VMware Virtual SAN 6.1 Proof Of Concept Guide

January 2016 Edition

Cormac Hogan
David Boone
Paudie O'Riordan
Brad Garvey

vmware®

Contents

1. Introduction

VMware customers love the simplicity, performance and integration of VMware® Virtual SAN™ since its launch.

Most customers choose to evaluate Virtual SAN before using it for production – always a good idea. We've made a list of issues occasionally encountered as people go through this process.

Follow this guide, and you'll have a great evaluation.

2. Before You Start

Plan on testing a reasonable hardware configuration that resembles what you plan to use in production. Refer to the VMware Virtual SAN 6.0 Design and Sizing Guide for information on supported hardware configurations, and consideration when deploying Virtual SAN.

2.1 All Flash or Hybrid

There are a number of additional considerations if you plan to deploy an all-flash Virtual SAN solution:

- All-flash is available in Virtual SAN since version 6.0.
- It requires a 10Gb Ethernet network; it is not supported with 1Gb NICs.
- The maximum number of all-flash hosts is 64.
- Flash devices are used for both cache and capacity.
- Flash read cache reservation is not used with all-flash configurations.
- There is a need to mark a flash device so it can be used for capacity – this is covered in the Virtual SAN Administrators Guide.
- Endurance now becomes an important consideration both for cache and capacity layers.

2.2 Three-node versus Four-node or Greater

While Virtual SAN fully supports 3-node configurations, they can behave differently than configurations with 4 or greater nodes. In particular, in the event of a failure you do not have the ability to rebuild components on another host in the cluster to tolerate another failure. Also with 3-node configurations, you will not have the ability to migrate all data from a node during maintenance. This is because virtual machines on a 3 node cluster cannot be configured to tolerate more than one failure.

If you plan to deploy a 3-node cluster, then that is what you should test. But if you plan on deploying larger clusters, we strongly recommend testing 4 or more nodes.

Further considerations with three-node clusters are covered in the failure testing section of this document.

For the purposes of this proof-of-concept guide, a 4-node configuration is used. However, the cluster size is 3-nodes to begin with, and the fourth node will be added during the course of the proof-of-concept.

2.3 Follow the vSphere Compatibility Guide Precisely

2.3.1 Why Is This Important?

We cannot overstate the importance of following the vSphere Compatibility Guide (VCG) for Virtual SAN to the letter. A significant number of our support requests are ultimately traced back to customers failing to follow these very specific recommendations. This on-line tool is regularly updated to ensure customers always have the latest guidance from VMware available to them.

2.3.2 Hardware, Drivers, and Firmware

The VCG makes very specific recommendations on hardware models for Storage I/O controllers, SSDs, PCI-E flash cards and disk drives. It also specifies which drivers have been fully tested, and – in many cases – identifies the firmware level required. The most direct way to check the controller's firmware version is by interrupting the boot process and looking into the controller's BIOS settings. The VMware Virtual SAN Diagnostics and Troubleshooting Reference Manual contains information about using '*esxcli hardware pci list*' and '*vmkload_mod -s*' to find the I/O controller's driver version.

2.3.3 Out-of-box Drivers versus Inbox (Shipped with ESXi and Listed on VCG)

Storage controller drivers provided as part of a server vendor's vSphere distribution may or may not be certified for use with Virtual SAN. When in question, go with the driver version specified in the VCG.

Some SSD and flash vendors are revising their firmware frequently, often with significant performance enhancements resulting. Check the VCG regularly for driver and firmware updates.

Although it's well documented, people sometimes forget that Virtual SAN can't claim a disk that already has a partition on it. So make sure to check that your disks are clean before trying to use them with Virtual SAN.

2.3.4 RAID-0 versus Pass-Through for Disks

The VCG will tell you if a controller supports RAID-0 or pass-through when presenting disks to ESXi hosts. RAID-0 is only supported when pass-through is not possible.

Check that you are using the correct configuration and that the configuration is uniform across all nodes.

2.3.5 Controller Configuration

Keep the controller configuration relatively simple. For controller with cache, either disable it, or – if that is not possible - set it to 100% read. For other vendor specific controller features such as HP SSD Smart Path, we recommend disabling them. This may only be possible from the BIOS of the controller in many cases.

2.4 Use Supported vSphere Software Versions

It is highly recommended that anyone who is considering an evaluation of Virtual SAN should pick up the latest versions of software. VMware continuously fixes issues encountered by customers, so by using the latest version of the software, you avoid issues already fixed.

In this proof-of-concept guide, Virtual SAN version 6.1 from vSphere 6.0u1 is used.

3. Virtual SAN POC Setup Assumptions and Prerequisites

The following assumptions are being made with regards to the deployment:

- Four servers are available, and are compliant with the Virtual SAN HCL.
- All servers have had ESXi 6.0u1 deployed. These steps will not be covered in this POC guide.
- A 6.0u1 vCenter Server has been deployed to manage these four ESXi hosts. These steps will not be covered in this POC guide.
- Services such as DHCP, DNS and NTP are available in the environment where the POC is taking place.
- Three out of four ESXi hosts should be placed in a cluster in vCenter.
- If using HP storage controllers, install the *hpssacli* VIB.
- The cluster must not have any features enabled, such as DRS, HA or Virtual SAN. These will be done throughout the course of the POC.
- Each host must have a management network and a vMotion network already configured. There is no Virtual SAN network configured. This will be done as part of the POC.
- For the purposes of testing Storage vMotion operations, an additional datastore type, such as NFS or VMFS, should be presented to all hosts. This is an *optional* POC exercise.
- A set of IP addresses, one per ESXi host will be needed for the Virtual SAN traffic VMkernel ports. The recommendation is that these are all on the same VLAN and network segment.

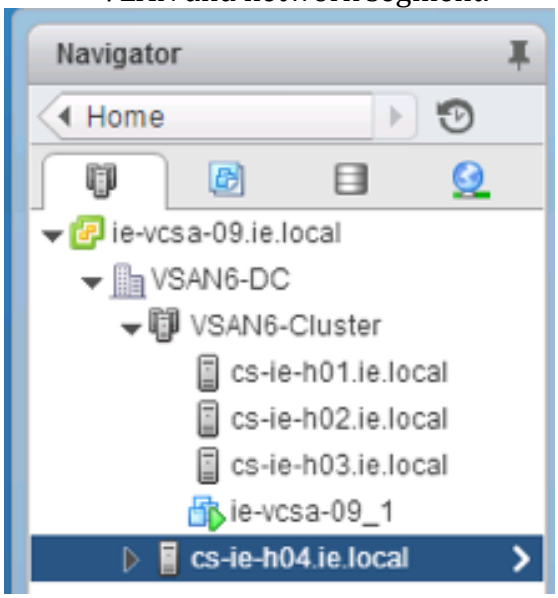


Figure 3.1: Initial cluster configuration example

From a network perspective, it is optimal to separate the Virtual SAN network from the management and vMotion networks. Below, management, VM and vMotion networks have their own uplinks via VSS (Virtual Standard Switch) vSwitch0.

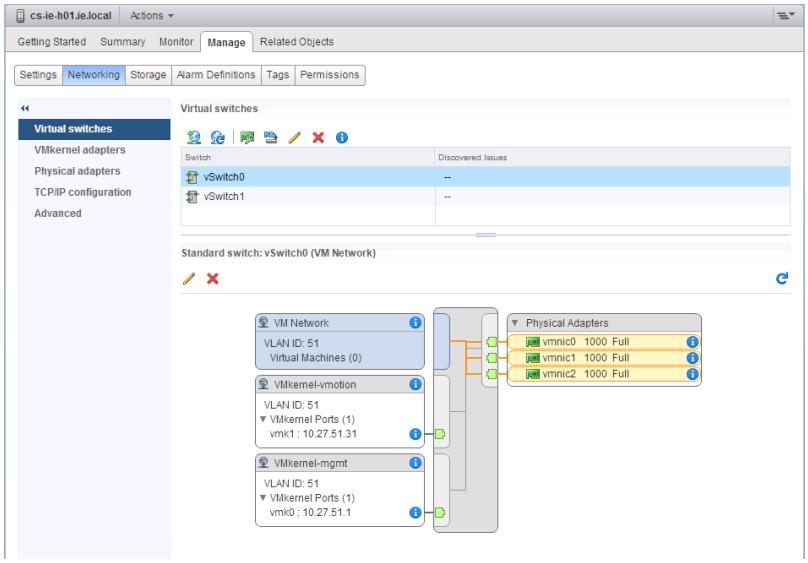


Figure 3.2: Initial host network configuration example – non-Virtual SAN networks

In this POC example, the Virtual SAN network is on its own VSS (vSwitch1) and the VSS has a number of uplinks in this configuration. You do not need to follow this design and you may use a much simpler "single uplink" VSS in your POC if you wish. In the next section, the steps to create a Virtual SAN VMkernel network interface will be shown.

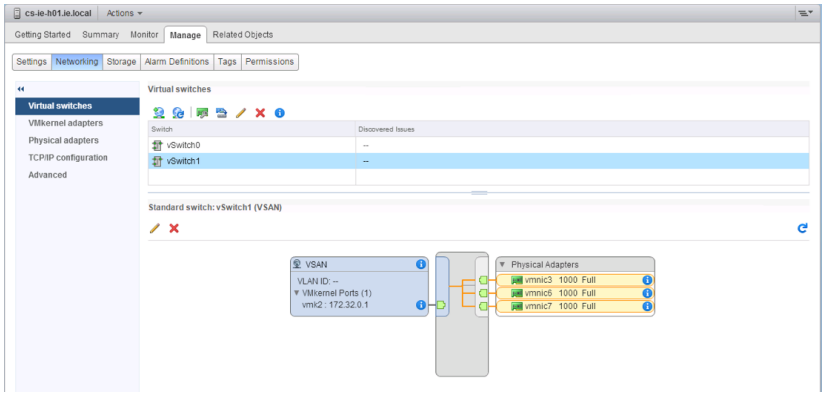
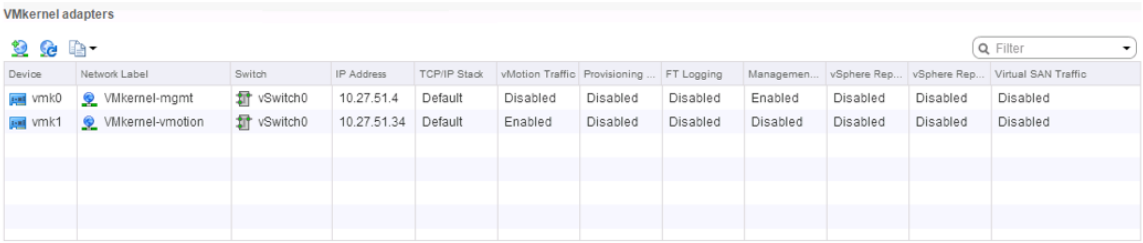


Figure 3.3: Initial host network configuration example – Virtual SAN network

It is considered best practice to dedicate 1GbE NICs to the Virtual SAN network. When using 10GbE networks, multiple traffic types may share the same uplink.



Device	Network Label	Switch	IP Address	TCP/IP Stack	vMotion Traffic	Provisioning ...	FT Logging	Managemen...	vSphere Rep...	vSphere Rep...	Virtual SAN Traffic
vmk0	VMkernel-mgmt	vSwitch0	10.27.51.4	Default	Disabled	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled
vmk1	VMkernel-vmotion	vSwitch0	10.27.51.34	Default	Enabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled

Figure 3.4: Initial host network configuration example

If you plan to use distributed switches in your POC, details on how to migrate from a VSS to a distributed switch are shown in appendix B of this guide.

4. Virtual SAN Network Setup

Anyone implementing a Virtual SAN POC should be aware of the VMware Virtual SAN 6.0 Design and Sizing Guide. The guide can be found here:

All ESXi hosts in a Virtual SAN Cluster communicate over a Virtual SAN network. This is a new VMkernel port type introduced in vSphere 5.5 specifically for Virtual SAN. The following example will demonstrate how to configure a Virtual SAN network on an ESXi host.

4.1 Advantages of Distributed Switch versus Standard Switch

If the plan is to test Network I/O Control (NIOC) functionality to provide Quality of Service (QoS) on the Virtual SAN traffic, then a distributed virtual switch (DVS) will be required. If you do not plan to use NIOC, then the evaluation may be done with a standard switch (VSS).

4.2 Creating a VMkernel Port for Virtual SAN

In many deployments, Virtual SAN may be sharing the same uplinks as the management and vMotion traffic, especially when 10GbE NICs are utilized. Later on, we will look at an optional workflow that migrates the standard vSwitches to a distributed switch for the purpose of providing Quality Of Service (QoS) to the Virtual SAN traffic through a feature called Network I/O Control. This is only available on distributed switches.

The Virtual SAN license also includes entitlement to distributed switch, even on the lower editions of vSphere (for use on the Virtual SAN enabled cluster only).

However, the assumption for this POC is that there is already a standard vSwitch created which contains the uplinks that will be used for Virtual SAN traffic. In this example, a separate vSwitch (vSwitch1) with dedicated 1Gbe NICs has been created for Virtual SAN traffic, while the management and vMotion network use different uplinks on a separate standard vSwitch.

To create a Virtual SAN VMkernel port, follow these steps:

Select an ESXi host in the inventory, then navigate to Manage > Networking > VMkernel Adapters. Click on the icon for "Add host networking", as highlighted below:

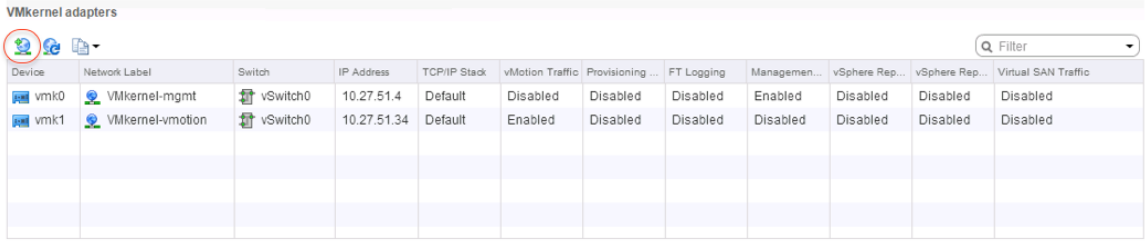


Figure 4.1: Add host networking

Ensure that VMkernel Network Adapter is chosen.

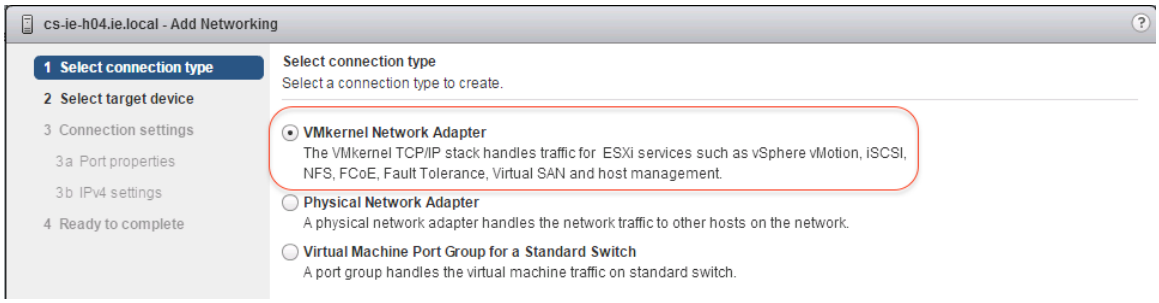


Figure 4.2: Select VMkernel Network Adapter type

The next step gives you the opportunity to build a new standard vSwitch for the Virtual SAN network traffic. In this example, an already existing vSwitch1 contains the uplinks for the Virtual SAN traffic. If you do not have this already configured in your environment, you can use an already existing switch or select the option to create a new standard vSwitch. When you are limited to 2 x 10GbE uplinks, it makes sense to use the same VSS. When you have many uplinks, some dedicated to different traffic types (as in this example), management can be a little easier if different VSS with their own uplinks are used for the different traffic types.

As there is an existing vSwitch in our environment that contains the network uplinks for the Virtual SAN traffic, the "browse" button is used to select it as shown below.

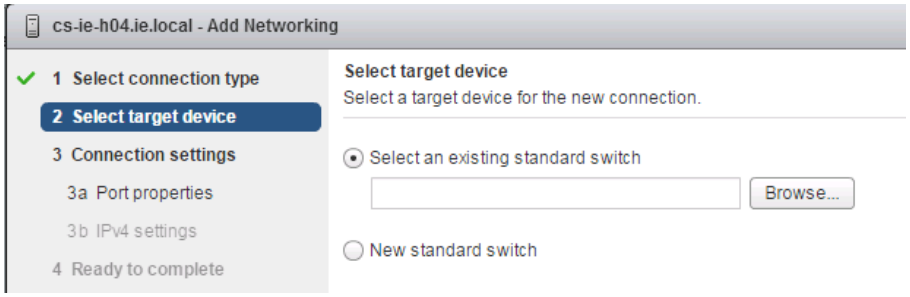


Figure 4.3: Select and existing standard switch via "Browse" button

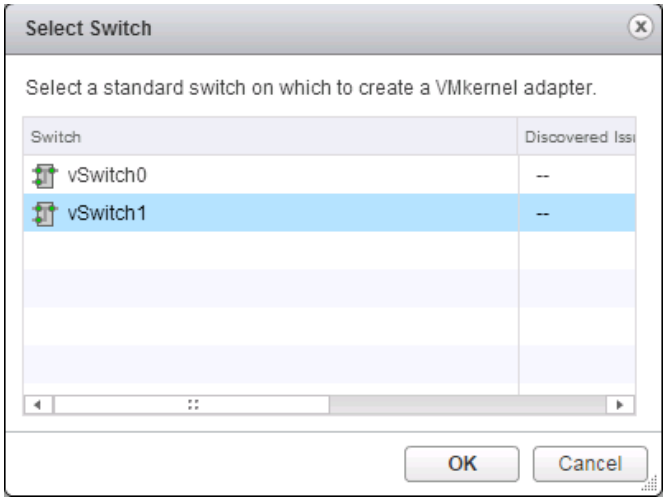


Figure 4.4: Choose a vSwitch

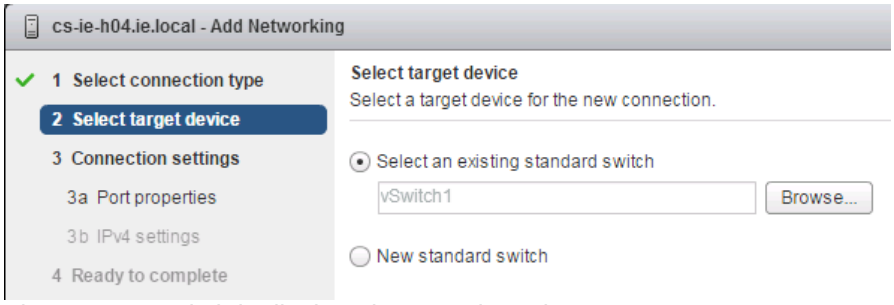


Figure 4.5: vSwitch is displayed once selected

The next step is to setup the VMkernel port properties, and choose the services, such as Virtual SAN traffic. This is what the initial port properties window looks like.

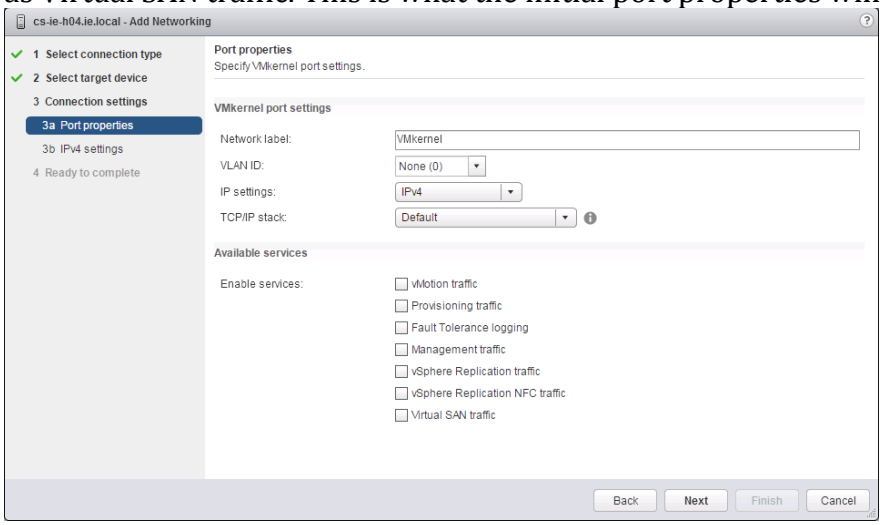


Figure 4.6: Default port properties

Here is what it looks like when populated with Virtual SAN specific information.

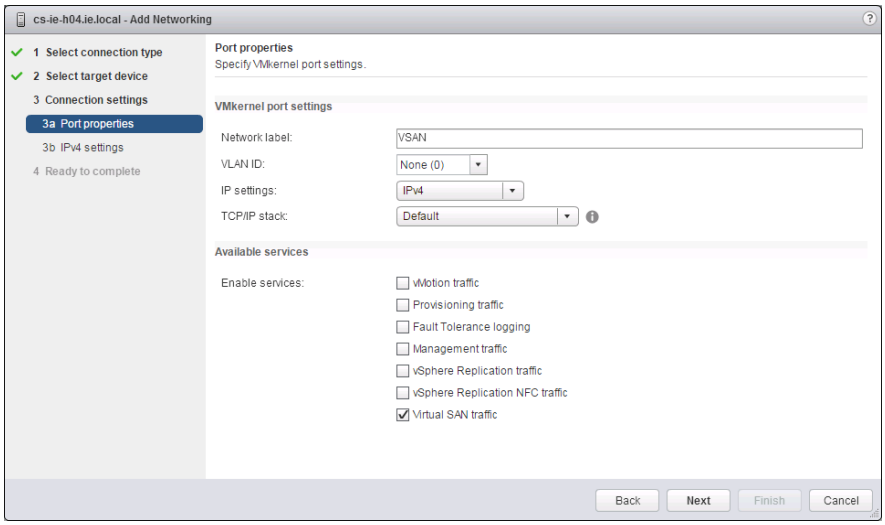


Figure 4.7: Port properties configured for Virtual SAN traffic

In the above example, the network label has been designated "Virtual SAN", and the Virtual SAN traffic does not run over a VLAN. If there is a VLAN used for the Virtual SAN traffic in your POC, change this from "None (0)" to an appropriate VLAN ID.

The next step is to provide an IP address and subnet mask for the Virtual SAN VMkernel interface. As per the assumptions and pre-requisites section earlier, you should have these available before you start. At this point, you simply add them, one per host by clicking on "Use static IPv4 settings" as shown below. Alternatively, if you plan on using DHCP IP addresses, leave the default setting which is "Obtain IPv4 settings automatically".

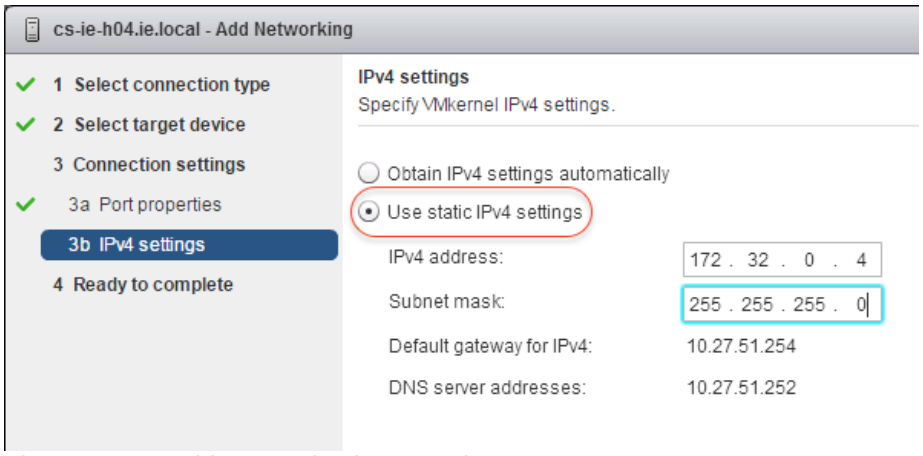


Figure 4.8: IP address and subnet mask

The final window is a review window. Here you can check that everything is as per the options selected throughout the wizard. If anything is incorrect, you can navigate back through the wizard. If everything looks like it is correct, you can click on the "Finish" button.

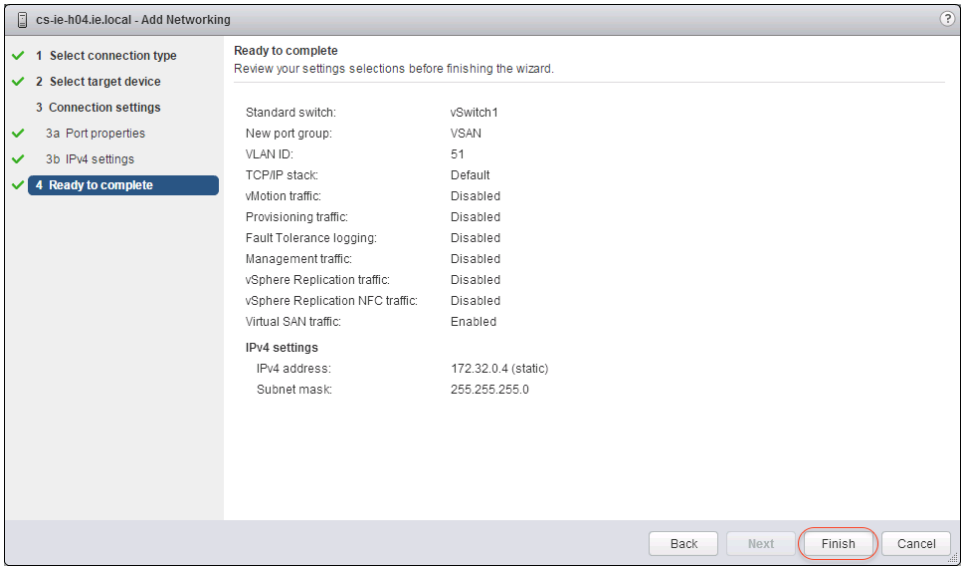


Figure 4.9: Review window

If the creation of the VMkernel port is successful, it will appear in the list of VMkernel ports, as shown below.

Device	Network Label	Switch	IP Address	TCP/IP Stack	vMotion Traffic	Provisioning ...	FT Logging	Managemen...	vSphere Rep...	vSphere Rep...	Virtual SAN Traffic
vmk0	VMkernel-mgmt	vSwitch0	10.27.51.4	Default	Disabled	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled
vmk1	VMkernel-vmotion	vSwitch0	10.27.51.34	Default	Enabled	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled
vmk2	VSAN	vSwitch1	172.32.0.4	Default	Disabled	Disabled	Disabled	Disabled	Disabled	Disabled	Enabled

Figure 4.10: VMkernel adapters with new Virtual SAN VMkernel adapter

That completes the Virtual SAN networking setup for that host. You must now repeat this for all other ESXi hosts, including the host that is not currently in the cluster you will use for Virtual SAN.

If you wish to use a DVS (distributed vSwitch), the steps to migrate from standard vSwitch (VSS) to DVS are documented in Appendix B—Migrating from Standard vSwitch to Distributed of this POC Guide.

5. Enabling Virtual SAN on the Cluster

Enabling Virtual SAN is quite simple, and can be done in just a few clicks in the vSphere web client. However, there is one decision that needs to be made when enabling Virtual SAN, and that is whether you want Virtual SAN to claim all of the unused local storage on the ESXi hosts, or if you (as the administrator) wish to decide which physical disks and flash devices to use for the Virtual SAN datastore.

To enable Virtual SAN, select the cluster object in the vCenter inventory, then select the Manage tab > Settings > Virtual SAN > General, as shown below.

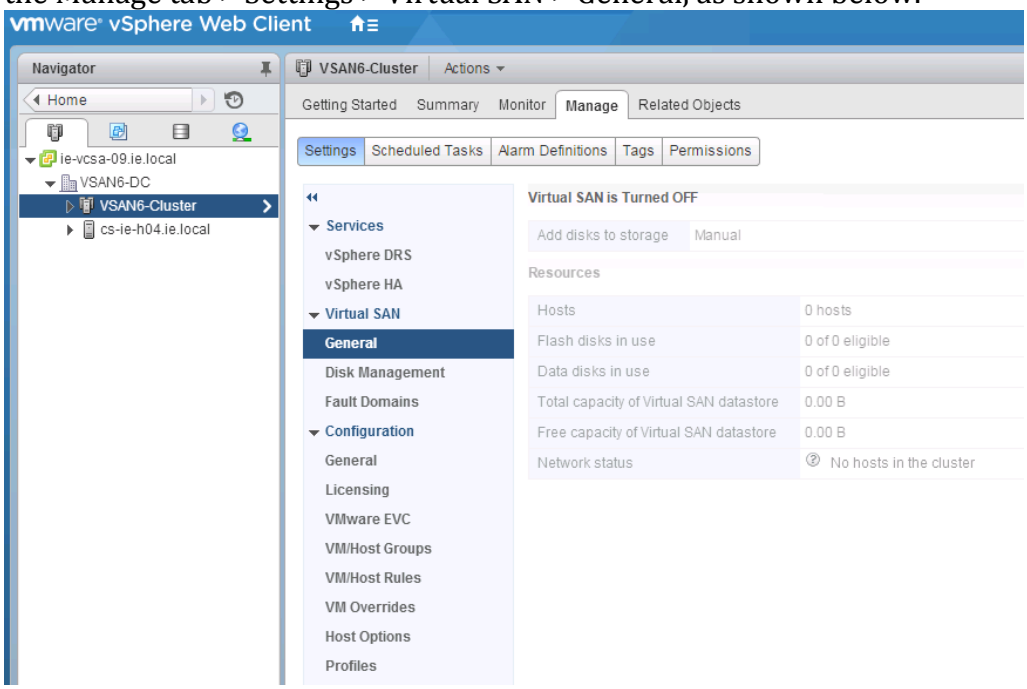


Figure 5.1: Virtual SAN is Turned OFF

To turn on (or enable) Virtual SAN, there is an "Edit" button located in the top right as shown below.

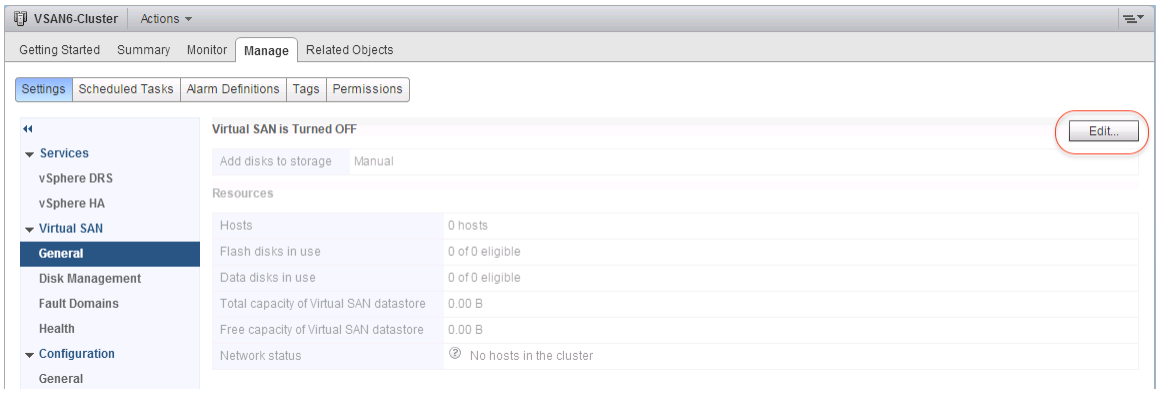


Figure 5.2: Virtual SAN Edit button

Simply click on this "Edit" button to start the process of enabling Virtual SAN. This opens the following pop-up, which provides the option to turn on Virtual SAN, and then add disks to storage manually or automatically.

Figure 5.3: Edit Virtual SAN Settings

When "Turn ON Virtual SAN" option is checked, the option to select a manual or automatic option is available for selection.

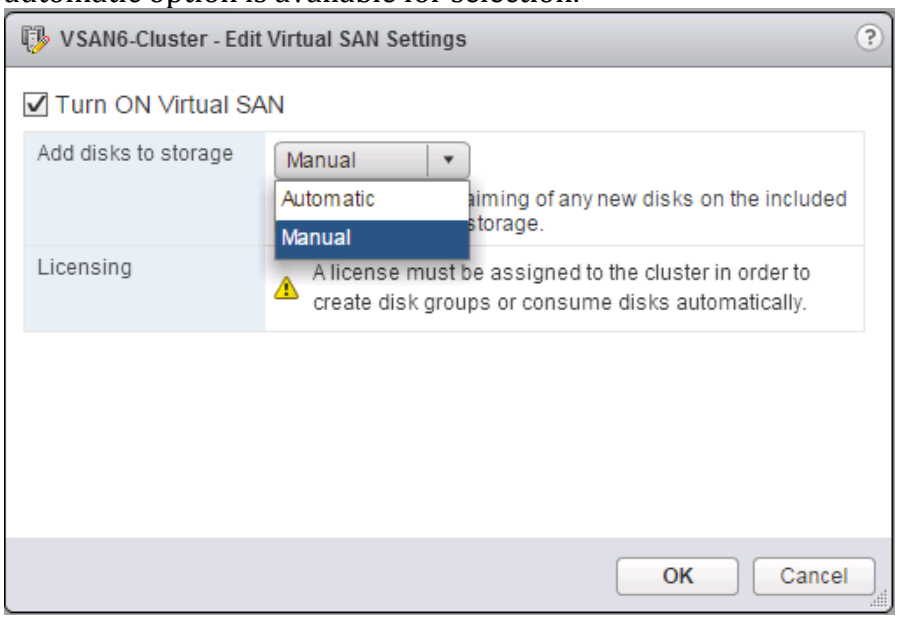


Figure 5.4: Add disks to storage

If networking is correctly configured, and each of the ESXi hosts can communicate, then the Virtual SAN Cluster will form. In the following example, manual disk claiming is chosen so as to provide for learning more about Virtual SAN disk groups during the POC.

5.1 Manual Disk Claiming—Create Disk Groups

At present, there are three hosts in the Virtual SAN Cluster. However, because the cluster is in manual mode, no flash devices or disks have been claimed. A General view of Virtual SAN currently looks something like this.

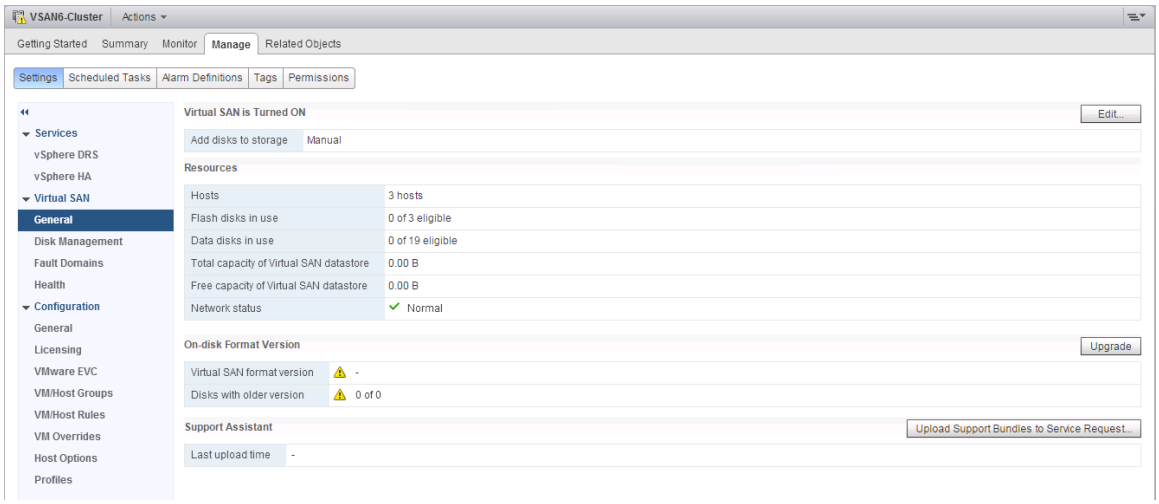


Figure 5.5: Virtual SAN enabled, no disks or flash device claimed

Any warnings against Virtual SAN format version and Disks with older version can be ignored for the moment. These appear as a result of the Health checks, but since there are no disks in the cluster as yet, these warning are displayed.

The next step is to claim some storage and flash devices for Virtual SAN and create the disk groups.

Navigate to Disk Management, just below the General view. You should observe that there are no disk groups associated with the hosts, nor are there any disks in use.

There are a number of icons here related to the claiming of disk groups that require further explaining:

	This allows you to build disks groups across all hosts in one step. Useful for small clusters, but cumbersome when lots of hosts and disks present
	Create a new disk group on a per host basis (visible when disk group selected)

Table 5.1: Disk group icons

For this POC, one flash device and two magnetic disks (HDD) are chosen for the disk group. This is repeated for all three hosts. Of course, you may wish to include additional devices in your POC.

As mentioned, since this is a small cluster, we are only going to create a disk group containing two physical disks for capacity. Either of the icons shown above can be chosen. In this example POC, the first icon shown above can be chosen.

This immediately pops up an option to "Select all eligible disks". We are not choosing this option in the POC, but it is a useful option to be aware of. If you are including all disks in all hosts, then you may certainly choose this option to speed things along.

Similarly, if one clicks on the check box next to a hostname, all disks belonging to that host will be used for creating disk groups. This is also not a feature we wish to use in the POC either, but once again a useful option to be aware of. If you wish to select all disks on a particular host for your POC, you may choose this option.

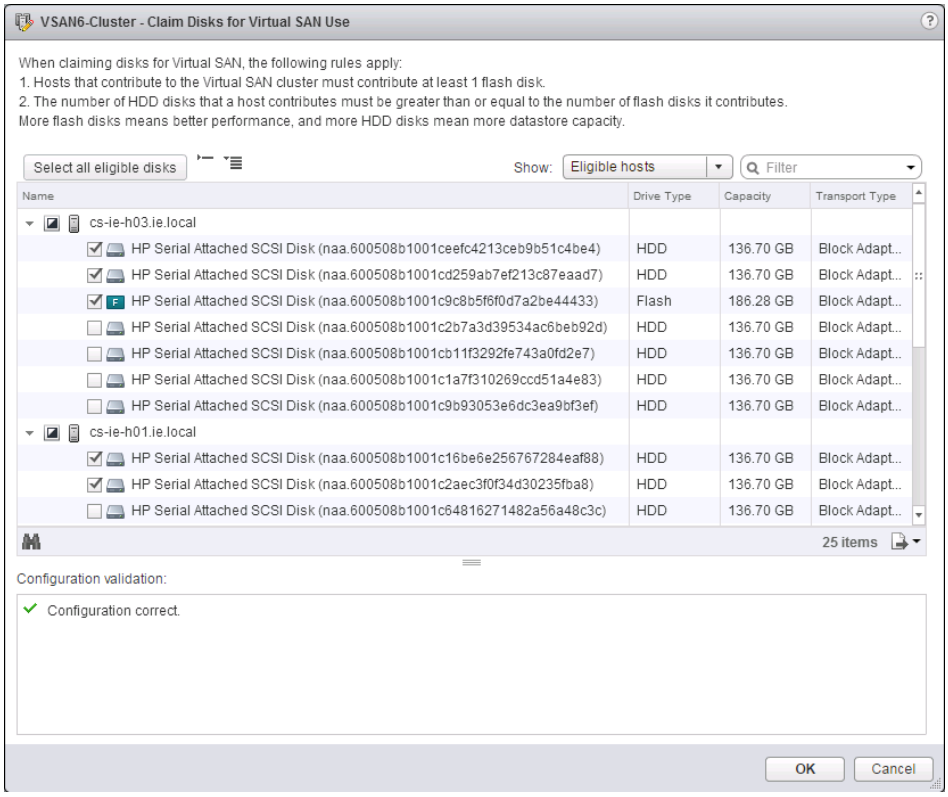


Figure 5.6: Claiming disks for disk groups

Note that there will be warnings posted if a flash device and magnetic disk devices are not chosen, since a disk group requires one flash device and at least one magnetic disk in hybrid configurations, which is what we are working on here. Click on the OK button to complete the configuration.

Once the configuration task completes, the Disk Management view should now be updated with a disk group per host added, as well as the "Disks in Use" column populated with the number of disks in use in the disk groups, which should be three (one flash device and two magnetic disks). The view should look similar to the following, although the total number of disks will vary depending on your POC.

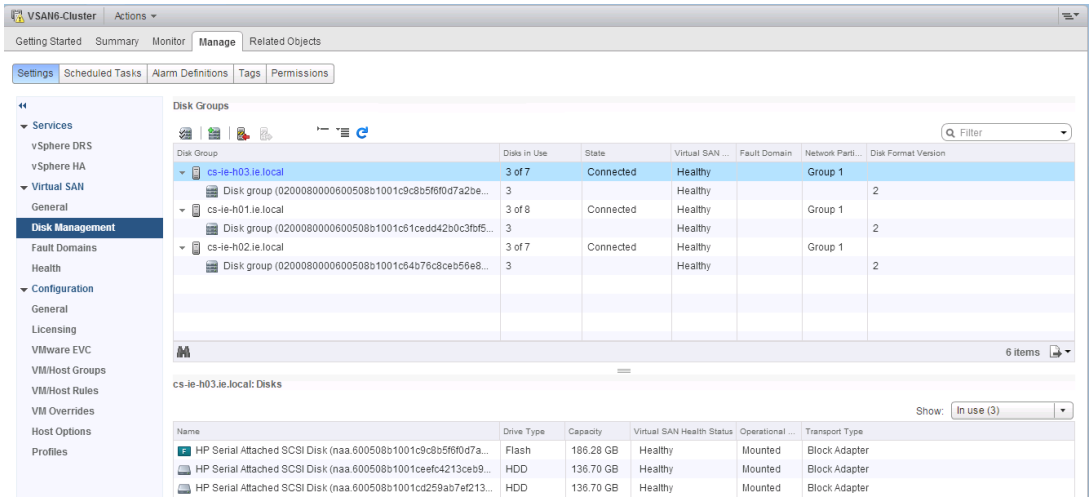


Figure 5.7: Disk groups created

Returning to the General view (and possibly refreshing the screen) should now show the number of flash disks in use (three, one per host) and data disks (six, two per host) that are now in use. It should also show the total capacity of the Virtual SAN datastore, which in this case is ~812GB. That is 6 x 136GB, less some overhead. Remember that flash devices do not contribute towards capacity, only the magnetic disk devices (in the case of hybrid configurations).

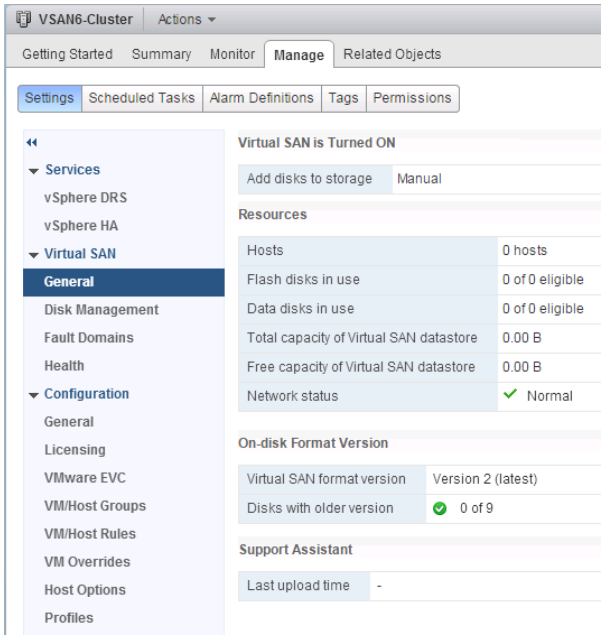


Figure 5.8: On-disk Format Version

6. Enable the Virtual SAN Health Check Plugin

Following on from the Virtual SAN 6.0 GA release, a new feature called Health Check plugin was released. This gives administrators valuable information regarding the state of the Virtual SAN Cluster, and is also extremely useful for POC activities as it quickly discovers issues.

There is an in-depth description of health checks, including how to install and configure it, as well as detailed information on the various checks that it carries out. Refer to the VMware Virtual SAN Health Check Plugin Guide.

Starting with vSphere 6.0 update 1 and vCenter 6.0 update 1, the Health Check plugin is pre-installed both in vCenter and as a VIB on each ESXi host. All that's required is to enable the health check services once Virtual SAN is enabled. This is done on a cluster-by-cluster basis at the cluster's Manage tab > Settings > Virtual SAN > Health. Once enabled, the new health check service in Virtual SAN 6.1 runs hourly by default and on-demand when you visit the cluster's Monitor tab > Virtual SAN > Health.

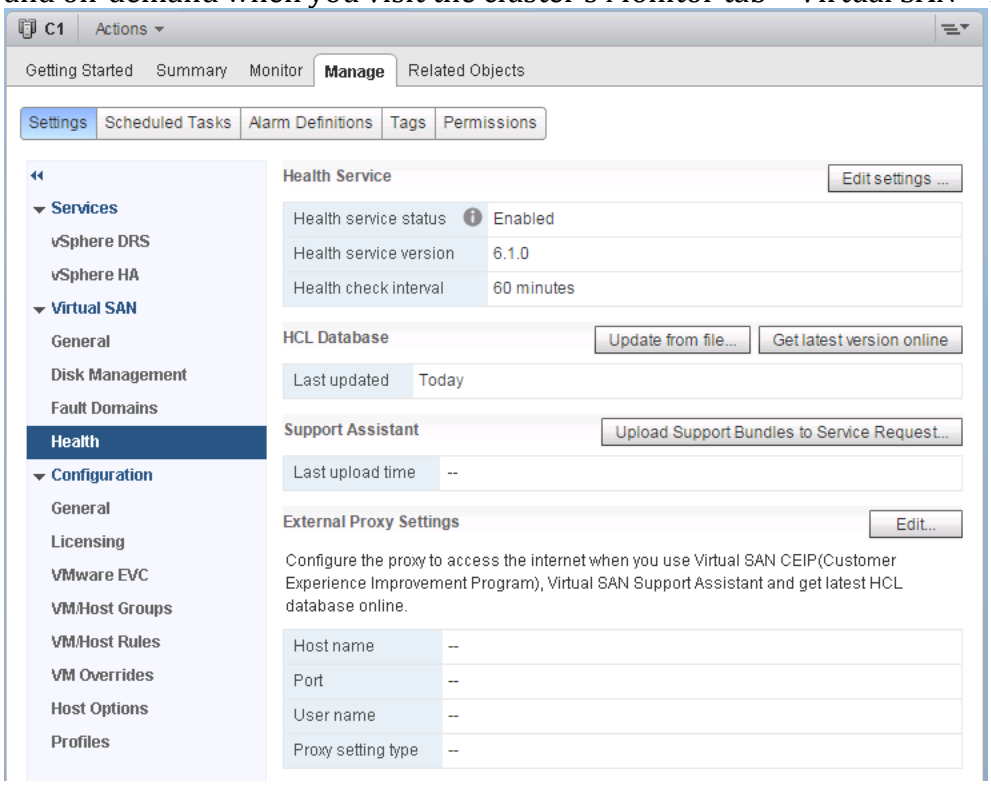


Figure 6.1: Managing Virtual SAN health check service

With the Virtual SAN -enabled cluster object selected in the inventory, navigate to the Monitor tab > Virtual SAN > Health. This will display the list of health check, and their status. Hopefully everything will show up as passed as per figure 6.2 below.

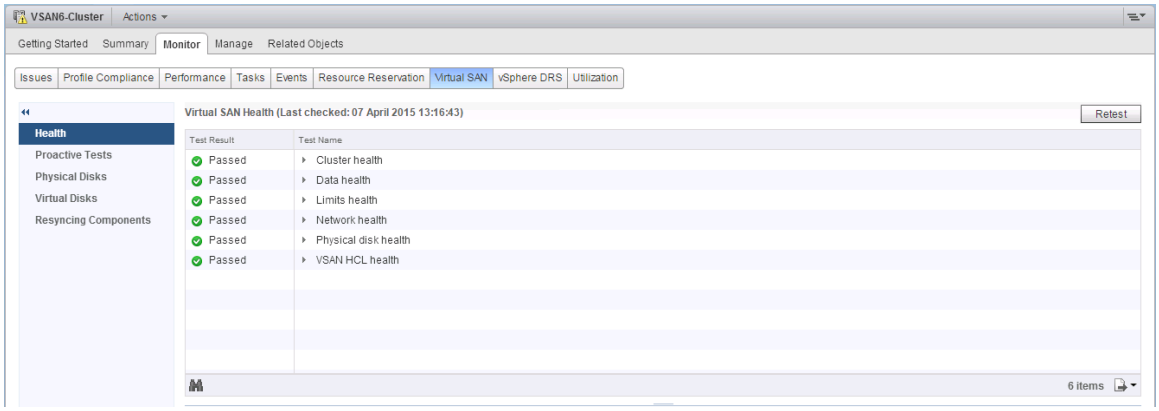


Figure 6.2: Top level list of health checks

6.1 Check Your Network Thoroughly

Once the Virtual SAN network has been created, and Virtual SAN is enabled, you should check that each ESXi host in the Virtual SAN Cluster is able communicate to all other ESXi hosts in the cluster. The easiest way to achieve this is via the Health Check Plugin.

6.1.1 Why Is This Important?

Virtual SAN is entirely dependent on the network: its configuration, reliability, performance, etc. One of the most frequent causes of requesting support is either an incorrect network configuration, or the network not performing as expected.

6.1.2 Check the Network Partition Groups after Creating Cluster

A network partition is when a subset of hosts (one or more) in unable to communicate to another subset of hosts. The Disk Management view (found under Virtual SAN Cluster > Manage tab > Settings) provides immediate information about whether or not there is a network partition in your cluster. If the network is functioning properly, all hosts will be in Group 1. Only if multicast routing is properly configured would Virtual SAN still function with multiple partition groups. Refer to the Network health tests under Cluster > Monitor > Virtual SAN > Health.

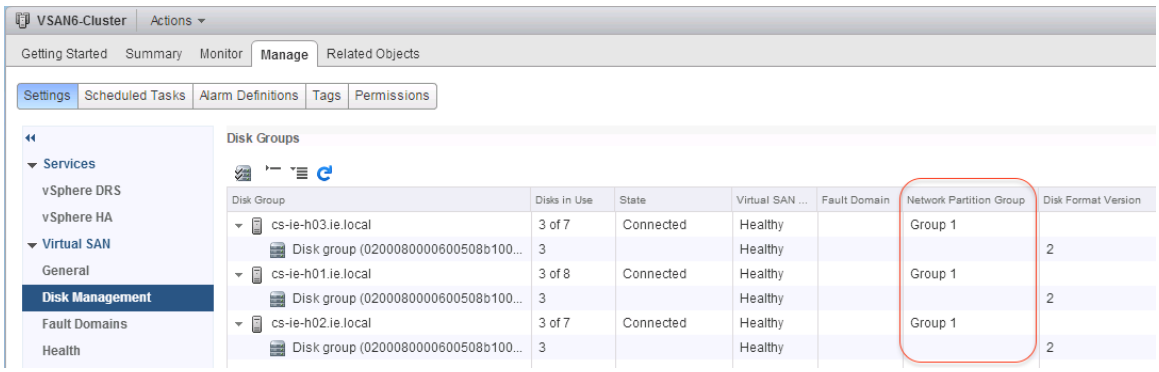


Figure 6.3: Network Partition Group info

6.1.3 Use the Health Check Plugin to Verify Virtual SAN Functionality

Running individual commands from one host to all other hosts in the cluster can be tedious and time consuming. Fortunately, since Virtual SAN 6.0 supports a new health check plugin, part of which tests the network connectivity between all hosts in the cluster. If for some reason the cluster will not form, and displays a "Network misconfiguration" in the General view, you should proceed with enabling the health check plugin, outlined in the previous section. This will reduce the time to detect and resolve the networking issue, or any other Virtual SAN misconfiguration issues in the cluster.

In the screenshot below, one can see that each of the health checks for networking has successfully passed.

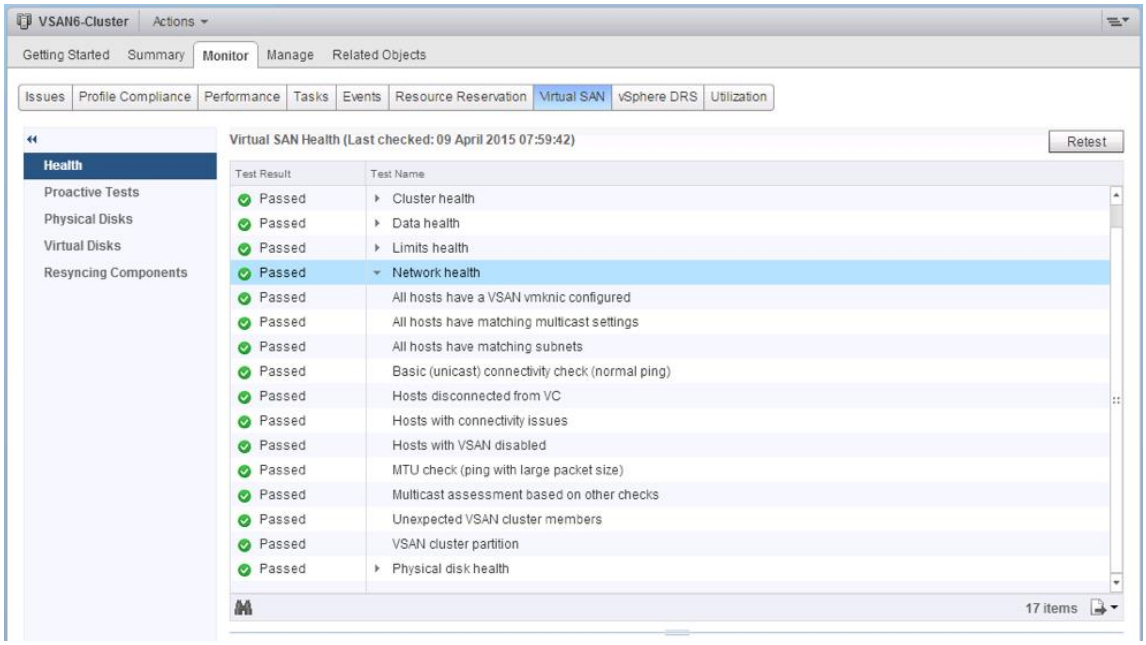


Figure 6.4: Network health checks all passed

If any of the network health checks fail, select the appropriate check and examine the details screen below for details on how to resolve the issue. Each details view also contains an *AskVMware* button where appropriate, which will take you to a VMware Knowledge Base article detailing the issue, and how to troubleshoot and resolve it.

For example, in this case where one host does not have a Virtual SAN vmknic configured, this is what is displayed.

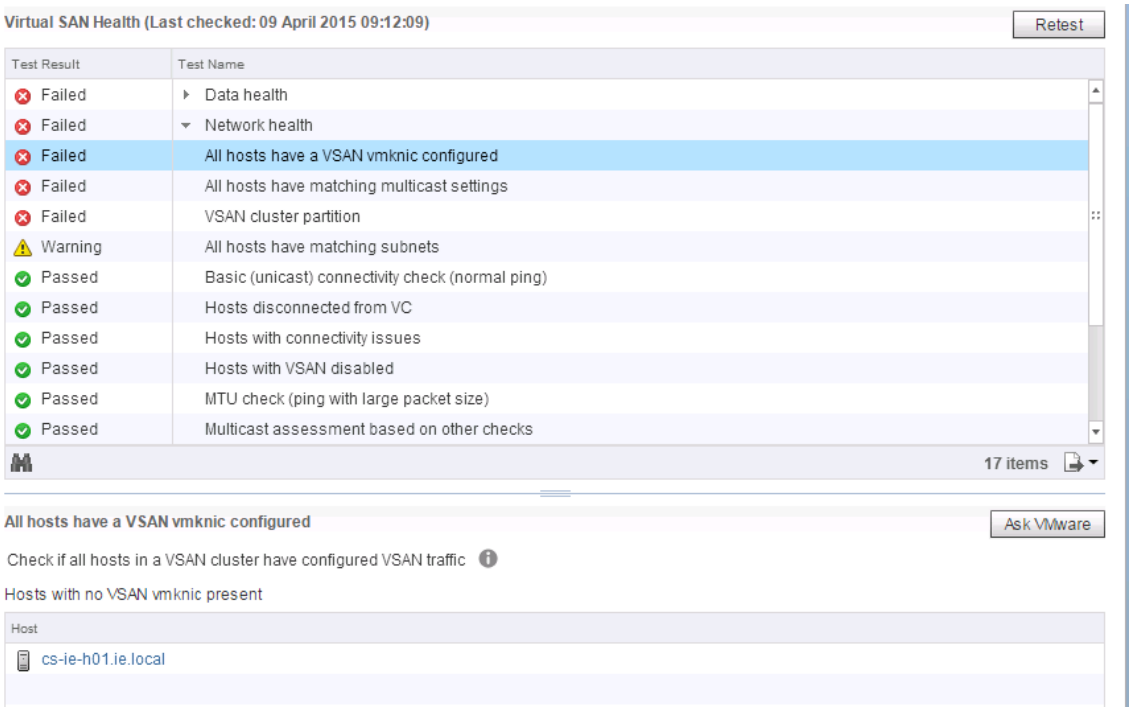


Figure 6.5: Network health failure example

Before going any further with this POC, it is worth downloading the latest version of the HCL database and running a "Retest" on the Health check screen. This will ensure everything in the cluster is optimal. It will also check the hardware against the VMware Compatibility Guide (VCG) for Virtual SAN, verify that the networking is functional, and that there are no underlying disk problems. All going well, after the Retest, everything should still display a "Passed" status.

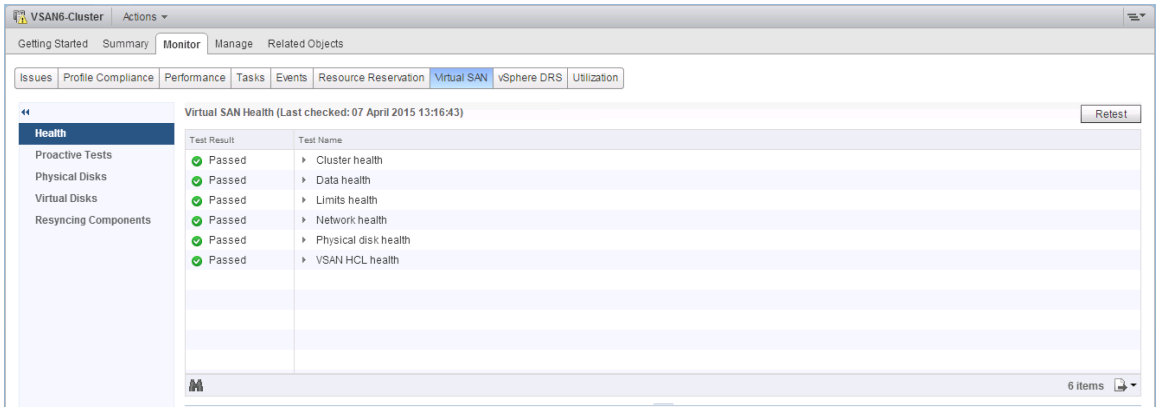


Figure 6.6: Virtual SAN Health checks

In particular at this juncture the Cluster health, Limits health and Physical disk health should be examined. The data health only becomes relevant once you start to deploy virtual machines to the Virtual SAN datastore.

Figure 6.7: Expanded Health check plugin Checks

6.1.4 Use the Troubleshooting Reference Manual to Verify Network Functionality

If you need to delve deeper into troubleshooting the network, there are a number of commands available for testing network connectivity between Virtual SAN hosts. These include *vmkping* and *tcpdump-uw*. How to use these commands, and the different parts of Virtual SAN functionality that they test, are outlined in the VMware Virtual SAN Diagnostics and Troubleshooting Reference Manual.

Virtual SAN, including the Health check monitoring, is now successfully deployed. The remainder of this POC guide will involve various tests and error injections to show how Virtual SAN will behave under these circumstances.

7. vSphere Functionality on Virtual SAN

This initial test is per VM testing, and will highlight the fact that general virtual machine operations are unchanged in Virtual SAN environments.

7.1 Deploy Your First VM

In this section, a VM is deployed to the Virtual SAN datastore using the default storage policy. This default policy is preconfigured and does not require any intervention unless you wish to change the default settings, which we do not recommend.

To examine the default policy settings, navigate to Home > VM Storage Policies.

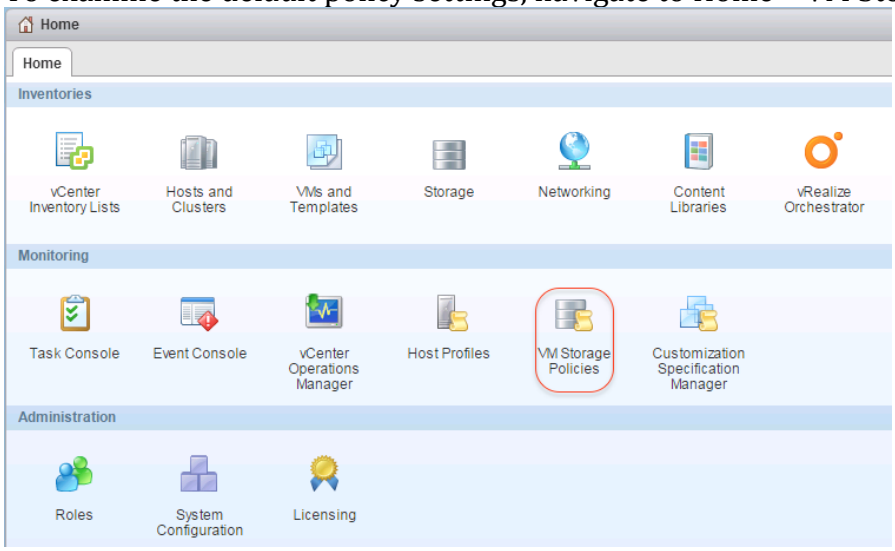


Figure 7.1: VM Storage Policies

From there, select Virtual SAN Default Storage Policy and then select the Manage tab. Under the Manage tab, select **Rule-Set 1: Virtual SAN** to see the settings on the policy:

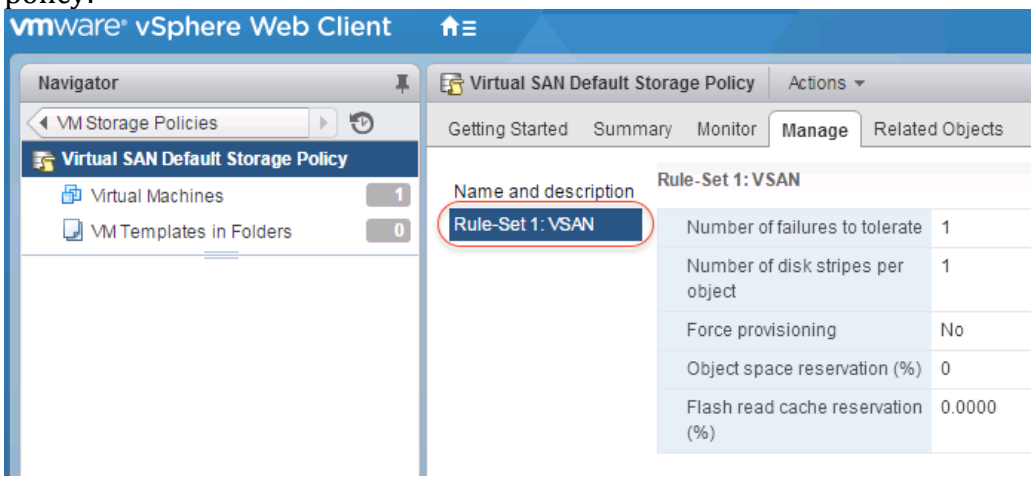
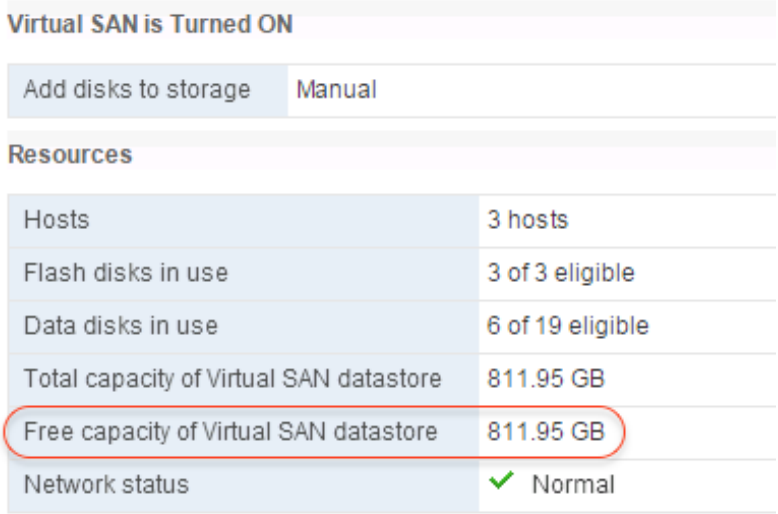


Figure 7.2: Rule-Set 1: Virtual SAN (default policy)

We will return to VM Storage Policies in more detail in a future chapter, but suffice to say that when a VM is deployed with the default policy, it should have a mirror copy of the VM data created. This second copy of the VM data is placed on storage on a different host to enable the VM to tolerate any single failure. Also note that object space reservation is set to 0%, meaning the object should be deployed as "thin". After we have deployed the VM, we will verify that Virtual SAN adheres to both of these capabilities.

One final item to check before we deploy the VM is the current free capacity on the Virtual SAN datastore. This can be viewed from the Virtual SAN Cluster > Manage tab > Settings > General view. In this POC, it is 811.95 GB.



Virtual SAN is Turned ON

Add disks to storage	Manual

Resources

Hosts	3 hosts
Flash disks in use	3 of 3 eligible
Data disks in use	6 of 19 eligible
Total capacity of Virtual SAN datastore	811.95 GB
Free capacity of Virtual SAN datastore	811.95 GB
Network status	✔ Normal

Figure 7.3: Current free capacity of Virtual SAN datastore

Make a note of the free capacity on your POC before continuing with the deploy VM exercise.

To deploy the VM, simply follow the steps provided in the wizard.

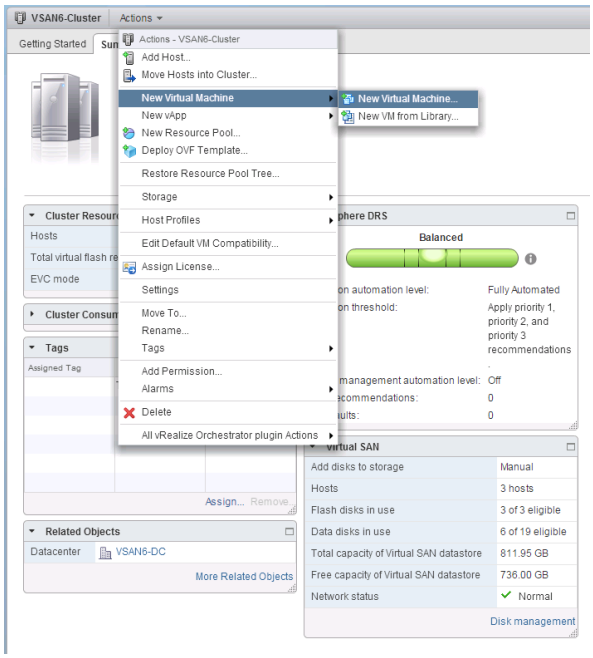


Figure 7.4: New Virtual Machine

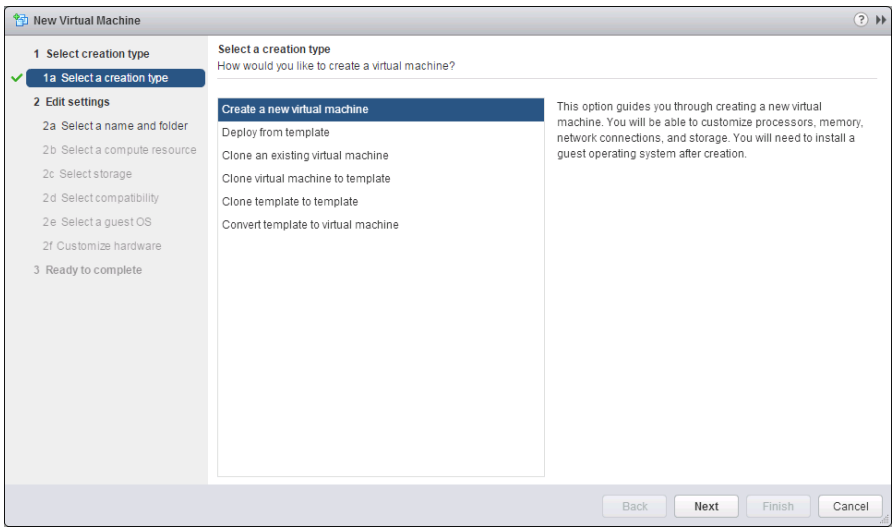


Figure 7.5: Create a new virtual machine

At this point a name for the VM must be provided, and then the Virtual SAN Cluster must be selected as a compute resource.

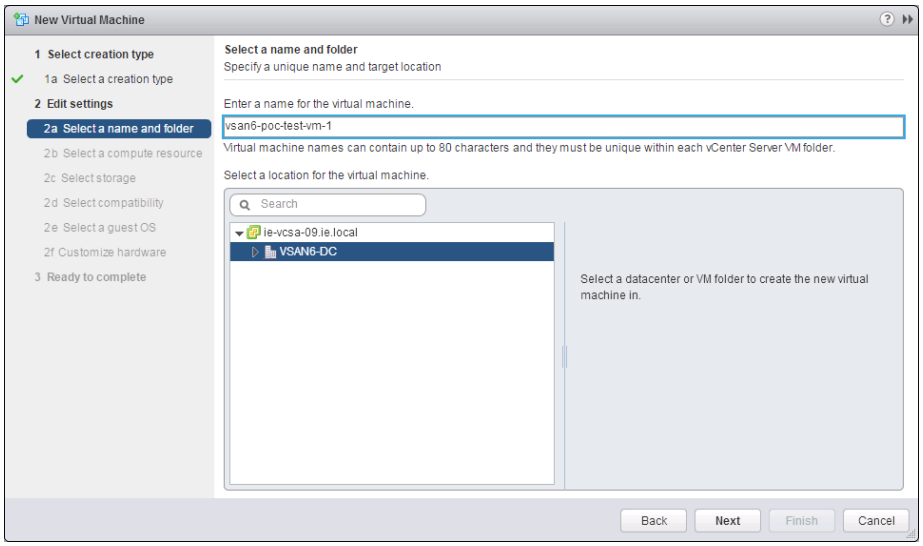


Figure 7.6: Select a name and folder

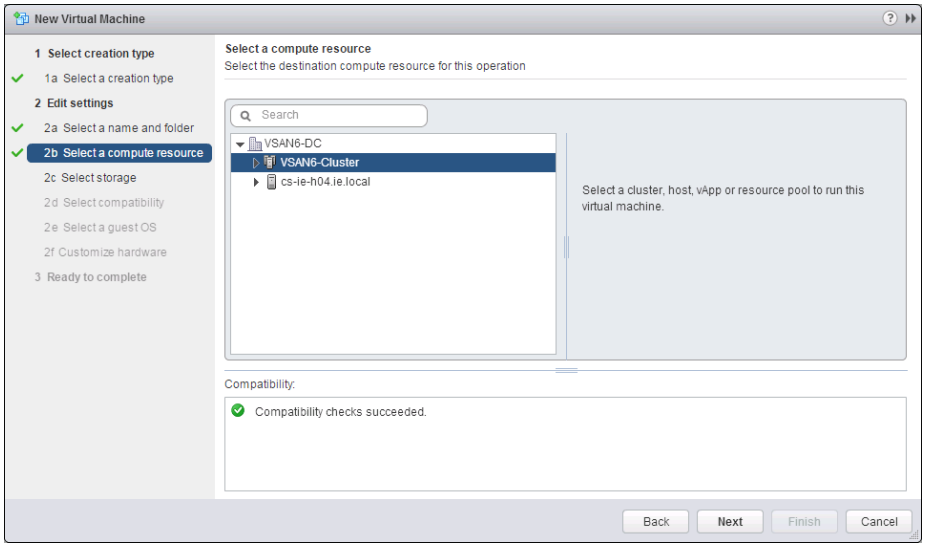


Figure 7.7: Select a compute resource

At this point, the virtual machine deployment process is almost identical to all other virtual machine deployments that you have done on other storage types. It is the next section that might be new to you. This is where a policy for the virtual machine is chosen.

From the next menu, you can either select the Virtual SAN datastore, and the "Datastore Default" policy will actually point to the "Virtual SAN Default Storage Policy" seen earlier.

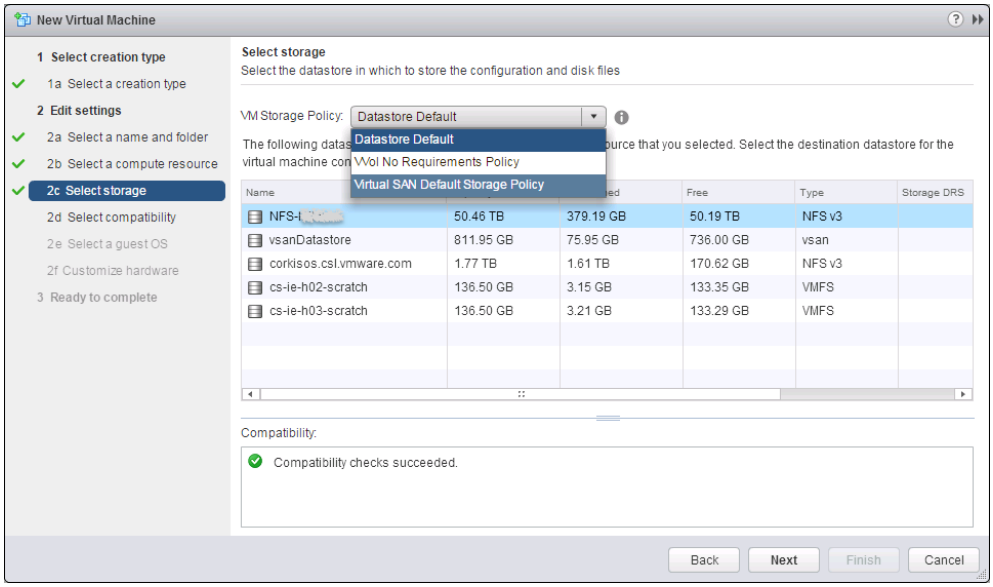


Figure 7.8: Select the Virtual SAN Default Storage Policy

Once the policy has been chosen, datastores are split into those that are compliant with the policy, and those that are non-compliant with the policy. As seen below, only the Virtual SAN datastore can understand the policy settings in the Virtual SAN Default Storage Policy so it is the only one that shows up as **Compatible** in the list of datastores.

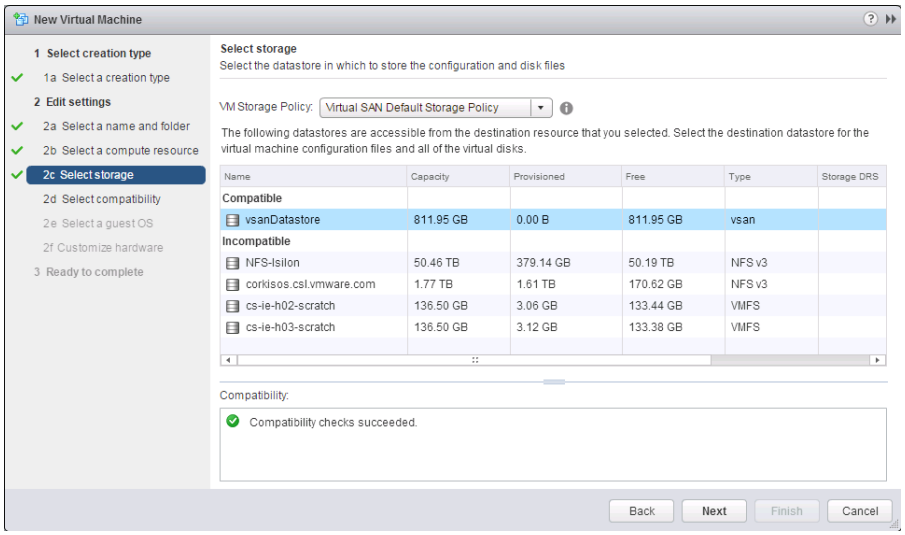


Figure 7.9: vsanDatastore is compatible with Virtual SAN Default Storage Policy

The rest of the VM deployment steps in the wizard are quite straightforward, and simply entail selecting ESXi version compatibility (leave at default), a guest OS (leave at default) and customize hardware (no changes). Essentially you can click through the remaining wizard screens without making any changes.

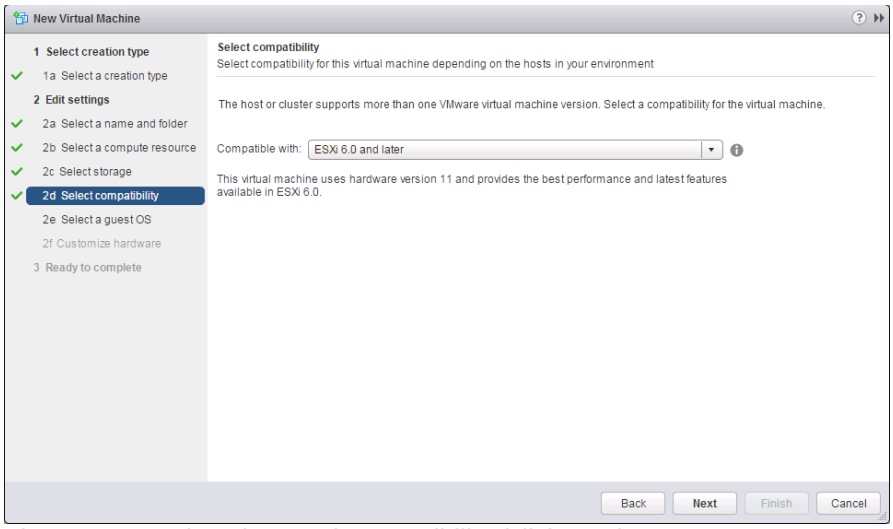


Figure 7.10: Select the ESXi compatibility (click next)

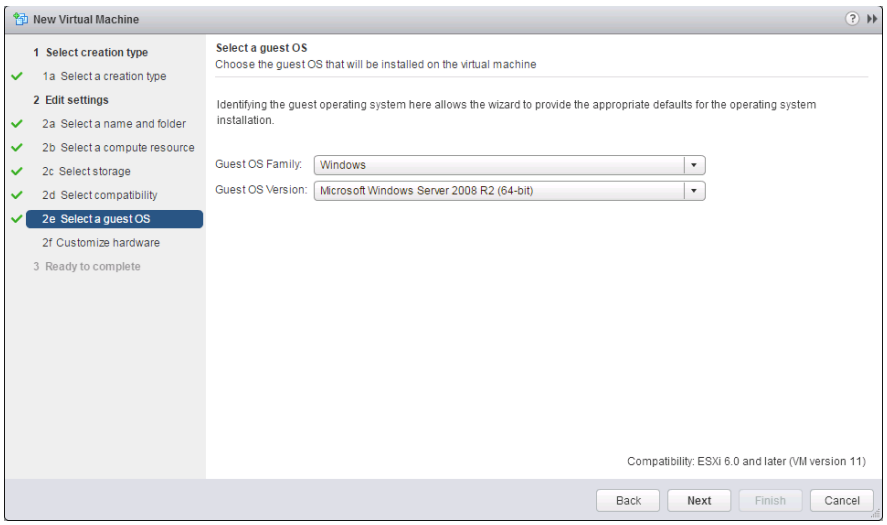


Figure 7.11: Select the guest OS (click next)

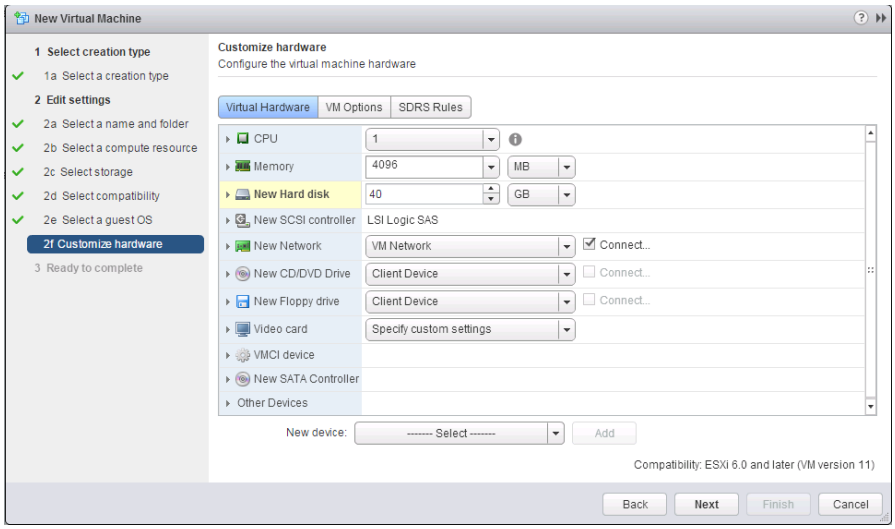


Figure 7.12: Customize hardware (click next)

The final step in the wizard is to click the "Finish" button to initiate the creation of the VM.

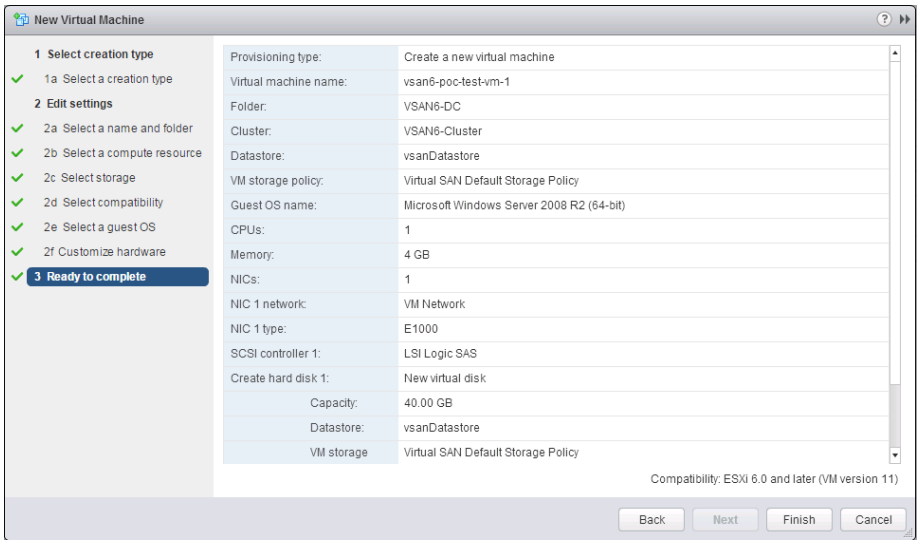


Figure 7.13: Finish VM creation

Once the VM is created, select the new VM in the inventory, navigate to the Manage tab, and then select "Policies". There should be two objects shown, "VM home" and "Hard disk 1". Both of these should show a compliance status of "Compliant" meaning that Virtual SAN was able to deploy these objects in accordance to the policy settings.

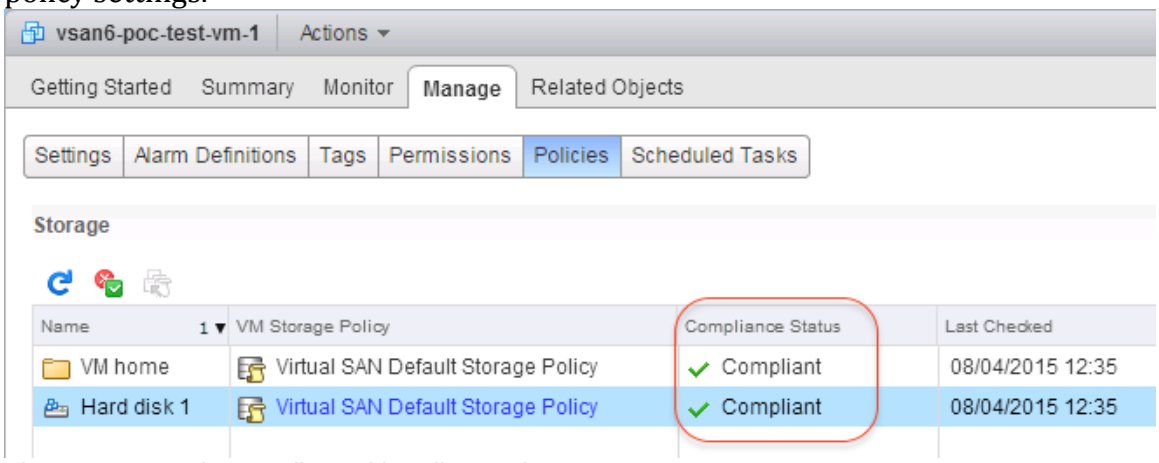


Figure 7.14: VM is compliant with policy settings

To verify this, navigate to the Monitor tab, and then select "Policies". Once again, both the "VM home" and "Hard disk 1" should be displayed. Select "Hard disk 1" and further down the window, select the "Physical Disk Placement" tab. This should display a RAID 1 configuration with two components, each component representing a mirrored copy of the virtual disk. It should also be noted that different components are located on different hosts. This implies that the policy setting to tolerate 1 failure is being adhered to.

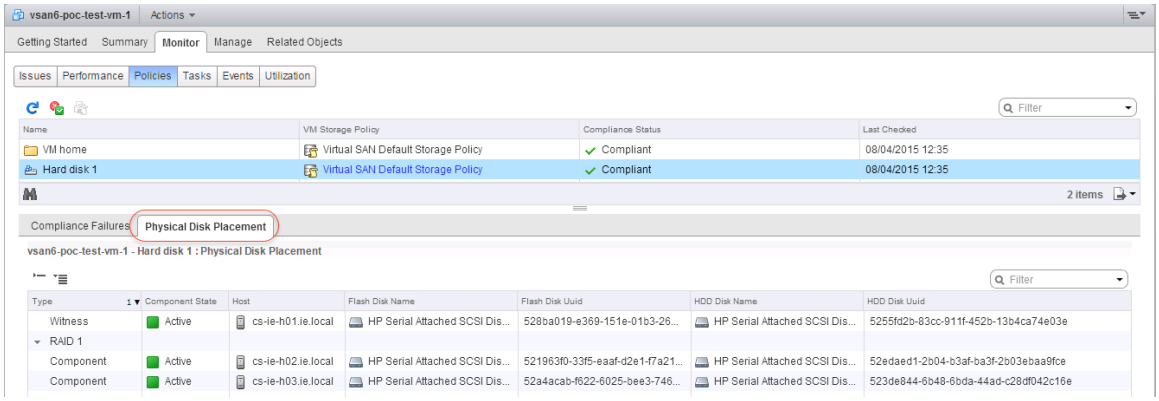


Figure 7.15: Physical Disk Placement displays underlying layout of objects

The witness item shown above is used to maintain a quorum. For more information on the purpose of witnesses, and objects and components in general, refer to the [VMware Virtual SAN 6.0 Design and Sizing Guide](#).

One final item is related to the "object space reservation" policy setting that defines how much space a VM reserves on the Virtual SAN datastore. By default, it is set to 0%, implying that the VM's storage objects are entirely "thin" and consume no unnecessary space.

If we examine Figure 7.12, we see that we requested that the VM be deployed with 40GB of disk space. However if we look at the free capacity after the VM has been deployed (as shown in figure 7.16 below), we see that the free capacity is very close to what it was before the VM was deployed, as previously captured in figure 7.3.

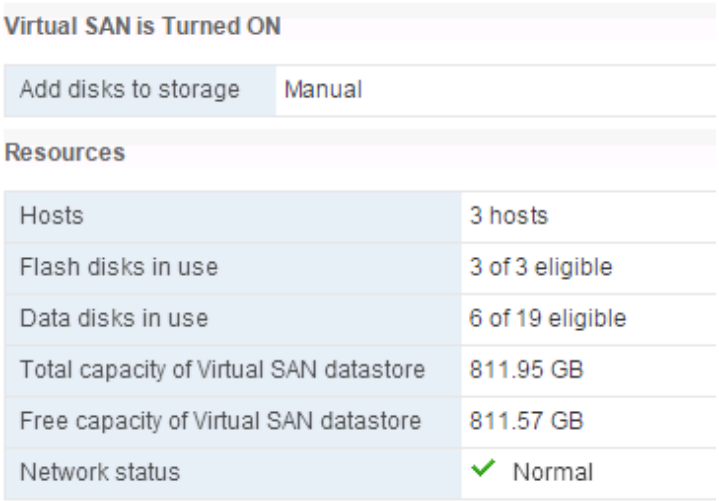


Figure 7.16: Free capacity after VM is created

Of course we have not installed anything in the VM such as a guest OS, but it shows that only a tiny portion of the Virtual SAN datastore has so far been used, verifying

that the object space reservation setting of 0% (essentially thin provisioning) is working correctly.

Do not delete this VM as we will use it for other POC tests going forward.

7.2 Snapshot VM

Using the virtual machine created previously, take a snapshot of it. The snapshot can be taken when the VM is powered on or powered off. The objective here is to see a successful snapshot delta object created, and see that the policy settings of the delta object are inherited directly from the base disk object.

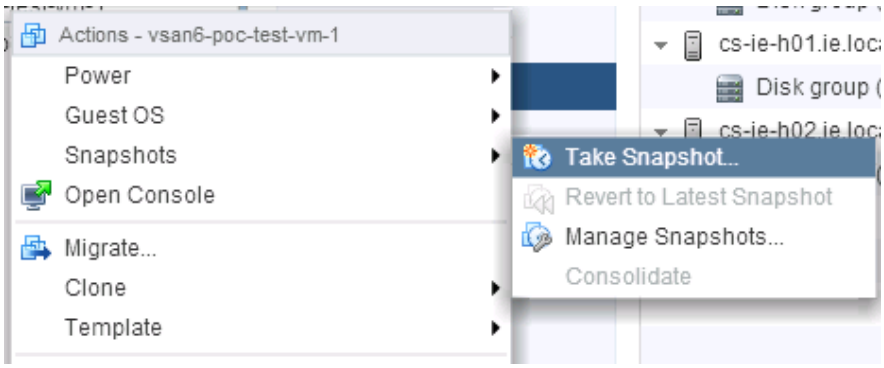


Figure 7.17: Take a VM snapshot

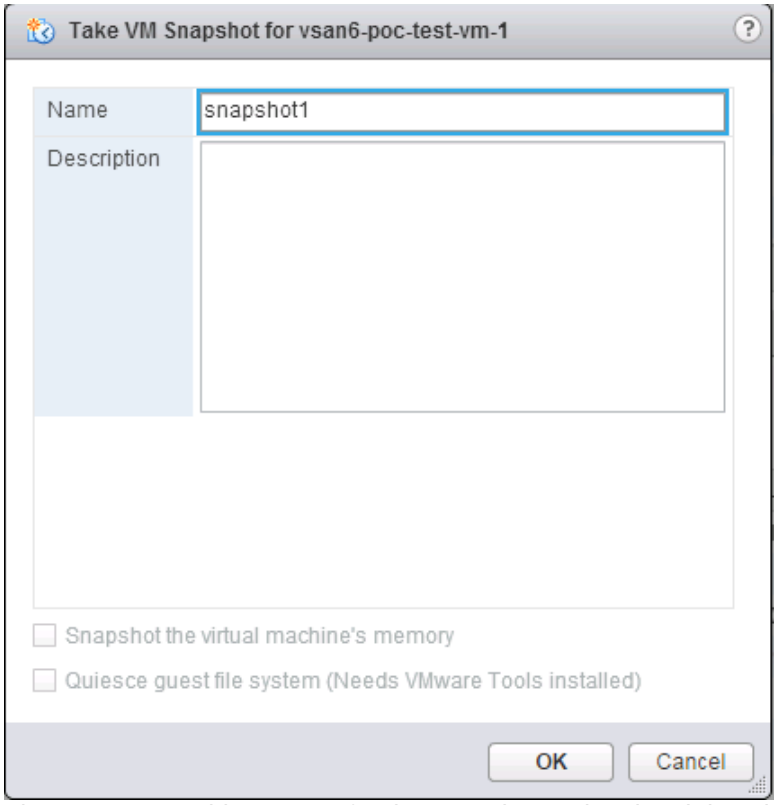


Figure 7.18: Provide a name for the snapshot and optional description

Once the snapshot has been requested, monitor tasks and events to ensure that it has been successfully captured. Once the snapshot creation has completed, additional actions will become available in the snapshot dropdown window. For example there

is a new action to "Revert to Latest Snapshot" and another action to "Manage Snapshots…".

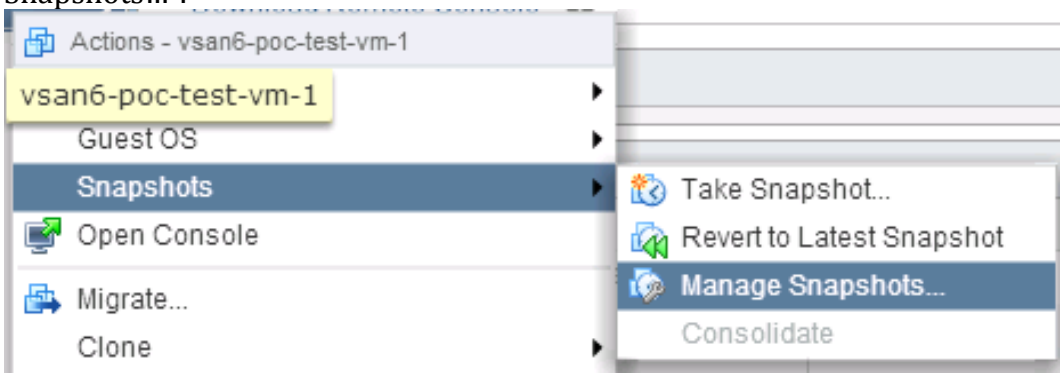


Figure 7.19: New snapshot actions

If the "Manage Snapshots…" option is chosen, the following is displayed. It includes details regarding all snapshots in the chain, the ability to delete one or all of them, as well as the ability to revert to a particular snapshot.

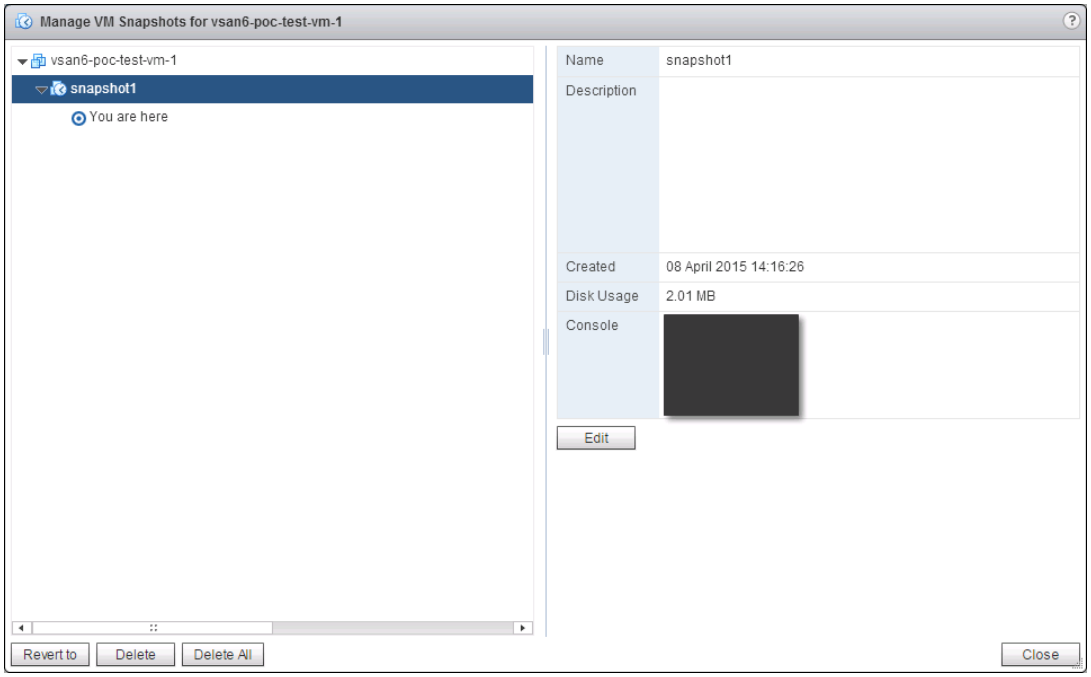


Figure 7.20: Manage Snapshots

There is unfortunately no way to see snapshot delta object information from the UI, like we can do for VMDKs and for VM home. Instead, the Ruby vSphere Console (RVC) must be relied on. To get familiar with RVC, see [VMware Ruby vSphere Console Command Reference for Virtual SAN].

The command needed to display snapshot information is:

```
vsan.vm_object_info <VM>
```

Here is an output based on the snapshot created previously:

```
 /ie-vcsa-09.ie.local/VSAN6-DC/vms> vsan.vm_object_info 1
 VM VSAN6-poc-test-vm-1:
   Namespace directory
     DOM Object: 95122555-8061-3328-cf10-001f29595f9f (v2, owner: cs-ie-h01.ie.local,
policy: forceProvisioning = 0, hostFailuresToTolerate = 1, spbmProfileId = aa6d5a82-1c88-
45da-85d3-3d74b91a5bad, proportionalCapacity = [0, 100], spbmProfileGenerationNumber = 0,
cacheReservation = 0, stripeWidth = 1)
       RAID_1
         Component: 96122555-80ad-3c97-dadf-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h01.ie.local, md: 52fc637f-ecf9-2b53-ff31-9e8d75d2b43f, ssd: 528ba019-e369-151e-01b3-
26b103d7de0f,
                                                          votes: 1, usage: 0.3 GB)
         Component: 96122555-dc90-3e97-9c6f-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h02.ie.local, md: 52edaed1-2b04-b3af-ba3f-2b03ebaa9fce, ssd: 521963f0-33f5-eaaf-d2e1-
f7a218b13be4,
                                                          votes: 1, usage: 0.3 GB)
       Witness: 96122555-fc7b-3f97-5d9a-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h03.ie.local, md: 527aade4-cec7-0661-b621-6e22d69c3042, ssd: 52a4acab-f622-6025-bee3-
746d436627cf,
                                                        votes: 1, usage: 0.0 GB)
   Disk backing: [vsanDatastore] 95122555-8061-3328-cf10-001f29595f9f/VSAN6-poc-test-vm-
1-000001.vmdk
     DOM Object: 2a2a2555-946f-292b-2e23-001f29595f9f (v2, owner: cs-ie-h01.ie.local,
policy: spbmProfileGenerationNumber = 0, forceProvisioning = 0, cacheReservation = 0,
hostFailuresToTolerate = 1, stripeWidth = 1, spbmProfileId = aa6d5a82-1c88-45da-85d3-
3d74b91a5bad, proportionalCapacity = [0, 100], objectVersion = 2)
       RAID_1
         Component: 2a2a2555-8ce3-a171-fb8e-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h01.ie.local, md: 5255fd2b-83cc-911f-452b-13b4ca74e03e, ssd: 528ba019-e369-151e-01b3-
26b103d7de0f,
                                                          votes: 1, usage: 0.0 GB)
         Component: 2a2a2555-78d0-a371-b90d-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h02.ie.local, md: 52edaed1-2b04-b3af-ba3f-2b03ebaa9fce, ssd: 521963f0-33f5-eaaf-d2e1-
f7a218b13be4,
                                                          votes: 1, usage: 0.0 GB)
       Witness: 2a2a2555-ce29-a571-da2b-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h03.ie.local, md: 527aade4-cec7-0661-b621-6e22d69c3042, ssd: 52a4acab-f622-6025-bee3-
746d436627cf,
                                                        votes: 1, usage: 0.0 GB)
     Disk backing: [vsanDatastore] 95122555-8061-3328-cf10-001f29595f9f/VSAN6-poc-test-
vm-1.vmdk
       DOM Object: 97122555-78d5-5580-bffc-001f29595f9f (v2, owner: cs-ie-h03.ie.local,
policy: forceProvisioning = 0, hostFailuresToTolerate = 1, spbmProfileId = aa6d5a82-1c88-
45da-85d3-3d74b91a5bad, proportionalCapacity = 0, spbmProfileGenerationNumber = 0,
cacheReservation = 0, stripeWidth = 1)
         RAID_1
           Component: 98122555-3ec9-d1d6-01f4-001f29595f9f (state: ACTIVE (5), host: cs-
ie-h02.ie.local, md: 52edaed1-2b04-b3af-ba3f-2b03ebaa9fce, ssd: 521963f0-33f5-eaaf-d2e1-
f7a218b13be4,
                                                            votes: 1, usage: 0.0 GB)
           Component: 98122555-5c6f-d3d6-55a7-001f29595f9f (state: ACTIVE (5), host: cs-
ie-h03.ie.local, md: 523de844-6b48-6bda-44ad-c28df042c16e, ssd: 52a4acab-f622-6025-bee3-
746d436627cf,
                                                            votes: 1, usage: 0.0 GB)
         Witness: 98122555-2028-d4d6-6ee6-001f29595f9f (state: ACTIVE (5), host: cs-ie-
h01.ie.local, md: 5255fd2b-83cc-911f-452b-13b4ca74e03e, ssd: 528ba019-e369-151e-01b3-
26b103d7de0f,
                                                          votes: 1, usage: 0.0 GB)
 /ie-vcsa-09.ie.local/VSAN6-DC/vms>
```

The three objects that are now associated with that virtual machine have a bold font in this document for clarity. There is the namespace directory (VM home), there is the disk *VSAN6-poc-test-vm-1.vmdk* and there is the snapshot delta `VSAN6-poc-test-vm-1-000001.vmdk`. The snapshot delta has been highlighted in blue above.

If you look closely, both of the disk backings have the same policy settings since every snapshot inherits its policy settings from the base disk. Both have a *stripeWidth* of 1, and *hostFailuresToTolerate* of 1 and an Object Space Reservation (shown as *proportionalCapacity* here) of 0%.

The snapshot can now be deleted from the VM. Monitor the VM's tasks and ensure that it deletes successfully. When complete, snapshot management should look similar to this.

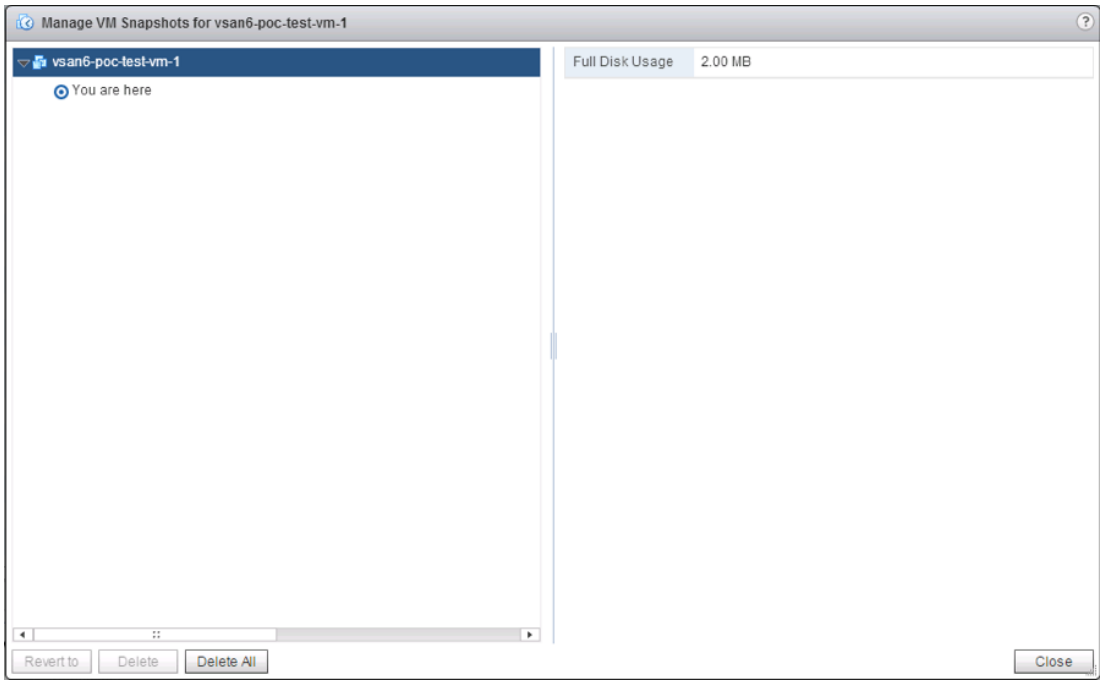


Figure 7.21: Manage Snapshots… Snapshot deleted

This completes the snapshot section of this POC. Snapshots in a Virtual SAN datastore are very intuitive because they utilize vSphere native snapshot capabilities. Starting with Virtual SAN 6.0, they are stored efficiently using "vsansparse" technology that improves the performance of snapshots compared to Virtual SAN 5.5. In Virtual SAN 6.1, snapshot chains can be up to 16 snapshots deep.

7.3 Clone a VM

The next POC test is cloning a VM. We will continue to use the same VM as before. This time make sure the VM is powered on first. There are a number of different cloning operations available in vSphere 6. These are shown here.

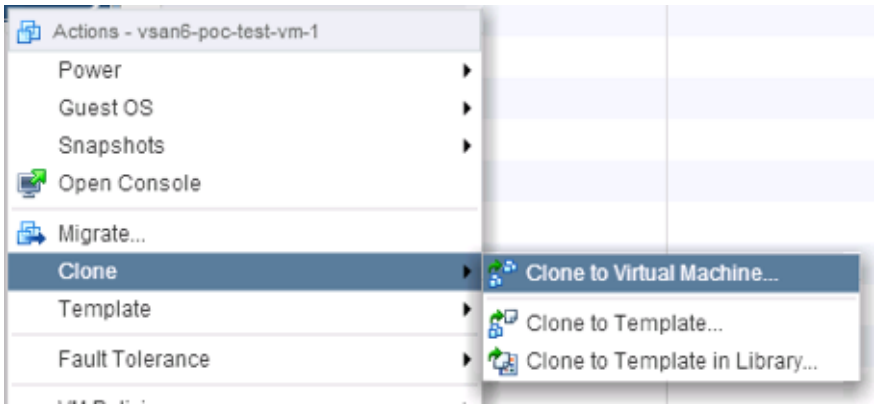


Figure 7.22: Clone operations

The one that we shall be running as part of this POC is the "Clone to Virtual Machine". The cloning operation is very much a "click, click, next" type activity. This next screen is the only one that requires human interaction. One simply provides the name for the newly cloned VM, and a folder if desired.

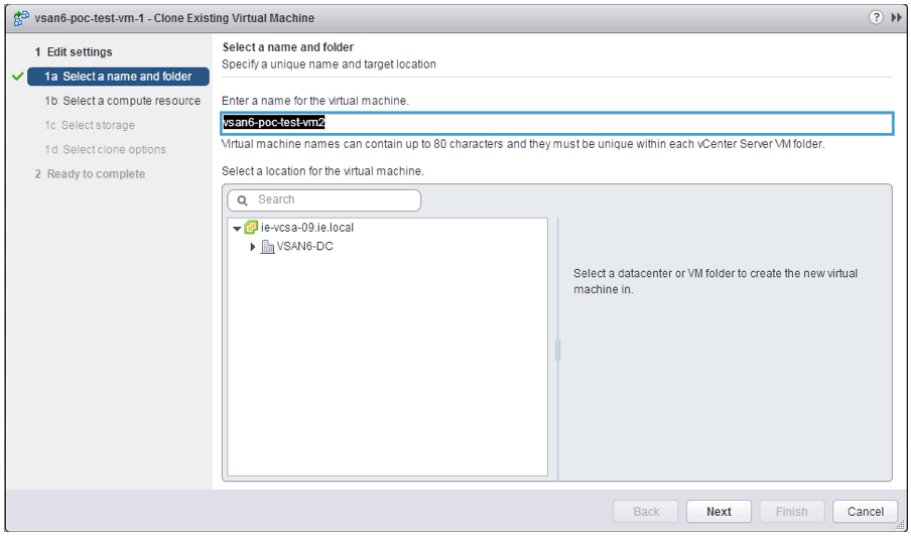


Figure 7.23: Select a name and folder

We are going to clone the VM in the Virtual SAN Cluster, so this must be selected as the compute resource.

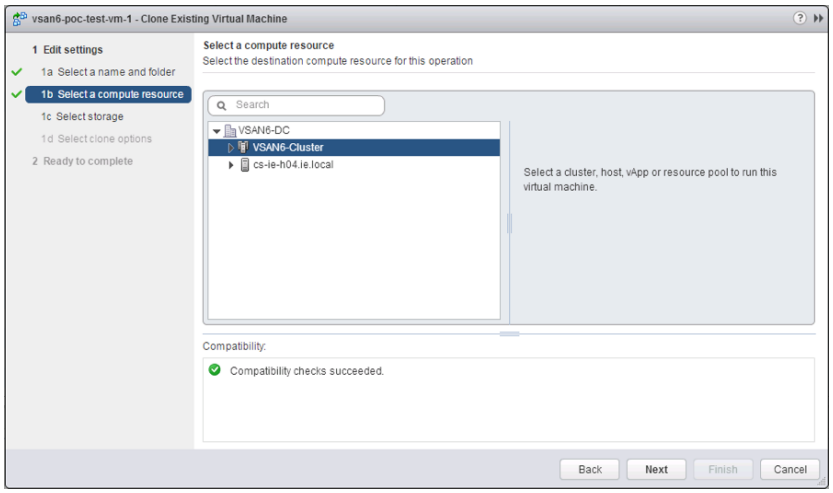


Figure 7.24: Select a compute resource

The storage will be the same as the source VM, namely the vsanDatastore. This will all be pre-selected for you if the VM being cloned also resides on the vsanDatastore.

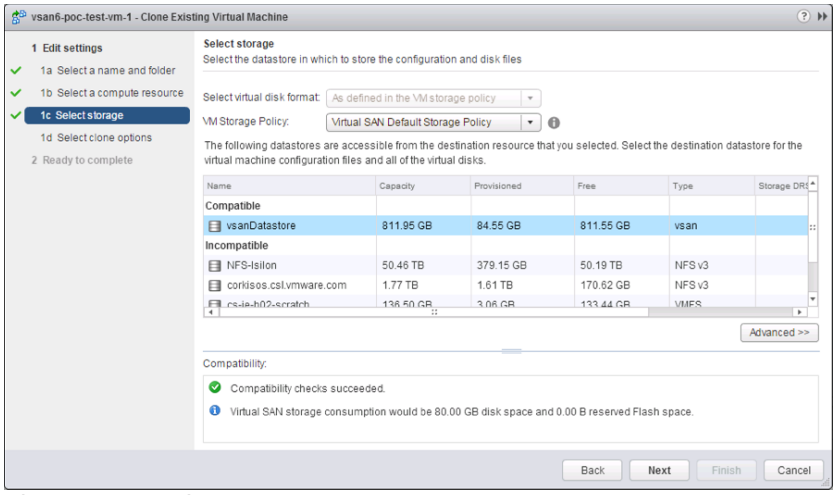


Figure 7.25: Select storage

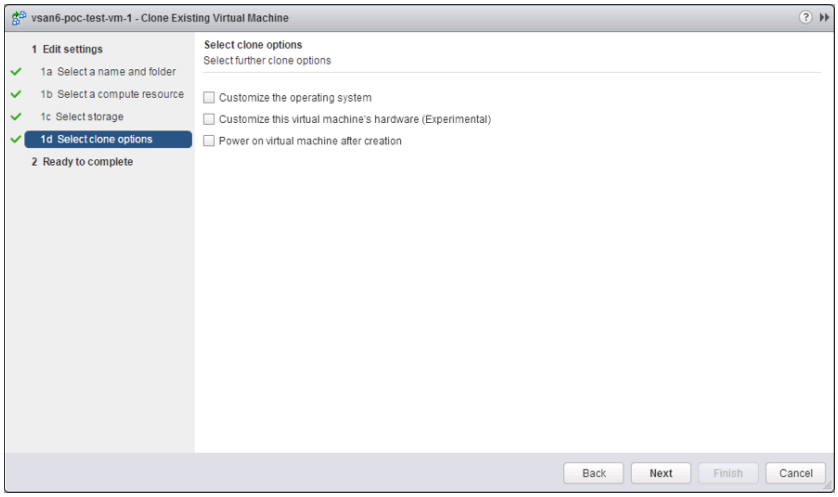


Figure 7.26: Select options (leave unchecked - default)

This will take you to the “Ready to Complete” screen. If everything is as expected, click Finish to commence the clone operation. Monitor the VM tasks for status of the clone operation.

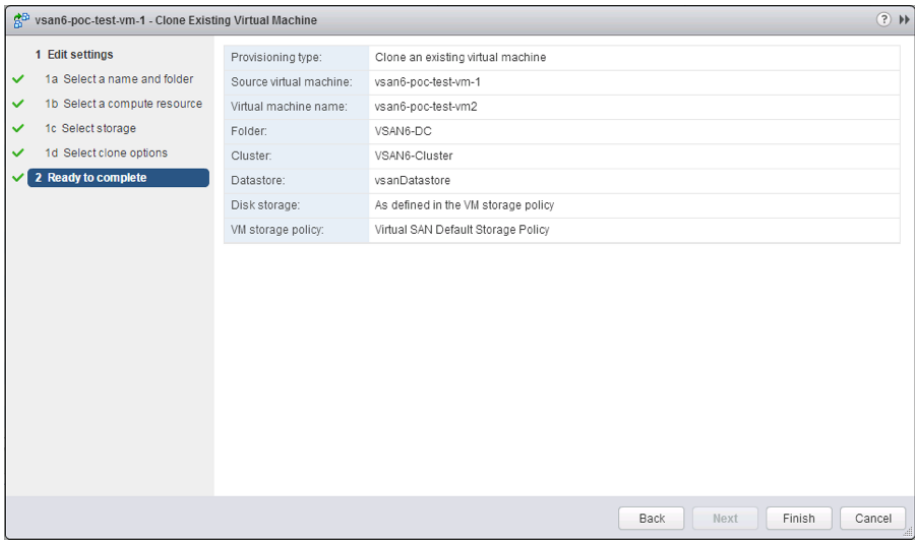


Figure 7.27: Ready to Complete

Do not delete the newly cloned VM. We will be using it in subsequent POC tests.

This completes the cloning section of this POC. Cloning with Virtual SAN has improved dramatically with the new on-disk (v2) format in version 6.0 and 6.1.

7.4 vMotion a VM between Hosts

The first step is to power-on the newly cloned virtual machine. We shall migrate this VM from one Virtual SAN host to another Virtual SAN host using vMotion.

Note: Take a moment to revisit the network configuration and ensure that the vMotion network is distinct from the Virtual SAN network. If these features share the same network, performance will not be optimal.

First, determine which ESXi host the VM currently resides on. Selecting the "Summary" tab of the VM does this. On this POC, the VM that we wish to migrate is on host *cs-ie-h01.ie.local*.

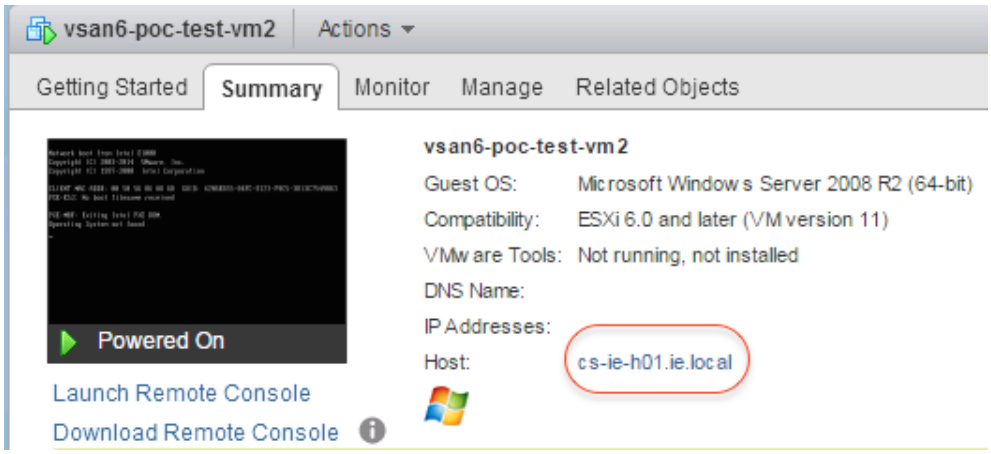


Figure 7.28: VM Summary tab – Host is displayed

Right click on the VM and select **Migrate**.

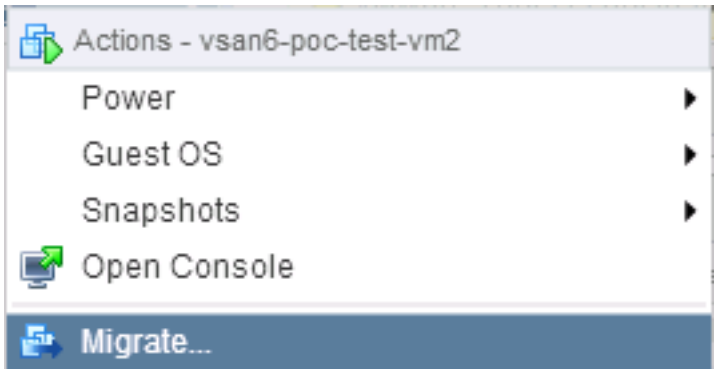


Figure 7.29: Migrate

Migrate allows you to migrate to a different compute resource (host), a different datastore or both at the same time. In this initial test, we are simply migrating the VM to another host in the cluster, so this initial screen can be left at the default of "Change compute resource only". The rest of the screens in the migration wizard are pretty self-explanatory.

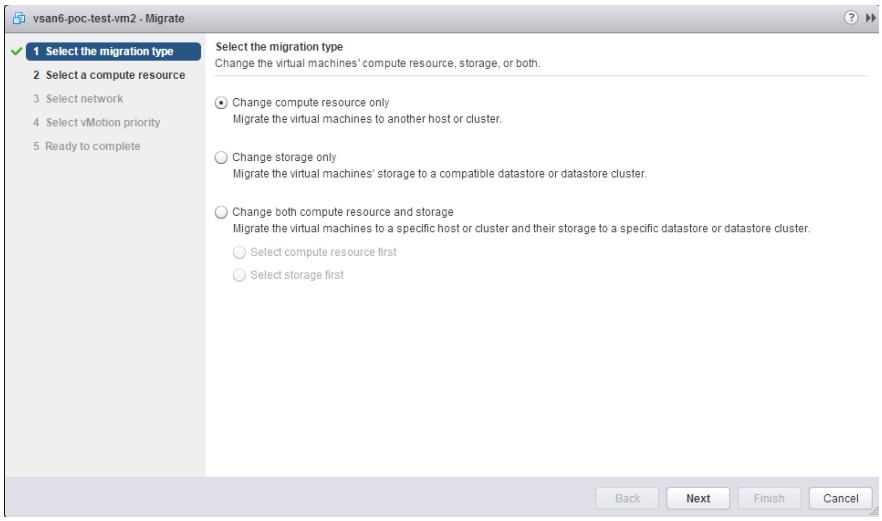


Figure 7.30: Change compute resources only

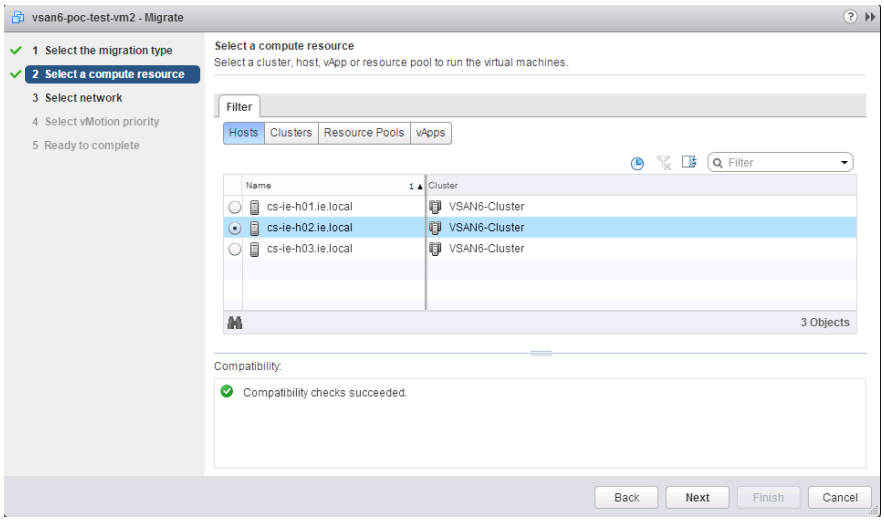


Figure 7.31: Select a destination host

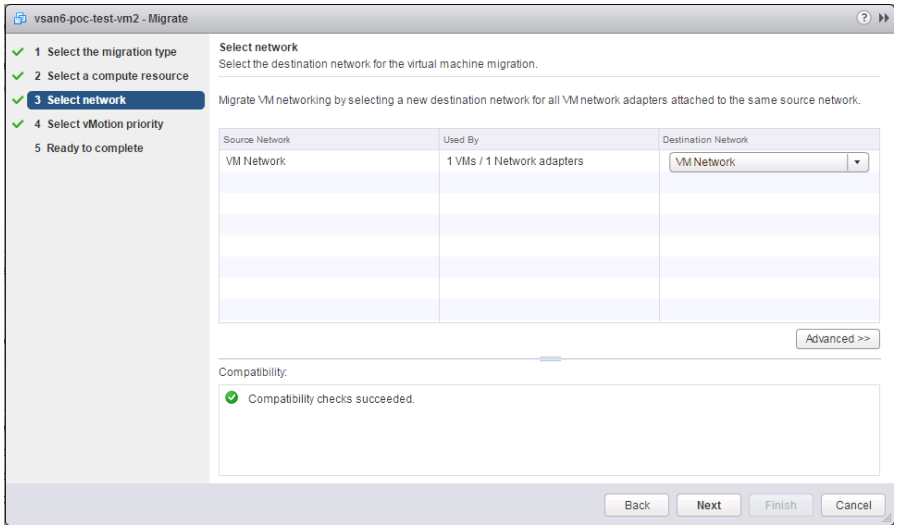


Figure 7.32: Select a destination network

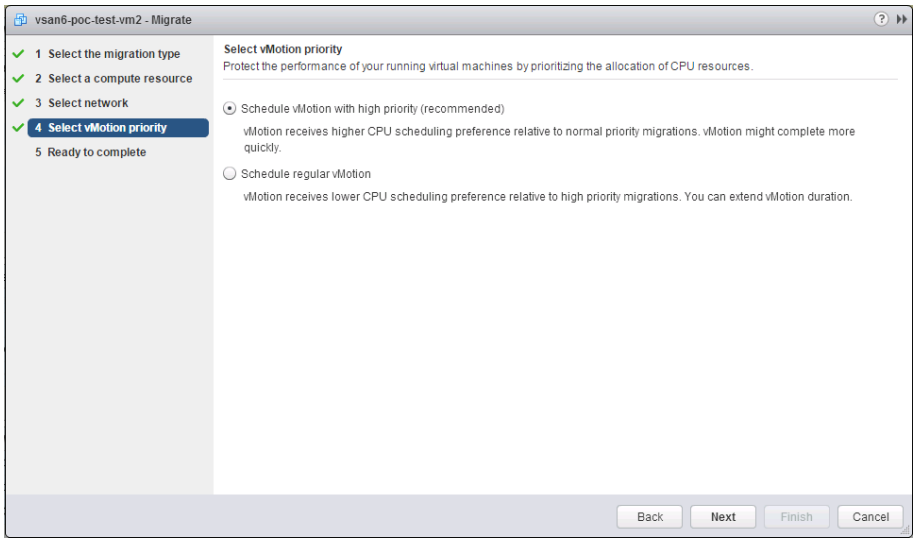


Figure 7.33: Priority can be left as high (default)

At the "Ready to Complete" window, click on Finish to initiate the migration. If the migration is successful, the summary tab of the virtual machine should show that the VM now resides on a different host.

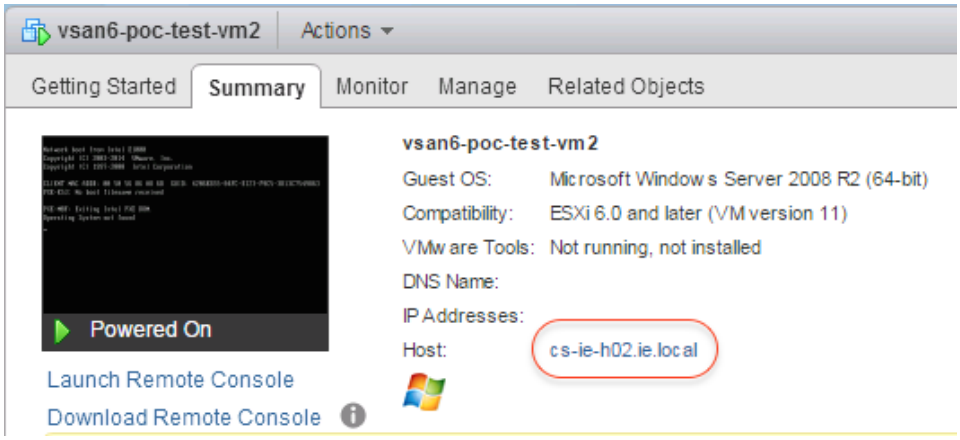


Figure 7.34: Verify VM has migrated to new host

Do not delete the migrated VM. We will be using it in subsequent POC tests.

This completes the "VM migration using vMotion" section of this POC. As you can see, vMotion works just great with Virtual SAN.

7.5 Optional: Storage vMotion a VM between Datastores

This test will only be possible if you have another datastore type available to your hosts, such as NFS/VMFS. If so, then the objective of this test is to migrate the VM from another datastore type into Virtual SAN. The VMFS datastore can even be a local VMFS disk on the host.

7.5.1 Mount an NFS Datastore to the Hosts

The steps to mount an NFS datastore to multiple ESXi hosts are described in the vSphere 6.0 Administrators Guide. See the Create NFS Datastore in the vSphere Client topic for detailed steps.

7.5.2 Storage vMotion a VM from Virtual SAN to Another Datastore Type

Currently the VM resides on the Virtual SAN datastore. Launch the migrate wizard, just like we did in the last exercise. However, on this occasion, to move the VM from the Virtual SAN datastore to the other datastore type you need to select "Change storage only".

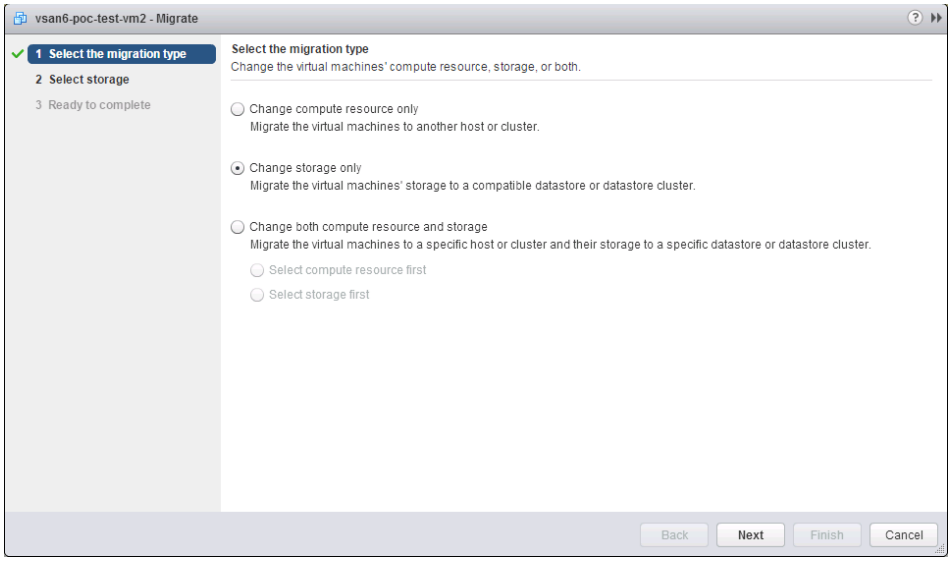


Figure 7.35: Change storage only

In this POC, we have an NFS datastore presented to each of the ESXi hosts in the Virtual SAN Cluster. This is the datastore where we are going to migrate the virtual machine to.

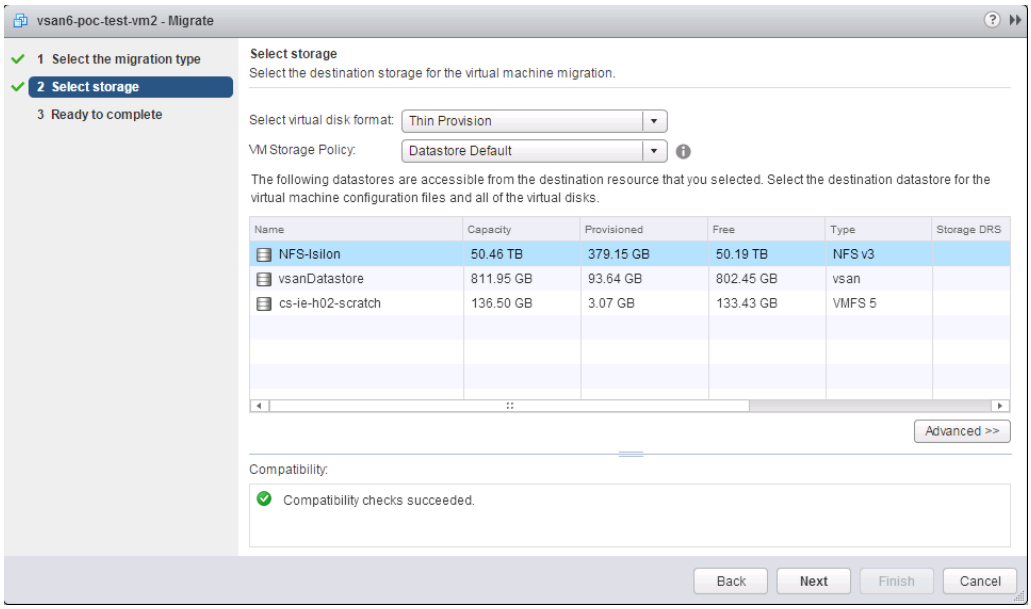


Figure 7.36: Select destination storage

One other item of interest in this step is that the VM Storage Policy should also be changed to "Datastore Default" as the NFS datastore will not understand the Virtual SAN policy settings.

At the "Ready to complete" screen, click "Finish" to initiate the migration:

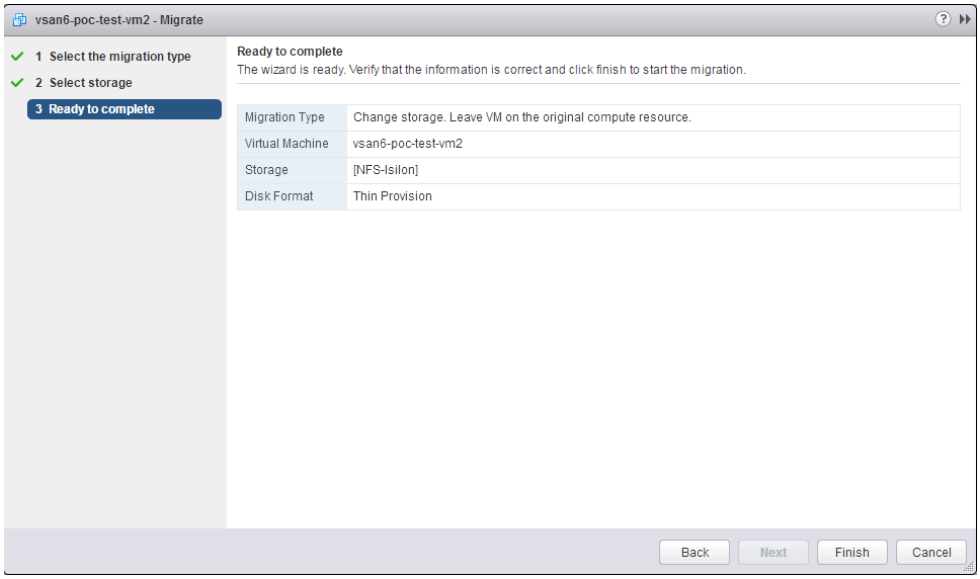


Figure 7.37: Ready to complete

Once the migration completes, the VM Summary tab can be used to examine the datastore on which the VM resides.

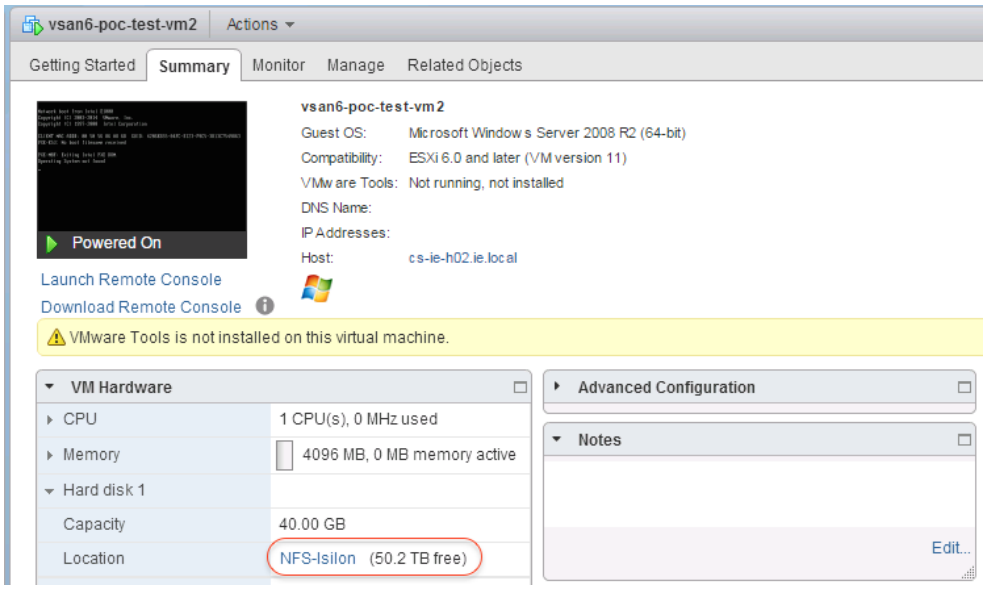


Figure 7.38: Verify VM has moved to new storage

7.5.3 Storage vMotion of VM to Virtual SAN from Another Datastore Type

Now Storage vMotion the virtual machine back to the Virtual SAN datastore to prove that Storage vMotion works in both directions. This now completes the optional "VM migration using Storage vMotion" section of this POC. Different storage policies can be chosen as part of the migration.

Storage vMotion works seamlessly with Virtual SAN.

8. Scale out Virtual SAN

One of the really nice features is the simplistic scale-out nature of Virtual SAN. If you need more compute or storage resources in the cluster, simply add another host to the cluster.

Let's remind ourselves about how our cluster currently looks. There are currently three hosts in the cluster, and there is a fourth host not in the cluster. We also created two VMs in the previous exercises.

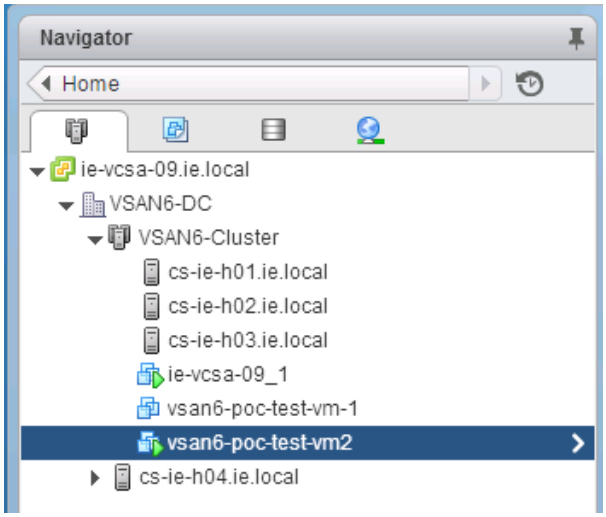


Figure 8.1: Current inventory status

Let us also remind ourselves of how big the Virtual SAN datastore is.

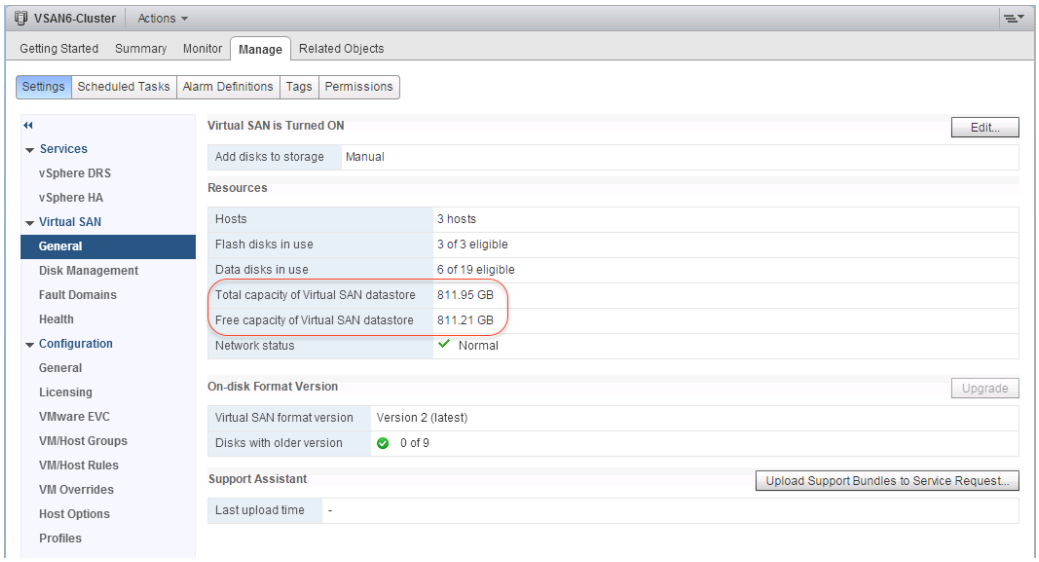


Figure 8.2: Total and Free Virtual SAN datastore capacity

In this POC, the Virtual SAN datastore is 811.95GB in size with 811.21GB free.

8.1 Add the Fourth Host to Virtual SAN Cluster

We will now proceed with adding a fourth host to the Virtual SAN Cluster.

Note: Back in section 5 of this POC guide, you should have already setup a Virtual SAN network for this host. If you have not done that, revisit section 5, and setup the Virtual SAN network on this fourth host.

Having verified that the networking is configured correctly on the fourth host, select the cluster object in the inventory, right click on it and select the option "Move Hosts into Cluster…" as shown below.

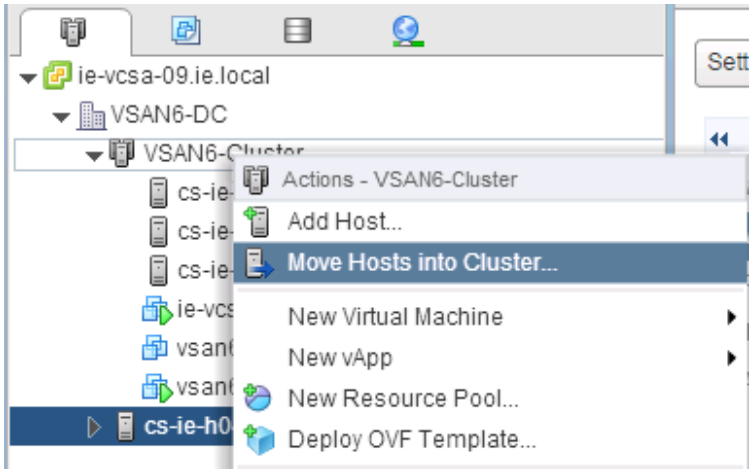


Figure 8.3: Move hosts into Cluster

You will then be prompted to select which host to move into the cluster. In this POC, there is only one additional host. Select that host.

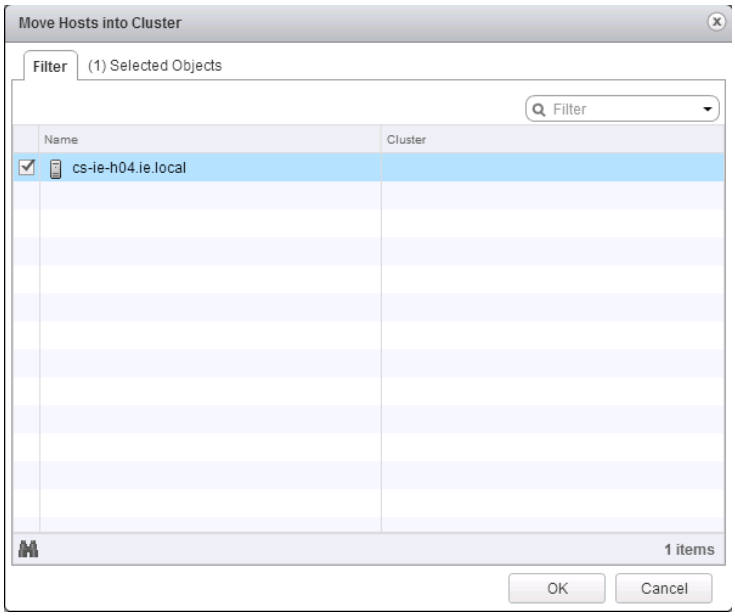


Figure 8.4: Select a host to move into the cluster

The next screen is related to resource pools. You can leave this at the default, which is to use the cluster's root resource pool, then click OK.

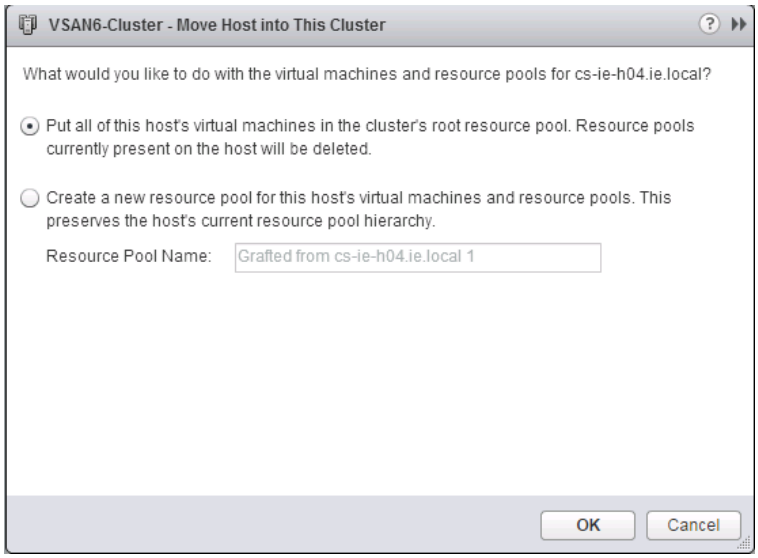


Figure 8.5: Resource Pools

This moves the host into the cluster. Next, navigate to the Manage tab > Settings > Virtual SAN > General view and verify that the cluster now contains the new node.

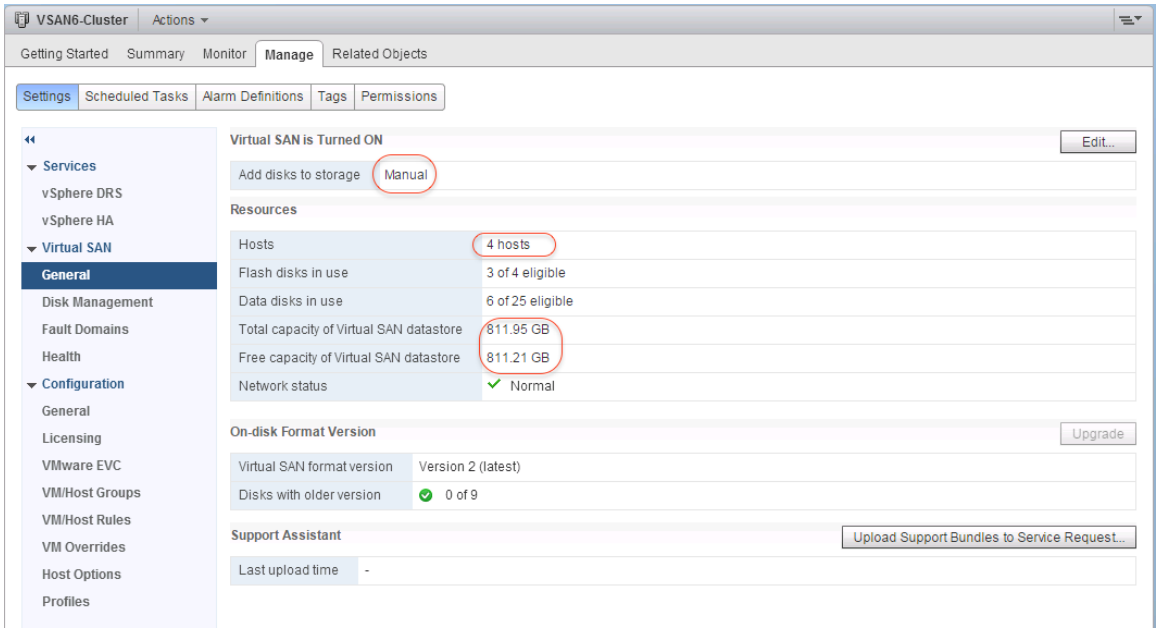


Figure 8.6: Resource Pools

As you can clearly see, there are now 4 hosts in the cluster. However, you will also notice that the Virtual SAN datastore has not changed with regards to total and free capacity. This is because the cluster was configured in "Manual" mode back in section 6. Therefore Virtual SAN will not claim any of the disks automatically. You will need to create a disk group for the new host and claim disks manually. At this point, it

would be good practice to re-run the health check tests. If there are any issues with the fourth host joining the cluster, use the Virtual SAN Health check to check where the issue lies. Verify that the host appears in the same network partition group as the other hosts in the cluster.

8.2 Manual Option: Create Disk Group on New Host

This process has already been covered in section 6.2. Navigate to the Disk Management section, select the new host and then click on the icon to create a new disk group:

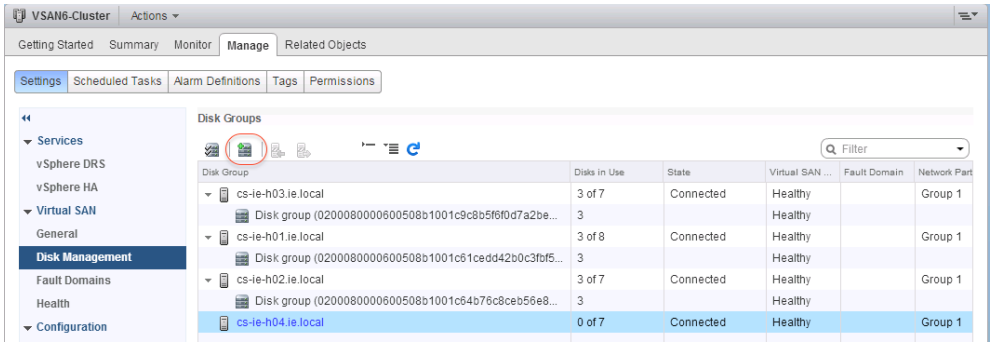


Figure 8.7: Create a new disk group

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

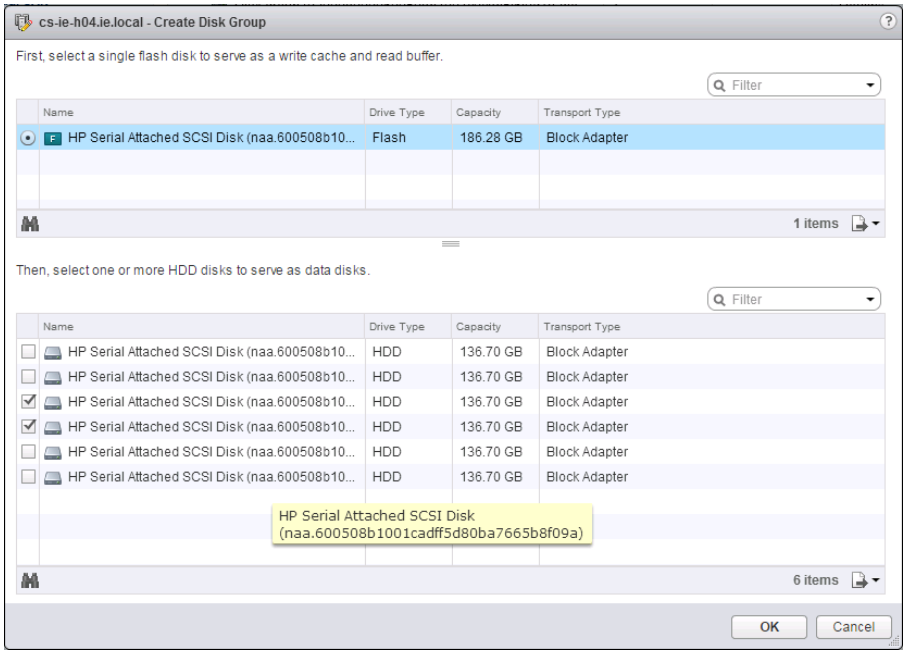


Figure 8.8: Select flash and capacity devices

8.3 Verify Virtual SAN Disk Group Configuration on New Host

Once the disk group has been created, the disk management view should be revisited to ensure that it is healthy.

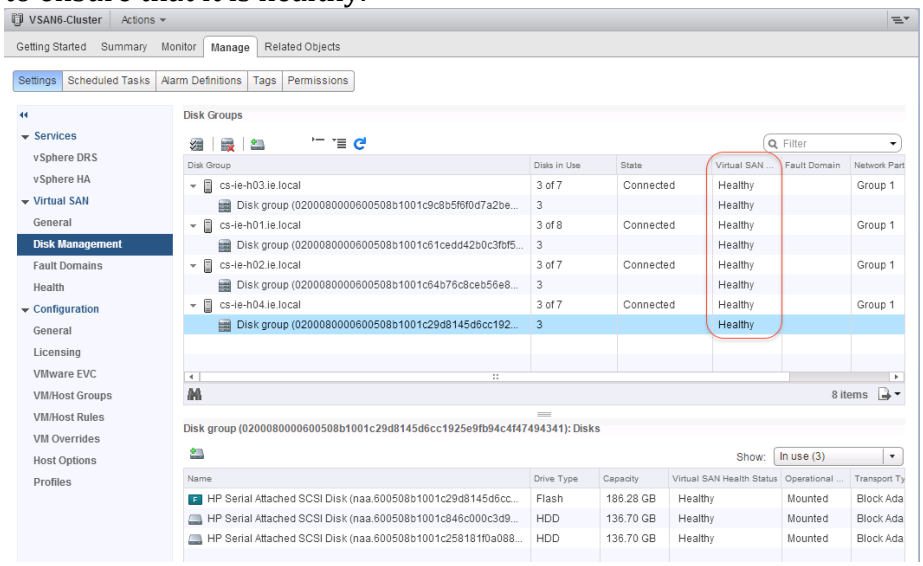


Figure 8.9: Check disk group health

8.4 Verify New Virtual SAN Datastore Capacity

The final step is to ensure that the Virtual SAN datastore has now grown in accordance to the capacity devices in the disk group that was just added on the fourth host. Return to the General tab and examine the total and free capacity field.

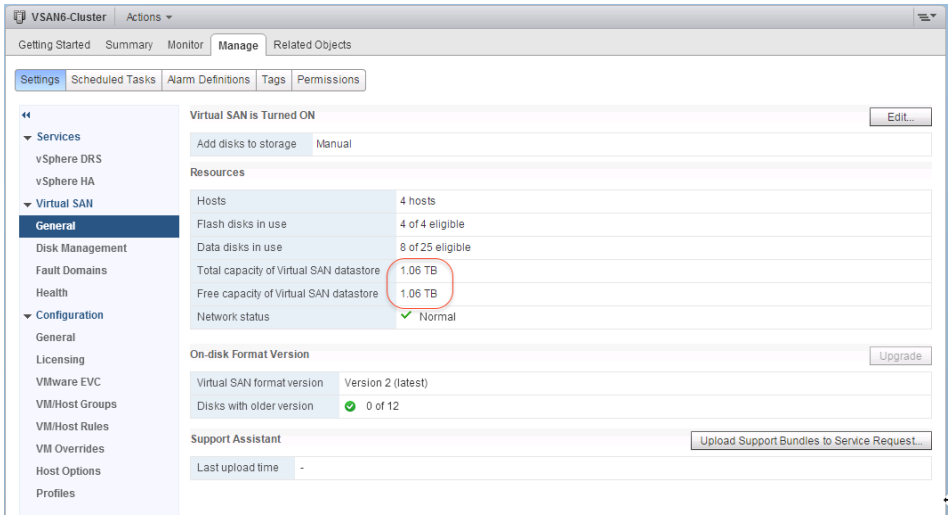


Figure 8.10: Virtual SAN Datastore capacity details

As we can clearly see, the Virtual SAN datastore has now grown in size to 1.06TB. Free space is shown as 1.06TB as the amount of space used is minimal.

This completes the “Scale Out” section of this POC. As seen, scale-out on Virtual SAN is simple but very powerful.

9. VM Storage Policies and Virtual SAN

VM Storage Policies form the basis of VMware's Software Defined Storage vision. Rather than deploying VMs directly to a datastore, a VM Storage Policy is chosen during initial deployment. The policy contains characteristics and capabilities of the storage required by the virtual machine. Based on the policy contents, the correct underlying storage is chosen for the VM.

If the underlying storage meets the VM storage Policy requirements, the VM is said to be in a compatible state.

If the underlying storage fails to meet the VM storage Policy requirements, the VM is said to be in an incompatible state.

In this section of the POC Guide, we shall look at various aspects of VM Storage Policies. The virtual machines that have been deployed thus far have used the default storage policy, which has the following settings:

- *NumberOfFailuresToTolerate* = 1
- *NumberOfDiskObjectsToStripe* = 1
- *ObjectSpaceReservation* = 0%
- *FlashReadCacheReservation* = 0%
- *ForceProvisioning = False*

We will create some additional policies in this section of the POC.

9.1 Create a New VM Storage Policy

In this part of the POC, we will build a policy that creates a stripe width of 2 for each storage object deployed with this policy. The VM Storage Policies can be accessed from the Home page on the vSphere web client as shown below.

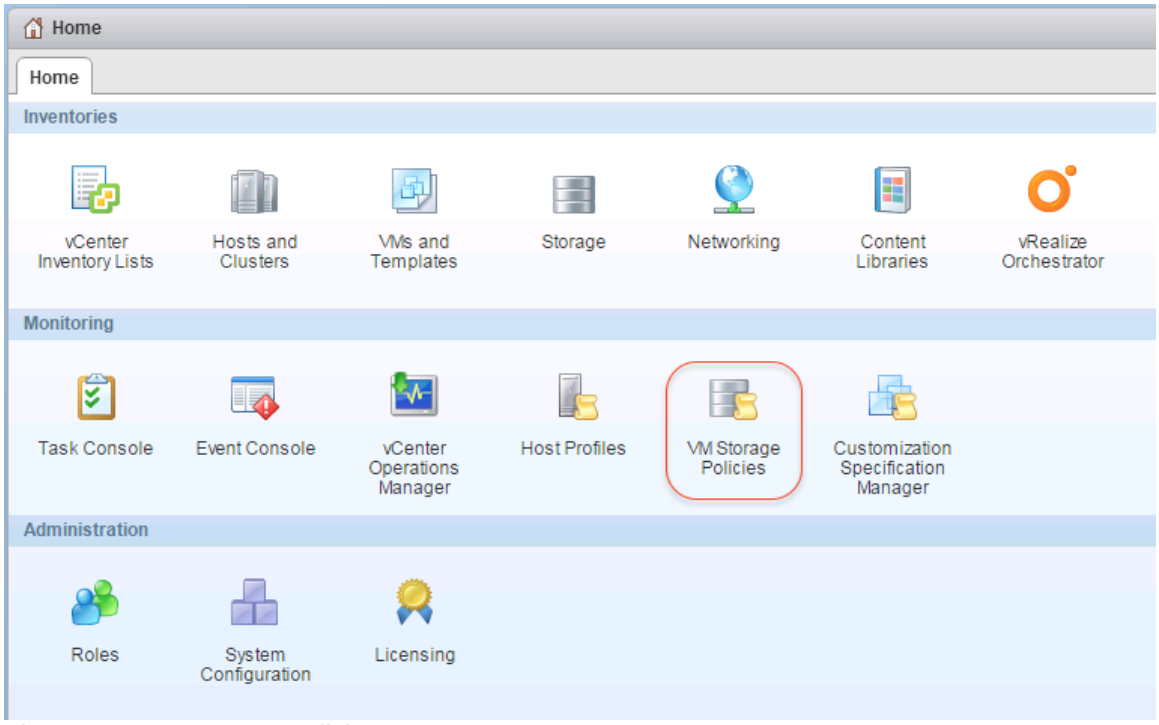


Figure 9.1: VM Storage Policies

There will be some existing policies already in place, such as the Virtual SAN Default Storage policy, which we've already used to deploy VMs in section 7 of this POC guide. There is another policy called "VVol No Requirements Policy", which is used for Virtual Volumes and is not applicable to Virtual SAN. There are a number of icons on this page that may need further explanation:

	Create a new VM Storage Policy
	Edit an existing VM Storage Policy
	Delete an existing VM Storage Policy
	Check the compliance of VMs using this VM Storage Policy
	Clone an existing VM Storage Policy

Table 9.1: VM Storage Policy icons

To create a new policy, click on the "Create a new VM Storage Policy" icon.

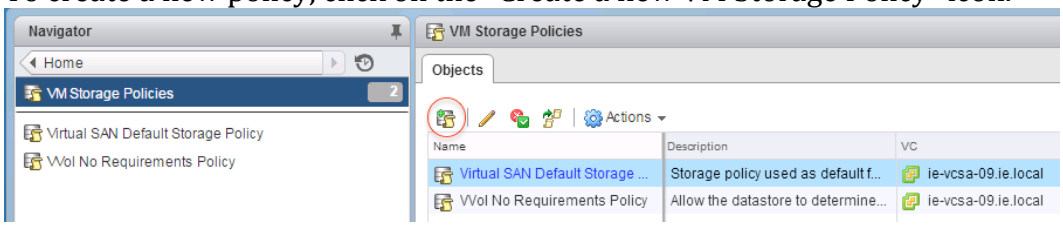


Figure 9.2: Create a new VM Storage Policy

The next step is to provide a name and an optional description for the new VM Storage Policy. Since this policy will contain a stripe width of 2, we have given it a name to reflect this. You may also give it a name to reflect that it is a Virtual SAN policy.

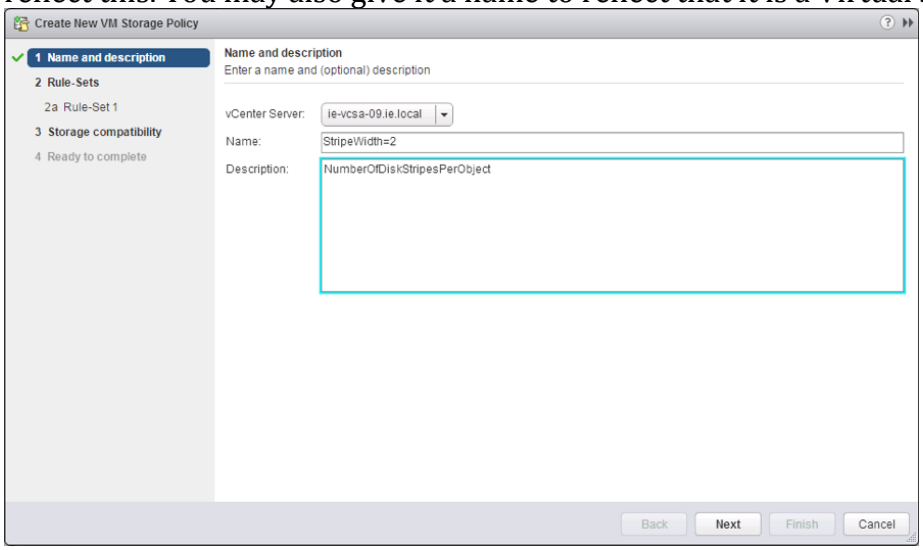


Figure 9.3: VM Storage Policy Name and Description

The next section contains a description of Rule-Sets and how to use them.

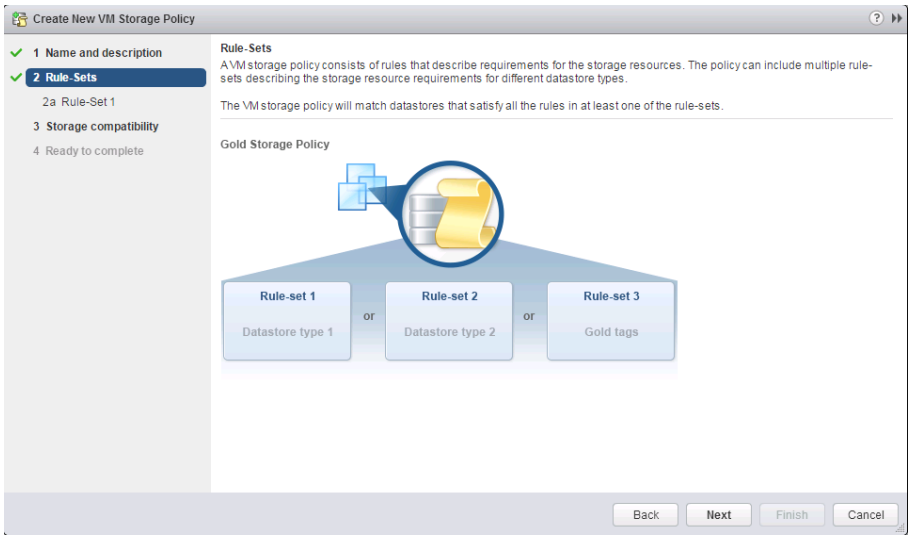


Figure 9.4: Rule-Sets

Now we get to the point where we create a set of rules for our Rule-Set (we are only creating a single Rule-Set in this VM Storage Policy). The first step is to select "Virtual

SAN" as the "Rules based on data services". Once this is selected, the five customizable capabilities associated with Virtual SAN are exposed. Since this VM Storage Policy is going to have a requirement where the stripe width of an object is set to two, this is what we select from the list of rules. It is officially called "*Number of disk stripes per object*".

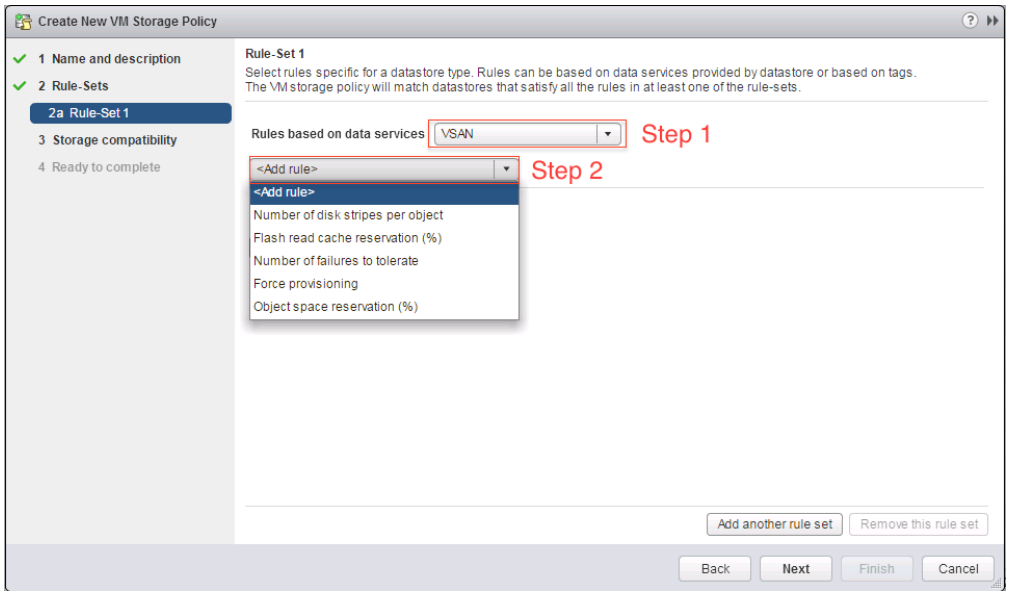


Figure 9.5: Number of disk stripes per object

We also want to set this value to 2. Once the disk stripe rule is chosen, change the default value from 1 to 2 as shown below. Notice also the Storage Consumption Model display on the right hand side, detailing how much disk space will be consumed based on the rules placed in the policy.

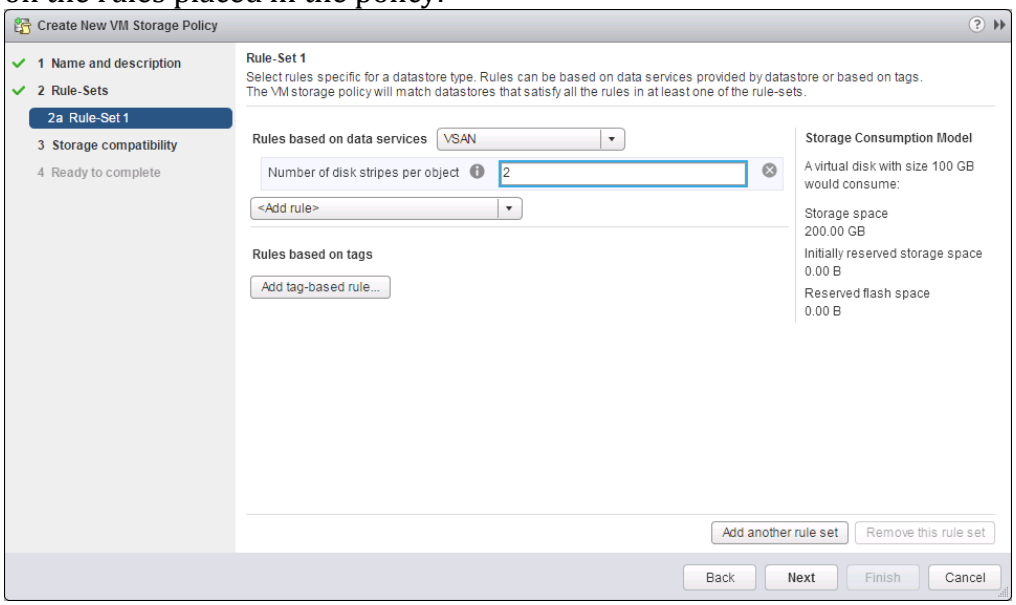


Figure 9.6: Setting Stripe Width to 2

Clicking next moves on to the Storage Compatibility screen. Note that this displays which storage "understands" the policy settings. In this case, the vsanDatastore is the only datastore that is compatible with the policy settings.

Note: This does not mean that the Virtual SAN datastore can successfully deploy a VM with this policy; it simply means that the Virtual SAN datastore understands the rules or requirements in the policy.

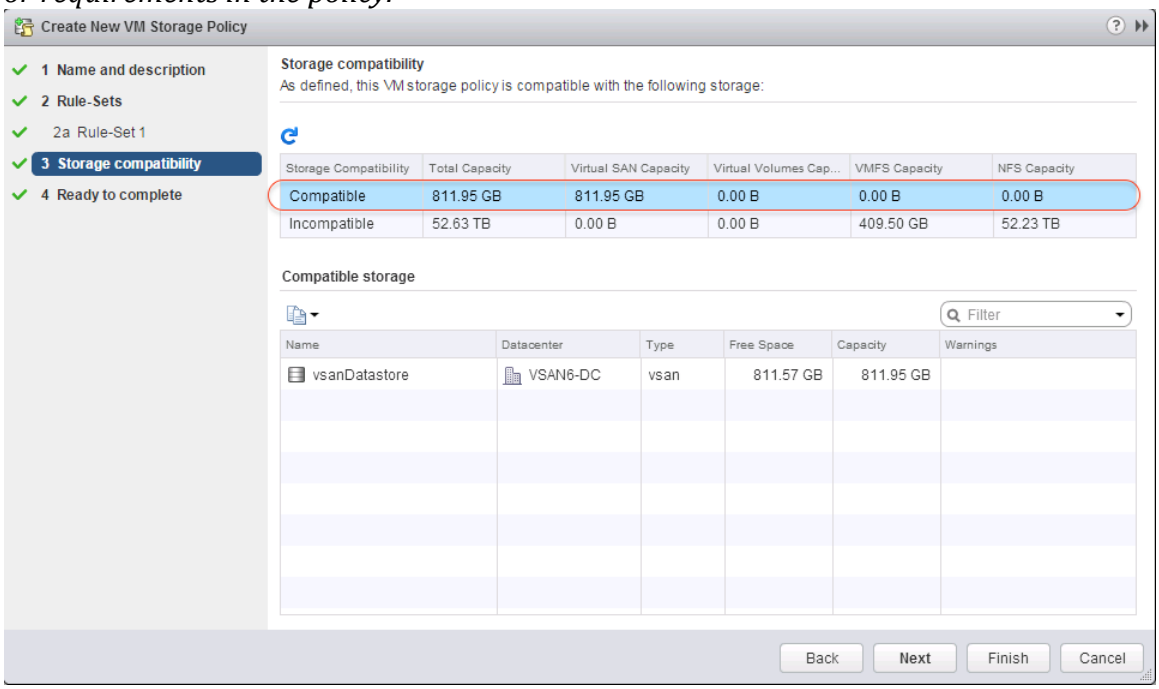


Figure 9.7: Storage Compatibility

At this point, you can click on next to review the settings once more, or alternatively, at this point, you can click "Finish" instead of reviewing the policy. On clicking "Finish", the policy is created.

Let's now go ahead and deploy a VM with this new policy, and let's see what effect it has on the layout of the underlying storage objects.

9.2 Deploy a New VM with the New VM Storage Policy

We have already deployed a VM back in 7.1. The steps will be identical, until we get to the point where the VM Storage Policy is chosen. This time, instead of selecting the default policy, we will select the newly created *StripeWidth=2* policy as shown below.

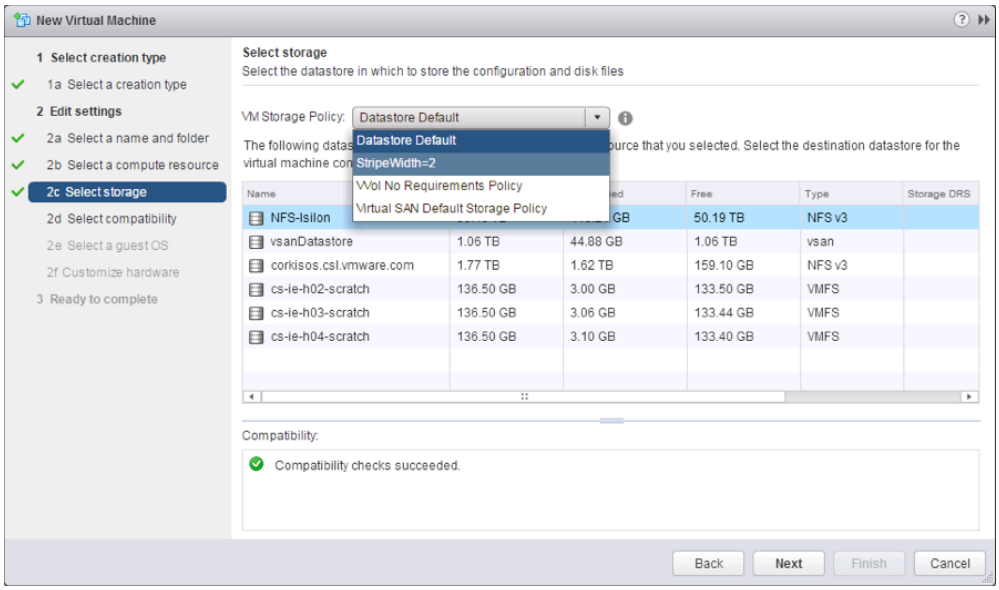


Figure 9.8: Selecting a non-default policy

And as before, the vsanDatastore should show up as the compatible datastore, and thus the one to which this VM should be provisioned if we wish to have the VM compliant with its policy.

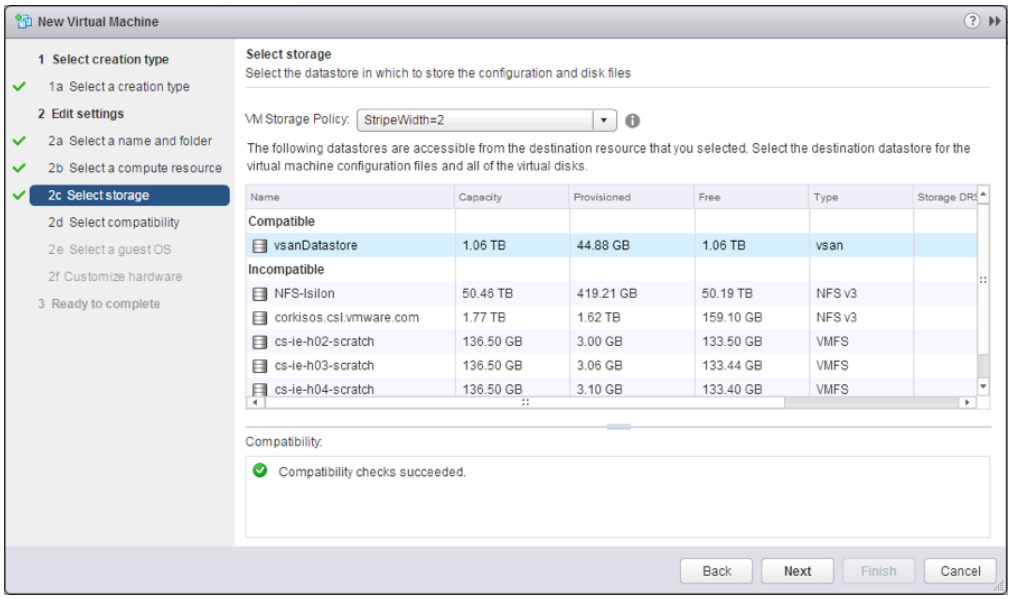


Figure 9.9: vsanDatastore is compatible with the policy

Let's now go ahead and examine the layout of this virtual machine, and see if the policy requirements are met; i.e. do the storage objects of this VM have a stripe width

of 2? First, ensure that the VM is compliant with the policy by navigating to VM > Manage tab > Policies, as shown here.

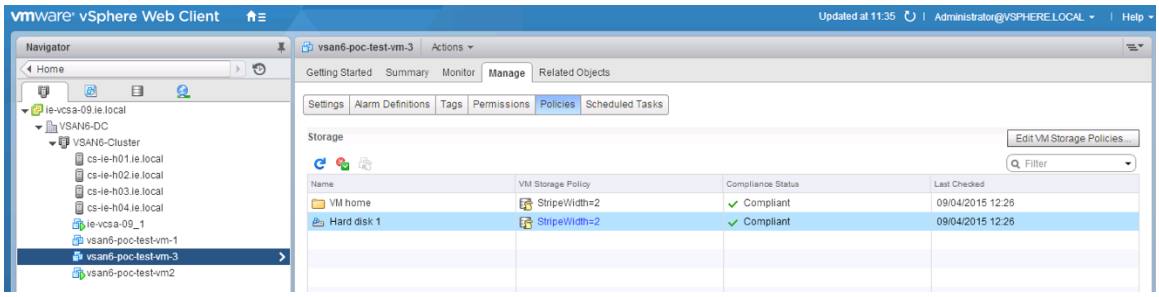


Figure 9.10: VM is compliant with the policy

The next step is to select the Monitor tab > Policies and check the layout of the VM's storage objects. The first object to check is the VM home namespace. Select it, and then select the "Physical Disk Placement" tab at the lower part of the window. This continues to show that there is only one mirrored component, but no stripe width (which is displayed as a RAID 0 configuration). Why? The reason for this is that the VM home namespace object does not benefit from striping so it ignores this policy setting. Therefore this behavior is normal and to be expected.

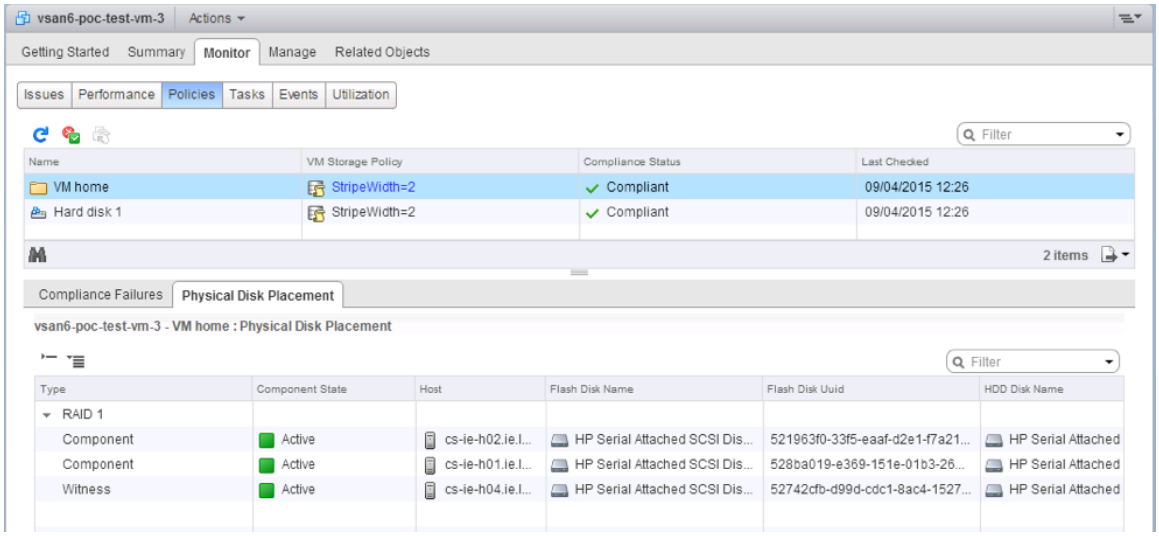


Figure 9.11: VM home namespace ignores stripe width policy setting

Now let's examine "Hard disk 1" and see if that layout is adhering to the policy. Here we can clearly see a difference. Each replica or mirror copy of the data now contains two components in a RAID 0 configuration. This implies that the hard disk storage objects are indeed adhering to the stripe width requirement that was placed in the VM Storage Policy.

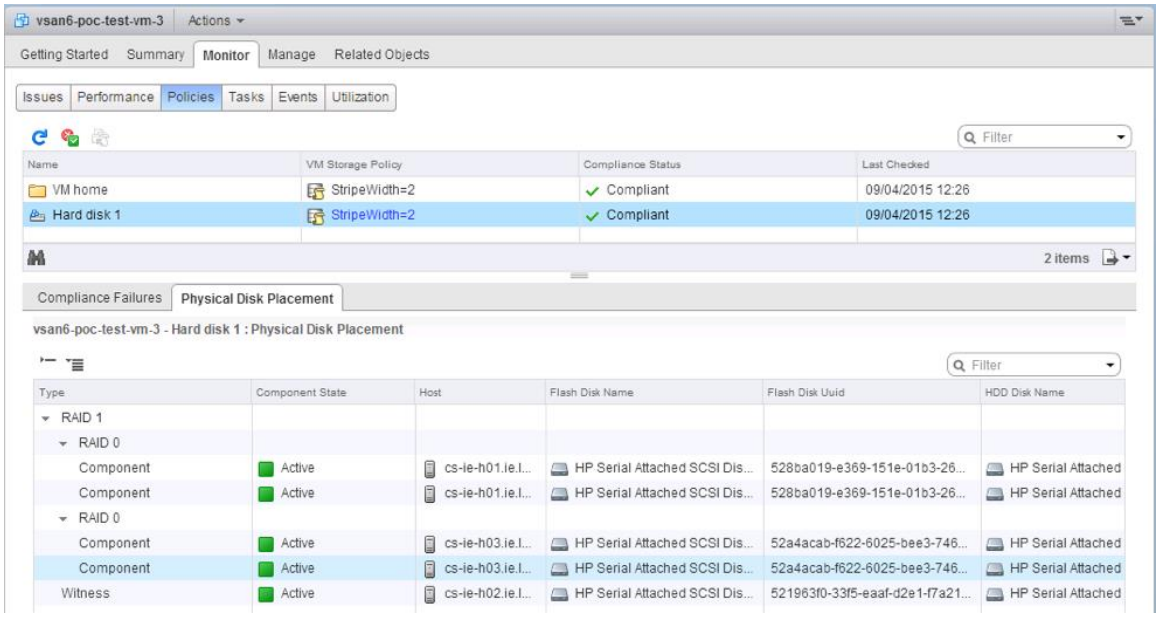


Figure 9.12: Hard disks adhere to stripe width policy setting

Note that each striped component must be placed on its own physical disk. There are enough physical disks to meet this requirement in this POC. However, a request for a larger stripe width would not be possible in this configuration. Keep this in mind if you plan a POC with a large stripe width value in the policy.

It should also be noted that snapshots taken of this base disk continue to inherit the policy of the base disk, implying that the snapshot delta objects will also be striped.

One final item to note is the fact that this VM automatically has a *NumberOfFailuresToTolerate*=1, even though it was not explicitly requested in the policy. We can tell this from the RAID 1 configuration in the layout. Virtual SAN will always provide availability to VMs via the *NumberOfFailuresToTolerate* policy setting, even when it is not requested via the policy. The only way to deploy a VM without a replica copy is by placing *NumberOfFailuresToTolerate*=0 in the policy.

A useful rule of thumb for *NumberOfFailuresToTolerate* is that in order to tolerate n failures in a cluster, you require a minimum of 2n + 1 hosts in the cluster (to retain a >50% quorum with n host failures).

9.3 Add a New VM Storage Policy to an Existing VM

Virtual Machines may also have new VM Storage Policies added after they have been deployed to the Virtual SAN datastore. The configuration of the objects will be changed when the new policy is added. That may mean the adding of new components to existing objects, for example in the case where the *NumberOfFailuresToTolerate* is increased. It may also involve the creation of new objects that are synced to the original object, and once synchronized, the original object is discarded. This is typically only seen when the layout of the object changes, such as increasing the *NumberOfDiskStripesPerObject*.

In this case, we will add the new StripeWidth=2 policy to one of the VMs created in section 7 which still only has the default policy (*NumberOfFailuresToTolerate*=1, *NumberOfDiskStripesPerObject*=1, *ObjectSpaceReservation*=0) associated with it.

To begin, select the VM that is going to have its policy changed from the vCenter inventory, then select the Manage tab > Policies view. This VM should currently be compliant with the Virtual SAN Default Storage Policy. Now click on the Edit VM Storage Policies button as highlighted below.

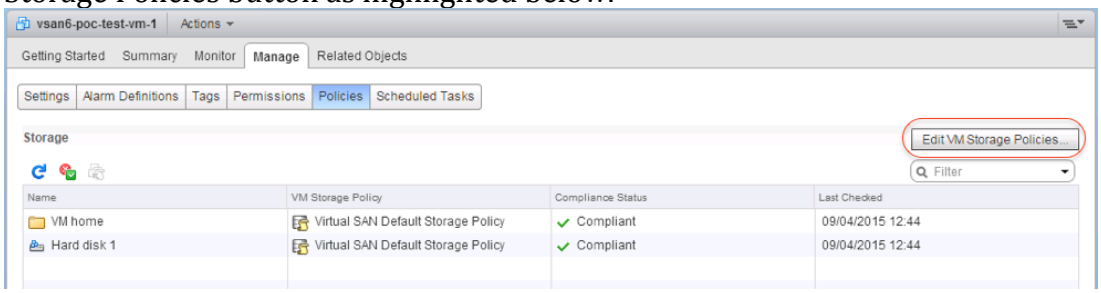


Figure 9.13: Edit VM Storage Policies

This takes you to the edit screen, where the policy can be changed.

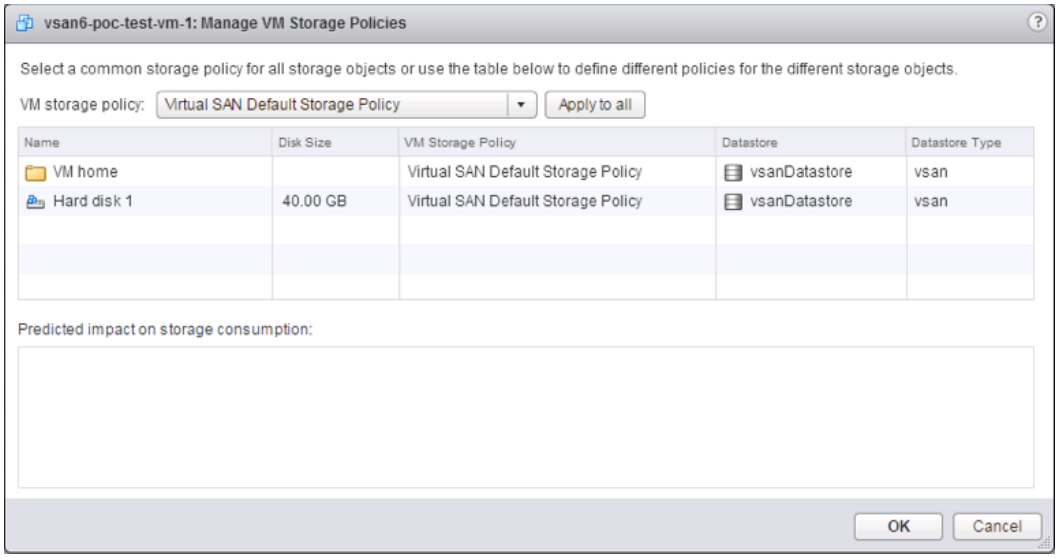


Figure 9.14: Manage VM Storage Policies

Select the new VM Storage Policy from the drop-down list. The policy that we wish to add to this VM is the StripeWidth=2 policy.

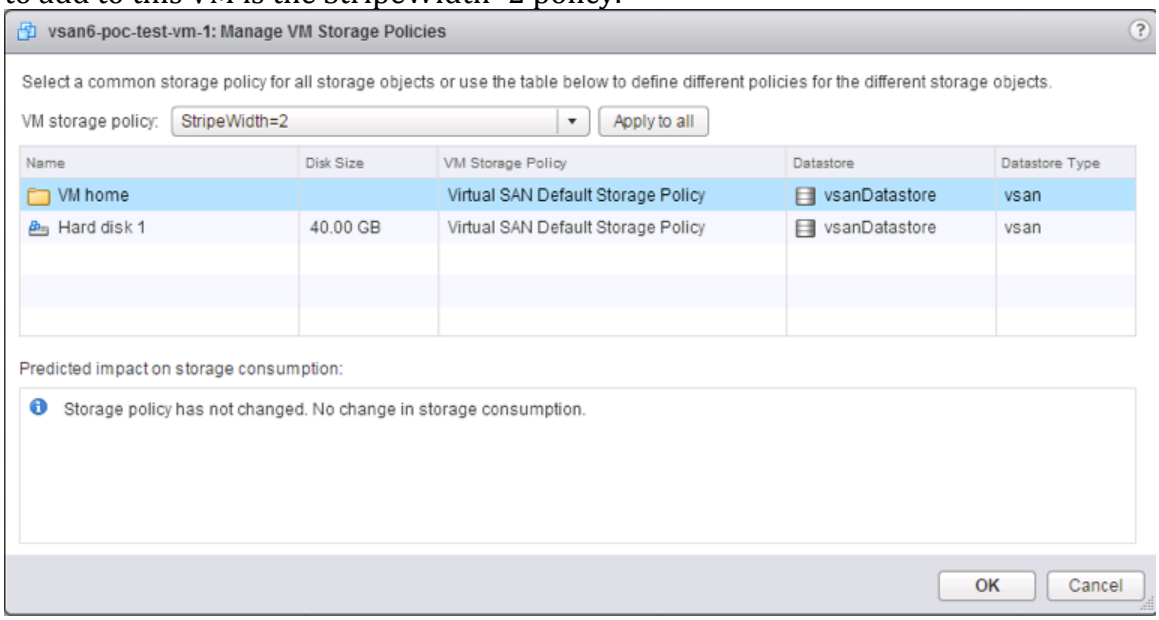


Figure 9.15: Select a new VM Storage Policies

Once the policy is select, click on the "Apply to all" button as shown below to ensure the policy gets applied to all storage objects and not just the VM home namespace object. The VM Storage Policy should now appear updated for all objects.

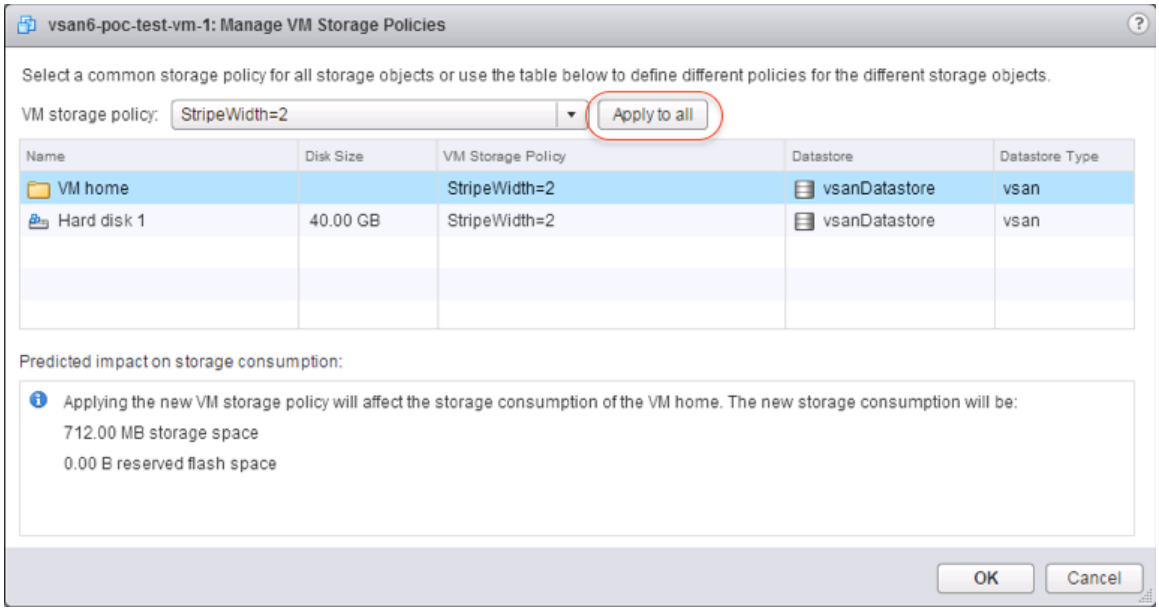


Figure 9.16: Apply to all

Next, click OK and initiate the policy change. Now when you revisit the Monitor tab > Policies view, you should see the changes in the process of taking effect (Reconfiguring) or completed, as shown below.

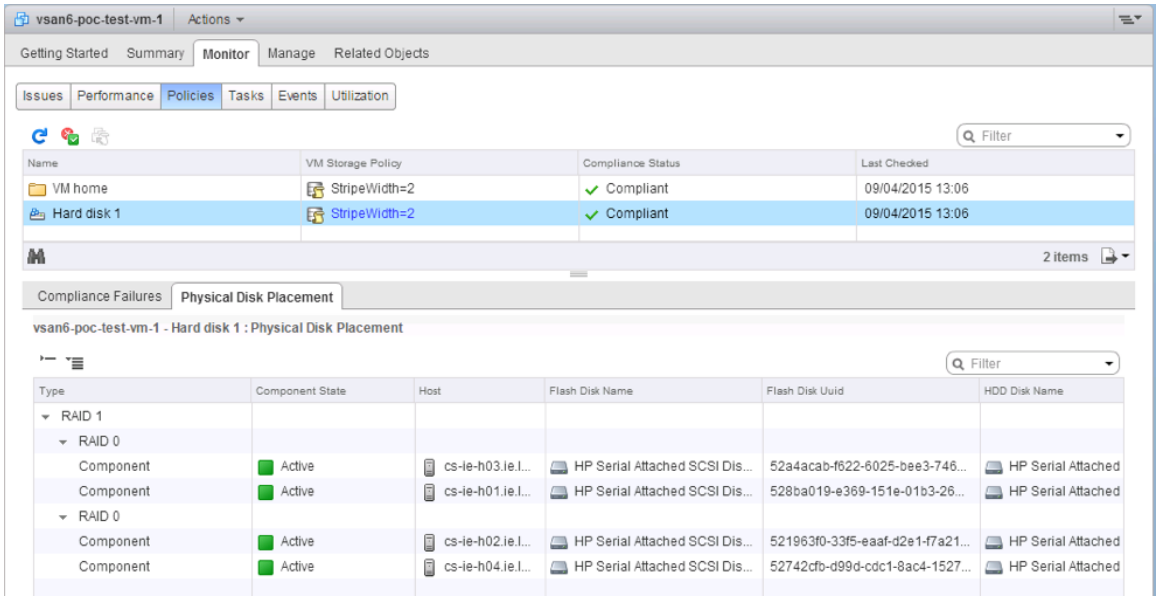


Figure 9.17: Reconfiguring complete – new policy in effect

This is useful when you only need to modify the policy of one or two VMs, but what if you need to change the VM Storage Policy of a significant number of VMs.

That can be achieved by simply changing the policy used by those VMs. All VMs using those VMs can then be "brought to compliance" by reconfiguring their storage object layout to make them compliant with the policy. We shall look at this next.

9.4 Modify a VM Storage Policy

We will modify the StripeWidth=2 policy created earlier to include an *ObjectSpaceReservation*=10%. This means that each storage object will now reserve 10% of the VMDK size on the Virtual SAN datastore. Since all VMs were deployed with 40GB VMDKs, the reservation value will be 4GB.

The first step in this task is to note the amount of free space in the Virtual SAN datastore, so you can compare it later and confirm that each VMDK has 4GB of space reserved. Next, revisit the VM Storage Policy section that we visited previously. This can be accessed once again via the Home page.

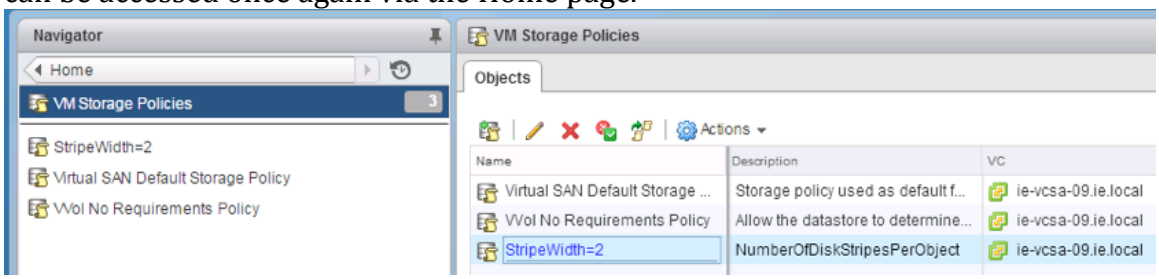


Figure 9.18: VM Storage Policies: Stripewidth

Select StripeWidth=2 policy in the left hand column, and then the Manage tab. Select "Rule-set 1: Virtual SAN" and then click on "Edit" button on the far right.

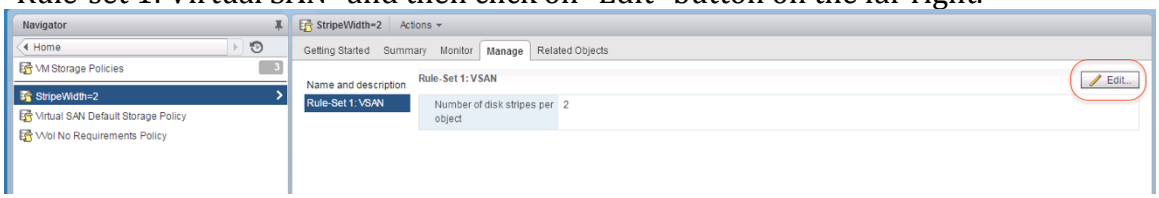


Figure 9.19: Edit Policy

From the <Add rule> drop down list, select *ObjectSpaceReservation* as a new capability to be added to the policy.

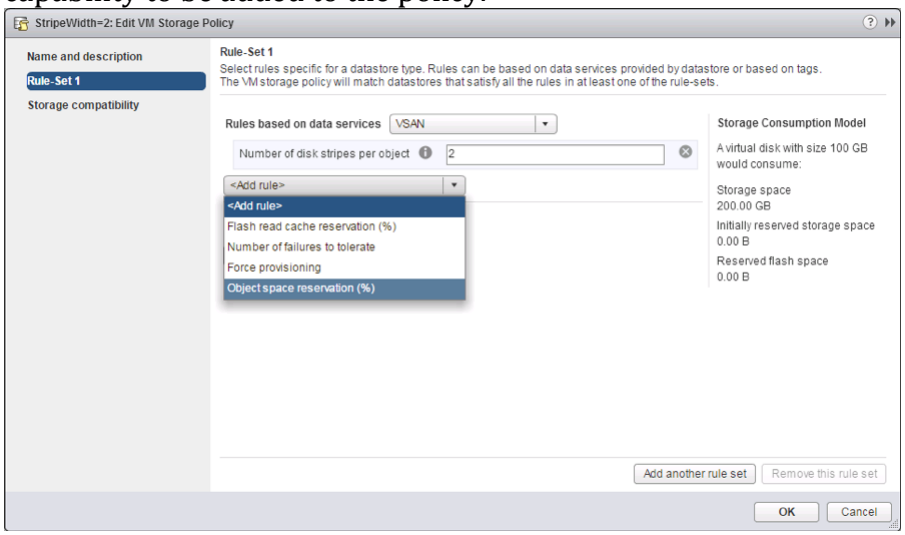


Figure 9.20: Add Object space reservation (%) as a rule to the policy

Set *ObjectSpaceReservation* to 10%. Note Storage Consumption calculations on right.

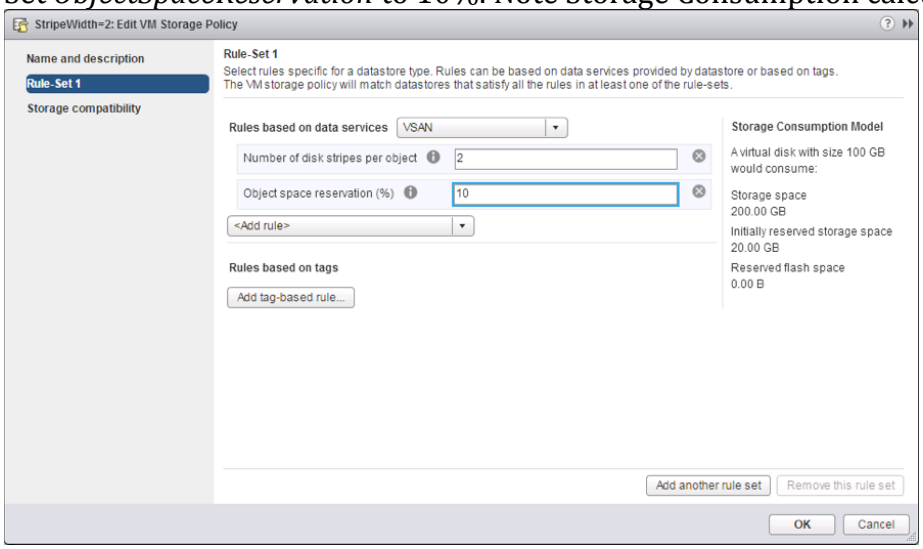


Figure 9.21: Set Object space reservation to 10%

After clicking OK to make the change. The wizard will prompt you as to whether you want to reapply this change to the virtual machines using this policy manually later (default) or automatically now. It also tells you how many VMs in the environment are using the policy and will be affected by the change. Leave it at the default, which is "Manually later", by clicking Yes. This POC guide will show you how to do this manually shortly.

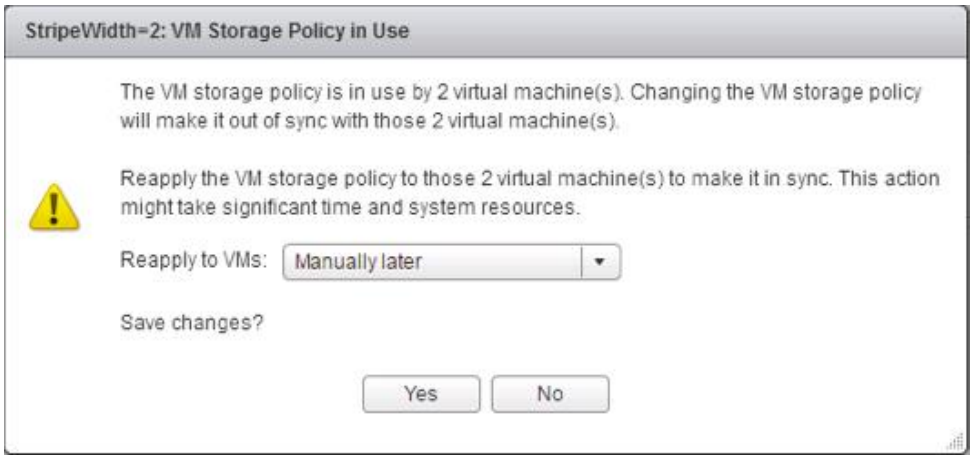


Figure 9.22: Manually later

Next, click on the Monitor tab next to the Manage tab. It will display the two VMs along with their storage objects, and the fact that they are no longer compliant with the policy. They are in an "Out of Date" compliance state as the policy has now been changed.

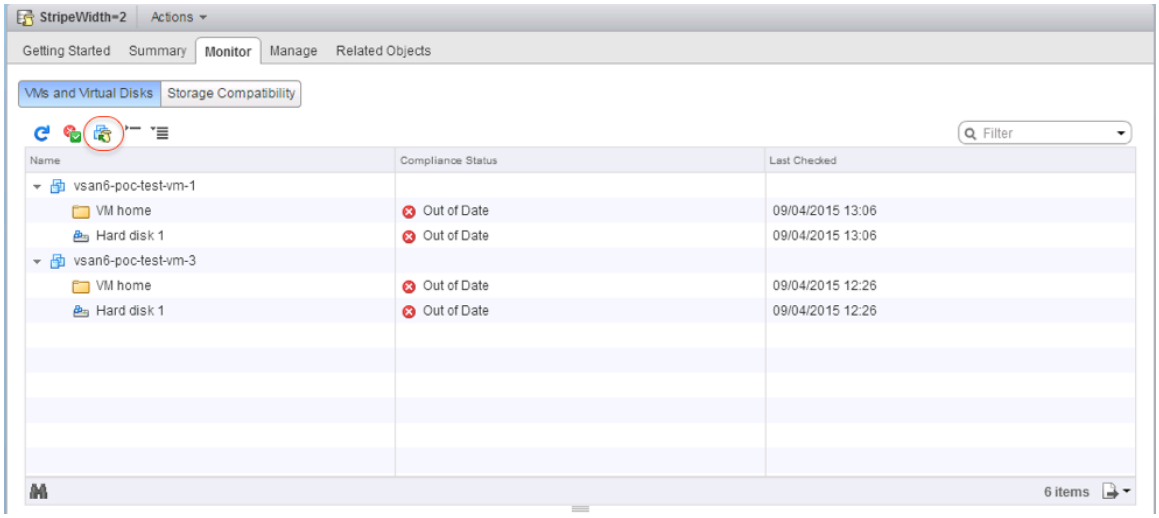


Figure 9.23: Out of Date

In order to bring the VM to a compliant state, we must manually reapply the VM Storage Policy to the objects. The button to do this action is highlighted in the previous screenshot. When this button is clicked, the following popup appears.

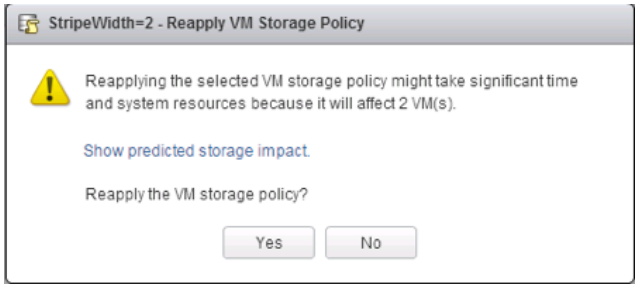


Figure 9.24: Reapply VM Storage Policy

When the reconfigure activity completes against the storage objects, and the compliance state is once again checked, everything should show as Compliant.

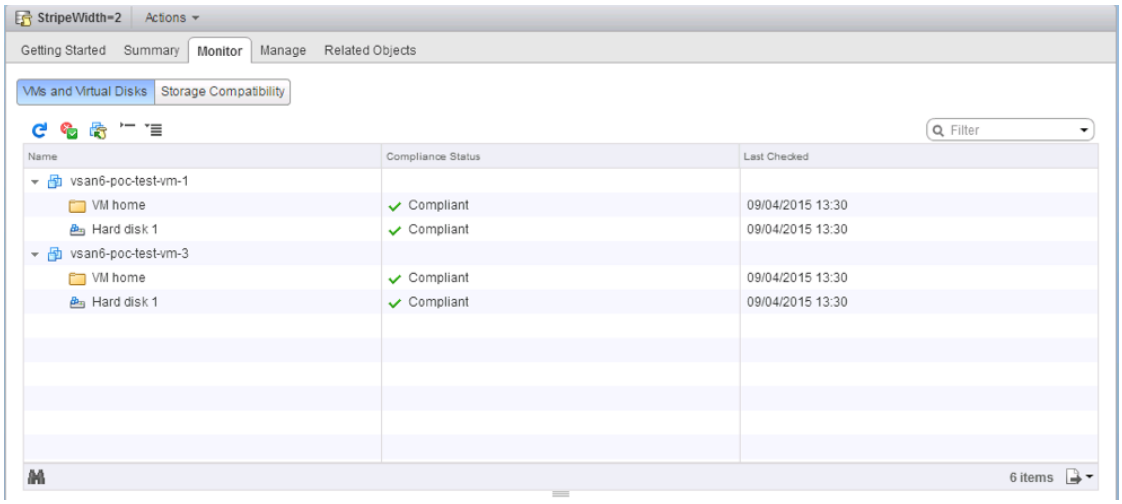


Figure 9.25: Compliant once again

Since we have now included an *ObjectSpaceReservation* value in the policy, what you may notice is that the amount of free capacity on the Virtual SAN datastore will have reduced.

For example, the two VMs with the new policy change have 40GB storage objects. Therefore there is a 10% *ObjectSpaceReservation* implying 4GB is reserved per VMDK. 4GB per VMDK, 1 VMDK per VM, 2 VMs equals 8GB reserved space, right? However the VMDK is also mirrored, so there is a total of 16GB reserved on the Virtual SAN datastore.

Checking the Virtual SAN datastore, we can see this reflected in the free capacity.

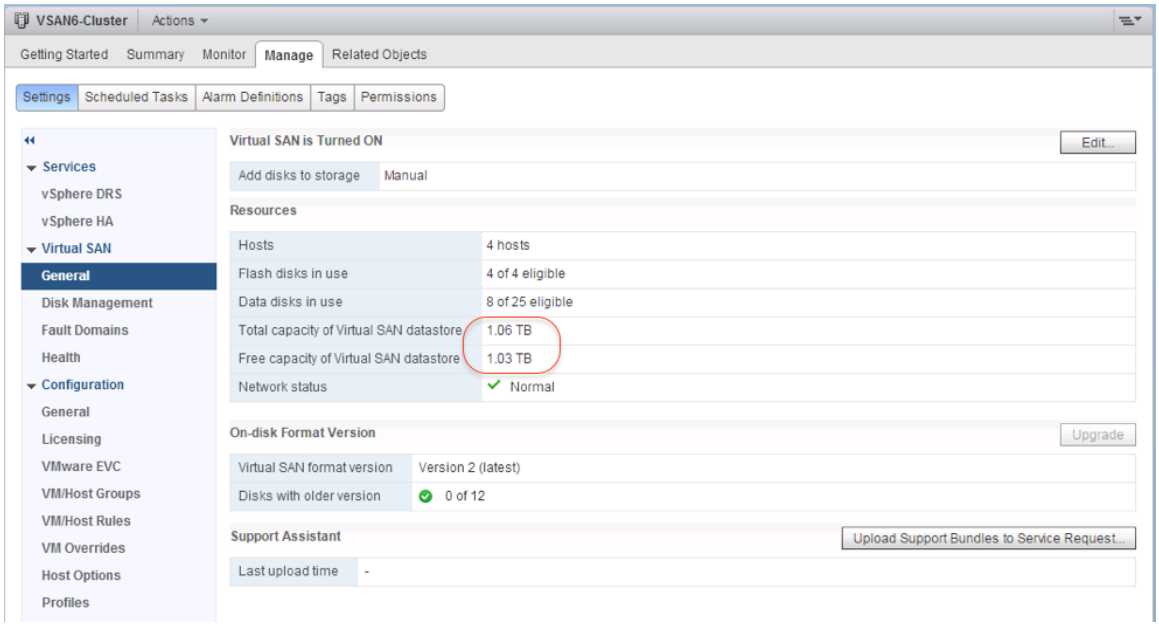


Figure 9.26: ObjectSpace Reservation consuming capacity

This completes the "VM Storage Policies" section of this POC. You should now appreciate how powerful VM Storage Policies are, and how characteristics of the underlying storage can be assigned to virtual machines on a granular per VMDK basis while using a single Virtual SAN datastore.

10. Virtual SAN Monitoring

When it comes to monitoring Virtual SAN, there are a number of areas that need particular attention. In no particular order, these are considerations when it comes to monitoring Virtual SAN:

- Monitor the Virtual SAN Cluster
- Monitor Virtual Devices in the Virtual SAN Cluster
- Monitor Physical Devices in Virtual SAN Datastores
- Monitor Resynchronization & Rebalance Operations in the Virtual SAN Cluster
- Examine Default Virtual SAN Alarms
- Triggering Alarms based on Virtual SAN VMkernel Observations Alarms

10.1 Monitor the Virtual SAN Cluster

The first item to monitor is the overall health of the cluster. The Manage > General view gives you a good idea as to whether all the flash and capacity devices that you expect to be in use are in fact in use. It also shows whether the network status is normal or not. Finally, it is a good indicator as to whether or not the expected capacity of the Virtual SAN datastore is correct, and if there are any capacity concerns looming.

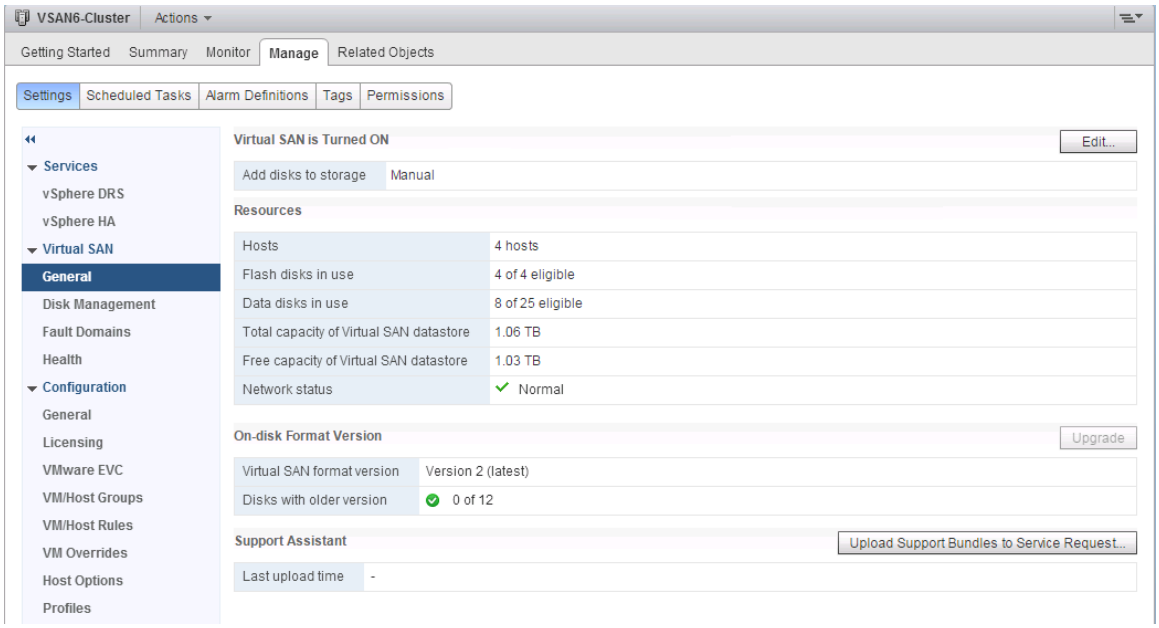


Figure 10.1: General view

Virtual SAN Health checks will display even more information regarding health and should be enabled as part of any Virtual SAN 6.1 POC.

10.2 Monitor Virtual Devices in the Virtual SAN Cluster

To monitor the virtual devices, navigate to Monitor > Virtual SAN > Virtual Disks. This will list the objects associated with each virtual machine, such as the VM home namespace and the hard disks. One can also see the policy, compliance state and health of an object. If one selects an object, physical disk placement and compliance failures are displayed in the lower half of the screen.

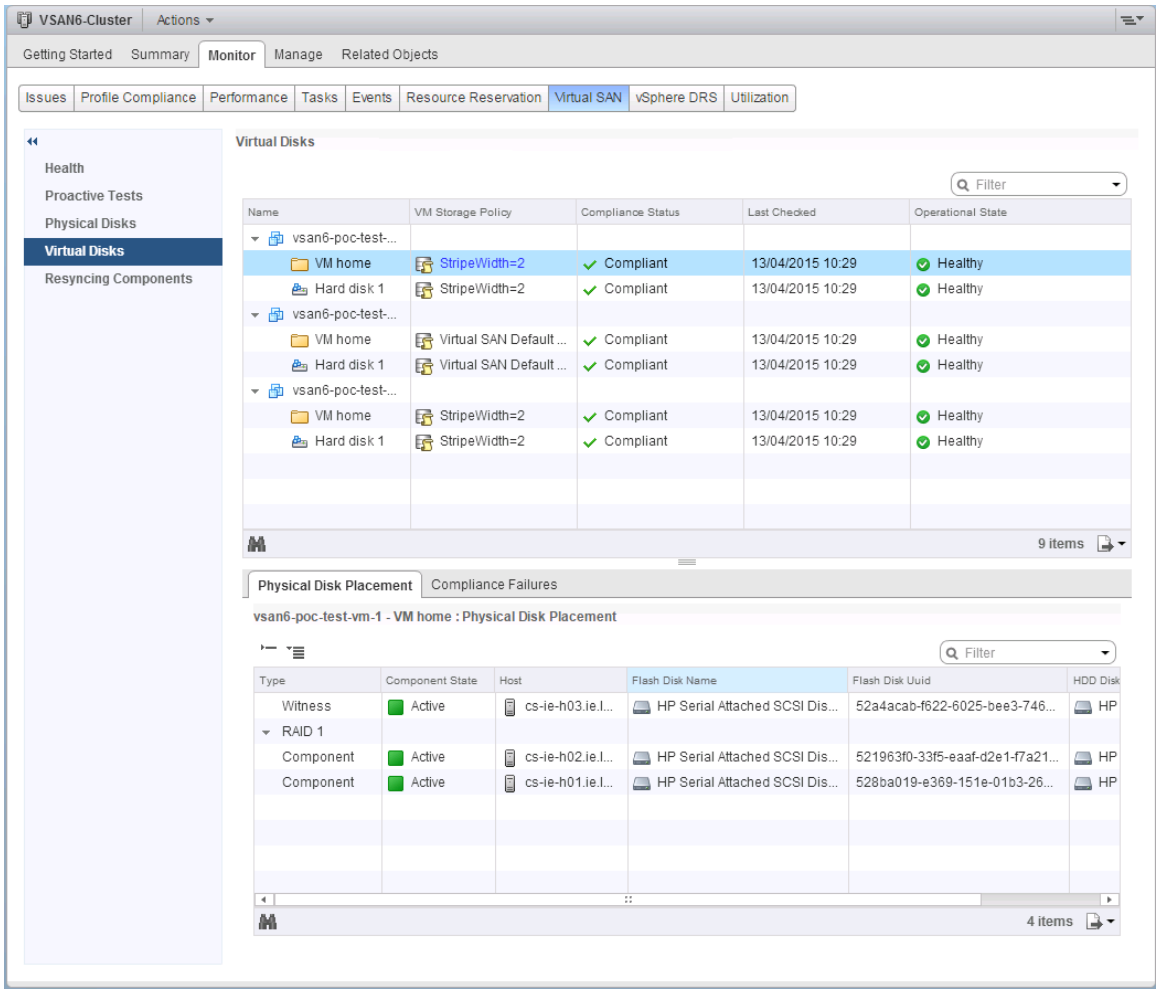


Figure 10.2: Virtual Disks view

All objects should be compliant and healthy. All components in the physical disk placement view should appear as “Active”.

10.3 Monitor Physical Devices in the Virtual SAN Cluster

In the same monitor > Virtual SAN view, physical disks can also be displayed. Where this view is very useful is when you wish to see which objects reside on a particular physical disk. In the view below, one of the magnetic disks is selected and in the lower half of the screen, the objects that have components residing on that physical disk are displayed.

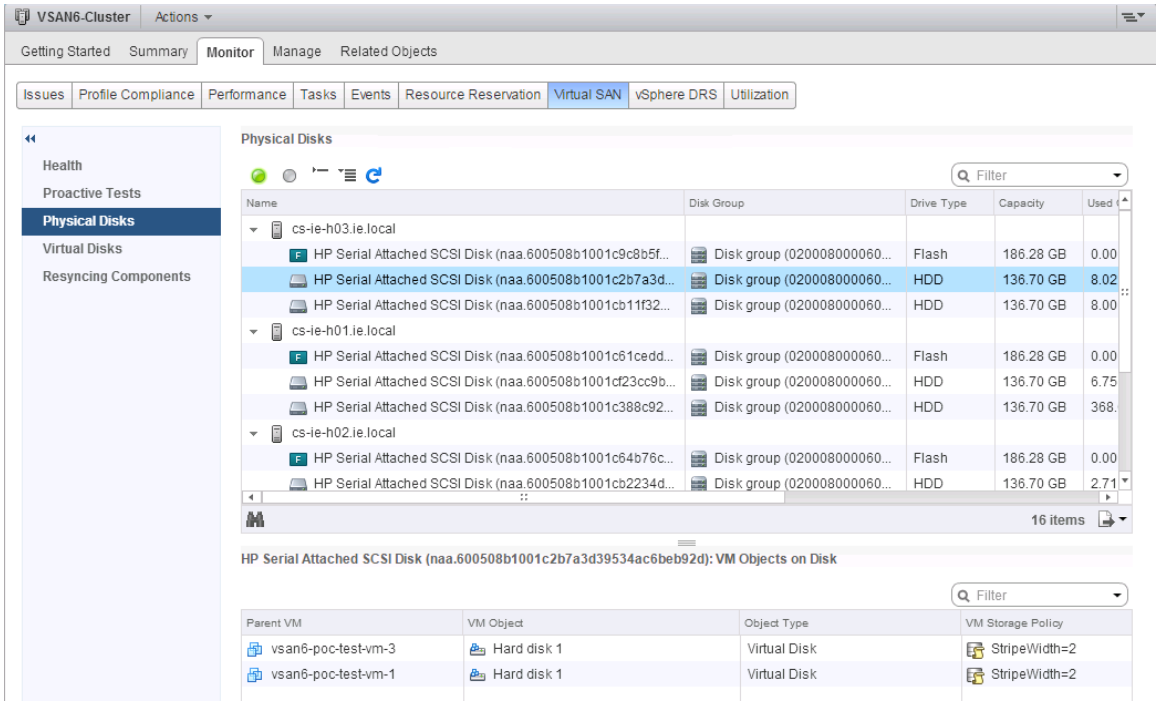


Figure 10.3: Physical Disks view

10.4 Monitor Resynchronization and Rebalance Operations

Another very useful view in this Monitor > Virtual SAN tab is "Resyncing components". This will display any rebuilding or rebalancing operations that might be taking place on the cluster. For example, if there was a device failure, resyncing or rebuilding activity could be observed here. Similarly, if a device was removed or a host failed, and the CLOMd (Cluster Logical Object Manager daemon) timer expired (60 minutes by default), rebuilding activity would also be observed in this case.

With regards to rebalancing, Virtual SAN attempts to keep all physical disks at less than 80% capacity. If any physical disks' capacity passes this threshold, Virtual SAN will move components from this disk to other disks in the cluster in order to rebalance the physical storage.

By default, there should be no resyncing activity taking place on the Virtual SAN Cluster, as shown below. Resyncing activity usually indicates:

(a) a failure of a device or host in the cluster
(b) a device has been removed from the cluster
(c) a physical disk have greater than 80% of its capacity consumed
(d) a policy change has been implemented which necessitates a rebuilding of a VM's object layout. In this case, the new object layout is created, synchronized to the original object, and then the original object is discarded.

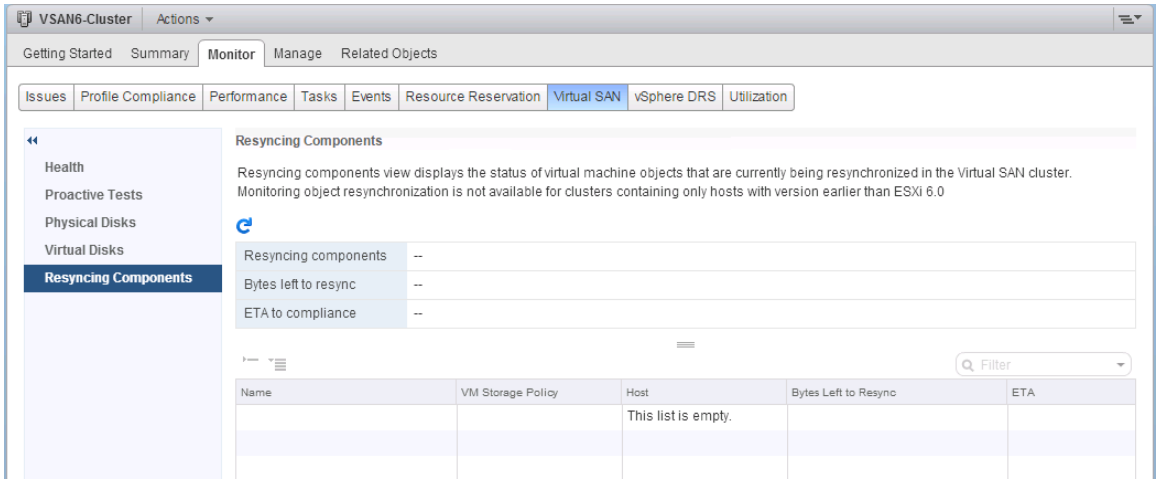


Figure 10.4: Resyncing components

10.5 Default Virtual SAN Alarms

There are at least 56 Virtual SAN alarms pre-defined in vCenter server 6.0u1. Some are shown here, and the majority relate to Virtual SAN Health issues:

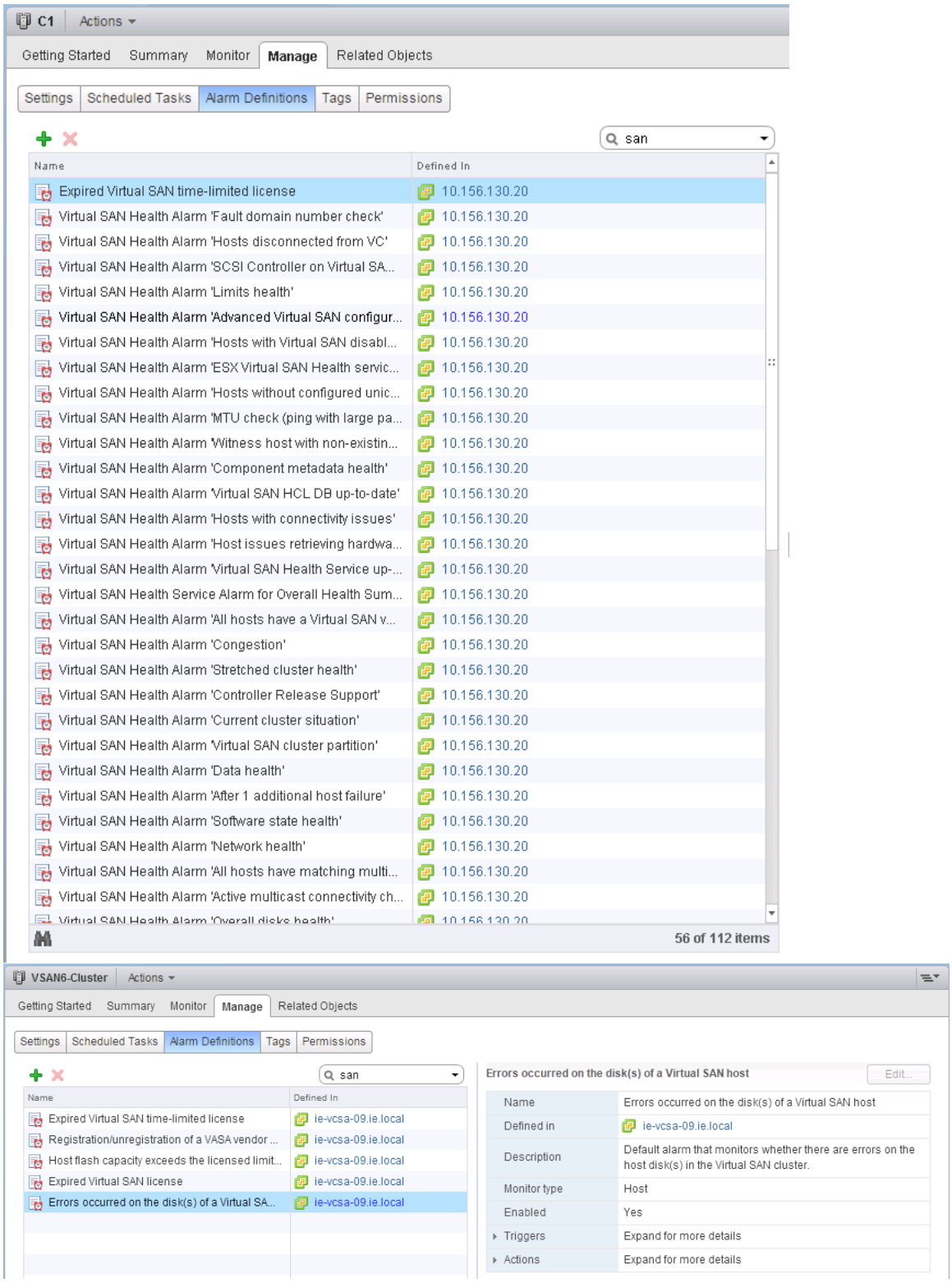


Figure 10.5: Alarm definitions

10.7 Monitor Virtual SAN with VSAN Observer

The VMware VSAN Observer is a performance monitoring and troubleshooting tool for Virtual SAN. The tool is launched from the Ruby vSphere Console (RVC) and can be utilized for monitoring performance statistics for Virtual SAN live mode or offline. When running in live mode, a web browser can be pointed at vCenter Server to see live graphs related to the performance of Virtual SAN.

The utility can be used to understand Virtual SAN performance characteristics. The utility is intended to provide deeper insights of Virtual SAN performance characteristics and analytics. VSAN Observer's user interface displays performance information of the following items:

- Host level performance statistics (client stats)
- Statistics of the physical disk layer
- Deep dive physical disks group details
- CPU Usage Statistics
- Consumption of Virtual SAN memory pools
- Physical and In-memory object distribution across Virtual SAN Clusters

The VSAN Observer UI depends on some JavaScript and CSS libraries (JQuery, d3, angular, bootstrap, font-awesome) in order to successfully display the performance statistics and other information. These library files are accessed and loaded over the Internet at runtime when the VSAN Observer page is rendered. The tool requires access to the libraries mentioned above in order to work correctly. This means that the vCenter Server requires access to the Internet. However with a little work beforehand, VSAN Observer can be configured to work in an environment that does not have Internet access.

Further discussion on VSAN Observer is outside the scope of this POC Guide. For those interested in learning more about Virtual SAN Observer, refer to the VMware Virtual SAN Diagnostics and Troubleshooting Reference Manual and Monitoring VMware Virtual SAN with VSAN Observer.

11. Performance Testing

Performance testing is an important part of evaluating any storage solution. Setting up a desirable test environment could be challenging, and customers may do it differently. Customers may also select from a variety of tools to run workloads, or choose to collect data and logs in different ways. These all add complexity to troubleshoot performance issues claimed by customers, and lengthen the evaluation process.

Virtual SAN Performance will depend on what devices are in the hosts (SSD, magnetic disks), on the policy of the virtual machine (how widely the data is spread across the devices), the size of the working set, the type of workload, and so on.

A major factor for virtual machine performance is the virtual hardware: how many virtual SCSI controllers, VMDKs, outstanding I/O and how many vCPUs can be pushing I/O. Use a number of VMs, virtual SCSI controllers and VMDKs for maximum performance.

Virtual SAN's distributed architecture dictates that reasonable performance is achieved when the pooled compute and storage resources in the cluster are well utilized. This usually means a number of VMs each running the specified workload should be distributed in the cluster and run in a consistent manner to deliver aggregated performance. Virtual SAN also depends on VSAN Observer for detailed performance monitoring and analysis, which as a separate tool is easy to become an afterthought of the testing.

11.1 Use VSAN Observer

Virtual SAN ships with a performance-monitoring tool called VSAN Observer. It is accessed via RVC – the Ruby vSphere Console. If you're planning on doing any sort of performance testing, plan on using VSAN Observer to observe what's happening.

Reference VMware Knowledgebase Article 2064240 for getting started with VSAN Observer – *http://kb.vmware.com/kb/2064240*. See detailed information in Monitoring VMware Virtual SAN with VSAN Observer.

11.2 Performance Considerations

There are a number of considerations you should take into account when running performance tests on Virtual SAN.

11.2.1 Single vs. Multiple Workers

Virtual SAN is designed to support good performance when many VMs are distributed and running simultaneously across the hosts in the cluster. Running a single storage test in a single VM won't reflect on the aggregate performance of a Virtual SAN-

enabled cluster. Regardless of what tool you are using – IOmeter, VDbench or something else – plan on using multiple "workers" or I/O processors to multiple virtual disks to get representative results.

11.2.2 Working Set

For the best performance, a virtual machine's working set should be mostly in cache. Care will have to be taken when sizing your Virtual SAN flash to account for all of your virtual machines' working sets residing in cache. A general rule of thumb is to size cache as 10% of your consumed virtual machine storage (not including replica objects). While this is adequate for most workloads, understanding your workload's working set before sizing is a useful exercise. Consider using VMware Infrastructure Planner (VIP) tool to help with this task – http://vip.vmware.com.

11.2.3 Sequential Workloads versus Random Workloads

Sustained sequential write workloads (such as VM cloning operations) run on Virtual SAN will simply fill the cache and future writes will need to wait for the cache to be destaged to the spinning magnetic disk layer before more I/Os can be written to cache, so performance will be a reflection of the spinning disk(s) and not of flash. The same is true for sustained sequential read workflows. If the block is not in cache, it will have to be fetched from spinning disk. Mixed workloads will benefit more from Virtual SAN's caching design.

11.2.4 Outstanding IOs

Most testing tools have a setting for Outstanding IOs, or OIO for short. It shouldn't be set to 1, nor should it be set to match a device queue depth. Consider a setting of between 2 and 8, depending on the number of virtual machines and VMDKs that you plan to run. For a small number of VMs and VMDKs, use 8. For a large number of VMs and VMDKs, consider setting it lower.

11.2.5 Block Size

The block size that you choose is really dependent on the application/workload that you plan to run in your VM. While the block size for a Windows Guest OS varies between 512 bytes and 1MB, the most common block size is 4KB. But if you plan to run SQL Server, or MS Exchange workloads, you may want to pick block sizes appropriate to those applications (they may vary from application version to application version). Since it is unlikely that all of your workloads will use the same block size, consider a number of performance tests with differing, but commonly used, block sizes.

11.2.6 Cache Warm up Considerations

Flash as cache helps performance in two important ways. First, frequently read blocks end up in cache, dramatically improving performance. Second, all writes are committed to cache first, before being efficiently destaged to disks – again, dramatically improving performance.

However, data still has to move back and forth between disks and cache. Most real-world application workloads take a while for cache to "warm up" before achieving steady-state performance.

11.2.7 Number of Magnetic Disk Drives in Hybrid Configurations

In the getting started section, we discuss how disk groups with multiple disks perform better than disk groups with fewer, as there are more disk spindles to destage to as well as more spindles to handle read cache misses. Let's look at a more detailed example around this.

Consider a Virtual SAN environment where you wish to clone a number of VMs to the Virtual SAN datastore. This is a very sequential I/O intensive operation. We may be able to write into the SSD write buffer at approximately 200-300 MB per second. A single magnetic disk can maybe do 100MB per second. So assuming no read operations are taking place at the same time, we would need 2-3 magnetic disks to match the SSD speed for destaging purposes.

Now consider that there might also be some operations going on in parallel. Let's say that we have another Virtual SAN requirement to achieve 2000 read IOPS. Virtual SAN is designed to achieve a 90% read cache hit rate (approximately). That means 10% of all reads are going to be read cache misses; for example, that is 200 IOPS based on our requirement. A single magnetic disk can perhaps achieve somewhere in the region of 100 IOPS. Therefore an additional 2 magnetic disks will be required to meet this requirement.

If we combine the destaging requirements and the read cache misses described above, your Virtual SAN design may need 4 or 5 magnetic disks per disk group to satisfy your workload.

11.2.8 Striping Considerations

One of the VM Storage Policy settings is *NumberOfDiskStripesPerObject.* That allows you to set a stripe width on a VM's VMDK object. While setting disk striping values can sometimes increase performance, that isn't always the case.

As an example, if a given test is cache-friendly (e.g. most of the data is in cache), striping won't impact performance significantly. As another example, if a given VMDK is striped across disks that are busy doing other things, not much performance is gained, and may actually be worse.

11.2.9 Guest File Systems Considerations

Many customers have reported significant differences in performance between different guest file systems and their settings; for example, Windows NTFS and Linux. If you are not getting the performance you expect, consider investigating whether it could be a guest OS file system issue.

11.2.10 Performance during Failure and Rebuild

When Virtual SAN is rebuilding one or more components, application performance can be impacted. For this reason, always check to make sure that Virtual SAN is fully rebuilt and that there are no underlying issues prior to testing performance. Verify there are no rebuilds occurring before testing with the following RVC command, which we discussed earlier:

- **vsan.check_state**
- **vsan.disks_stats**
- **vsan.resync_dashboard**

11.3 Performance Testing Option 1: Virtual SAN Health Check

Virtual SAN Health Check comes with its own Storage Performance Test. This negates the need to deploy additional tools to test the performance of your Virtual SAN environment. To run the storage performance test is quite simple; navigate to the cluster's Monitor tab > Virtual SAN > Proactive Tests, select Storage Performance Test, then click on the Go arrow highlighted below.

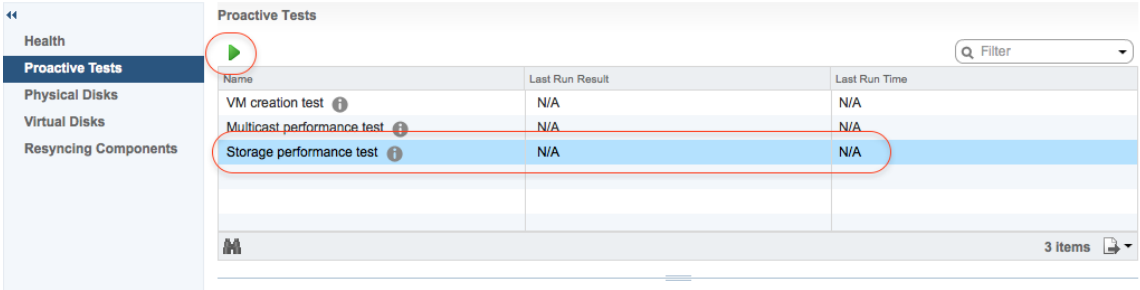


Figure 11.1: Storage Performance Test

A popup is then displayed, showing the duration of the test (default 10 minutes) along with the type of workload that will be run. The user can change this duration, for example, if a burn-in test for a longer period of time is desired.

There are a number of different workloads that can be chosen from the drop-down menu.

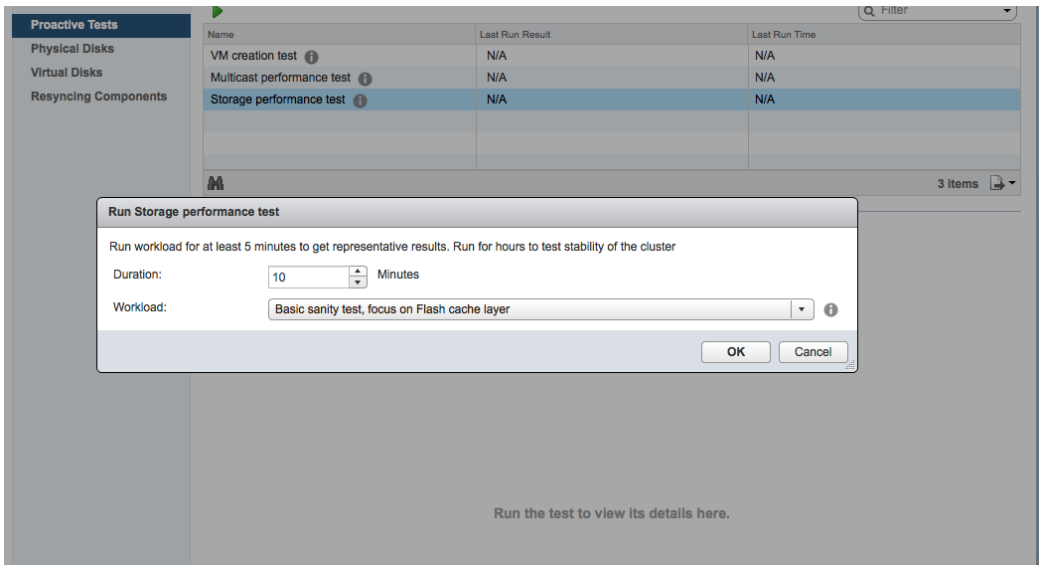


Figure 11.2: Storage Performance Test duration and workload

To learn more about the test that is being run, click on the (i) symbol next to the workload. This will describe the type of workload that the test will initiate.

When the test completed, the Storage Load Test results are displayed, including test name, workload type, IOPS, throughput, average latency and maximum latency. Keep in mind that a sequential write pattern will not benefit from caching, so the results that are shown from this test are basically a reflection of what the capacity layer (in this case, the magnetic disks) can do.

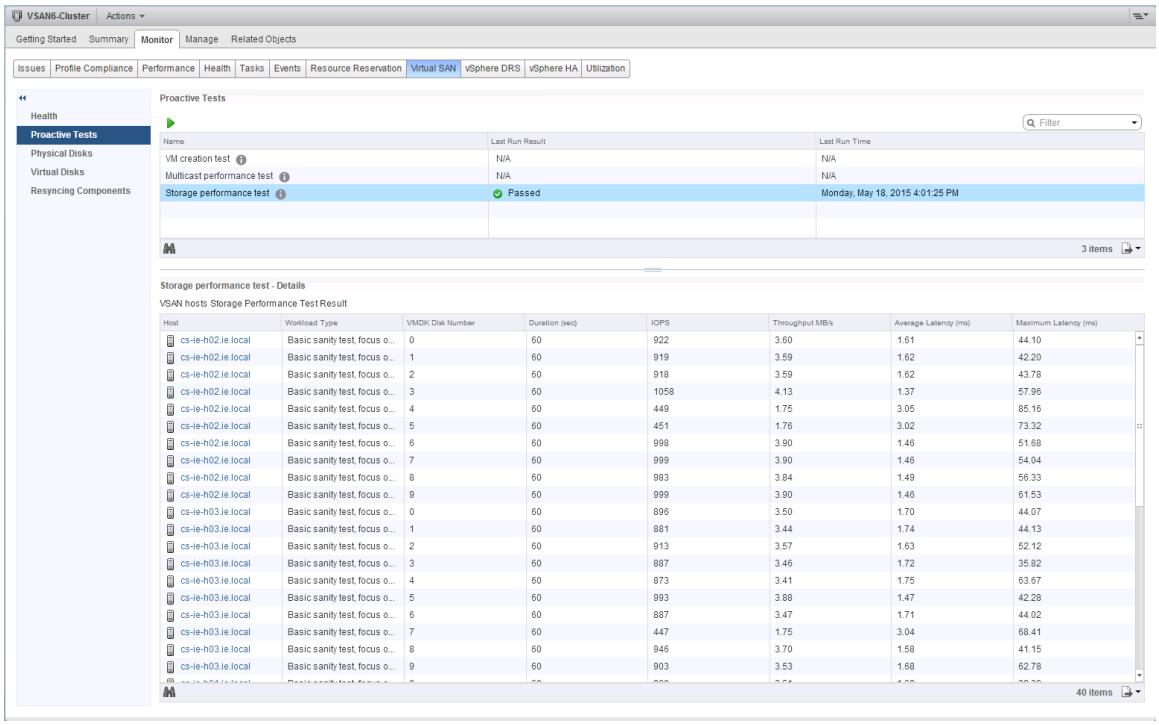


Figure 11.3: Virtual SAN Cluster Storage Load Test results

The proactive test could then be repeated with different workloads

As before, when the test completes, the results are once again displayed. You will notice a major difference in results when the workload can leverage the caching layer versus when it cannot.

11.4 Performance Testing Option 2: HCIbench

In a hyper-converged architecture, each server is intended to support both many application VMs, as well as contribute to the pool of storage available to applications. This is best modeled by invoking many dozens of test VMs, each accessing multiple stored VMDKs. The goal is to simulate a very busy cluster. Unfortunately, popular storage performance testing tools do not directly support this model. As a result performance testing a hyper-converged architecture such as Virtual SAN presents a different set of challenges. To accurately simulate workloads of a production cluster it is best to deploy multiple VMs dispersed across hosts with each VM having multiple disks. In addition, the workload test needs to be run against each VM and disk simultaneously.

To address the challenges of correctly running performance testing in hyper-converged environments, VMware has created a storage performance testing automation tool called HCIbench that automates the use of the popular Vdbench testing tool. Users simply specify the parameters of the test they would like to run, and HCIbench instructs Vdbench what to do on each and every node in the cluster.

HCIbench aims to simplify and accelerate customer Proof of Concept (POC) performance testing in a consistent and controlled way. The tool fully automates the end-to-end process of deploying test VMs, coordinating workload runs, aggregating test results, and collecting necessary data for troubleshooting purposes. Evaluators choose the profiles they are interested in; HCIbench does the rest quickly and easily.

This section provides an overview and recommendations for successfully using HCIbench. For complete documentation and use procedures, refer to the HCIbench Installation and User guide which is accessible from the download directory.

11.4.1 Where to Get HCIbench

HCIbench and complete documentation can be downloaded from the following location: HCIbench Automated Testing Tool.

This tool is provided free of charge and with no restrictions. Support will be provided solely on a best-effort basis as time and resources allow, by the VMware Virtual SAN Community Forum.

11.4.2 Deploying HCIbench

Step 1 – Deploy the OVA

To get started, you deploy a single HCIbench appliance called *HCIbench.ova*. The process for deploying the HCIbench OVA is no different from deploying any other OVA.

Step 2 – HCIbench Configuration

After deployment, navigate to http://Controller_VM_IP:8080/ to start configuration and kick off the test.

There are three main sections in this configuration file:

- vSphere Environment Information

  In this section, all the parameters are required except for the **Network Name** field. You must provide the vSphere environment information where the Virtual SAN Cluster is configured, including vCenter IP address, vCenter credential, name of the datacenter, name of the Virtual SAN Cluster, and name of the Datastore.

  - The **Network Name** parameter defines which network the Vdbench Guest VMs should use. The default value is VM Network.
  - If DHCP serviceses are not available, the **Enable DHCP Service on the Network** parameter allows user to enable DHCP service on the network which "HCIBench Internal Network" mapped on.
  - The **Datastore Name** parameter specifies the datastores to be tested. All VM data will be deployed on this datastore. For the purposes of this guide the Virtual SAN datastore should be specified. Testing multiple datastores in parallel is also supported. You can enter the datastore names, one per line. In this case, virtual machines are distributed evenly across the datastores. For example, if you enter two datastores and 100 virtual machines, 50 virtual machines will be deployed on each datastore.

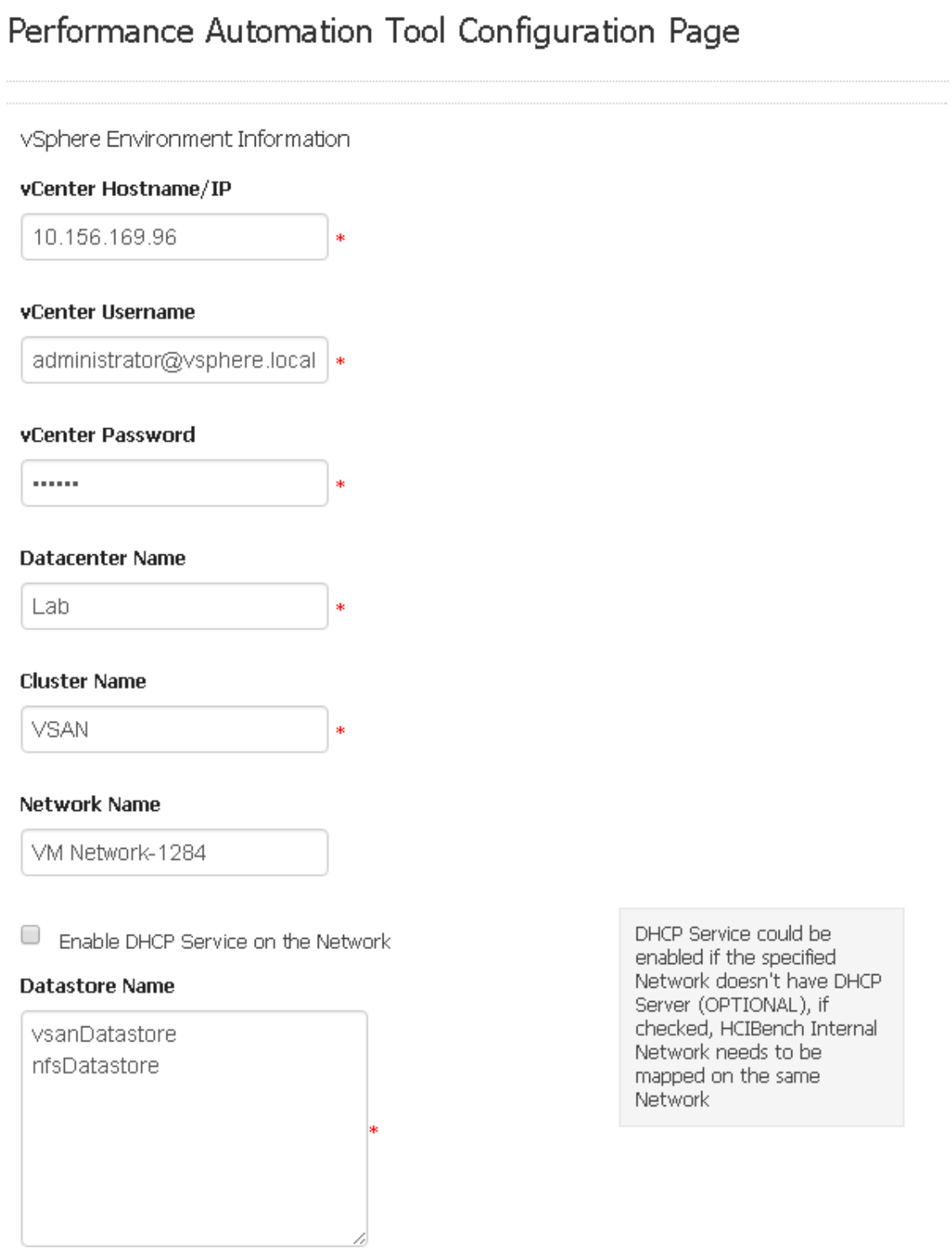


Figure 11.4: Performance Automation Tool Configuration

Step 3 – Virtual SAN Cluster Hosts Information

Configuring the Cluster Hosts information is optional. If this parameter is lift unchecked HCIbench will create a VDbench Guest VM, then clone it to all hosts in the

Virtual SAN Cluster in a round-robin fashion. The naming convention of Vdbench Guest VMs deployed in this mode is “vdbench-vc-<*DATASTORE_NAME*>-<#>”.

If this option is checked, each hosts you wish to deploy HCIbench guest VMs on must be manually added to the Hosts section. As a best practice it is recommended to leave the Cluster host information parameter unchecked and let HCIbench evenly distribute virutal machines on each host.

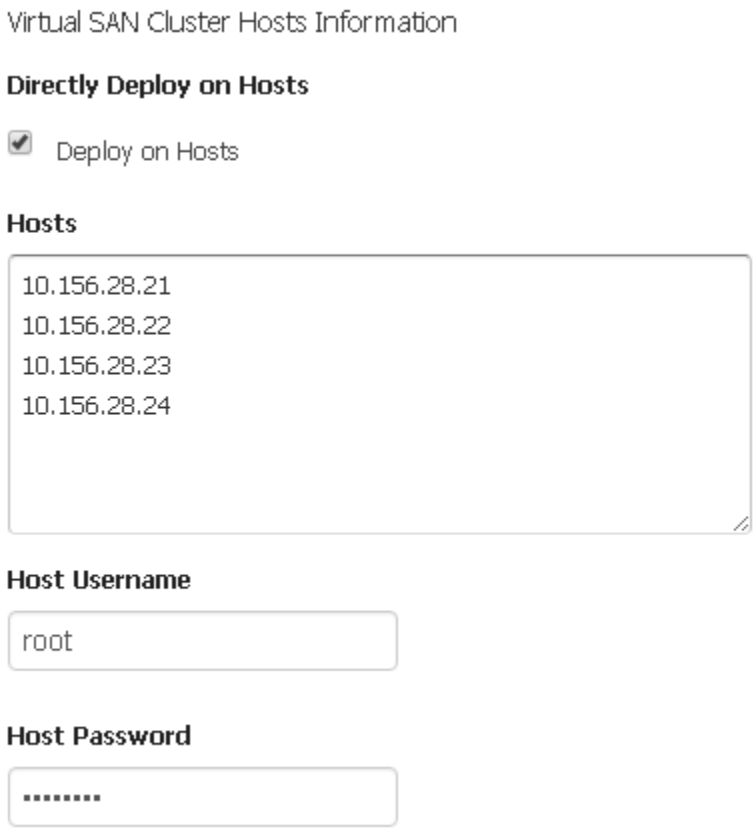


Figure 11.5: Virtual SAN Cluster Hosts Information

Step 4 - VDbench Guest VM Specification

In this section, the only required parameter is **Number of VMs** that specifies the total number of Vdbench Guest VMs to be deployed for testing. If you enter multiple datastores, these VMs are deployed evenly on the datastores. The **Number of Data Disk** and **Size of Data Disk** parameters are optional:

- The **Number of Data Disk** parameter specifies how many VMDKs to be tested are added to each Vdbench Guest VM.
- The **Size of Data Disk** parameter specifies the size (GB) of each VMDK to be tested. The total number of simulated workload instances is **Number of VM** * (times) **Number of Data Disk**.

The default value of both parameters is 10.

NOTE: Prior to setting the number and size of each data disk careful consideration should be given to ensure that there is sufficeent compute and storage resrouces to supporte the target workload. In addition, the cumulative size of all test VMs should not exceed the size of cache available on the cluster as a whole. You should take a carefule sizing exercise to make sure there is sufficient compute and storage resoruces to support the target level of workload instances.

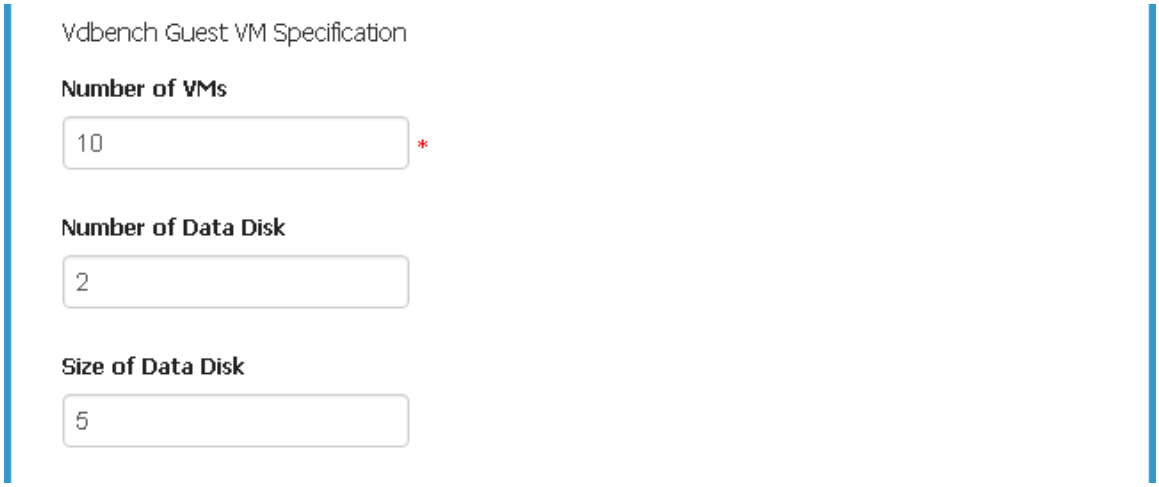


Figure 11.5: Vdbench Guest VM Specification

Step 5 – Download and add vdbench zip file, and add parameter file

Once this is done, users need to provide access to the ***vdbench*** tool. Due to licensing issues, we are not allowed to distribute the vdbench benchmarking tool, so it needs to be downloaded from Oracle if you do not have it already. There is a link provided to the Oracle website to down the vdbench zip file, but you will need to have an account on Oracle's site to access it. Once the vdbench zip file has been downloaded locally, you must then uploaded to the appliance. The next part of the setup is to generate a vdbench parameter file, which has information such as I/O size, R/W ratio and whether the I/O should be random or sequential in nature. You should also state how long you want the test to run (3600 seconds = 1 hour below), as well as whether you want to *dd* the storage first (initialize it). Finally, decide if you want the benchmark VMs cleaned up once the test completes. Save the configuration. To make sure that everything is OK, run the validate test. This will verify that all the configuration parameters are correct, and will state whether it is OK to start the test.

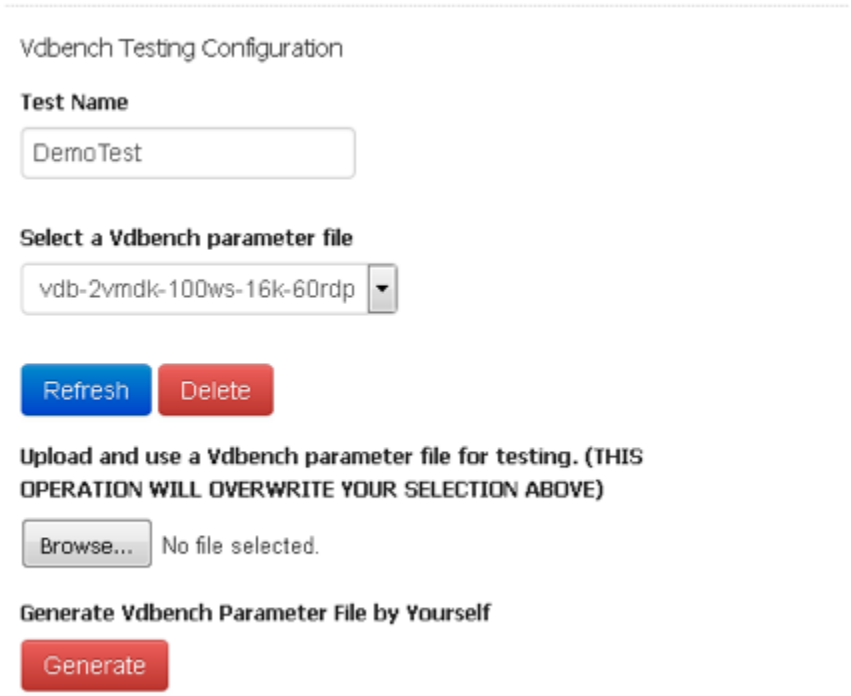


Figure 11.6: Vdbench Testing Configuration

11.4.3 Considerations for Defining Test Workloads

Working set

Working set is one of the most important factors for correctly running performance test and obtaining accurate results. For the best performance, a virtual machine's working set should be mostly in cache. Care will have to be taken when sizing your Virtual SAN flash to account for all of your virtual machines' working sets residing in cache. A general rule of thumb is to size cache as 10% of your consumed virtual machine storage (not including replica objects). While this is adequate for most workloads, understanding your workload's working set before sizing is a useful exercise. Consider using VMware Infrastructure Planner (VIP) tool to help with this task – http://vip.vmware.com.

The following process is an example of sizing an appropriate working set for performance testing with HCIbench. Consider a four node cluster with one 400GB SSD per node. This gives the cluster a total cache size of 1.6TB. The total cache available in Virtual SAN is split 70% for read cache and 30% for write cache. This gives the cluster in our example 1120GB of available read cache and 480GB of available write cache. In order to correctly fit the HCIbench within the available the total capacity of all VMDKs should not exceed 1,120GB.

Designing a test scenario with 4 VMs per host, each VM having 5 X 10GB VMDKs, resulting in a total size of 800GB. This will allow the test working set to fit within cache. The default setting for both the number and size of data disks is 10. This value should. If the total of the **Size of Data Disk** parameter should exceed the total cache size of the cluster. The total size of data disk is the **Number of VM** * (times) **Number of Data Disk**.

Sequential workloads versus random workloads

Before doing performance tests it is important to understand the performance characteristics of the production workload to be tested. Different applications have different performance characteristics. Understanding these characteristics is crucial to successful performance testing. When it is not possible to test with the actual application or application specific testing tool it is important to design a test which matches the production workload as closely as possible. Different workload types will perform differently on Virtual SAN.

Sustained sequential write workloads (such as VM cloning operations) run on Virtual SAN will simply fill the cache and future writes will need to wait for the cache to be destaged to the spinning magnetic disk layer before more I/Os can be written to cache, so performance will be a reflection of the spinning disk(s) and not of flash. The same is true for sustained sequential read workflows. If the block is not in cache, it will have to be fetched from spinning disk. Mixed workloads will benefit more from Virtual SAN's caching design.

HCIbench allows you to change the percentage read and the percentage random parameters. As a starting point it is recommended to set the percentage read parameter to 70 and the percentage random parameter to 30%.

Initializing Storage

During configuration of the workload the recommendation is to select the option to initialize storage. This option will zero the disks for each VM being used in the test, helping to alleviate a first write penalty during the performance testing phase.

Test Run Considerations

As frequently read blocks end up in cache, read performance will improve. In a production environment active blocks will already be in cache. When running any kind of performance testing it is important to keep this in mind. As a best practice performance tests should include at least a 15 minute warm up period. Also keep in mind that the longer testing runs the more accurate the results will be. In addition to the cache warming period HCIbench tests should be configured to for at least an hour.

Results

After the Vdbench testing is completed, the test results are collected from all Vdbench instances in the test VMs. And you can view the results at http://Controller_VM_IP/results in a web browser. You can find all of the original result files produced by Vdbench instances inside the subdirectory corresponding to a test run. In addition to the text files, there is another subdirectory named iotest-vdbench-<*VM#*>vm inside, which is the statistics directory generated by Virtual SAN Observer. Virtual SAN performance data can be viewed by opening the stats.html file within the test directory.

12. Testing Hardware Failures

12.1 Understanding Expected Behavior

When doing failure testing with Virtual SAN, it is important to understand the expected behavior for different failure scenarios. You should compare the results of your test to what is expected. The previous section should be read to understand expected failure behaviors.

12.2 Important: Test one Thing at a Time

By default, virtual machines are deployed on Virtual SAN with the ability to tolerate one failure. If you do not wait for the first failure to be resolved, and then try to test another failure, you will have introduced two failures to the cluster. Virtual Machines will not be able to tolerate the second failure and will become inaccessible.

12.3 VM Behavior when Multiple Failures Encountered

Previously we discussed VM operational states and availability. To recap, a VM remains accessible when a full mirror copy of the objects are available, as well as greater than 50% of the components that make up the VM; the witnesses are there to assist with the latter requirement.

Let's talk a little about VM behavior when there are more failures in the cluster than the *NumberOfFailuresToTolerate* setting in the policy associated with the VM.

12.3.1 VM Powered on and VM Home Namespace Object Goes Inaccessible

If a running VM has its VM Home Namespace object go inaccessible due to failures in the cluster, a number of different things may happen. Once the VM is powered off, it will be marked "inaccessible" in the vSphere web client UI. There can also be other side effects, such as the VM getting renamed in the UI to its ".vmx" path rather than VM name, or the VM being marked "orphaned".

12.3.2 VM Powered on and Disk Object Goes Inaccessible

If a running VM has one of its disk objects go inaccessible, the VM will keep running, but its VMDK's I/O is stalled. Typically, the Guest OS will eventually time out I/O. Some operating systems may crash when this occurs. Other operating systems, for example some Linux distributions, may downgrade the filesystems on the impacted VMDK to read-only. The Guest OS behavior, and even the VM behavior is not Virtual SAN specific. It can also be seen on VMs running on traditional storage when the ESXi host suffers an *APD* (All Paths Down).

Once the VM becomes accessible again, the status should resolve, and things go back to normal. Of course, data remains intact during these scenarios.

12.4 What Happens when a Server Fails or is Rebooted?

A host failure can occur in a number of ways. It could be a crash, or it could be a network issue (which is discussed in more detail in the next section). However, it could also be something as simple as a reboot, and that the host will be back online when the reboot process completes. Once again, Virtual SAN needs to be able to handle all of these events.

If there are active components of an object residing on the host that is detected to be failed (due to any of the stated reasons) then those components are marked as ABSENT. I/O flow to the object is restored within 5-7 seconds by removing the ABSENT component from the active set of components in the object.

The ABSENT state is chosen rather than the DEGRADED state because in many cases a host failure is a temporary condition. A host might be configured to auto-reboot after a crash, or the host's power cable was inadvertently removed, but plugged back in immediately. Virtual SAN is designed to allow enough time for a host to reboot before starting rebuilds on other hosts so as not to waste resources. Because Virtual SAN cannot tell if this is a host failure, a network disconnect or a host reboot, the 60-minute timer is once again started. If the timer expires, and the host has not rejoined the cluster, a rebuild of components on the remaining hosts in the cluster commences.

If a host fails, or is rebooted, this event will trigger a "Host connection and power state" alarm, and if vSphere HA is enabled on the cluster, it will also cause a" vSphere HA host status" alarm and a "Host cannot communicate with all other nodes in the Virtual SAN Enabled Cluster" message.

If *NumberOfFailuresToTolerate=1* or higher in the VM Storage Policy, and an ESXi host goes down, VMs not running on the failed host continue to run as normal. If any VMs with that policy were running on the failed host, they will get restarted on one of the other ESXi hosts in the cluster by vSphere HA, as long as it is configured on the cluster.

Caution: If VMs are configured in such a way as to not tolerate failures, (*NumberOfFailuresToTolerate=0*), a VM that has components on the failing host will become inaccessible through the vSphere web client UI.

12.5 Simulate Host Failure without vSphere HA

Without vSphere HA, any virtual machines running on the host that fails will not be automatically started elsewhere in the cluster, even though the storage backing the virtual machine in question is unaffected.

Let's take an example where a VM is running on a host (cs-ie-h02.ie.local).

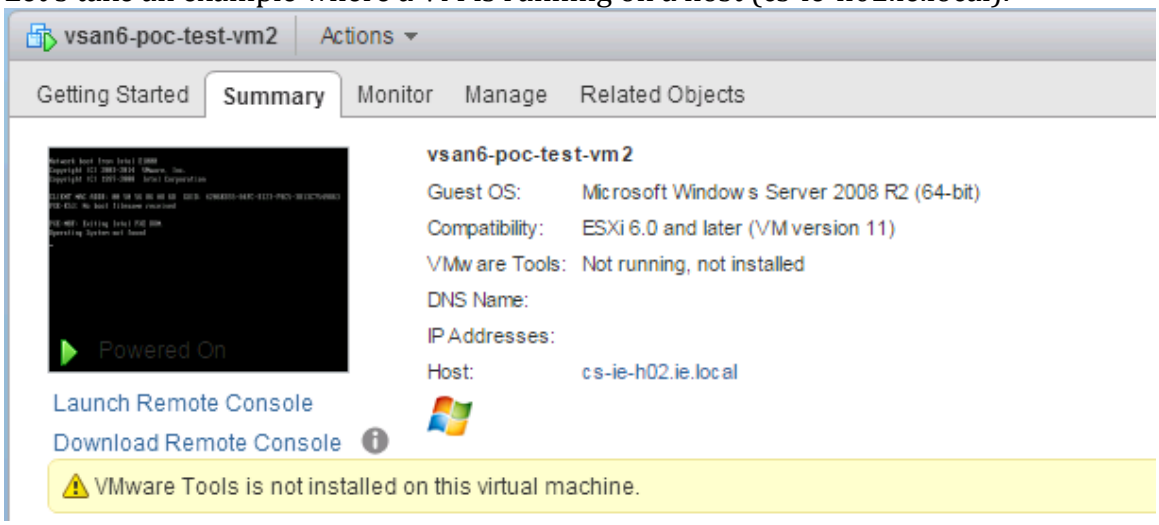


Figure 12.1: host failure without vSphere HA

It would also be a good test if this VM also had components located on the local storage of this host. However it does not matter if it does not as the test will still highlight the benefits of vSphere HA.

Next, the host is rebooted:

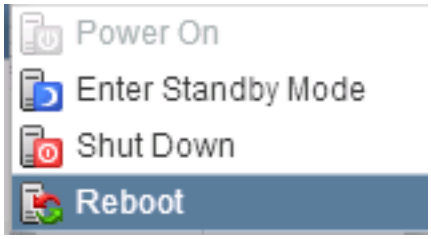


Figure 12.2: Reboot the host

As expected, the host is not responding in vCenter, and the VM becomes disconnected. The VM will remain in a disconnected state until the ESXi host has fully rebooted, as there is no vSphere HA enabled on the cluster, so the VM cannot be restarted on another host in the cluster.

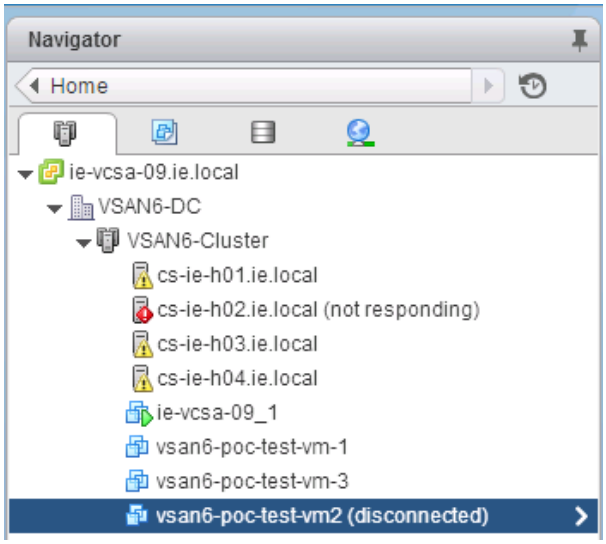


Figure 12.3: ESXi host not responding, VM disconnected

If you now examine the policies of the VM, you will see that it is non-compliant. You can also see the reason why in the lower part of the screen. This VM should be able to tolerate one failure, but due to the failure currently in the cluster (for example: one ESXi host is currently rebooting), this VM cannot tolerate another failure, thus it is non-compliant with its policy.

What can be deduced from this is that not only was the VM's compute running on the host which was rebooted, but that it also had some components residing on the storage of the host that was rebooted. We can confirm this when the host fully reboots.

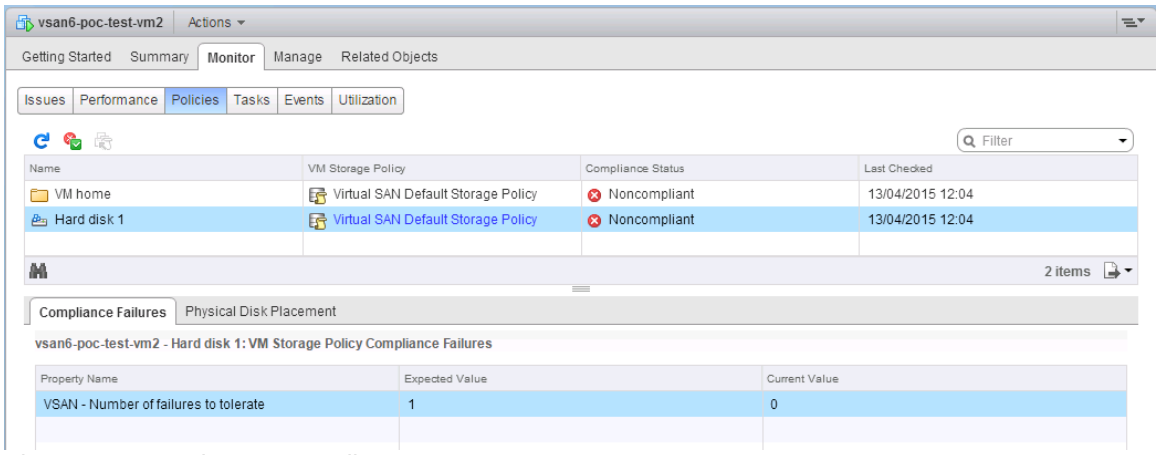


Figure 12.4: VM is non-compliant

Once the ESXi host has rebooted, we see that the VM is no longer disconnected. However it is left in a powered off state.

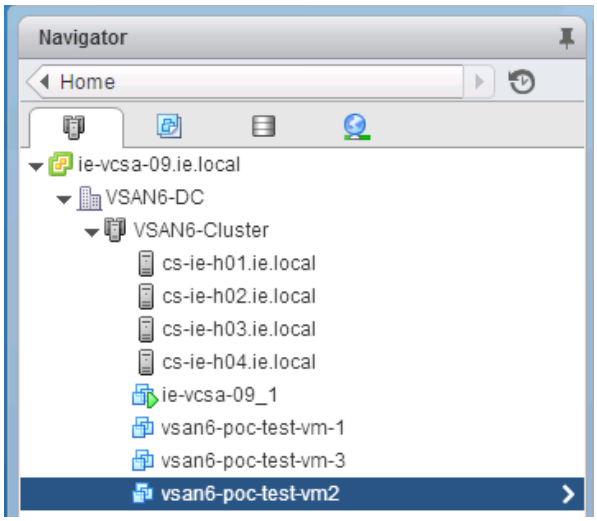


Figure 12.5: ESXi host rebooted, VM powered off

And as mentioned previously, if the physical disk placement is examined, we can clearly see that the storage on the host that was rebooted, cs-ie-h02.ie.local, was used to store components belonging to the VM.

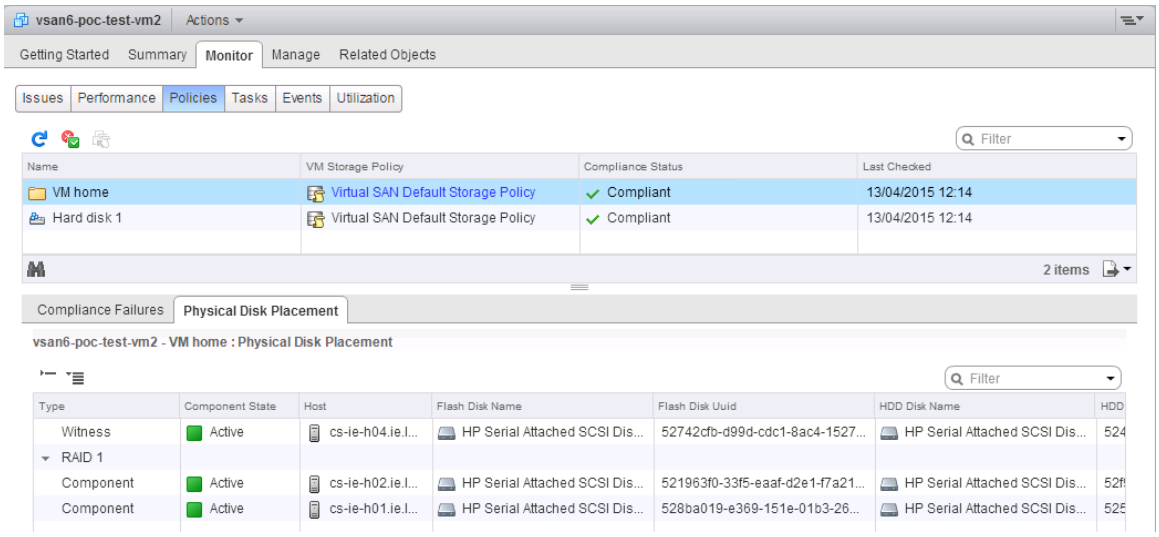


Figure 12.6: Components on host that was rebooted

12.6 Simulate Host Failure with vSphere HA

Let's now repeat the same scenario, but with vSphere HA enabled on the cluster. First, power on the VM from the last test.

Next, select the cluster object, and navigate to the Manage tab, then Settings > Services > vSphere HA. vSphere HA is turned off currently.

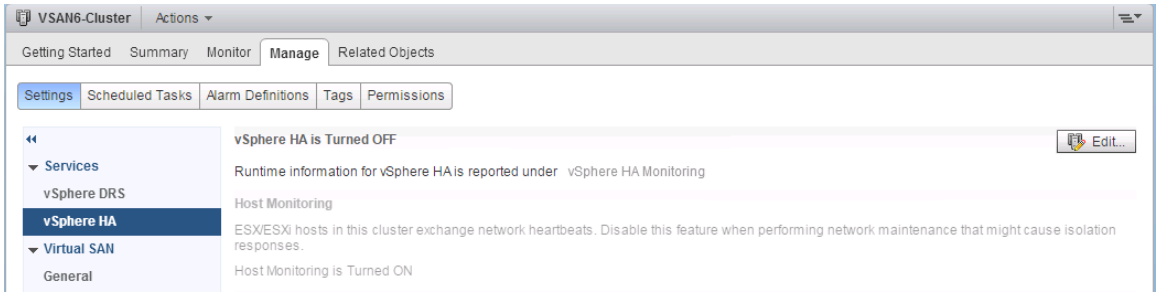


Figure 12.7: vSphere HA is turned off

Click on the "Edit" button to enable vSphere HA. When the wizard pops up, click on the "Turn on vSphere HA" checkbox as shown below, then click OK.

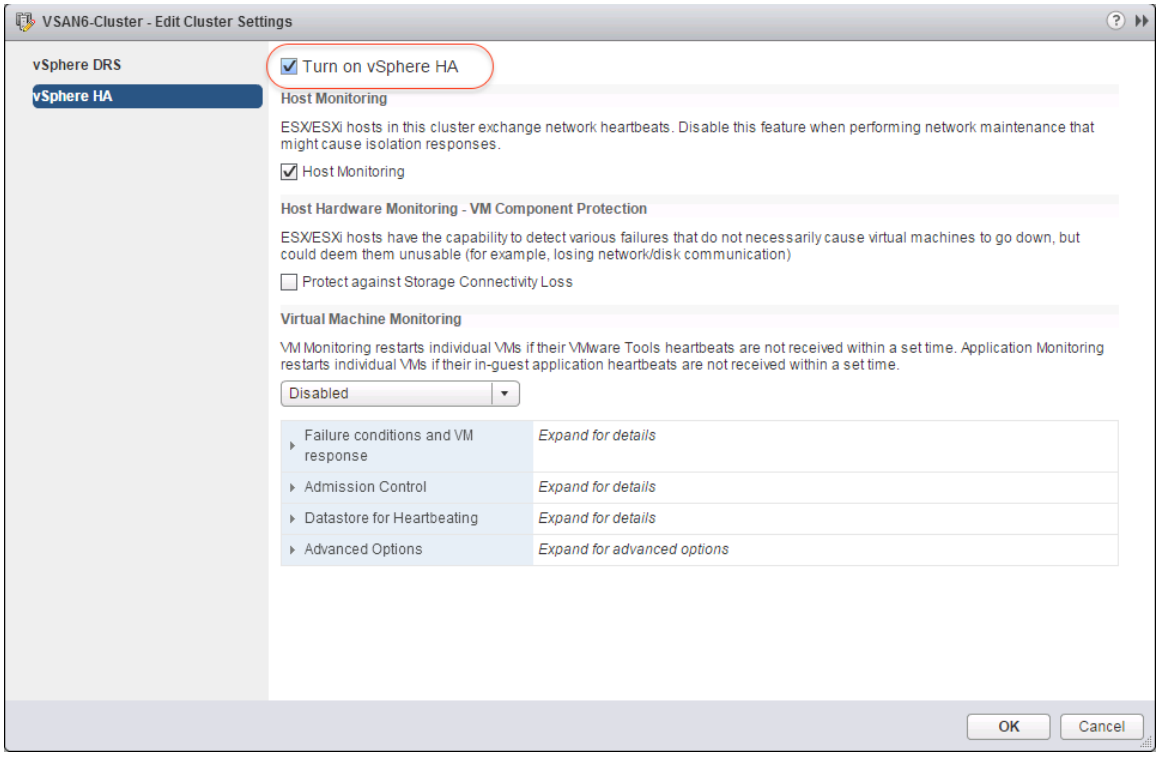


Figure 12.8: Turn on vSphere HA

This will launch a number of tasks on each ode in the cluster. These can be monitored via the Monitor > Tasks view. When the configuring of vSphere HA tasks complete, select the cluster object, then the Summary tab, then the vSphere HA

window and ensure it is configured and monitoring. The cluster should now have Virtual SAN, DRS and vSphere HA enabled.

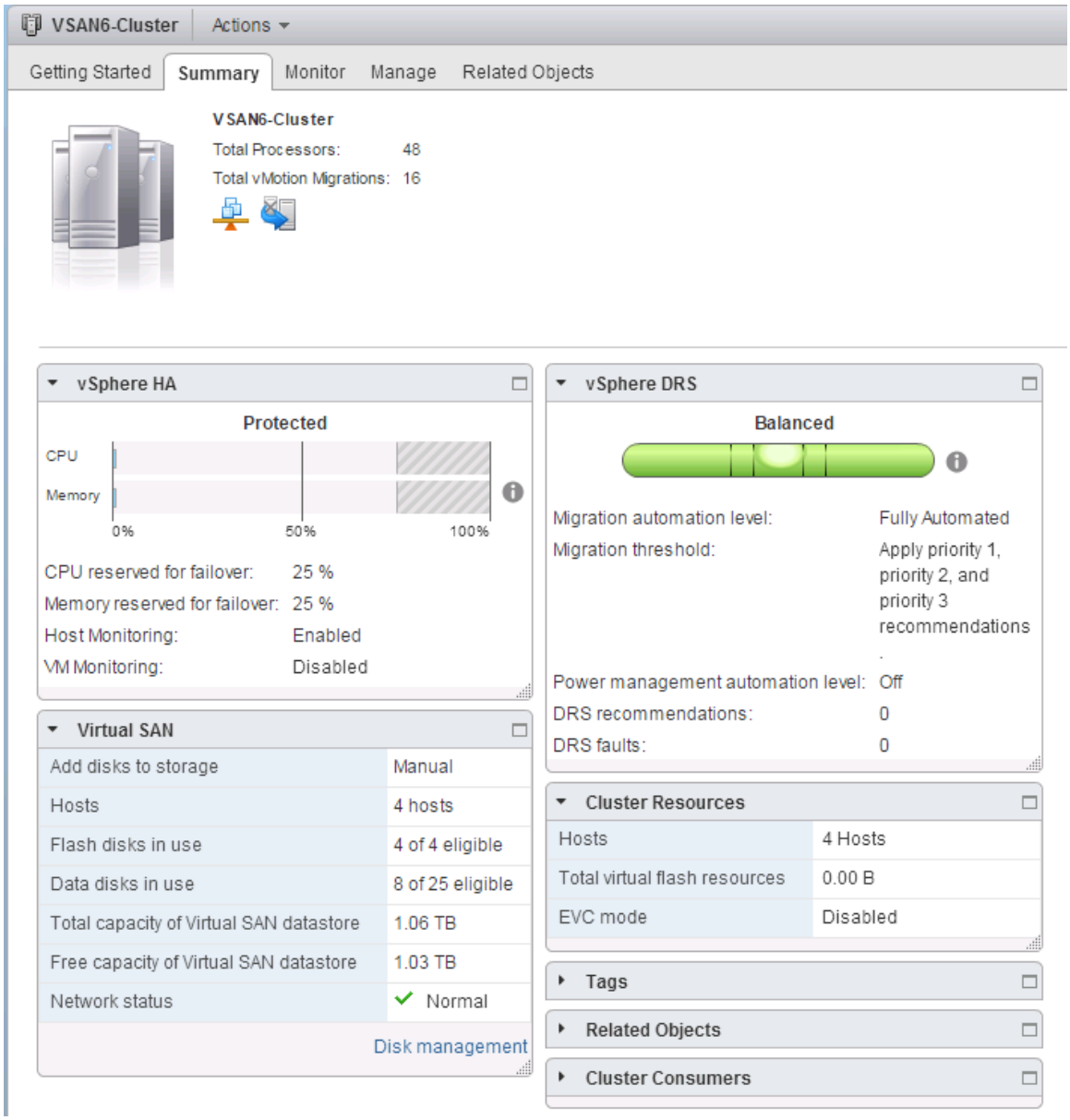


Figure 12.9: Virtual SAN, DRS and vSphere HA enabled

Verify that the test VM is still residing on host cs-ie-h02.ie.local. Now repeat the same test as before by rebooting host cs-ie-h02.ie.local and examine the differences with vSphere HA enabled.

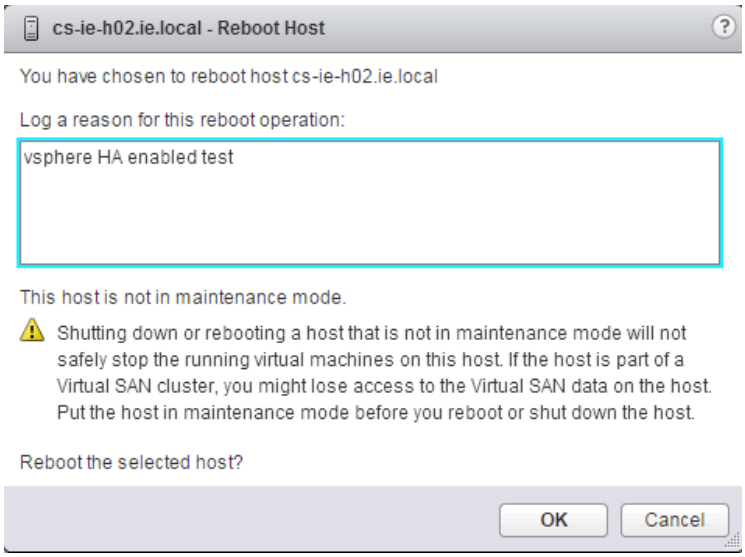


Figure 12.10: Reboot the host, this time with vSphere HA enabled

On this occasion, a number of HA related events should be displayed on the Summary tab of the host being rebooted (you may need to refresh the web client to see these):

Figure 12.11: vSphere HA messages

However, rather than the VM becoming disconnected for the duration of the host reboot like was seen in the last test, the VM in instead restarted on another host, in this case cs-ie-h03.ie.local.

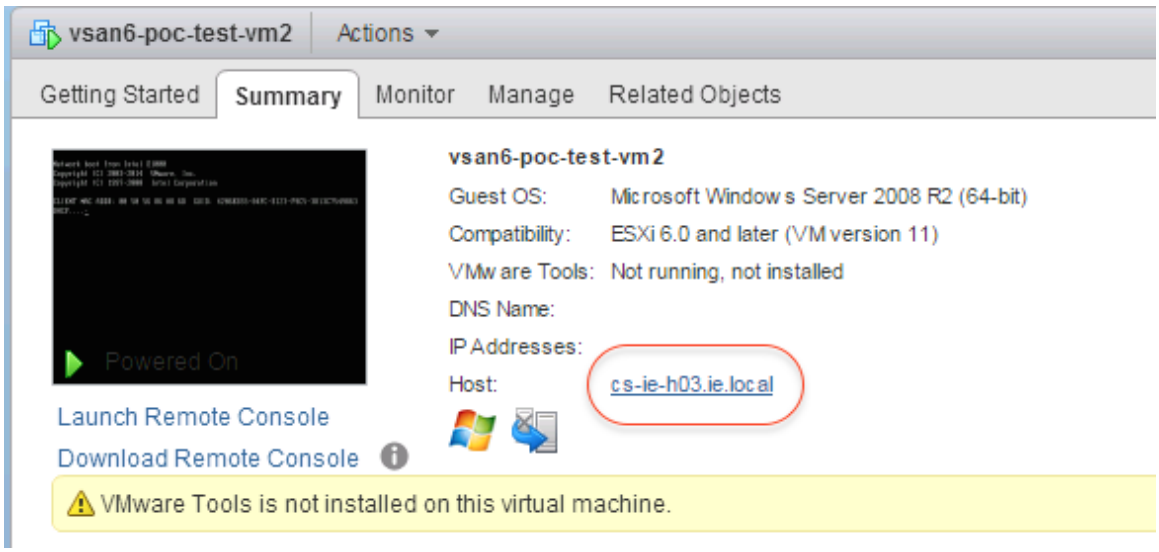


Figure 12.12: VM restarted on a different host

If you remember earlier we stated that there were some components belonging to the objects of this VM residing on the local storage of the host that was rebooted. These components now show up as “Absent” in the VM > Monitor > policies > Physical Disk Placement view as shown below.

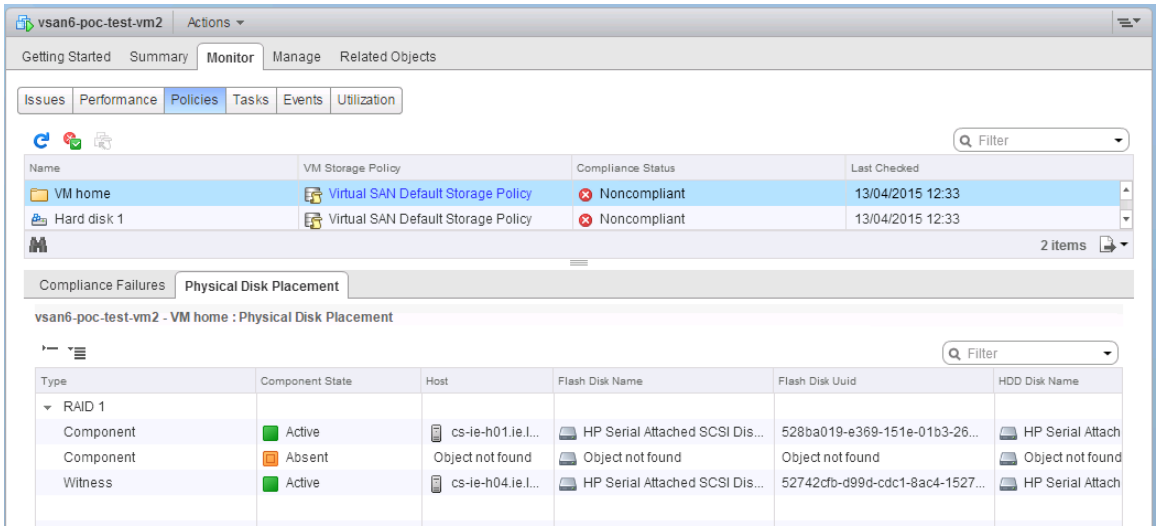


Figure 12.13: Absent components

However, once the ESXi host completes rebooting, assuming it is within 60 minutes, these components will be rediscovered, resynchronized and placed back in an Active state.

Should the host be disconnected for longer than 60 minutes (the CLOMD timeout delay default value), the “Absent” components will be rebuilt elsewhere in the cluster.

12.7 Disk is Pulled Unexpectedly from ESXi Host

When a magnetic disk is pulled from an ESXi hosts that is using it to contribute storage to Virtual SAN without first decommissioning the disk, all the Virtual SAN components residing on the disk go ABSENT and are inaccessible.

The ABSENT state is chosen over DEGRADED because Virtual SAN knows the disk is not lost, but rather just removed. If the disk gets put back into the server before the 60-minute timeout, no harm is done and Virtual SAN syncs it back up. In this scenario, Virtual SAN is back up with full redundancy without wasting resources on an expensive rebuild.

12.7.1 Expected Behaviors

- If the VM has a policy that includes *NumberOfFailuresToTolerate=1* or greater, the VM's objects will still be accessible from another ESXi host in the Virtual SAN Cluster.
- The disk state is marked as ABSENT and can be verified via vSphere web client UI.
- At this point, all in-flight I/O is halted while Virtual SAN reevaluates the availability of the object (e.g. VM Home Namespace or VMDK) without the failed component as part of the active set of components.
- If Virtual SAN concludes that the object is still available (based on a full mirror copy and greater than 50% of the components being available), all in-flight I/O is restarted.
- The typical time from physical removal of the disk, Virtual SAN processing this event, marking the component ABSENT halting and restoring I/O flow is approximately 5-7 seconds.
- If the same disk is placed back on the same host within 60 minutes, no new components will be re-built.
- If 60 minutes passes, and the original disk has not been reinserted in the host, components on the removed disk will be built elsewhere in the cluster, if capacity is available, including any newly inserted disks claimed by Virtual SAN.
- If the VM Storage Policy has *NumberOfFailuresToTolerate=0*, the VMDK will be inaccessible if one of the VMDK components (think one component of a stripe or a full mirror) resides on the removed disk. To restore the VMDK, the same disk has to be placed back in the ESXi host. There is no other option for recovering the VMDK.

12.8 SSD is Pulled Unexpectedly from ESXi Host

When a solid-state disk drive is pulled without decommissioning it, all the Virtual SAN components residing in that disk group go ABSENT and are inaccessible. In other words, if an SSD is removed, it will appear as a removal of the SSD as well as all associated magnetic disks backing the SSD from a Virtual SAN perspective.

12.8.1 Expected Behaviors

- If the VM has a policy that includes *NumberOfFailuresToTolerate=1* or greater, the VM's objects will still be accessible.
- Disk group and the disks under the disk group states will be marked as ABSENT and can be verified via the vSphere web client UI.
- At this point, all in-flight I/O is halted while Virtual SAN reevaluates the availability of the objects without the failed component(s) as part of the active set of components.
- If Virtual SAN concludes that the object is still available (based on a full mirror copy and greater than 50% of components being available), all in-flight I/O is restarted.
- The typical time from physical removal of the disk, Virtual SAN processing this event, marking the components ABSENT halting and restoring I/O flow is approximately 5-7 seconds.
- When the same SSD is placed back on the same host within 60 minutes, no new objects will be re-built.
- When the timeout expires (default 60 minutes), components on the impacted disk group will be rebuilt elsewhere in the cluster, if capacity is available.
- If the VM Storage Policy has *NumberOfFailuresToTolerate=0*, the VMDK will be inaccessible if one of the VMDK components (think one component of a stripe or a full mirror) exists on disk group whom the pulled SSD belongs to. To restore the VMDK, the same SSD has to be placed back in the ESXi host. There is no option to recover the VMDK.

12.9 What Happens When a Disk Fails?

If a disk drive has an unrecoverable error, Virtual SAN marks the disk as DEGRADED as the failure is permanent.

12.9.1 Expected Behaviors

- If the VM has a policy that includes *NumberOfFailuresToTolerate=1* or greater, the VM's objects will still be accessible.
- The disk state is marked as DEGRADED and can be verified via vSphere web client UI.
- At this point, all in-flight I/O is halted while Virtual SAN reevaluates the availability of the object without the failed component as part of the active set of components.
- If Virtual SAN concludes that the object is still available (based on a full mirror copy and greater than 50% of components being available), all in-flight I/O is restarted.
- The typical time from physical removal of the drive, Virtual SAN processing this event, marking the component DEGRADED halting and restoring I/O flow is approximately 5-7 seconds.
- Virtual SAN now looks for any hosts and disks that can satisfy the object requirements. This includes adequate free disk space and placement rules (e.g. 2 mirrors may not share the same host). If such resources are found, Virtual SAN will create new components on there and start the recovery process immediately.
- If the VM Storage Policy has *NumberOfFailuresToTolerate=0*, the VMDK will be inaccessible if one of the VMDK components (think one component of a stripe) exists on the pulled disk. This will require a restore of the VM from a known good backup.

12.10 What Happens When an SSD Fails?

An SSD failure follows a similar sequence of events to that of a disk failure with one major difference; Virtual SAN will mark the entire disk group as DEGRADED. Virtual SAN marks the SSD and all disks in the disk group as DEGRADED as the failure is permanent (disk is offline, no longer visible, and others).

12.10.1 Expected Behaviors

- If the VM has a policy that includes *NumberOfFailuresToTolerate=1* or greater, the VM's objects will still be accessible from another ESXi host in the Virtual SAN Cluster.
- Disk group and the disks under the disk group states will be marked as DEGRADED and can be verified via the vSphere web client UI.
- At this point, all in-flight I/O is halted while Virtual SAN reevaluates the availability of the objects without the failed component(s) as part of the active set of components.
- If Virtual SAN concludes that the object is still available (based on available full mirror copy and witness), all in-flight I/O is restarted.
- The typical time from physical removal of the drive, Virtual SAN processing this event, marking the component DEGRADED halting and restoring I/O flow is approximately 5-7 seconds.
- Virtual SAN now looks for any hosts and disks that can satisfy the object requirements. This includes adequate free SSD and disk space and placement rules (e.g. 2 mirrors may not share the same hosts). If such resources are found, Virtual SAN will create new components on there and start the recovery process immediately.
- If the VM Storage Policy has *NumberOfFailuresToTolerate=0*, the VMDK will be inaccessible if one of the VMDK components (think one component of a stripe) exists on disk group whom the pulled SSD belongs to. There is no option to recover the VMDK. This may require a restore of the VM from a known good backup.

Warning: Test one thing at a time during the following POC steps. Failure to resolve the previous error before introducing the next error will introduce multiple failures into Virtual SAN which it may not be equipped to deal with, based on the NumberOfFailuresToTolerate setting, which is set to 1 by default.

12.11 Virtual SAN Disk Fault Injection Script for POC Failure Testing

When the Virtual SAN Health Check VIB is installed (installed by default in vSphere 6.0U1), a python script to help with POC disk failure testing is available on all ESXi hosts. The script is called `vsanDiskFaultInjection.pyc` and can be found on the ESXi hosts in the directory `/usr/lib/vmware/vsan/bin`. To display the usage, run the following command:

```
[root@cs-ie-h01:/usr/lib/vmware/Virtual SAN /bin]
python./vsanDiskFaultInjection.pyc -h
Usage:
            injectError.py -t -r error_durationSecs -d deviceName
            injectError.py -p -d deviceName
            injectError.py -c -d deviceName

Options:
  -h, --help            show this help message and exit
  -u                    Inject hot unplug
  -t                    Inject transient error
  -p                    Inject permanent error
  -c                    Clear injected error
  -r ERRORDURATION      Transient error duration in seconds
  -d DEVICENAME, --deviceName=DEVICENAME
[root@cs-ie-h01:/usr/lib/vmware/vsan/bin]
```

Warning: This command should only be used in pre-production environments during a POC. It should not be used in production environments. Using this command to mark disks as failed can have a catastrophic effect on a Virtual SAN Cluster.

Readers should also note that this tool provides the ability to do "hot unplug" of drives, which is similar to the testing that was done with the *hpssacli* command previously. This is an alternative way of creating a similar type of condition. However, in this POC guide, this script is only being used to inject permanent errors.

12.12 Pull Magnetic Disk/Capacity Tier SSD and Replace before Timeout Expires

In this first example, we shall remove a disk from the host using the *vsanDiskFaultInjection.pyc* python script rather than physically removing it from the host.

It should be noted that the same tests can be run by simply removing the disk from the host. If physical access to the host is convenient, literally pulling a disk would test exact physical conditions as opposed to emulating it within software.

Also note that not all I/O controllers support hot unplugging drives. Check the Virtual SAN Compatibility Guide to see if your controller model supports the hot unplug feature.

We will then examine the effect this operation has on Virtual SAN, and virtual machines running on Virtual SAN. We shall then replace the component before the CLOMD timeout delay expires (default 60 minutes), which will mean that no rebuilding activity will occur during this test.

Pick a host with a running VM.

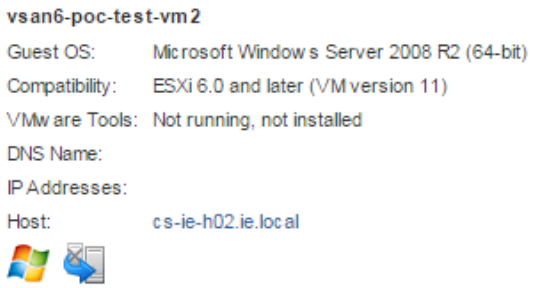


Figure 12.14: Select host with running VM

Next, navigate to the VM's Monitor tab > Policies, select a Hard Disk and then select Physical Disk Placement tab in the lower half of the screen. Identify a Component object. The column that we are most interested in is HDD Disk Name, as it contains the NAA SCSI identifier of the disk. The objective is to remove one of these disks from the host (other columns may be hidden by right clicking on them).

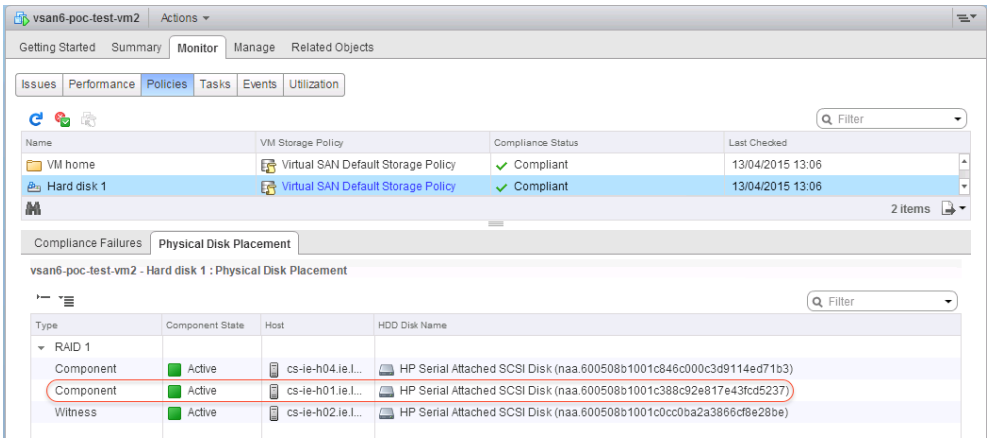


Figure 12.15: Display disk identifiers

From figure 12.15, let us say that we wish to remove the disk containing the component residing on host cs-ie-h01.ie.local. That component resides on physical disk with an NAA ID string of naa.600508b1001c388c92e817e43fcd5237. Make a note of your NAA ID string. Next, SSH into the host with the disk to pull. Inject a hot unplug event using the *vsanDiskFaultInjection.pyc* python script:

```
[root@cs-ie-h01:~] python /usr/lib/vmware/vsan/bin/vsanDiskFaultInjection.pyc -u
-d naa.600508b1001c388c92e817e43fcd5237
Injecting hot unplug on device vmhba1:C0:T4:L0
vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x1
vsish -e set /storage/scsifw/paths/vmhba1:C0:T4:L0/injectError 0x004C0400000002
```

Let's now check out the VM's objects and components and as expected, the component that resided on that disk in host cs-ie-h01 quickly shows up as absent.

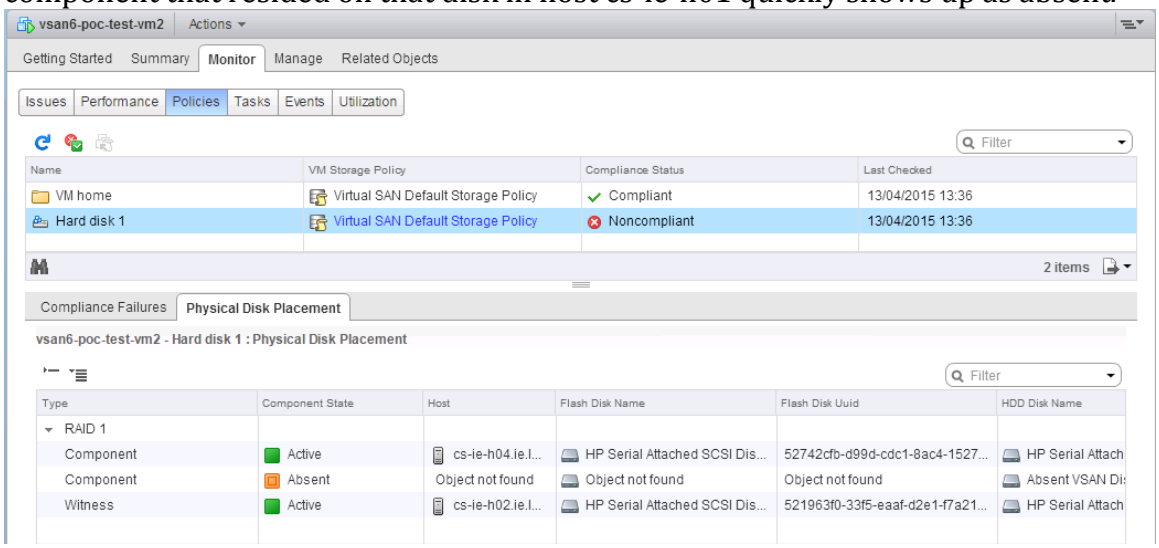


Figure 12.16: Disk Removed, Component Absent

To put the disk drive back in the host, one simply rescans the host for new disks. Navigate to the host > Manage > Storage > Storage Devices and click the rescan button.

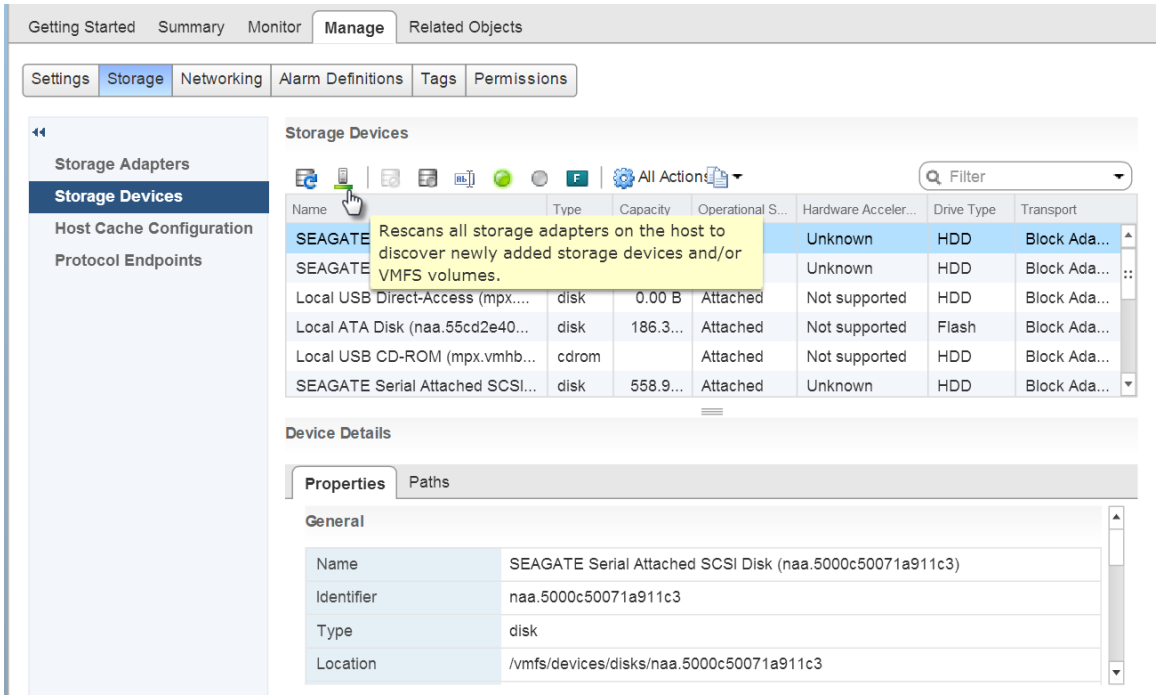


Figure 12.17: Rescan storage adapters

Look at the list of storage devices for the NAA ID that was removed. If for some reason, the disk doesn't return after refreshing the screen, try rescanning the host

again. If it still doesn't appear, reboot the ESXi host. Once the NAA ID is back, clear any hot unplug flags set previously with the *–c* option:

```
[root@cs-ie-h01:~] python /usr/lib/vmware/vsan/bin/vsanDiskFaultInjection.pyc -c
-d naa.600508b1001c388c92e817e43fcd5237
Clearing errors on device vmhba1:C0:T4:L0
vsish -e set /storage/scsifw/paths/vmhba1:C0:T4:L0/injectError 0x00000
vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x00000
```

12.13 Pull Magnetic Disk/Capacity Tier SSD and Do not Replace before Timeout Expires

In this example, we shall remove the magnetic disk from the host, once again using the *vsanDiskFaultInjection.pyc* script. However, this time we shall wait longer than 60 minutes before scanning the HBA for new disks. After 60 minutes, Virtual SAN will rebuild the components on the missing disk elsewhere in cluster.

The same process as before can now be repeated. However this time we shall leave the disk drive removed for more than 60 minutes and see the rebuild activity take place. Begin by identifying the disk on which the component resides.

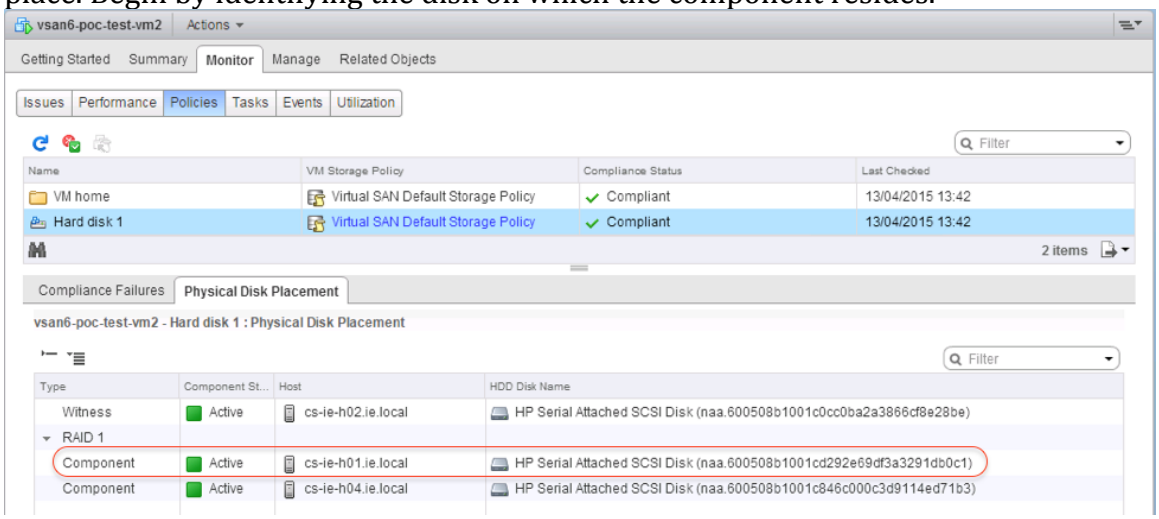


Figure 12.18: Identify NAA id

```
 [root@cs-ie-h01:~] date
 Mon Dec 14 13:36:02 UTC 2015
 [root@cs-ie-h01:~] python /usr/lib/vmware/vsan/bin/vsanDiskFaultInjection.pyc
-u -d naa.600508b1001c388c92e817e43fcd5237
Injecting hot unplug on device vmhba1:C0:T4:L0
vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x1
vsish -e set /storage/scsifw/paths/vmhba1:C0:T4:L0/injectError 0x004C0400000002
```

At this point, we can once again see that the component has gone *absent*. After 60 minutes have elapsed, the component should now be rebuilt.

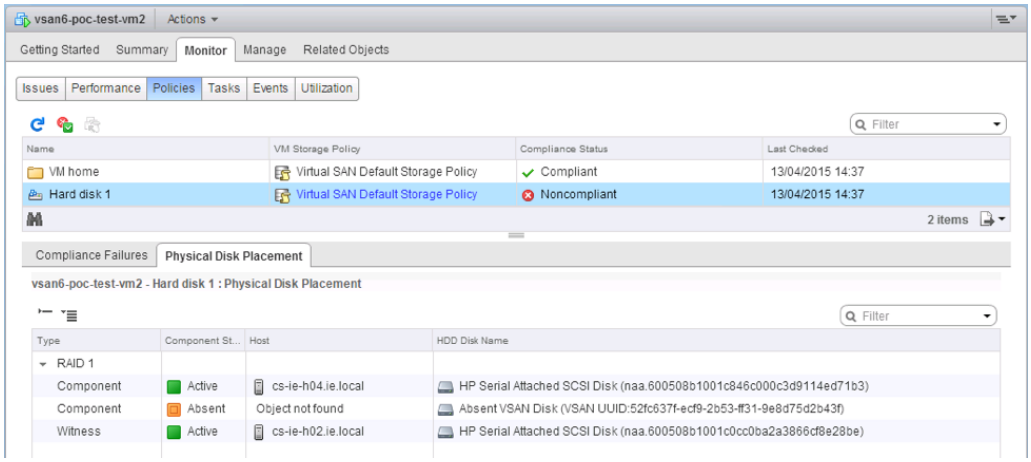


Figure 12.19: Component is absent

After the 60 minutes has elapsed, the component should be rebuilt on a different disk in the cluster. That is what is observed. Note the component resides on a new disk (NAA id is different).

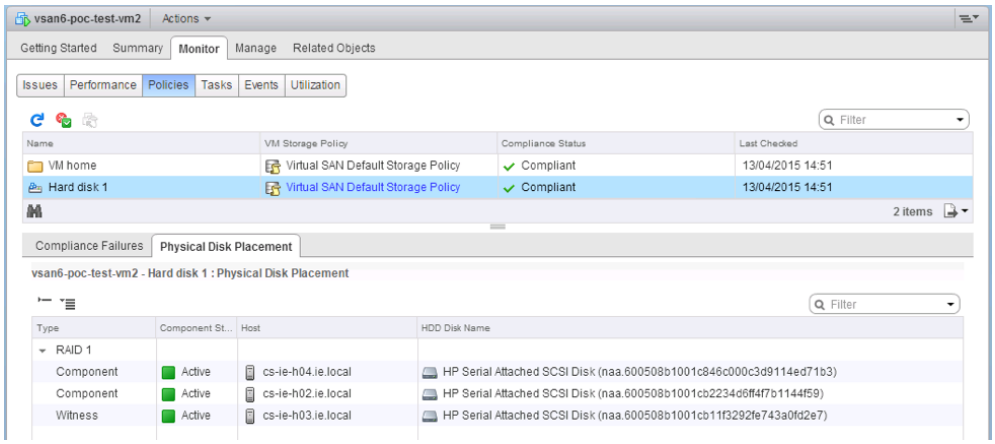


Figure 12.20: Component is rebuilt

The removed disk can now be re-added by scanning the HBA:

Navigate to the host > Manage > Storage > Storage Devices and click the rescan button. See Figure 12.18 above for a screenshot.

Look at the list of storage devices for the NAA ID that was removed. If for some reason, the disk doesn't return after refreshing the screen, try rescanning the host again. If it still doesn't appear, reboot the ESXi host. Once the NAA ID is back, clear any hot unplug flags set previously with the *–c* option:

```
[root@cs-ie-h01:~] python /usr/lib/vmware/vsan/bin/vsanDiskFaultInjection.pyc -c
-d naa.600508b1001c388c92e817e43fcd5237
Clearing errors on device vmhba1:C0:T4:L0
vsish -e set /storage/scsifw/paths/vmhba1:C0:T4:L0/injectError 0x00000
vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x00000
```

That completes this part of the POC.

12.14 Pull Cache Tier SSD and Do Not Reinsert/Replace

For the purposes of this test, we shall remove an SSD from one of the disk groups in the cluster. Navigate to the cluster > Manage > Settings > Virtual SAN > Disk Management. Select a disk group from the top window and identify its SSD in the bottom window. If All-Flash, make sure it's the Flash device in the "Cache" Disk Role. Make a note of the SSD's NAA ID string.

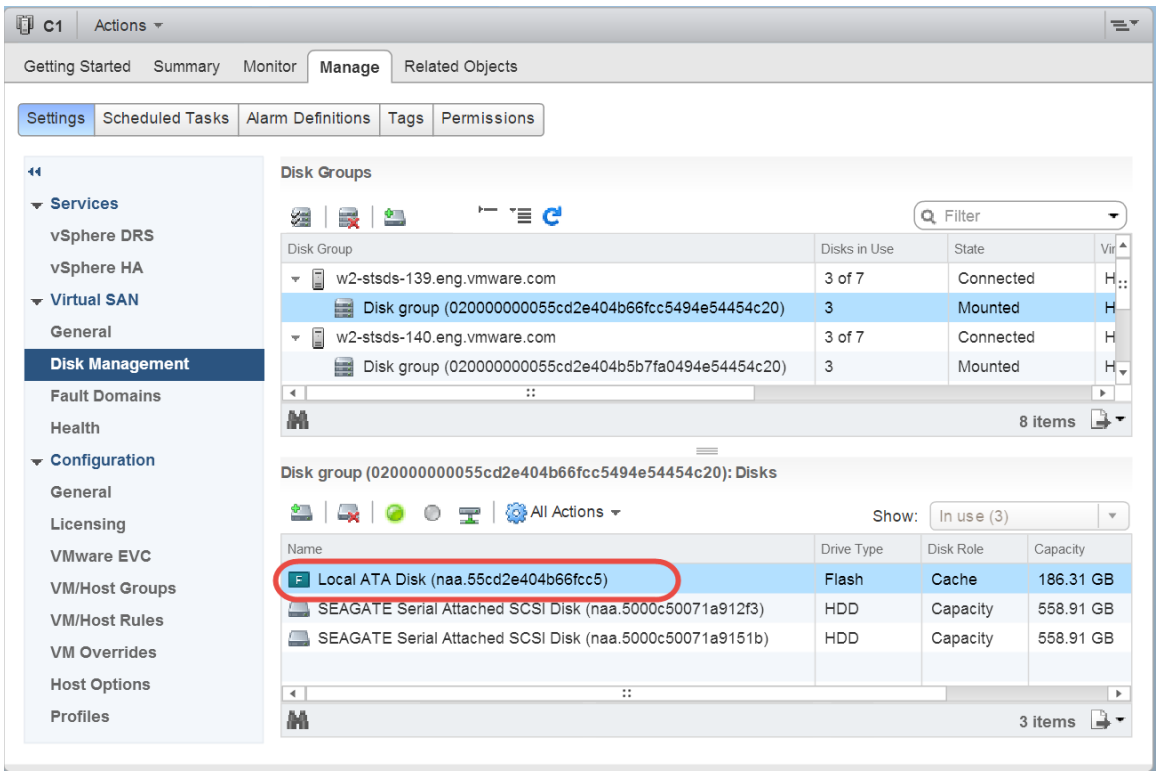


Figure 12.21: Locate a caching-tier SSD

In the above screenshot, we have located an SSD on host w2-stsds-139 with an NAA ID string of naa.55cd2e404b66fcc5. Next, SSH into the host with the SSD to pull. Inject a hot unplug event using the *vsanDiskFaultInjection.pyc* python script:

```
[root@w2-stsds-139:~] python /usr/lib/vmware/vsan/bin/vsanDiskFaultInjection.pyc
-u -d naa.55cd2e404b66fcc5
Injecting hot unplug on device vmhba1:C0:T4:L0
vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x1
vsish -e set /storage/scsifw/paths/vmhba1:C0:T4:L0/injectError 0x004C0400000002
```

Now we observe the impact that losing an SSD (flash device) has on the whole disk group.

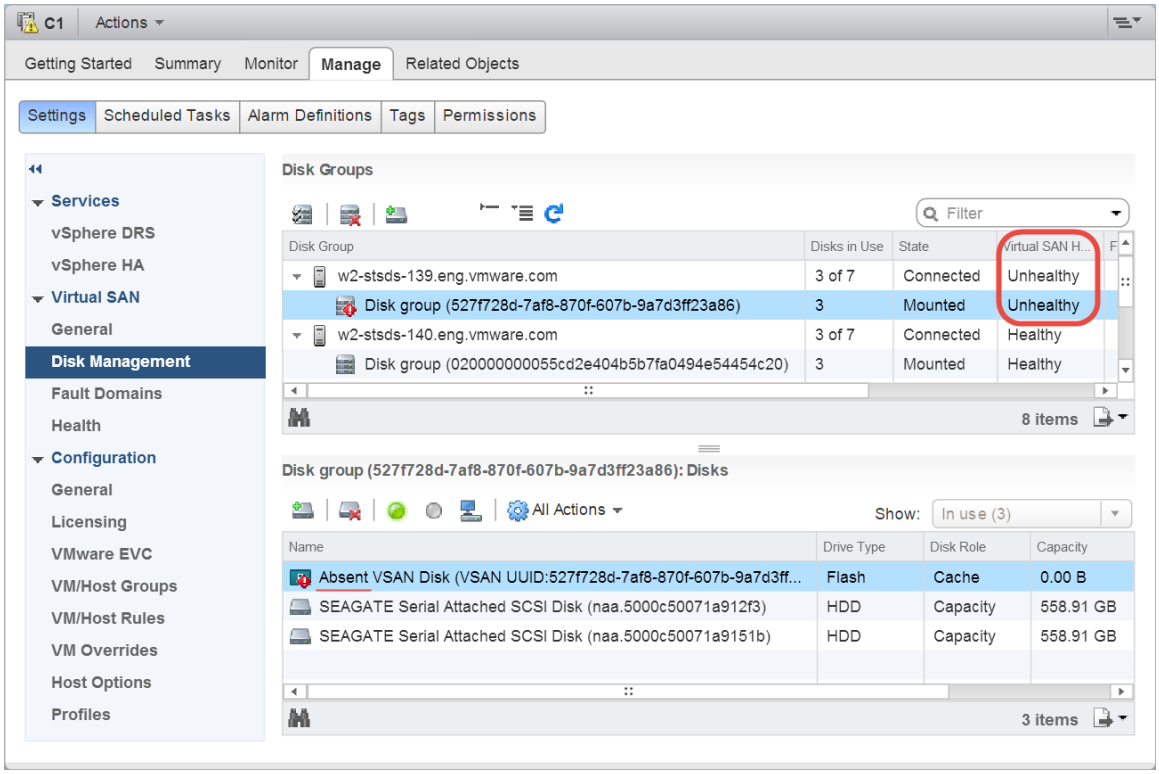


Figure 12.22: Absent cache tier SSD = Unhealthy Disk Group

And finally, let's look at the components belonging to the virtual machine. This time, any components that were residing on that disk group are absent.

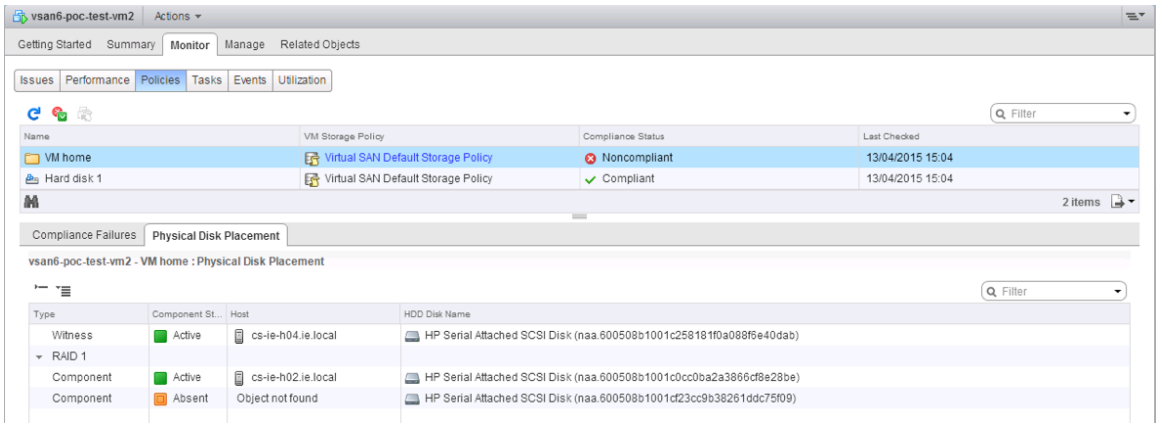


Figure 12.23: SSD removed – all components absent

To show that this impacts all VMs, here is another VM that had a component on local storage on host cs-ie-h01.ie.local.

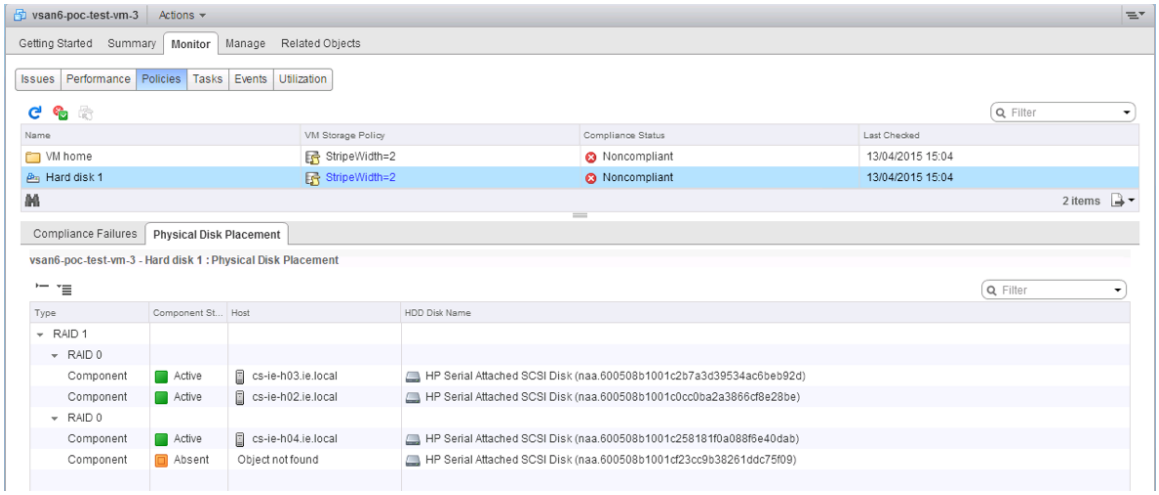


Figure 12.24: SSD removed – all components absent

If you search all your VMs, you will see that each VM that had a component on the disk group on cs-ie-h07 now has absent components. This is expected since an SSD failure impacts the whole of the disk group.

After 60 minutes have elapsed, new components should be rebuilt in place of the absent components. If you manage to refresh at the correct moment, you should be able to observe the additional components synchronizing with the existing data.

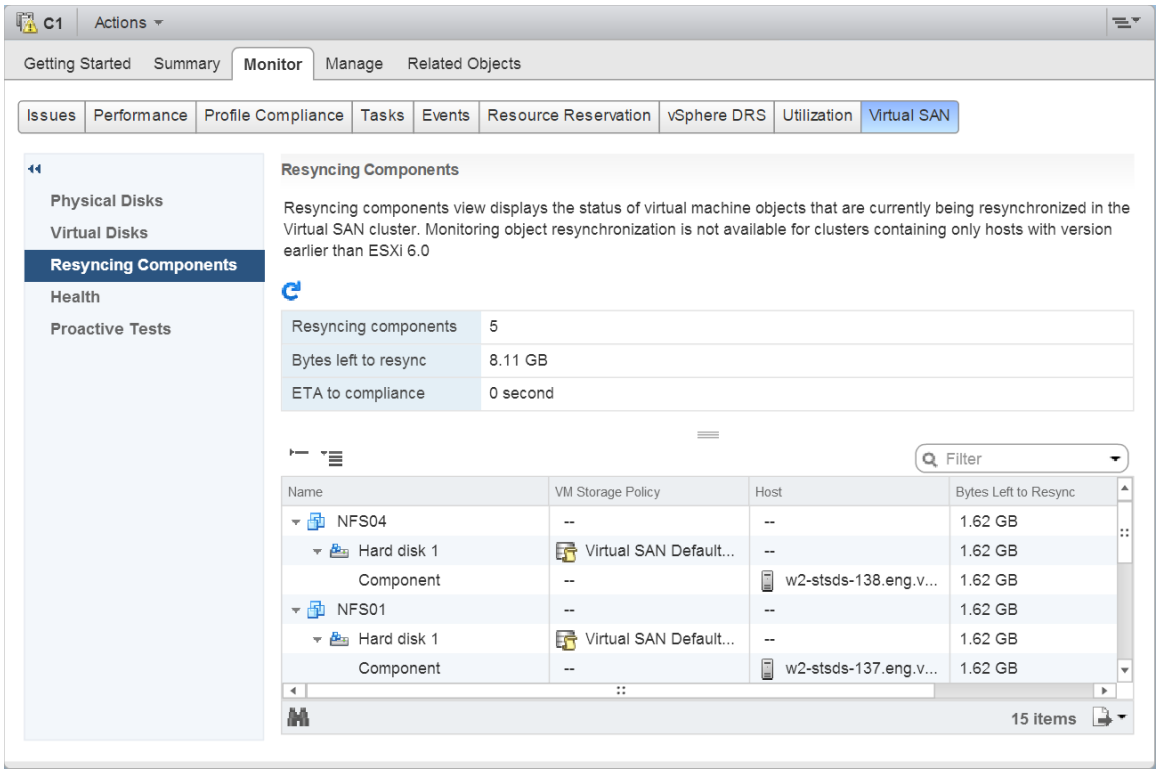


Figure 12.25: New components resynchronizing after clomd timeout expires

To complete this POC, re-add the SSD logical device back to the host by rescanning the HBA:

Navigate to the host > Manage > Storage > Storage Devices and click the rescan button. See Figure 12.18 above for a screenshot.

Look at the list of storage devices for the NAA ID of the SSD that was removed. If for some reason, the SSD doesn't return after refreshing the screen, try rescanning the host again. If it still doesn't appear, reboot the ESXi host. Once the NAA ID is back, clear any hot unplug flags set previously with the *–c* option:

```
[root@cs-ie-h01:~] python /usr/lib/vmware/vsan/bin/vsanDiskFaultInjection.pyc -c
-d naa.55cd2e404b66fcc5
Clearing errors on device vmhba1:C0:T4:L0
vsish -e set /storage/scsifw/paths/vmhba1:C0:T4:L0/injectError 0x00000
vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x00000
```

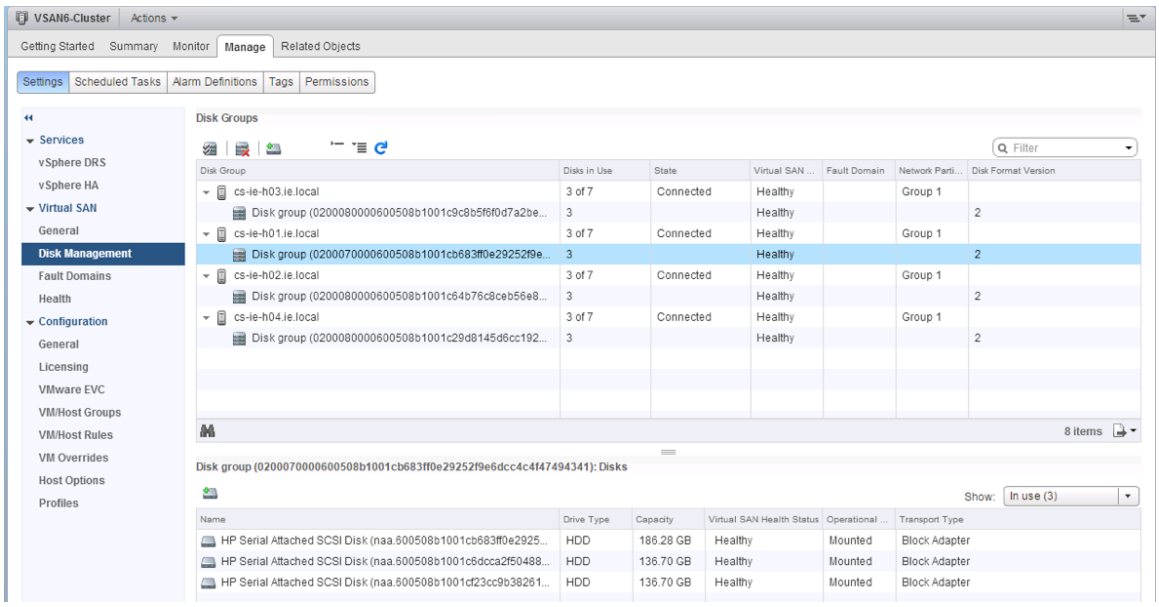


Figure 12.26: Verify that the disk group is back in a health state

Warning: If you delete an SSD drive that was marked as an SSD, and a logical RAID 0 device was rebuilt as part of this test, you may have to mark the drive as an SSD once more.

12.15 Checking Rebuild/Resync Status

Virtual SAN 6.0 displays details on resyncing components. Navigate to Monitor tab > Virtual SAN > Resyncing Components.

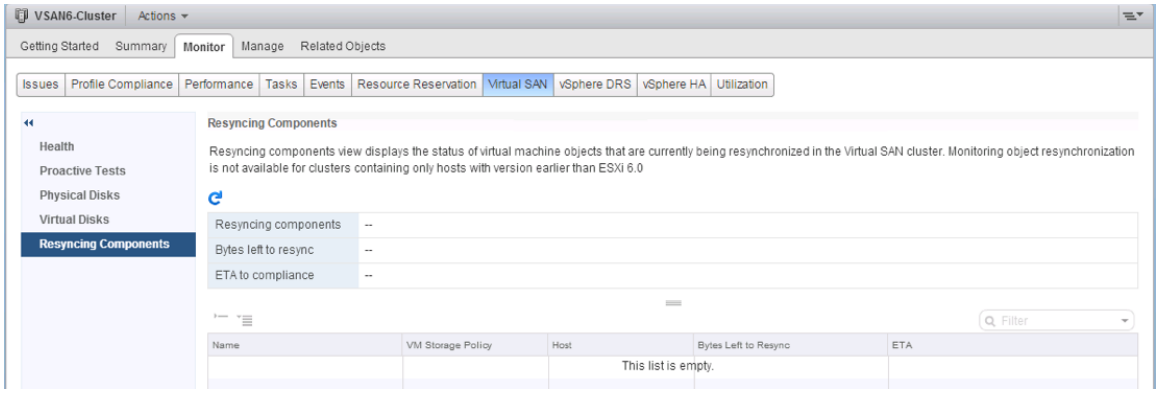


Figure 12.27: Resyncing Components

To check the status of component resync/rebuild on a Virtual SAN Cluster using RVC commands, the following command will display useful information:

- **vsan.resync_dashboard**

When resynchronization is complete, this command will report “0 bytes to sync”.

12.16 Injecting a Disk Error

The first step is to select a host, and the select a disk that is part of a disk group on that host. The –d DEVICENAME argument requires the SCSI identifier of the disk, typically the NAA id. You might also wish to verify that this disk does indeed contain VM components. This can be done by selecting a VM, then selecting the Monitor > Policies > Physical Disk Placement tab.
cs-ie-03, and has an NAA id of `600508b1001c1a7f310269ccd51a4e83`:

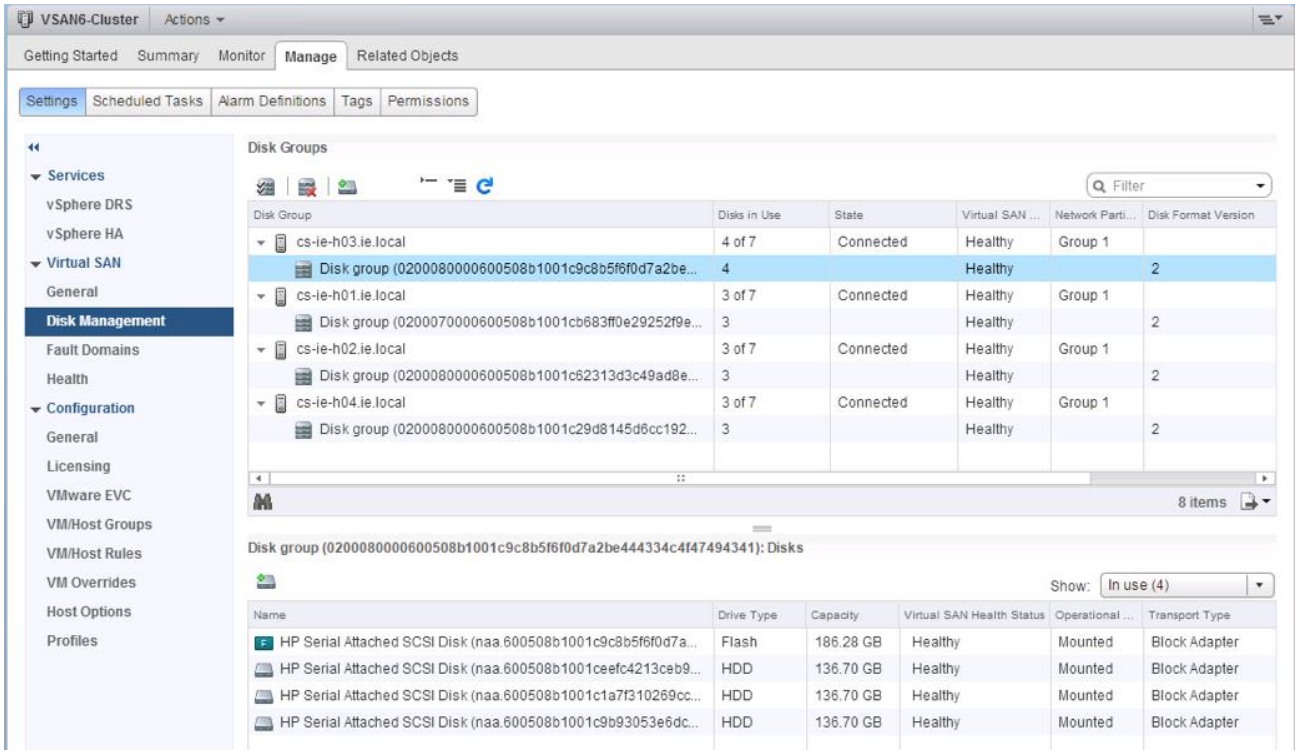


Figure 12.28: Healthy Disk Group

The error can only be injected from the command line of the ESXi host. To display the NAA ids of the disks on the ESXi host, you will need to SSH to the ESXi host, login as the *root* user, and run the following command:

```
[root@cs-ie-h03:/usr/lib/vmware/vsan/bin] esxcli storage core device list| grep
 ^naa
 naa.600508b1001ceefc4213ceb9b51c4be4
 naa.600508b1001cd259ab7ef213c87eaad7
 naa.600508b1001c9c8b5f6f0d7a2be44433
 naa.600508b1001c2b7a3d39534ac6beb92d
 naa.600508b1001cb11f3292fe743a0fd2e7
 naa.600508b1001c1a7f310269ccd51a4e83
 naa.600508b1001c9b93053e6dc3ea9bf3ef
 naa.600508b1001c626dcb42716218d73319
```

Once a disk has been identified, and has been verified to be part of a disk group, and that the disk contains some virtual machine components, we can go ahead and inject the error as follows:

```
[root@cs-ie-h03:/usr/lib/vmware/vsan/bin] python vsanDiskFaultInjection.pyc -p
-d naa.600508b1001c1a7f310269ccd51a4e83
 Injecting permanent error on device vmhba1:C0:T0:L4
 vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x1
 vsish -e set /storage/scsifw/paths/vmhba1:C0:T0:L4/injectError
0x03110300000002
 [root@cs-ie-h03:/usr/lib/vmware/vsan/bin]
```

Before too long, the disk should display an error and the disk group should enter an unhealthy state.

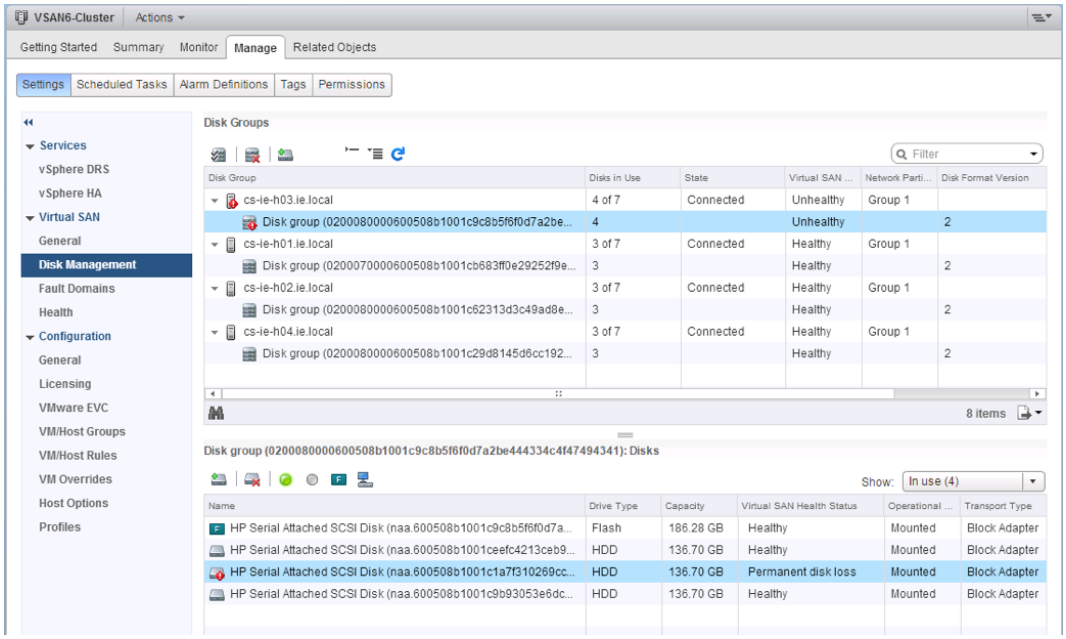


Figure 12.29: Unhealthy Disk Group

Notice that the disk group is in an Unhealthy state and the status of the disk is "Permanent disk loss". This should place any components on the disk into a degraded state (which can be observed via the VM's Physical Disk Placement tab, and initiate an immediate rebuild of components. Navigating to Cluster > Monitor > Virtual SAN > Resyncing Components should reveal the components resyncing.

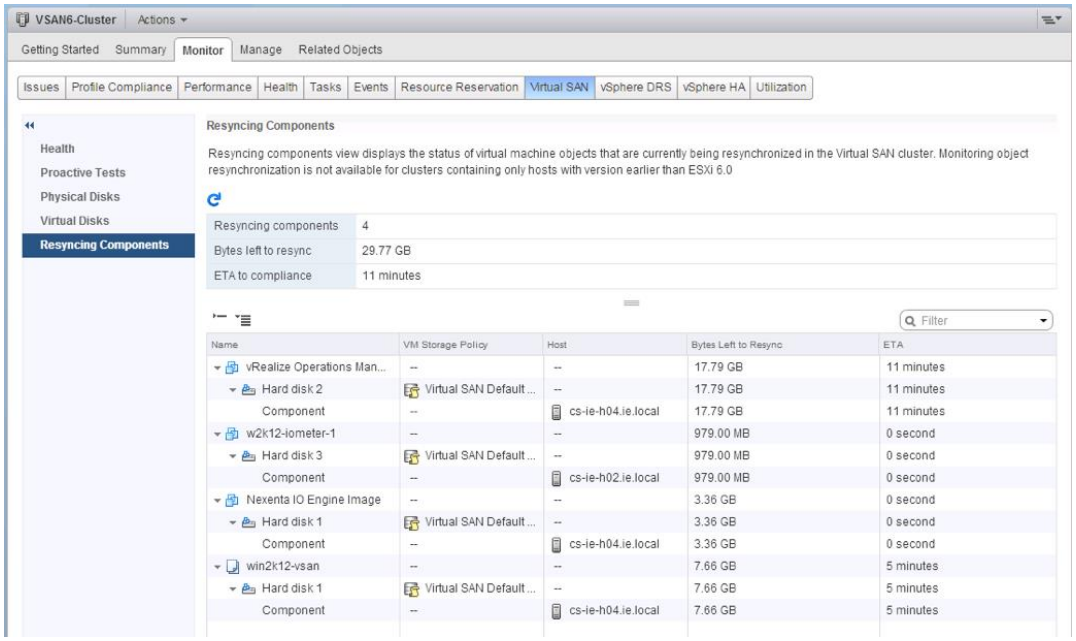


Figure 12.30: Resyncing components after disk failure

12.16.2 Clear a Permanent Error

At this point, we can clear the error. We use the same script that was used to inject the error, but this time we provide a –c (clear) option:

```
[root@cs-ie-h03:/usr/lib/vmware/vsan/bin] python vsanDiskFaultInjection.pyc -c
-d naa.600508b1001c1a7f310269ccd51a4e83
 Clearing errors on device vmhba1:C0:T0:L4
 vsish -e set /storage/scsifw/paths/vmhba1:C0:T0:L4/injectError 0x00000
 vsish -e set /reliability/vmkstress/ScsiPathInjectError 0x00000
 [root@cs-ie-h03:/usr/lib/vmware/vsan/bin]
```

Note however that since the disk failed, it will have to be removed, and re-added from the disk group. This is very simple to do. Simply select the disk in the disk group, and remove it by clicking on the icon highlighted below.

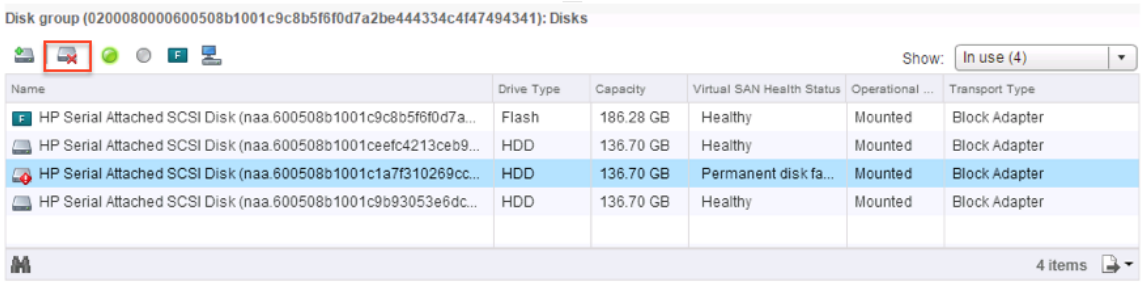


Figure 12.31: Remove disk from disk group

This will display a pop-up window regarding which action to take regarding the components on the disk. You can choose to migrate the components or not. By default it is shown as “Evacuate Data”, shown here.

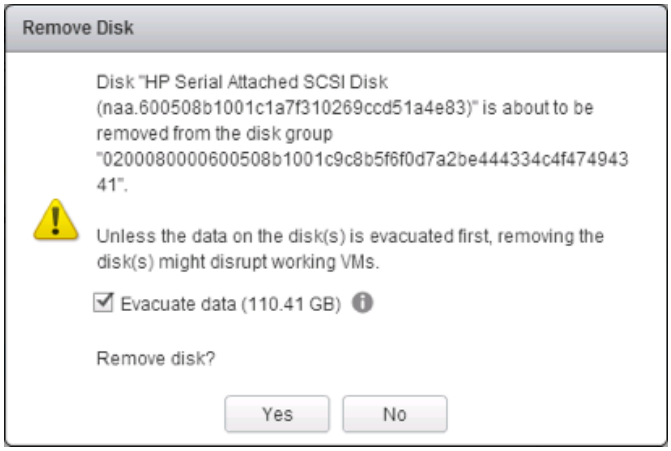


Figure 12.32: Data is evacuated by default, but can be unchecked in this test

For the purposes of this POC, you can *uncheck* this option as you are adding the disk back in the next step. When the disk has been removed and re-added, the disk group will return to a healthy state. That completes the disk failure test.

12.17 When Might a Rebuild of Components Not Occur?

There are a couple of reasons why a rebuild of components might not occur.

12.17.1 Lack of Resources

Verify that there are enough resources to rebuild components before testing with the following RVC command:

- **vsan.whatif_host_failures**

Of course, if you are testing with a 3-node cluster, and you introduce a host failure, there will be no rebuilding of objects. Once again, if you have the resources to create a 4-node cluster, then this is a more desirable configuration for evaluation Virtual SAN.

12.17.2 Underlying Failures

Another cause of a rebuild not occurring is due to an underlying failure already present in the cluster. Verify there are none before testing with the following RVC command:

- **vsan.hosts_info**
- **vsan.check_state**
- **vsan.disks_stats**

If these commands reveal underlying issues (ABSENT or DEGRADED components for example), rectify these first or you risk inducing multiple failures in the cluster, resulting in inaccessible virtual machines.

13. Virtual SAN Management

In this section, we shall look at a number of management tasks, such as the behavior when placing a host into maintenance mode, and the evacuation of a disk and a disk group from a host. We will also look at how to turn on and off the identifying LEDs on a disk drive.

13.1 Put a Host into Maintenance Mode

There are a number of options available when placing a host into maintenance mode. The first step is to identify a host that has a running VM, as well as components belonging to virtual machine objects.

Select the Summary tab of the virtual machine to verify which host it is running on.

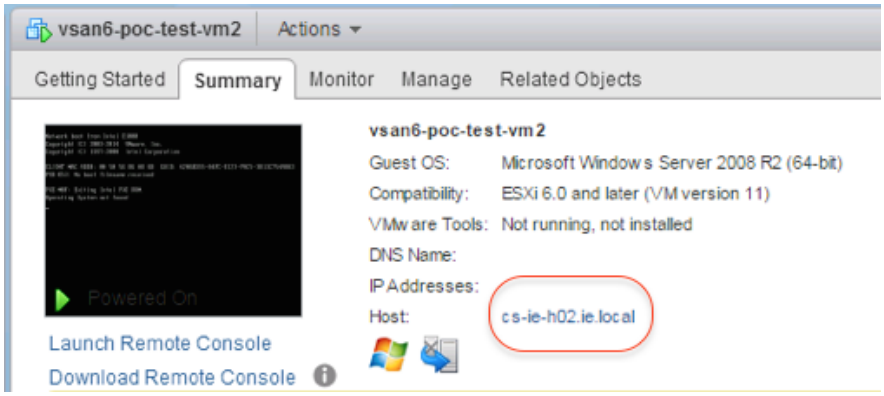


Figure 13.1: VM Summary tab

Then select the Monitor tab > Policies > Physical Disk Placement and verify that there are components also residing on the same host.

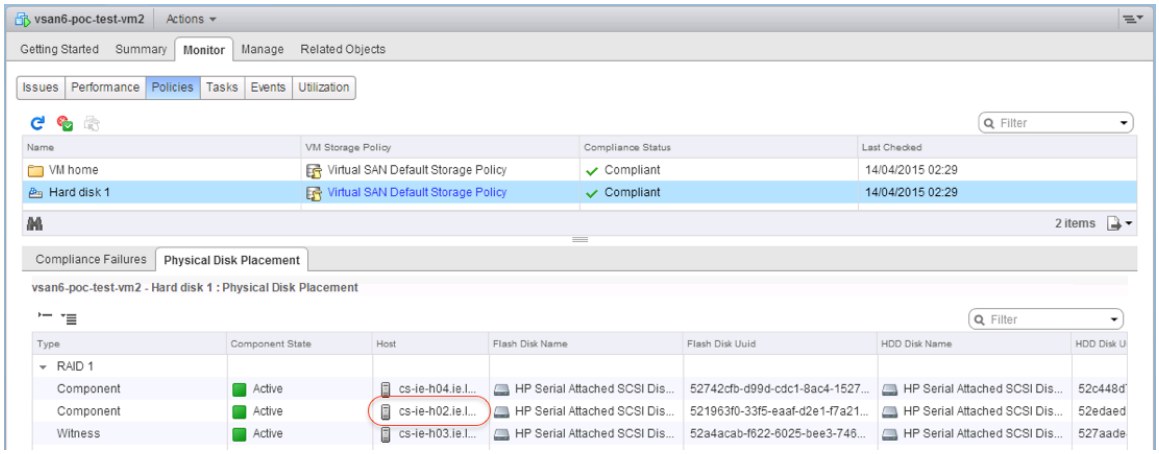


Figure 13.2: Physical Disk Placement

From the screenshots shown here, we can see that the VM selected is running on host cs-ie-h02 and also has components residing on that host. This is the host that we shall place into maintenance mode.

Right click on the host, select Maintenance Mode from the dropdown menu, then select the option "Enter Maintenance Mode" as shown here.

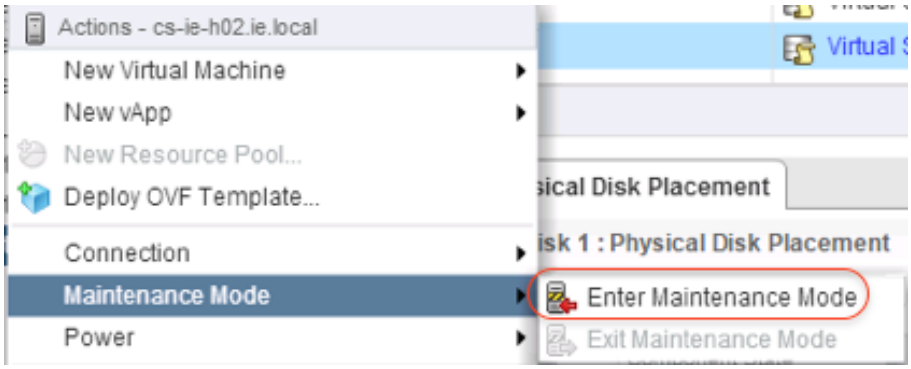


Figure 13.3: Enter Maintenance Mode

There are three options displays when maintenance mode is selected; (i) Ensure accessibility, (ii) Full data migration and (iii) No data migration.

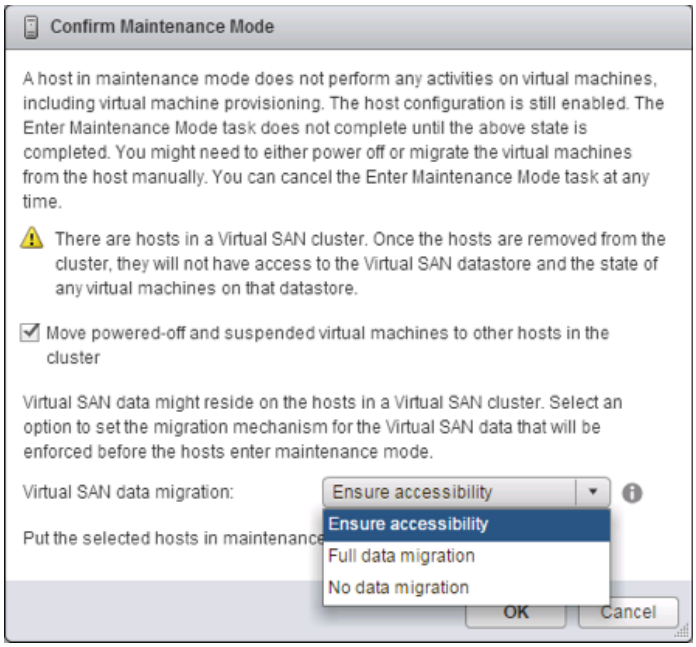


Figure 13.4: Maintenance Mode options

In this first part of the maintenance mode testing, we shall select the option "Ensure accessibility". This means that although components may go missing, the VMs shall remain accessible.

When this option is chosen, a popup is displayed regarding migrating running virtual machines. Since this is a fully automated DRS cluster, the virtual machines should be automatically migrated.

Figure 13.5: Migration warning

After the host has entered maintenance mode, we can now examine the state of the components that were on the local storage of this host. What you should observe is that these components are now in an "Absent" state. However the VM remains accessible as we chose the option "Ensure Accessibility" when entering Maintenance Mode.

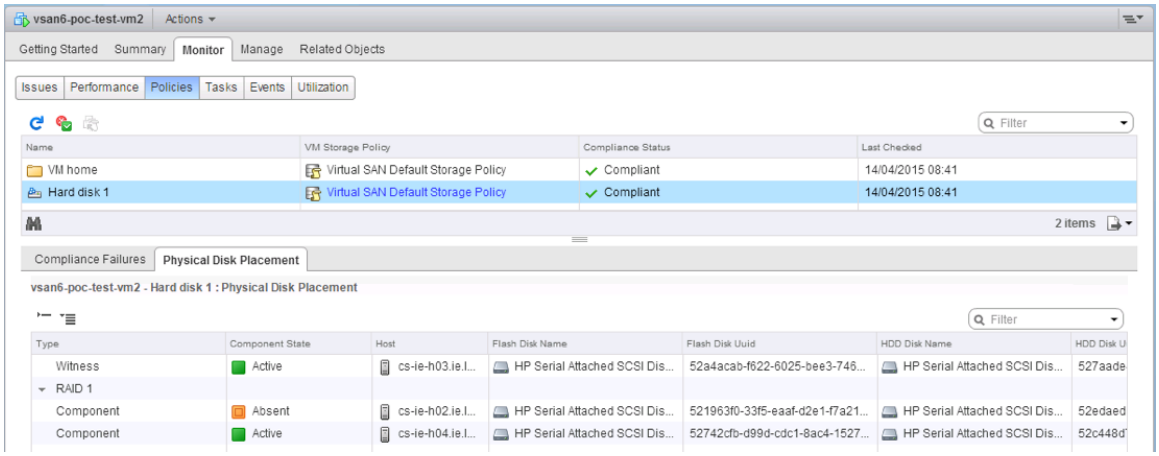


Figure 13.6: Components are Absent during Maintenance Mode

The host can now be taken out of maintenance mode. Simply right click on the host as before, select Maintenance Mode and then Exit Maintenance Mode.

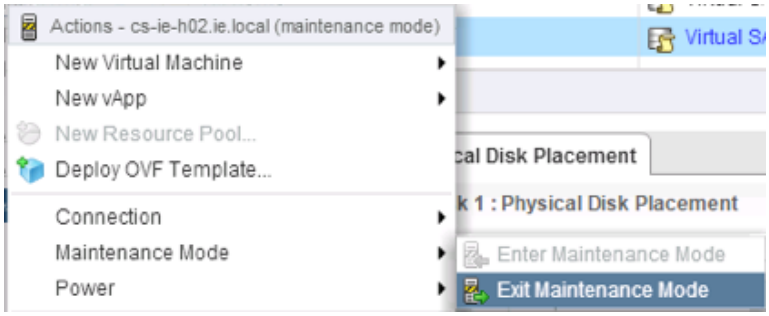


Figure 13.7: Exit Maintenance Mode

After exiting Maintenance Mode, the "Absent" component becomes Active once more. This is assuming that the host exited maintenance mode before the vsan.ClomdRepairDelay expires (default 60 minutes).

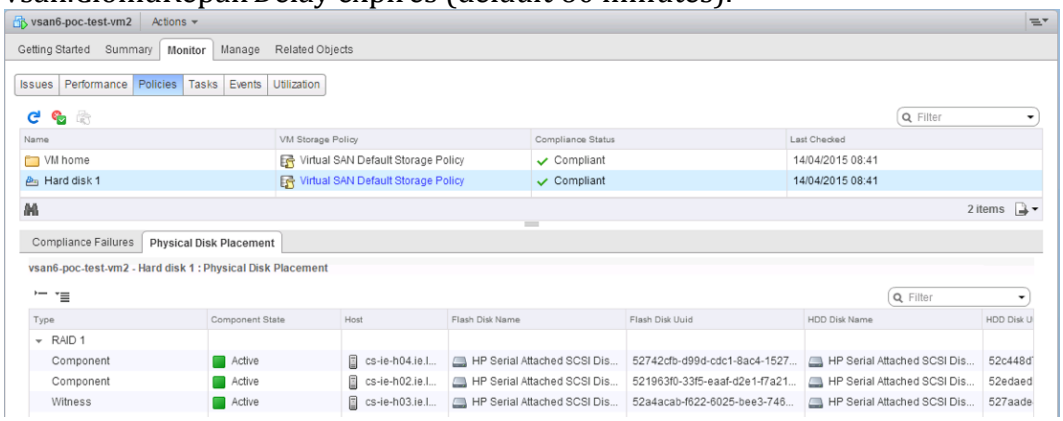


Figure 13.8: Component is Active once more

We shall now place the host into maintenance mode once more, but this time instead of "Ensure Accessibility", we shall choose "Full data migration". This means that although components on the host in maintenance mode will no longer be available, those components will be rebuilt elsewhere in the cluster, implying that there is full availability of the virtual machine objects.

Note: This is only possible when NumberOfFailuresToTolerate = 1 and there are 4 or more hosts in the cluster. It is not possible with 3 hosts and NumberOfFailuresToTolerate = 1, as another host needs to be available to rebuild the components. This is true for higher values of NumberOfFailuresToTolerate also.

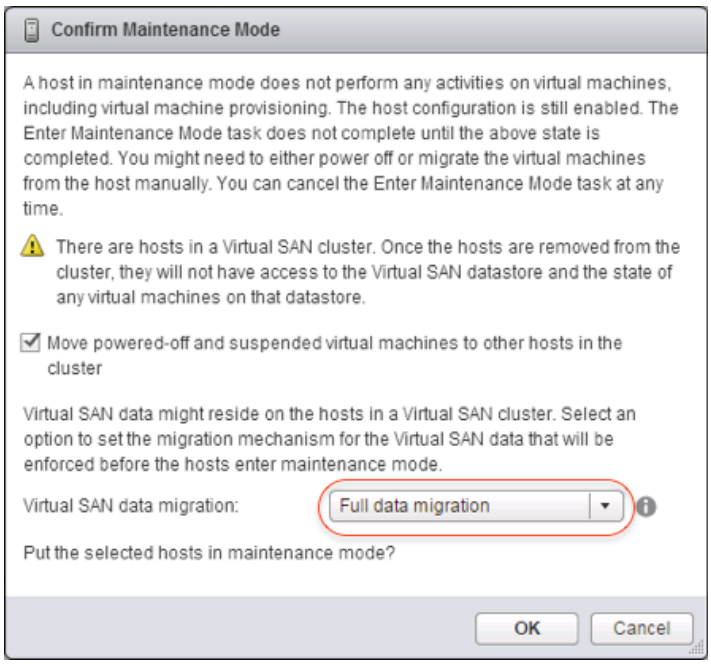


Figure 13.9: Full data migration

Now if the components on host cs-ie-h02.ie.local are monitored, you will see that no components are placed in an “Absent” state, but rather they are rebuilt on the other hosts in the cluster. When the host enters maintenance mode, you will notice that all components of the virtual machines are active, but none reside on the host placed into maintenance mode.

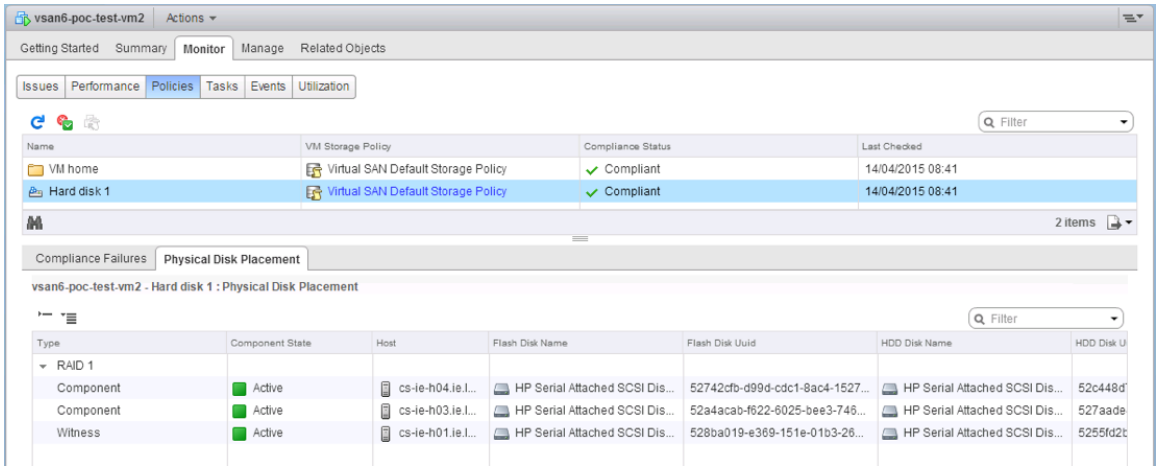


Figure 13.10: All components are Active when host is in mode (full data migration)

Exit maintenance mode. This completes this part of the POC.

13.2 Remove and Evacuate a Disk

In this example, we show a feature introduced in version 6.0. This is the ability to evacuate a disk prior to removing it from a disk group.

Note: The cluster must be left in manual mode. The operations are not available when a cluster is in automatic mode.

Navigate to the cluster > Manage tab > Virtual SAN > Disk Management, and select a disk group in one of the hosts as shown below. Then select one of the capacity disks from the disk group, also shown below. Note that the disk icon with the red x becomes visible. This is not visible if the cluster is in automatic mode.

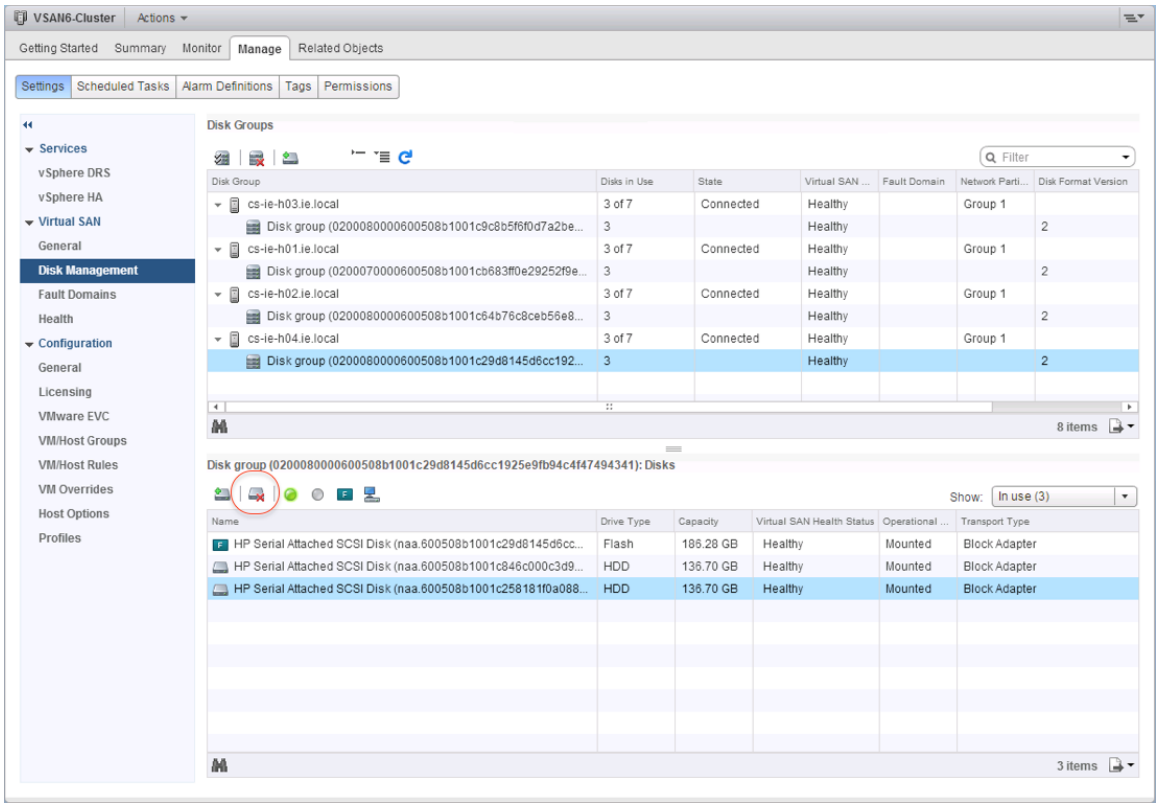


Figure 13.11: Remove a disk

Make a note of the devices in the disk group, as you will need these later to rebuild the disk group. There are a number of new icons on this view of disk groups in Virtual SAN 6.0. It is worth spending some time understanding that they mean. The following table should help to explain that.

	Add a disk to the selected disk group
	Remove (and optionally evacuate data) from a disk in a disk group
	Turn on the locator LED on the selected disk
	Turn off the locator LED on the selected disk
	Tag a device as a flash device (useful when RAID 0, non-passthru in use)
	Tag a device as a local device (useful when SAS controllers in use)

Table 13.1: Disk group icons

To continue with the option of removing a disk from a disk group and evacuating the data, click on the icon to remove a disk highlighted earlier. This pops up the following window, which gives you the option to evacuate data (selected automatically). Click "Yes" to continue:

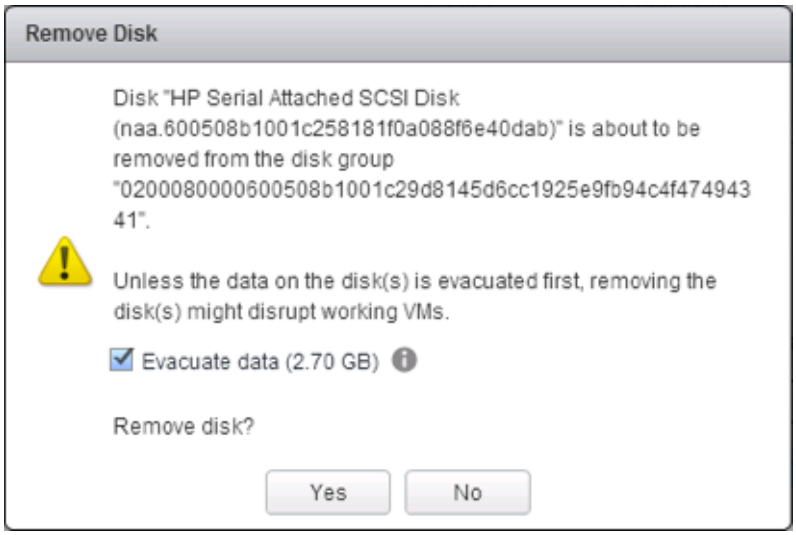


Figure 13.12: Evacuate data

When the operation completes, there should be one less disk in the disk group, but if you examine the components of your VMs, there should be none found to be in an "Absent" state. All components should be "Active", and any that were originally on the disk that was evacuated should now be rebuilt elsewhere in the cluster.

13.3 Evacuate a Disk Group

Let's repeat the previous task for the rest of the disk group. Instead of removing the original disk, let's now remove the whole of the disk group. Make a note of the devices in the disk group, as you will need these later to rebuild the disk group.

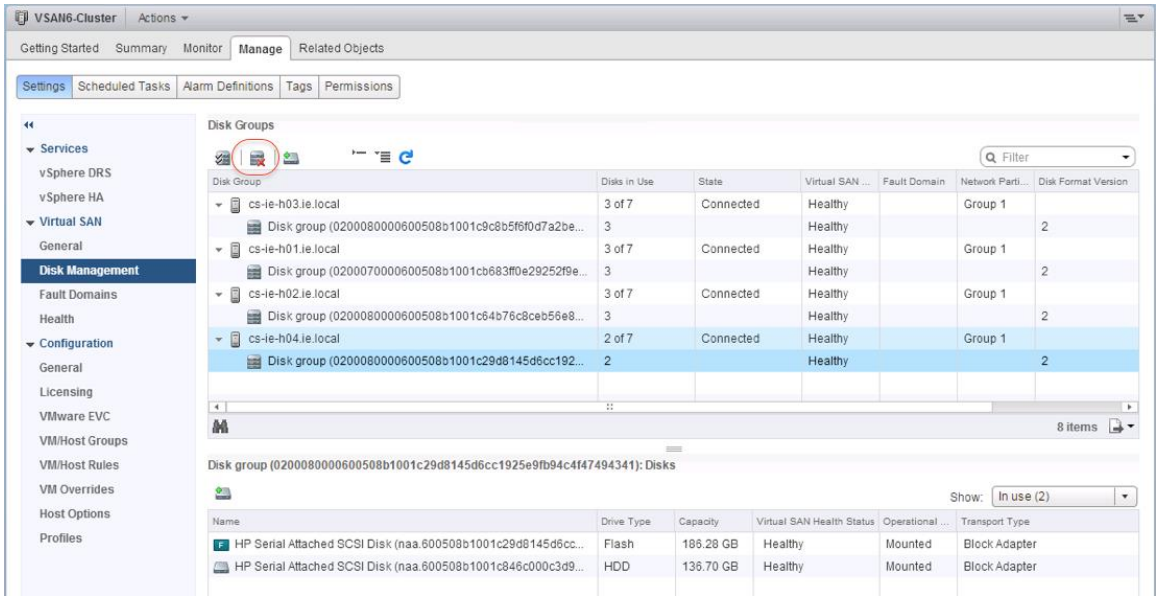


Figure 13.13: Delete disk group

As before, you are prompted as to whether or not you wish to evacuate the data from the disk group. The amount of data is also displayed, and the option is selected by default. Click "Yes" to continue.

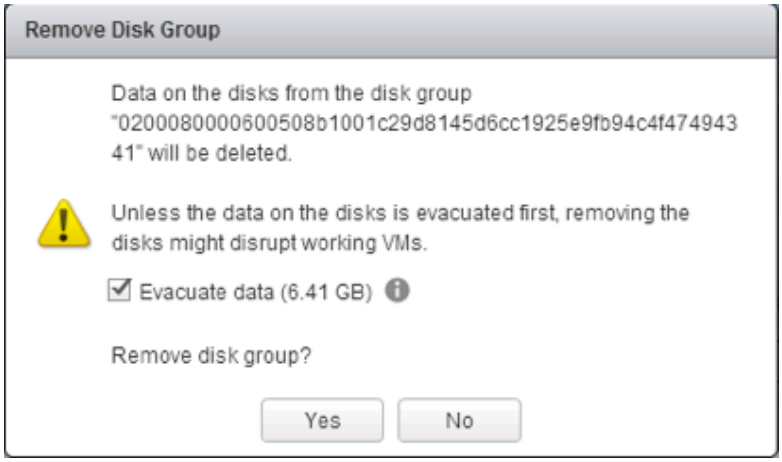


Figure 13.14: Evacuate data

Once the evacuation process has completed, the disk group should no longer be visible in the Disk Groups view.

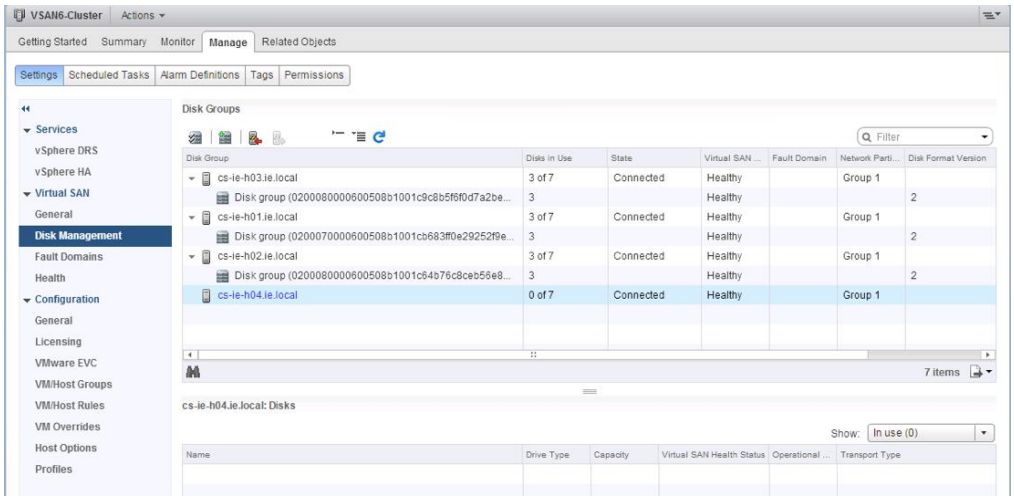


Figure 13.15: Disk group now removed and evacuated

Once again, if you examine the components of your VMs, there should be none found to be in an "Absent" state. All components should be "Active", and any that were originally on the disk that was evacuated should now be rebuilt elsewhere in the cluster.

13.4 Add Disk Groups Back Again

At this point, we can recreate the deleted disk group. This was already covered in section 6.1 of this POC guide. Simply select the host that the disk group was removed from, and click on the icon to create a new disk group. Once more, select a flash device and the two magnetic disk devices that you previously noted were members of the disk group. Click OK to recreate the disk group.

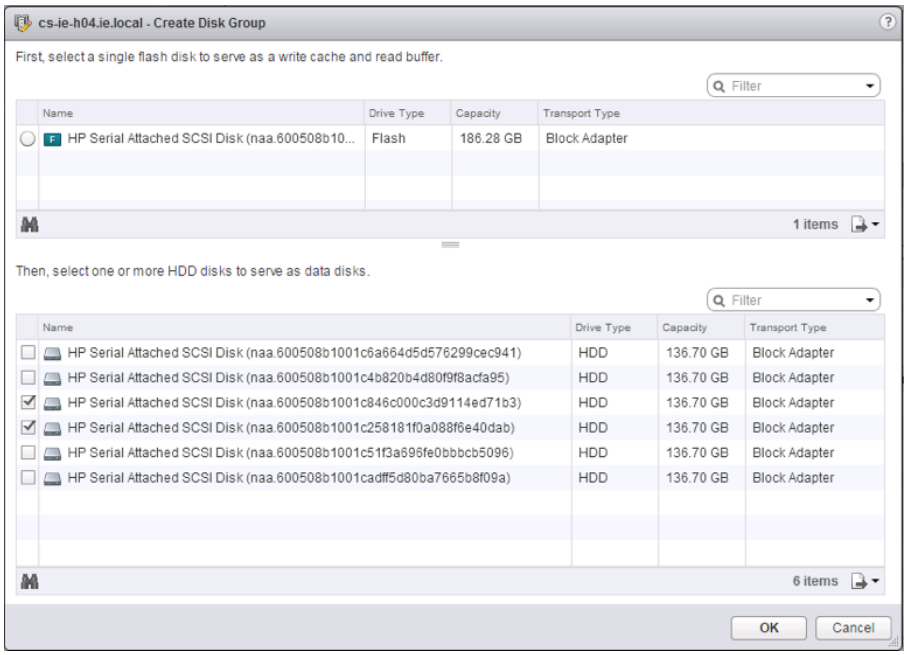


Figure 13.16: Recreate disk group

13.5 Turning on and off Disk LEDs

Our final maintenance task is to turn on and off the locator LEDs on the disk drives. This is a new feature of Virtual SAN 6.0. In chapter 12, we spoke about the importance of the *hpssacli* utility for removing and adding logical devices. This was a "nice to have". However for turning on and off the disk locator LEDs, the utility is a necessity when using HP controllers. Refer to section 12.10 for information on how to locate and install this utility.

Note: This is not an issue for LSI controllers, and all necessary components are shipped with ESXi for these controllers.

The icons for turning on and off the disk locator LEDs are shown in table 13.1. To turn on a LED, select a disk in the disk group and then click on the icon highlighted below.

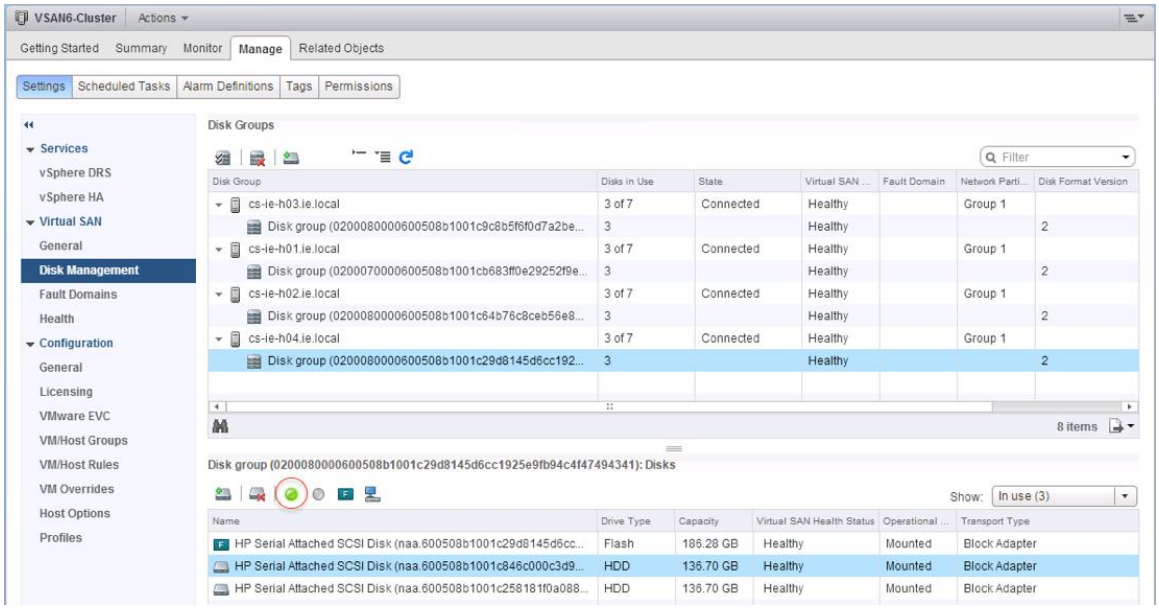


Figure 13.17: Turn on disk locator LED

This will launch a task to "Turn on disk locator LEDs". To see if the task was successful, go to the Monitor tab and check the Events. If there is no error, the task was successful. At this point you can also take a look in the data center and visually check if the LED of the disk in question is lit.

Once completed, the locator LED can be turned off by clicking on the "Turn off disk locator LEDs" as highlighted in the screen shot below. Once again, this can be visually checked in the data center if you wish.

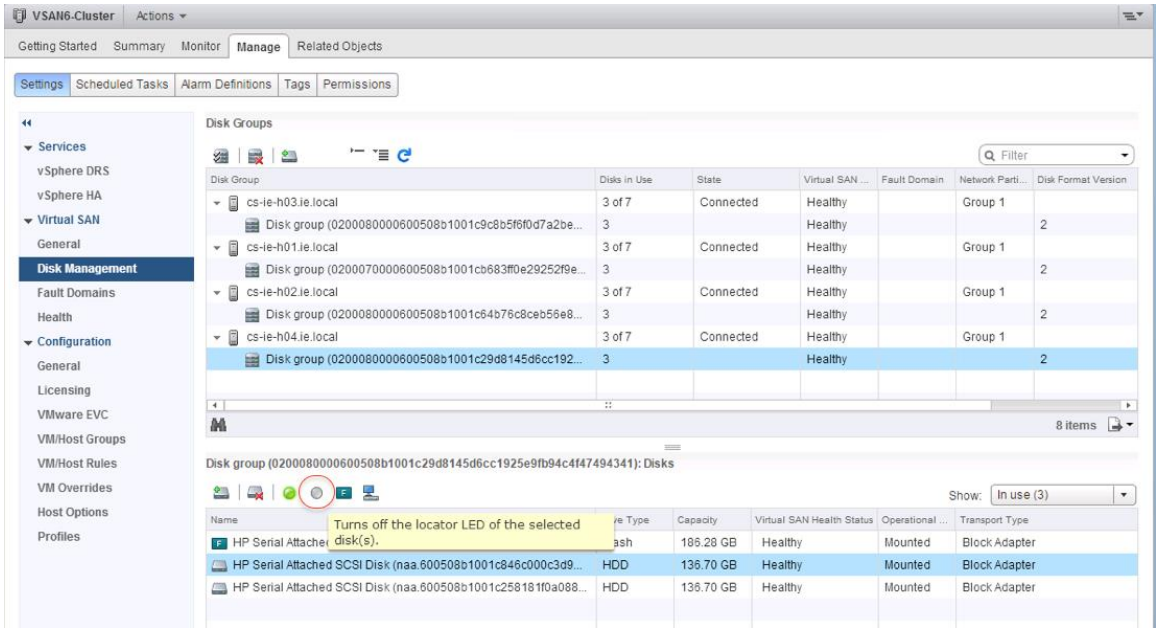


Figure 13.18: Turn off disk locator LED

This completes this section of the Virtual SAN 6.0 Proof-Of-Concept (POC) guide. Before handing over the environment to the customer, do one final check on the health and ensure all checks pass.

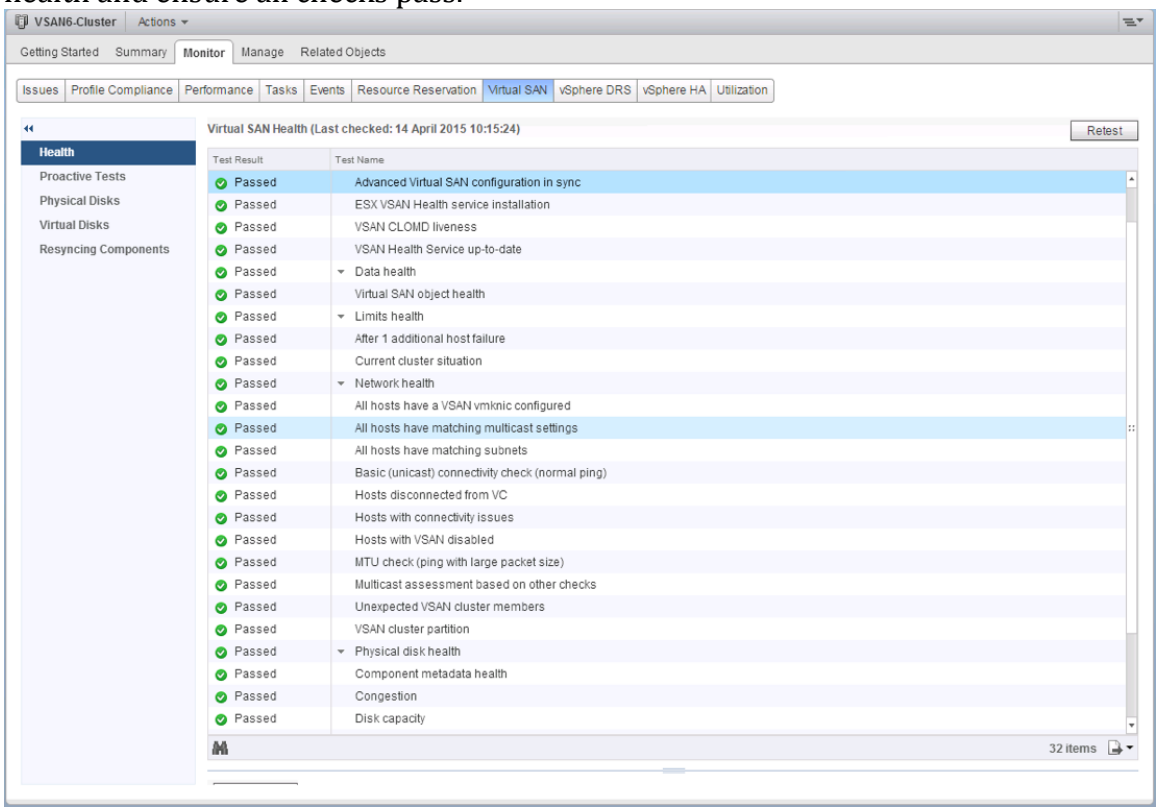


Figure 13.19: Final health check

14. Virtual SAN 6.1 Stretched Cluster Configuration

As per of the vSphere 6.0U1 release in September 2015, a number of new Virtual SAN features were included. The features included a Stretched Cluster solution, which is the purpose of this report. Note that the Virtual SAN version in vSphere 6.0U1 is Virtual SAN 6.1.

A good working knowledge of how Virtual SAN Stretched Cluster is designed and architected is assumed. Readers unfamiliar with the basics of Virtual SAN Stretched Cluster are urged to review the relevant documentation before proceeding with this part of the proof-of-concept. Details on how to configure a Virtual SAN Stretched Cluster are found in the Virtual SAN 6.1 Stretched Cluster Guide.

14.1 Virtual SAN 6.1 Stretched Cluster Network Topology

As per the *Virtual SAN 6.1 Stretched Cluster Guide,* a number of different network topologies are supported for Virtual SAN Stretched Cluster. The network topology deployed in this lab environment is a full layer 3 stretched Virtual SAN network. L3 multicast is implemented for the Virtual SAN network between data sites, and L3 unicast is implemented for the Virtual SAN network between data sites and the witness site. While VMware also supports stretched L2 between the data sites, L3 is the only supported network topology for the Virtual SAN network between the data sites and the witness site. The VM network is a stretched L2 between both data sites.

14.2 Virtual SAN 6.1 Stretched Cluster Hosts

There are four ESXi hosts in this cluster, two ESXi hosts on data site A (the "preferred" site) and two hosts on data site B (the "secondary" site). There is one disk-group per host (all flash). The witness host/appliance is deployed on a 3rd, remote data center. The configuration is referred to as 2+2+1.

Disk Group	Disks in Use	State	Virtual SAN ...	Fault Domain	Network Parti...
esxi-a-pref.rainpole.com	2 of 2	Connected	Healthy	Preferred	Group 1
esxi-b-pref.rainpole.com	2 of 2	Connected	Healthy	Preferred	Group 1
esxi-c-scnd.rainpole.com	2 of 2	Connected	Healthy	Secondary	Group 1
esxi-d-scnd.rainpole.com	2 of 2	Connected	Healthy	Secondary	Group 1
esxi-e-witn.rainpole.com...	2 of 2	Connected	Healthy		Group 1

Figure 14.1: Hosts in Virtual SAN cluster

VMs are deployed on both the "Preferred" and "Secondary" sites of the Virtual SAN Stretched Cluster. VMs are running/active on both sites.

14.3 Virtual SAN 6.1 Stretched Cluster Diagram

Below is a diagram detailing the POC environment used for the Stretched Cluster testing.

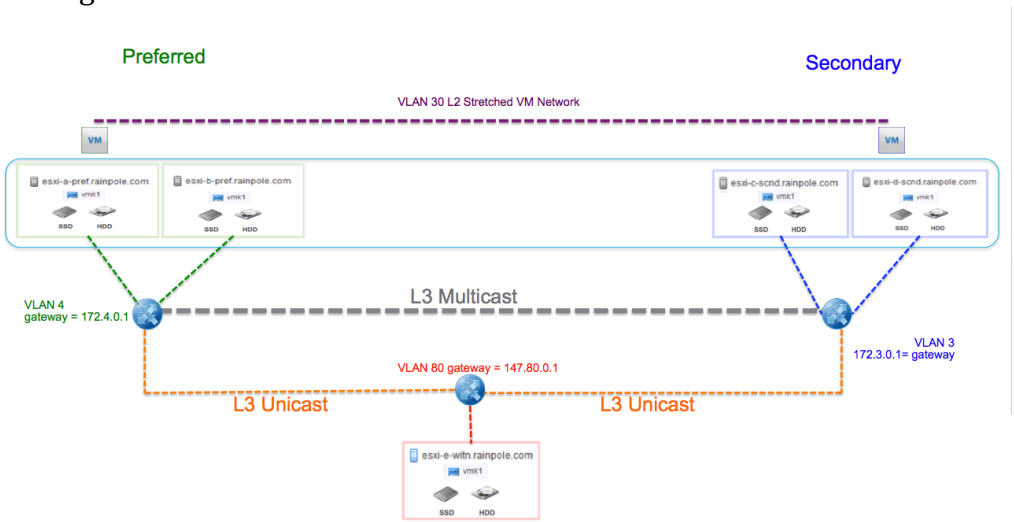


Figure 14.2: Virtual SAN Stretch Cluster network diagram

- This configuration uses L3 (route) for the Virtual SAN network between all sites.
- Static routes are required to enable communication between sites.
- The Virtual SAN network VLAN for the ESXi hosts on the preferred site is VLAN id 4. The gateway is 172.4.0.1.
- The Virtual SAN network VLAN for the ESXi hosts on the secondary site is VLAN id 3. The gateway is 172.3.0.1.
- The Virtual SAN network VLAN for the witness host on the witness site is VLAN id 80.
- The VM network is stretched L2 between the data sites. This is VLAN id 30. Since no VMs are run on the witness, there is no need to extend this network to the third site.

14.4 Preferred Site Details

In Virtual SAN Stretched Clusters, "preferred" site simply means the site that the witness will 'bind' to in the event of an inter-site link failure between the data sites. Thus, this will be the site with the majority of VM components, so this will also be the site where <u>all</u> VMs will run when there is an inter-site link failure between data sites.

In this example, Virtual SAN traffic is enabled on vmk1 on the hosts on the preferred site, which is sitting on routable VLAN 4.

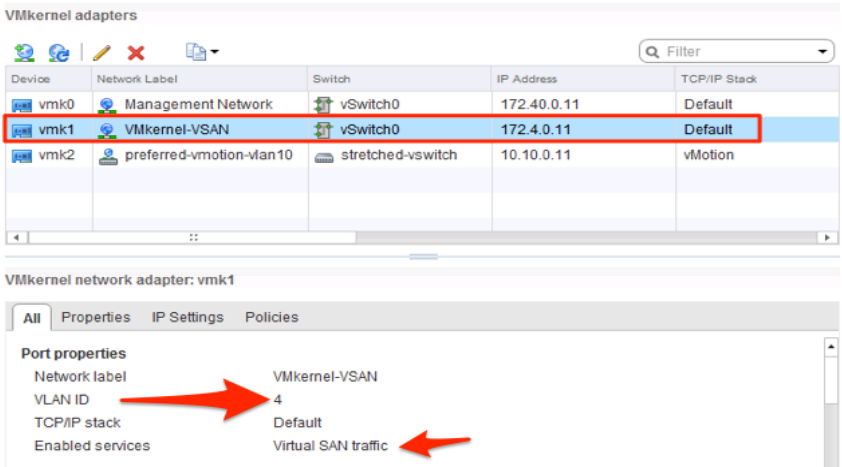


Figure 14.3: Virtual SAN preferred site networking details

Static routes need to be manually configured on these hosts. This is because the default gateway is on the management network, and if the preferred site hosts tried to communicate to the secondary site hosts, the traffic would be routed via the default gateway and thus via the management network. Since the management network and the Virtual SAN network are entirely isolated, there would be no route.

Since this is L3 everywhere, including between the data sites, the Virtual SAN interface on the preferred site, vmk1, has to route to “Secondary site (VLAN 3)” and “Witness Site (VLAN 80)”.

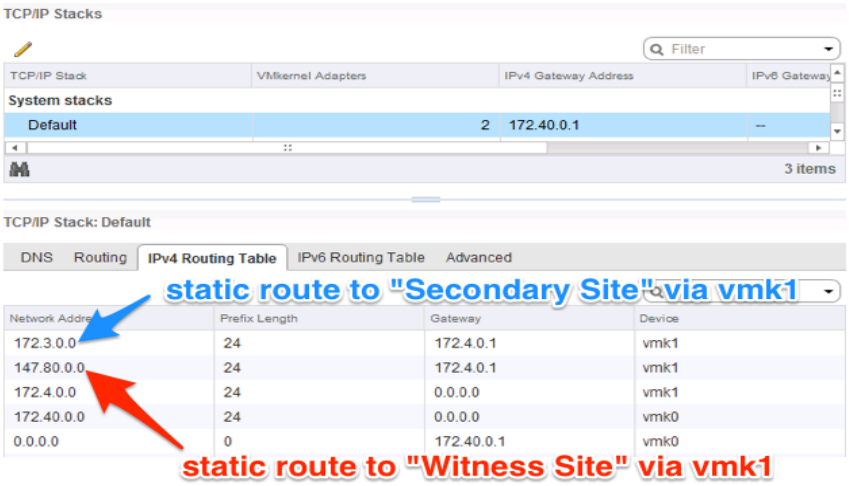


Figure 14.4: Primary site routing table with static routes to remote sites

14.4.1 Commands to Add Static Routes

The following command is used to add static routes is as follows:

```
esxcli network ip route ipv4 add -n REMOTE-NETWORK  -g LOCAL-GATEWAY
```

To add a static route from a preferred host to hosts on the secondary site in this POC:

```
esxcli network ip route ipv4 add -n 172.3.0.0/24 -g 172.4.0.1
```

To add a static route from a preferred host to the witness host in this POC:

```
esxcli network ip route ipv4 add –n 147.80.0.0/24 –g 172.4.0.1
```

Note: L3 Multicast routing must be enabled between VLAN 3 and 4. This is configured on the physical switch or router.

14.5 Secondary Site Details

The secondary site is the site that contains ESXi hosts whose objects do not bind with the witness components in the event of an inter-site link failure. However that is the only significant difference. Under normal conditions, the secondary site behaves exactly like the preferred site, and virtual machines may also be deployed there. In this POC, Virtual SAN traffic is enabled on vmk1, which is sitting on routable VLAN 3.

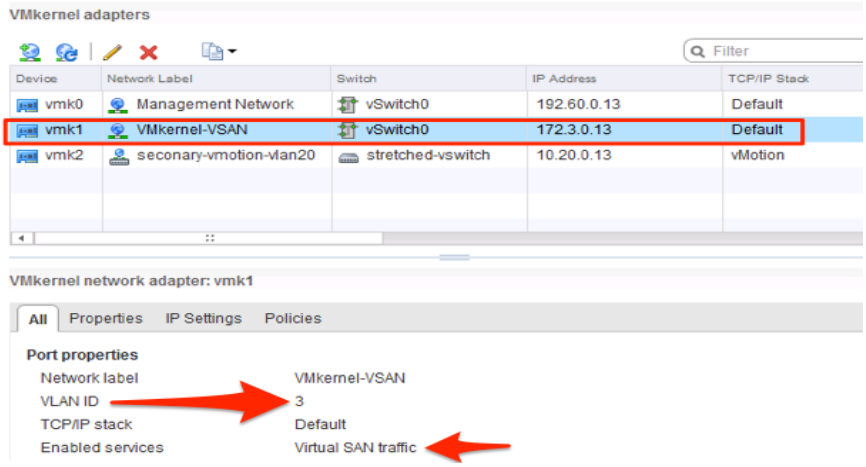


Figure 14.5: Virtual SAN secondary site networking details

Once again, static routes need to be manually configured on the Virtual SAN network interface, vmk1, to route to "Preferred site (VLAN 4)" and "Witness Site (VLAN 80)".

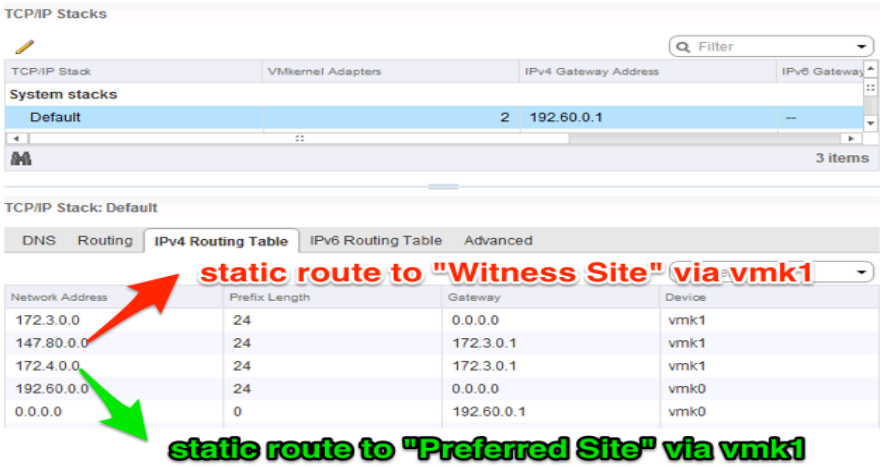


Figure 14.6: Secondary site routing table with static routes to remote sites

14.5.1 Commands to Add Static Routes

The following command is used to add static routes is as follows:

```
esxcli network ip route ipv4 add -n REMOTE-NETWORK  -g LOCAL-GATEWAY
```

To add a static route from a secondary host to hosts on the preferred site in this POC:

```
esxcli network ip route ipv4 add -n 172.4.0.0/24 -g 172.3.0.1
```

To add a static route from a secondary host to the witness host in this POC:

```
esxcli network ip route ipv4 add –n 147.80.0.0/24 –g 172.3.0.1
```

Note: L3 Multicast routing must be enabled between VLAN 3 and 4. This is configured on the physical switch or router.

14.6 A note on IGMP v3

IGMP Version 2, specified in [RFC-2236], added support for "low leave latency". That is, a reduction in the time it takes for a multicast router to learn that there are no longer any members of a particular group present on an attached network.

IGMP Version 3 adds support for "source filtering". That is, the ability for a system to report interest in receiving packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address.

It should be noted that in our POC testing with the DELL network switch, the Stretched Cluster would not configure properly after failures until the network switch was forced to talk IGMP v3 between VLANs.

Recommendation: Use IGMP v3 for multicast configurations.

14.7 Witness Site Details

The witness site only contains a single host for the Stretched Cluster, and the only VM objects stored on this host are "witness" objects. No data components are stored on the witness host. In this POC, we are using the witness appliance, which is an "ESXi host running in a VM". If you wish to use the witness appliance, it should be downloaded from VMware. This is because it is preconfigured with various settings, and also comes with a pre-installed license. Note that this download requires a login to My VMware.

Alternatively, customers can use a physical ESXi host for the appliance.

Virtual SAN traffic must be enabled on the Virtual SAN interface of the witness appliance, in this case vmk1, which is sitting on routable VLAN 80 (tagged on the underlying physical ESXi).

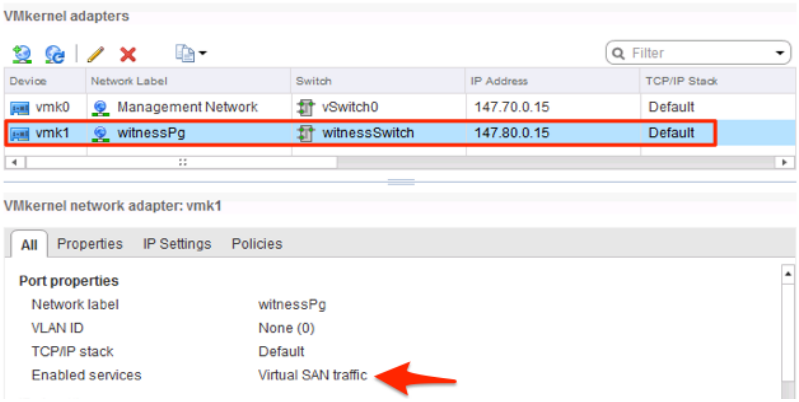


Figure 14.7: Virtual SAN witness host networking details

Once again, static routes should be manually configured on Virtual SAN vmk1 to route to "Preferred site (VLAN 4)" and "Secondary Site (VLAN 3)".

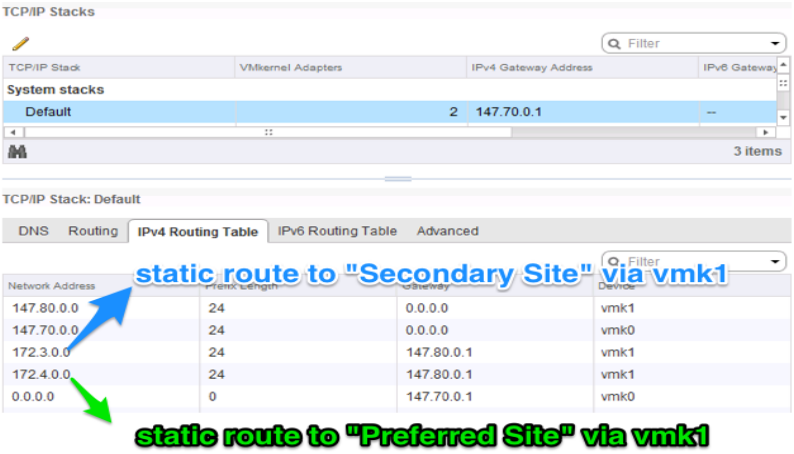


Figure 14.8: Witness host routing table with static routes to remote sites

14.7.1 Commands to Add Static Routes

The following command is used to add static routes is as follows:

```
esxcli network ip route ipv4 add -n REMOTE-NETWORK  -g LOCAL-GATEWAY
```

To add a static route from the witness host to hosts on the preferred site in this POC:

```
esxcli network ip route ipv4 add -n 172.4.0.0/24 -g 172.80.0.1
```

To add a static route from the witness host to hosts on the secondary site in this POC:

```
esxcli network ip route ipv4 add –n 147.3.0.0/24 –g 172.80.0.1
```

Note: L3 Multicast is not required for Witness Virtual SAN Traffic. Also VLAN tagging is enabled on ESXi host hosting witness appliance.

14.8 vSphere HA Settings

vSphere HA plays a critical part in Stretched Cluster. HA is required to restart virtual machines on other hosts and even the other site depending on the different failures that may occur in the cluster. The following section covers the recommended settings for vSphere HA when configuring it in a Stretched Cluster environment.

14.8.1 Response to Host Isolation

The recommendation is to "Power off and restart VMs" on isolation, as shown below. In cases where the virtual machine can no longer access the majority of its object components, it may not be possible to shut down the guest OS running in the virtual machine. Therefore the "Power off and restart VMs" option is recommended.

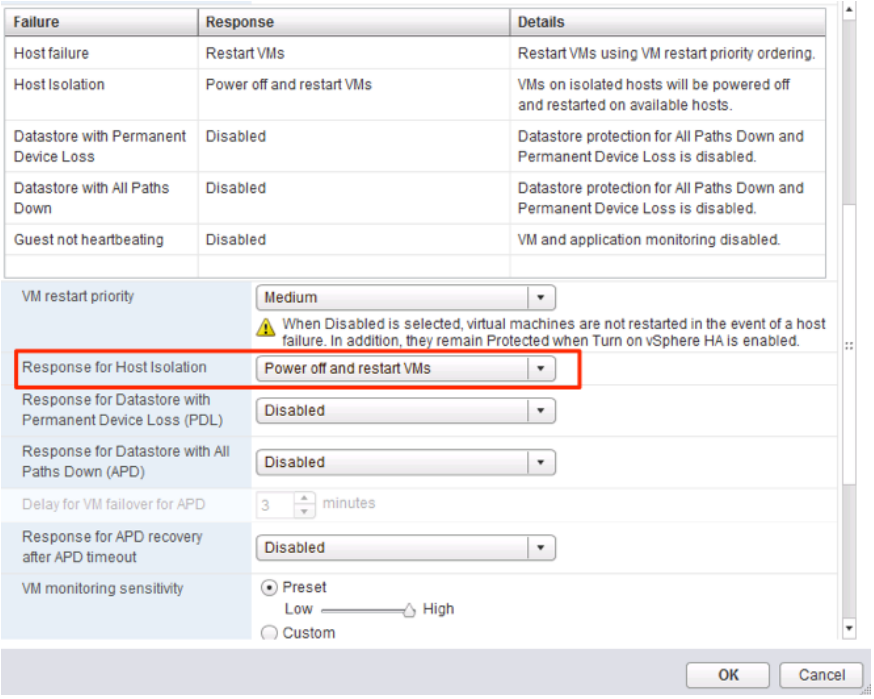


Figure 14.9: vSphere HA Host Isolation recommended setting

14.8.2 Admission Control

If a full site fails, the desire is to have all virtual machines run on the remaining site. To allow a single data site to run all virtual machines if the other data site fails, the recommendation is to set Admission Control to 50% for CPU and Memory as shown below.

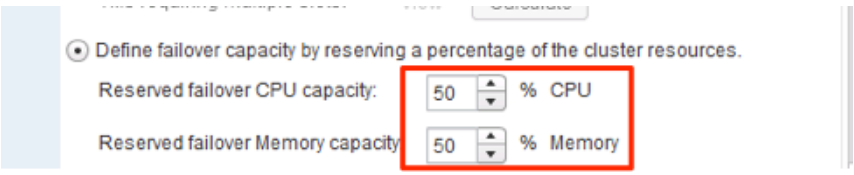


Figure 14.10: vSphere HA Admission Control setting recommendation

14.8.3 Advanced Settings

The default isolation address uses the default gateway of the management network. This will not be useful in a Virtual SAN Stretched Cluster, when the Virtual SAN network is broken. Therefore the default isolation response address should be turned off. This is done via the advanced setting *das.usedefaultisolationaddress* to false.

To deal with failures occurring on the Virtual SAN network, VMware recommends setting two isolation addresses, each of which is local to one of the data sites. In this POC, one address is on VLAN 4, which is reachable from the hosts on the preferred sites. The other address is on VLAN 3, which is reachable from the hosts on the secondary site. Use advance settings *das.isolationaddress0* and *das.isolationaddress1* to set these isolation addresses respectively.

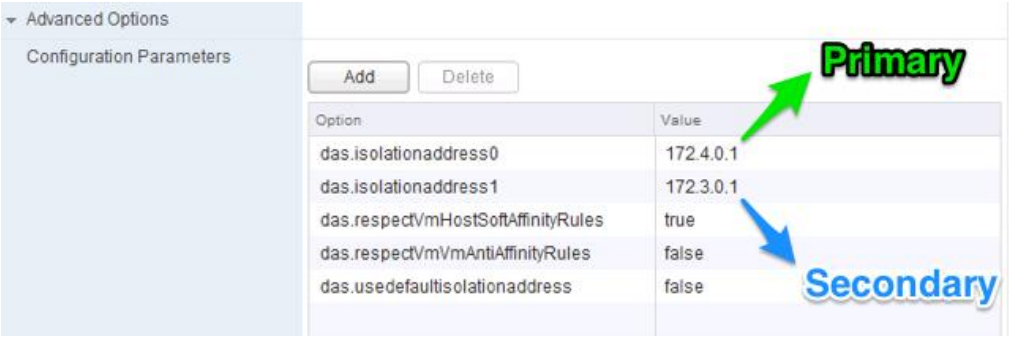


Figure 14.11: vSphere HA advanced options isolation address recommendations

These advanced settings are added in the Advanced Options > Configuration Parameter section of the vSphere HA UI. The other advanced settings get filled in automatically based on additional configuration steps. There is no need to add them manually.

14.9 VM Host Affinity Groups

The next step is to configure VM/Host affinity groups. This allows administrators to automatically place a virtual machine on a particular site when it is powered on. In the event of a failure, the virtual machine will remain on the same site, but placed on a different host. The virtual machine will be restarted on the remote site only when there is a catastrophic failure or a significant resource shortage.

To configure VM/Host affinity groups, the first step is to add hosts to the host groups. In this example, the Host Groups are named Preferred and Secondary, as shown below.

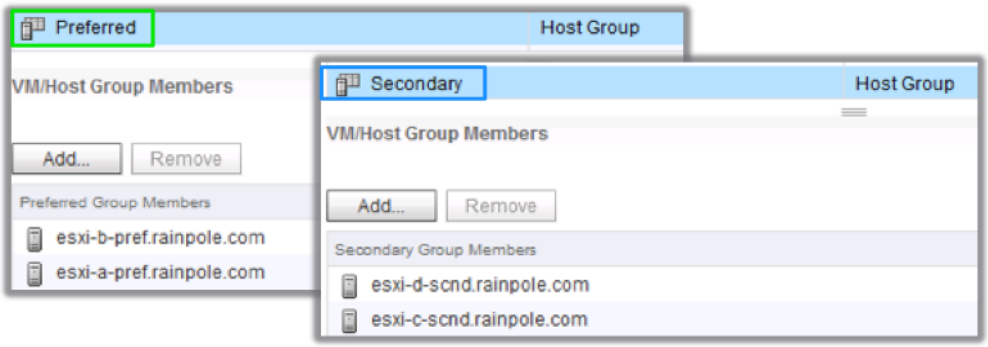


Figure 14.12: Host affinity groups

The next step is to add the virtual machines to the host groups. Note that these virtual machines must be created in advance.

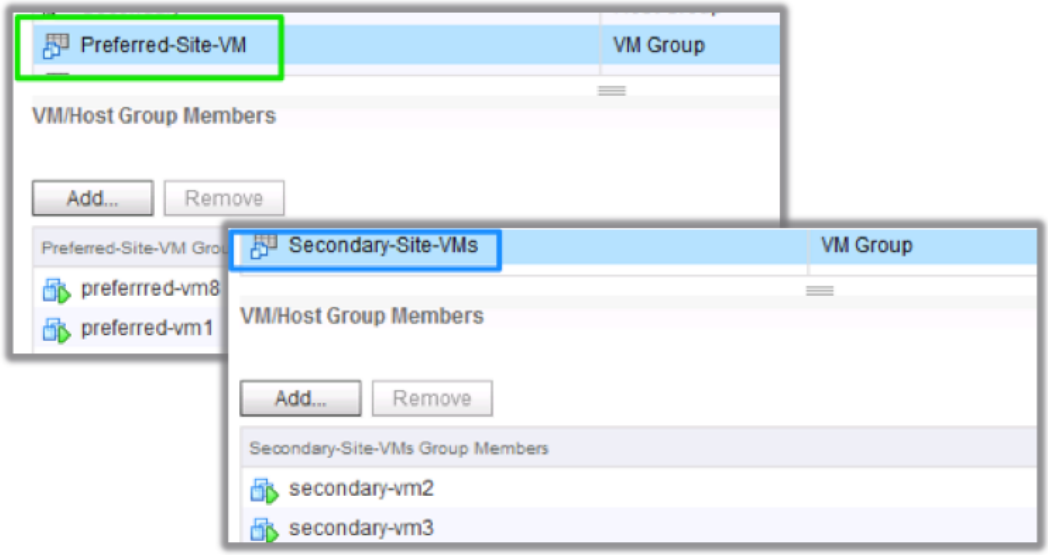


Figure 14.13: Host affinity groups with VMs

Note that these VM/Host affinity rules are "should" rules and not "must" rules. "Should" rules means that every attempt will be made to adhere to the affinity rules. However, if this is not possible (due lack of resources), the other site will be used for hosting the virtual machine.

Also note that the vSphere HA rule settings is set to "should". This means that if there is a catastrophic failure on the site to which the VM has affinity, HA will restart the virtual machine on the other site. If this was a "must" rule, HA would not start the VM on the other site.

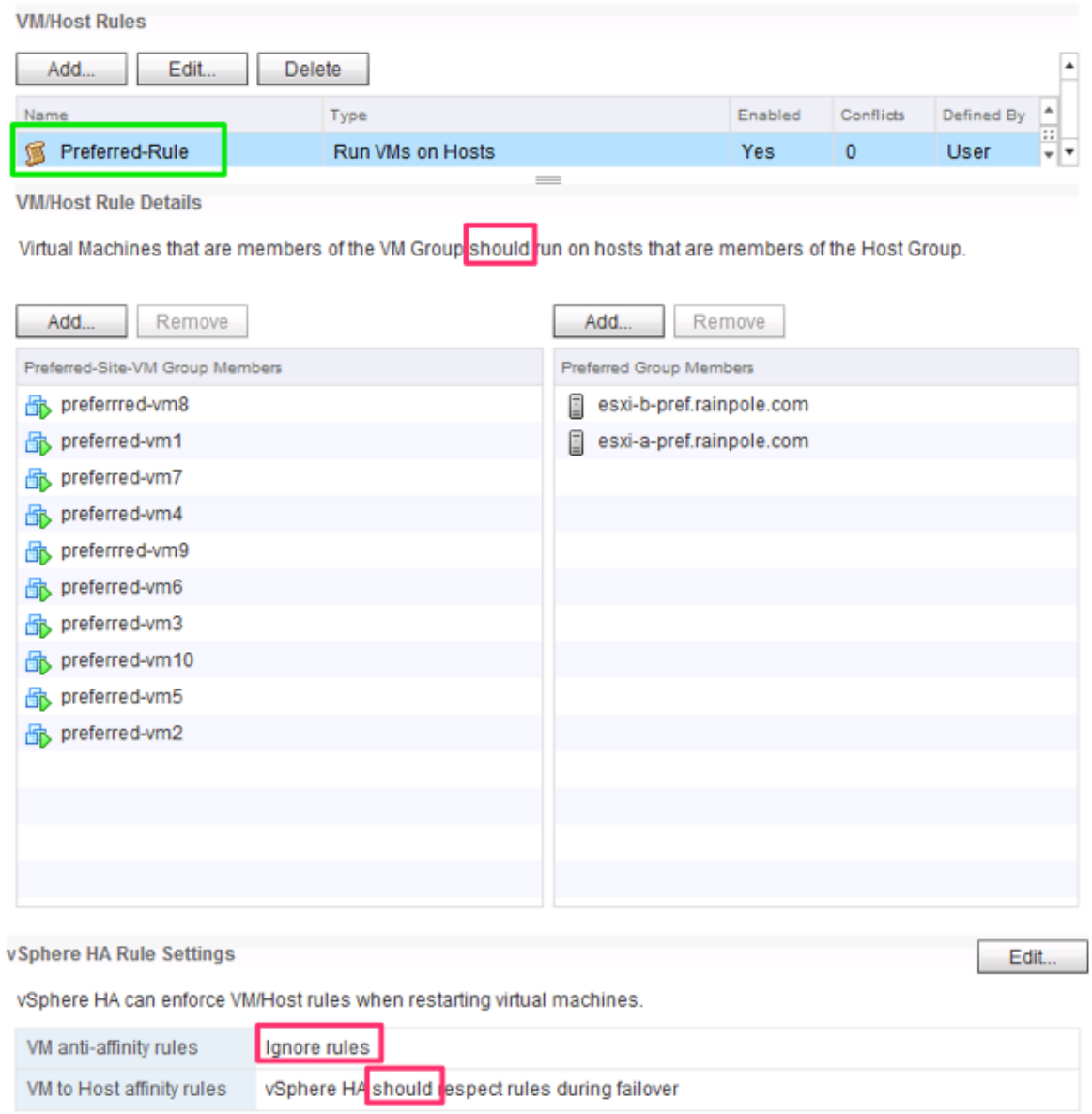


Figure 14.14: Set vSphere HA VM to Host affinity rules to "should", not "must"

The same settings are necessary on both the primary VM/Host group and the secondary VM/Host group.

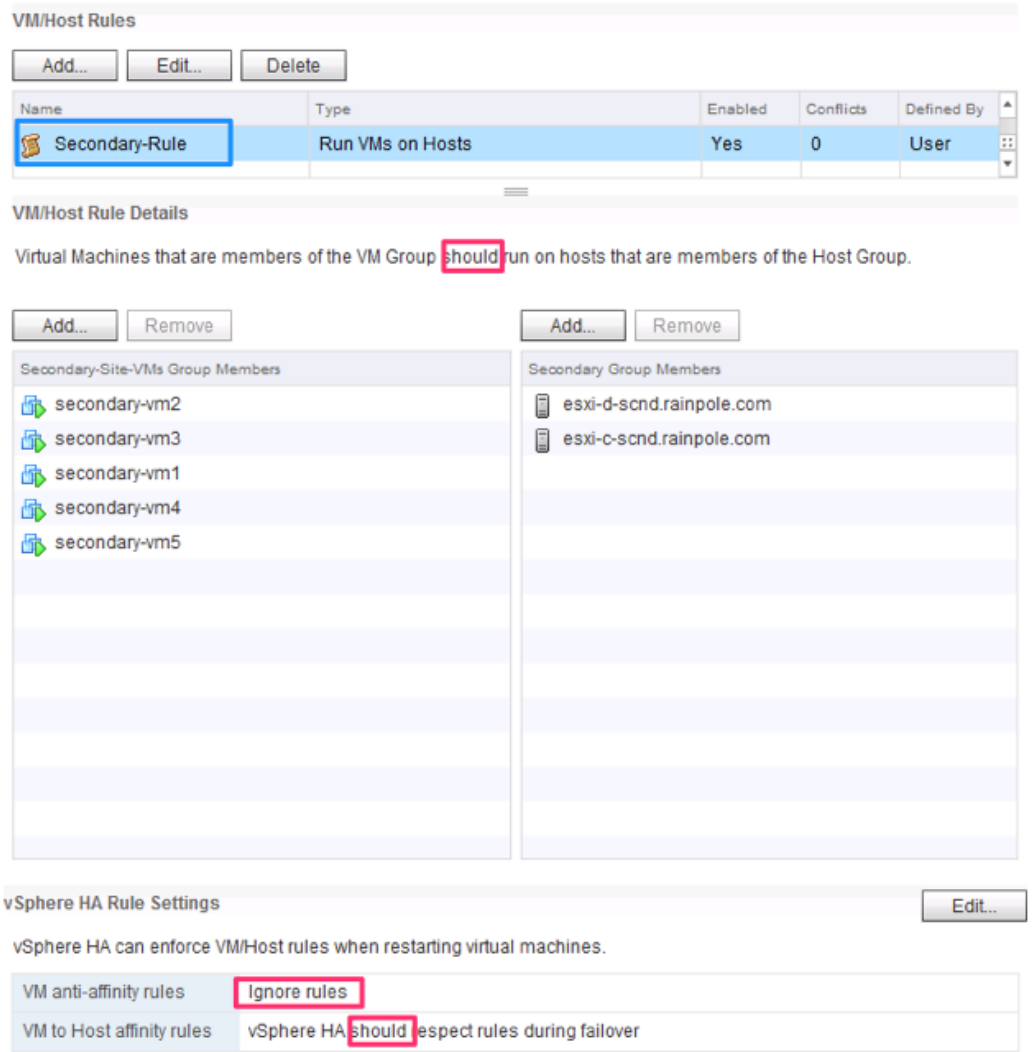


Figure 14.15: Set vSphere HA VM to Host affinity rules to “should” on Secondary too

14.10 DRS Settings

In this POC, partially automated mode has been chosen. However, this could be set to Fully Automated if customers wish, but note that it should be changed back to partially automated when a full site failure occurs. This is to avoid failback of VMs occurring whilst rebuild activity is still taking place. More on this later.

Figure 14.16: Virtual SAN stretch cluster DRS settings

15. Virtual SAN Stretched Cluster Network Failover Scenarios

In this section, we will look at how to inject various network failures in a Virtual SAN Stretched Cluster configuration. We will see how the failure manifests itself in the cluster, focusing on the Virtual SAN health check and the alarms/events as reported in the vSphere web client.

15.1 Network Failure between Secondary Site and Witness

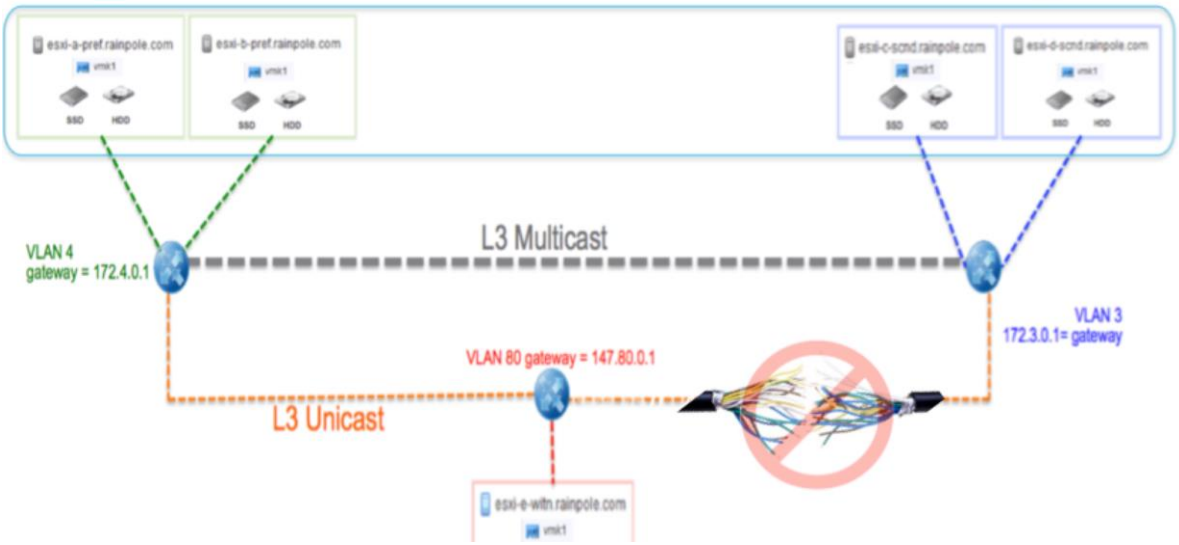


Figure 15.1: Path failure between secondary site and witness site

15.1.1 Trigger the Event

To make the secondary site lose access to the witness site, one can simply remove the static route on the witness host that provides a path to the secondary site.

On witness host issue:

```
esxcli network ip route ipv4 remove -g 147.80.0.1 -n 172.3.0.0/24
```

On secondary host(s) issue:

```
esxcli network ip route ipv4 remove -g 172.3.0.1 -n 147.80.0.0/24
```

15.1.2 Cluster Behavior on Failure

To begin with, the **Cluster Summary** view shows one configuration issue related to 0 witness hosts.

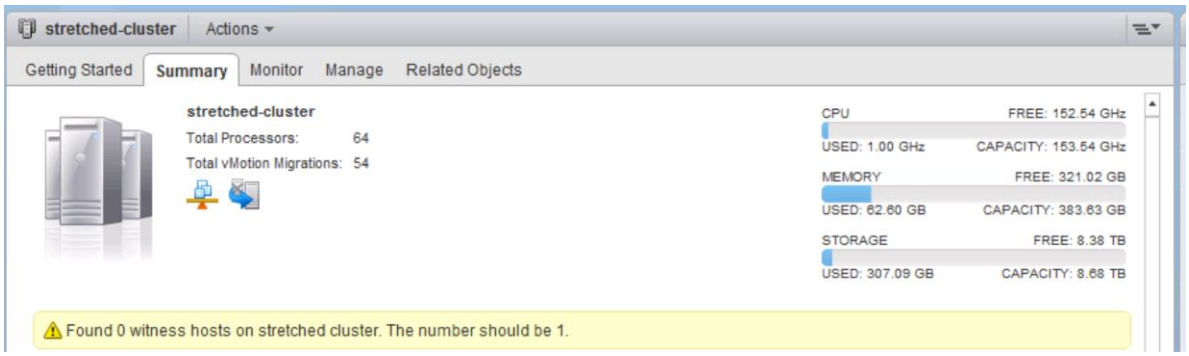


Figure 15.2: Cluster summary view – 0 witness hosts

This same event is visible in the Cluster > Monitor > Issues > All Issues view.

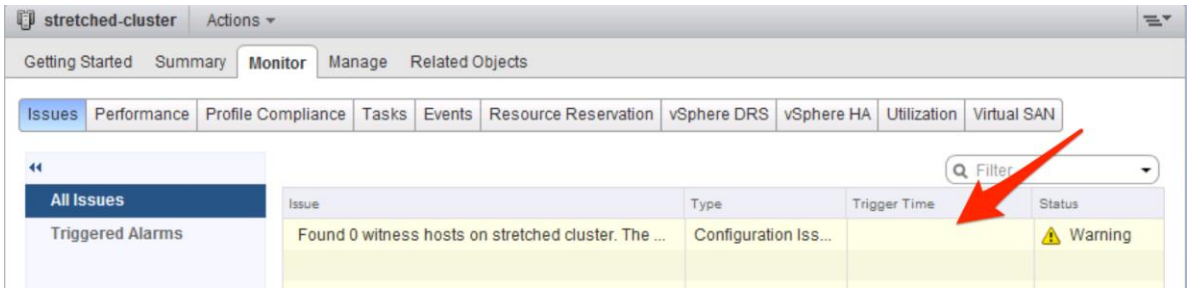


Figure 15.3: Cluster Issue – missing witness

Note that this event may take some time to trigger. Next, looking at the **health check** alarms, a number of them get triggered (Triggering alarms from health check test failures is a new feature in Virtual SAN 6.1).

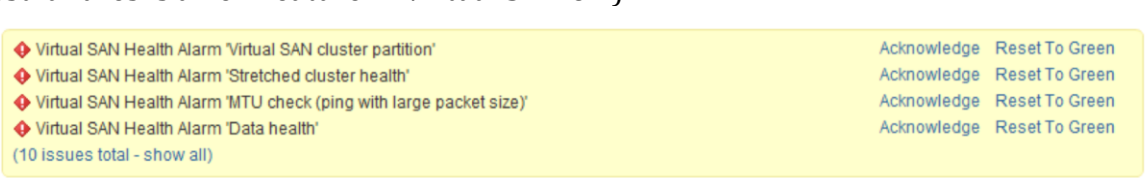


Figure 15.4: Virtual SAN Health Alarms triggered

In the **Cluster summary** view, an error is also shown. This directs the administrator to go to "Monitor Virtual SAN health".

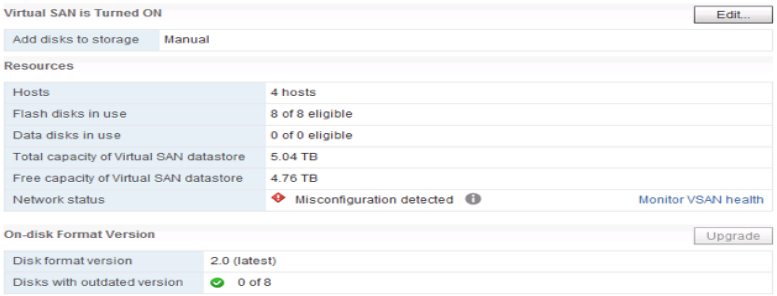


Figure 15.5: Virtual SAN cluster summary view

On navigating to the Virtual SAN Health > Monitor view, there are a lot of checks showing errors. One should also note that there is a set of new Stretched Cluster health checks in 6.1. These are also failing.

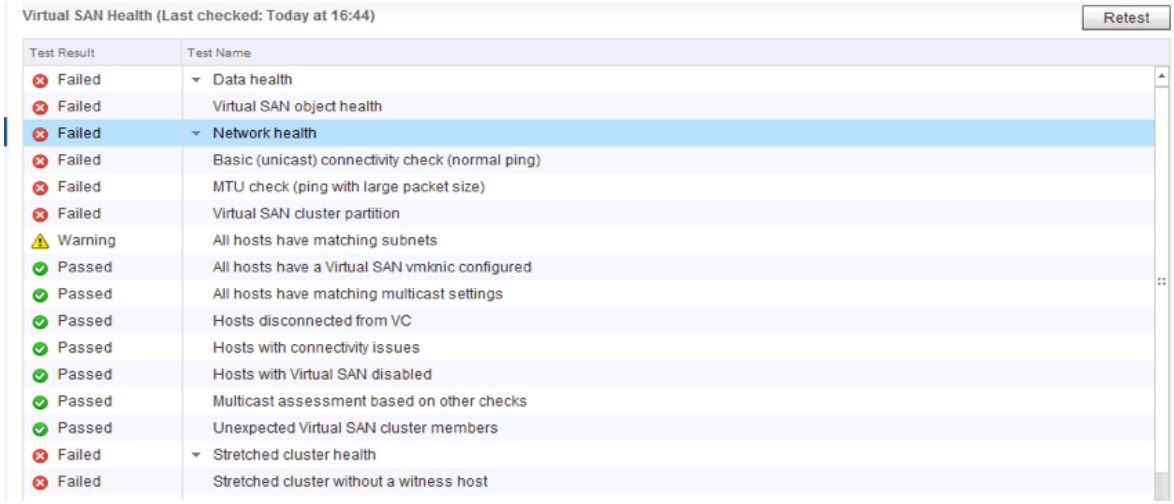


Figure 15.6: Virtual SAN Health Check detects the problems

One final place to examine is the virtual machines. Navigate to a VM on the secondary site, then Monitor > Policies > Physical Disk Placement. It should show the witness absent from secondary site perspective. However the virtual machines should still be running and fully accessible.

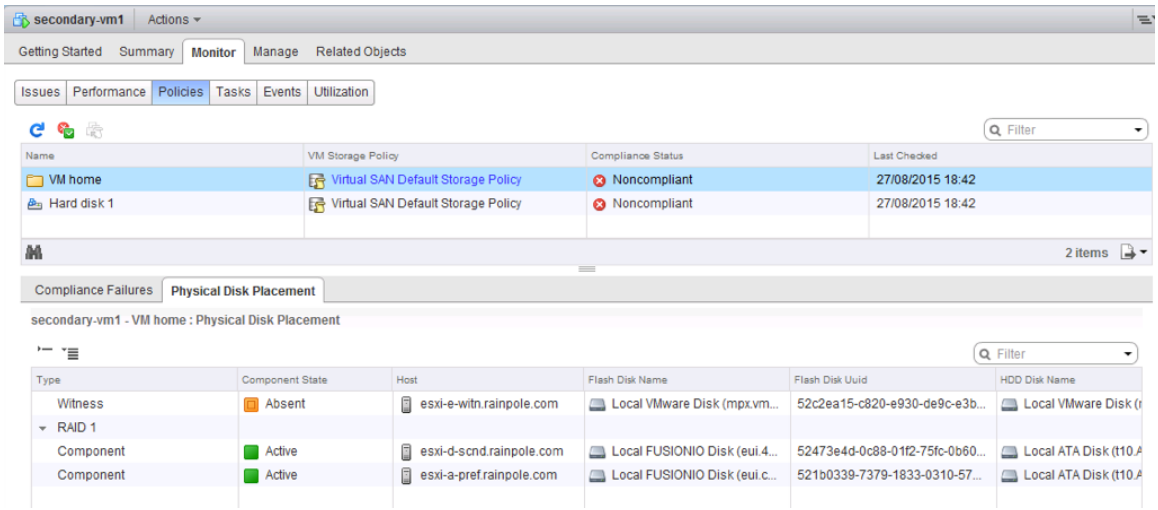


Figure 15.7: VM shows the witness component is absent

Returning to the health check client, selecting "Basic (unicast) connectivity check (normal ping), you can see that the Secondary Site can't talk to witness or vice versa.

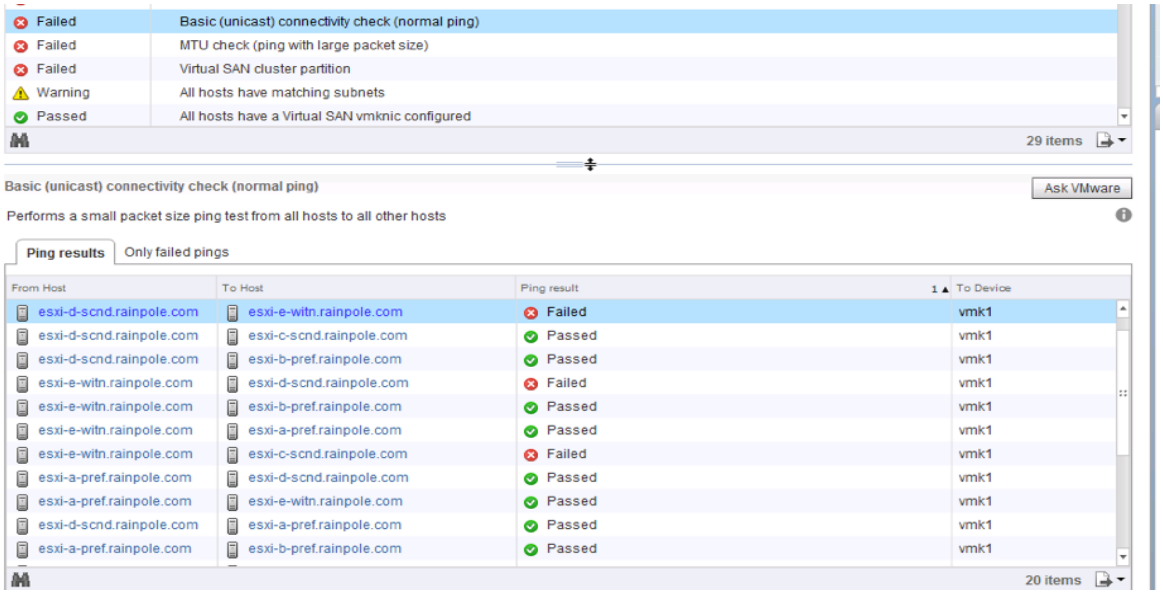


Figure 15.8: Virtual SAN Health Check ping test results

15.1.3 Conclusion

Loss of the witness does not impact the running virtual machines on the secondary site. There is still a quorum of components available per object, available from the data sites. Since there is only a single witness host/site, and only three fault domains, there is no rebuilding/resyncing of objects.

15.1.4 Repair the Failure

Add back the static routes that were removed earlier, and rerun the health check tests. Verify that all tests are passing before proceeding. **Remember to test one thing at a time**.

15.2 Network Failure between Preferred Site and Witness

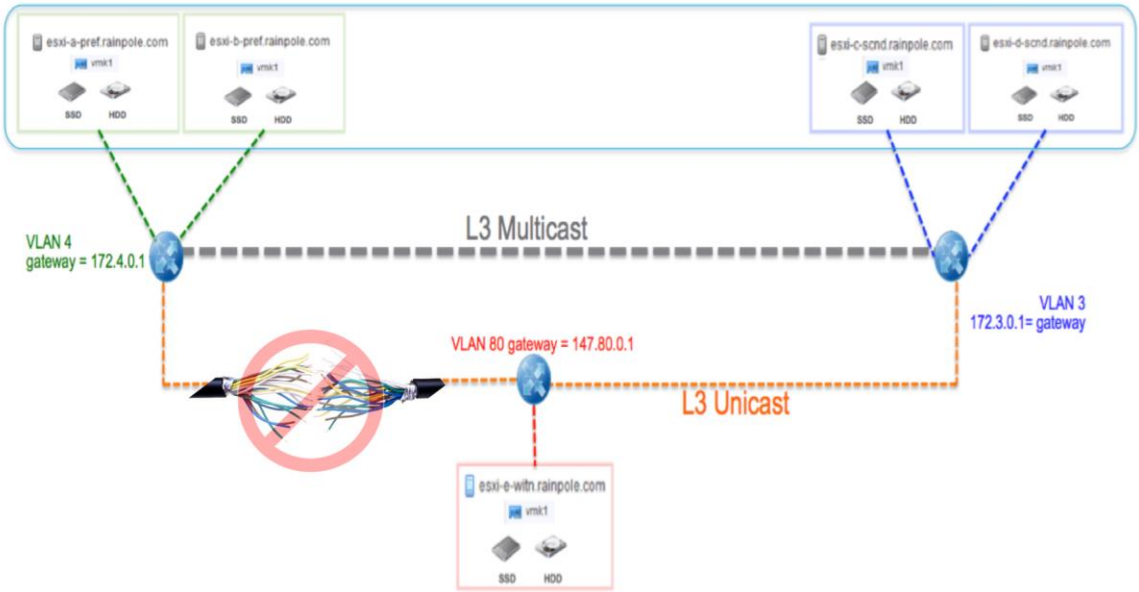


Figure 15.9: Path failure between preferred site and witness site

15.2.1 Trigger the Event

To make the preferred site lose access to the witness site, one can simply remove the static route on the witness host that provides a path to the preferred site.

On witness host issue:

```
esxcli network ip route ipv4 remove -g 147.80.0.1 -n 172.4.0.0/24
```

On preferred host(s) issue:

```
esxcli network ip route ipv4 remove -g 172.4.0.1 -n 147.80.0.0/24
```

15.2.2 Cluster Behavior on Failure

As per the previous test, it may take some time for alarms to trigger when this event occurs. However, the events are similar to those seen previously.

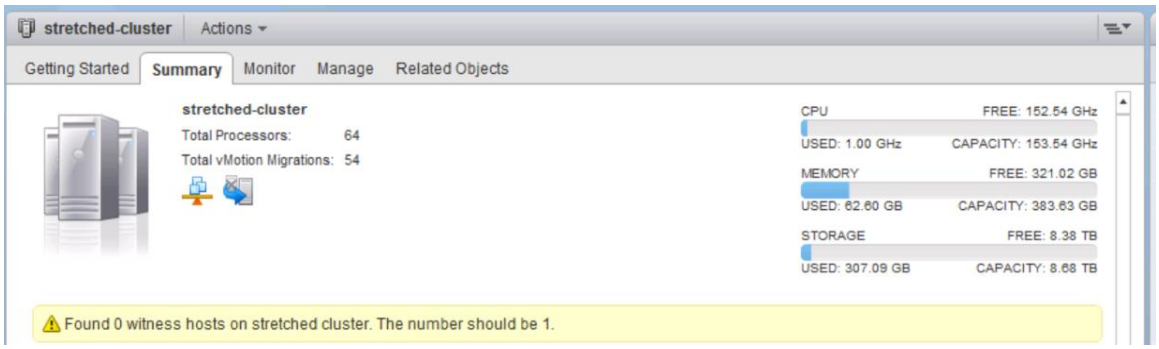


Figure 15.10: Cluster summary view – 0 witness hosts

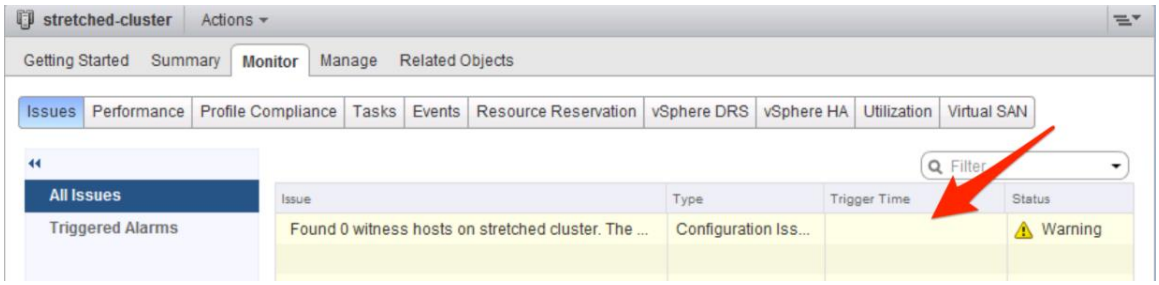


Figure 15.11: Cluster issue – missing witness

One can also see various health checks fail, and their associated alarms being raised.

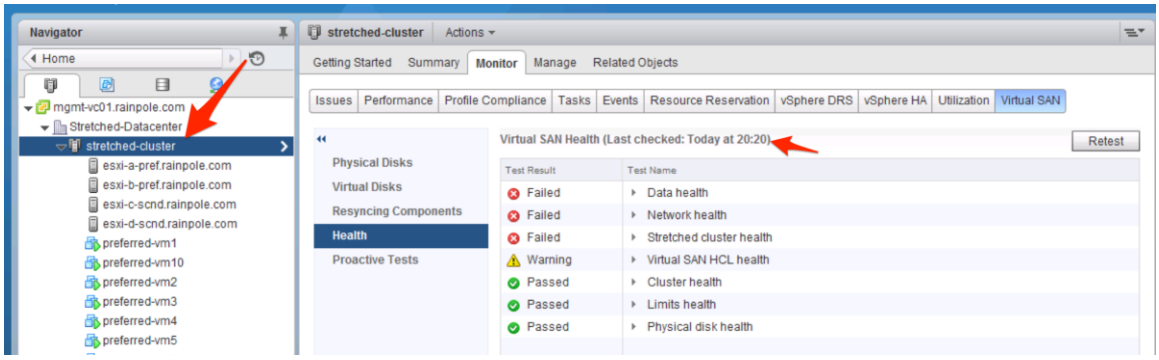


Figure 15.12: Virtual SAN Health Check detects the problems

Just like the previous test, the witness component goes absent.

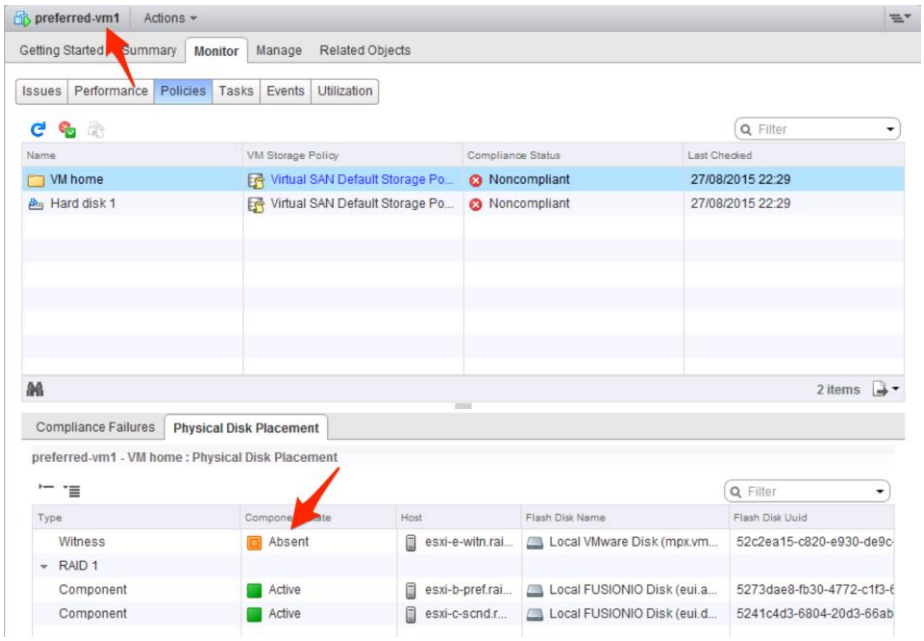


Figure 15.13: VM's storage policy is out of compliance

We did not look at the "Data health" health check during the previous test. If this health check **"Virtual SAN object health"** is selected, it displays X number of objects with "reduced-availability-with-no-rebuild-delay-timer". In this POC, there are 52 objects impacted by the failure.

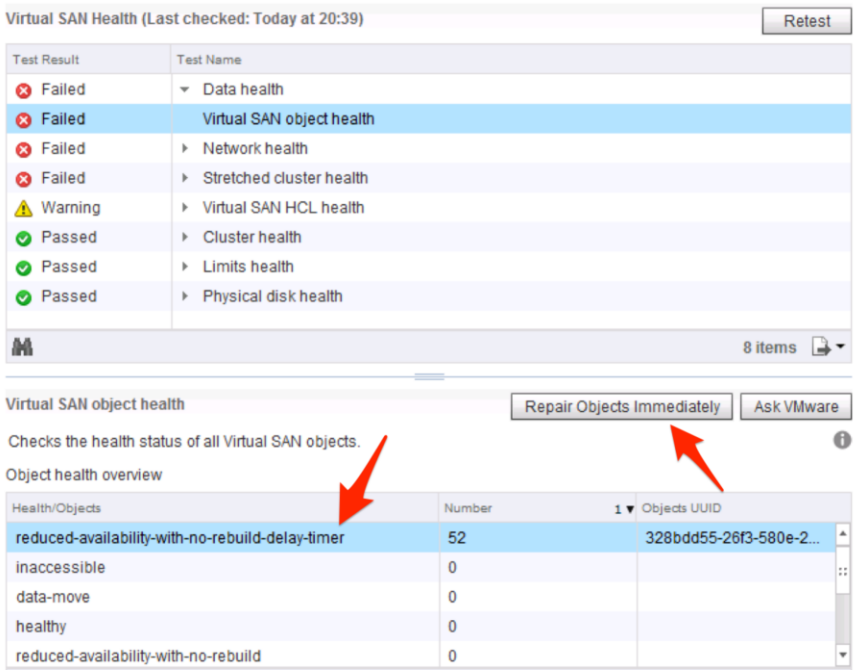


Figure 15.14: Virtual SAN Health Check for object health

This health check behavior appears whenever components go 'absent' and Virtual SAN is waiting for the 60-minute *clomd* timer to expire before starting any rebuilds. If an administrator clicks on "Repair Objects Immediately", the objects switch state and now the objects are no longer waiting on the timer, and will start to rebuild immediately under general circumstances. However in this POC, with only three fault domains and no place to rebuild witness components, there is no syncing/rebuilding.

15.2.3 Conclusion

Just like the previous test, a witness failure has no impact on the running virtual machines on the preferred site. There is still a quorum of components available per object, as the data sites can still communicate. Since there is only a single witness host/site, and only three fault domains, there is no rebuilding/resyncing of objects.

15.2.4 Repair the Failure

Add back the static routes that were removed earlier, and rerun the health check tests. Verify that all tests are passing before proceeding. **Remember to test one thing at a time.**

15.3 Network Failure between Witness and Both Data Sites

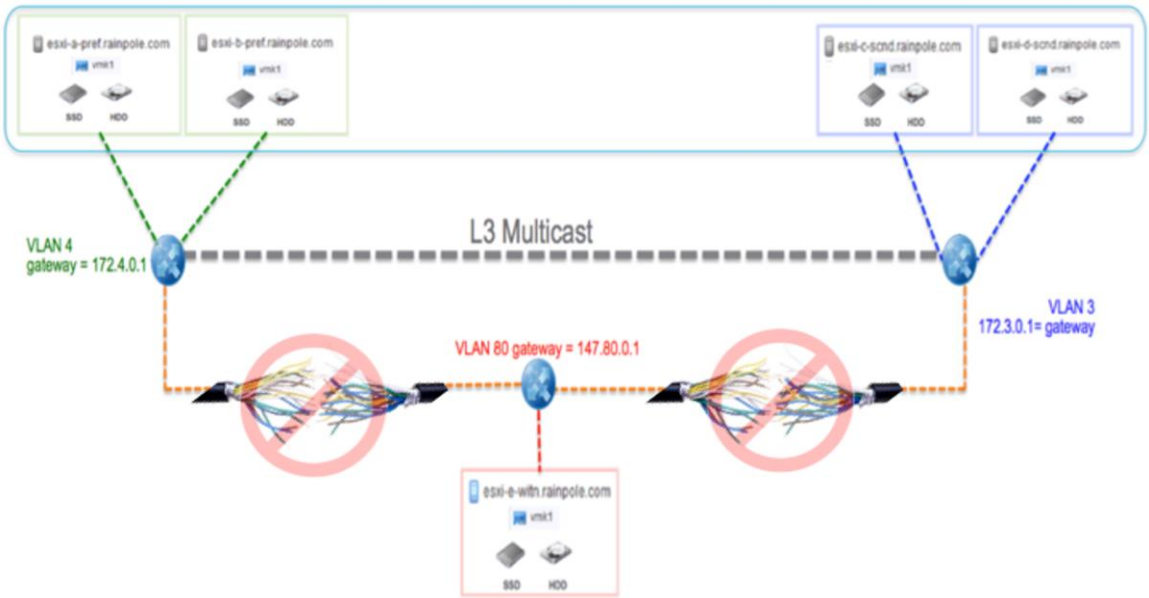


Figure 15.15: Complete witness site outage

15.3.1 Trigger the Event

To introduce a network failure between the preferred and secondary data sites and the witness site, one can simply remove the static route on the witness host that provides a path to both the preferred and secondary sites, and remove the static routes to the witness on the preferred and secondary hosts.

On Witness host issue:

```
esxcli network ip route ipv4 remove -g 147.80.0.1 -n 172.3.0.0/24
esxcli network ip route ipv4 remove -g 147.80.0.1 -n 172.4.0.0/24
```

On Preferred host(s) issue:

```
esxcli network ip route ipv4 remove -g 172.4.0.1 -n 147.80.0.0/24
```

On Secondary host(s) issue:

```
esxcli network ip route ipv4 remove -g 172.3.0.1 -n 147.80.0.0/24
```

15.3.2 Cluster Behavior on Failure

The events observed are for the most part identical to those observed in failure scenario #1 and #2.

15.3.3 Conclusion

When the Virtual SAN network fails between the witness site and both the data sites (as in the witness site fully losing its WAN access), it does not impact the running virtual machines. There is still a quorum of components available per object, available from the data sites. However, as explained previously, since there is only a single witness host/site, and only three fault domains, there is no rebuilding/resyncing of objects.

15.3.4 Repair the Failure

Add back the static routes that were removed earlier, and rerun the health check tests. Verify that all tests are passing. **Remember to test one thing at a time.**

16. Further Information

16.1 VMware Virtual SAN Community

16.2 Links to Existing Documentation

- VMware Virtual SAN Resources
- Administering VMware Virtual SAN
- VMware Compatibility Guide
- VMware Virtual SAN Diagnostics and Troubleshooting Reference Manual
- VMware Virtual SAN 6.0 Design and Sizing Guide
- Virtual SAN Hosted Evaluation
- VMware Virtual SAN Health Check Plugin Guide

16.3 VMware Support

- My VMware
- How to file a Support Request in My VMware
- Location of log files for VMware Products
- Location of ESXi 5.1 and 5.5 log files
- Collecting Virtual SAN support logs and uploading to VMware

Appendix A—Fault Domains

In this four-node environment, we now look at the benefits of failure domains, a new feature introduced in vSphere 6.0. In this scenario, we will assume that the 4 nodes are in two racks, something as follows.

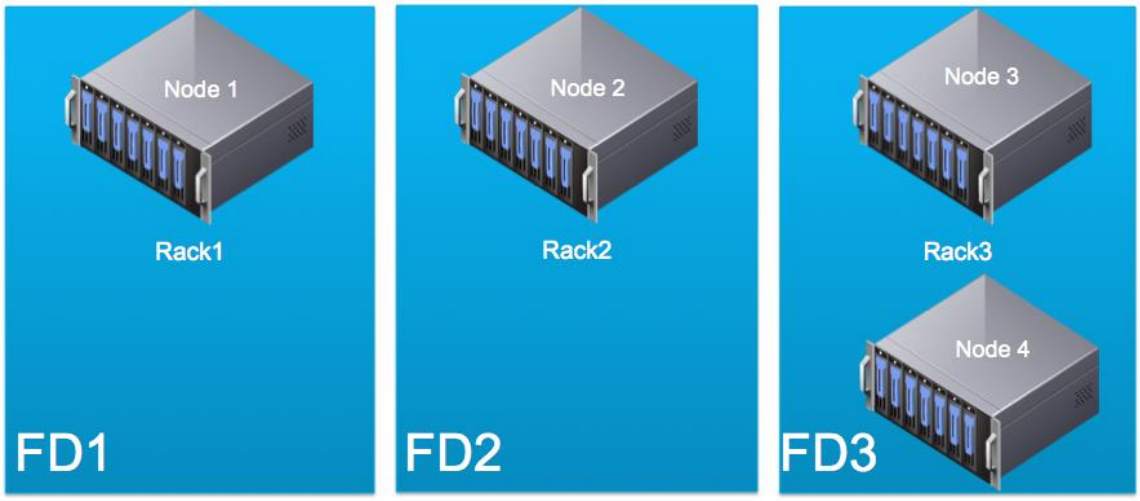


Figure A.1 Fault Domains

The objective now is to match the rack with fault domains. This implies that if there is a rack failure, the virtual machine components will have been distributed in a fashion such that they remain available even when a complete rack fails.

A1. Setting up Fault Domains

As shown above, we will create three fault domain, two of which only contain a single host, but one which contains two hosts. Navigate to the Manage tab, and under Virtual SAN select Fault Domains as shown below. Initially, there are no hosts in any fault domains.

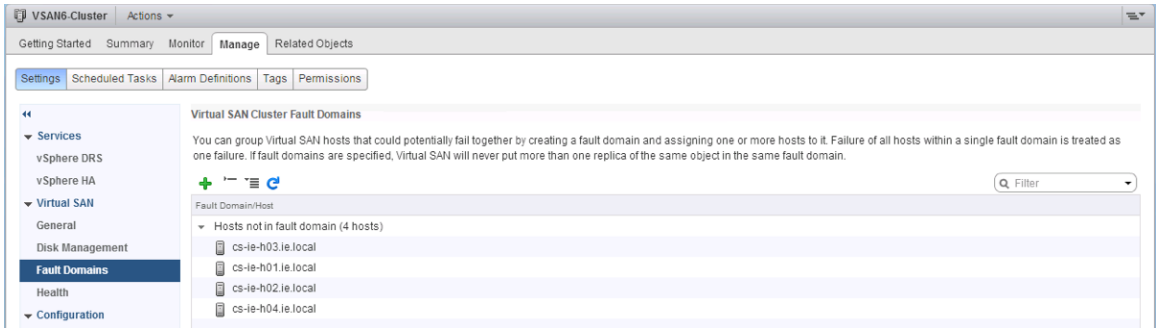


Figure A.2 No hosts in Fault Domains

Click on the green "+" symbol to create a fault domain. Initially, we will add host cs-ie-h01.ie.local to the first fault domain. Let's call the domain FD1.

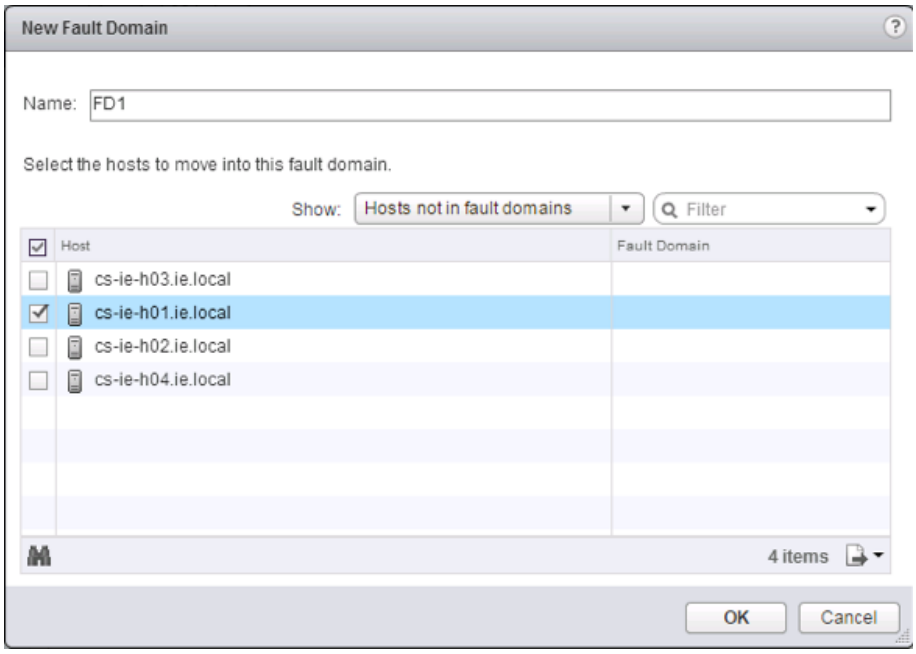


Figure A.3 Add single host to Fault Domain FD1

Repeat this operation for the second fault domain, but this time add host cs-ie-h02.ie.local to this domain FD2. For the third fault domain, add the remaining two hosts, cs-ie-h03 and cs-ie-h04 as shown here.

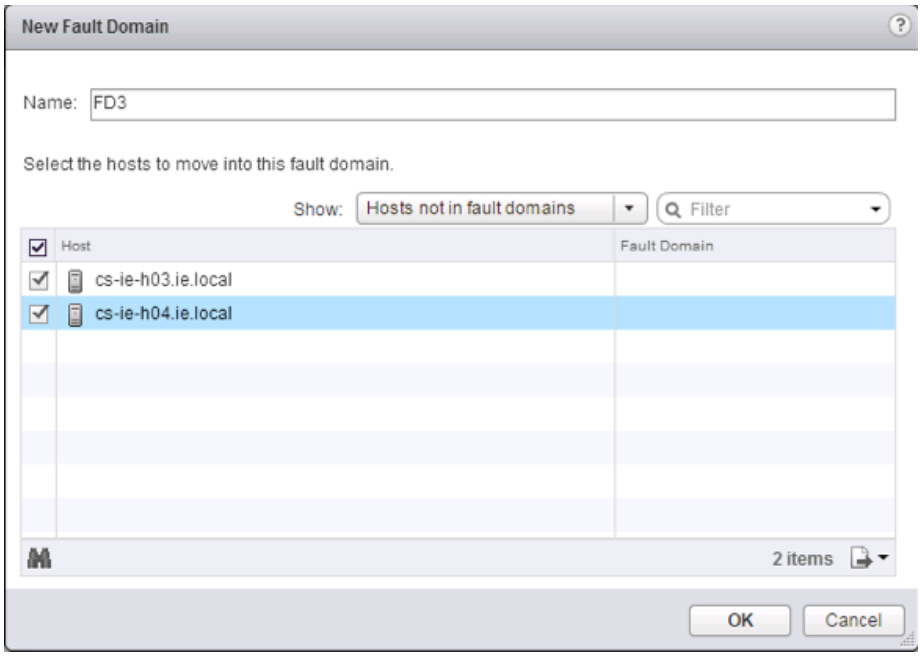


Figure A.4 Add two hosts to third Fault Domain FD3

At this point, three fault domains have been created.

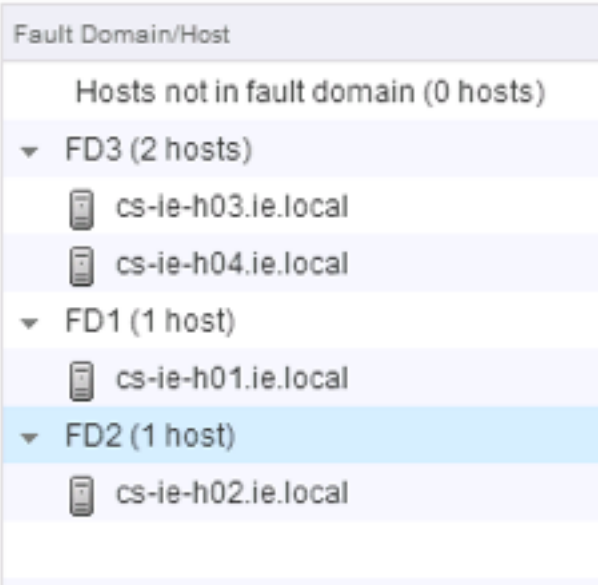


Figure A.5 Fault Domain Overview

A2. Create a Policy to Leverage Fault Domains

The next step is to create a VM storage policy that highlights the behavior of fault domains. In the event of a failure of any single rack, there should still be enough components available belonging to the VM to continue running. In essence, there should still be a full copy of the data even when a rack fails. Let's create a policy so that we can observe how a VM's components. We have chosen a policy that has *NumberOfFailuresToTolerate* = 1 and *NumberOfDiskStripesPerObject* = 3.

We have already created policies back in chapter 9. Here are the steps once again.

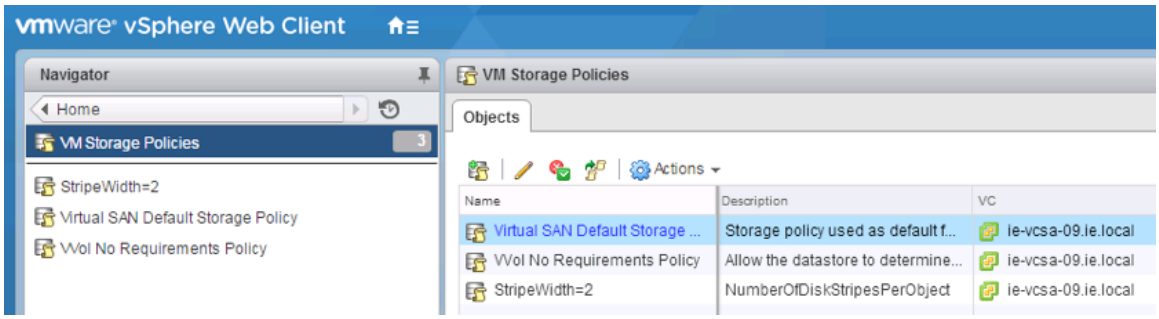


Figure A.6 Navigate to VM Storage Policies

Click on the "Create New Policy" icon. Give it a name and an optional description.

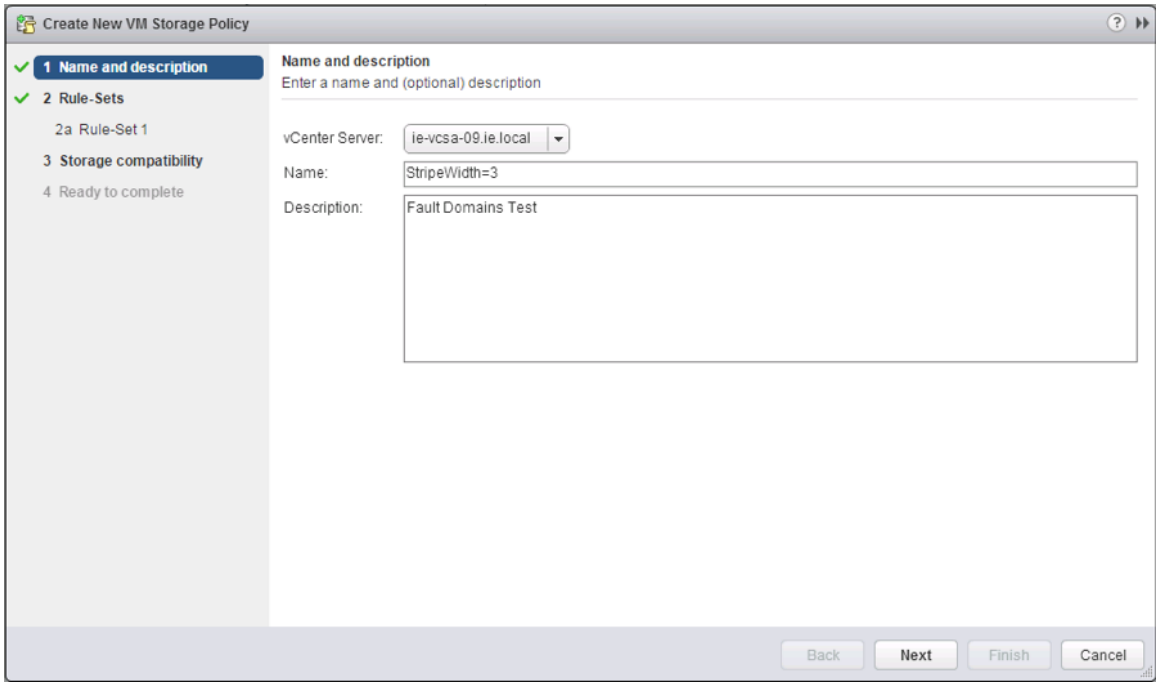


Figure A.7 Give a name and description to the policy

Click through the rule-set description.

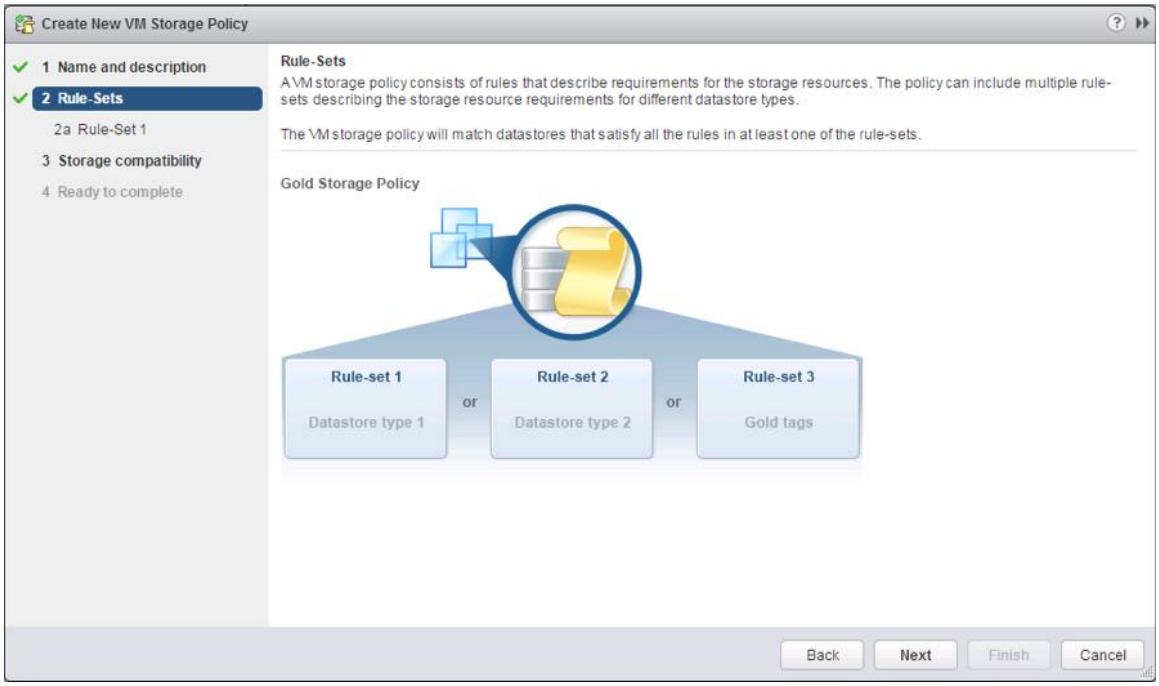


Figure A.8 Rule-set description

In the Rule-Set 1 window, select Virtual SAN as the "Rules based on data service". Then add the rule "Number of disk stripes per object" and set the value to 3. There is no need to add "Number of failures to tolerate" as this is automatically set to 1 for every policy unless you explicitly set it to a value of 0.

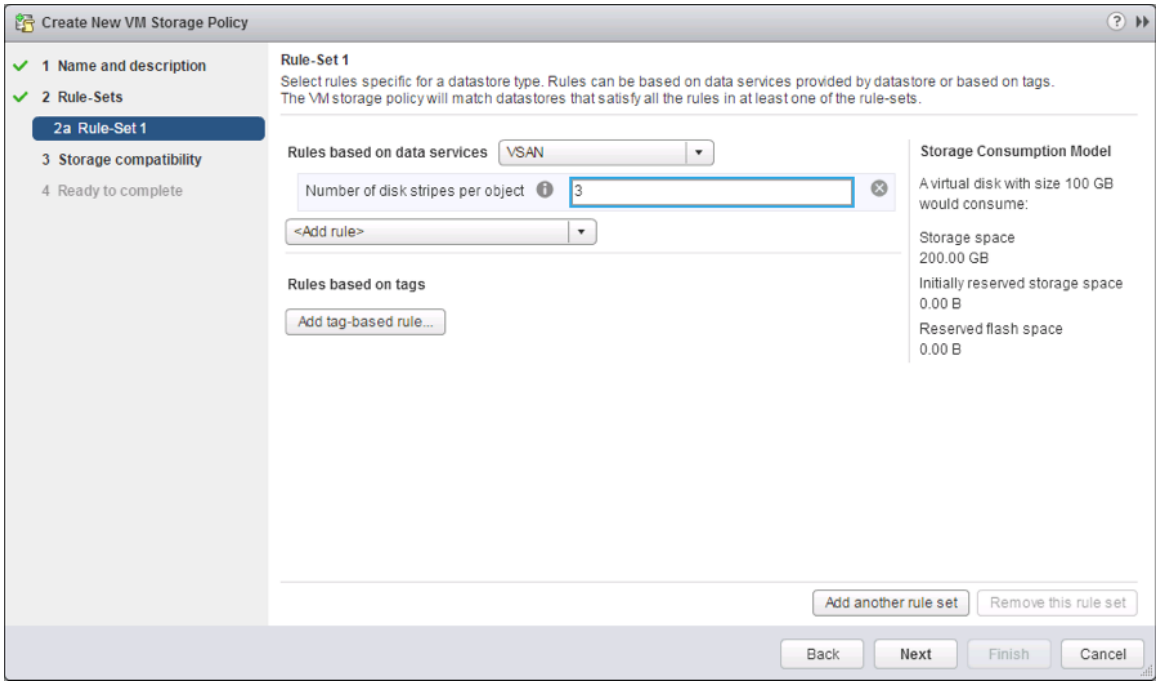


Figure A.9 Number of disk stripes per object

The Virtual SAN datastore should appear as compatible, in other words it understands the policy settings.

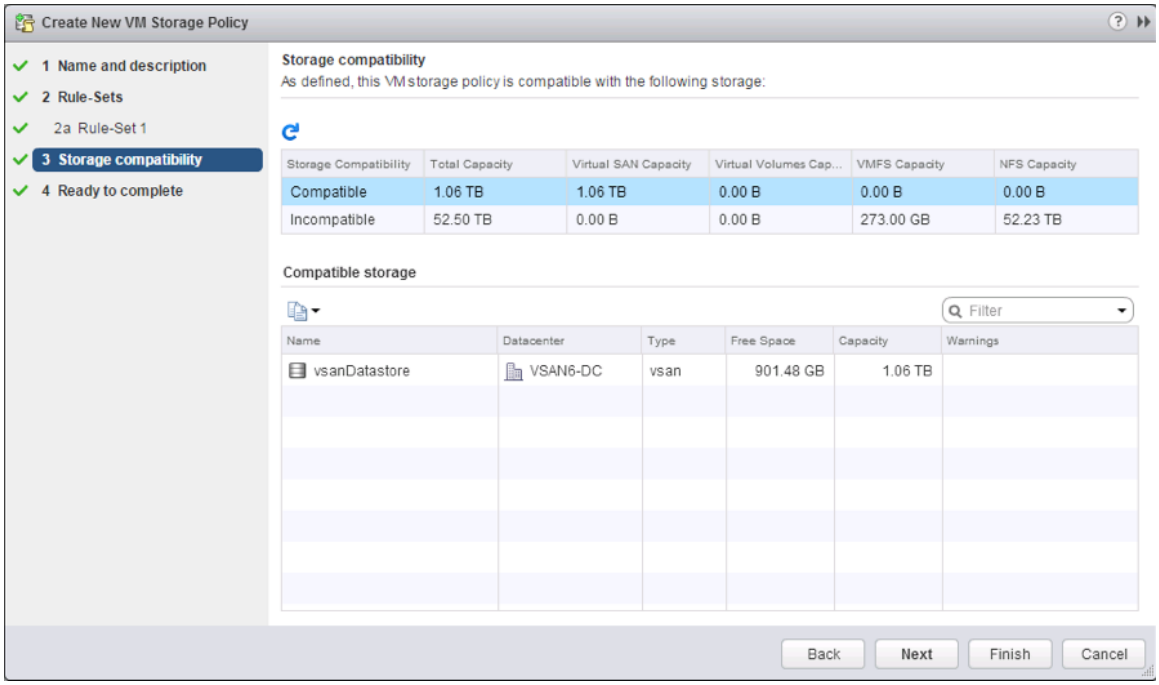


Figure A.10: vsanDatastore shows as compatible

The final step is to click on Finish and create the policy.

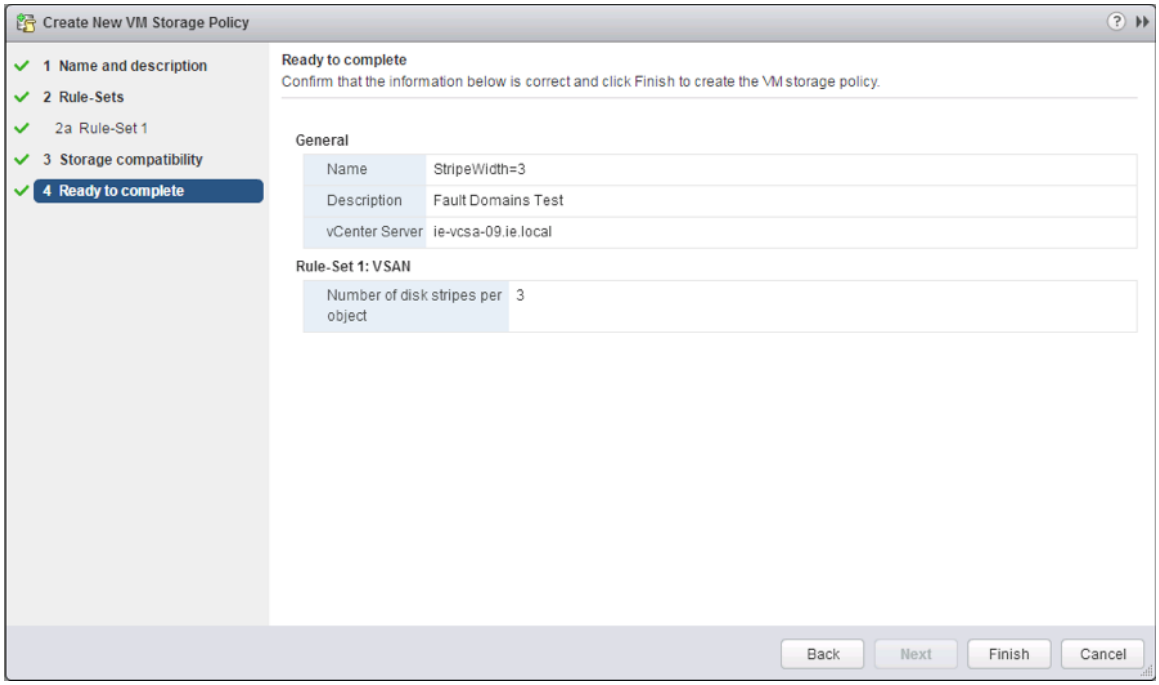


Figure A.11 Finish creating the policy

We can now go ahead and deploy a VM with this policy, and afterwards we shall examine the layout and see if it is taking Fault Domains into account.

A3. Create a VM and Check the Fault Domains

At this point, a new VM can be deployed. The only inputs required for this VM are to provide it with a name and to choose the newly created policy with a StripeWidth = 3.

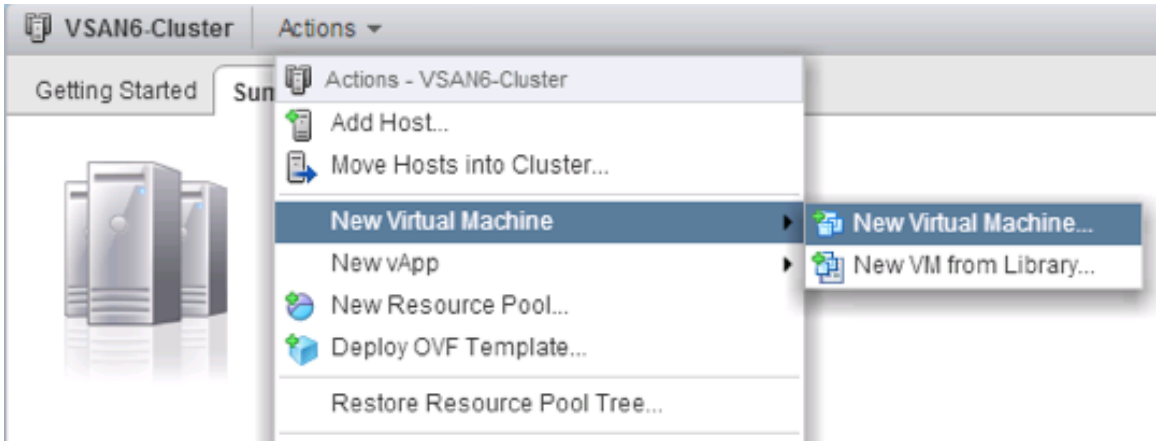


Figure A.12 Create a new VM

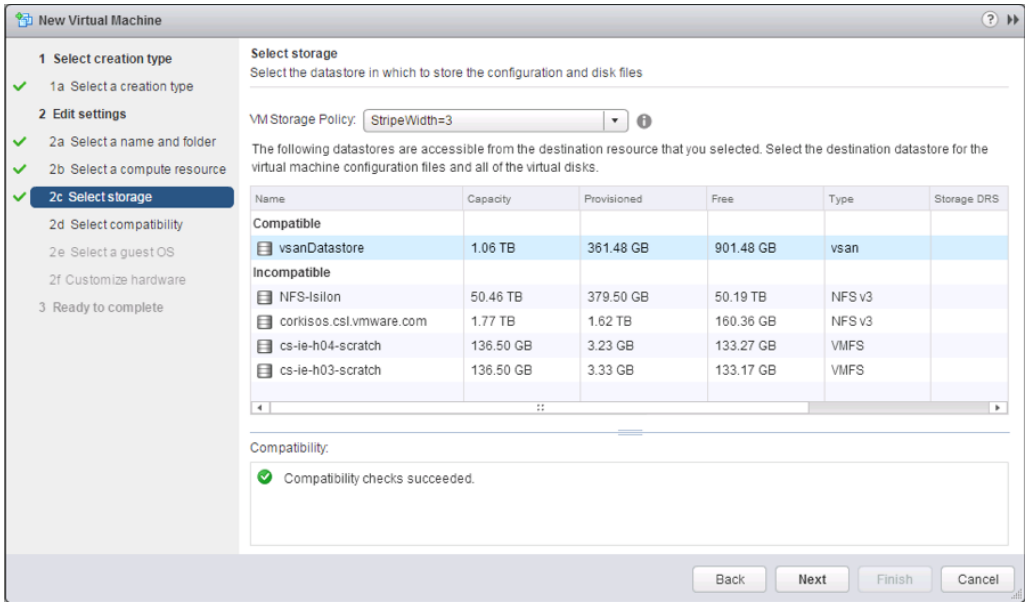


Figure A.13 Select the new VM Storage Policy

The rest of the VM creation options can be left at the default. Once the virtual machine has been deployed, check the Manage tab > Policies and verify that the VM is compliant with the policy. It should be compliant as shown below.

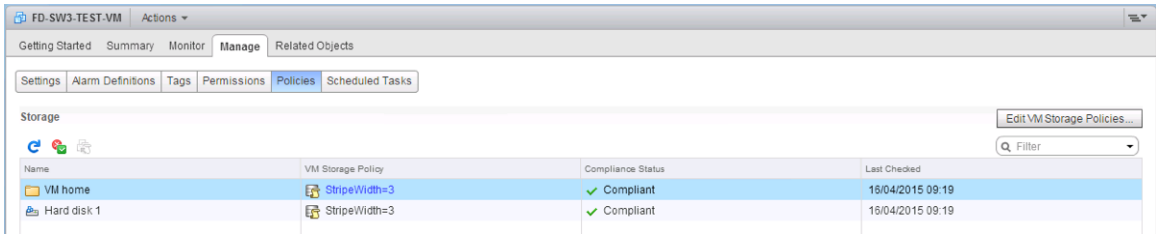


Figure A.14 VM Storage Policy is Compliant

Finally check the distribution of VM components under the Monitor tab > Policies.

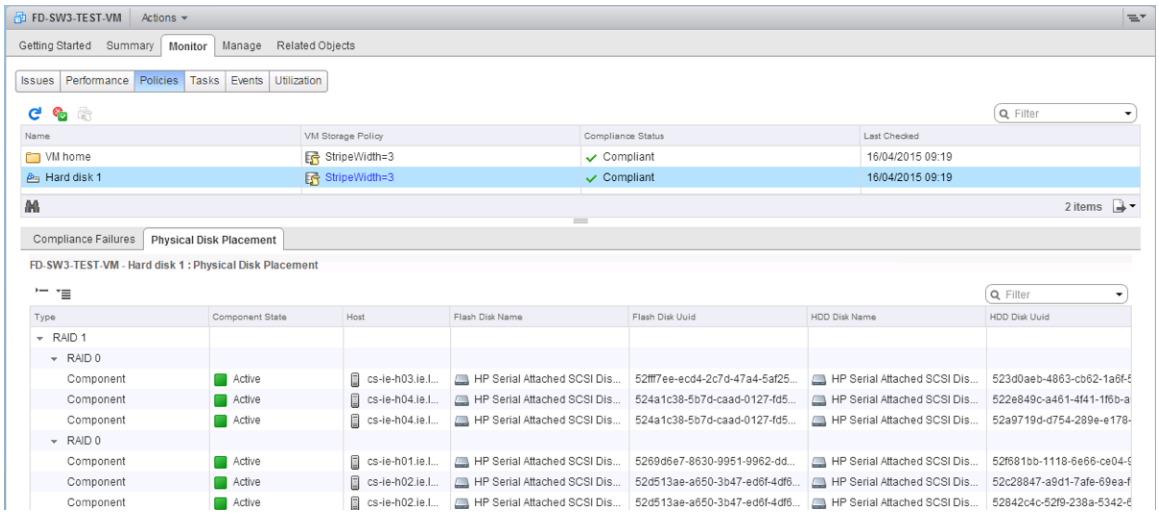


Figure A.15 Component distribution

The questions that need to be asked now are related to rack failures and fault domain failures. For example, if rack 1 were to fail, is there still a full copy of the data? The answer is yes. What about rack 2? Yes, there is still a full copy of the data. What about rack 3, which houses hosts 3 and 4? The answer is yes, once again there would be a fully copy of the data even if rack 3 failed.

One additional item to highlight here is the lack of witnesses. This is something new in Virtual SAN 6.0. Certain configurations do not need witnesses as a new voting mechanism has been introduced which gives components extra votes. Therefore in some configurations, such as this one, witnesses are not needed, reducing the overall component count.

Appendix B—Migrating from Standard vSwitch to Distributed

Before we begin, this procedure is rather complicated, and can easily go wrong. The only real reason why one would want to migrate from VSS (standard vSwitches) to a DVS (Distributed vSwitch) is to make use of the Network I/O Control feature that is only available with DVS. This will then allow you to place QoS (Quality of Service) on the various traffic types such as Virtual SAN traffic.

Warning: Ensure that you have console access to the ESXi hosts during this exercise. All going well, you will not need it. However, should something go wrong, you may well need to access the console of the ESXi hosts.

B.1 Create Distributed Switch

To begin with, create the distributed switch. This is a relatively straight forward exercise.

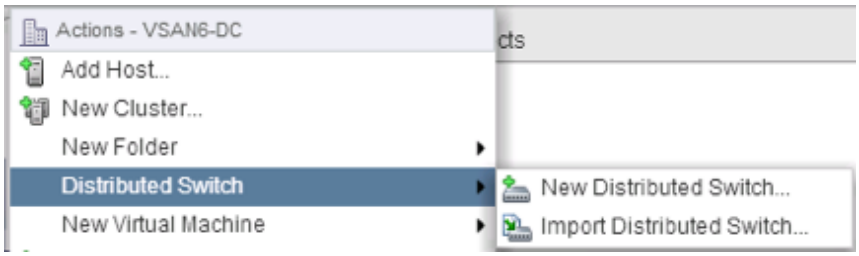


Figure B.1 Create a new distributed switch

Provide it with a name.

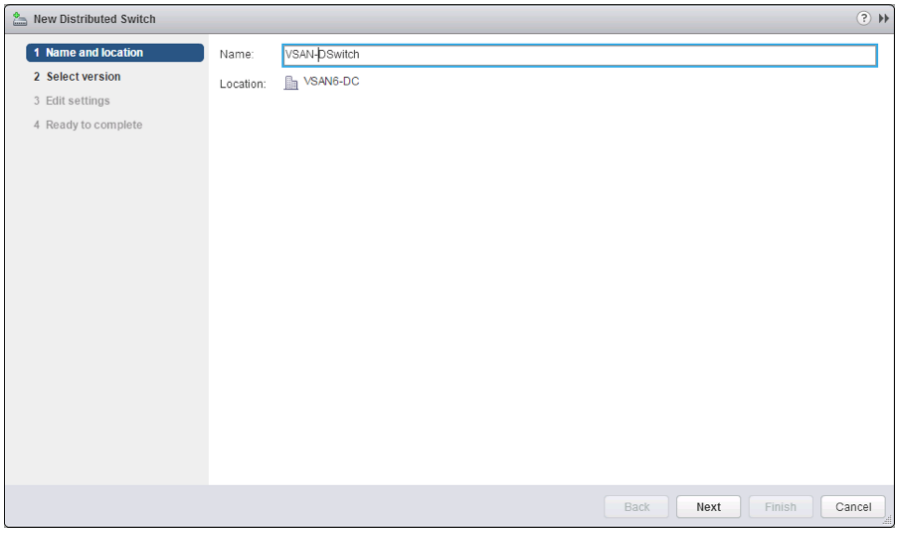


Figure B.2: Provide a name for the new distributed switch

Select the version of the DVS. In this example, we shall use the latest version, 6.0.0.

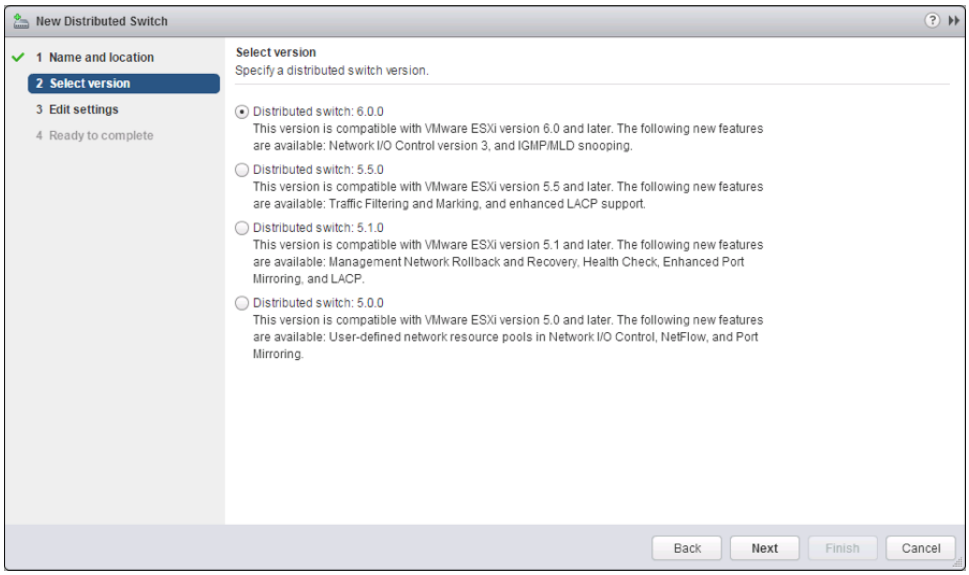


Figure B.3: Select the distributed switch version

At this point, we get to add the settings. First, you will need to determine how many uplinks you are currently using for networking. In our POC, we are using six; one for management, one for vMotion, one for virtual machines and three for Virtual SAN. Therefore, when we are prompted for the number of uplinks, we select "6". This may differ in your environment but you can always edit it later on.

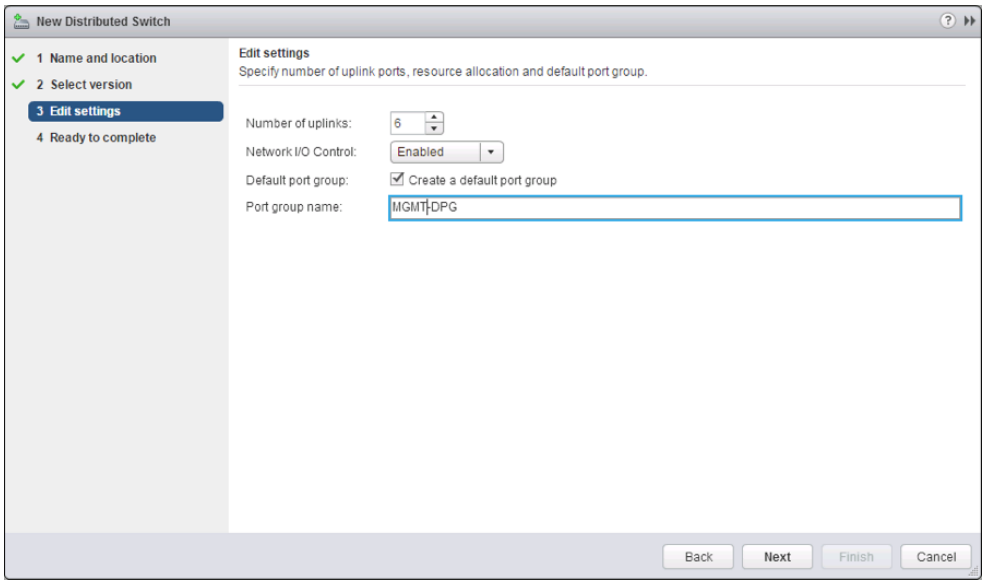


Figure B.4: Select the number of uplinks

Another point to note here is that a default portgroup can be created. You can certainly create a port group at this point, but there will be additional port groups that need to be created shortly. At this point, the distributed switch can be completed.

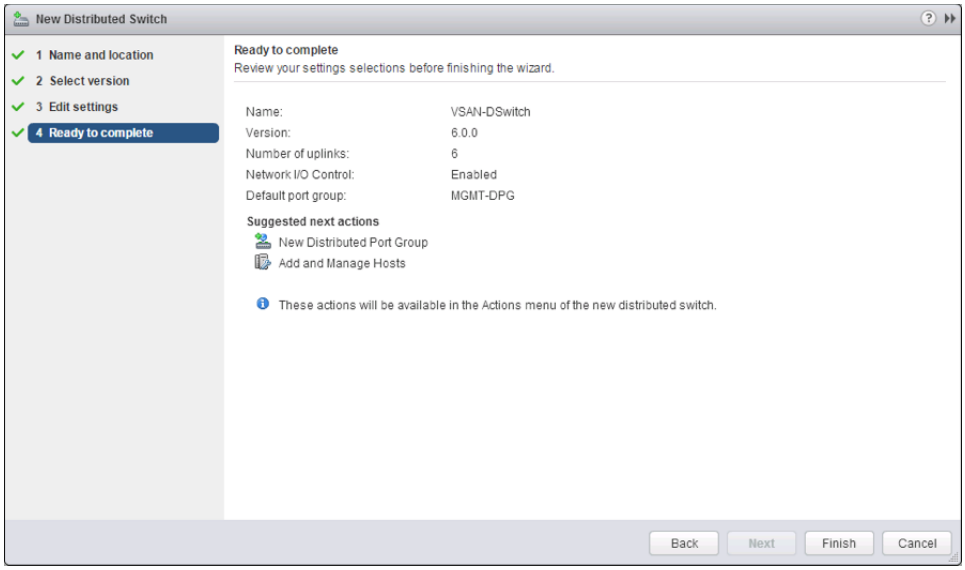


Figure B.5: Complete the creation of the DVS

As alluded to earlier, configure and create the additional port groups.

B.2 Create Port Groups

In the previous exercise, a single default port group was created for the management network. There was little in the way of configuration that could be done at that time. It is now important to edit this port group to make sure it has all the characteristics of the management port group on the VSS, such as VLAN and NIC teaming and failover settings. Select the distributed port group, and click on the Edit button shown below.

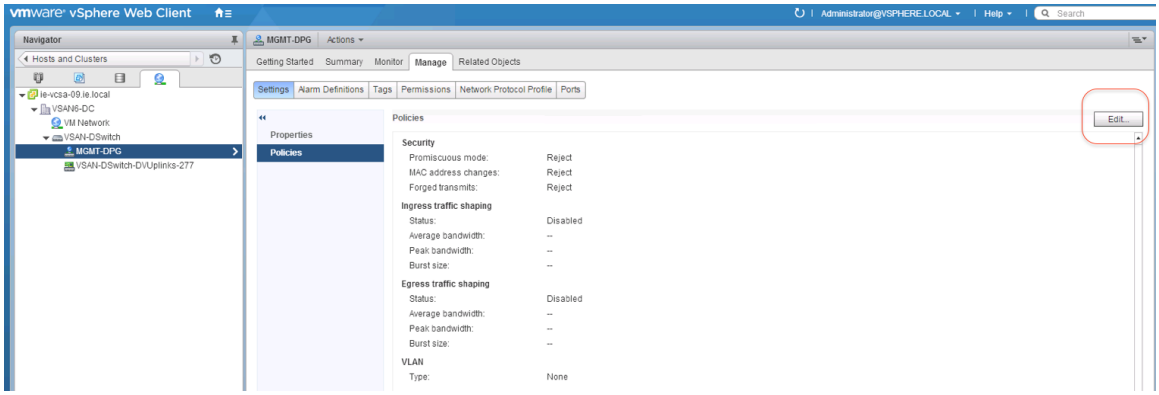


Figure B.6: Edit the distributed port group

For some port groups it may be necessary to change the VLAN. Since the management VLAN in this POC is on 51, we need to tag the distributed port group accordingly.

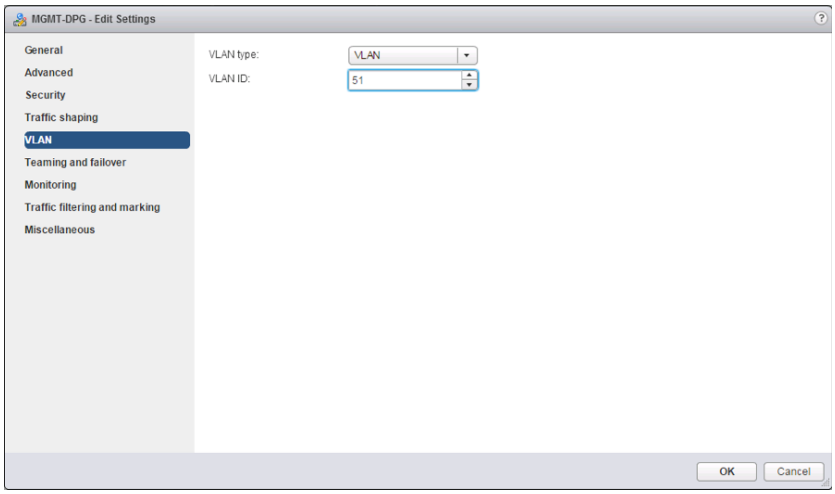


Figure B.7: Tag the distributed port group with a VLAN

That is the management distributed port group taken care of. You will also need to create distributed port groups for vMotion, virtual machine networking and of course Virtual SAN networking. In the "Getting Started" tab of the distributed switch, there is a basic task link called "Create a new port group".

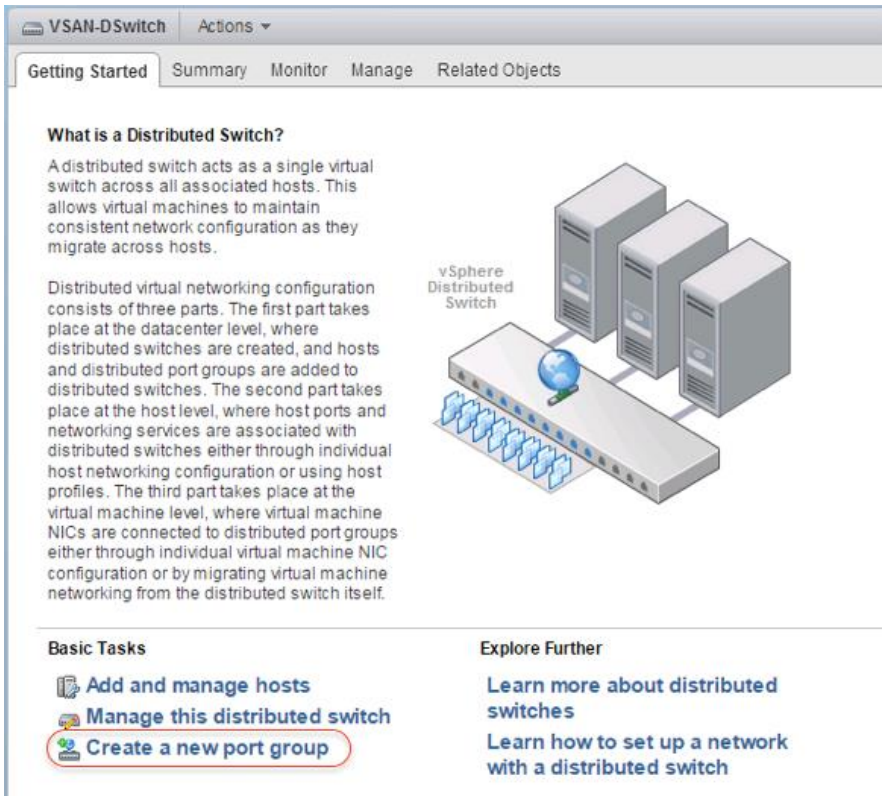


Figure B.8: Create a new distributed port group

In this exercise, we shall create a port group for the vMotion network.

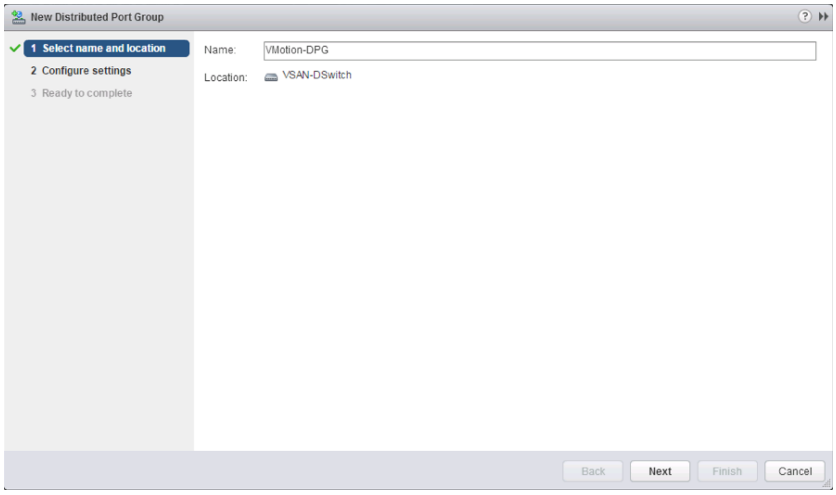


Figure B.9: Provide a name for the new distributed port group

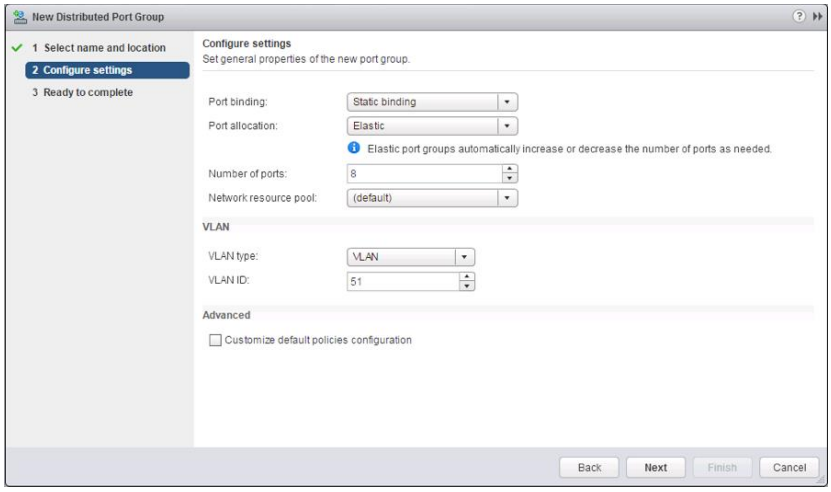


Figure B.10: Configure distributed port group settings, such as VLAN

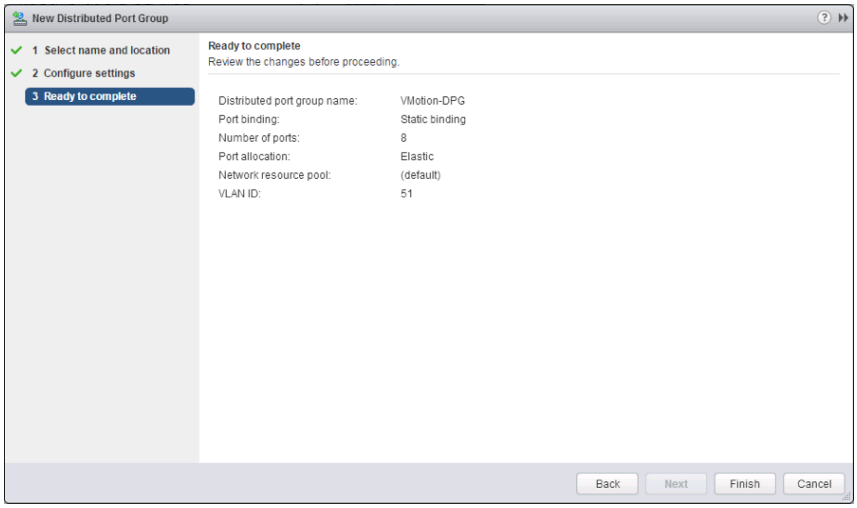


Figure B.11: Finish creating the new distributed port group

Once all the distributed port groups are created on the distributed switch, the uplinks, VMkernel networking and virtual machine networking can be migrated to the distributed switch and associated distributed port groups.

Warning: While the migration wizard allows many uplinks and many networks to be migrated concurrently, we recommend migrating the uplinks and networks step-by-step to proceed smoothly and with caution. For that reason, this is the approach we use here.

B.3 Migrate Management Network

To begin, let's migrate just the management network (vmk0) and its associated uplink, which in this case is vmnic0 from VSS to DVS. To begin, select "Add and manage hosts" from the basic tasks in the Getting started tab of the DVS.

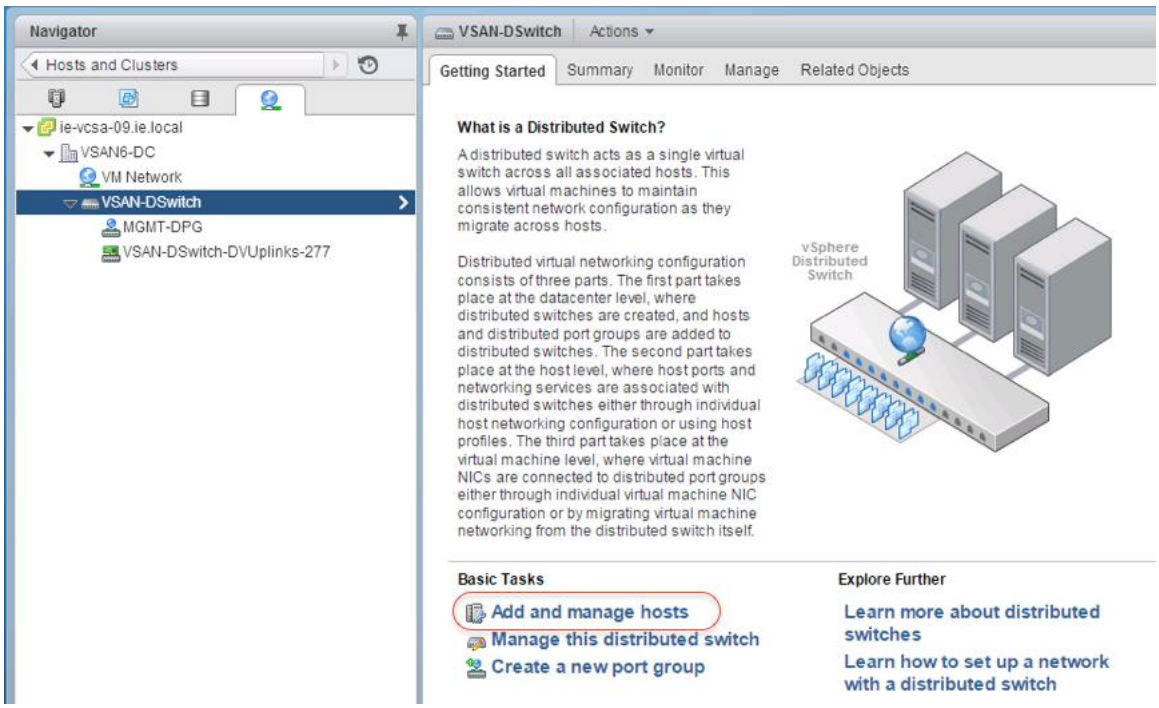


Figure B.12: Add and manage hosts

The first step is to add hosts to the DVS.

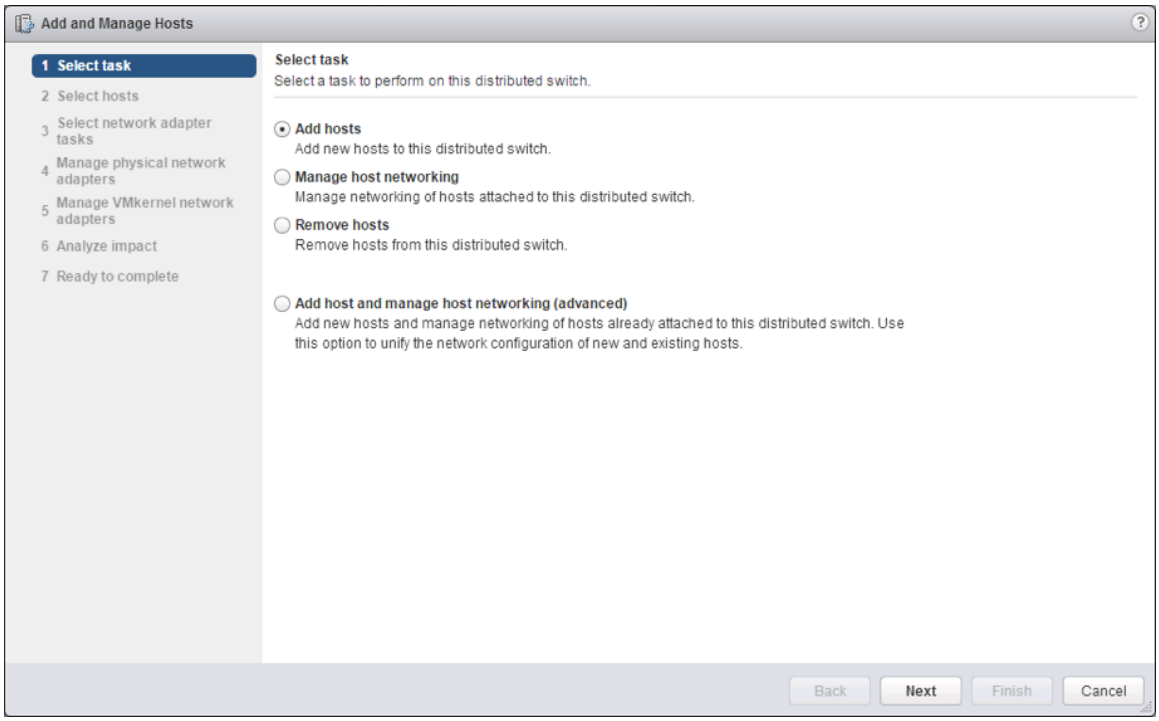


Figure B.13: Add hosts

Click on the green + and add all four hosts from the cluster.

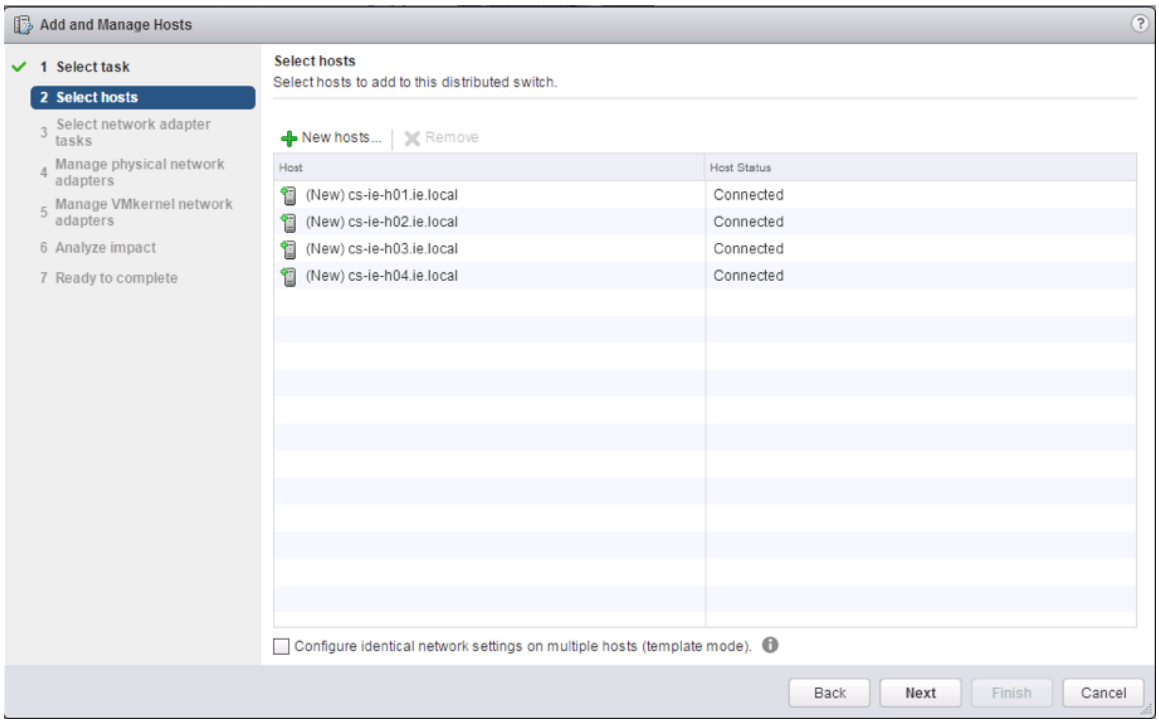


Figure B.14: Select all hosts in the cluster

The next step is to manage both the physical adapters and VMkernel adapters. To repeat, what we wish to do here is migrate both vmnic0 and vmk0 to the DVS.

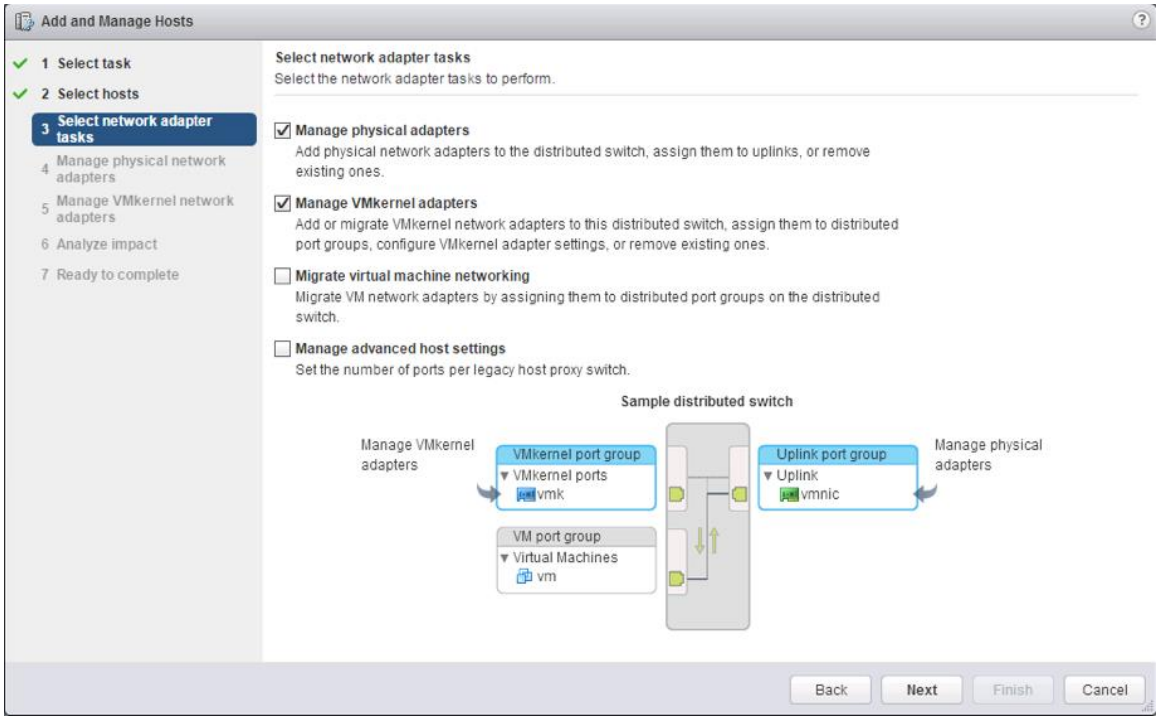


Figure B.15: Select physical adapters and VMkernel adapters

Next, select an appropriate uplink on the DVS for physical adapter vmnic0. In this example we chose Uplink1.

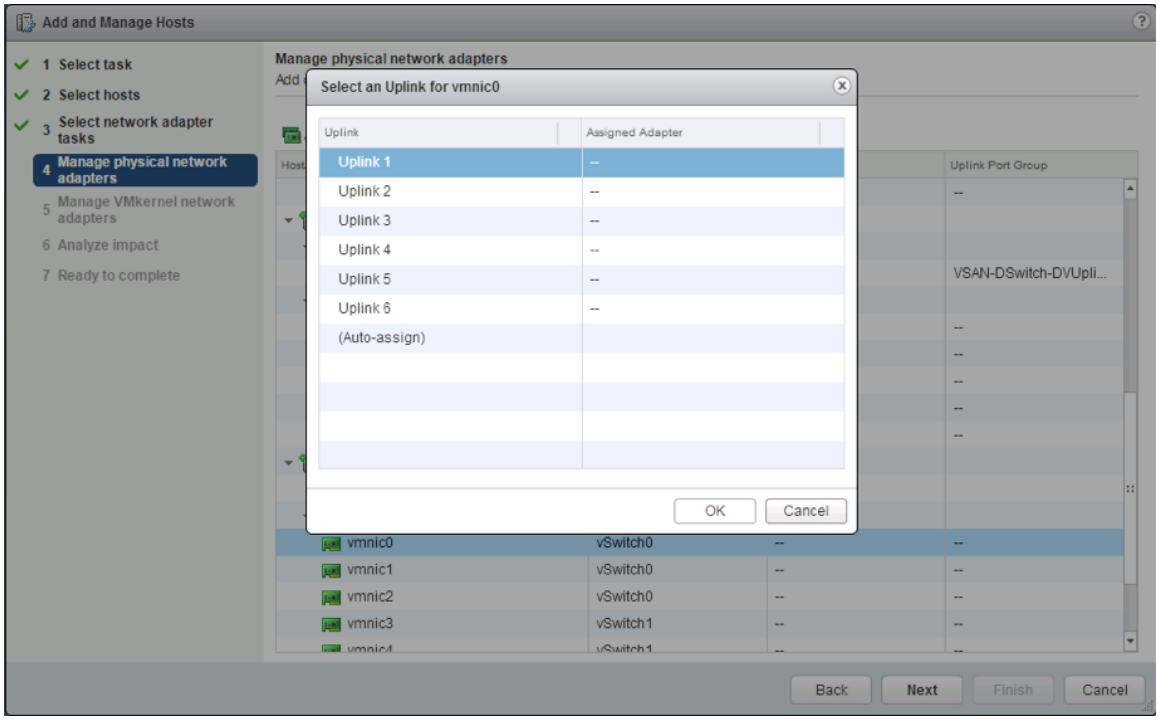


Figure B.16: Assign uplink (uplink1) to physical adapter vmnic0

With the physical adapter selected and an uplink chosen, the next step is to migrate the management network on vmk0 from the VSS to the VDS. We are going to leave vmk1 and vmk2 for the moment and just migrate vmk0.

Select vmk0, and then click on the "Assign port group" as shown below. The port group assigned should be the newly created distributed port group created for the management network earlier. Remember to do this for each host.

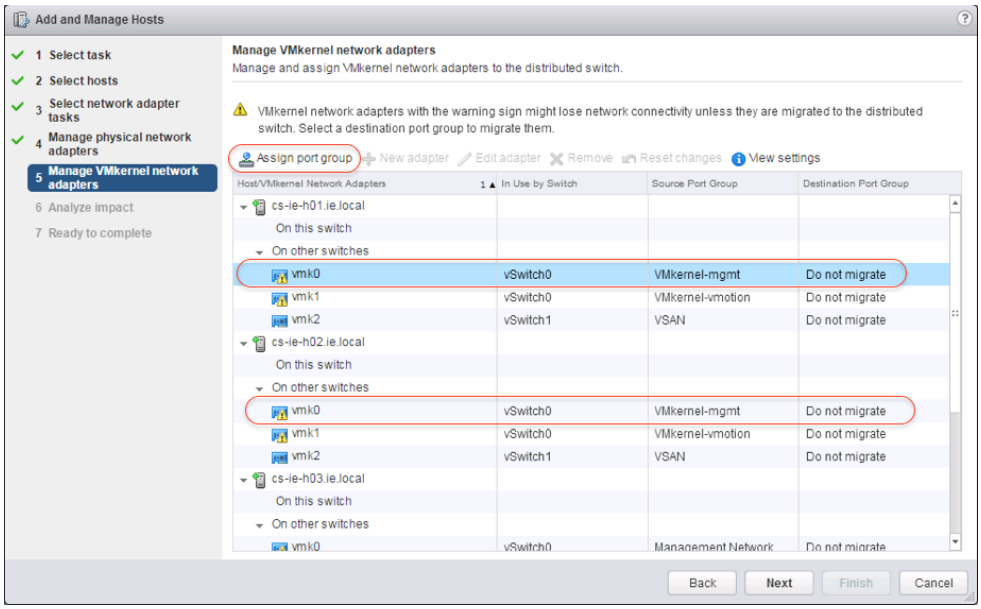


Figure B.17: Assign port group for vmk0

Click through the analyze impact screen since it only checks iSCSI and is not relevant to the Virtual SAN POC.

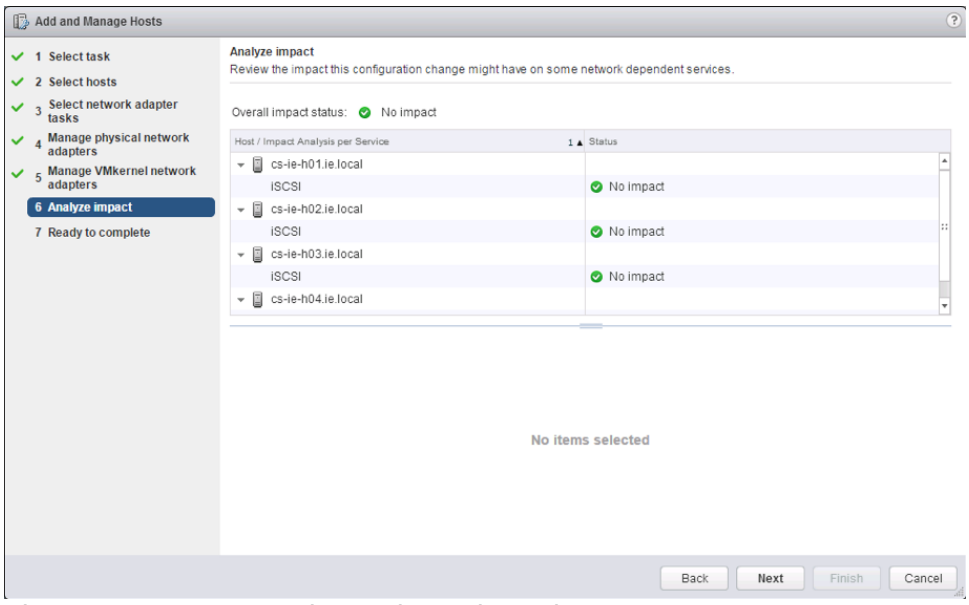


Figure B.18: Impact on iSCSI (not relevant)

At the finish screen, you can examine the changes. We are adding 4 hosts, 4 uplink s (vmnic0 from each host) and 4 VMkernel adapters (vmk0 from each host).

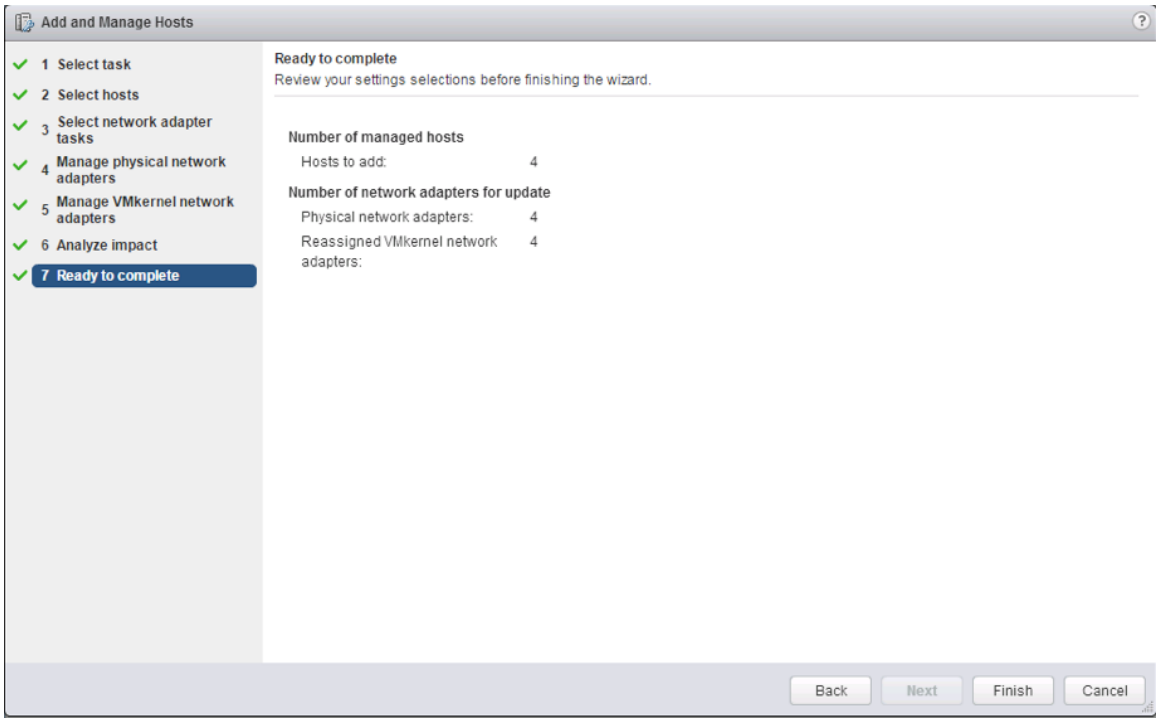


Figure B.19: Ready to complete

When the networking configuration of each host is now examined, you should observe the new DVS, with one uplink (vmnic0) and the vmk0 management port on each host.

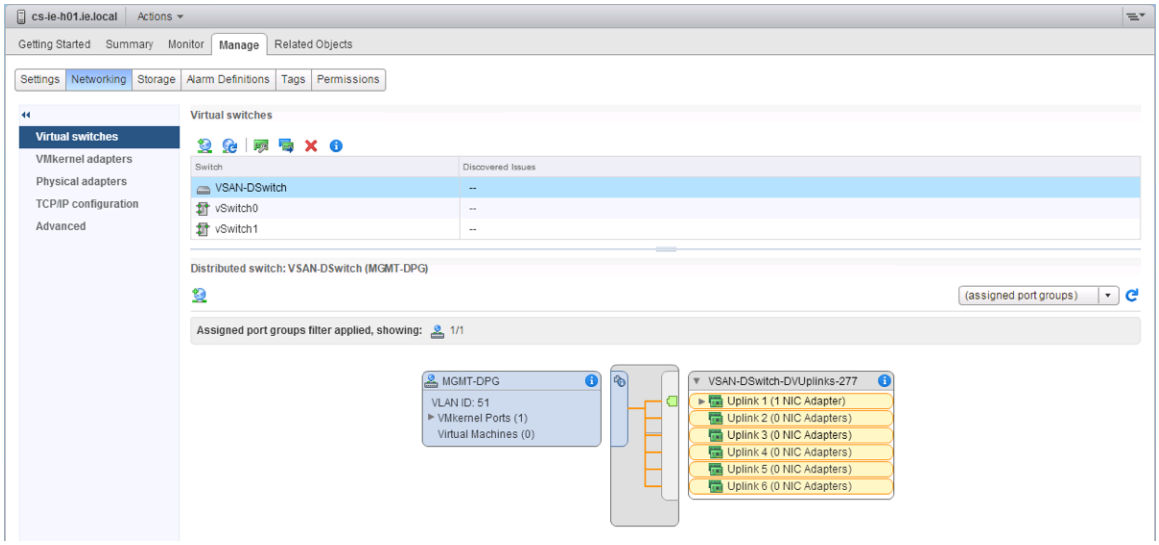


Figure B.20: Management network migration to DVS complete

You will now need to repeat this for the other networks.

B.4 Migrate vMotion

Migrating the vMotion network takes the exact same steps as the management network. Before you begin, ensure that the distributed port group for the vMotion network has all the same attributes as the port group on the standard (VSS) switch. Then it is just a matter of migrating the uplink used for vMotion (in this case vmnic1) along with the VMkernel adapter (vmk1). As mentioned already, this takes the same steps as the management network.

When the migration completes, the individual host network configuration should look similar to the following diagram.

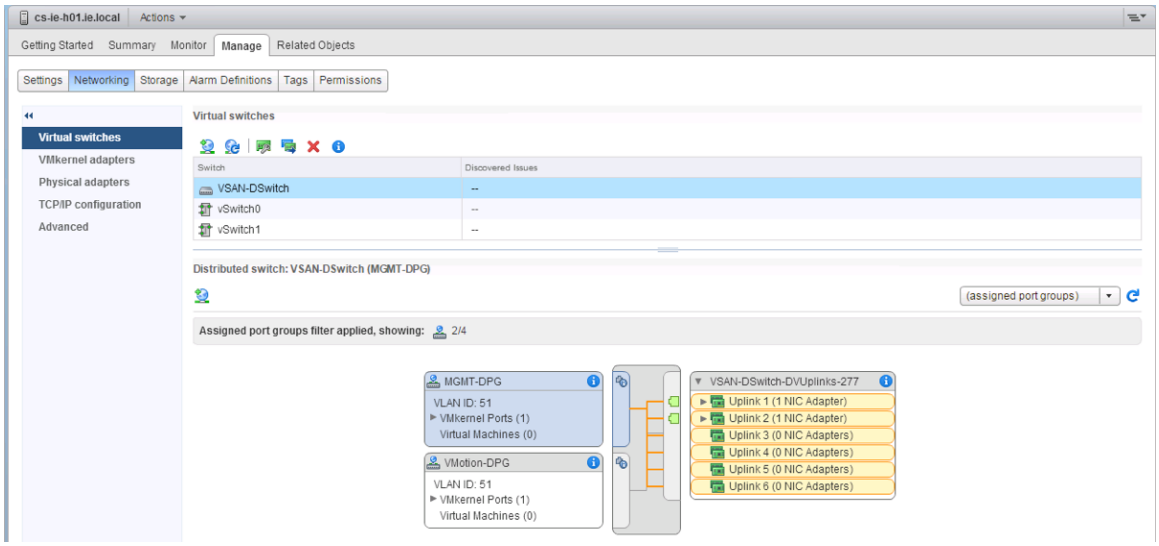


Figure B.21: vMotion network migration to DVS complete

B.5 Migrate Virtual SAN Network

If you are using a single uplink for the Virtual SAN network, then the process becomes the same as before.

However, if you are using more than one uplink, then there are additional steps to be taken. If the Virtual SAN network is using a feature such as Link Aggregation (LACP), or it is on a different VLAN to the other VMkernel networks, then you will need to place some of the uplinks into an unused state for certain VMkernel adapters.

For example, in this scenario, VMkernel adapter vmk2 is used for Virtual SAN. However uplinks vmnic3, 4 and 5 are used for Virtual SAN and they are in a LACP configuration. Therefore for vmk2, all other vmnics (0, 1 and 2) must be placed in an unused state. Similarly, for the management adapter (vmk0) and vMotion adapter (vmk0), the Virtual SAN uplinks/vmnics should be placed in an unused state.

Modifying the settings of the distributed port group and changing the path policy/failover appropriately do this.

In the manage physical network adapter, the steps are similar as before except that now you are doing this for multiple adapters.

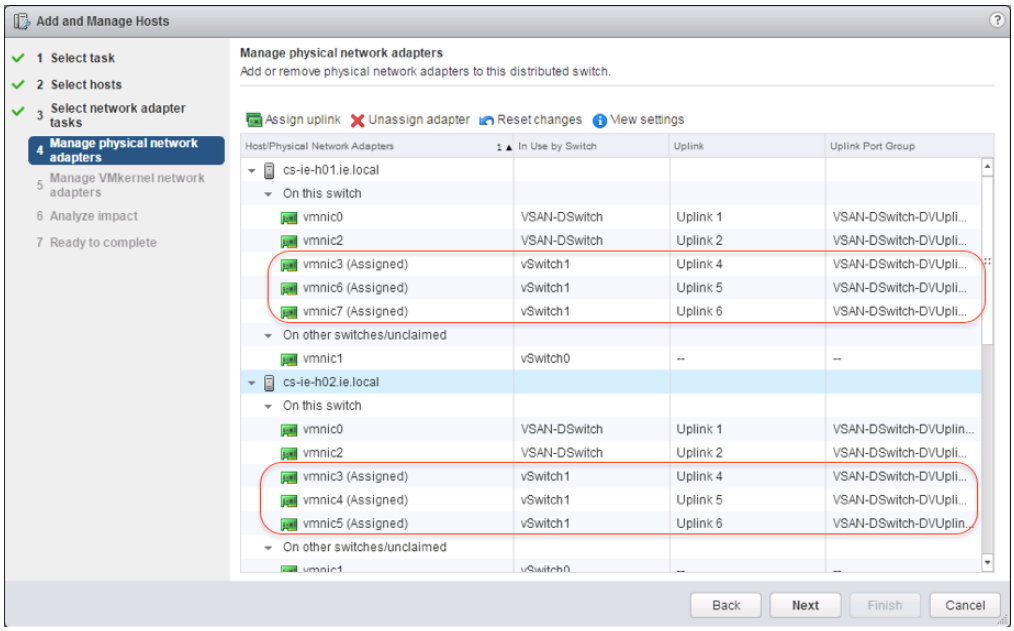


Figure B.22: Multiple uplinks used by the Virtual SAN network

As before, vmk2 (the Virtual SAN VMkernel adapter) should be assigned to the distributed port group for Virtual SAN.

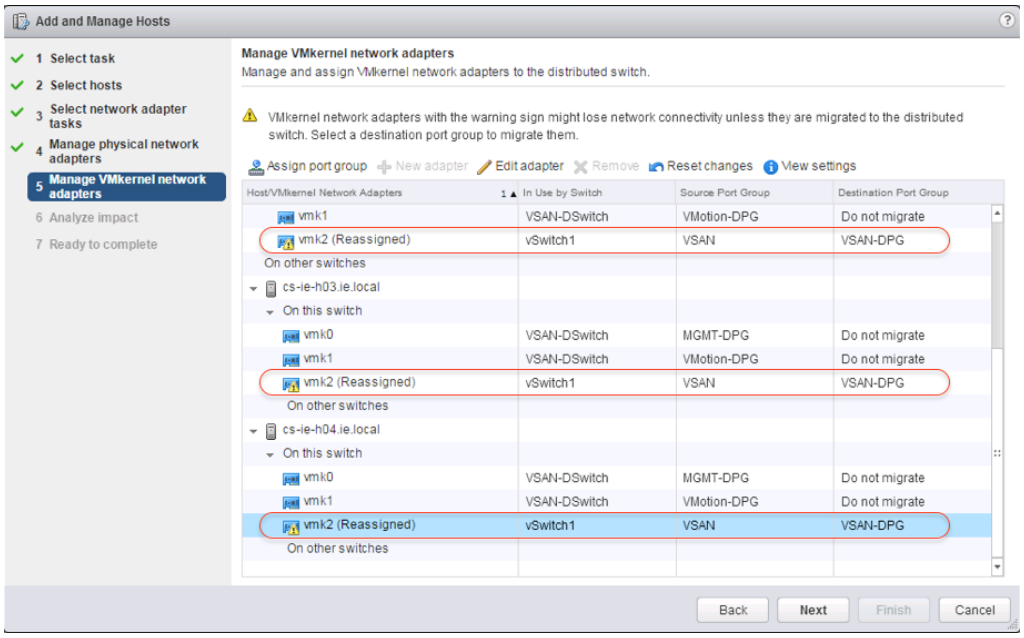


Figure B.23: Assign distributed port group for Virtual SAN networking

Note: If you are only now migrating the uplinks for the Virtual SAN network, you may not be able to change the distributed port group settings until after the migration. During this time, Virtual SAN may have communication issues. After the migration, move to the distributed port group settings and make any policy changes and mark any uplinks that should be unused. Virtual SAN networking should then return to normal when this task is completed. Use the Health Check plugin to verify that everything is functional once the migration is completed.

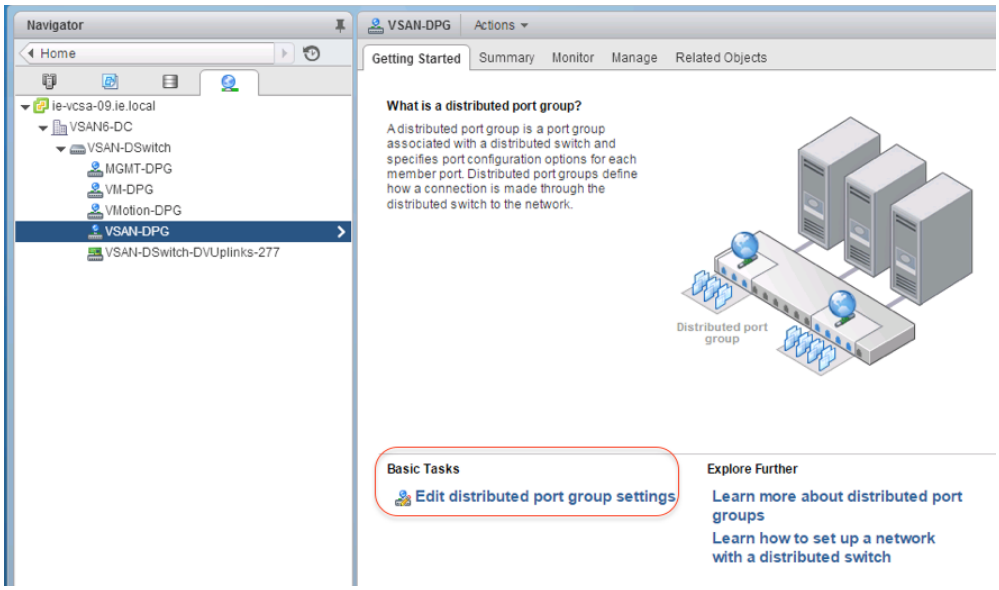


Figure B.24: Change distributed port group settings

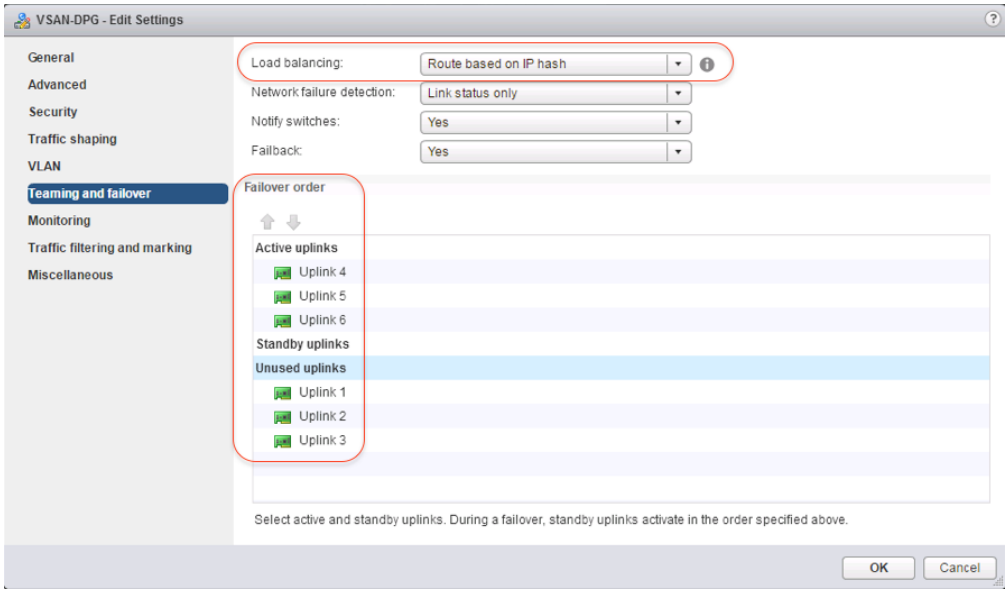


Figure B.25: Showing load balancing and unused uplinks

That completed the VMkernel adapter migrations. The final step is to move the VM networking.

B.6 Migrate VM Network

This is the final step of migrating the network from a standard vSwitch (VSS) to a distributed switch (DVS). Once again, we use the "Add and manage hosts", the same link used for migrating the VMkernel adapters. The task is to manage host networking.

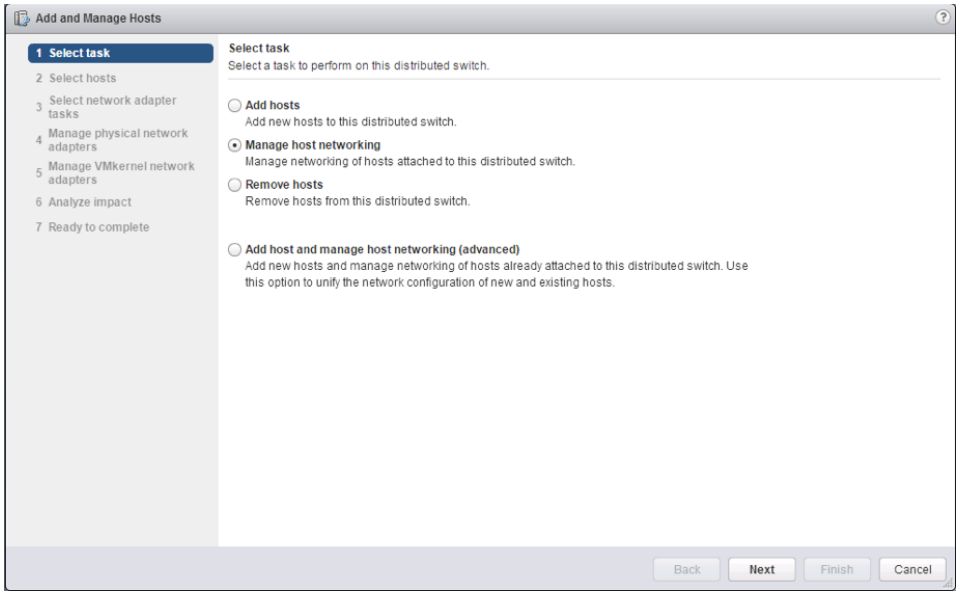


Figure B.26: Manage host networking

Select all the hosts in the cluster, as all hosts will have their virtual machine networking migrated to the distributed switch.

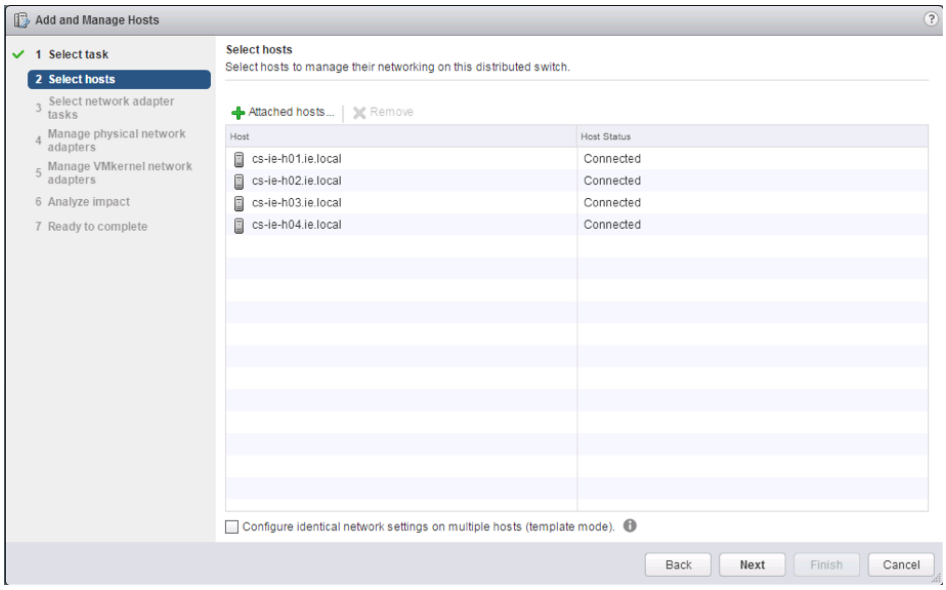


Figure B.27: Select all hosts

On this occasion, we do not need to move any uplinks. However, if the VM networking on your hosts used a different uplink, then this of course would also need to be migrated from the VSS. In this example, the uplink has already been migrated.

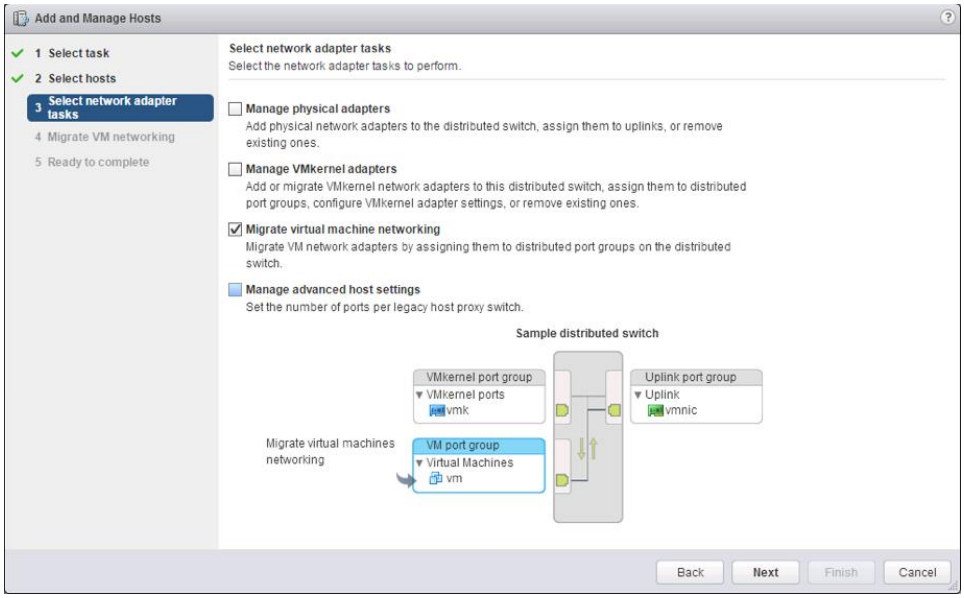


Figure B.28: Migrate virtual machine networking

Select the VMs that you wish to have migrated from a virtual machine network on the VSS to the new virtual machine distributed port group on the DVS. Click on the "Assign port group" option like we have done many times before, and select the distributed port group, name VM-DPG here.

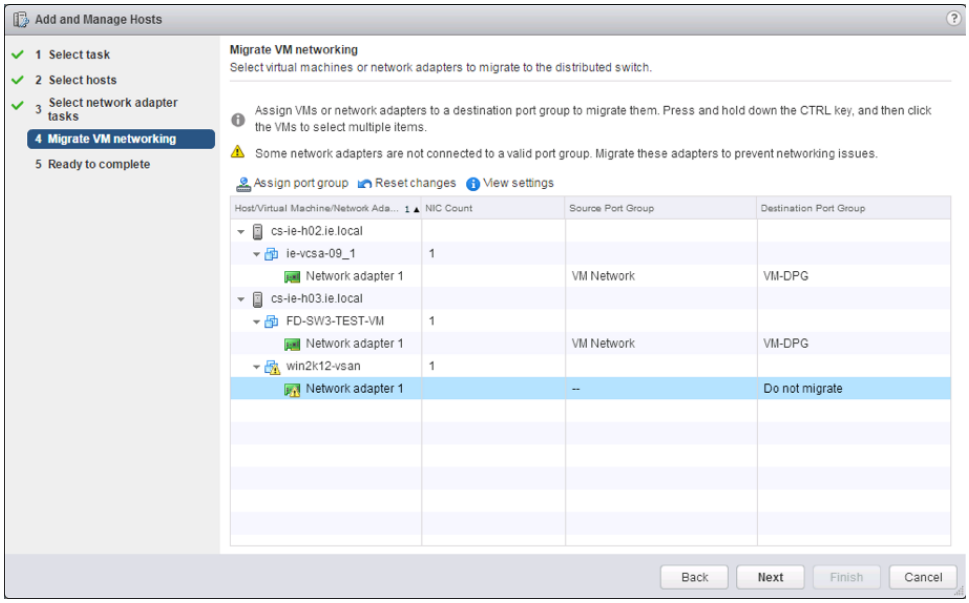


Figure B.29: Assign port groups for the VMs

Reviewing the final screen. In this case we are only moving to VMs. Note that any templates using the original VSS virtual machine network will need to be converted

to virtual machines, edited and the new distributed port group for virtual machines will need to be selected as the network. This step cannot be achieved through the migration wizard.

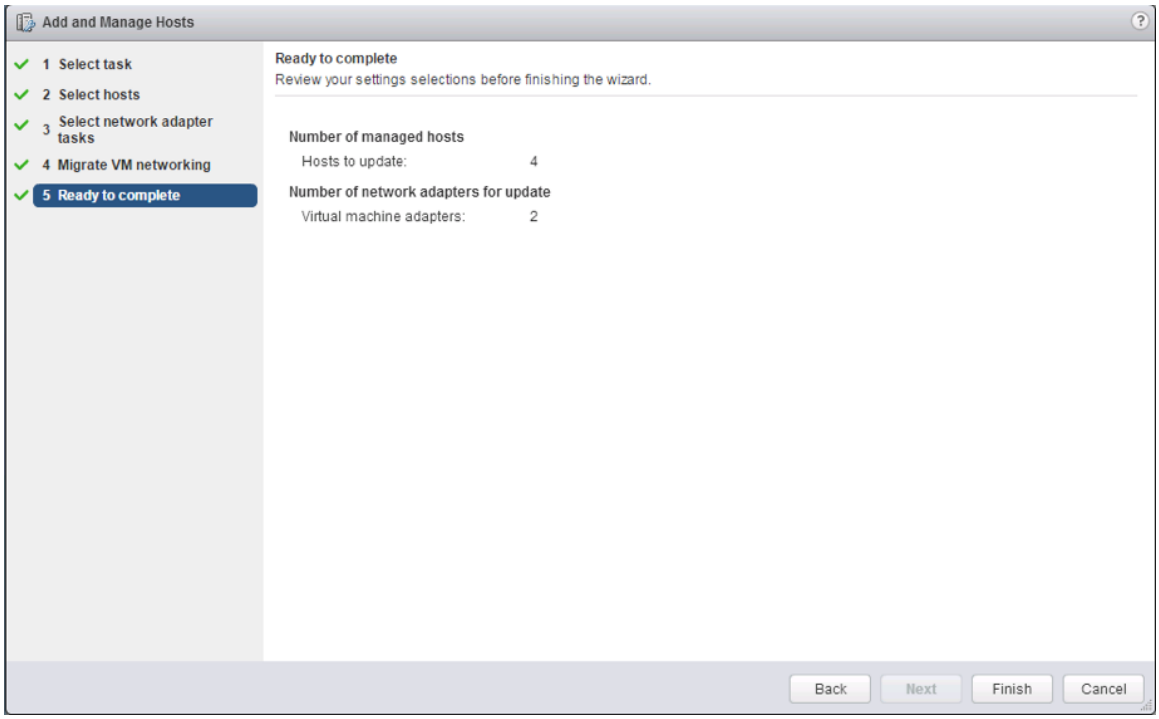


Figure B.30: Finish

The VSS should no longer have any uplinks of port groups and can be safely removed.

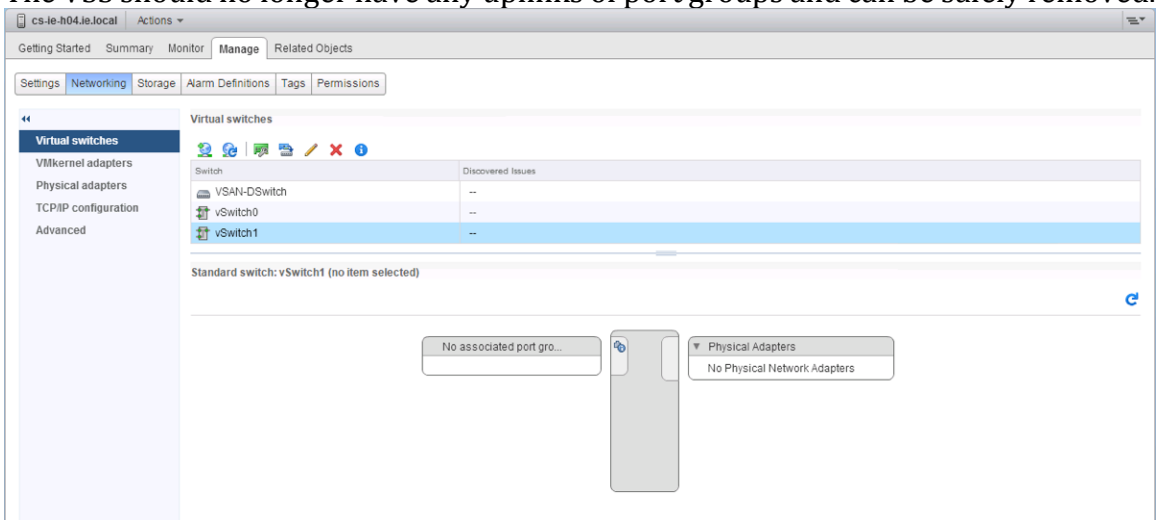


Figure B.31: VSS no longer in use

This completes the migration from a standard vSwitch (VSS) to a distributed vSwitch (DVS).