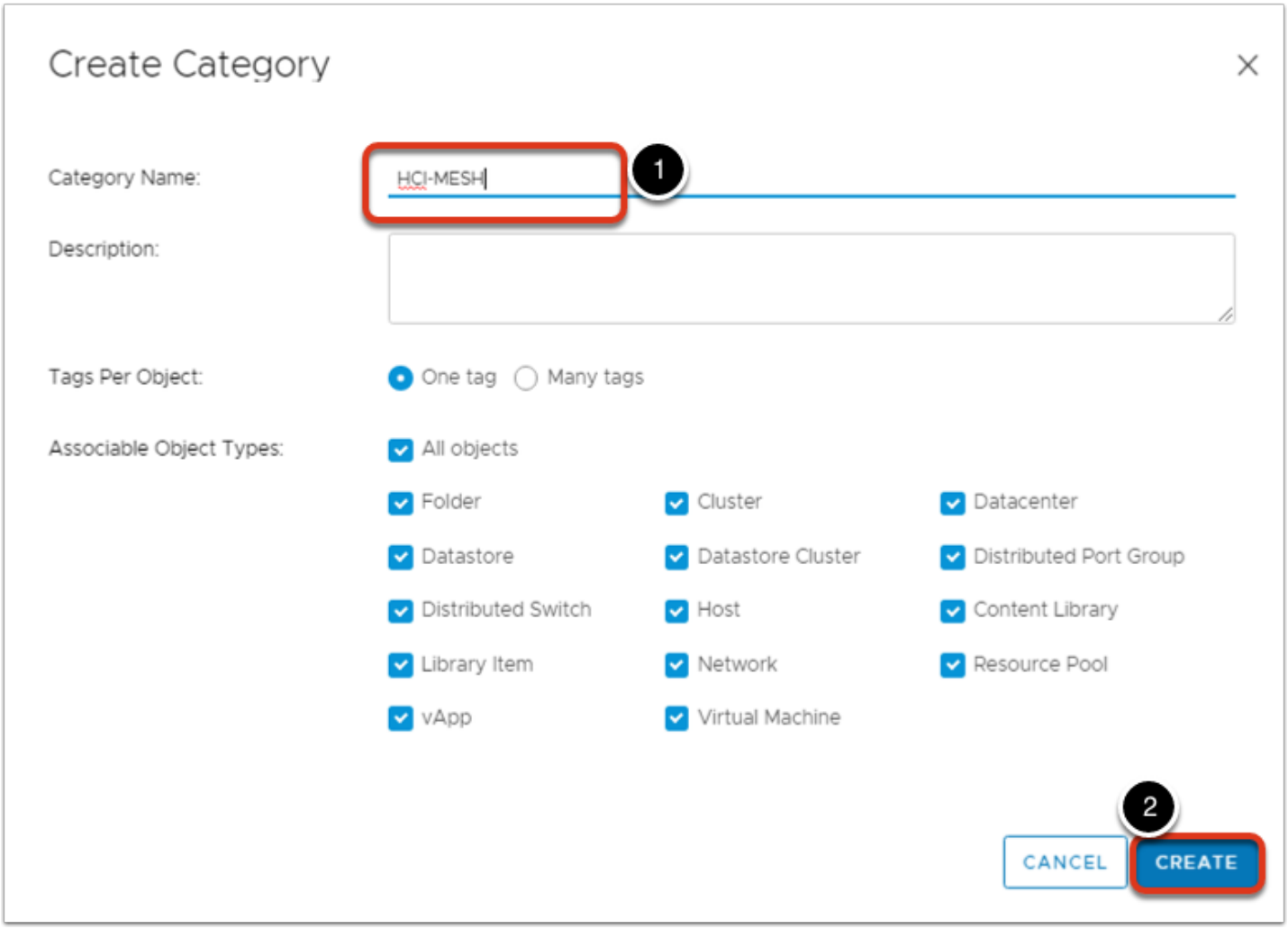
1. Select **Menu**
2. Select **Tags & Custom Attributes**

Create Categories



1. Select **CATAGORIES**
2. Click on **NEW**

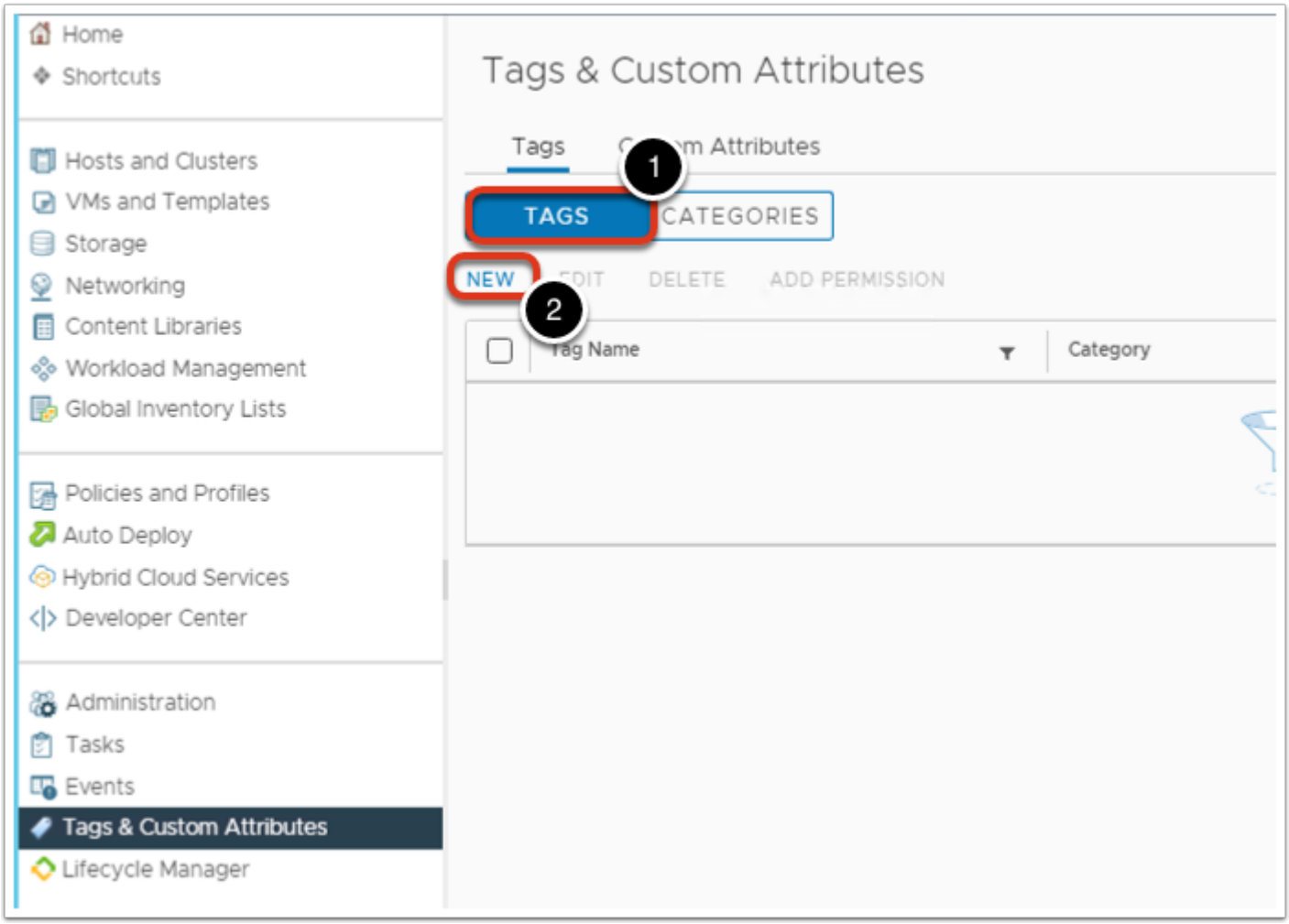
Create Categories



To begin, we will create a new tag category which is called **HCI-Mesh**.

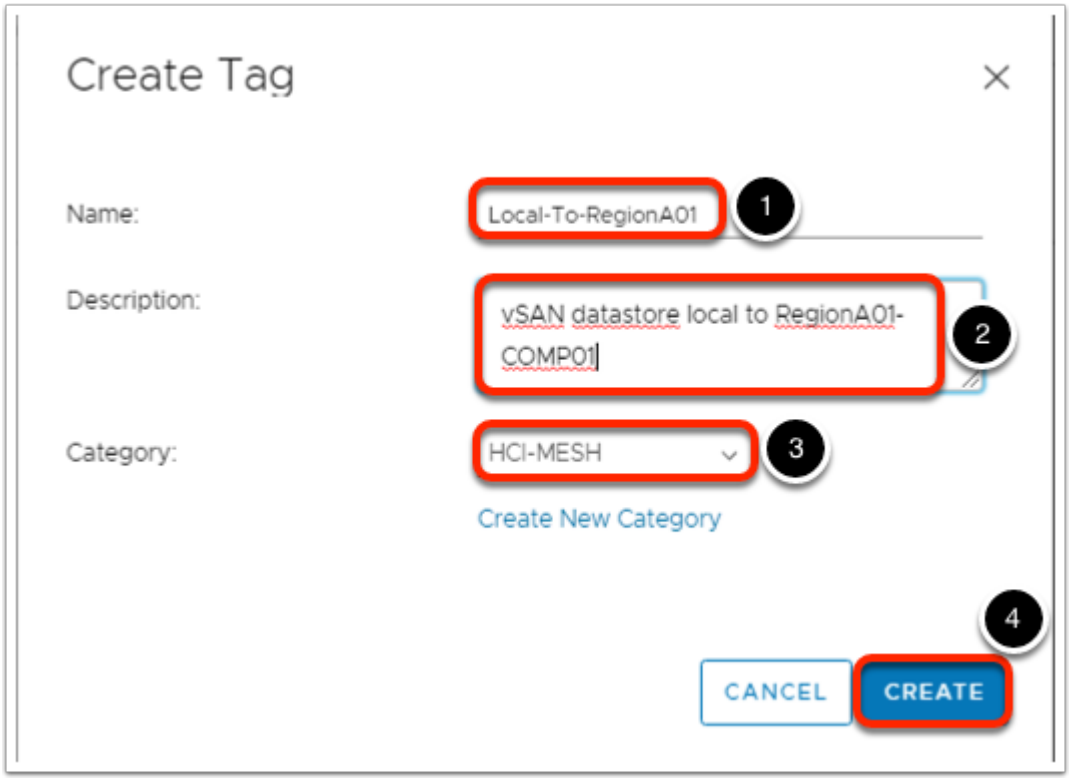
1. Enter Category Name: **HCI-MESH**
2. Click on **CREATE**

Create Tags



1. Select **TAGS**
2. Click **NEW**

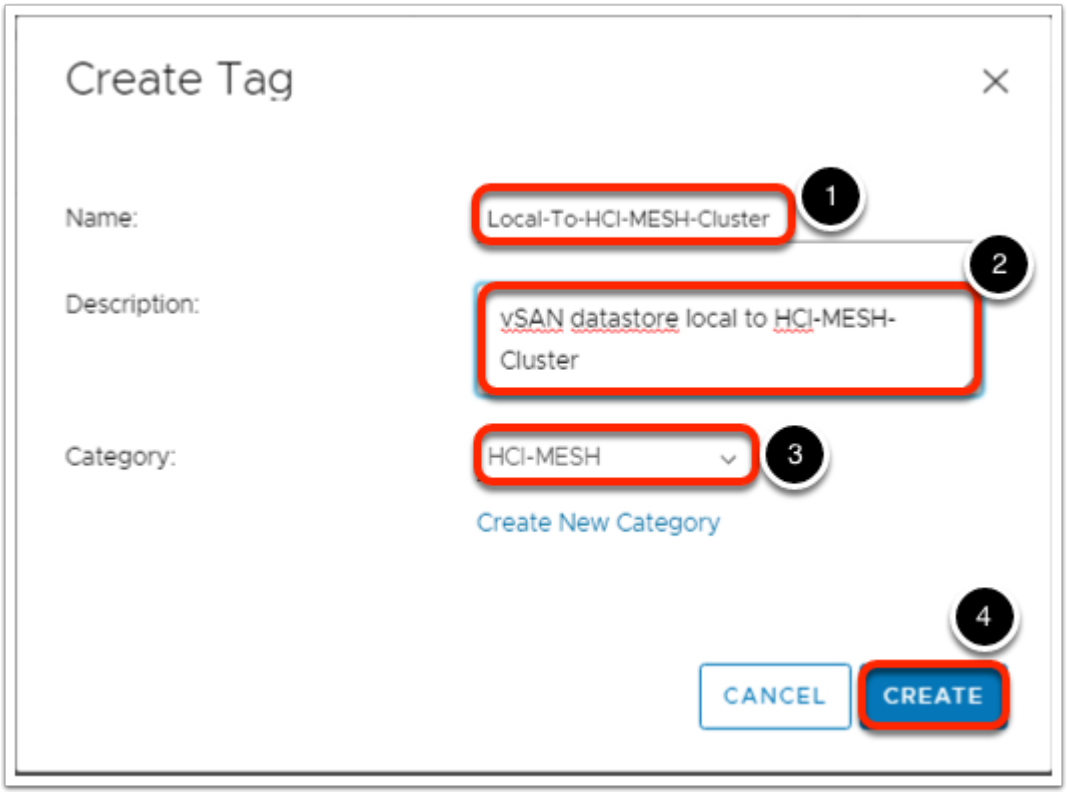
Create Tags (local to RegionA01-COMP01 Cluster)



In this Tag category, we will create 2 tags. One will be used to tag the vSAN datastore on the “RegionA01-COMP01” cluster, and the other will be used to tag the vSAN datastore in the “HCI-MESH-Cluster” cluster.

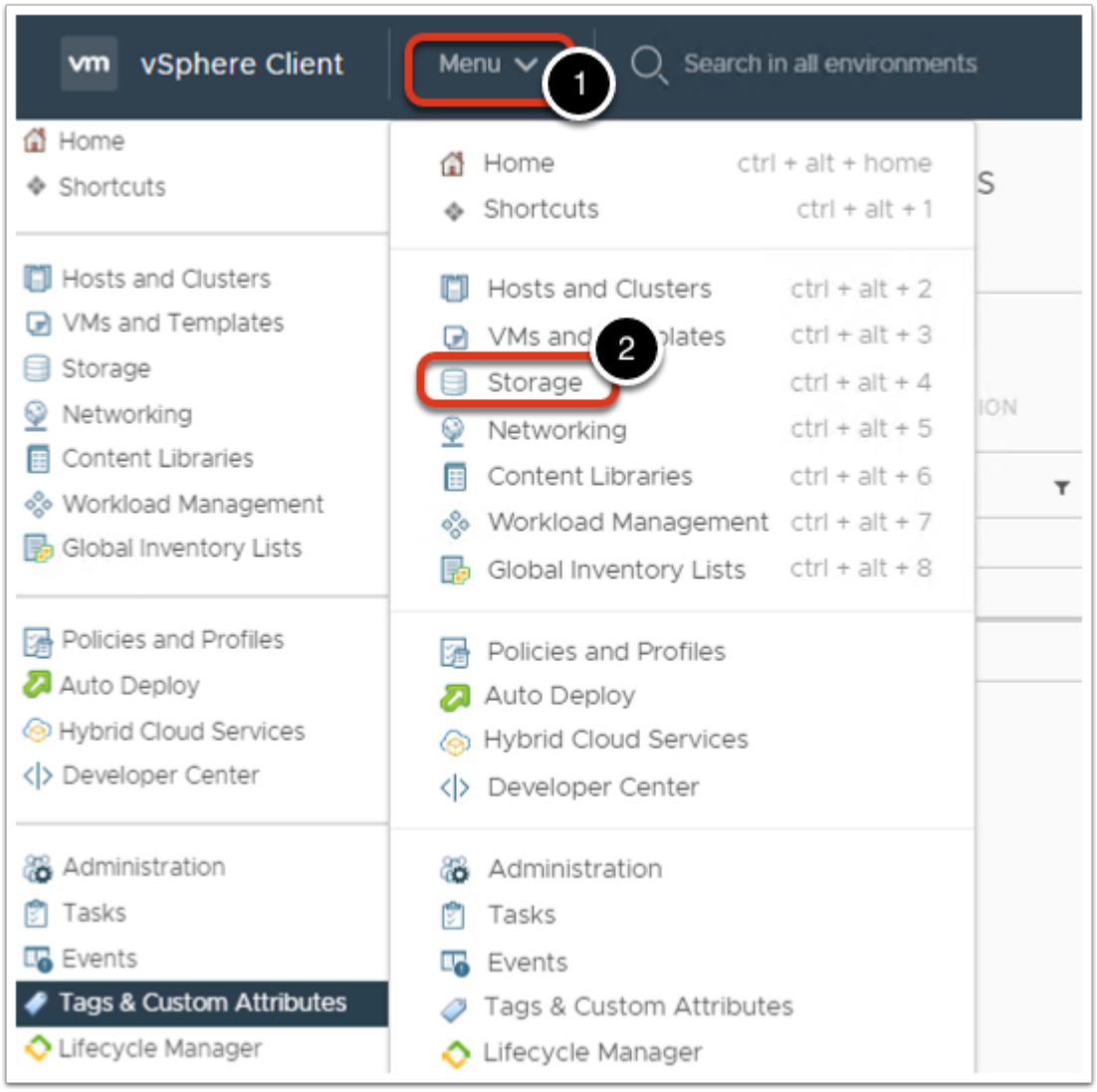
1. Name: **Local-to-RegionA01**
2. Description: **vSAN datastore local to RegionA01-COMP01**
3. Category: **HCI-MESH**
4. Click **CREATE**
5. We will then create a second Tag for the HCI-MESH cluster, click on **NEW** (not shown)

Create Tags (local to HCI-MESH Cluster Cluster)



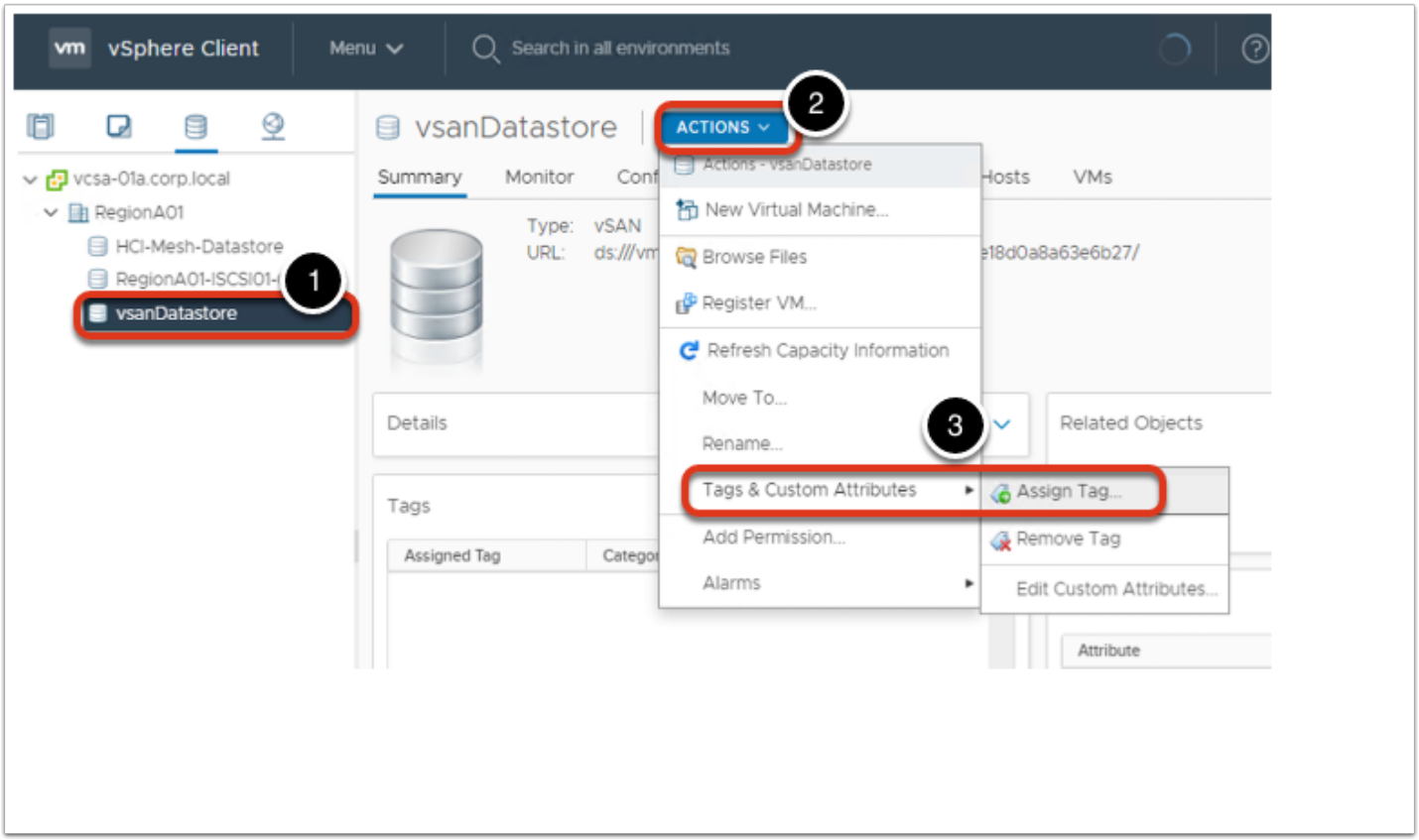
1. Name: **Local-to-HCI-MESH-Cluster**
2. Description: **vSAN datastore local to HCI-MESH-Cluster**
3. Category: **HCI-MESH**
4. Click **CREATE**

Assign Tags to Datastore



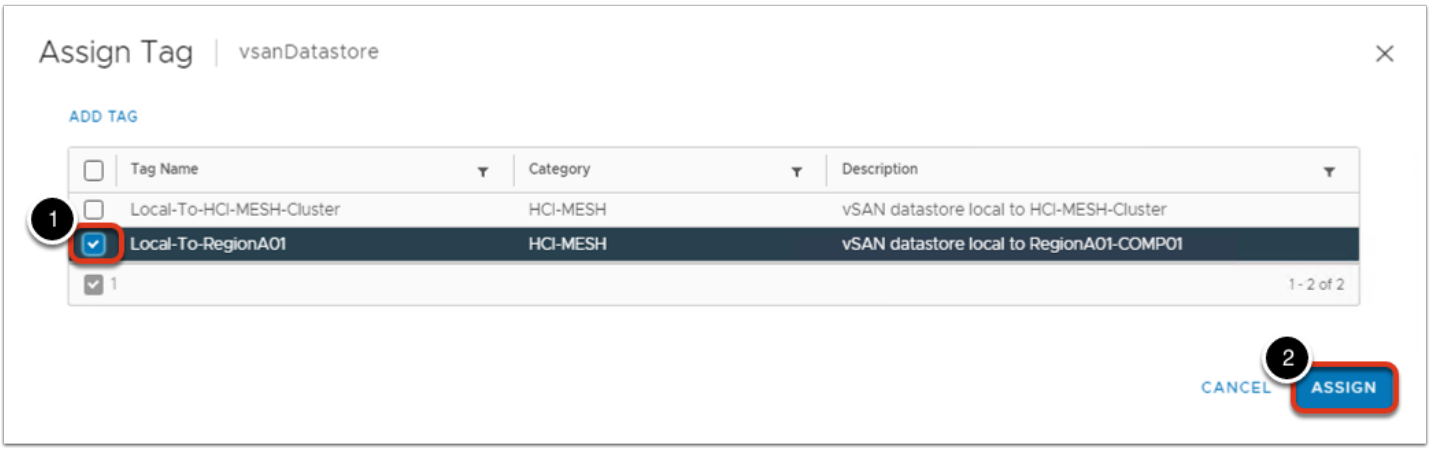
1. Select **Menu**
2. Select **Storage**

Assign Tags to Datastore



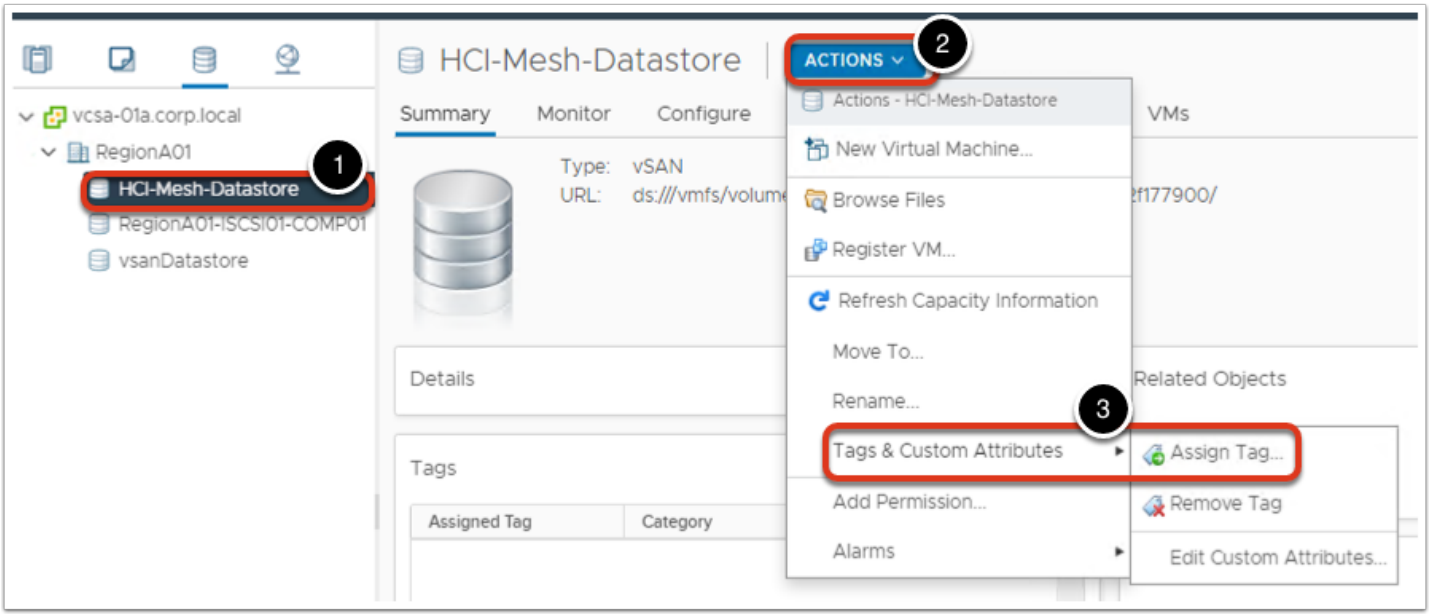
1. Select storage, **vsanDatastore**
2. Select **Actions**
3. Select **Tags & Custom Attributes > Assign Tags**

Assign Tags



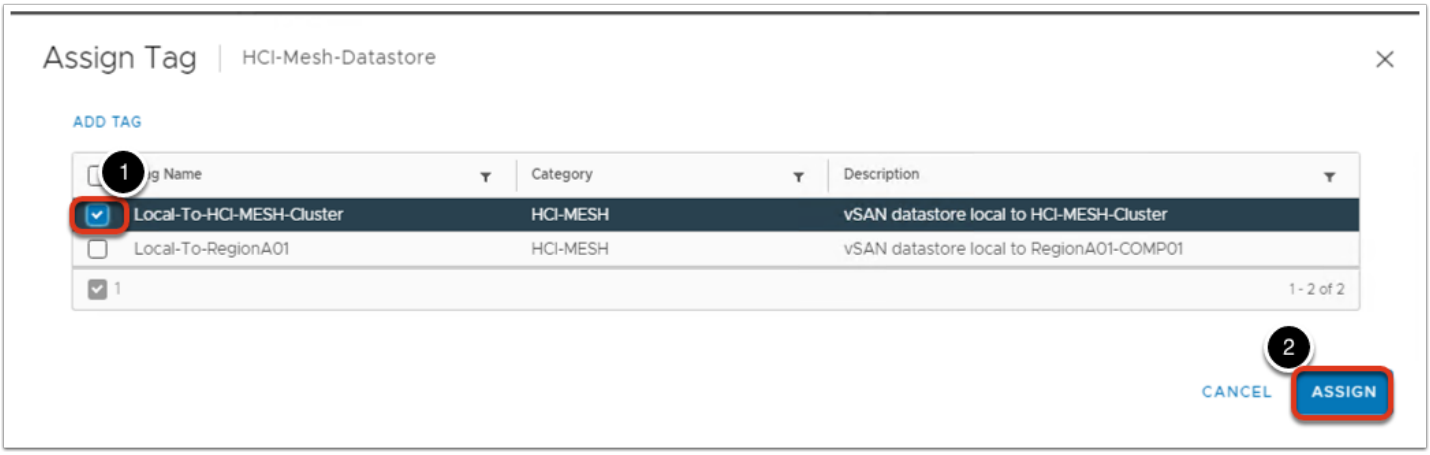
1. Click **Local-To-RegionA01**
2. Click **ASSIGN**

Assign other Tags to datastore



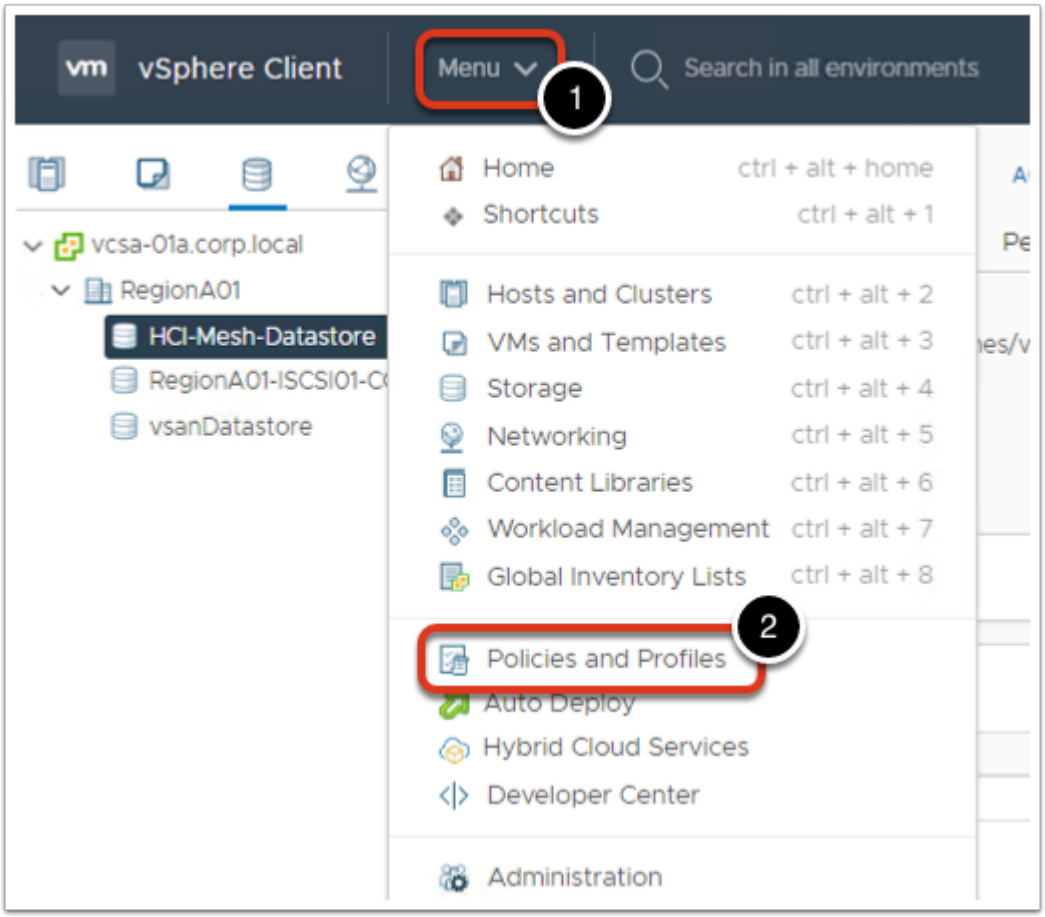
1. Select storage, **HCI-Mesh-Datastore**
2. Select **Actions**
3. Select **Tags & Custom Attributes > Assign Tags**

Assign Tags to Datastore



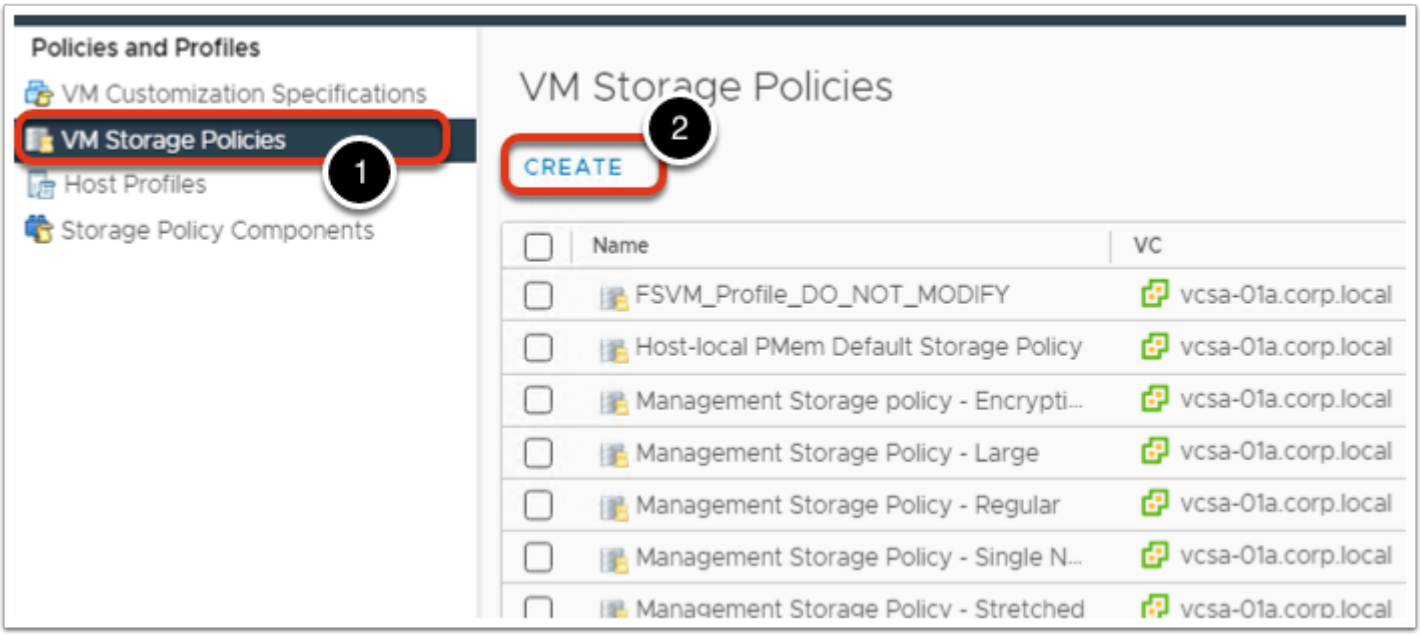
1. Click **Local-To-HCI-MESH-Cluster**
2. Click **ASSIGN**

Create Storage Policy with Tags



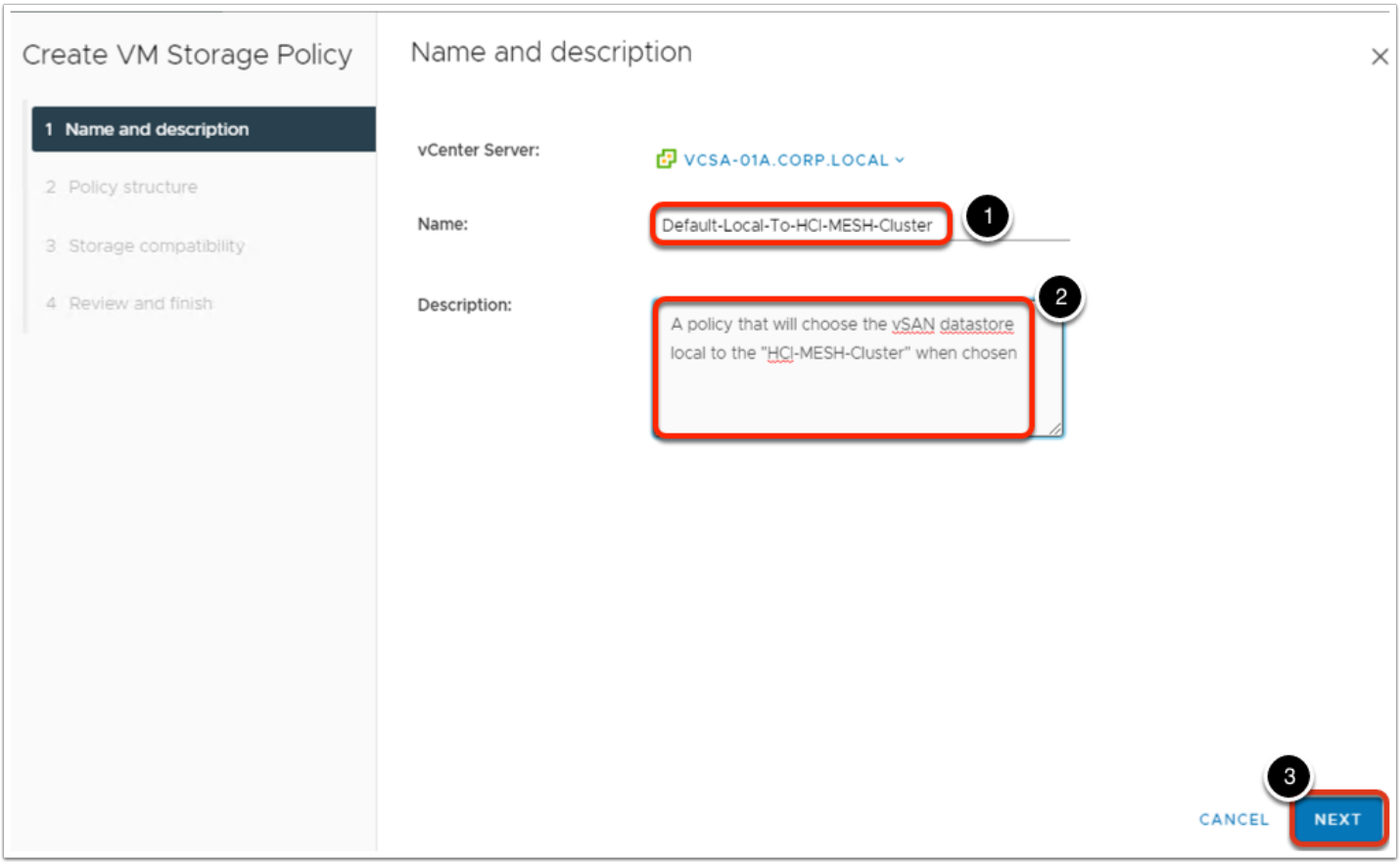
1. Select **Menu**
2. Select **Policies and Profiles**

Create Storage Policy with Tags



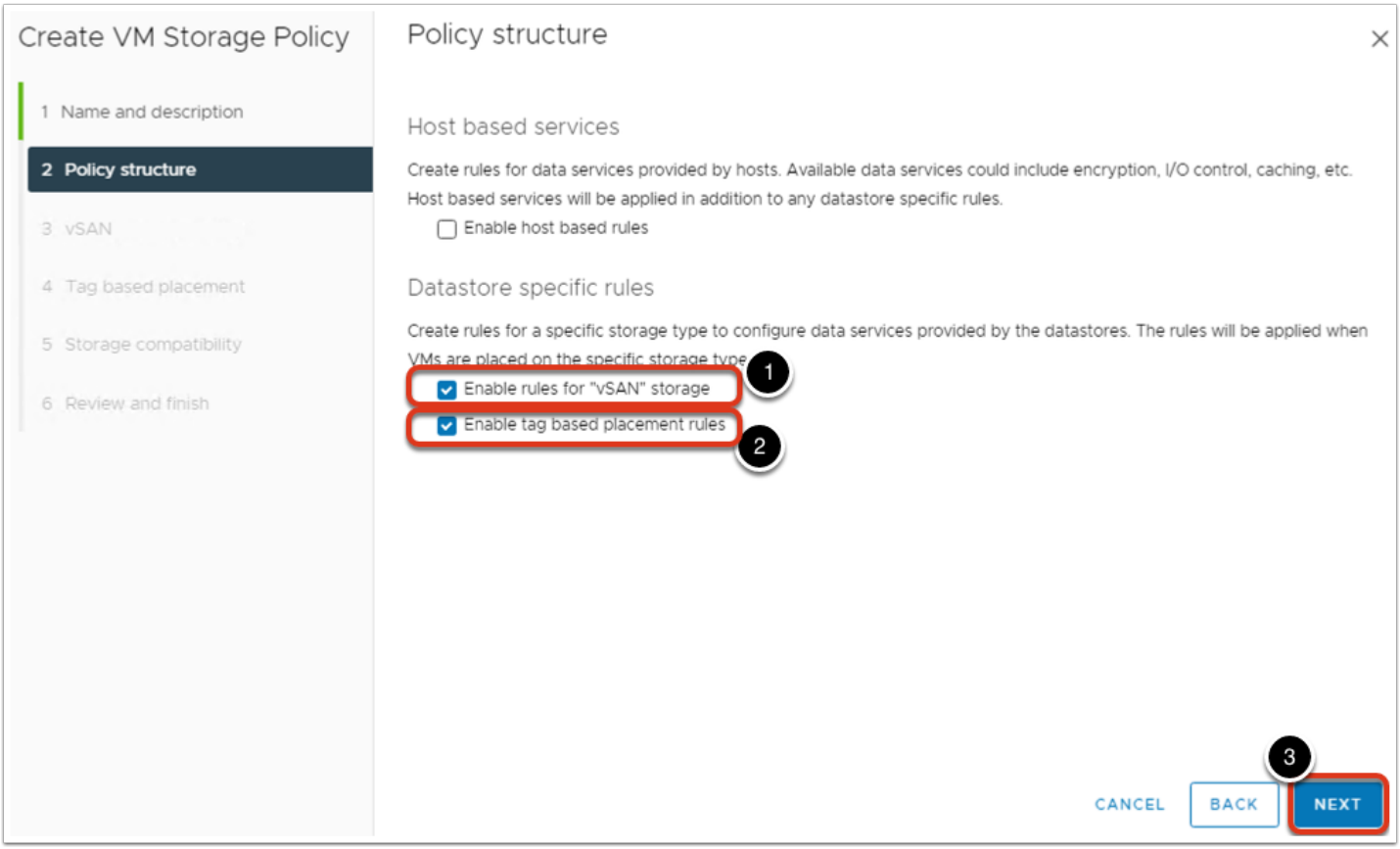
1. Select **VM Storage Policies**
2. Click **CREATE**

Create VM Storage Policy - Name & Description



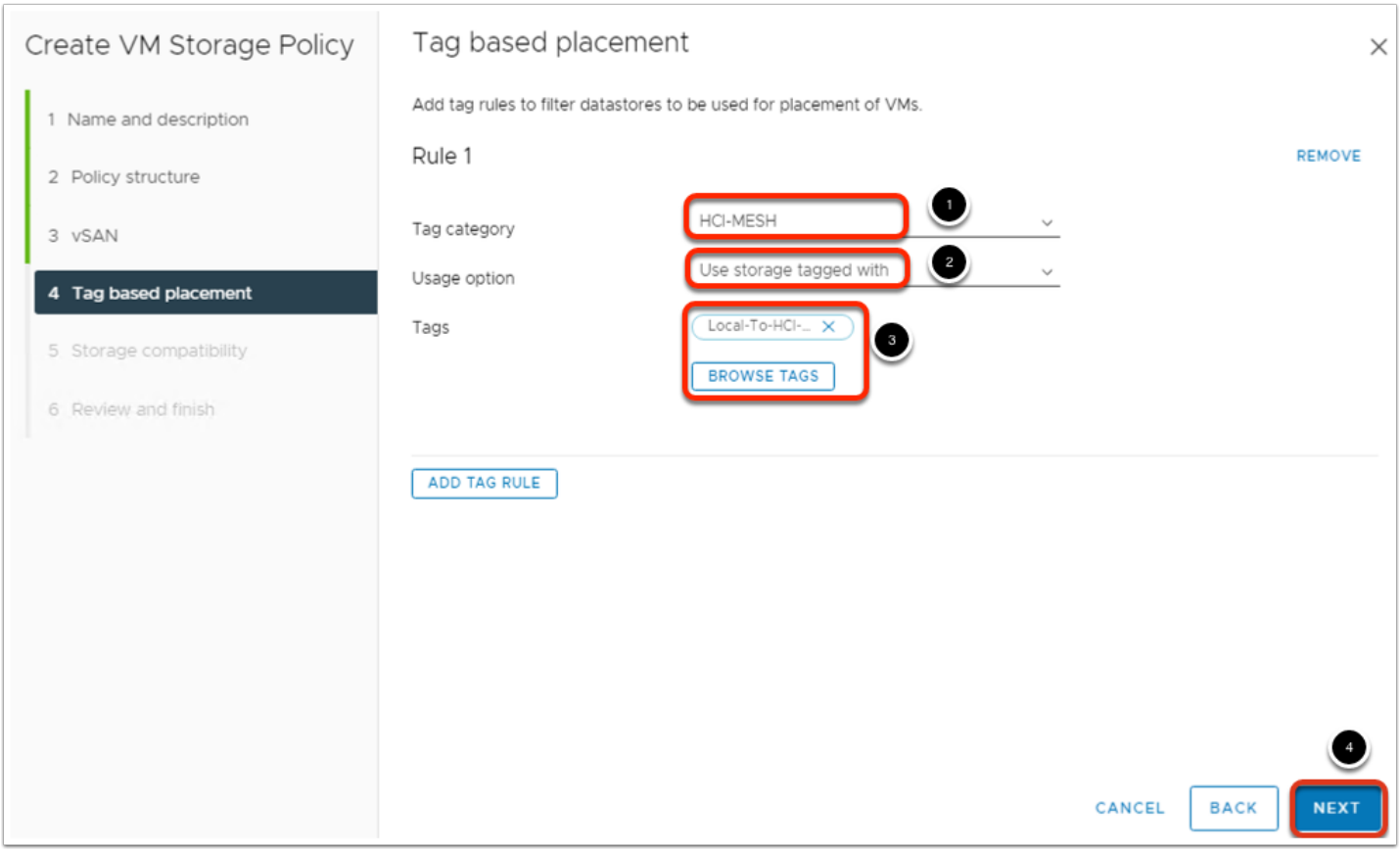
1. Enter Name: **Default-Local-To-HCI-MESH-Cluster**
2. Description (optional): **A policy that will choose the vSAN datastore local to the "HCI-MESH-Cluster" when chosen**
3. Click **NEXT**

Create VM Storage Policy - Policy Structure



1. Click **Enable rules for "vSAN" storage**
2. Click **Enable tag based placement rules**
3. Click **NEXT -** then for next section 'vSAN' click **NEXT** again

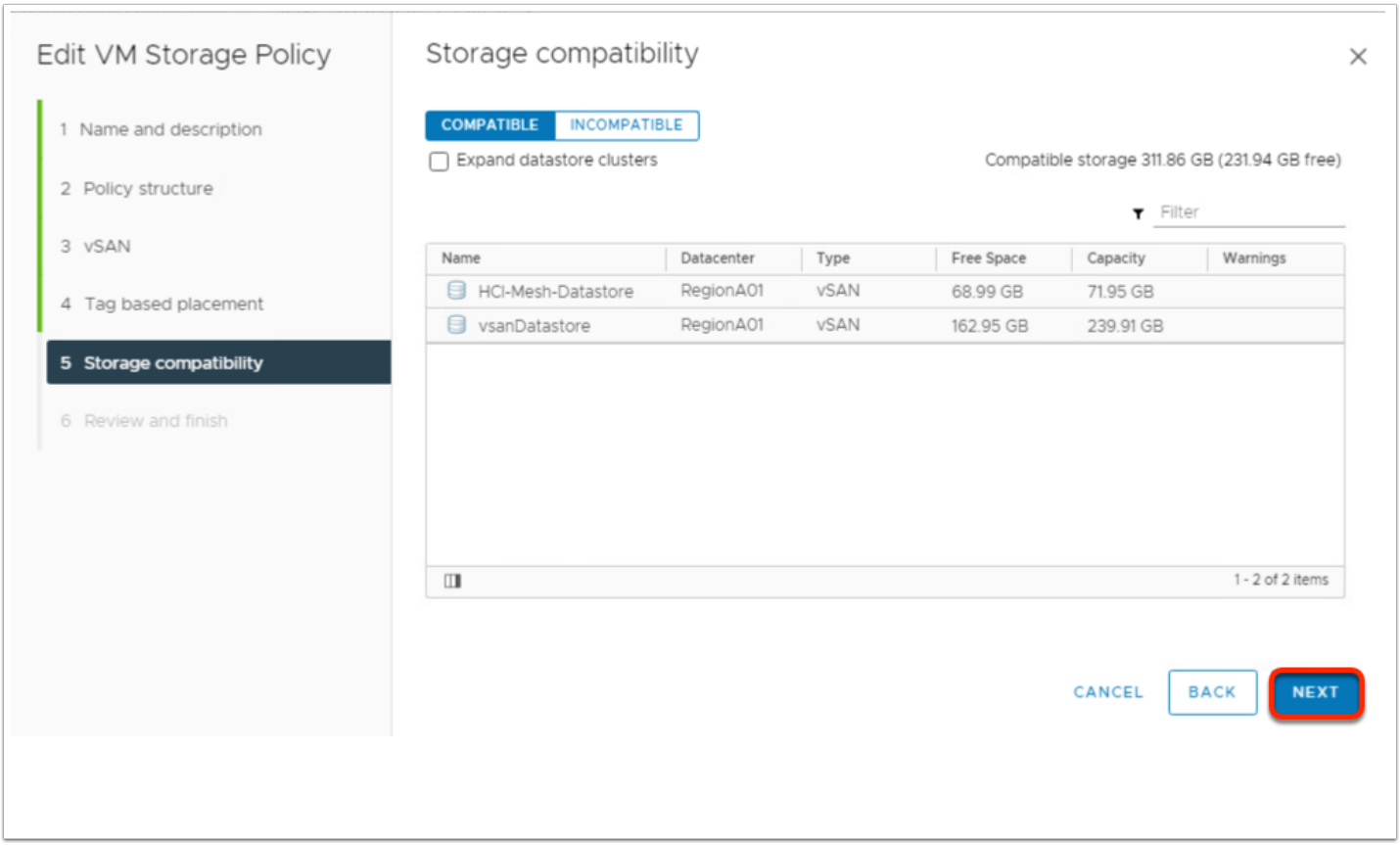
Create VM Storage Policy - Tag based placement



You can set the Tag category in the previous page. This is just an alternative way.

1. Select Tag category: **HCI-MESH**
2. Select Usage option: **Use storage tagged with**
3. Browse Tags: Select **Local-To-HCI-MESH-Cluster** and click **OK** (pop-up window to chose the tag not picture)
4. Click **NEXT**

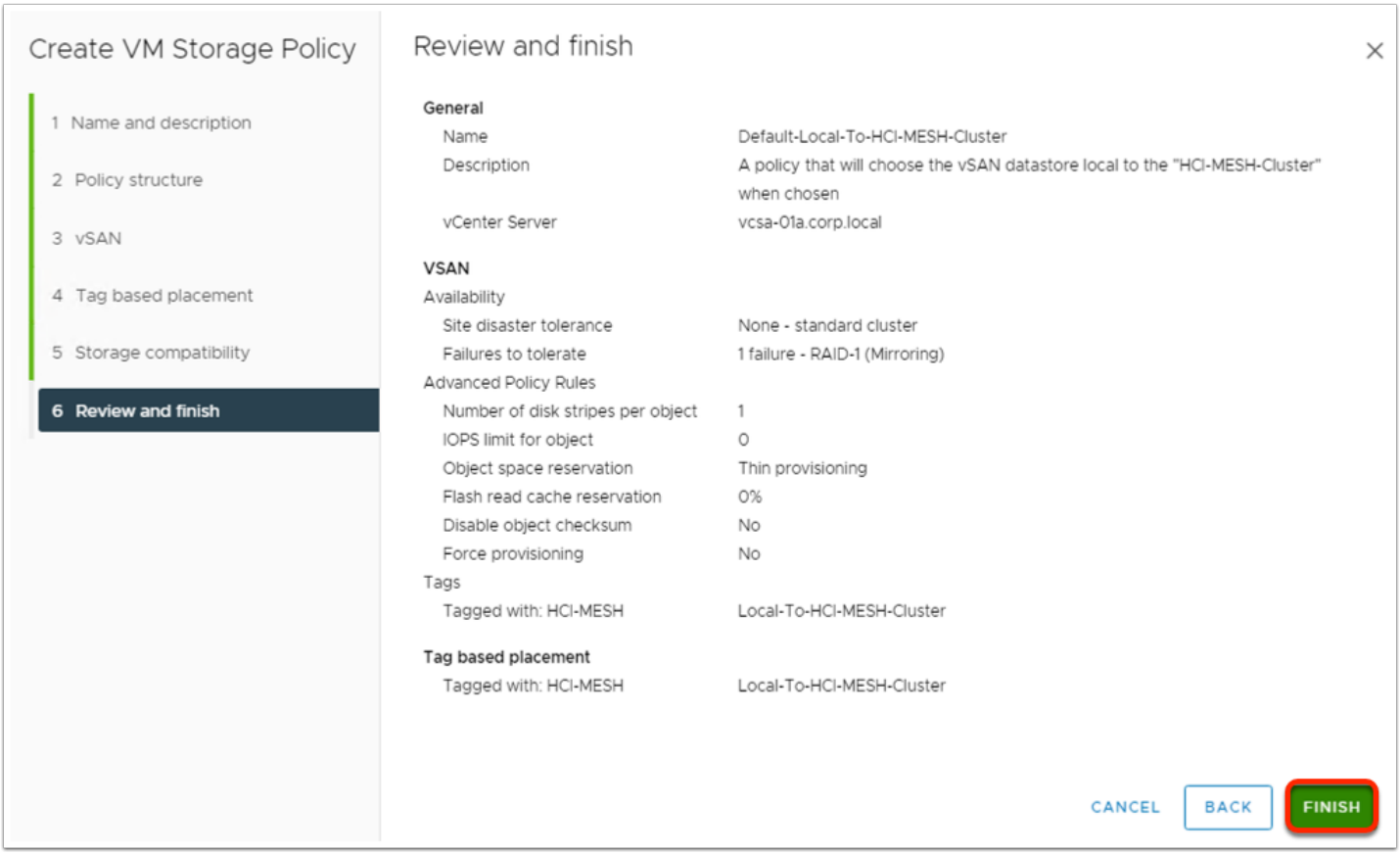
Create VM Storage Policy - storage compatibility



Whichever way that you decide to do it, it really doesn’t matter as long as the vSAN datastore that you want to be selected with this rule shows up as the only compatible one in the Storage compatibility view.

Click **NEXT**

Create VM Storage Policy - Review and Finish

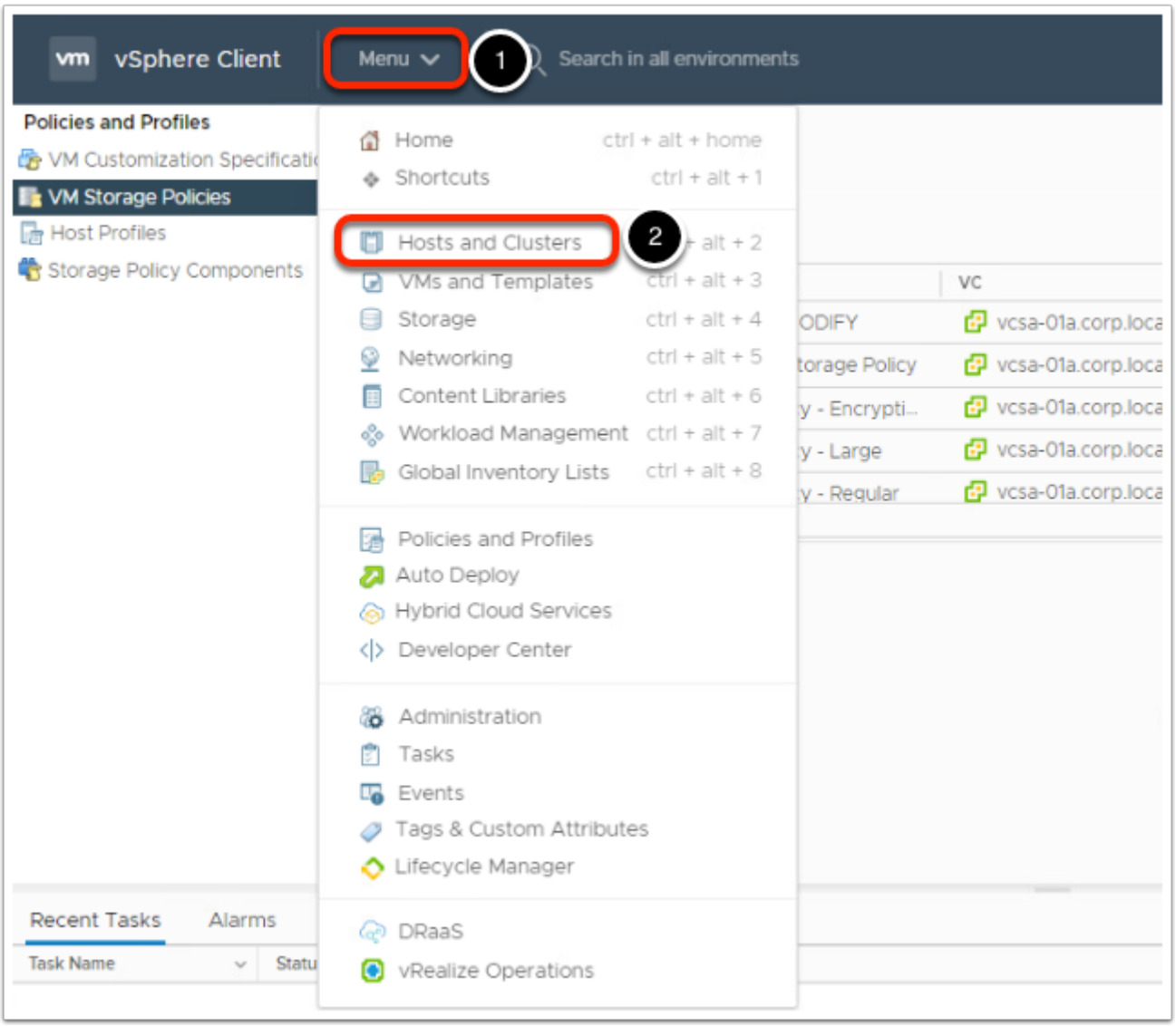


Review what you have enter. You notice since we did the tag placement in both places, it is outline twice.

Click **FINISH.**

Apply VM for HCI Mesh

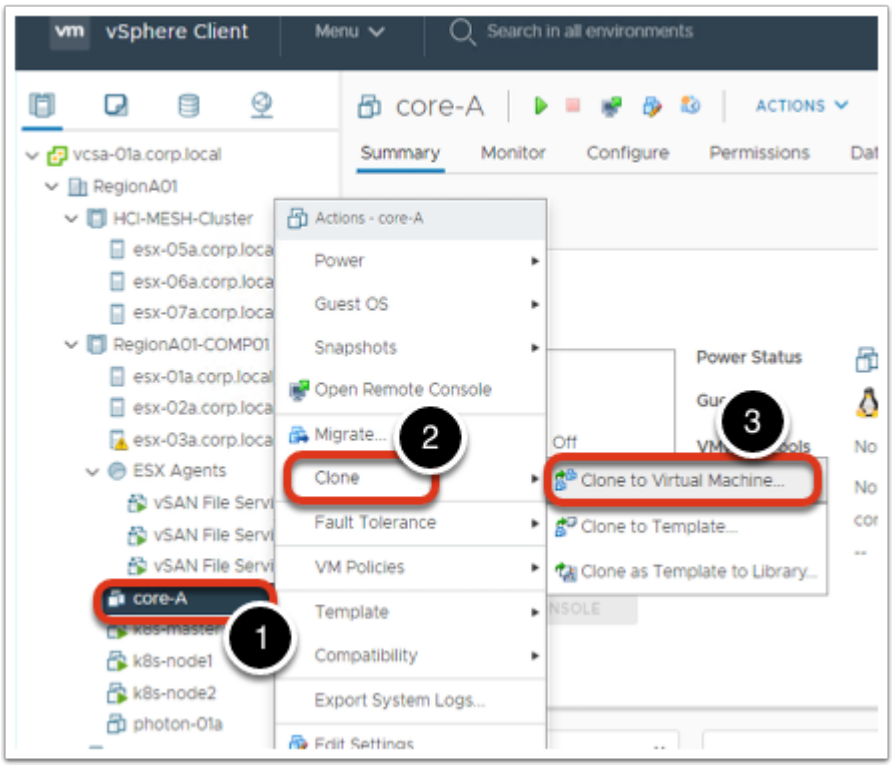
We will now deploy a VM and apply the newly created Storage Policy for the vSAN Mesh Datastore



1. Click on **Menu**
2. Select **Hosts and Clusters**

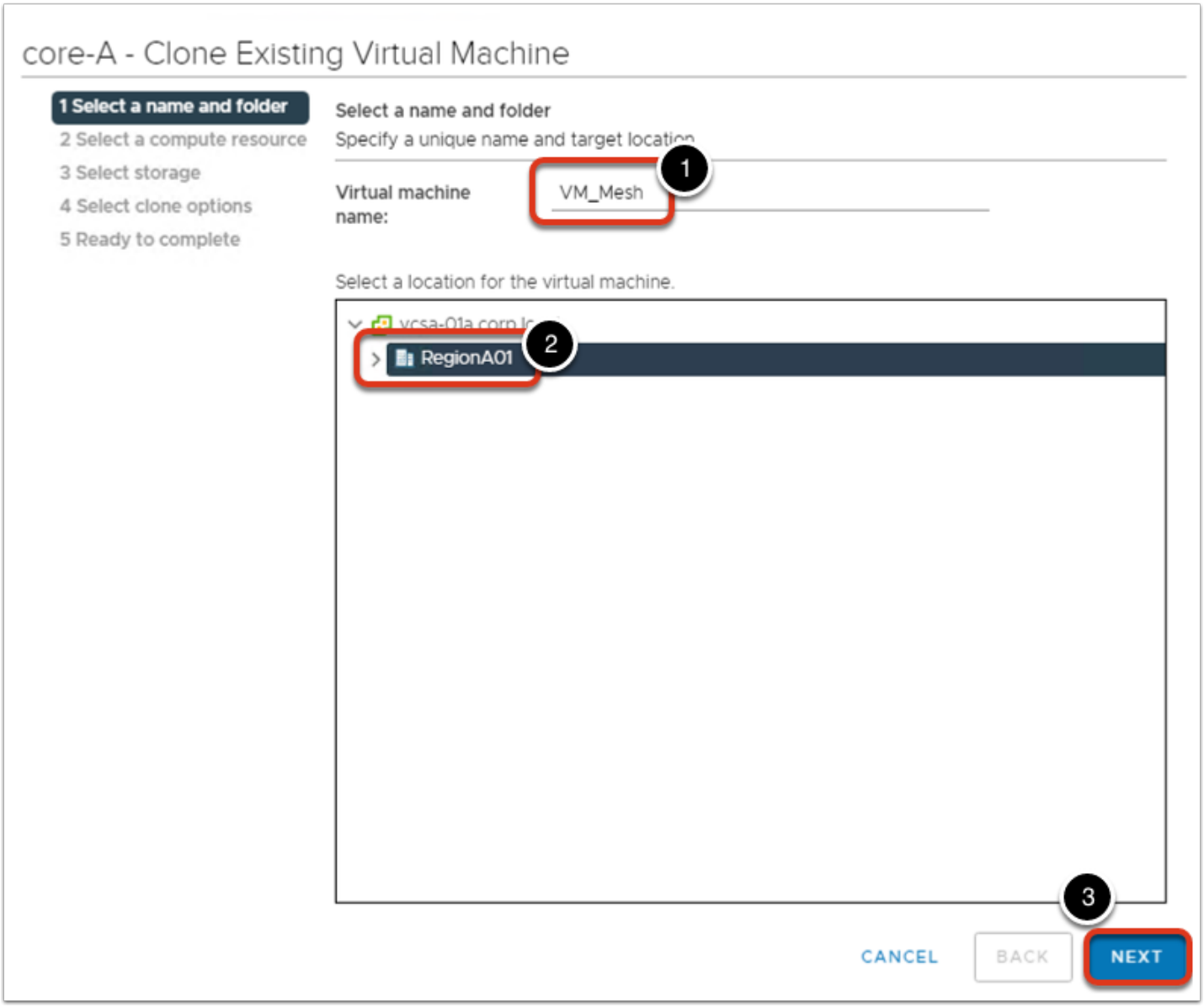
Deploy VM with HCI Storage Policy

We will clone the VM called **core-A** (which currently resides on an iSCSI VMFS datastore) to the vSAN Mesh Datastore and apply the new storage policy.



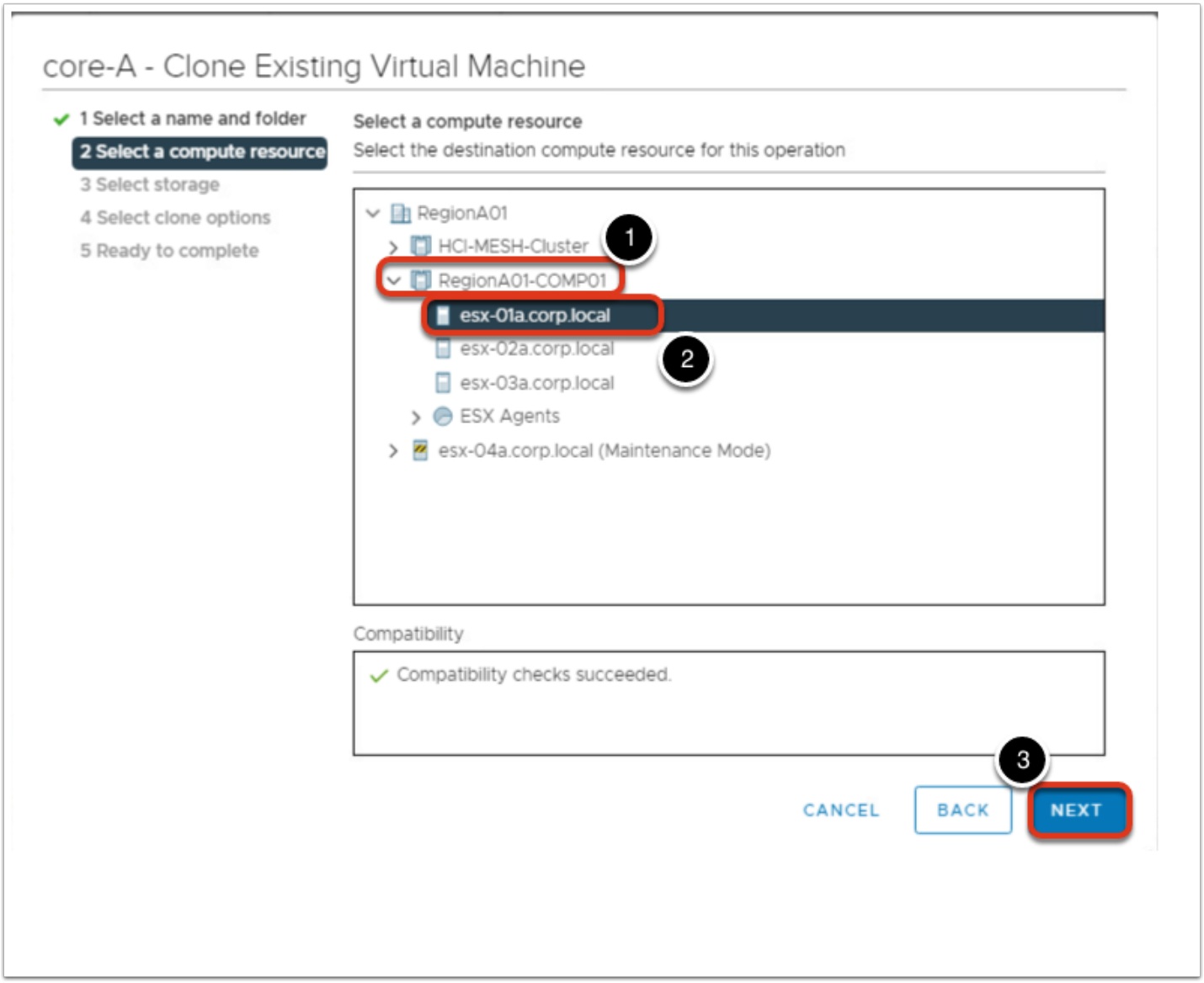
1. Expand the ESXi Cluster called RegionA01-COMP01 and right click the VM called **core-A**
2. Select **Clone**
3. Select **Clone to Virtual Machine**

Deploy VM to Storage Policy



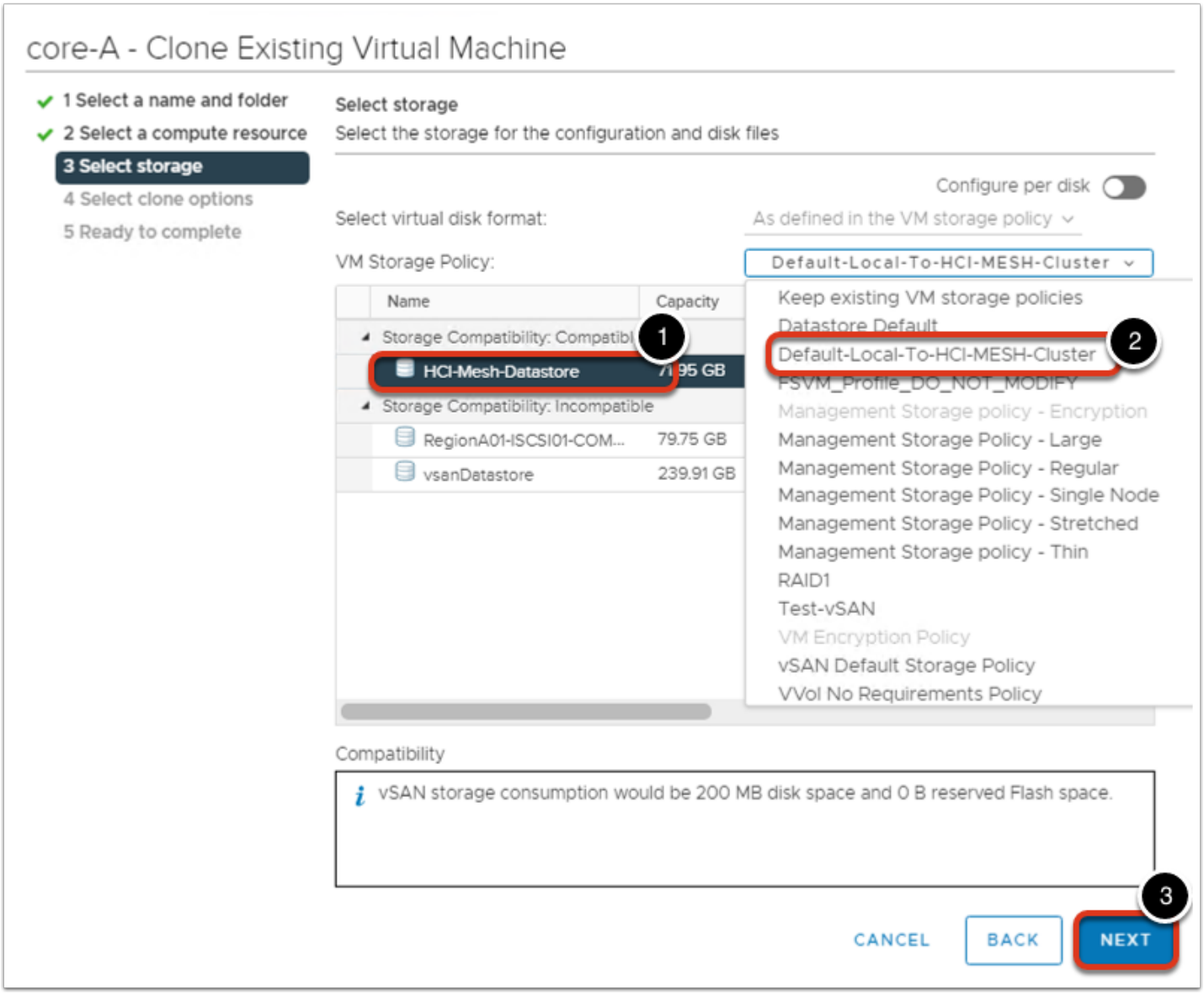
1. Give the Virtual Machine a name: VM_Mesh
2. Select **vcsa-01a.corp.local** > **RegionA01**
3. Click **NEXT**

Deploy VM to Storage Policy



1. Expand the compute resource called **RegionA01-COMP01**
2. Select the ESXi host called **esx-01a.corp.local**
3. Click **NEXT**

Deploy VM to Storage Policy



1. Click on **HCI-Mesh-Datastore**
2. For the VM Storage Policy dropdown, select **Default-Local-To-HCI-MESH-Cluster**

The resulting list of compatible datastores will be presented, in our case the HCI-Mesh-Datastore. In the lower section of the screen we can see the vSAN storage consumption would be 200.00 MB disk space and 0.00 B reserved Flash space.

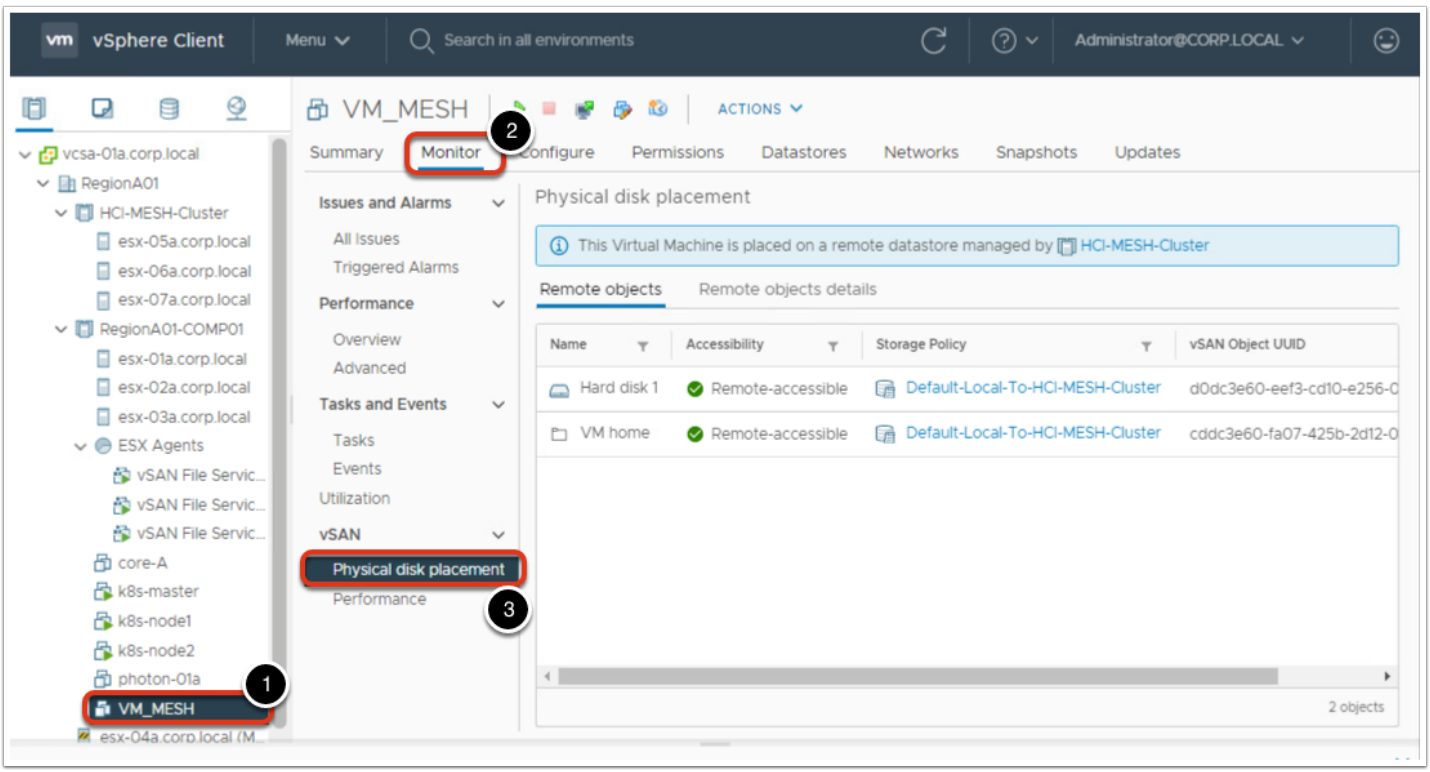
Since we have a VM with 100 MB disk and the Default Storage Policy, the VSAN disk consumption will be 200.00 MB disk.

3. Click **NEXT** then, Click **NEXT** on the **Select clone options** (not picture), then Click **FINISH** (not picture)

Monitor HCI Mesh

Next, we will examine the disk and components placement details in regards to vSAN HCI-Mesh datastores

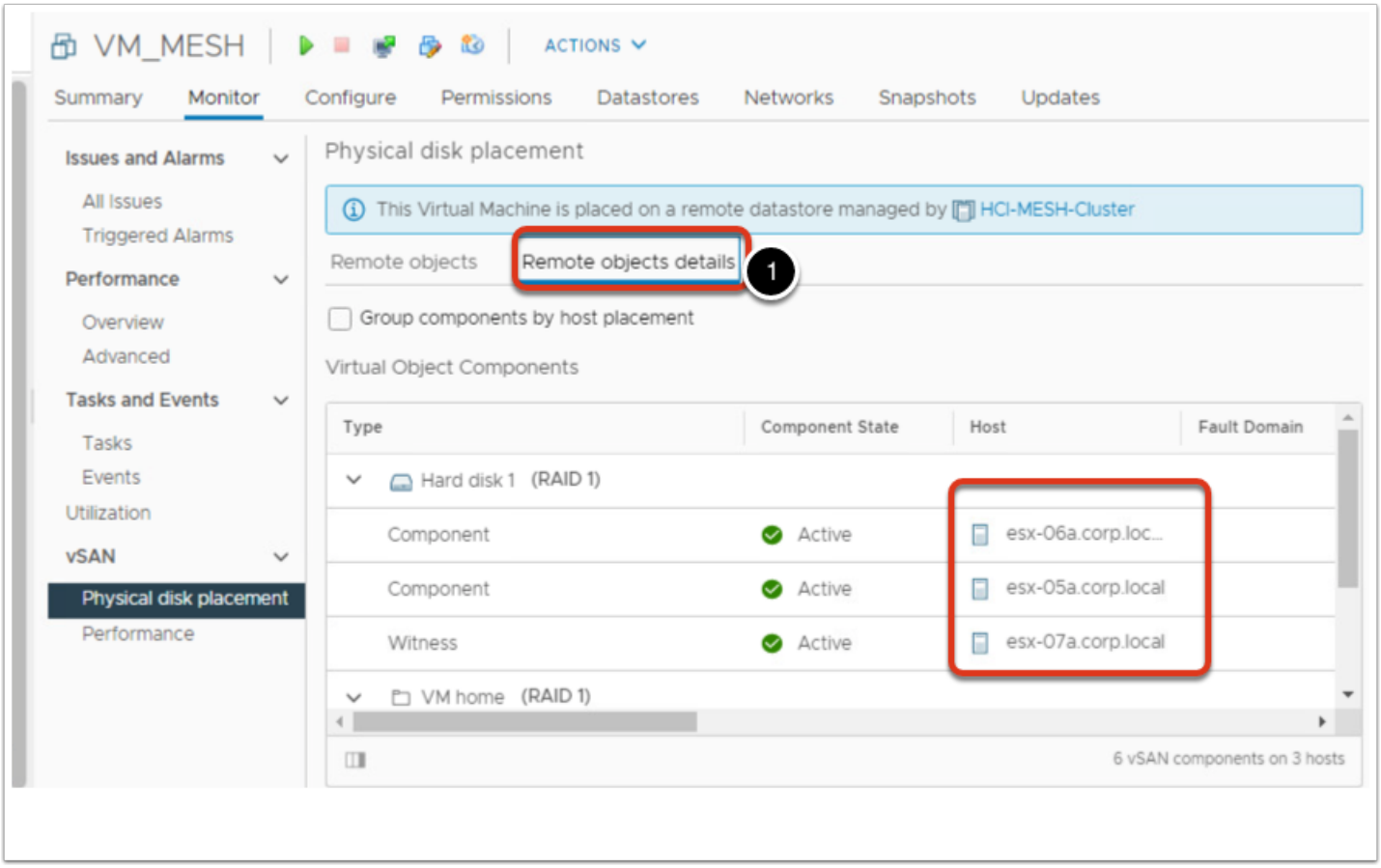
Monitor HCI Mesh



1. Select VM, **VM_MESH**
2. Click **Monitor**
3. Select **vSAN > Physical disk placement**

You can see that the VM you created is accessible to the remote datastore.

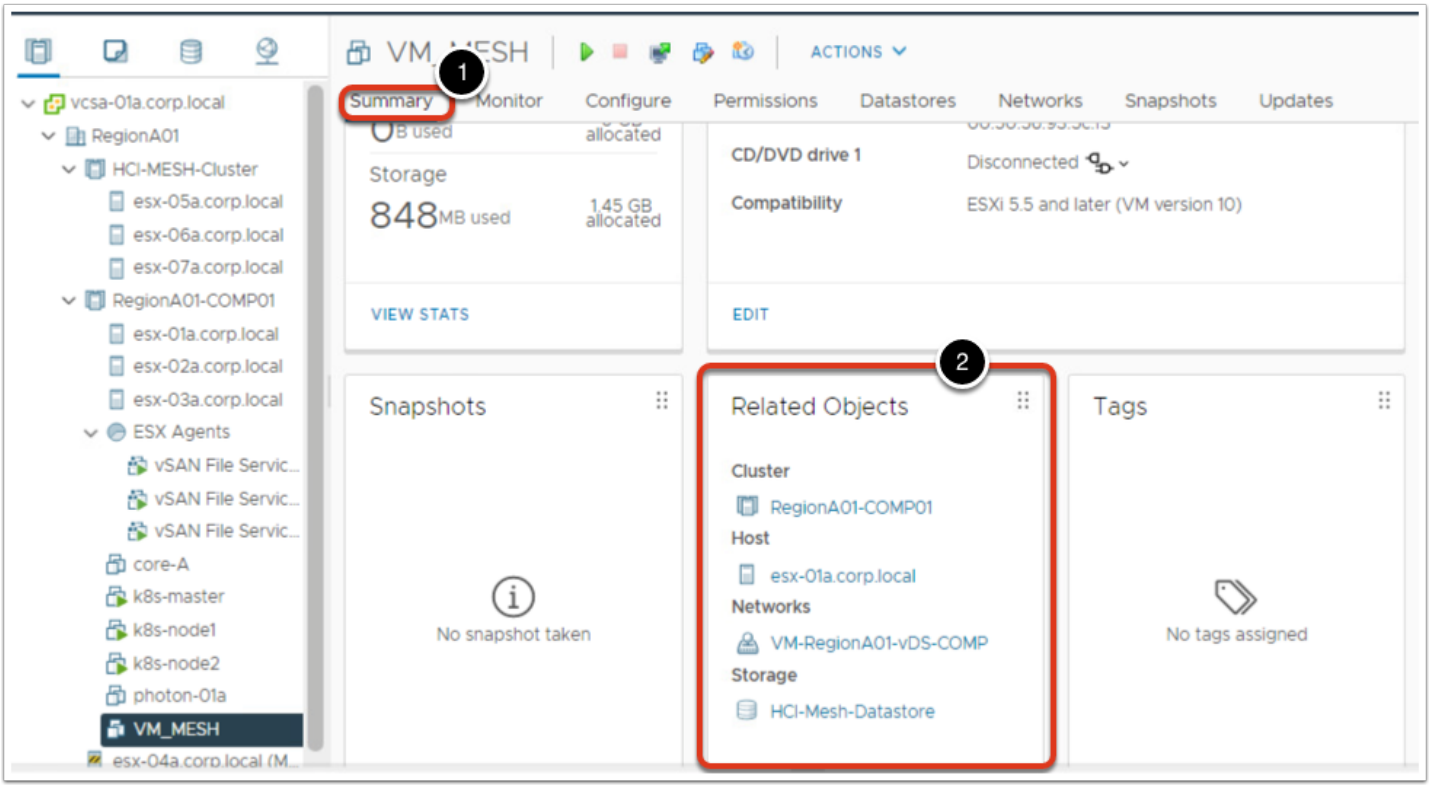
Monitor HCI MESH



1. Select **Remote objects details**

You can see the components and witness is in vSAN Cluster, HCI-MESH-Cluster

Monitor HCI MESH Summary



1. Select Summary
2. Look for **Related Objects**

You can see the vm, VM_Mesh's CPU and Memory is located on host, esx-01a.corp.local but the storage is on the HCI-Mesh-Datastore which is located in the vSAN Cluster, HCI-MESH-Cluster.

This concludes this module.

Module 8 - vSAN Encryption and Security

Module 8 - vSAN Encryption

vSAN Encryption Overview

You can encrypt data-in transit in your vSAN Cluster and encrypt data-at-rest in your vSAN datastore.

vSAN can encrypt data in transit across hosts in the vSAN cluster. Data-in-transit encryption protects data as it moves around the vSAN cluster.

vSAN can encrypt data at rest in the vSAN datastore. Data-at-rest encryption protects data on storage devices, in case a device is removed from the cluster.

vSAN Data-In Transit Encryption

- vSAN can encrypt data in transit, as it moves across hosts in your vSAN cluster.
- vSAN uses AES-256 bit encryption on data in transit.
- vSAN data-in-transit encryption is not related to data-at-rest-encrytion. 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 client servers is encrypted.
- vSAN uses symmetric keys that are generated dynamically and shared between hosts. Hosts dynamically generate an encryption key when they establish a connection, and they use the key to encrypt all traffic between the hosts. You do not need a key management server to perform data-in-transit encryption.
- Each host is authenticated when it joins the cluster, ensuring connections only to trusted hosts are allowed. When a host is removed from the cluster, it is authentication certificate is removed.

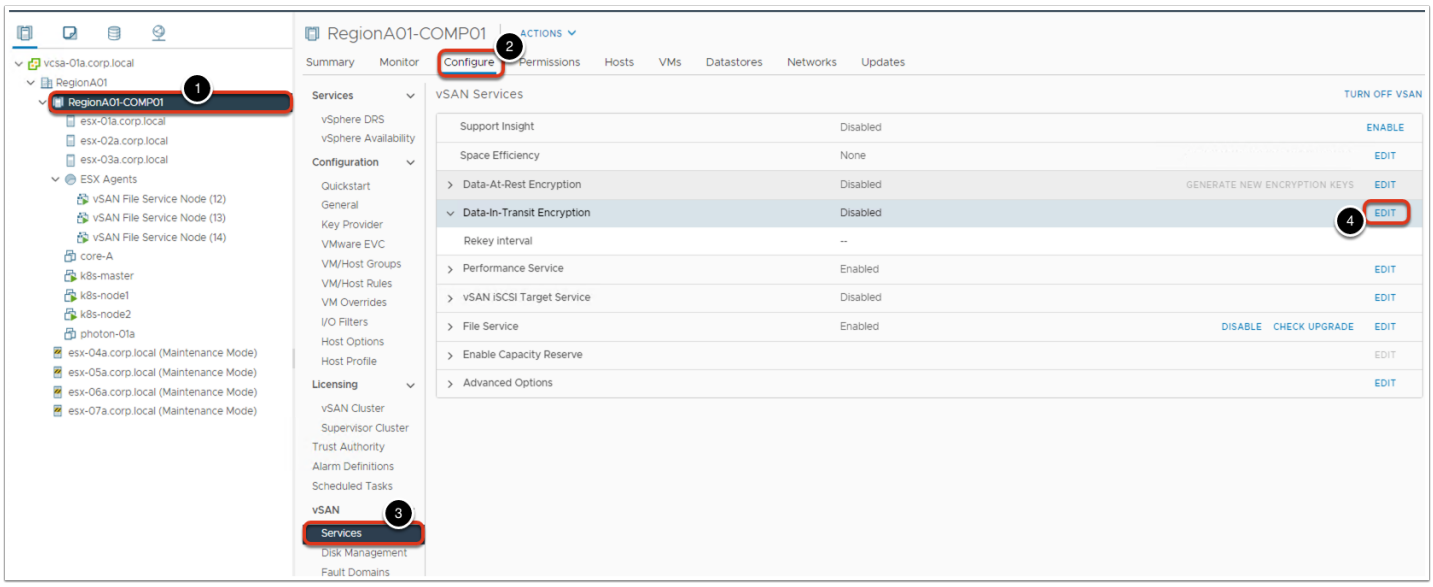
vSAN Data-At-Rest Encryption

- vSAN can encrypt data at rest in your vSAN datastore.
- When you enable encryption, vSAN encrypts everything in the vSAN datastore. All files are encrypted, so all virtual machines and their corresponding data are protected. Only administrators with encryption privileges can perform encryption and decryption tasks.
- vSAN uses encryption keys as follows:
    - vCenter Server requests an AES-256 Key Encryption Key (KEK) from the KMS.vCenter Serverstores only the ID of the KEK, but not the key itself
    - The ESXi host encrypts disk data using the industry standard AES-256 XTS mode. Each disk has a different randomly generated Data Encryption Key (DEK)
    - Each ESXi host uses the KEK to encrypt its DEKs, and stores the encrypted DEKs on disk. The host does not store the KEK on disk. If a host reboots, it

requests the KEK with the corresponding ID from the KMS. The host can then decrypt its DEKs as needed
- A host key is used to encrypt core dumps, not data. All hosts in the same cluster use the same host key. When collecting support bundles, a random key is generated to re-encrypt the core dumps. You can specify a password to encrypt the random key

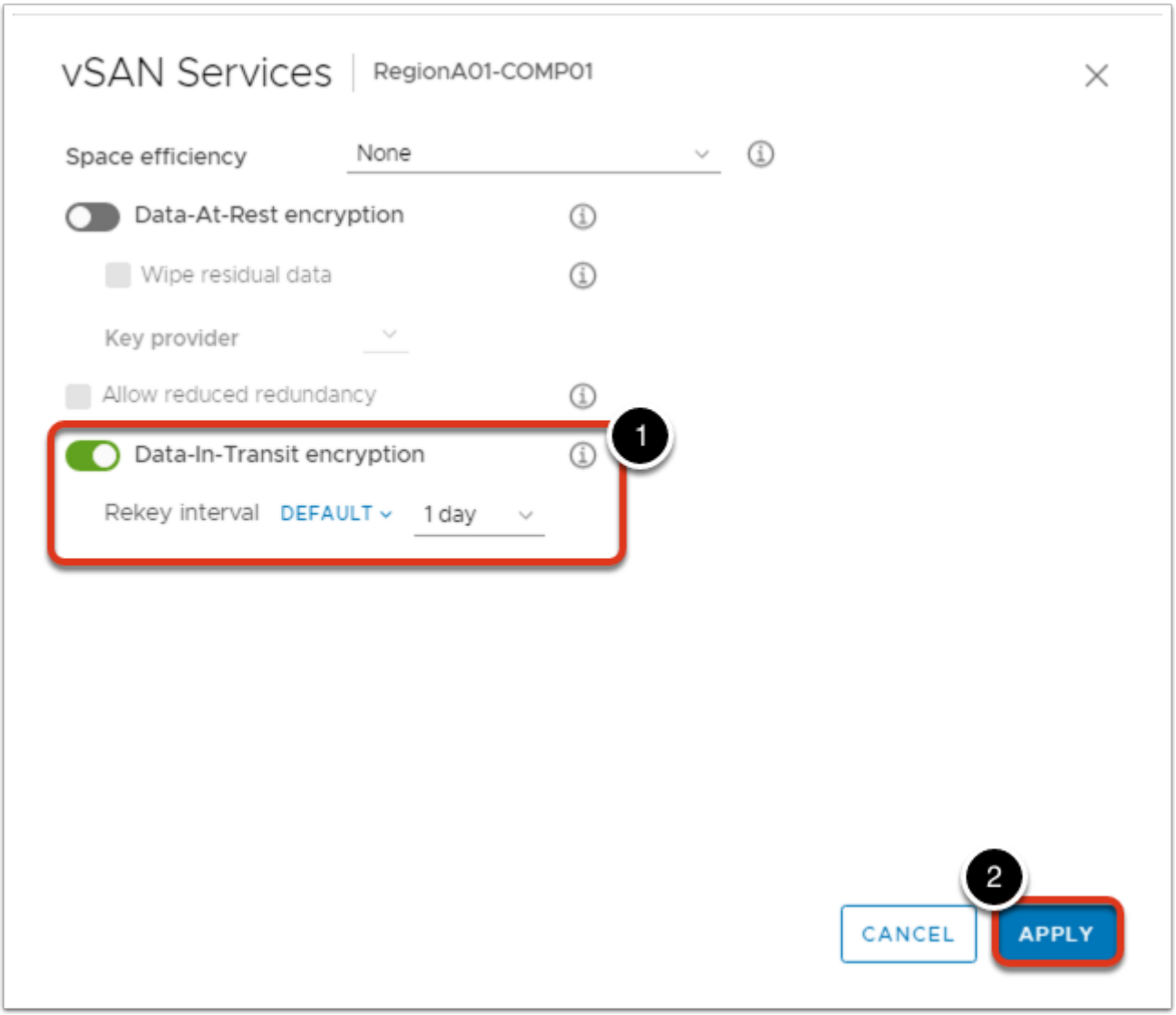
Data-In-Transit Encryption

Enable Data-In-Transit Encryption



1. Select vSAN Cluster, **RegionA01-COMP01**
2. Click **Configure**
3. Select **vSAN > Services**
4. Click **ENTER** in the **Data-At-Transit Encryption**

vSAN Services Encryption



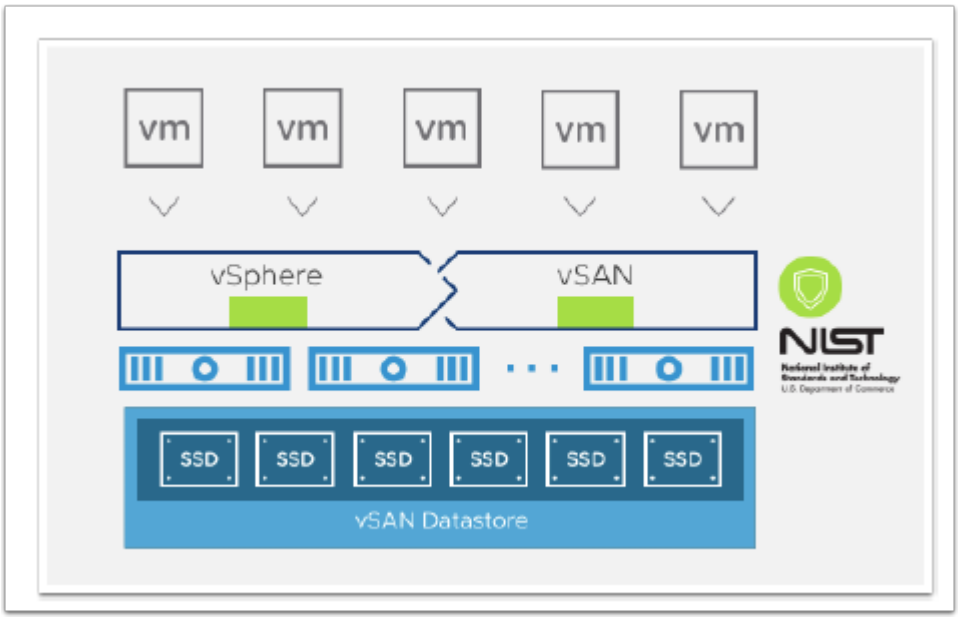
When you enable Data-In-Transit Encryption, rekey interval can be set. By default, it is set for 1 day but you can customize it as needed.

1. **Toggle** on Data-In-Transit encryption to be enable
2. Click on **Apply**

DISA STIG (FIPS 140-2) Validated

vSAN offered the first native HCI encryption solution for data-at-rest since vSAN 6.7. vSAN Encryption is the first FIPS 140-2 validated software solution, meeting stringent US Federal Government requirements. vSAN Encryption delivers lower data protection costs and greater flexibility by being hardware agnostic and by offering simplified key management. This is also the first HCI solution with a DISA-approved STIG.

FIPS 140-2 Validation



vSAN takes an important step forward with improved security since vSphere 6.7, with FIPS 140-2 validation. Since vSAN is integrated into the hypervisor, it uses the kernel module used in vSphere, and as of vSphere 6.7, has achieved FIPS 140-2 validation.

Organizations that require this level of validation can be confident that VMware vSphere, paired with VMware vSAN, will allow them to meet their security requirements.

vSAN Encryption

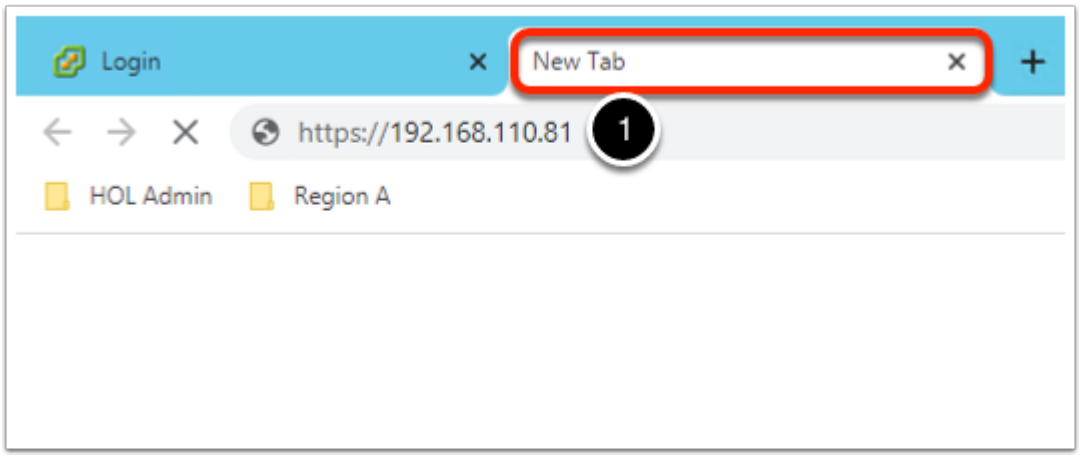
vSAN can perform data at rest encryption. Data is encrypted after all other processing, such as deduplication, is performed. Data at rest encryption protects data on storage devices, in case a device removed from the cluster.

Using encryption on your vSAN cluster requires some preparation. After your environment is set up, you can enable encryption on your vSAN cluster.

vSAN encryption requires an external Key Management Server (KMS), the vCenter Server system, and your ESXi hosts. vCenter Server requests encryption keys from an external KMS. The KMS generates and stores the keys, and vCenter Server obtains the key IDs from the KMS and distributes them to the ESXi hosts.

vCenter Server does not store the KMS keys, but keeps a list of key IDs.

Validate HyTrust KeyControl

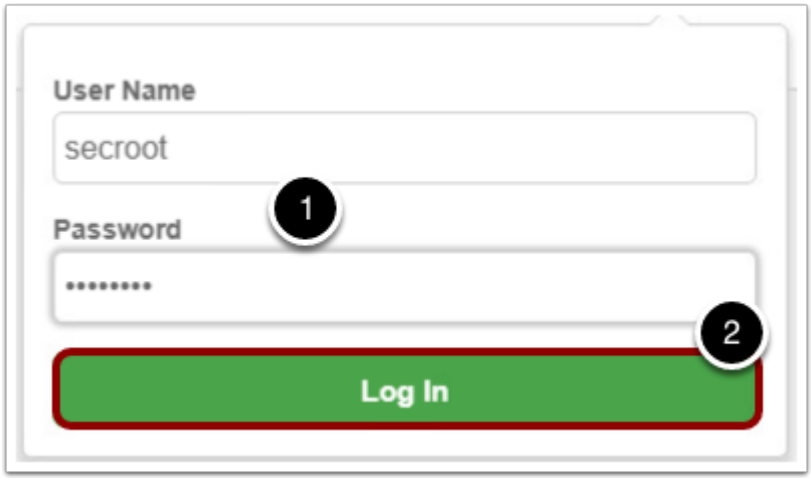


1. Open a New Chrome Browser Window or Tab and enter the following URL to connect to the HyTrust KeyControl interface:

```
https://192.168.110.81
```

2. Select **Advanced** (Not Shown)
3. Click **Proceed to 192.168.110.81 (unsafe)**

Validate HyTrust KeyControl, cont.

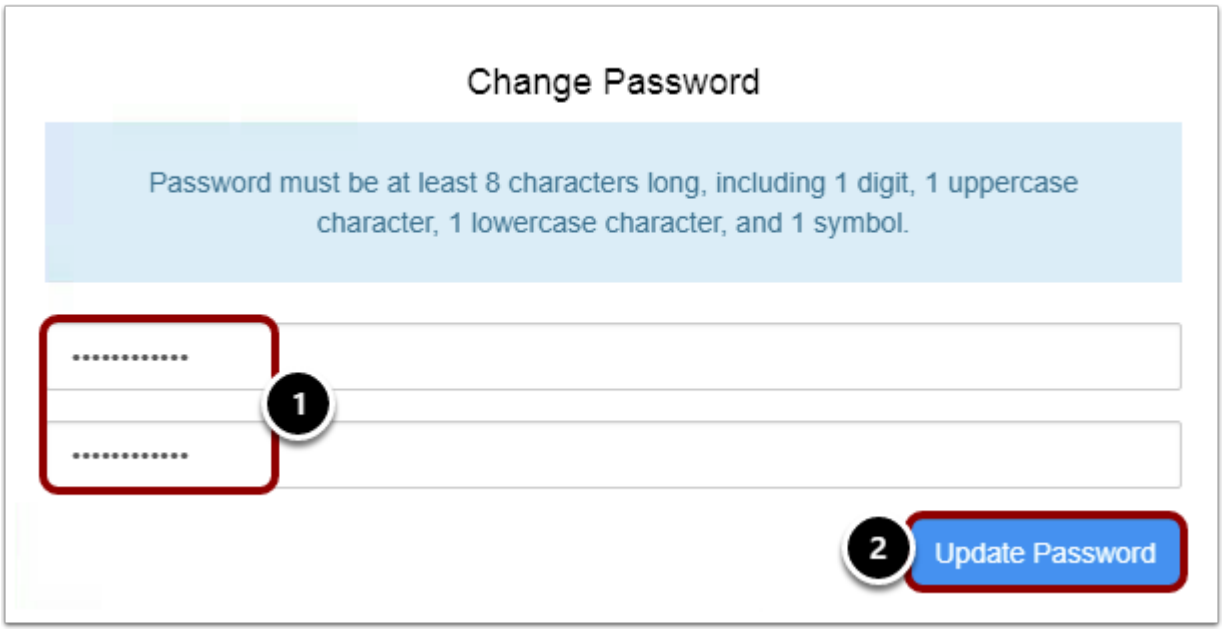


1. Use the following credentials to authenticate

```
User Name: secroot
Password: VMware1!
```

2. Click **Log In**

Change Password



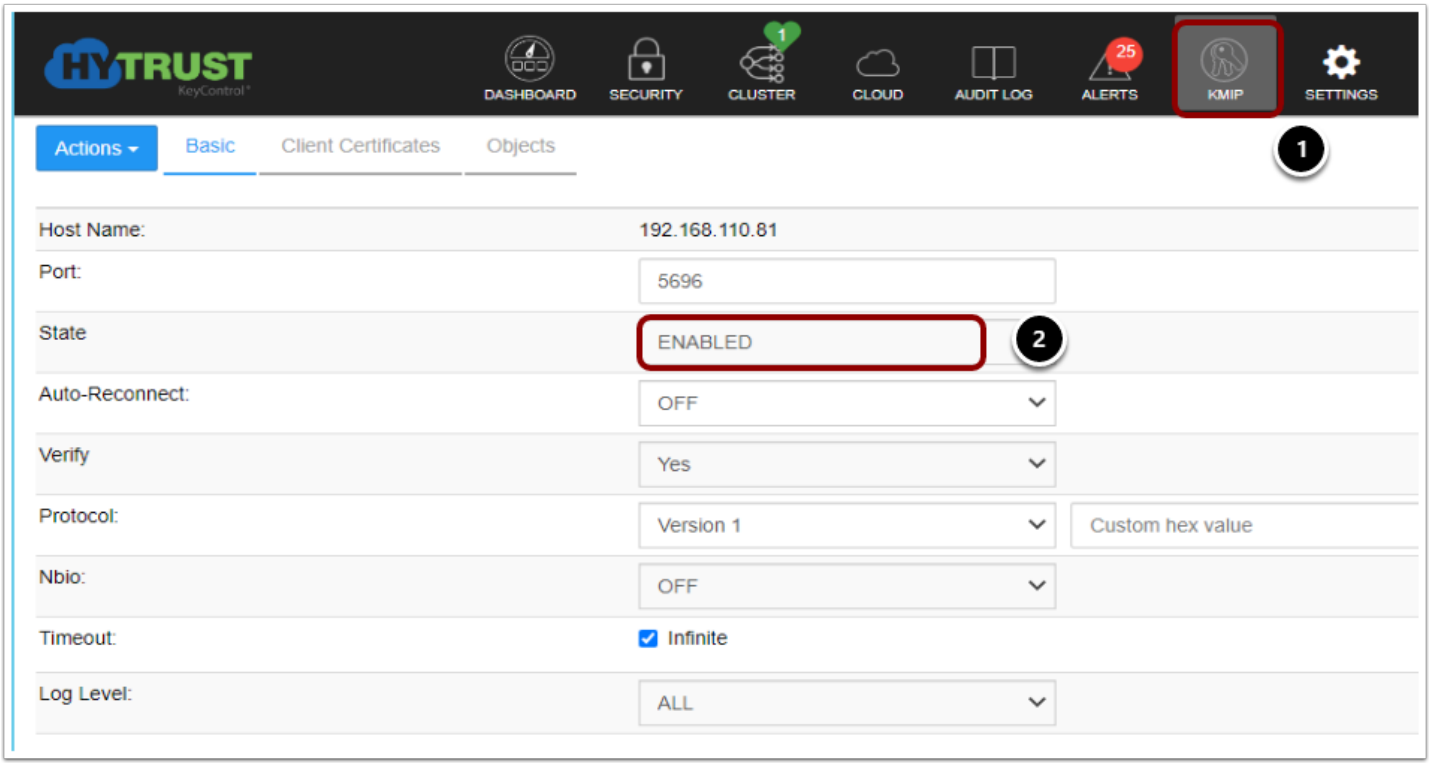
Note: If you receive a System Recovery needed warning, please click here to resolve, otherwise:

1. Enter the following new password

```
Password: !Password123
```

2. Click **Update Password**

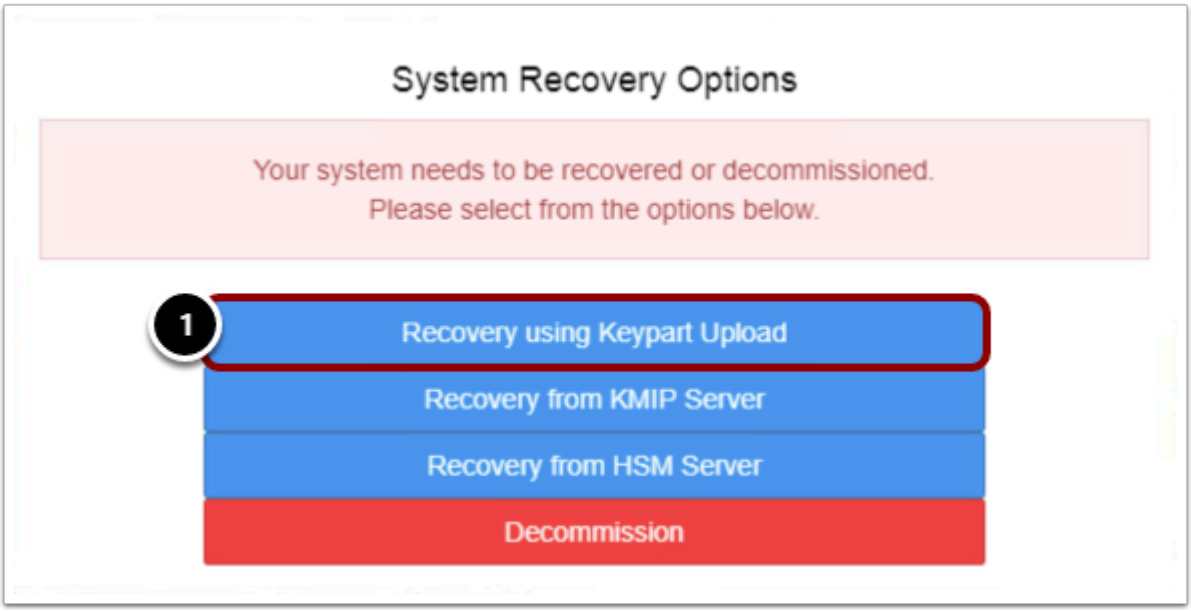
KMIP



1. Select **KMIP**
2. Note that the State of the KMS is **Enabled**

We have confirmed the functional state of the HyTrust KeyControl KMS instance. Click here to begin enabling vSAN Encryption.

System Recovery Options



1. Open a New Chrome Tab and use the following URL to connect to the HyTrust KeyControl interface:

```
https://192.168.110.81
```

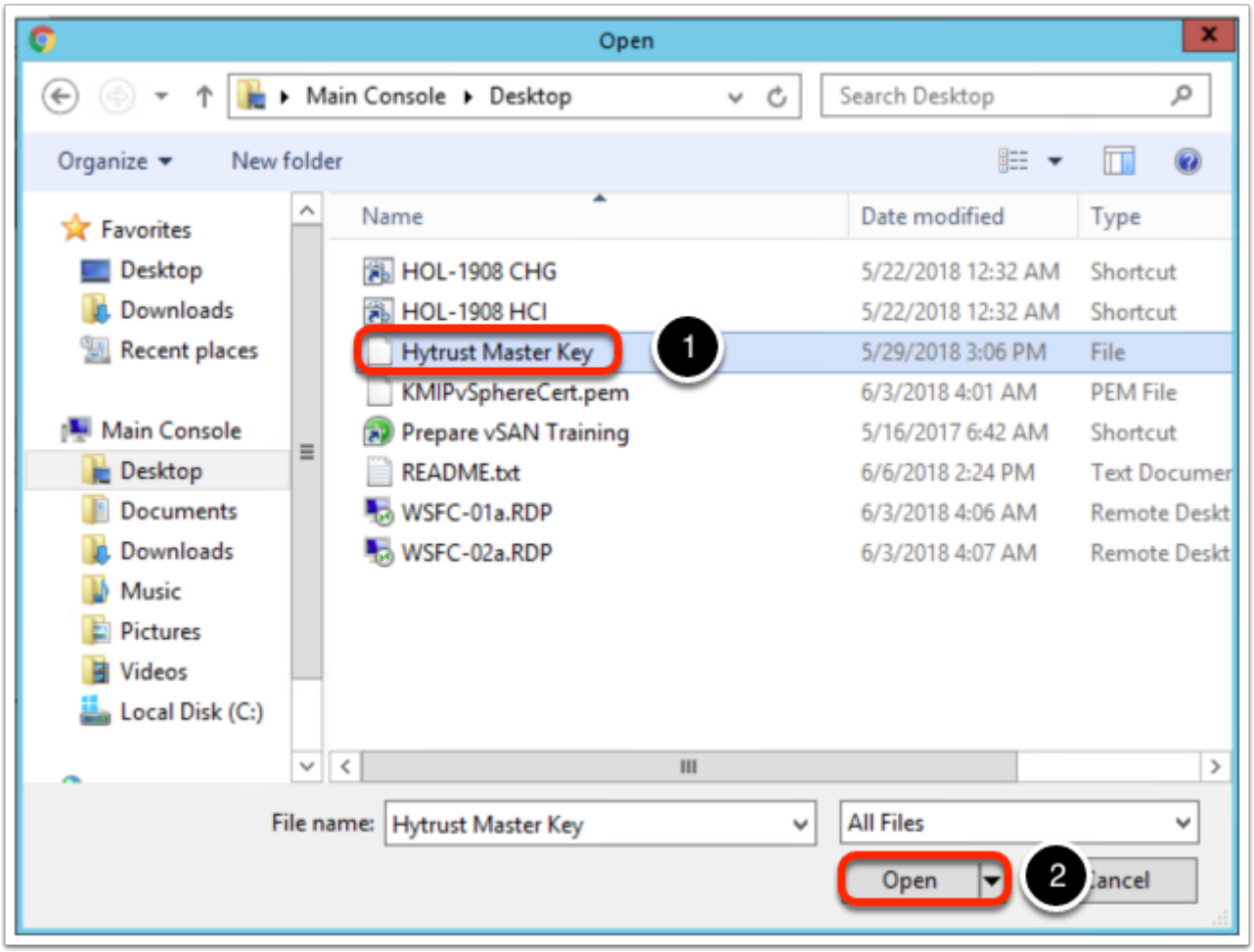
2. Use the following credentials to authenticate and click **Log In**

```
User Name:  secroot
Password: VMware1!
```

Recover Admin Key

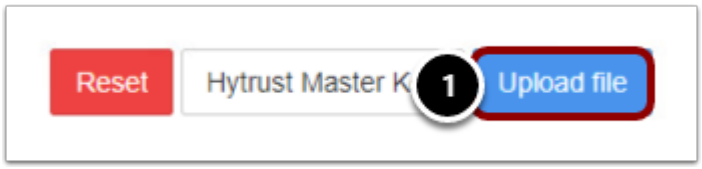
1. Click **Browse**

Open Dialog



1. Select **Hytrust Master Key**
2. Click **Open**

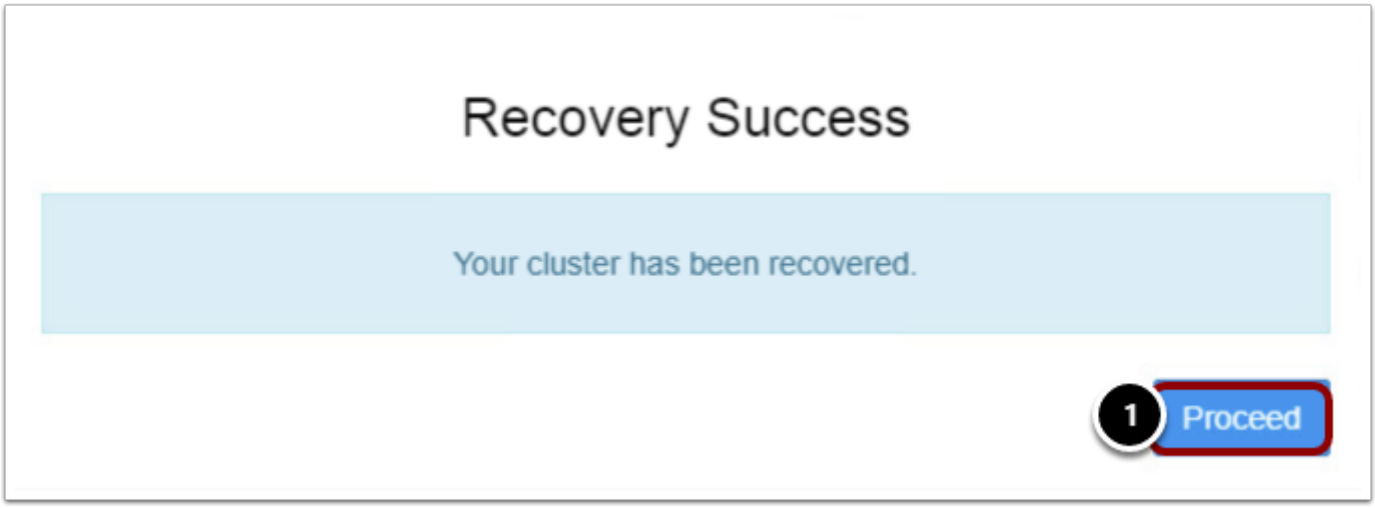
Upload File



1. Click **Upload File**

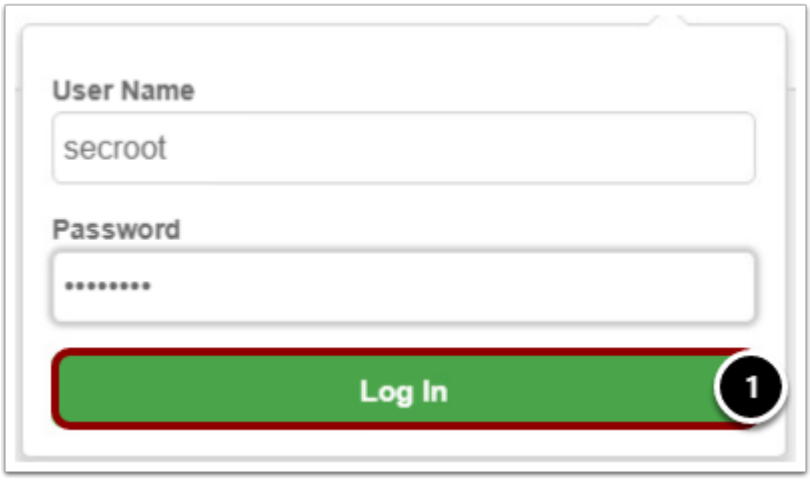
Allow the process to complete (note that this may take a few minutes, thank you for your patience)!

Recovery Success



1. Click **Proceed**

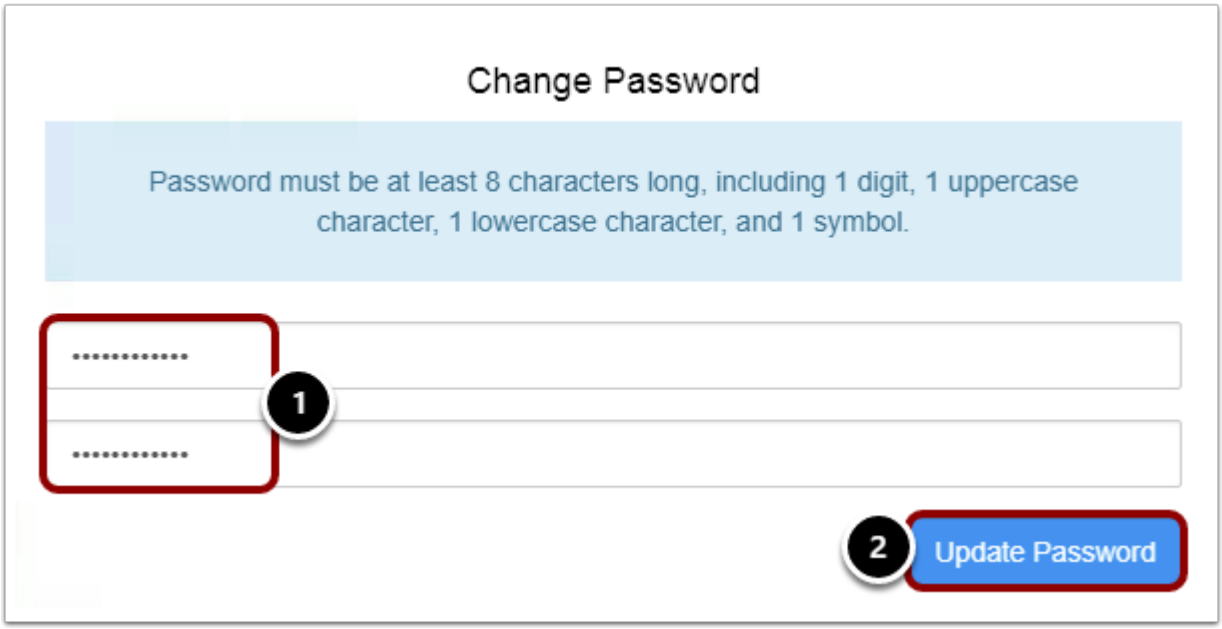
HyTrust Login



1. Use the following credentials to authenticate and click **Log In**

```
User Name:  secroot
Password: VMware1!
```

Change Password



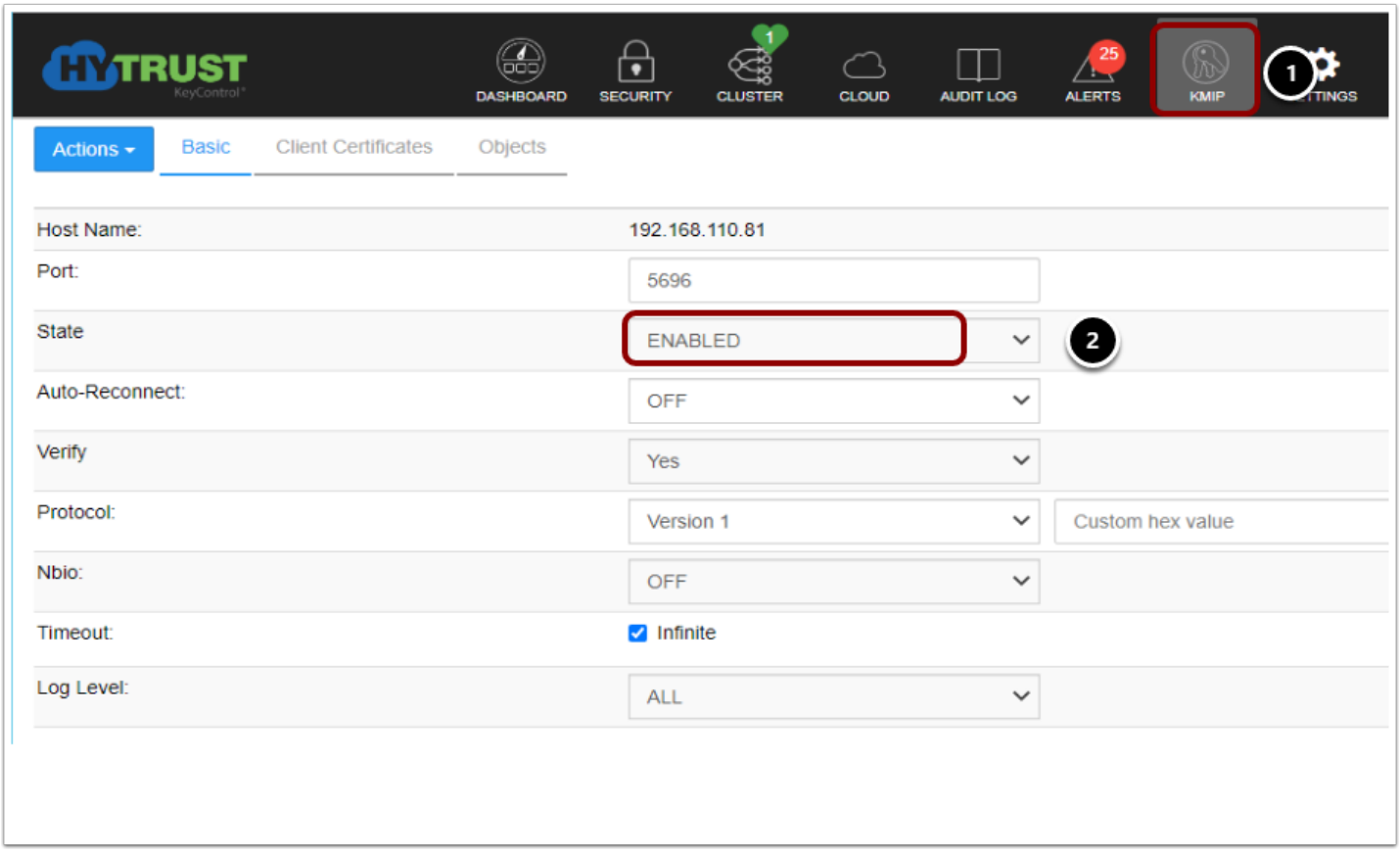
Note: If you receive a System Recovery needed warning, please click here to resolve, otherwise:

1. Enter the following new password

```
Password: !Password123
```

2. Click **Update Password**

KMIP



1. Select **KMIP**
2. Note that the State of the KMS is **Enabled**

We have confirmed the functional state of the HyTrust KeyControl KMS instance and are now ready to configure vSAN Encryption.

Configuring the Key Management Server

A Key Management Server (KMS) cluster provides the keys that you can use to encrypt the vSAN datastore.

Before you can encrypt the vSAN Datastore, you must set up a KMS cluster to support encryption. That task includes adding the KMS to vCenter Server and establishing trust with the KMS.

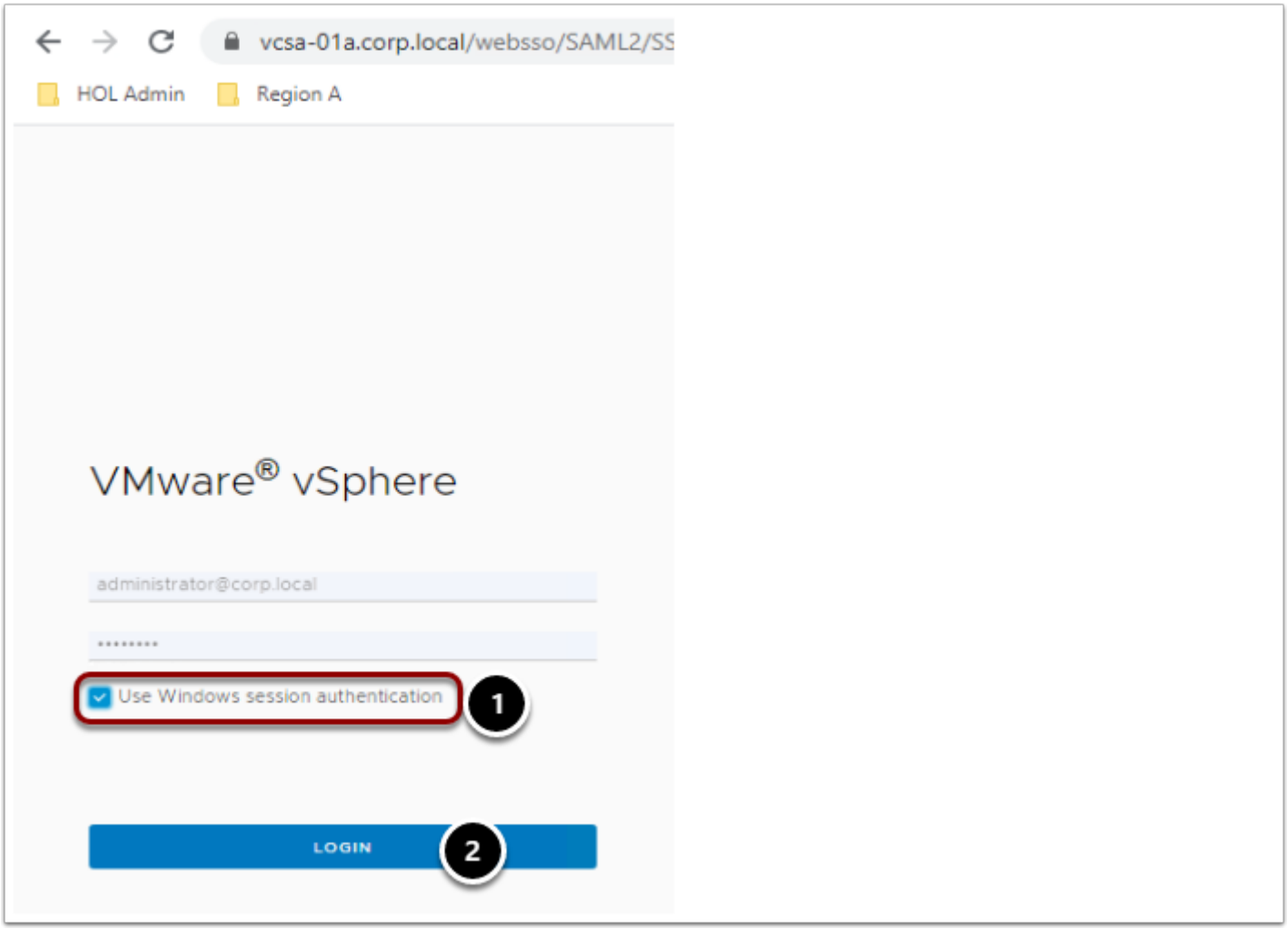
The vCenter Server provisions encryption keys from the KMS cluster.

The KMS must support the Key Management Interoperability Protocol (KMIP) 1.1 standard.

Launch vSphere Client

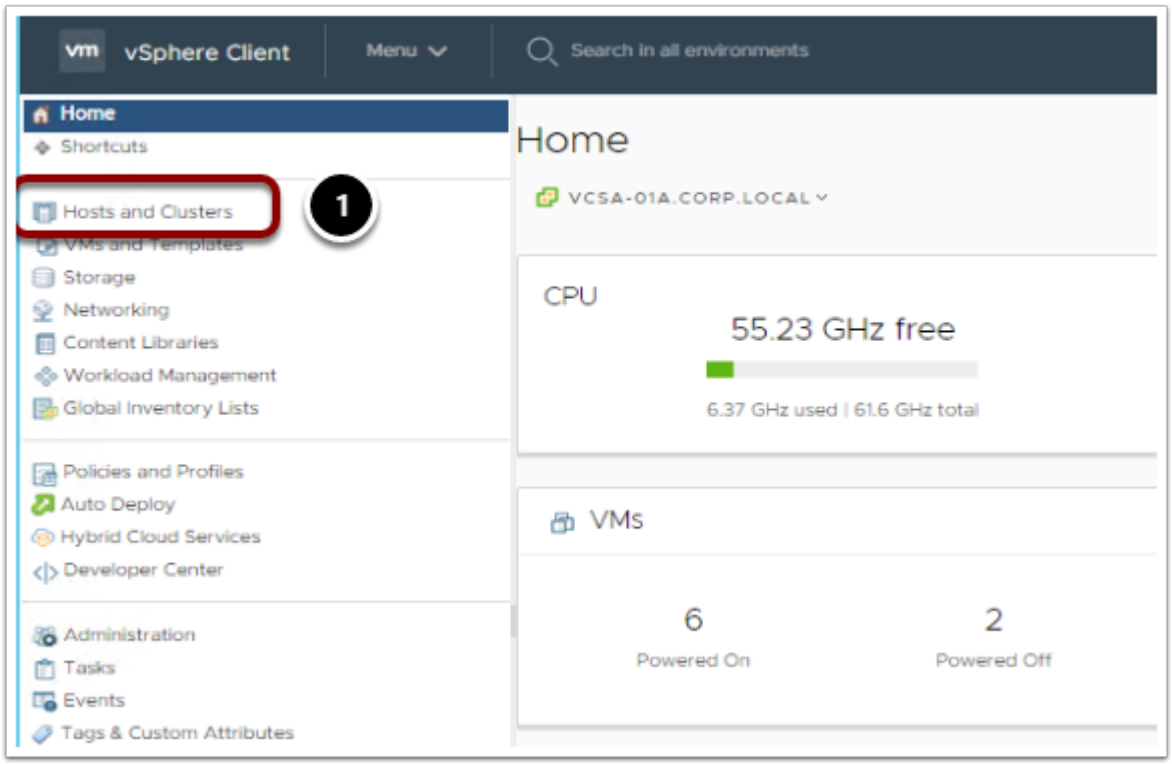
1. If Chrome is not already running, Click on the Chrome Icon on the Windows Quick Launch Task Bar.

Login to vSphere Client



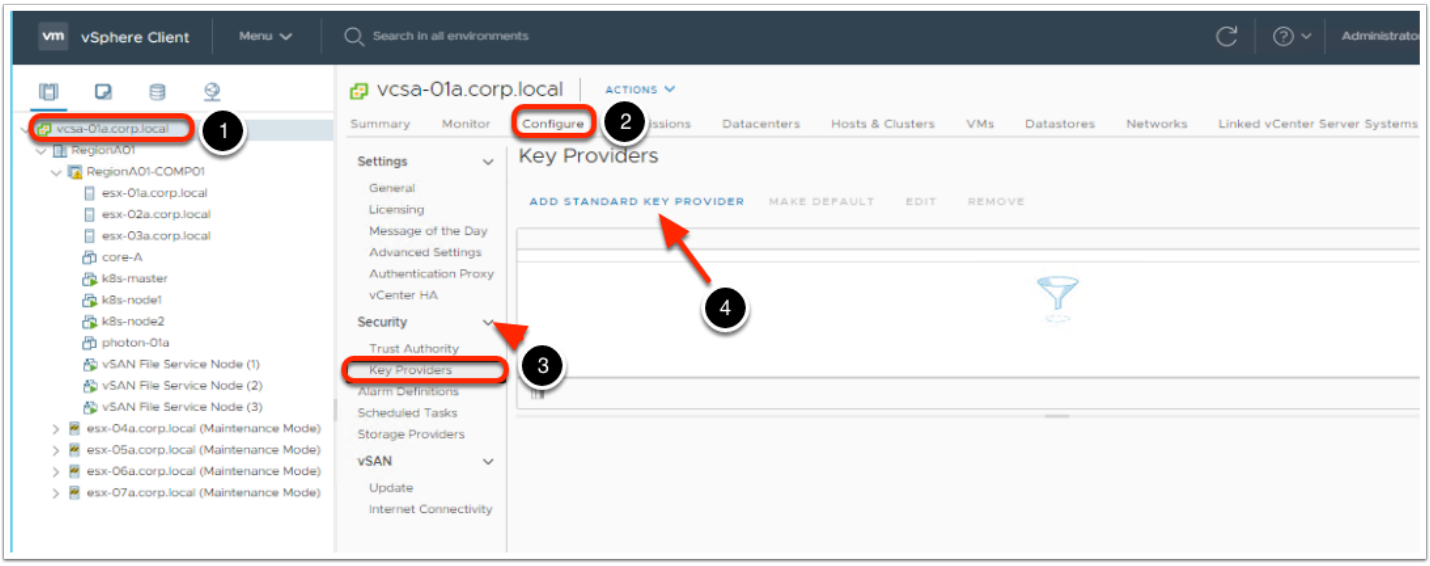
1. On the vSphere Web Client login screen, select "**Use Windows session authentication**"
2. Click **Login**

Select Hosts and Clusters



1. Select **Hosts and Clusters**

Add Key Management Server settings



A Key Management Server (KMS) cluster provides the keys that you can use to encrypt the vSAN datastore.

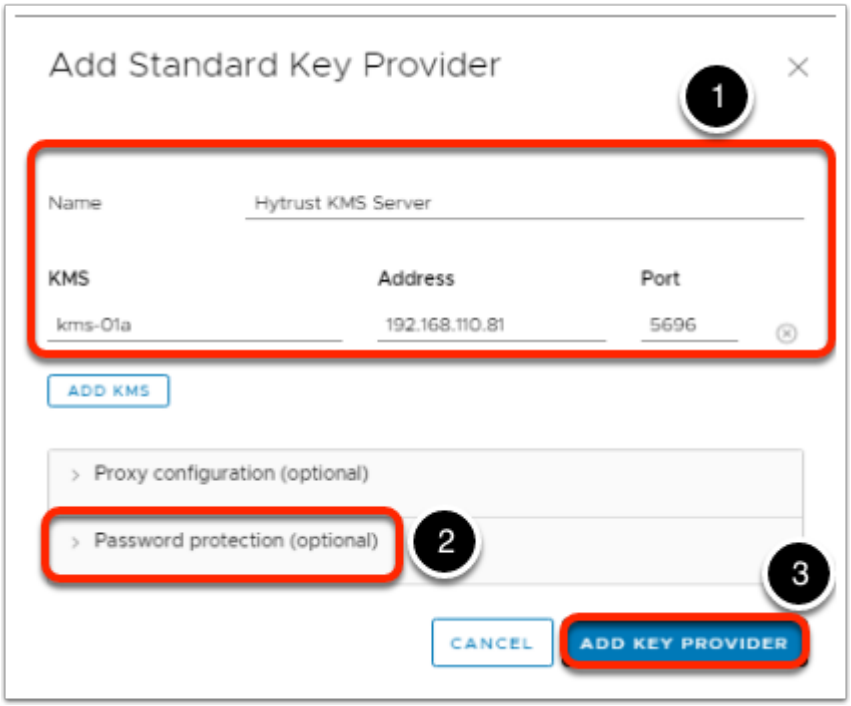
Before you can encrypt the vSAN datastore, you must set up a KMS cluster to support encryption.

That task includes adding the KMS to vCenter Server and establishing trust with the KMS.

vCenter Server provisions encryption keys from the KMS cluster.

1. Select the vCenter Server called v**csa-01a.corp.local**
2. Select **Configure**
3. Select **Security > Key Providers**
4. Click **ADD STANDARD KEY PROVIDER**

Add Key Management Server



1. Enter the following information to create the KMS Cluster:

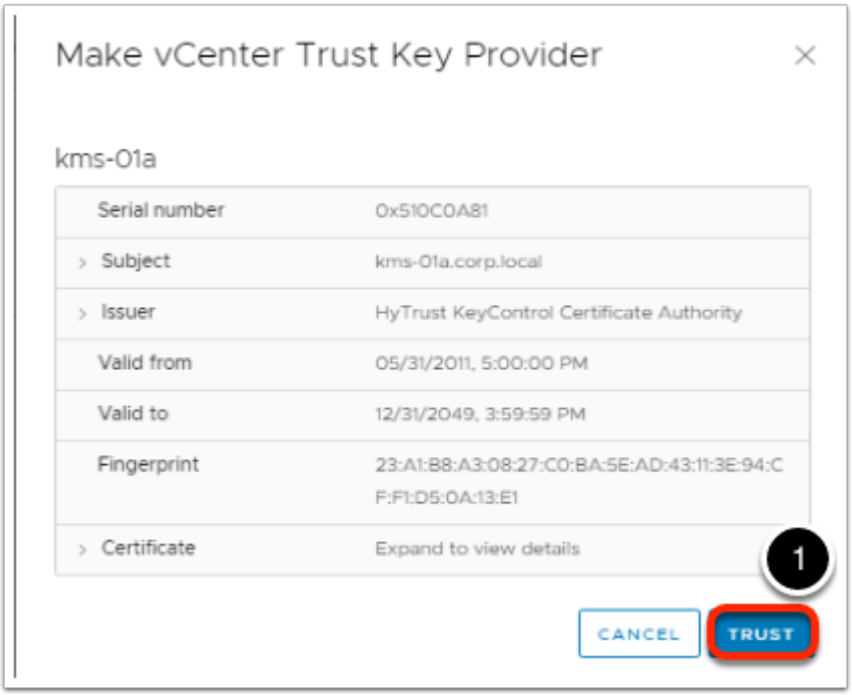
```
Name : Hytrust KMS Server
KMS : kms-01a
Address : 192.168.110.81
Port : 5696
```

2. Click **Password protection (optional)** and remove the Password (if the fields are already populated)

The remaining settings can be left at the defaults.

3. Click **ADD KEY PROVIDER**

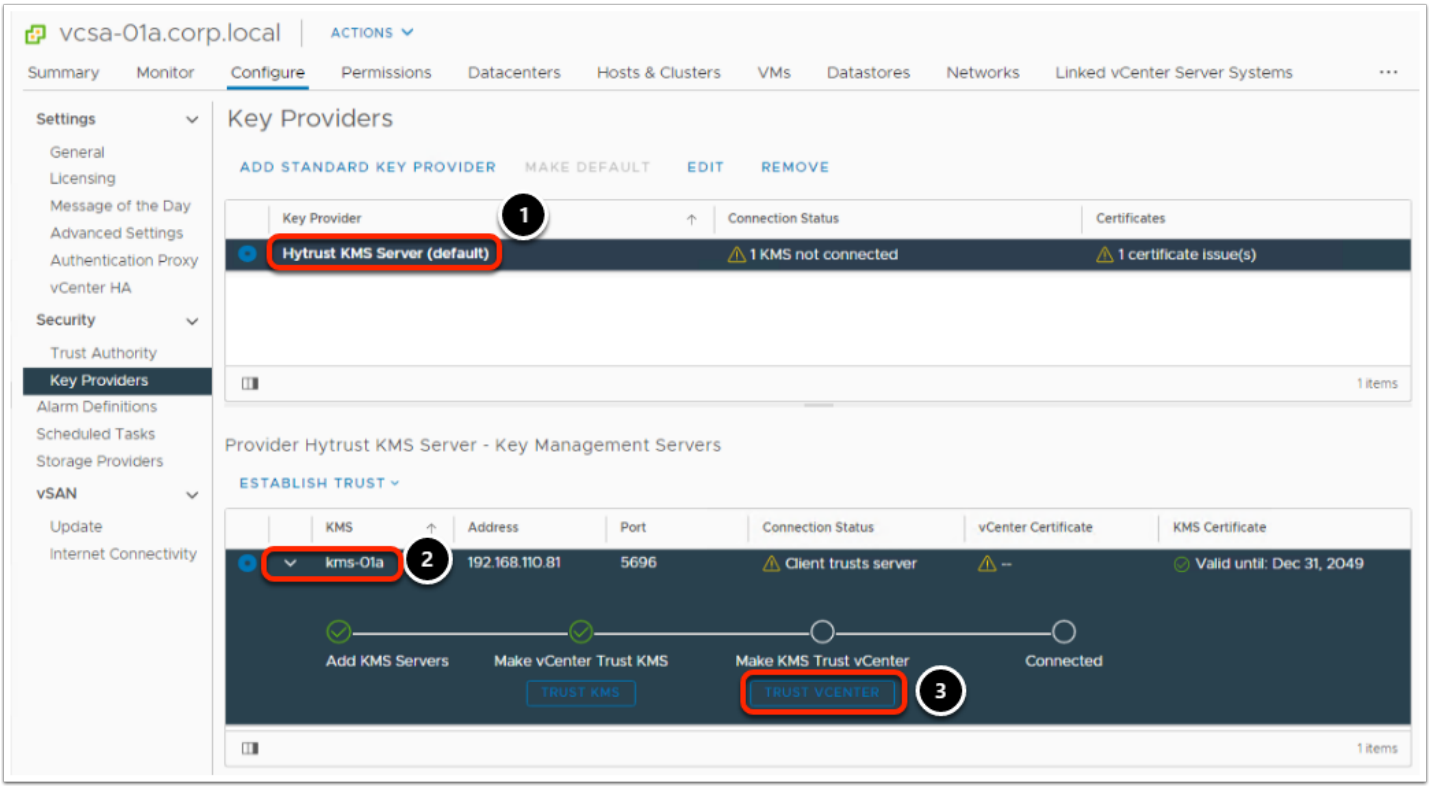
Add Key Management Server



On the Trust Certificate dialog box.

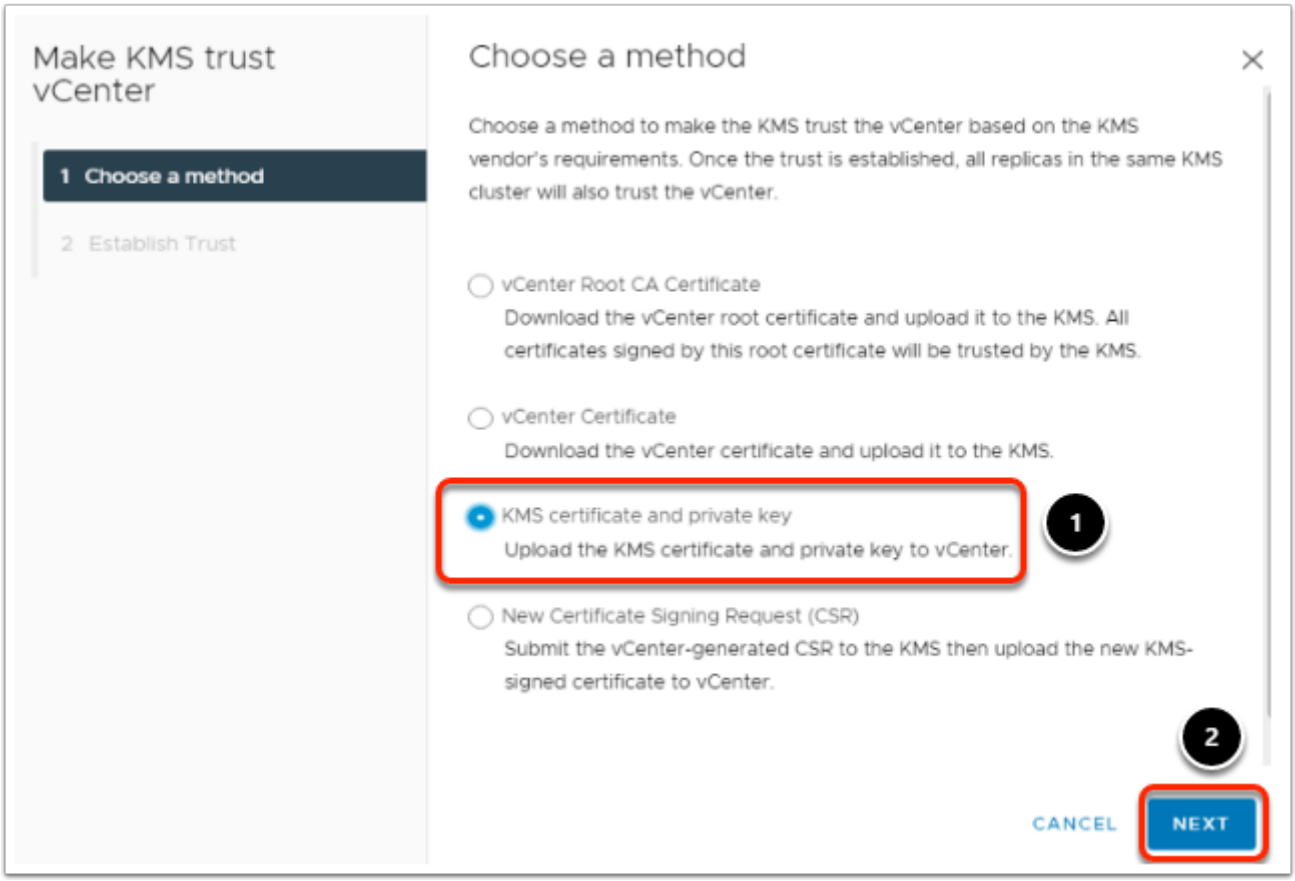
1. Click **TRUST**

Add Key Management Server



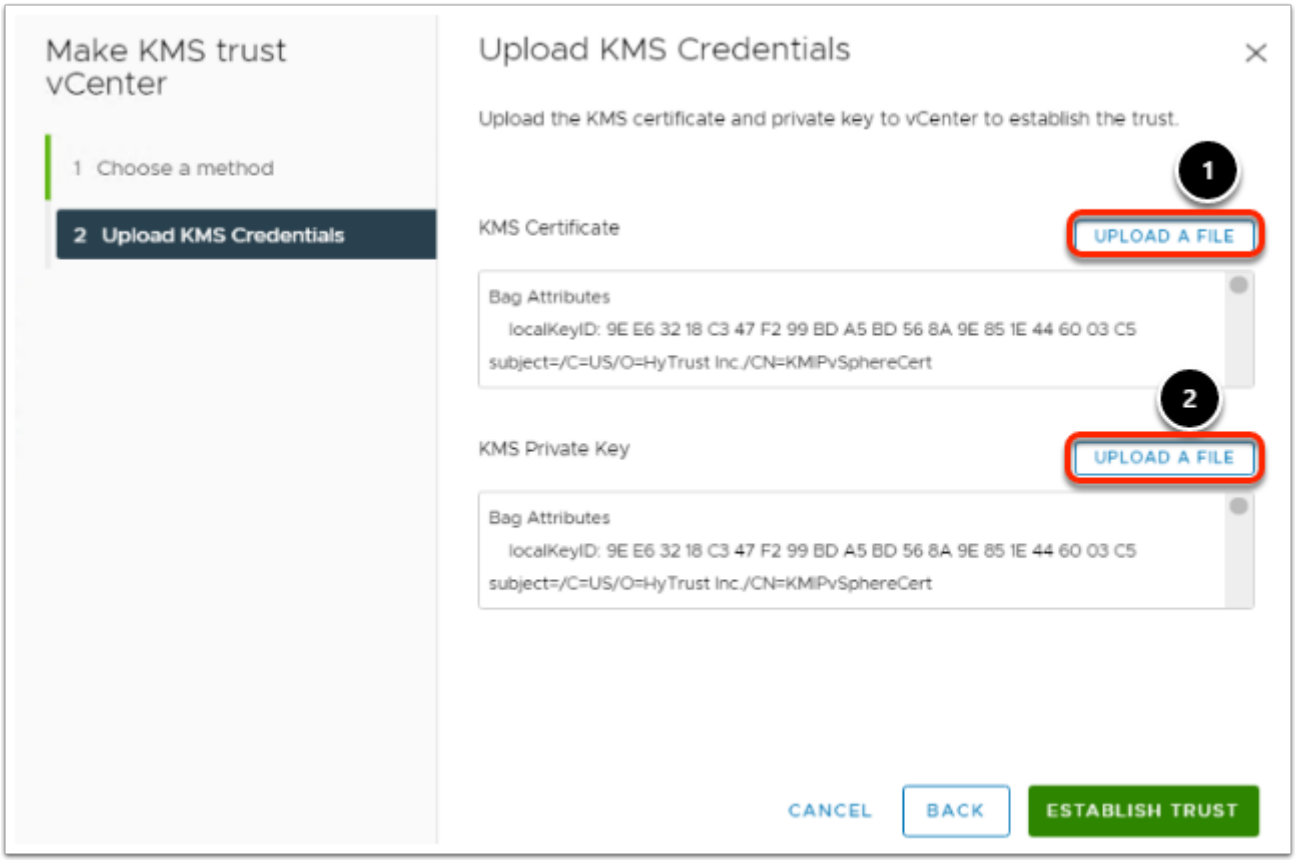
1. On the top panel, select **Hytrust KMS Server (default)**
2. In the bottom panel, Expand the **kms-01a** that you just added to see additional information.
3. Click **TRUST VCENTER**

Add Key Management Server



1. Select **KMS certificate and private key**
2. Click **NEXT**

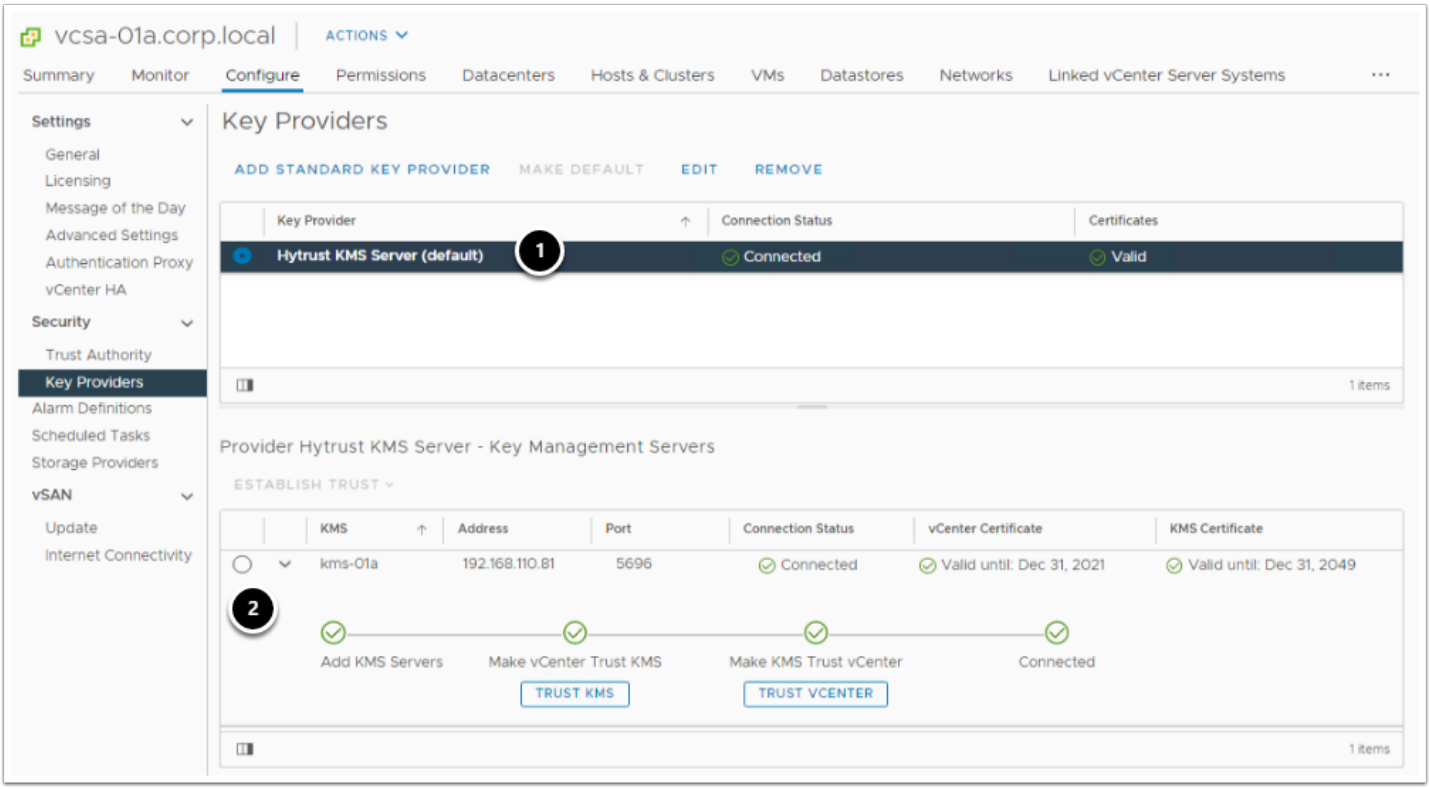
Add Key Management Server



1. For the KMS Certificate, click **UPLOAD A FILE** and Browse to **C:\HytrustLicense\KMIPvSphereCert.pem** and click **Open**
2. For the KMS Private Key, click **UPLOAD A FILE** and Browse to **C:\HytrustLicense\KMIPvSphereCert.pem** and click **Open**

- Click **ESTABLISH TRUST**

Verify Key Management Server



1. Verify that the HyTrust Key Management Server has been added.
2. Verify that the Connection Status is Green and the Certificates are valid.

Enabling vSAN Encryption

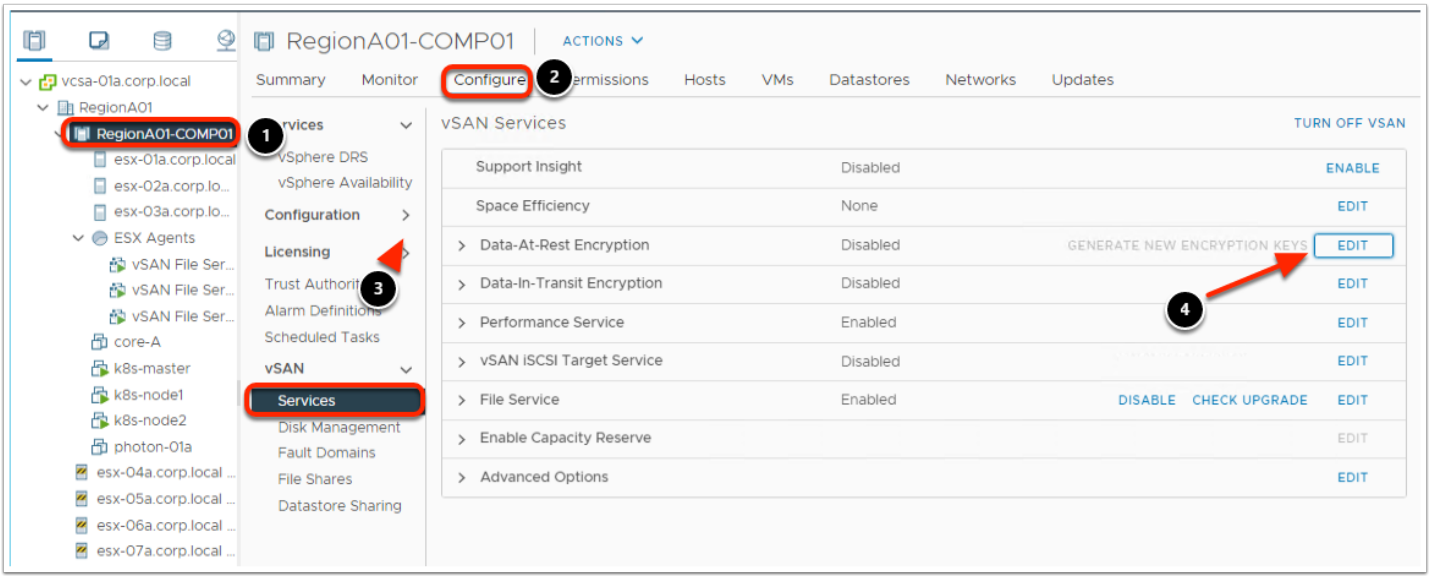
Since vSAN 6.6, we are introducing another option for native data-at-rest encryption, vSAN Encryption.

vSAN Encryption is the industry's first native HCI encryption solution; it is built right into the vSAN software. With a couple of clicks, it can be enabled or disabled for all items on the vSAN datastore, with no additional steps.

Because it runs at the hypervisor level and not in the context of the virtual machine, it is virtual machine agnostic, like VM Encryption.

And because vSAN Encryption is hardware agnostic, there is no requirement to use specialized and more expensive Self-Encrypting Drives (SEDs), unlike the other HCI solutions that offer encryption.

Enabling vSAN Encryption



You can enable encryption by editing the configuration parameters of an existing vSAN cluster.

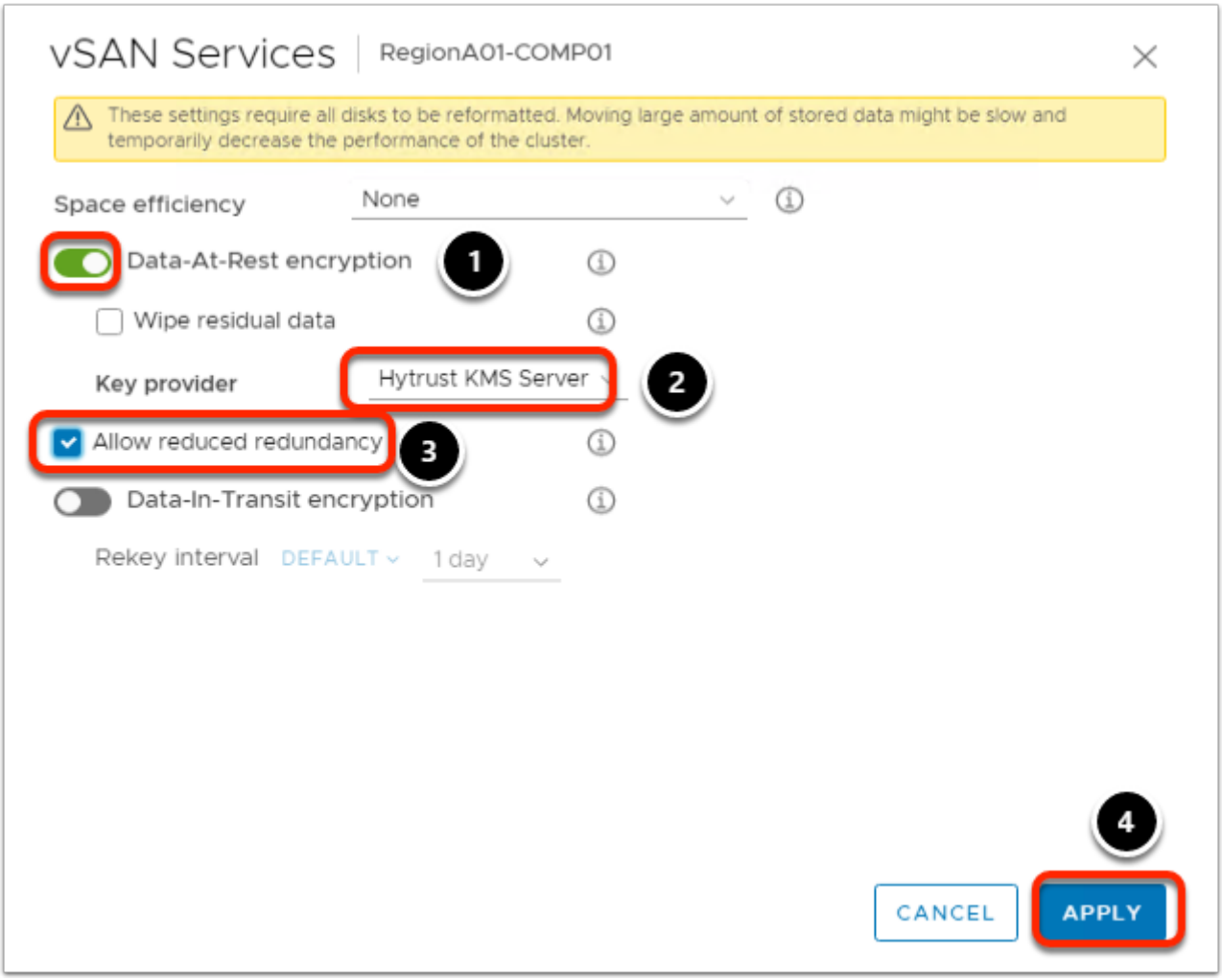
1. Select the Cluster called **RegionA01-COMP01**
2. Select **Configure**
3. Select **vSAN > Services**
4. For **Data-At-Rest Encryption**, Click **EDIT**

Turning on encryption is a simple matter of clicking a checkbox. Encryption can be enabled when vSAN is enabled or after and with or without virtual machines (VMs) residing on the vSAN datastore.

Note that a rolling disk reformat is required when encryption is enabled.

This can take a considerable amount of time – especially if large amounts of existing data must be migrated as the rolling reformat takes place.

Enabling vSAN Encryption



Enabling vSAN Encryption is a one click operation.

1. Click to enable **Data-at-Rest Encryption**
2. Verify the KMS Server is selected (**Hytrust KMS Server**), if you have multiple KMS clusters in your environment, you can choose from here.
3. Select option to **Allow Reduced Redundancy**

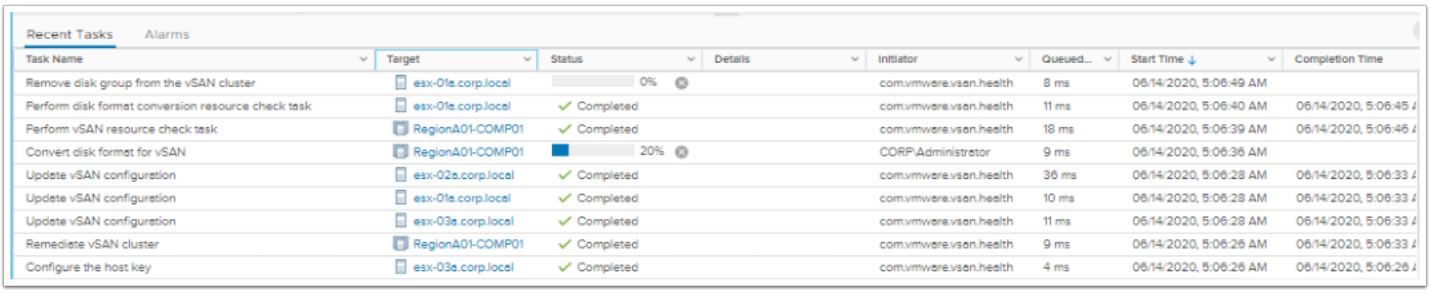
Enabling vSAN Encryption has an option to Erase disk before use. Do not enable this option.

Click on the information button **(i)** for these options to get additional information on these options.

4. Click **APPLY**

The Erase disks before use option will significantly reduce the possibility of data leak and increase the attackers cost to reveal sensitive data. This option will also increase the cost of time to consume disks.

Monitor Recent Tasks



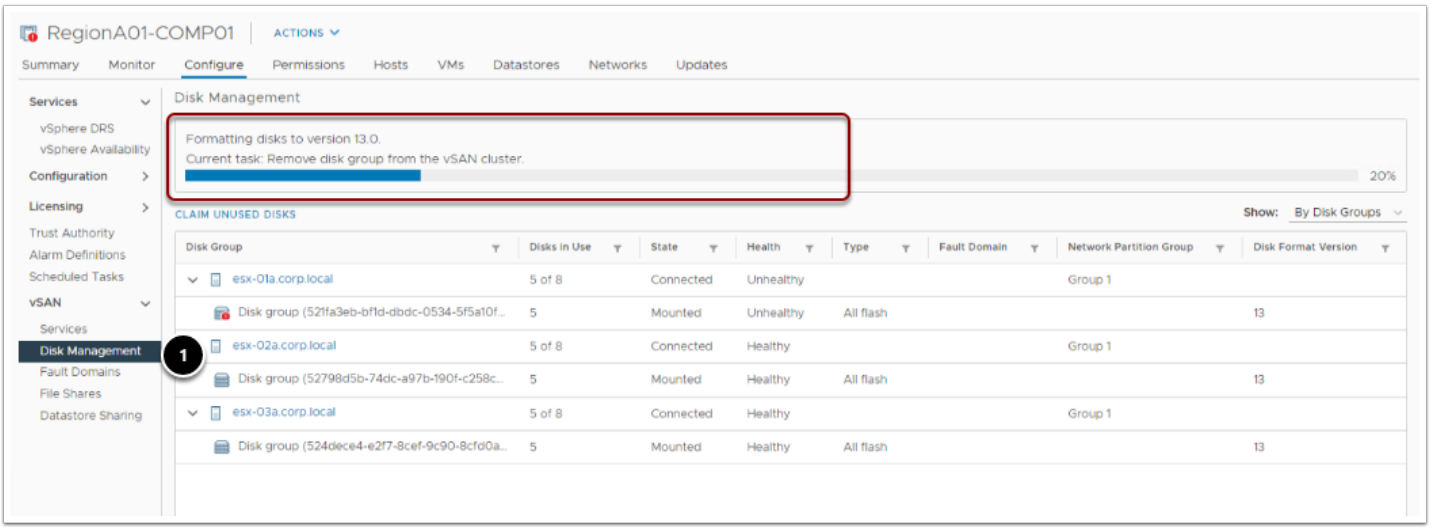
You can monitor the vSAN Encryption process from the **Recent Tasks** window.

To enable vSAN Encryption, the following operation take place.

- Data is migrated from each vSAN Disk Group
- That vSAN Disk Group is removed
- The vSAN Disk Group is recreated with Encryption enabled

This process is repeated for each of the Disk Groups in the vSAN Cluster.

Monitor Formatting Progress



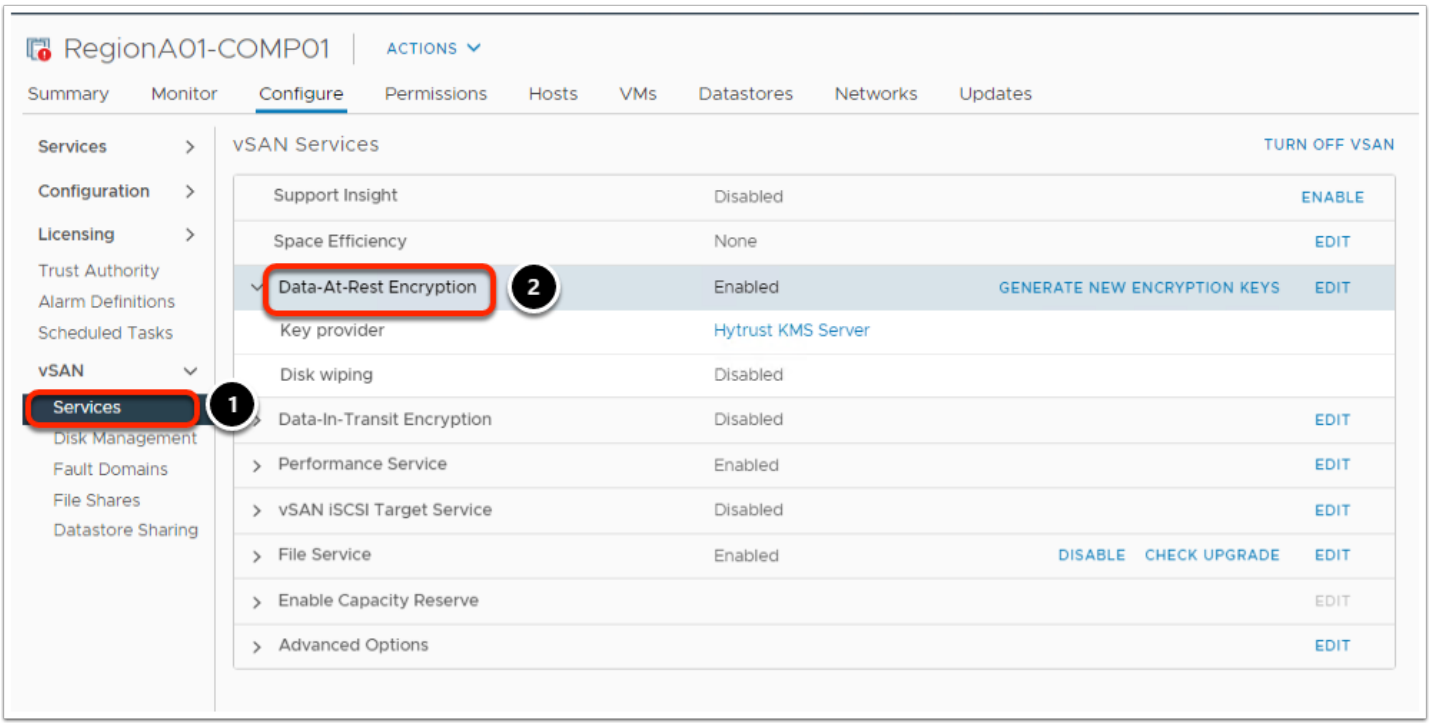
1. You can also monitor the vSAN Encryption process from the **Configure -> vSAN -> Disk Management** screen

Enabling vSAN Encryption will take a little time. Each of the Disk Groups in the vSAN Cluster have to be removed and recreated. If you have vSAN Files

Services enabled, the vSAN File Services agents need to be redeployed again and repair itself.

You can monitor these tasks in the Recent Tasks Window.

Enabling vSAN Encryption



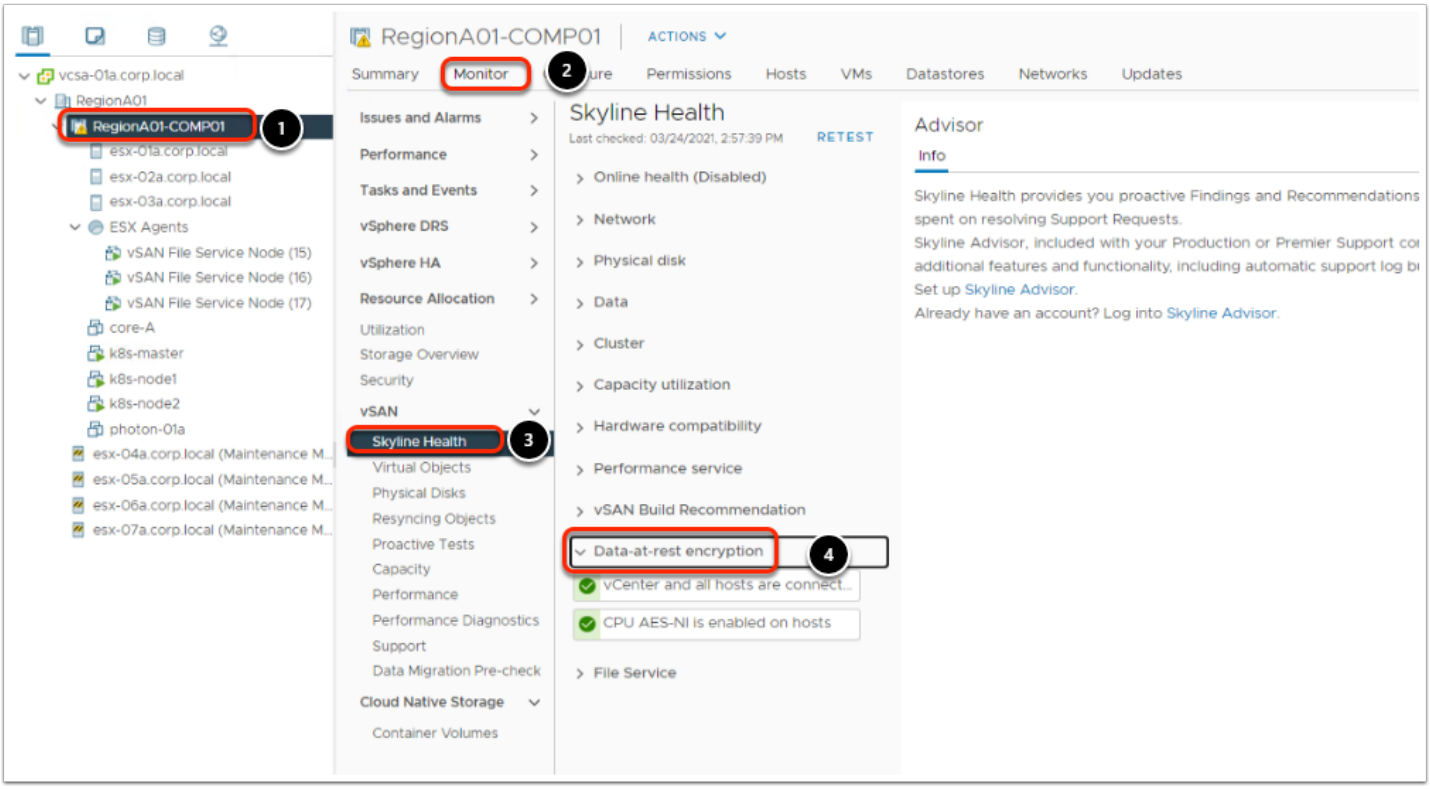
1. Select **vSAN > Services**
2. Expand **Data-At-Rest Encryption**

Once the rolling reformat of all the disk groups task has completed, Encryption of data at rest is enabled on the vSAN cluster.

vSAN encrypts all data added to the vSAN datastore.

You have the option to generate new encryption keys, in case a key expires or becomes compromised.

vSAN Encryption Health Check

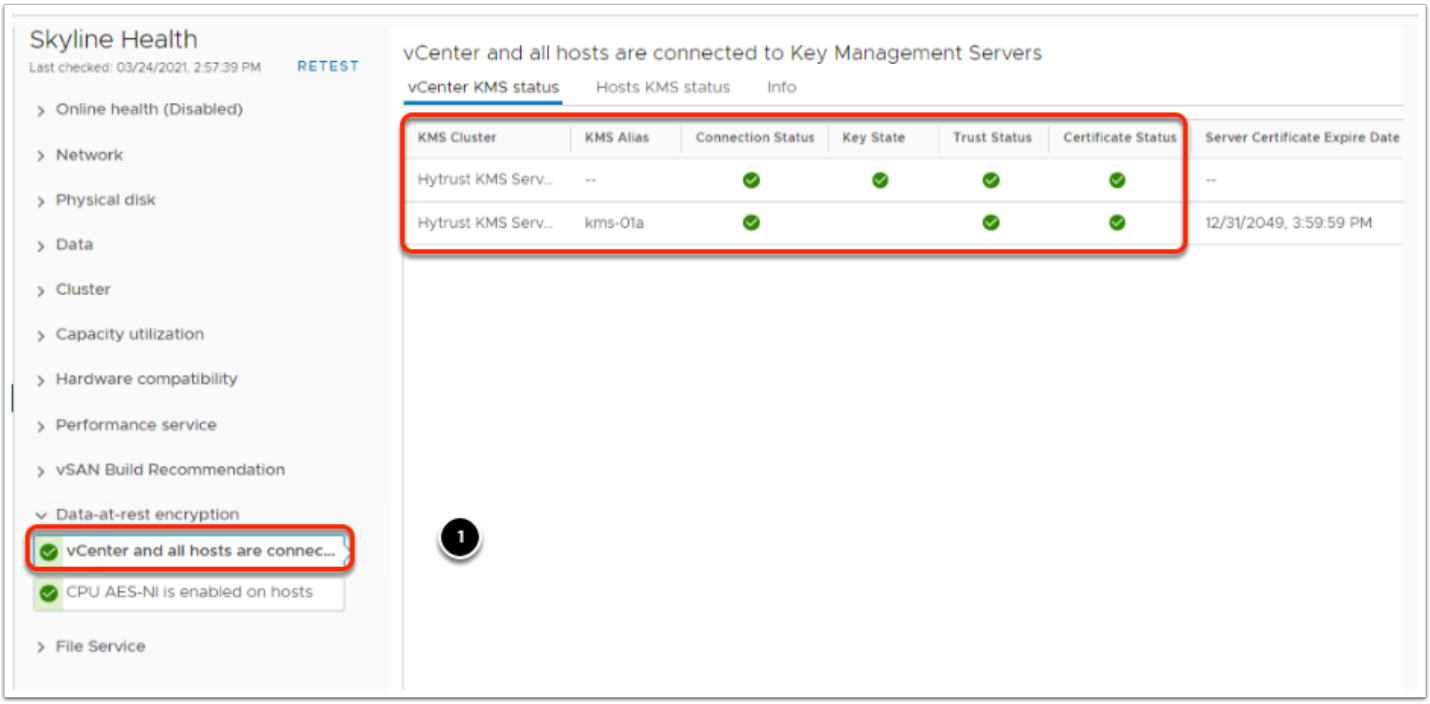


There are vSAN Health Checks to verify that your vSAN Encryption is enabled and healthy.

1. Select the Cluster called **RegionA01-COMP01**
2. Select **Monitor**
3. Select **vSAN > Skyline Health**
4. Expand the **Data-at-rest encryption** health service

There are 2 Health Checks associated with vSAN Encryption.

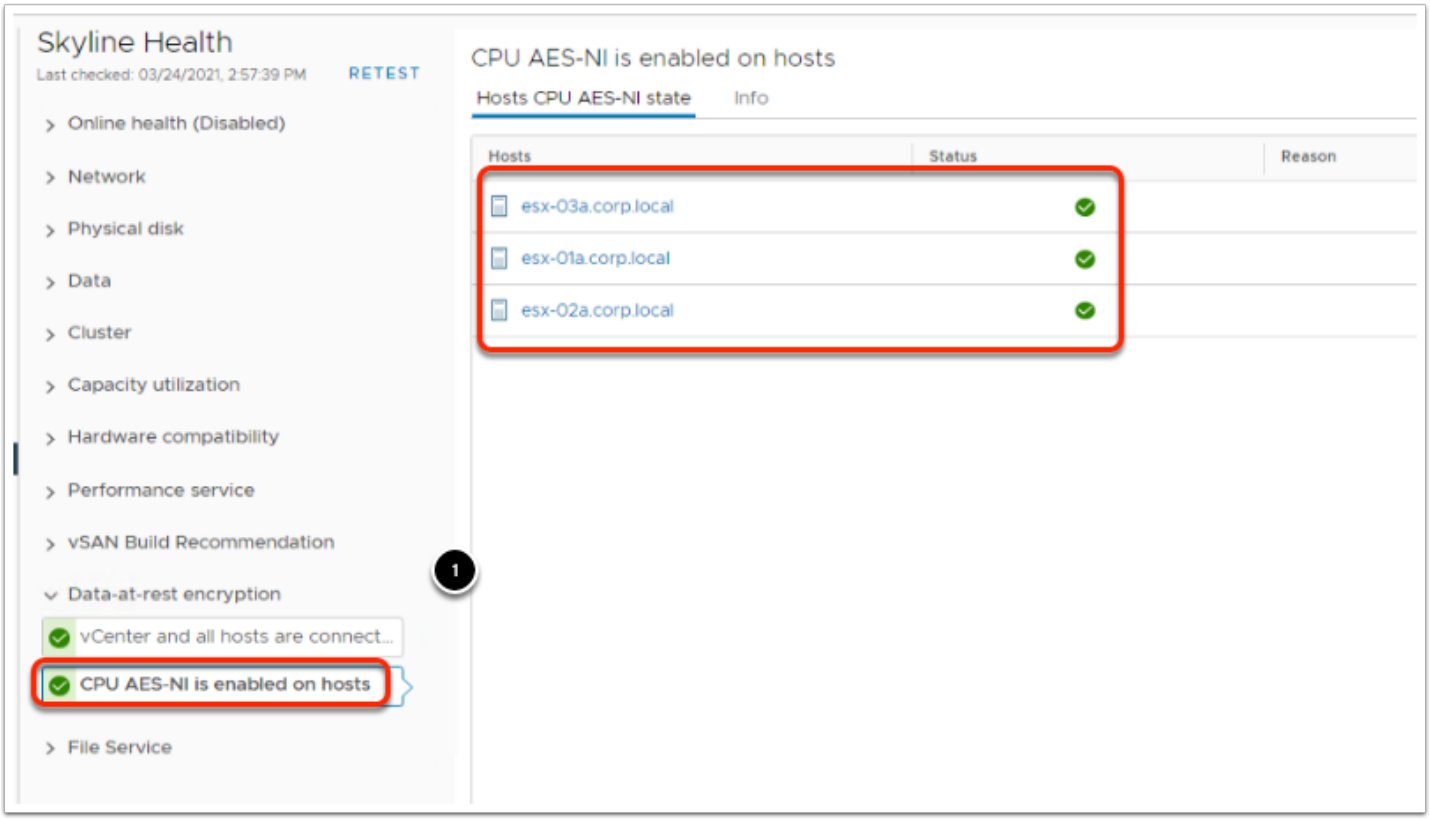
vSAN Encryption Health Check



1. Select **vCenter and all hosts are connected to Key Management Servers**

This vSAN Health Check verifies that the vCenter Server can connect to the Key Management Servers

vSAN Encryption Health Check



1. Select **CPU AES-NI is enabled on hosts**

This check verifies whether ESXi hosts in the vSAN cluster have **CPU AES-NI** feature enabled.

> Advanced Encryption Standard Instruction Set (or the Intel Advanced Encryption Standard New Instructions; AES-NI) is an extension to the x86 instruction set architecture for microprocessors from Intel and AMD. The purpose of the instruction set is to improve the speed of applications performing encryption and decryption using the Advanced Encryption Standard (AES).

Conclusion

In this lesson we explored vSAN security parameters including DISA STIG (FIPS 104-2) Validation and vSAN Data-at-rest Encryption.

You Finished Module 8

Congratulations on completing Module 8.

If you are looking for additional information on topic:

- VMware Security Certification
- vSAN Datasheet (including FIPS 140-2 Information)
- vSAN Encryption
- Go to VMware Storage Hub for all things related to vSAN.

Test Your Skills

Now that you've completed this lab, try testing your skills with VMware Odyssey, our newest Hands-on Labs gamification program. We have taken Hands-on Labs to the next level by adding gamification elements to the labs you know and love. Experience the fully automated VMware Odyssey as you race against the clock to complete tasks and reach the highest ranking on the leaderboard. Try the vSAN Odyssey lab

- **HOL-2108-05-ODY - VMware Odyssey - vSAN - Advanced Game**

How to End Lab

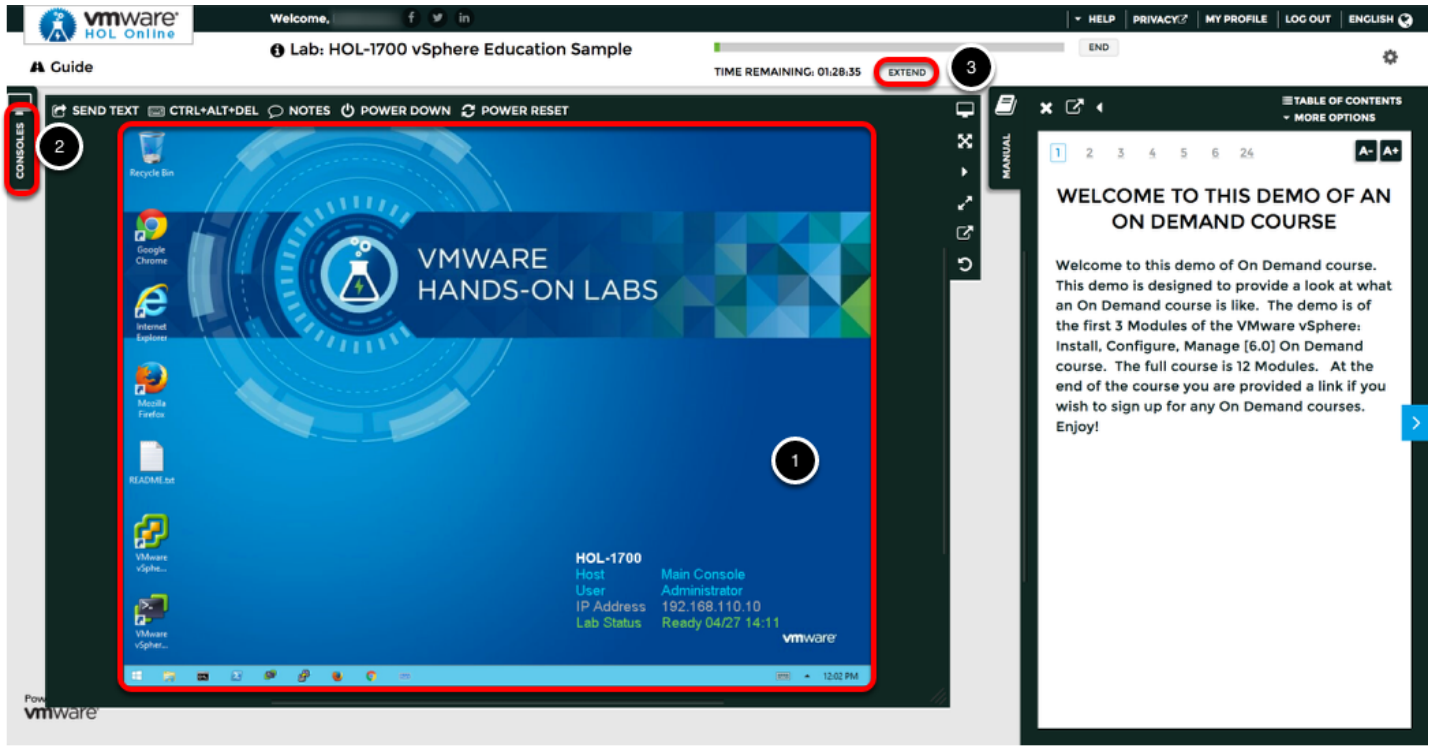
If you would like to end your lab click on the **END** button.

Appendix

Hands-on Labs Interface

Welcome to Hands-on Labs! This overview of the interface and features will help you to get started quickly. Click next in the manual to explore the Main Console.

Location of the Main Console



1. The area in the large RED box contains the Main Console. The Lab Manual is on the tab to the right of the Main Console.
2. Some labs have additional consoles found on separate tabs in the upper left. The lab manual will direct you to open another specific console if necessary.
3. Your lab starts with 90 minutes on the timer. The lab can not be saved. Your lab will end when the timer expires. Click the **EXTEND** button to increase the time allowed. If you are at a VMware event, you can extend your lab time twice up to 30 minutes. Each click gives you an additional 15 minutes. Outside of VMware events, you can extend your lab time up to 9 hours and 30 minutes. Each click gives you an additional hour.

Alternate Methods of Keyboard Data Entry

In this lab you will input text into the Main Console. Besides directly typing it in, there are two very helpful methods of entering data which make it easier to enter complex data.

Click and Drag Lab Manual Content Into Console Active Window

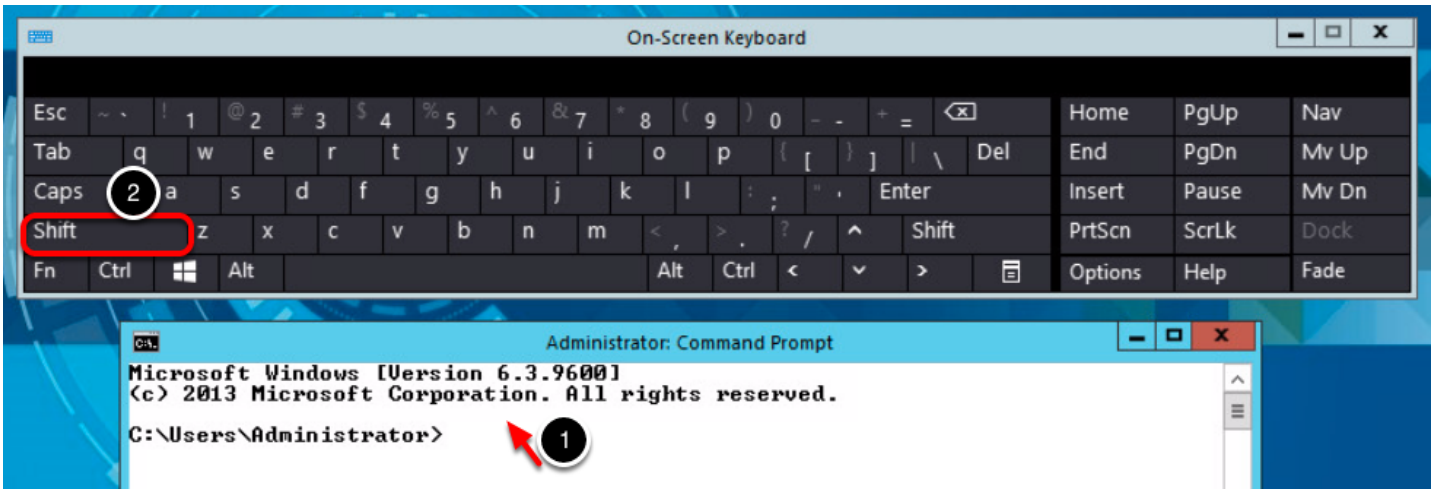
You can also click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

Accessing the Online International Keyboard

You can also use the Online International Keyboard found in the Main Console.

1. Click on the keyboard icon found on the Windows Quick Launch Task Bar.

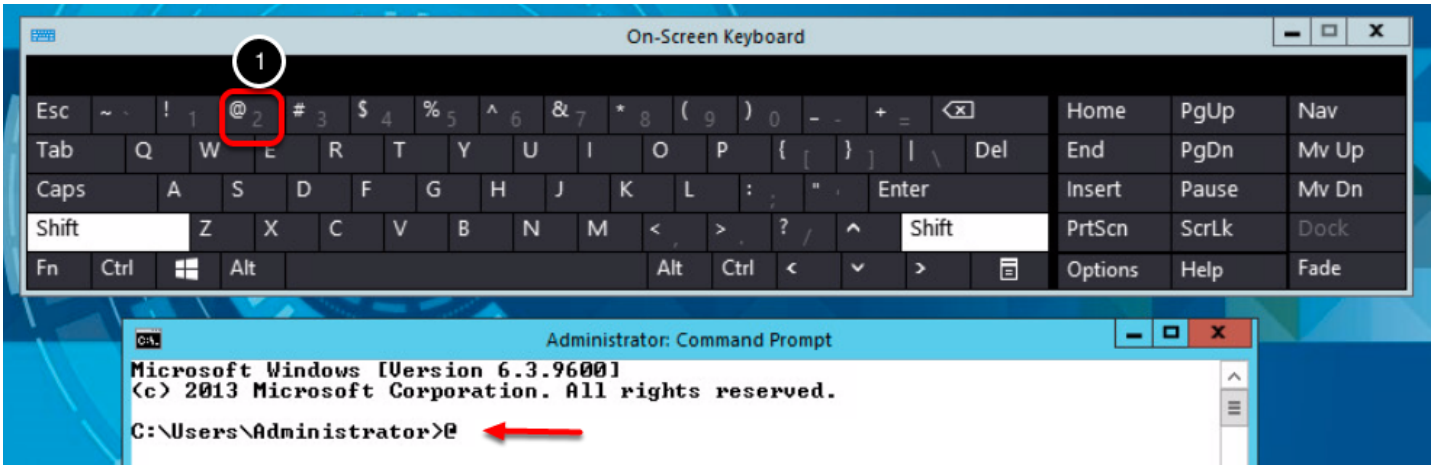
Click once in active console window



In this example, you will use the Online Keyboard to enter the "@" sign used in email addresses. The "@" sign is Shift-2 on US keyboard layouts.

1. Click once in the active console window.
2. Click on the **Shift** key.

Click on the @ key



1. Click on the "**@**" key.

Notice the @ sign entered in the active console window.

Return to Lab Guidance

[Click Here](#) to return to the Lab Guidance page.

Conclusion

Thank you for participating in the VMware Hands-on Labs. Be sure to visit http://hol.vmware.com/ to continue your lab experience online.

Lab SKU: HOL-2108-01-HCI

Version: 20210610-174032