



Aidan Finn
27 April 2011

<http://www.aidanfinn.com/>

Building a Hyper-V Cluster using the Microsoft iSCSI Software Target

How to build a small production or lab environment using the free iSCSI storage solution

This could be where your company promotes its products

Contact me via my blog for more information.

Table of Contents

Introduction.....	5
Architecture.....	6
Preparing the Servers	8
Hyper-V Host Operating Systems.....	8
System Center.....	8
Host1 and Host2	9
Storage Server.....	9
Networking Protocols	9
Installing the iSCSI Target.....	12
System Center Storage on the Target	14
Adding the System Center Target	14
Adding System Center Storage	17
The System Center iSCSI Initiator.....	20
Add the System Center Target	20
Formatting the System Center Volume.....	23
Hyper-V Cluster Storage on the Target	25
Adding the Hyper-V Cluster Target	25
Adding the Hyper-V Cluster Storage	26
Building the Hyper-V Cluster.....	29
Configuring the Cluster	32
Storage Renaming.....	32
Cluster Network Selection	33
Hyper-V Virtual Machine Networking.....	34
Enable and Configure Cluster Shared Volume (CSV).....	35
Specify CSV Network.....	37
Build Some Highly Available Virtual Machines.....	38
Specify the Cluster's Live Migration Network	39
Additional Storage Operations.....	41
Adding a CSV.....	41
Extending a CSV	42
Summary.....	45
Credits.....	46

About The Author 47

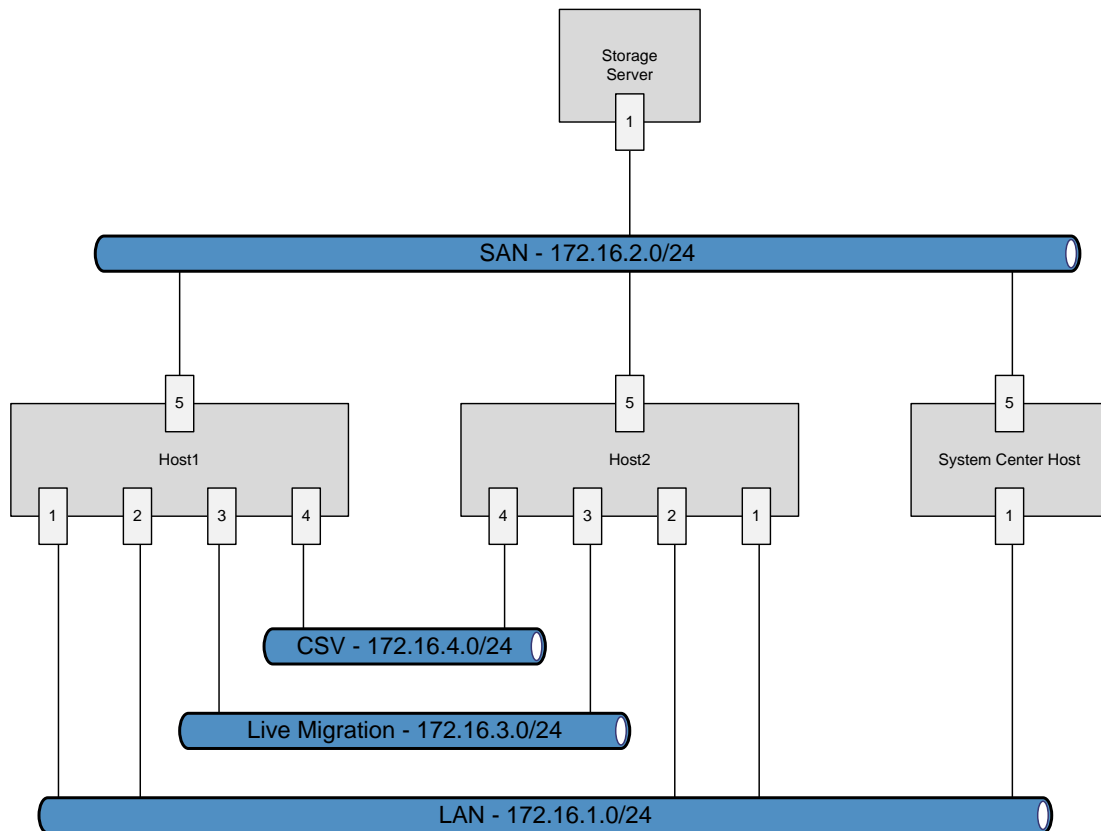
Introduction

The Microsoft iSCSI Software Target is a free iSCSI storage solution. It is included as a part of Windows Storage Server 2008 R2, and it is a free download for Windows Server 2008 R2. This allows a Windows Server to become a shared storage solution for many computers. It also provides an economic way to provide an iSCSI "SAN" for a Failover Cluster, such as Hyper-V.

This document will detail how to build a 2 node Hyper-V cluster, using the Microsoft iSCSI Software Target for shared storage, which is managed by System Center running on virtual machines, hosted on another Hyper-V server and stored on the same shared storage.

Architecture

The final solution that is being built is illustrated below:



Note that it is strongly recommended that the following switches (at least 1 Gbps) are used:

- SAN: a dedicated switch for the SAN to isolate it from general networking. This is for security and performance reasons.
- CSV: You can use a cross over cable in a 2 host cluster. A switch is required for more than 3 hosts.
- Live Migration: You can use a cross over cable in a 2 host cluster. A switch is required for more than 3 hosts.
- LAN: A switch will be required.

Note that you *could* use the same switch (ideally with dedicated VLANs) for CSV, Live Migration, and the LAN.

There are four servers in this lab. The details of the servers are detailed in the following table. You can substitute server names as appropriate.

Role	Computer Name	Description
Host1	InfraCluster1	Clustered Hyper-V host
Host2	InfraCluster2	Clustered Hyper-V host
System Center	InfraHost1	Non-clustered Hyper-V host for System Center VMs

Storage Server	SAN	iSCSI target used to provide storage to Hyper-V hosts
----------------	-----	---

The following table details the NIC functions and IP configurations for each server in the architecture:

Role	NIC1	NIC2	NIC3	NIC4	NIC5
System Center	Parent - 172.16.1.1	VM - No IP Binding	N/A	N/A	iSCSI1 - 172.16.2.1
Host1	Parent - 172.16.1.10	VM - No IP Binding	LM - 172.16.3.1	CSV - 172.16.4.1	iSCSI1 - 172.16.2.2
Host2	Parent - 172.16.1.11	VM - No IP Binding	LM - 172.16.3.2	CSV - 172.16.4.2	iSCSI1 - 172.16.2.3
Storage Server	SAN - 172.16.2.250	N/A	N/A	N/A	N/A

You'll notice that there is only one iSCSI NIC per host in this architecture. At the time of writing this document Microsoft stated on TechNet ([http://technet.microsoft.com/library/gg214141\(ws.10\).aspx](http://technet.microsoft.com/library/gg214141(ws.10).aspx)) that "When used with Microsoft iSCSI Software Target, MPIO is not supported for iSCSI initiators when the initiators are members of a failover cluster". So that meant no MPIO for us, and therefore no fault tolerant iSCSI NICs. Well, that's what they said on TechNet. It turns out that this wasn't true. It was later discovered that the page in question was not updated correctly and MPIO is supported when initiators are members of a failover cluster (credit #1). This document has not been updated since.

Tip #1: Use an A3 sheet of paper or a white board to plan out your NICs and IP addresses. Diagram everything out and document each NIC and IP configuration for that NIC. Use that as a reference when configuring your servers.

Tip #2: Insert each NIC cable one at a time. That way you can identify and configure each Local Area Connection. I usually rename them according to their role. For example, *Local Area Connection X* would be renamed to *Parent*. This allows me or anyone else to easily troubleshoot networking if there is an issue later on.

Tip #3: Log into each cluster host in turn, and ping each of the network connections of each of the other hosts and the iSCSI target IP address. This will help you track down any networking issues that could cause storage or networking issues.

The more time you spend on network configuration and testing, the less time you'll spend trying to troubleshoot when your storage or failover cluster don't seem to work.

Preparing the Servers

Hyper-V Host Operating Systems

It should go without saying that you should use some variation of Windows Server 2008 R2, ideally with Service Pack 1. Windows Server 2008 R2 adds a whole bunch, with the headlines being Cluster Shared Volume and Live Migration. Throw in Service Pack 1 and the big news is Dynamic Memory which will optimise memory assignment to virtual machines.

What editions should you use to create a Hyper-V cluster? You have three choices:

- Windows Server 2008 R2 Enterprise Edition: Includes Failover Clustering and Live Migration. This license gives you 4 free installs of Windows Server Enterprise or lower for your virtual machines on the licensed host.
- Windows Server 2008 R2 Datacenter Edition: Includes Failover Clustering and Live Migration. This license gives you unlimited free installs of Windows Server Enterprise or lower for your virtual machines on the licensed host. The host is licensed per CPU.
- Hyper-V Server 2008 R2: This is a free version of Windows Server that is intended to only be a Hyper-V host and nothing else. It has no GUI and can be managed locally using command prompt (and a basic text based wizard), and remotely via System Center Virtual Machine Manager and the remote server administration tools. There are no free guest OSs for the guests running on these hosts.

Your iSCSI target machine can run any purchased version of Windows Server 2008 R2. Your System Center Hyper-V host can run any Windows machine with Hyper-V.

Talk to a Microsoft licensing specialist if you need more details. This stuff is a nest of twists and turns.

Now the big question: do you want a Core or a Full installation? Microsoft employees seem to have to encourage you towards core because of the “smaller installation footprint and attack surface, and it requires less CPU and memory”. Hmm, A couple of MB of RAM and a few GB of disk space are saved, while adding unbelievable complexity to have command prompt only administration on the local machine. I tried Core in the past, really wanting it to work. Added administrative effort, as well as the inability to run many tools, ended that plan, so I always advise a full installation of Windows Server.

System Center

Note that in this demo lab, this machine is the domain controller. This machine will also be a Hyper-V host. This is not recommended in a production environment. In fact: don't do this in a production environment! Use at least one dedicated *physical* domain controller.

The server will be prepared as follows:

1. Install Windows Server 2008 R2 with Service Pack 1.
2. Configure networking as described in the architecture.
3. Enable the Active Directory Services role and use *DCPROMO* to create a new forest/domain.
4. Patch the server.

5. Enable the Failover Clustering feature.
6. Enable the Hyper-V role.
7. Optionally add the features for managing Hyper-V and Failover Clustering (for centralised management from this host).

Host1 and Host2

The Hyper-V cluster hosts should be prepared as follows:

1. Install Windows Server 2008 R2 with Service Pack 1.
2. Configure networking as described in the architecture.
3. Patch the servers.
4. Enable the Failover Clustering feature.
5. Enable the Hyper-V role.
6. Join the servers to the domain.

Note: if you do not want to use Service Pack 1 for Windows Server 2008 R2 (I recommend that you do) then make sure you download and install of the patches for Hyper-V from [http://technet.microsoft.com/library/ff394763\(WS.10\).aspx](http://technet.microsoft.com/library/ff394763(WS.10).aspx). Note that there is a patch that resolves a memory leak issue for iSCSI clients.

Storage Server

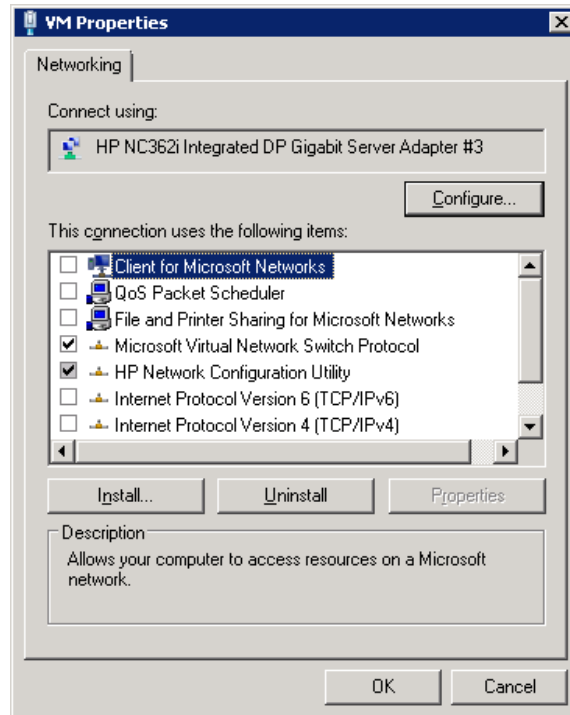
You will set up the storage server as follows. You can alternatively use Windows Storage Server 2008 R2. In that case you will not need to download and install the iSCSI target, as will be described at a later point.

1. Install Windows Server 2008 R2 with Service Pack 1.
2. Configure networking as described in the architecture.
3. Patch the server.
4. Optionally (for ease of administration) join the server to the domain.

Networking Protocols

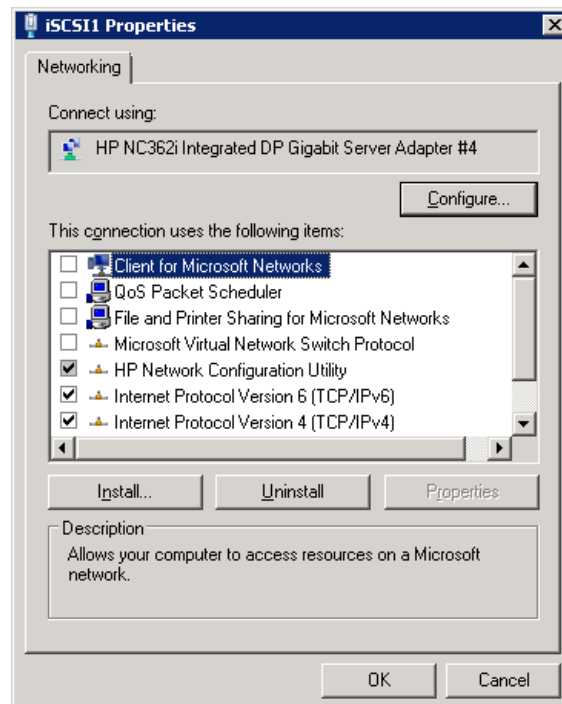
In a test lab, you won't necessarily need to do this step. But it doesn't harm to get used to it for (hopefully) later production deployments.

There are a number of NIC roles to look at. Unbind everything from the VM NIC except for the Microsoft Network Switch Protocol. Leaving this NIC with unnecessary protocols, such as IP, could create a security vulnerability. For example, it could pick up a DHCP address from a VM VLAN, thus giving the host network access to a VM network that it should not have or vice versa.



Note: in my lab, I have used HP servers. The HP Network Configuration Utility is the HP NIC teaming solution. I have left this bound because it will be uninstalled from the server if I unbind it. Please read documentation from your OEM to learn how to do NIC teaming in Hyper-V.

The iSCSI NICs are left with just the IP protocols bound.



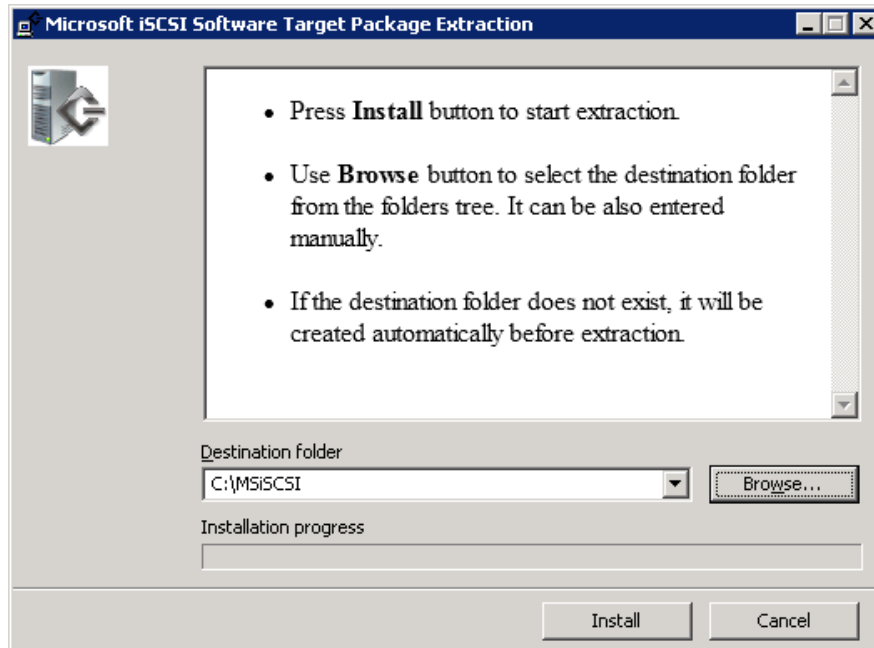
Other things to note:

- You can unbind IPv6 from the NICs if your network does not use IPv6. However, Microsoft has not tested this configuration in Failover Clustering. It probably will never cause an issue at all if you do unbind IPv6. However, it is probably safer to leave it bound.
- You need to leave NTLM authentication enabled for the Hyper-V hosts for CSV to function.
- The CSV network also requires the typical Microsoft networking protocols to remain bound for CSV to function.
- You can enable Jumbo frames on the Live Migration network (from end to end) to improve Live Migration speeds.
- Some people recommend moving the Parent NIC to the top of the binding order.

Installing the iSCSI Target

You can download the free iSCSI target from Microsoft. Use a search engine to find “Microsoft iSCSI Software Target”. The current version at the time of writing was 3.3.

This will download a single file called *iSCSITargetDLC.exe*. Run this file on your desired storage server. You will be asked to specify a location for extracting the contents of this file. *C:\MSiSCSI* was specified in the below screenshot.

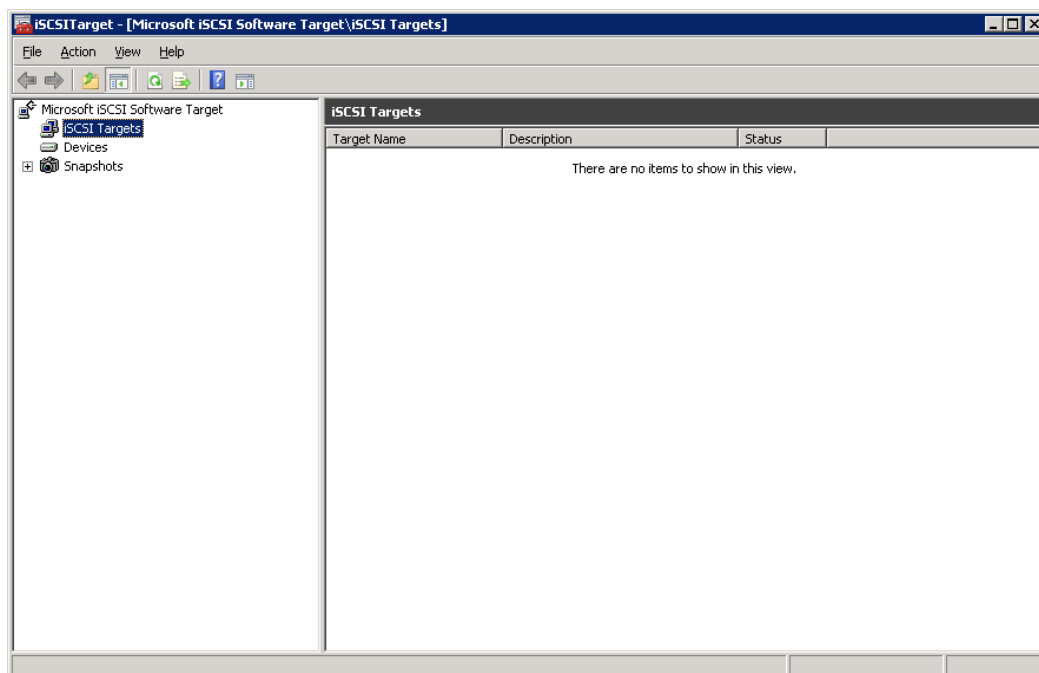


A web page can be run from the extracted contents (*Index.htm*). You can see this page below. Click on the iSCSI Software Target (x64) hyperlink.



This will start the installer. You may need to allow the browser to start it. A splash screen will appear and you will have to accept the following EULA. You can probably accept the default installation path to keep things simple. The iSCSI target software is installed.

A new administrative tool called Microsoft iSCSI Software Target is added to Administrative Tools on the server. This tool is shown below.

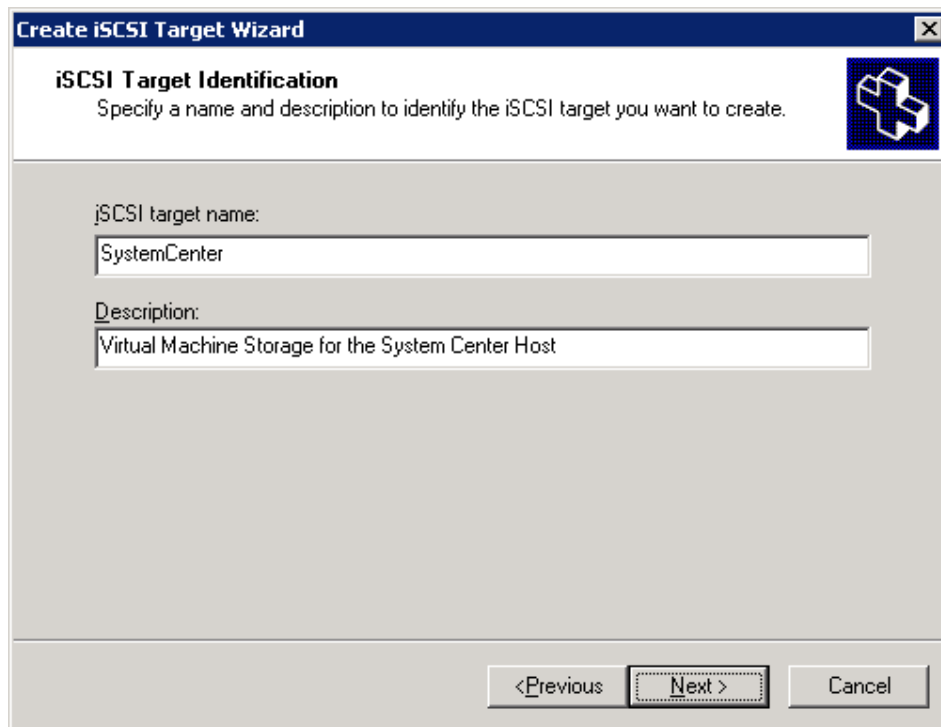


System Center Storage on the Target

In this section we will add a target and devices to provide virtual machine storage for the System Center Hyper-V host.

Adding the System Center Target

Launch the iSCSI target admin console, right-click on iSCSI Targets, and select Create iSCSI Target. This will open the wizard in the following screenshot. You can skip the greeting and describe the new target in the identification screen as is displayed here.

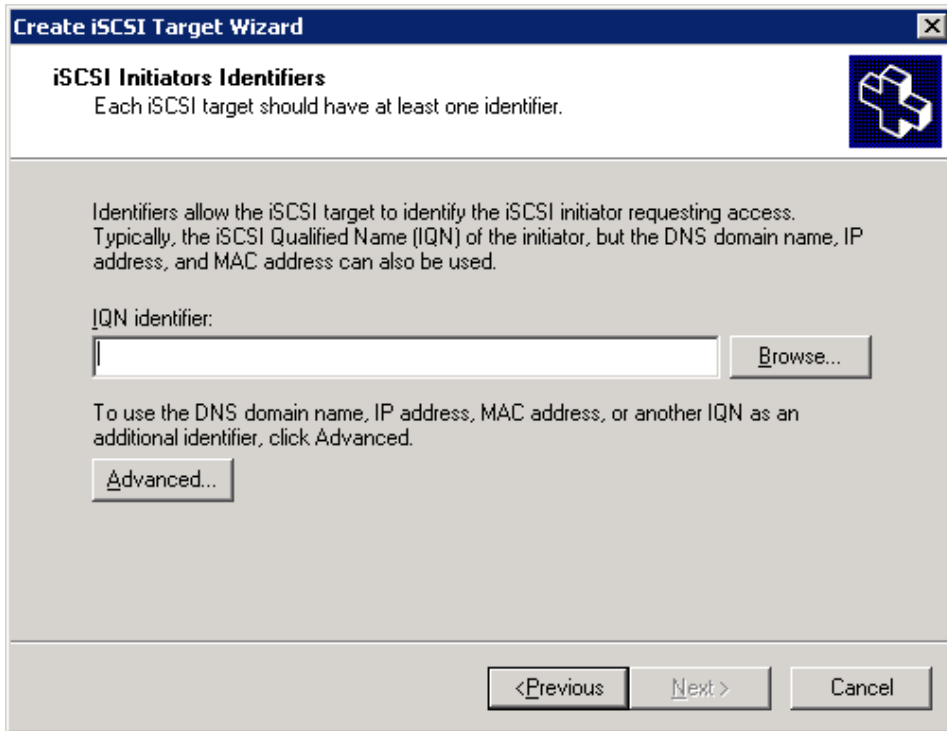


The identifiers screen (below) allows you to specify:

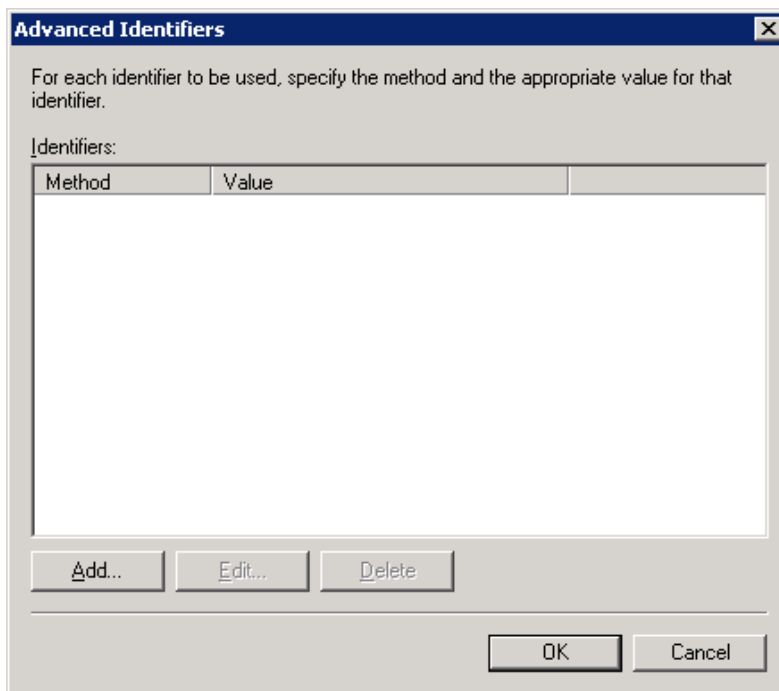
- What iSCSI initiators (clients requiring storage) can connect to the target
- How they authenticate

You can have many iSCSI initiators connecting to a storage server. You can create many iSCSI targets. For example, we are creating a target just for the System Center host called SystemCenter. That will allow the System Center Hyper-V host to see its storage. Later we will create a target for the clustered Hyper-V hosts. Those hosts will be able to see the storage associated with that target. They will not be able to see the storage associated with the SystemCenter target.

The screen below is quite basic. We can click Advanced to specify multiple machines or to use an access method other than IQN.



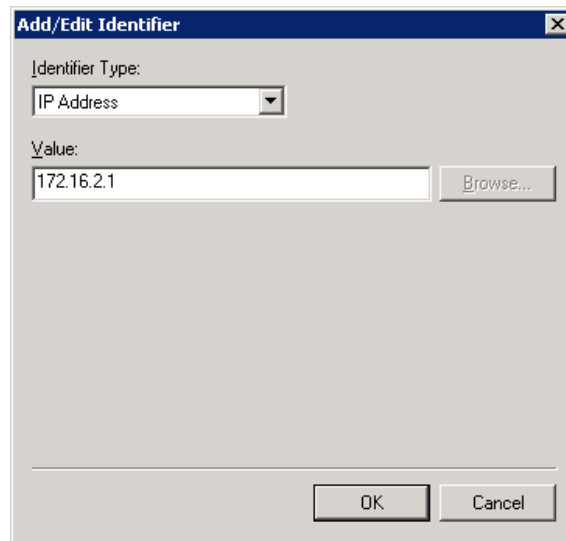
That brings up the below Advanced Identifiers screen.



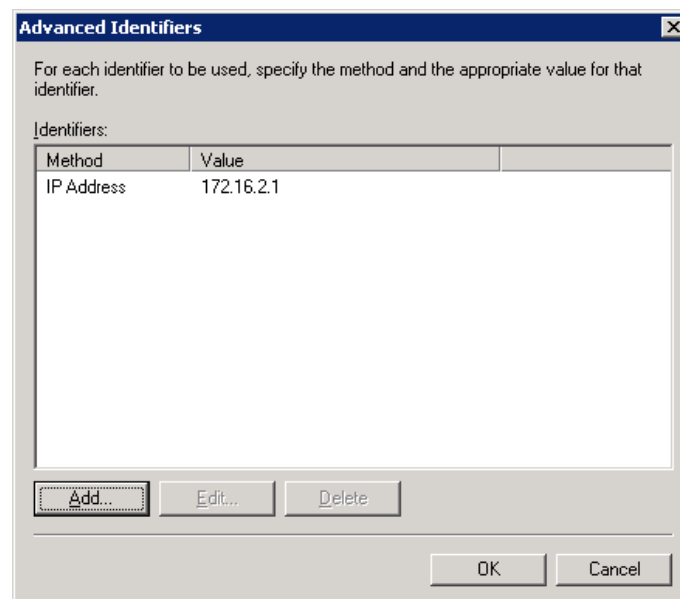
Click Add to add an identifier, i.e. specify an iSCSI client/initiator. There are 4 methods that you can use:

- IQN: the iSCSI Qualified Name for the client in question.
- DNS Domain Name: the DNS name for the client.
- IP Address: the IP address of the iSCSI NIC being used on the client.
- MAC Address: the MAC address of the iSCSI NIC being used on the client.

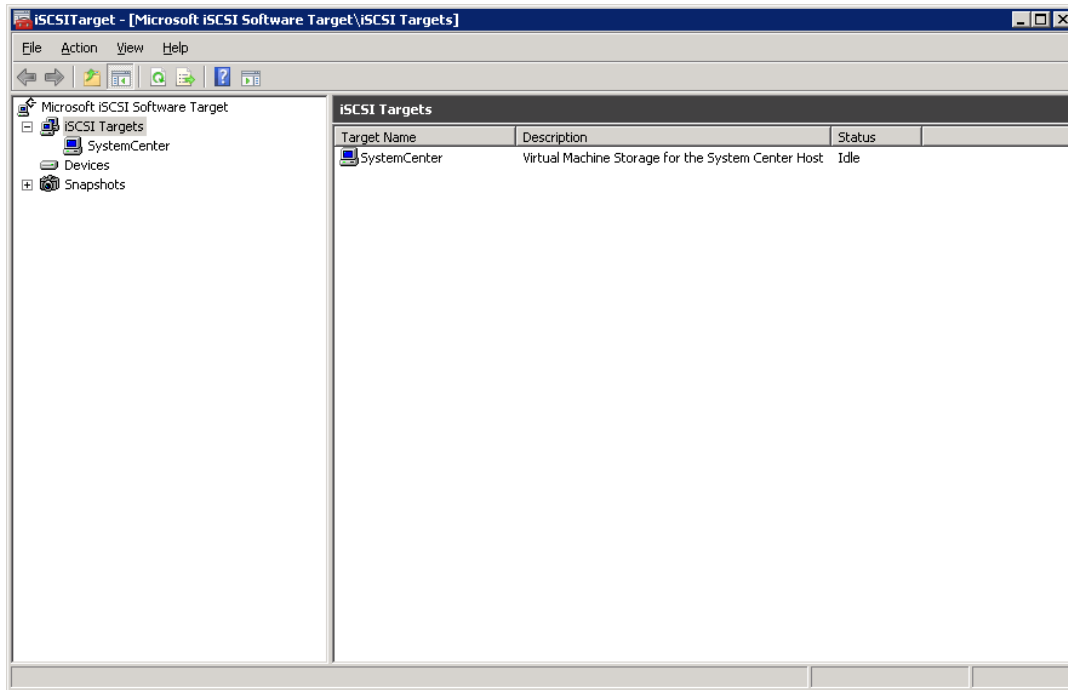
This is a lab so simplicity is required. Select IP address as the identifier type. Enter the IP addresses of the iSCSI NIC used in the System Center host.



The IP address of the System Center host iSCSI NIC should be displayed as shown below.



Finish the wizard and your new target is visible in the iSCSI target admin console.



The System Center host now has an iSCSI target to connect to. Now we need to give it a reason to connect. We will provision some storage.

Adding System Center Storage

You have two ways to add storage:

1. Add storage under *Devices* and later associate it with the iSCSI target.
2. Create the storage by directly adding it to the iSCSI target

I am lazy so we will do the latter. Right-click the SystemCenter target and select Create Virtual Disk For iSCSI Target. Note the option to Add Existing Virtual Disk To iSCSI Target.

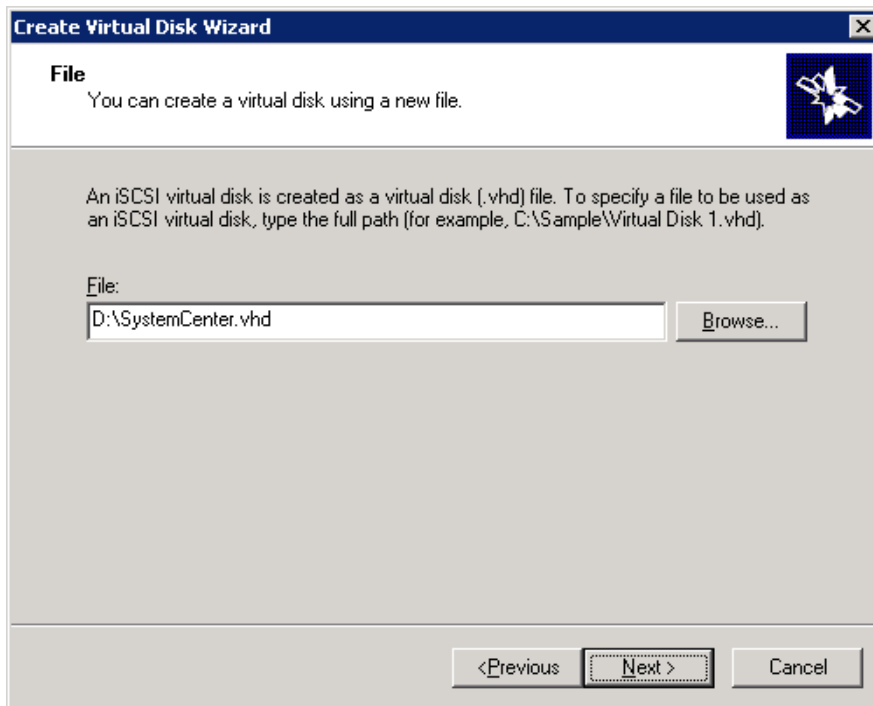
Skip the splash screen.

The Microsoft iSCSI target creates fixed-size VHDs (devices) on an NTFS partition on the storage server. Each VHD (device) is seen as a LUN by the iSCSI initiator. Do not confuse these with virtual machine VHDs. We aren't even to the point of talking about virtual machines yet.

You could optimise performance of your storage by having a dedicated physical LUN in your storage server for each VHD. For example, you could create a VHD/device on a RAID 10 LUN if you wanted some high speed storage, and you could create a VHD on a RAID5 LUN if you required economic storage.

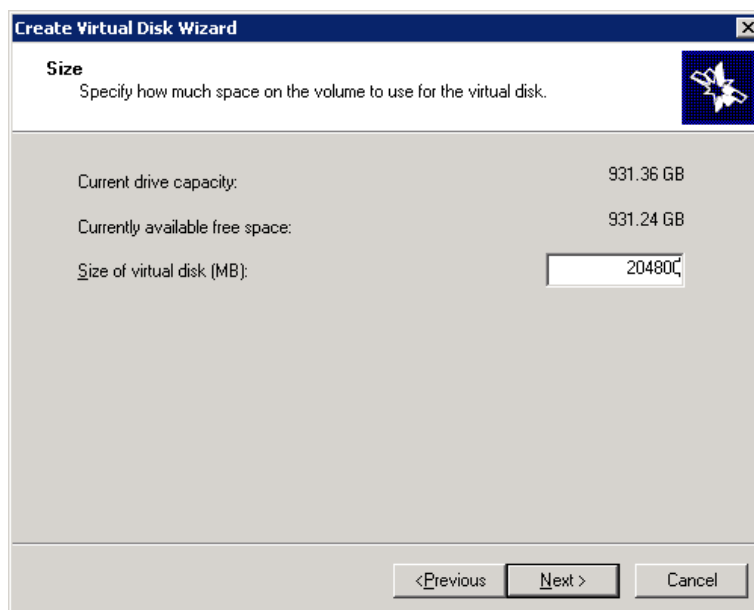
This storage server is a lab machine and has a D: drive for all VHDs (devices) that will be used for iSCSI storage. We will create a VHD/device called *SystemCenter.vhd* on the D: drive of this storage server.

The System Center host will connect to the VHD/device and see it as a volume. We can then create files, such as virtual machines, and store them in that iSCSI connected volume. All of that data will be contained within the *SystemCenter.vhd* file on the storage server.

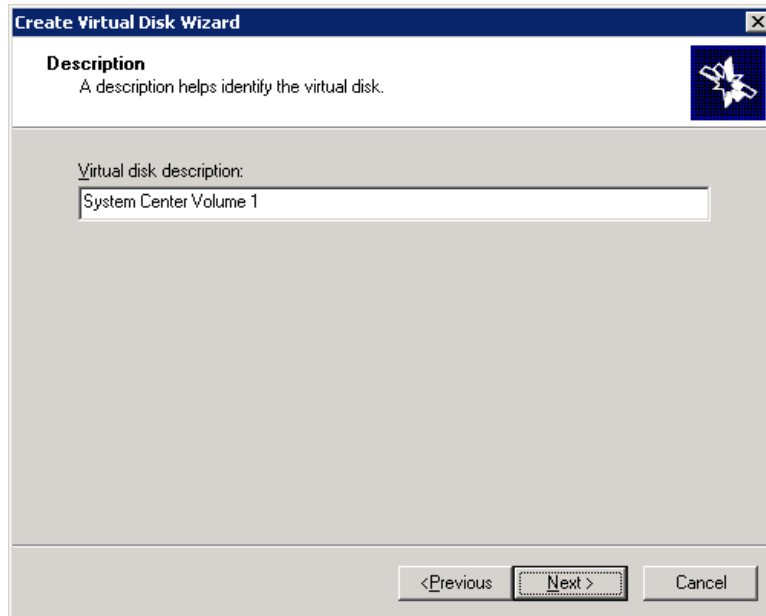


The next screen is a little confusing so be careful. It shows you the total size of the volume where you are creating the VHD/device, as well as the available free space. That data is shown as GB.

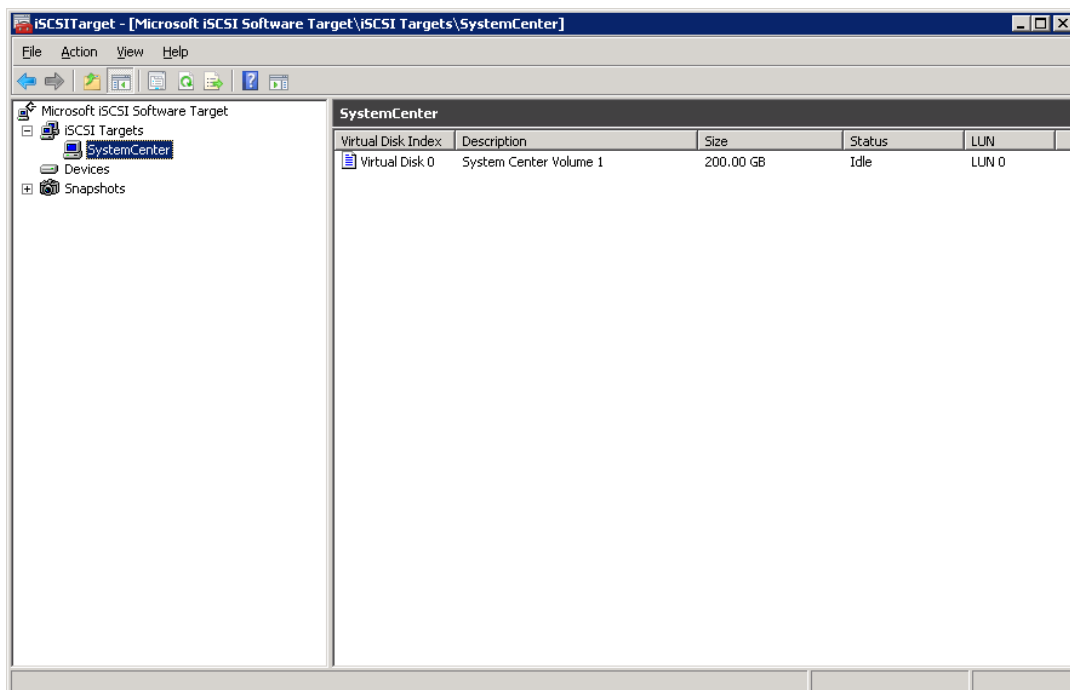
You are asked to specify the size of the new VHD/device in MB. I am creating a modest 200 GB VHD/device which is 204800 MB.



Finally you add a description for the VHD/device. Remember that you could have many VHDs or devices over time.



The new device is visible in the iSCSI admin console:

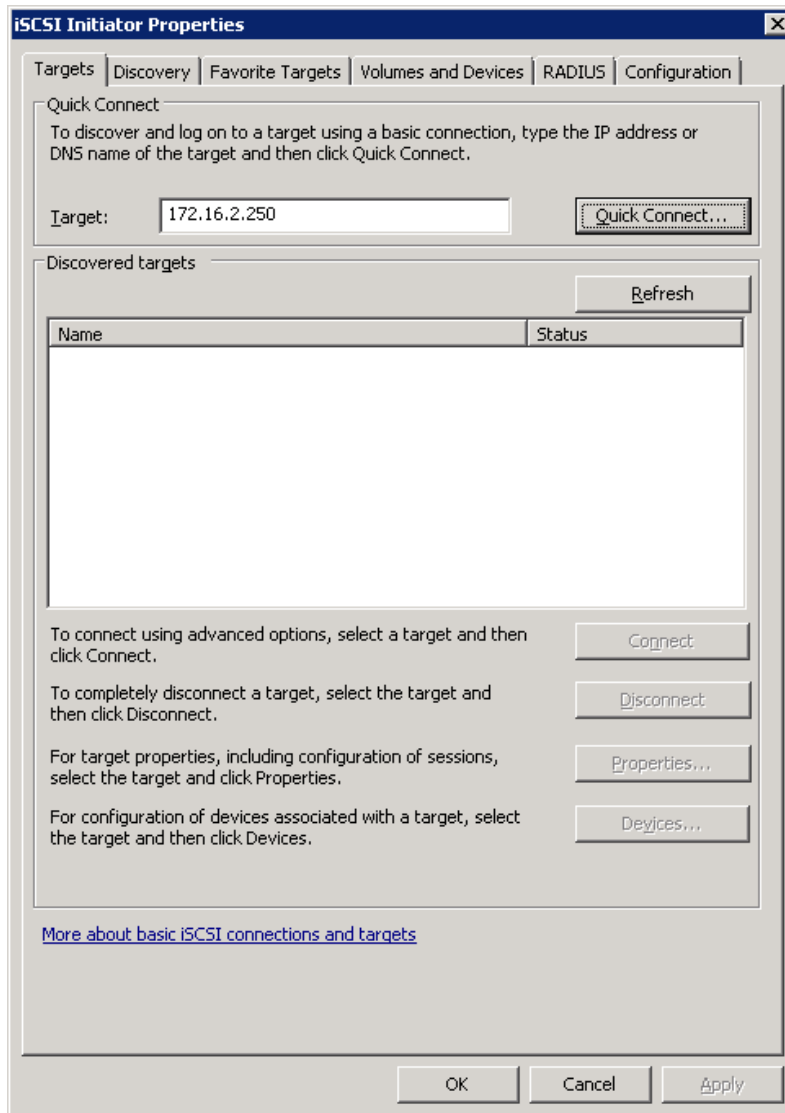


The System Center iSCSI Initiator

Log into the System Center Hyper-V host. Launch iSCSI Initiator from Administrative Tools.

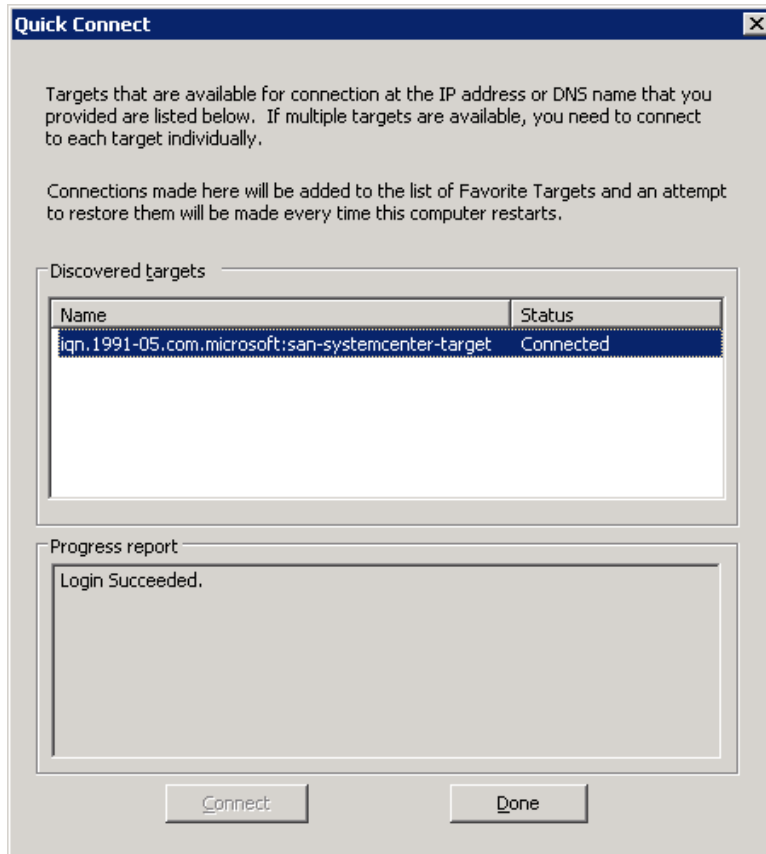
Add the System Center Target

You should type in the address of the iSCSI target (the storage server's NIC for receiving iSCSI communications) and click Quick Connect.

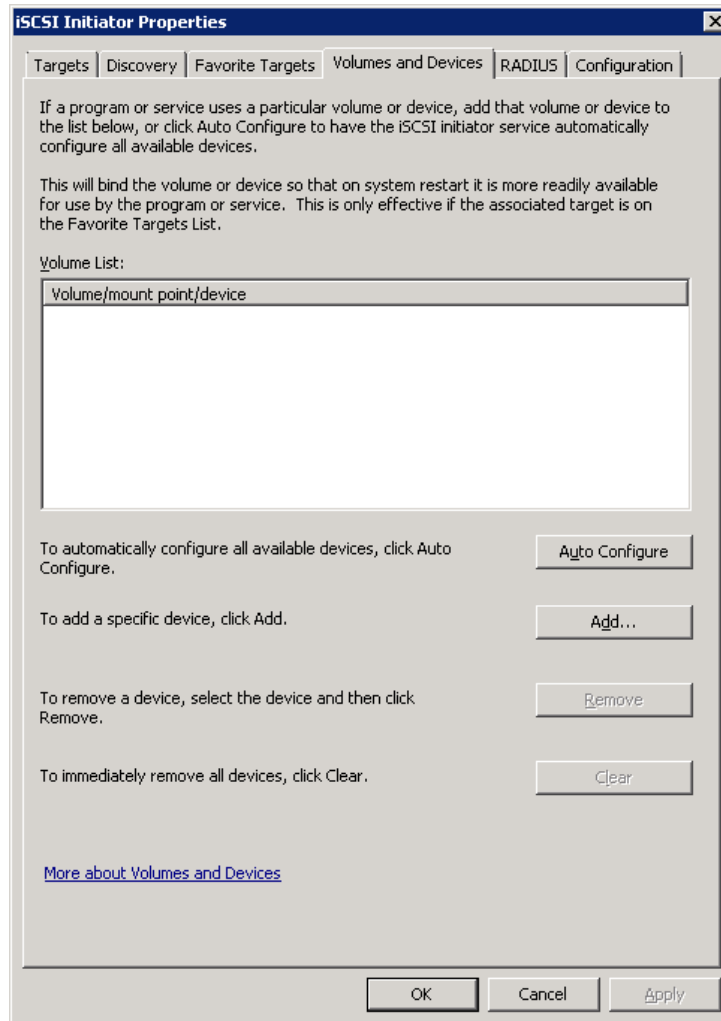


Assuming that you configured the IP address of the iSCSI NIC on the System Center host and the iSCSI Identifier on the storage server target correctly, this should succeed. If it doesn't succeed, check that you can ping successfully, and recheck your IP configurations and target configuration.

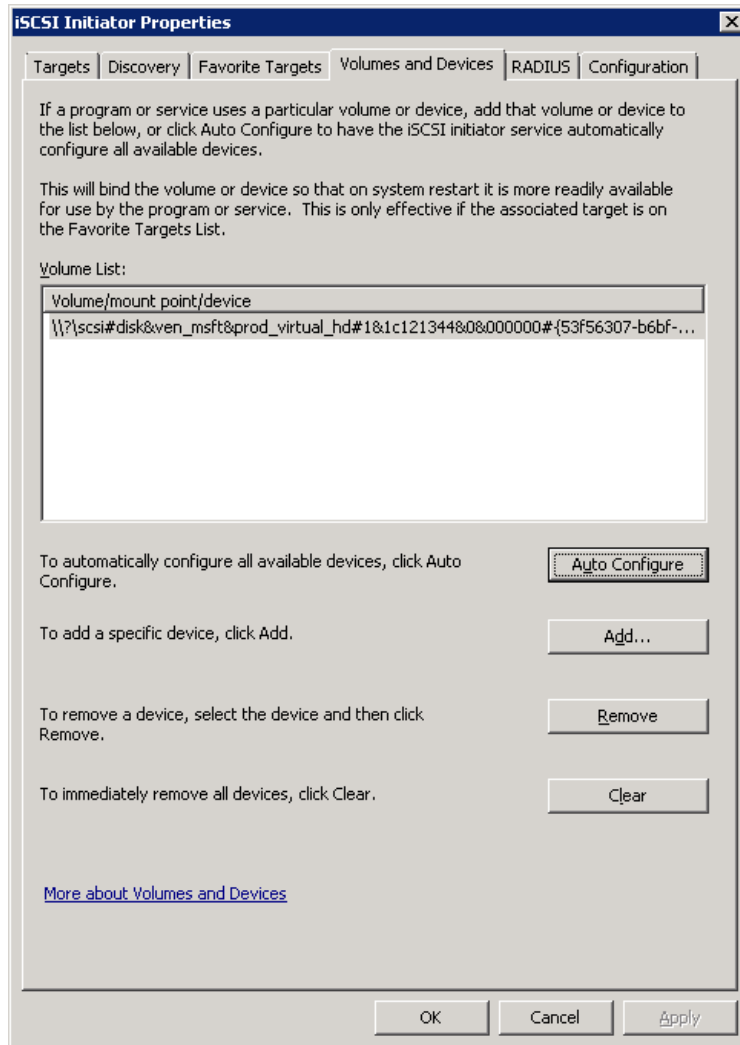
If it does succeed then you'll get this screen (not the bottom half where the progress report is).



Browse to the Volumes And Devices tab and click Auto Configure to quickly add the available devices as volumes on this server.



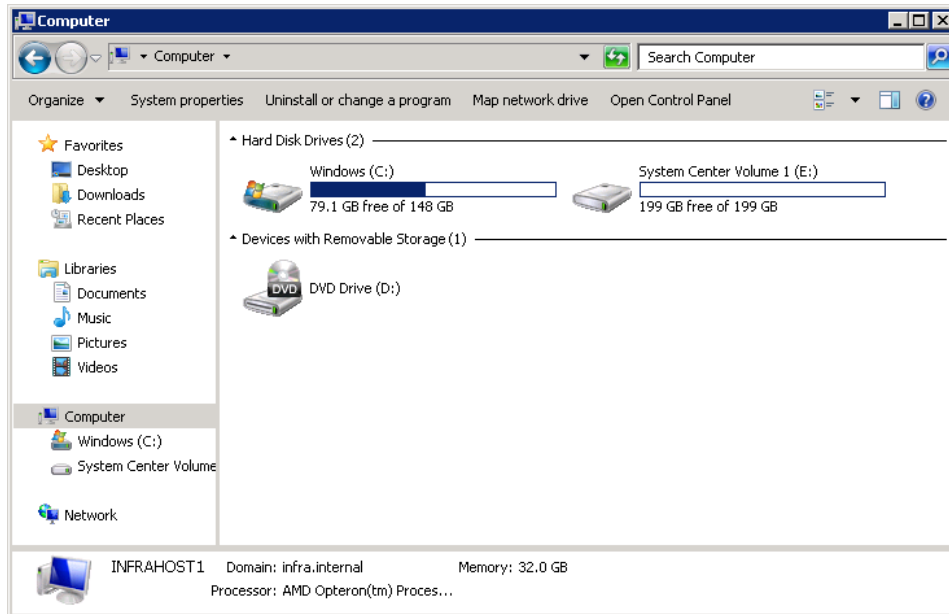
This will find every VHD/device that is associated with every iSCSI target that this server can see. In this case, it should find one device, as shown below.



Formatting the System Center Volume

This is good ol' fashioned disk management. Launch Disk Management, bring your volume online, initialize it (GPT for multi-TB volumes), and format it. I suggest labelling it using the name that was used to create the device on the storage server. The end result should look something like the below Windows Explorer screenshot.

This System Center host has one internal C: drive. The E: drive is the iSCSI mounted volume that we earlier created as a VHD/device, which was associated with the iSCSI target, and permitted the System Center host's iSCSI NIC IP address to connect.



You now have a host that is ready to deploy virtual machines on. In this lab, those virtual machines would run the System Center components such as:

- System Center Virtual Machine Manager.
- System Center Operations Manager.
- System Center Data Protection Manager – don't use VHDs for the storage pools. Instead, provision a new target on the storage server, and use the iSCSI initiator in the DPM VM to connect to the storage.

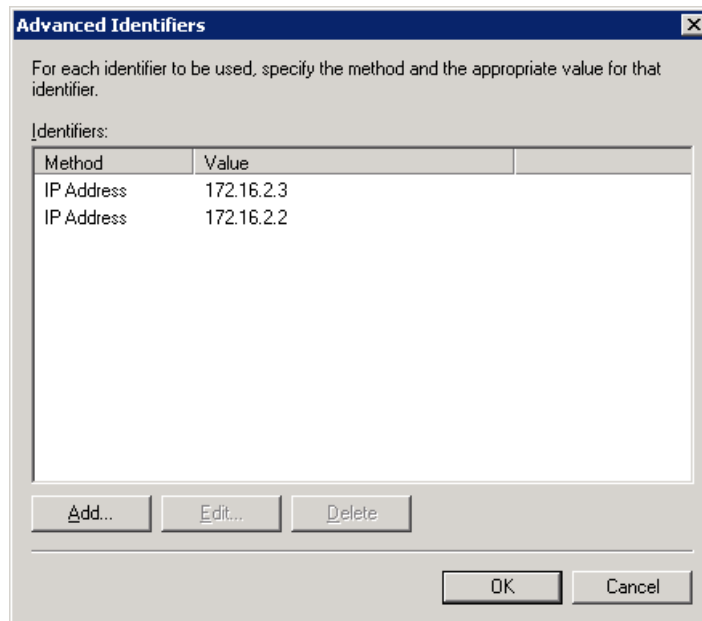
Now we get to do something similar for the cluster hosts and create the Hyper-V cluster!

Hyper-V Cluster Storage on the Target

This section will deal with creating storage on the iSCSI target for the Hyper-V cluster. The cluster will comprise of Host1 and Host2.

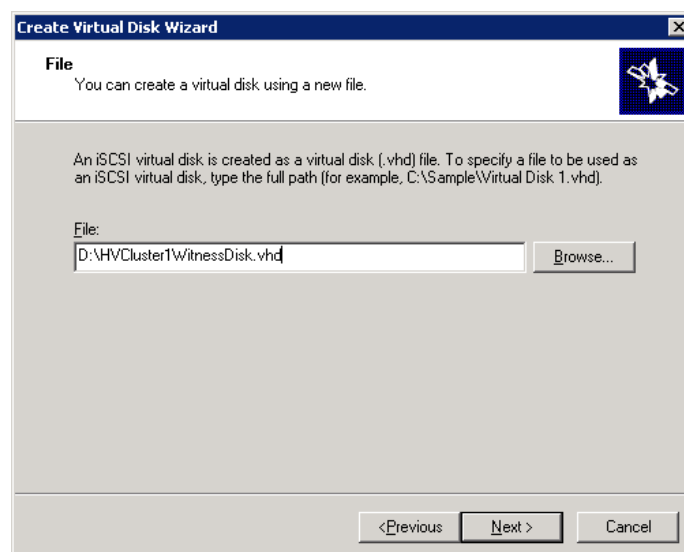
Adding the Hyper-V Cluster Target

Create a new target as you previously did for the System Center host. However, this time you will specify the 2 IP addresses used by the iSCSI NICs of both Host1 and Host2.



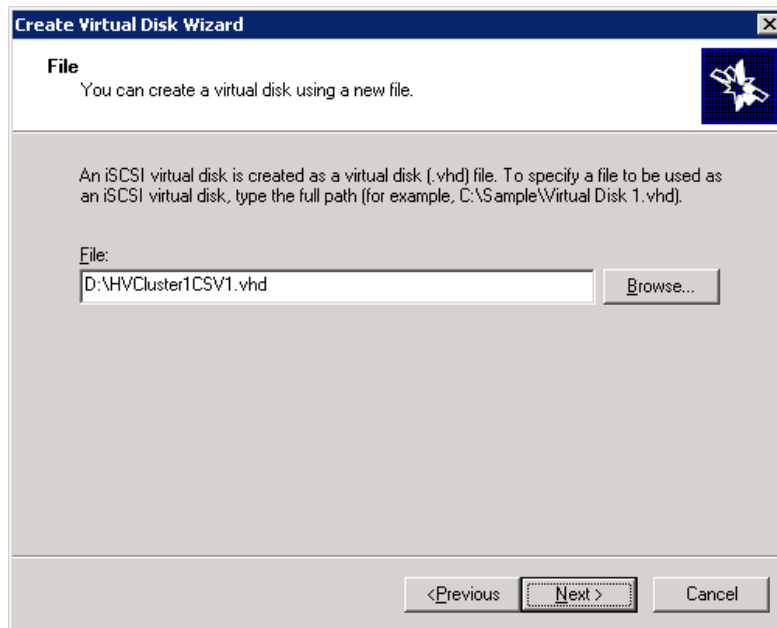
This will be a two node failover cluster so it will require a witness disk (or quorum disk depending on the screen or documentation you are reading).

Create a new device and attach it to the target. It will only need to be 1024MB in size.

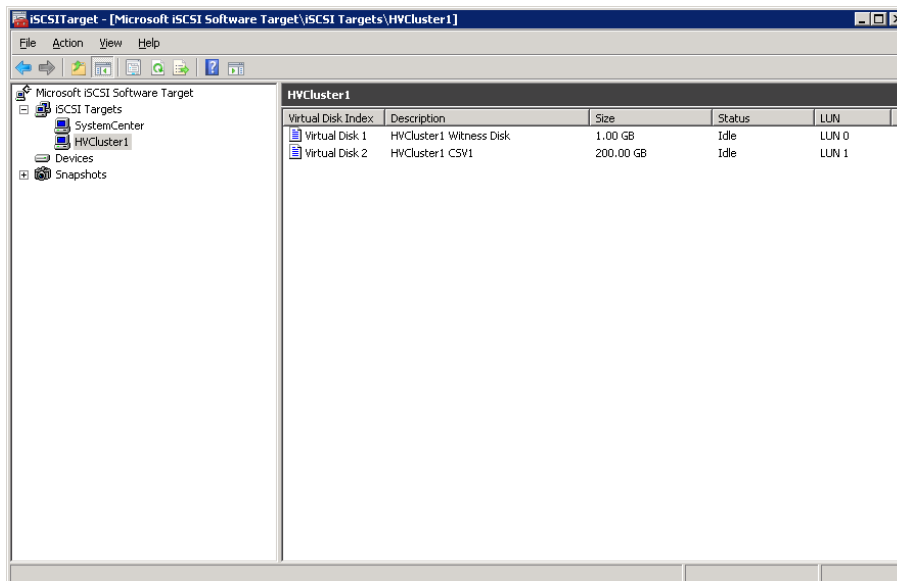


That first VHD/device will only be used as a witness disk. What use is a Hyper-V cluster without any storage for virtual machines? Now we will add a second VHD/device to the target. This will be used

to create a Cluster Shared Volume (CSV) for the cluster. We will eventually have more than one CSV in the cluster so this will be known as CSV1. Make it 20 GB (204800 MB) in size.



You now should have two devices in the Hyper-V cluster target.



Adding the Hyper-V Cluster Storage

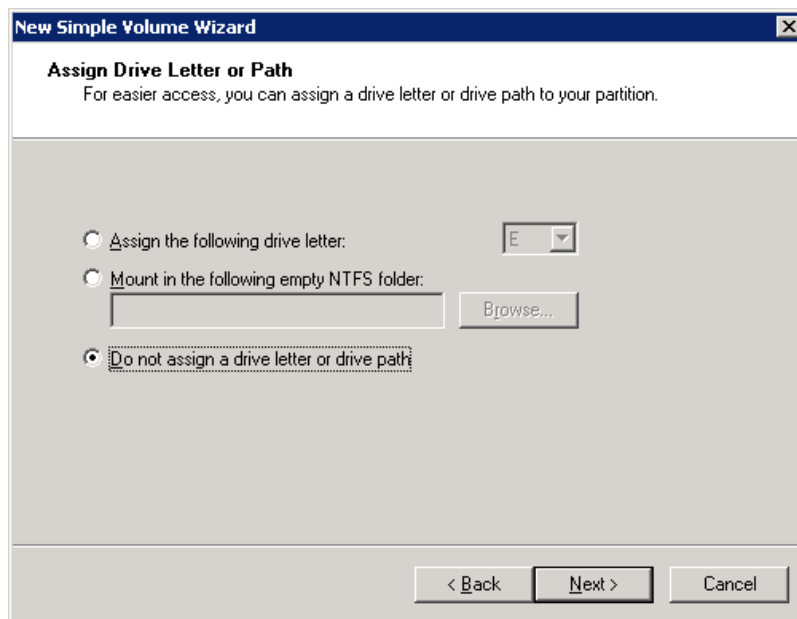
In this section, you will see how to connect each of the soon-to-be-clustered Hyper-V hosts to the storage that was just created on the storage server.

The process of adding storage for each of the cluster nodes is identical to what we saw earlier for the System Center host. You will do the following:

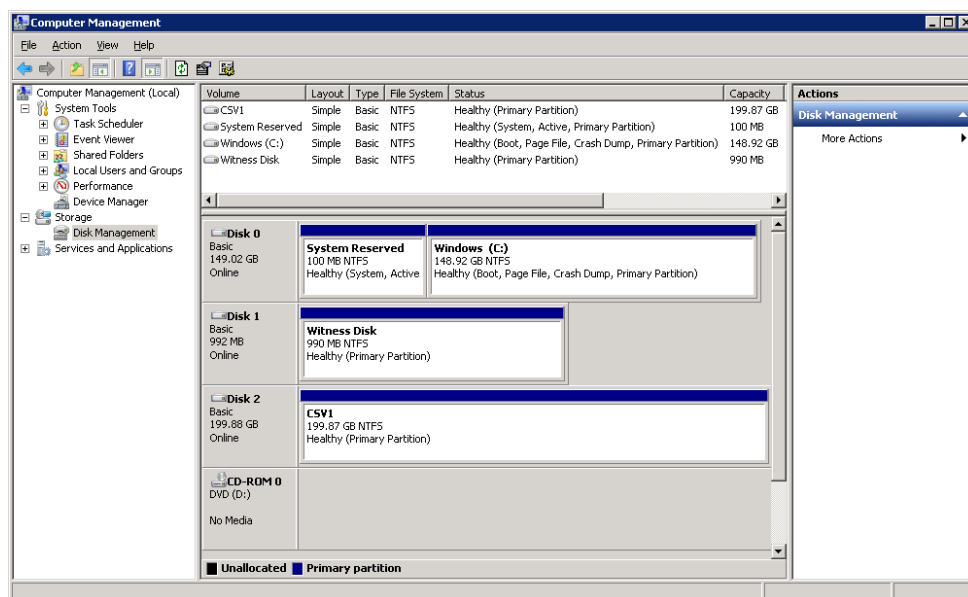
1. Log into Host 1.
2. Add the target 172.16.2.250 in the iSCSI initiator.
3. Use Auto Configure to add the Volumes and Devices. Both of the configured VHDs/devices (witness and CSV1) will be added and appear.

4. Log into Host 2.
5. Add the target 172.16.2.250 in the iSCSI initiator.
6. Use Auto Configure to add the Volumes and Devices. Both of the configured VHDs/devices (witness and CSV1) will be added and appear.
7. Log into either host.
8. Use Disk Management to bring the new volumes online, initialize them, and format them. Label the disks using the names of the VHDs/devices for name standardisation. You **should not** repeat this step on other hosts in the cluster.

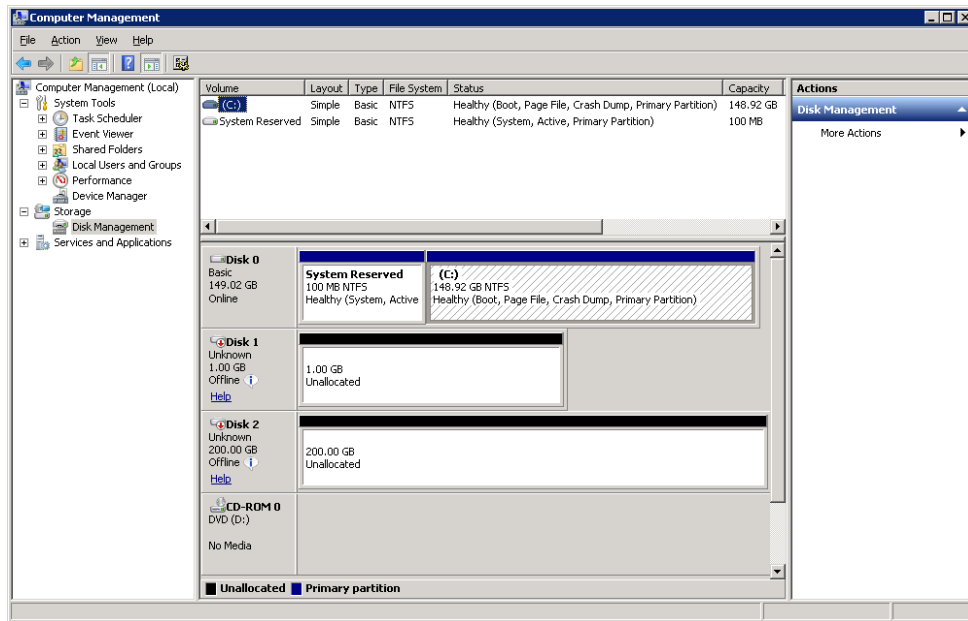
Use the option to not use a drive letter (GUID drives) when you format the cluster volumes (as seen below). It is pointless giving these drives a letter. The witness disk shouldn't be used for anything other than providing quorum to the cluster. And the CSV volume will be reveal via CSV.



You can see in the following screenshot how the disks appear after you have formatted them.



You should log into the other host to verify that the disk are visible. The disks will not be online (see below). That is because Windows has a “shared nothing” rule when it comes to storage. Only 1 server can bring a disk online at a time, even if more than one machine is able to see the disk. Don’t go messing with bringing the shared disks online in here.



Failover clustering manages this operation for us by treating the cluster disks as resources. Resources can be failover over and back between hosts. In Windows Server 2008 Hyper-V clusters, this meant that a volume could only be used by one host, and normally meant that we ran one virtual machine per LUN or volume. That was a nightmare to manage, especially when decommissioning virtual machines – you had to double or triple check that any storage you were about to delete was no longer being used, otherwise you could erase a production VM!

Windows Server 2008 R2 introduced Cluster Shared Volume (CSV). This allows many virtual machines, running across many clustered hosts in the same host cluster, to reside on a single shared volume. They can live migrate between hosts, independent of each other. Despite this sharing, we still have the rule of “shared nothing” with the CSV volume. One host, the CSV coordinator (a role that can failover), owns the volume. It delegates read/write permissions to VM files to those hosts that are hosting the VMs. However, low level operations such as VSS backup of the CSV, require that the CSV coordinator have exclusive access to the volume. This causes Redirected I/O to initialise across the CSV LAN (see the NIC configuration in the architecture) where non CSV-coordinators will redirect their storage I/O traffic from their direct storage link via the CSV coordinator for all traffic related to VMs that are stored on the CSV that is being manipulated.

This means that you need to size your CSVs, plan the number of them, determine the number of VMs per CSV, and plan which VMs/VHDs are stored on which CSV. There are a lot of variables which are out of the scope of this document. However, you can learn more in *Mastering Hyper-V Deployment* (Sybex, 2011), available at <http://tinyurl.com/699626a>.

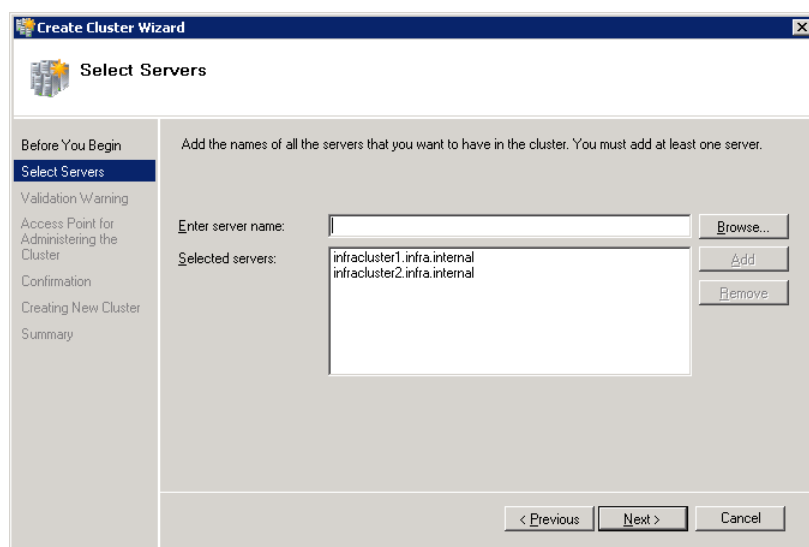
Building the Hyper-V Cluster

Now for the fun bit of this deployment!

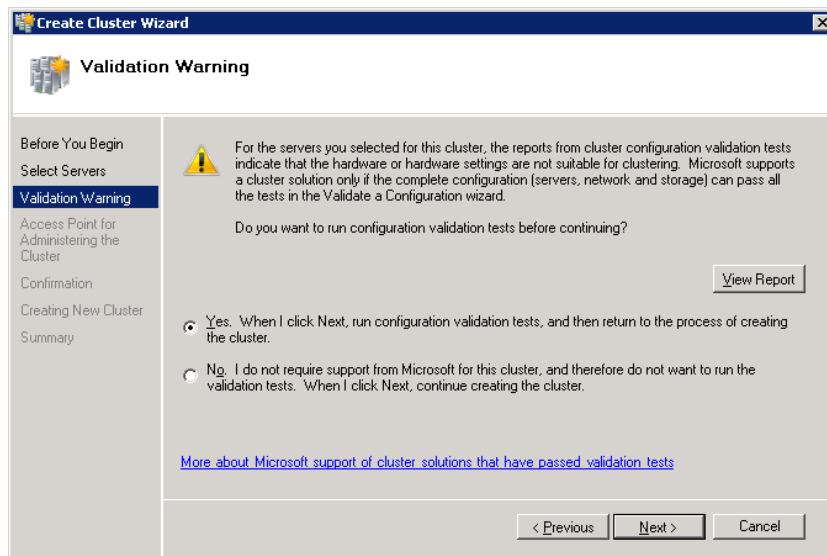
Before you create a failover cluster, have a static IP address on the LAN and a cluster name ready for the new cluster object that will appear in your domain and on your network.

Launch Failover Cluster Manager from Administrative Tools. In there, click the hyperlink or action to Create A Cluster.

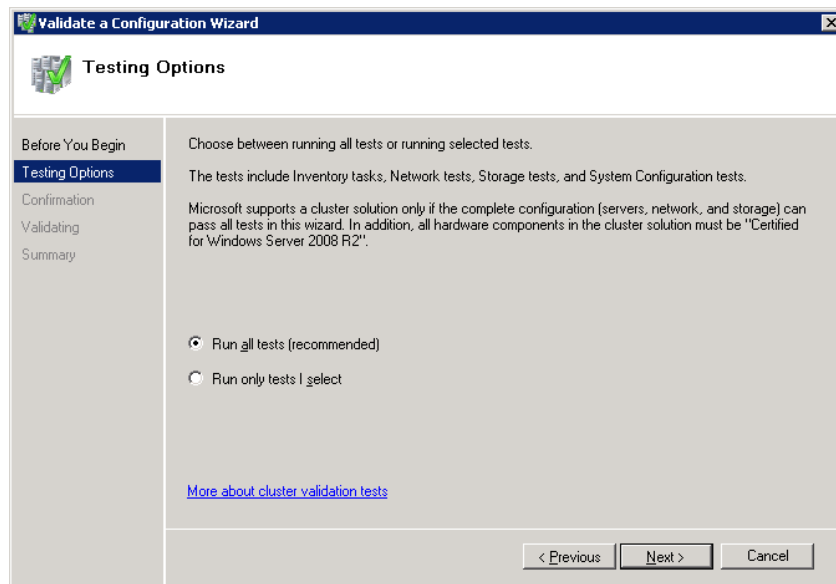
You will be asked to specify the name of each host that you want to add to the new cluster. Type in the names and the machines will be validated. This is the most basic of tests, for example, checking to see that the hosts are responsive.



The real validation tests come next. Microsoft used to have a very specific set of hardware and configurations that they would support in a failover cluster. Now they will support anything where the cluster is validated using the next step (this can be re-run in an existing cluster). It is recommended that you run this validation wizard because it (a) can find configuration issues, and (b) provide your cluster with support from Microsoft.

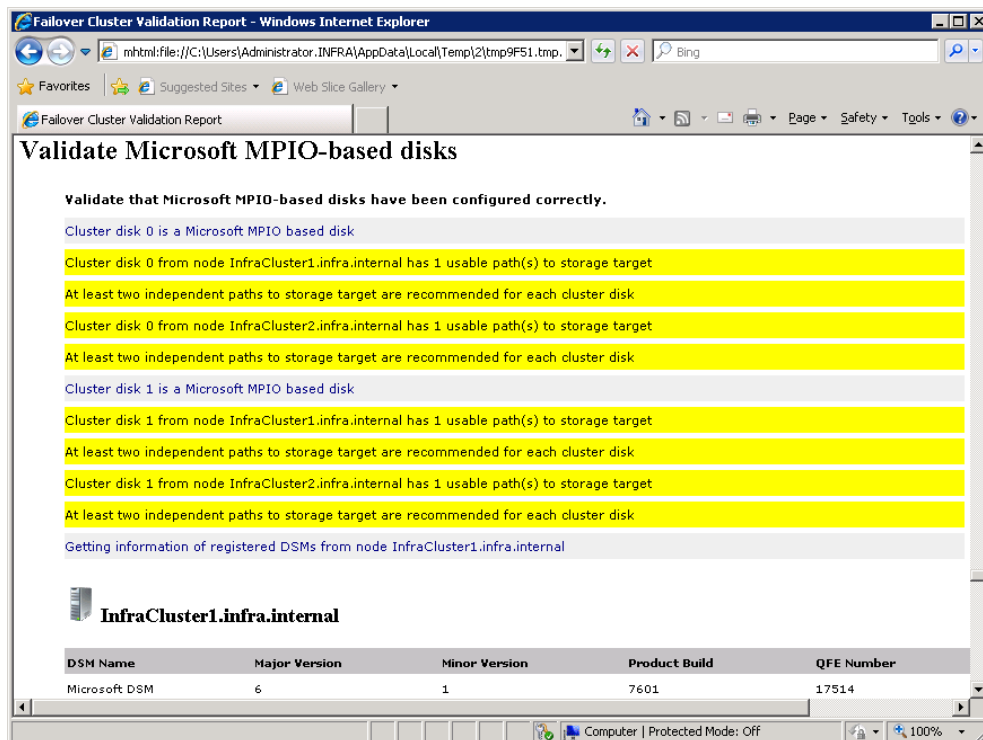


You will have a choice of what tests to run. You can run all tests (recommended) or just a selection of tests that you choose.



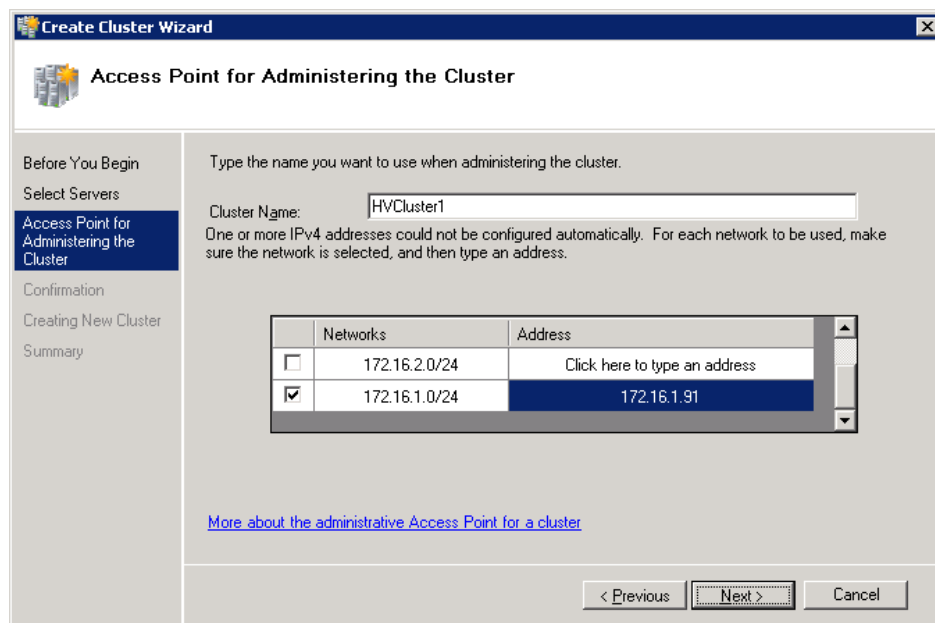
The tests take a few minutes. Eventually it returns with a report (that you can save). The overall result can be a pass (where everything is fine), a warning (where you can create a cluster but there may or may not be an issue later on), or a fail (where you should not create a cluster). You can drill down into each category of test to see the result of each individual test and learn more about any issues that may or may not have been found.

If you followed the architecture in this document then you will get some non-critical warnings including one regarding the storage in this architecture for the clustered Hyper-V hosts. Failover Clustering would prefer to find MPIO disks. But we don't have support to use MPIO with the Microsoft iSCSI Software Target from our clustered hosts so we have not configured it.

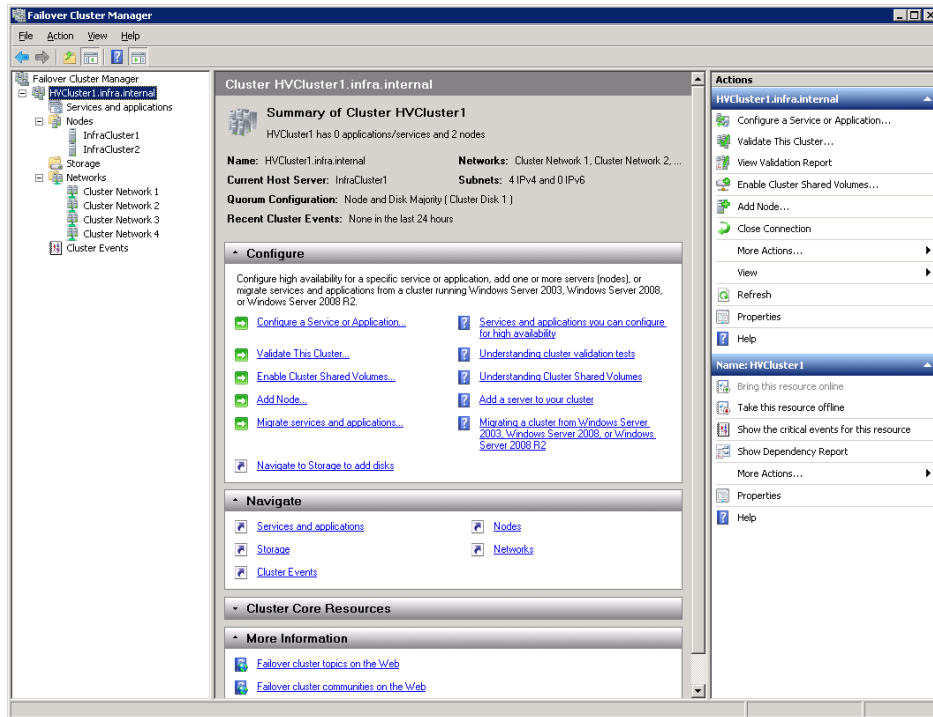


Once done with the validation wizard, you can return to creating the cluster. The next screen allows you to define the cluster object name and IP address as they will appear on the network. An active directory object will be created and its IP address (as will appear in DNS) is configured here.

You only need to configure an IP address for the parent NIC. This means you can de-select the other NICs.



The wizard completes, and hopefully you'll end up with a working cluster in a few moments.



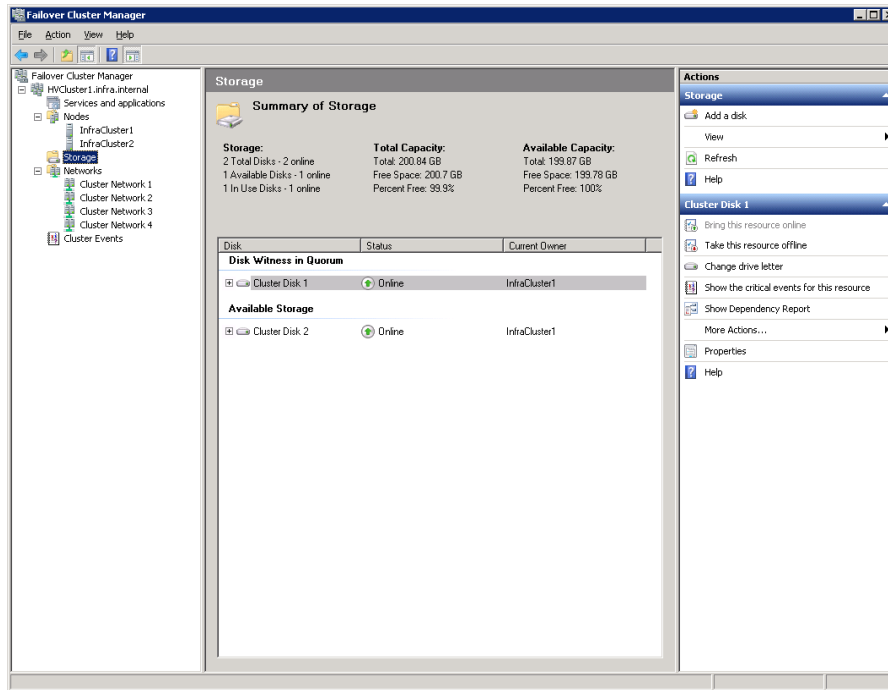
Configuring the Cluster

Don't go rushing into creating a VM. You should do a little tidying up first.

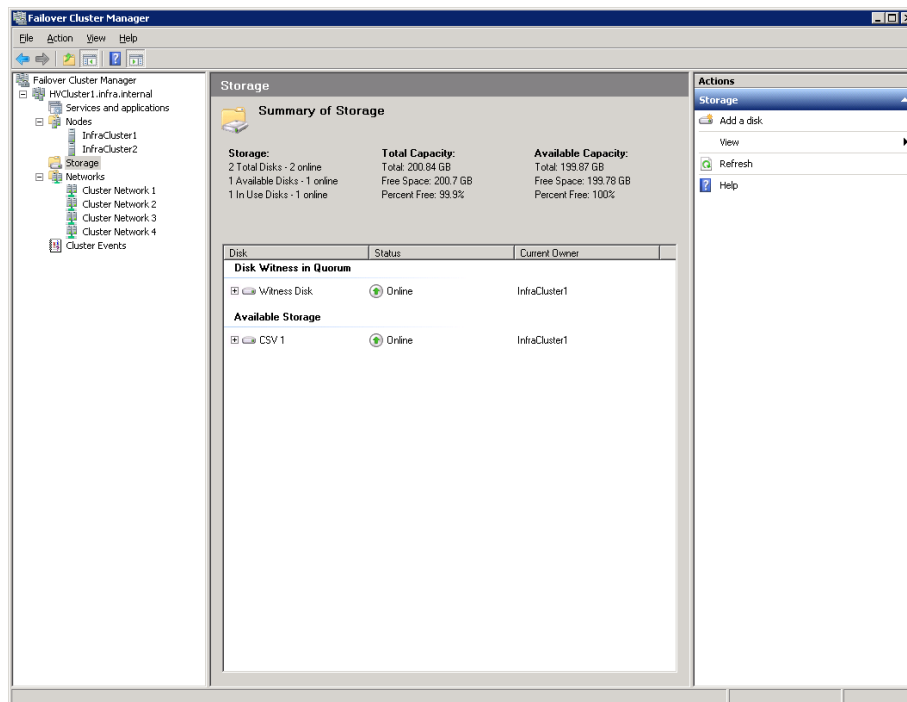
Storage Renaming

First thing to note is that the cluster has automatically selected Disk And Node Majority as the quorum configuration. That's because there are an even number of hosts in the cluster. The cluster automatically selected the smallest, commonly available disk as the witness disk. In this case, that is the 1 GB volume that is on the storage server; Perfect!

If you look at the following screenshot, you can see that the 2 cluster volumes appear as rather anonymously labelled storage devices in the cluster. Imagine if you add more and more volumes. You'll have no idea what is what.

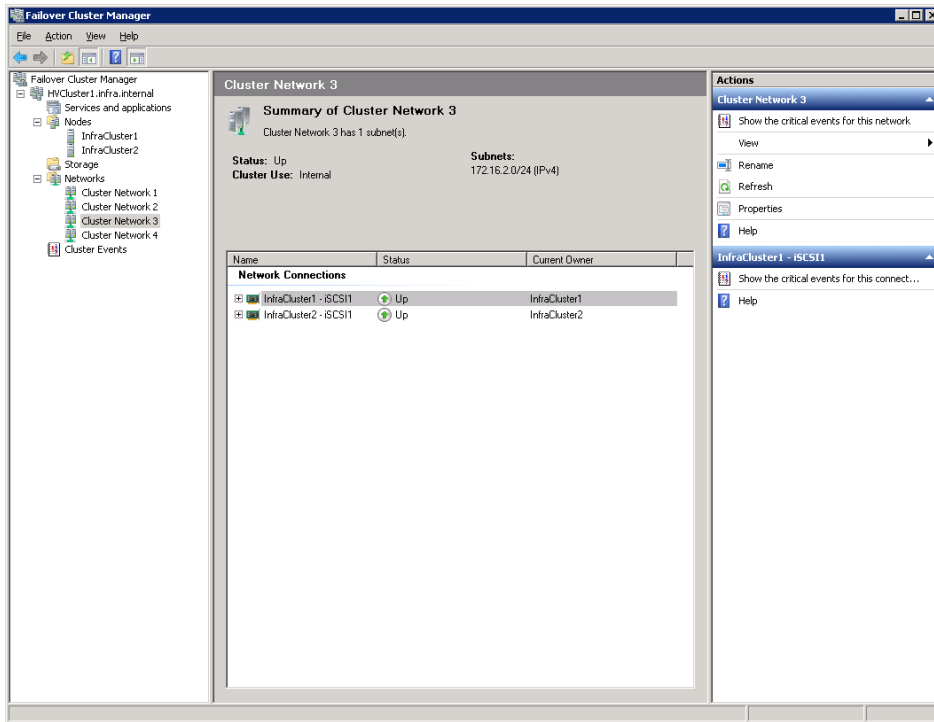


Edit the properties of the storage volumes to match the iSCSI target device names and the NTFS labels. This gives you a form of documentation for the end-to-end storage.

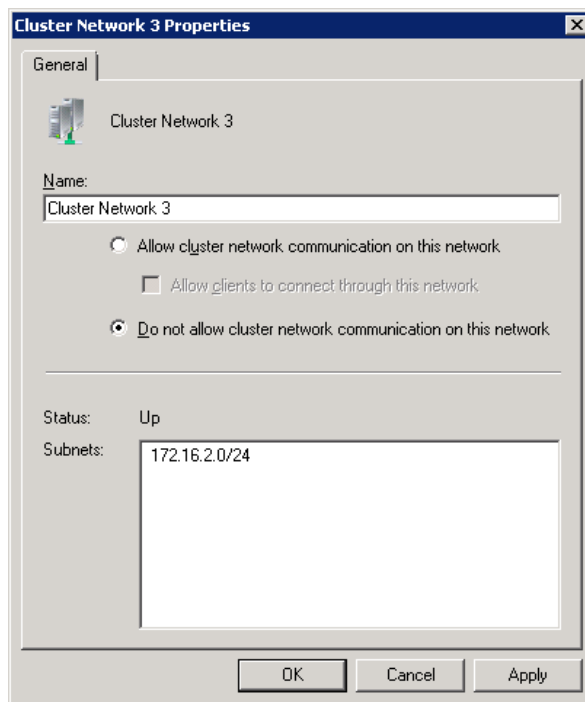


Cluster Network Selection

Browsing through Networks, you will find each of the host NICs that the cluster will try to use for cluster communications. You should find a pair of Parent NICs, a pair of CSV NICs, and a pair of LM (Live Migration) NICs. You can see below, that we also have a pair of iSCSI NICs. We do not want cluster communications to cross that network.



You can edit the properties of a cluster network to select the option to Do Not Allow Cluster Network Communication On This Network.

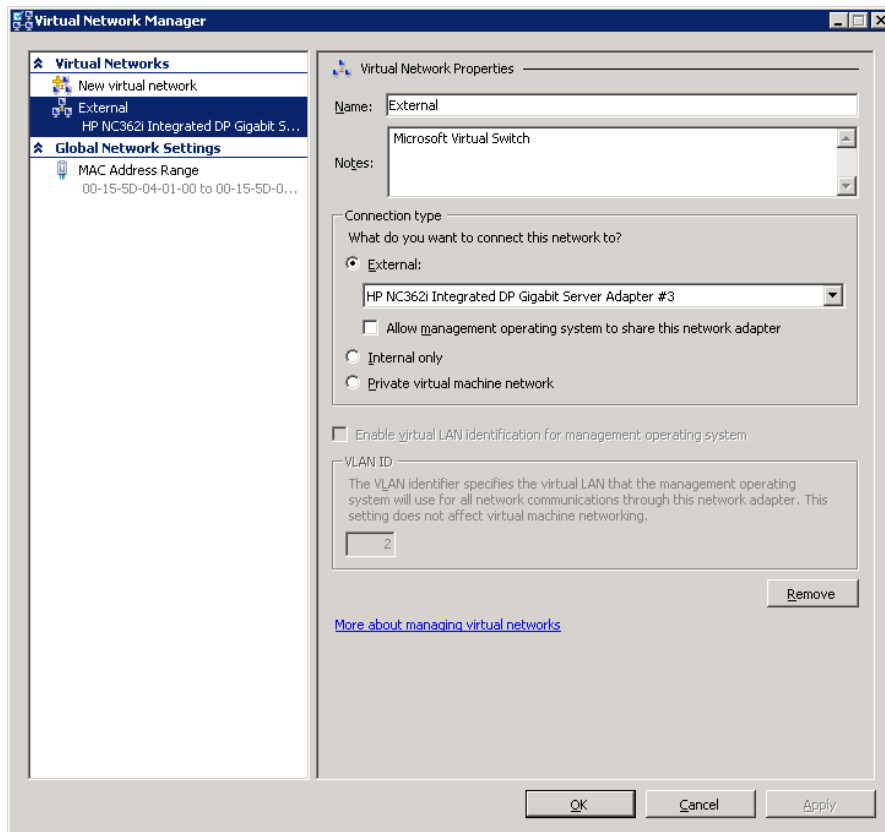


Hyper-V Virtual Machine Networking

The next step is to configure the virtual network (switch) that the virtual machines will communicate on. This is a lengthy subject. We'll be sticking to creating a very basic network here.

Fire up the Hyper-V Manager console for each host in the Hyper-V cluster. Click Virtual Network Manager. Create an identically named External network on each Hyper-V host. Ensure that this is bound to the physical NIC that will be used for virtual machine communications (labelled as VM in this lab). The confusing bit here is that the labels are not presented. You need to find the Device Name from Network Connections.

Clear the option to Allow Management Operating System To Share This Network Adapter. We are doing this because we have a dedicated NIC for the parent (Parent - for management) and a dedicated NIC for virtual machine communications (VM).

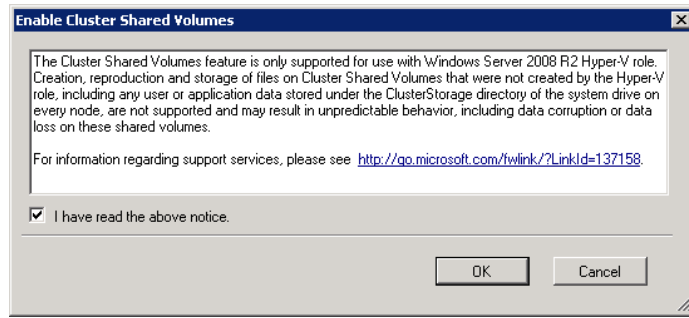


Enable and Configure Cluster Shared Volume (CSV)

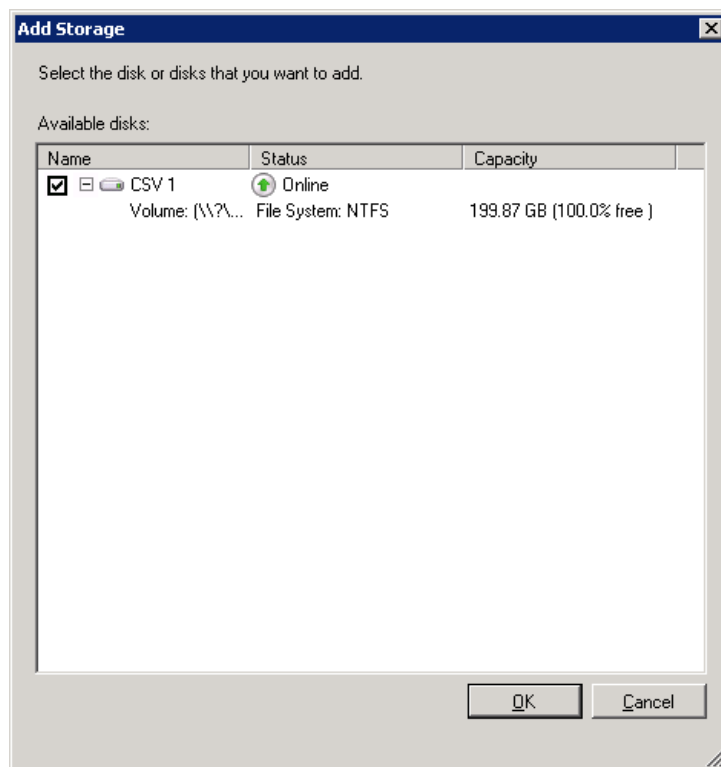
As it stands we have a single volume for storing virtual machines. It is not yet a CSV volume, and can only be accessed by a single host at one time. We will enable CSV to allow all hosts in the cluster to access the volume. Remember that “shared nothing” still applies and the CSV coordinator (a fault tolerant role) is the single owner of the volume.

In Failover Cluster Manager, you will right-click the cluster and select Enable Cluster Shared Volumes.

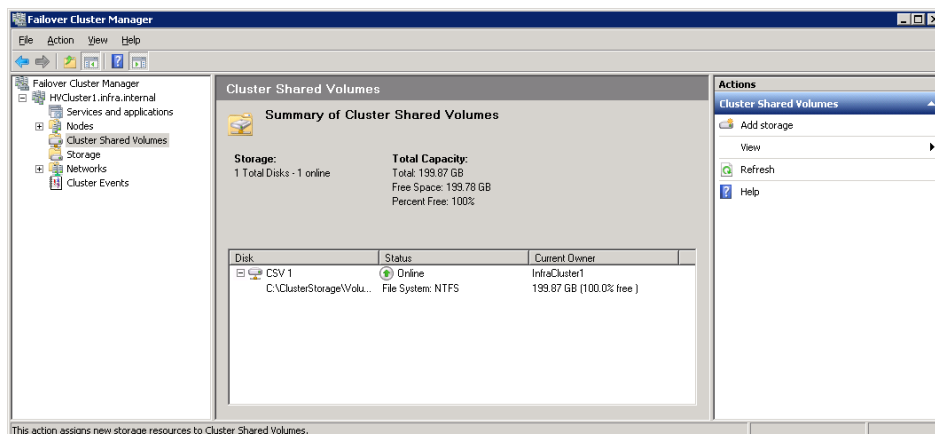
WARNING! CSV is only to be used with Hyper-V. Microsoft are pretty determined that you know this, as displayed in the following warning that you must accept to continue enabling CSV.



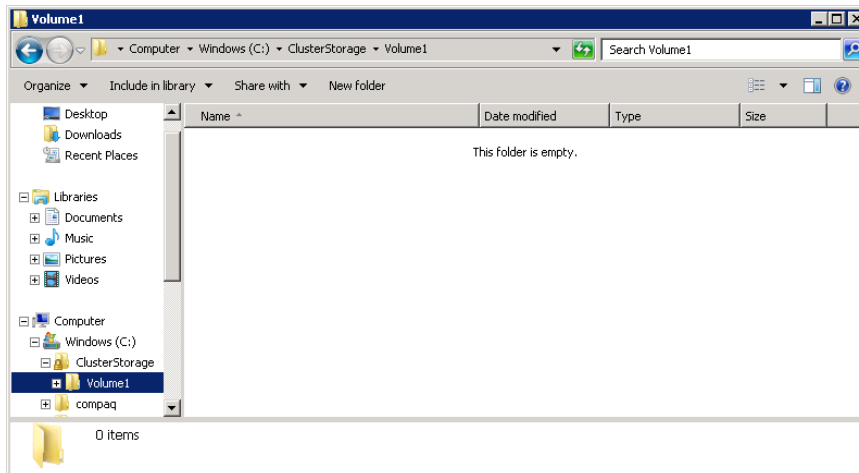
Cluster Shared Volumes now appears in Failover Cluster Manager. Navigate into there and select Add Storage. This will allow you to select the volume labelled as CSV 1 and convert it into a CSV volume.



If all is well with your cluster and networking, the new CSV volume will appear in Failover Cluster Manager.



The CSV volume appears on each host as a mounted volume under *C:\ClusterStorage*.



Specify CSV Network

Failover Clustering will automatically try to configure the private (non-routed) network with the lowest metric as the CSV network (the network used for CSV communications). You can determine which NIC/network is being used for CSV using PowerShell.

Log into a node in the cluster and launch Windows PowerShell Modules from Administrative Tools. This launches a PowerShell window and imports the modules that are required for administration of the roles/features on this server.

Run the following cmdlet:

```
Get-ClusterNetwork | ft Name, Metric, AutoMetric, Role
```

This returns something like the following:

Name	Metric	AutoMetric	Role
Cluster Network 1	1000	True	1
Cluster Network 2	1100	True	1
Cluster Network 3	10000	True	0
Cluster Network 4	1300	False	3

Cluster Network 1 has the lowest metric. That is the network that is currently being used for CSV communications. Returning to Networks in Failover Cluster Manager, I find that this happens to contain the NICs that I labelled as CSV. That is nice! However, if it wasn't my desired CSV network I could manipulate the metric of my desired CSV cluster network to get what I want:

```
( Get-ClusterNetwork "Cluster Network 2" ).Metric = 900
```

That PowerShell cmdlet will reduce the metric of *Cluster Network 2* to be the lowest on the cluster and therefore ensure that it is the preferred network for CSV communications.

Build Some Highly Available Virtual Machines

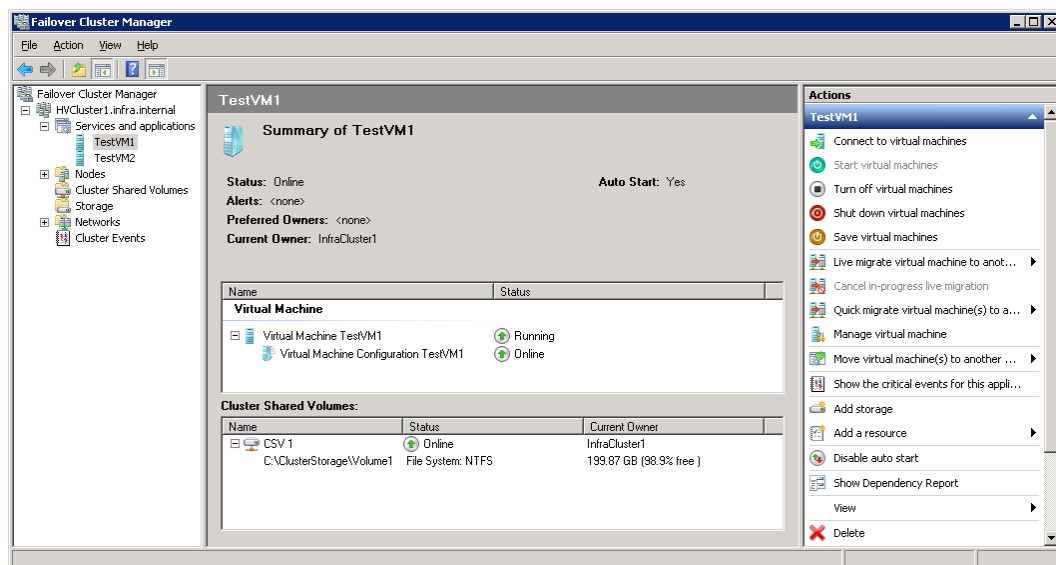
Yes! It is time for some real fun.

You will use Failover Cluster Managers to create a couple of virtual machines. No, I didn't Hyper-V Manager. There are a few reasons for that:

- If you use Hyper-V Manager then you'll have to create Virtual Machine resources in Failover Cluster Manager afterwards.
- Anytime you reconfigure a VM in Hyper-V Manager you have to update the cluster resource configuration in Failover Cluster Manager.

Failover Cluster Manager reveals the Hyper-V VM management actions and takes care of the resource update action for you. You can do VM management under Services And Applications.

Here you can see a pair of VMs have been created. One is on Host1 and the other is on Host2. To test, you would install operating systems (Linux will require static MAC address for Quick or Live Migration) and the latest integration components. Live Migrate the VMs between hosts, verifying network connectivity and VM functionality. Check that a single CSV can have VMs on different hosts. Ensure that you can move VMs back and forth.



In Cluster Shared Volumes check that you can move the CSV coordinator role between hosts (right-click the CSV in the centre pane, select Move This Shared Volume To Another Node, and select another host in the cluster).

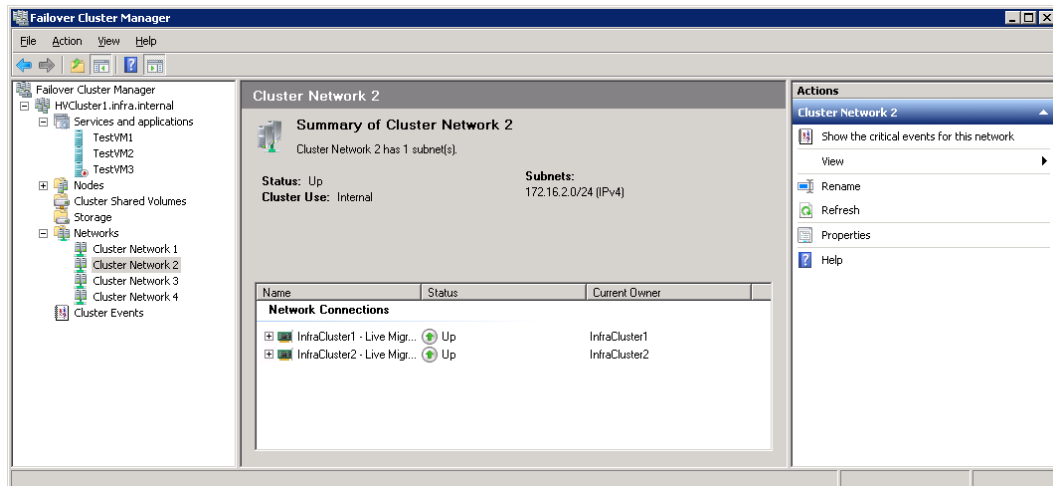
For extreme tests, you can power off a host and verify that VMs restart on the other host.

If all is well, you now have a fully functioning Hyper-V cluster, with highly available virtual machines, using the Microsoft iSCSI target for shared storage.

Specify the Cluster's Live Migration Network

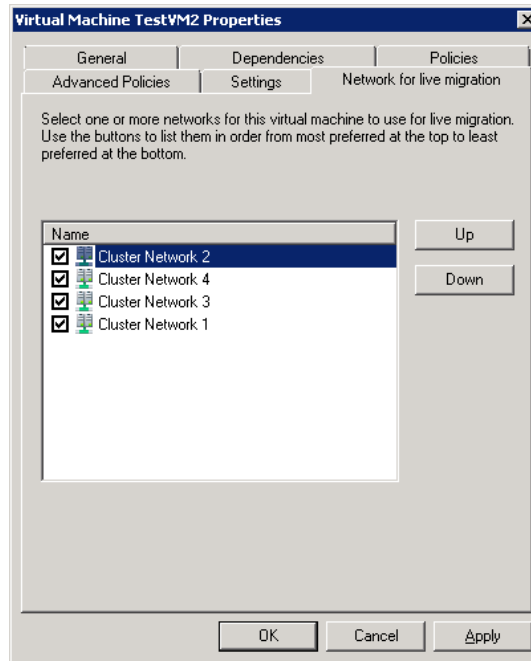
This one is a bit quirky. Earlier we configured the CSV network using Failover Clustering PowerShell cmdlets. Now we're going to configure the cluster's Live Migration network by editing the properties of a single VM. That sounds a bit weird, but that's how it's done!

First thing is to find out which of the cluster's networks contains the Live Migration NICs. You can find that out by browsing through your Networks in Failover Cluster Manager.

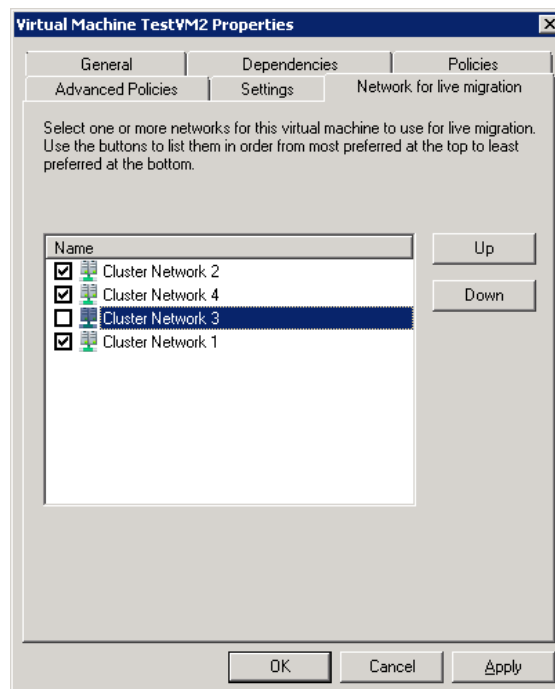


Armed with that information, in Failover Cluster Manager, navigate to one of the virtual machine resources under Services And Applications. You have seen an example in the previous section, "Build Some Highly Available Virtual Machines", of a virtual machine resource. Notice how the Virtual Machine and Cluster Shared Volume are present in the centre pane? We're going to work with the Virtual Machine. You should right-click on it, in the centre pane, select Properties, and navigate to the Network For Live Migration tab.

By default, Failover Clustering could use just any old NIC for Live Migration. That isn't good if you're trying to optimise the performance of the cluster. You can see in the next screen shot that all NICs are enabled for Live Migration. You can also see in what order Live Migration will try to use cluster networks. In this case it is Cluster Network 2 first (luckily our Live Migration network), then Cluster Network 4 (Parent), and so on, until it can't find a functioning network to use.



We can use the Up and Down buttons to reorder the NICs. That's not a problem here. I am happy with the Live Migration network (luckily) being the first selected. I'm OK with Parent being second – I don't want CSV to be impacted by Live Migration if the Live Migration Network is impacted. What I am not happy with is Cluster Network 3. Failover Clustering included one of my iSCSI networks. I can deselect that and exclude it from ever being used by Live Migration.



You've now taken complete control over the networking of your Hyper-V cluster.

Additional Storage Operations

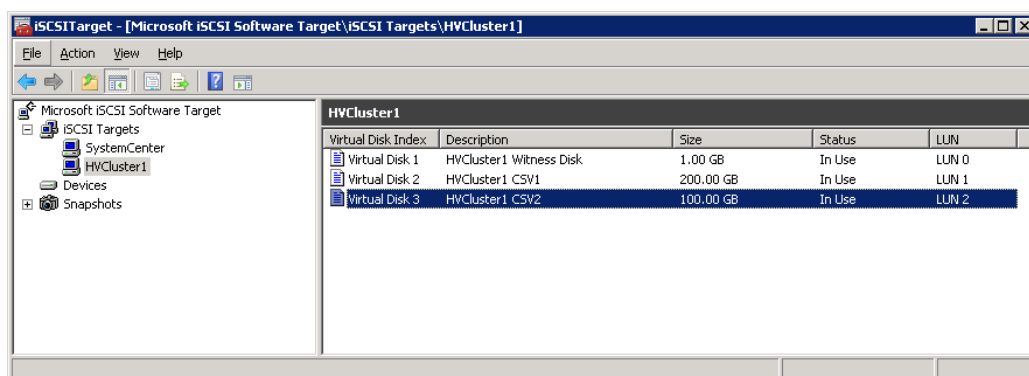
This section will detail some common storage operations you might need to do in a Hyper-V cluster.

Adding a CSV

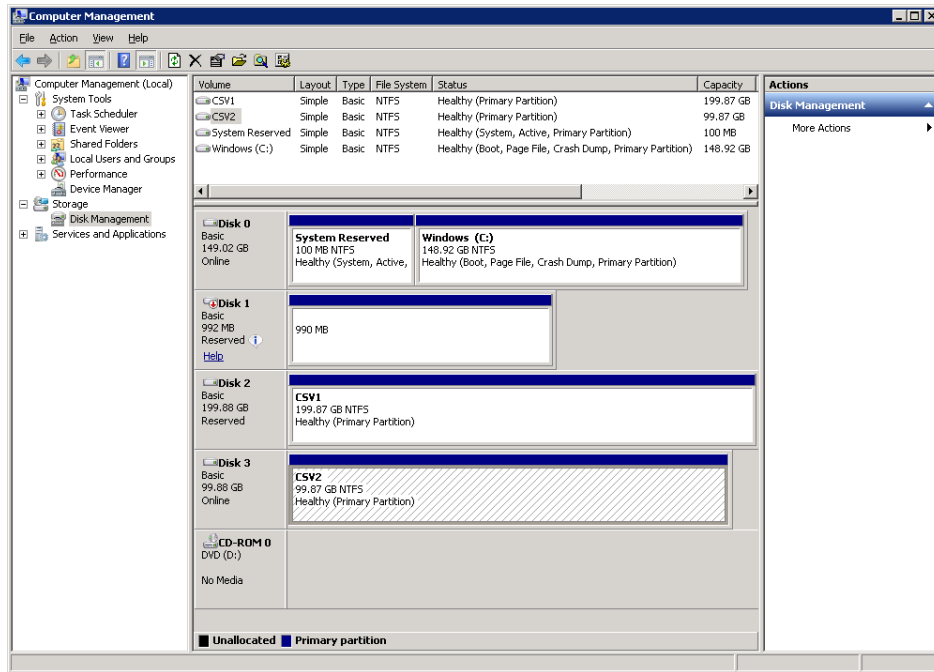
A production Hyper-V cluster will usually have more than one CSV. Some Hyper-V engineers use a rule of thumb of having one CSV per host in the cluster. Others use more scientific calculations based on disk performance, backup operations, and other such factors.

You can add more CSVs to an existing cluster. The steps will be as follows:

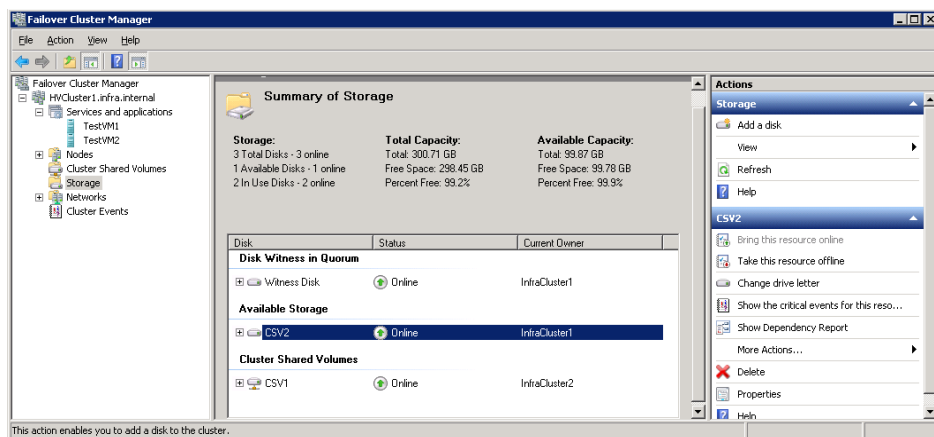
1. Log into the storage server.
2. Use the iSCSI Target administration console to add an additional device to the cluster target. This means that you are reusing the target and the identification methods, allowing all hosts in the cluster to have instant visibility of the new volume.



3. Log into Host1.
4. Using the iSCSI Initiator, click Auto Configure in Volumes and Devices to add the new device.
5. Log into Host2.
6. Using the iSCSI Initiator, click Auto Configure in Volumes and Devices to add the new device.
7. Use Disk Management on one of the hosts to bring online, initialize (GPT), format, and label the new volume.



- Use Storage in the Failover Clustering Manager to add the new storage device and rename it.



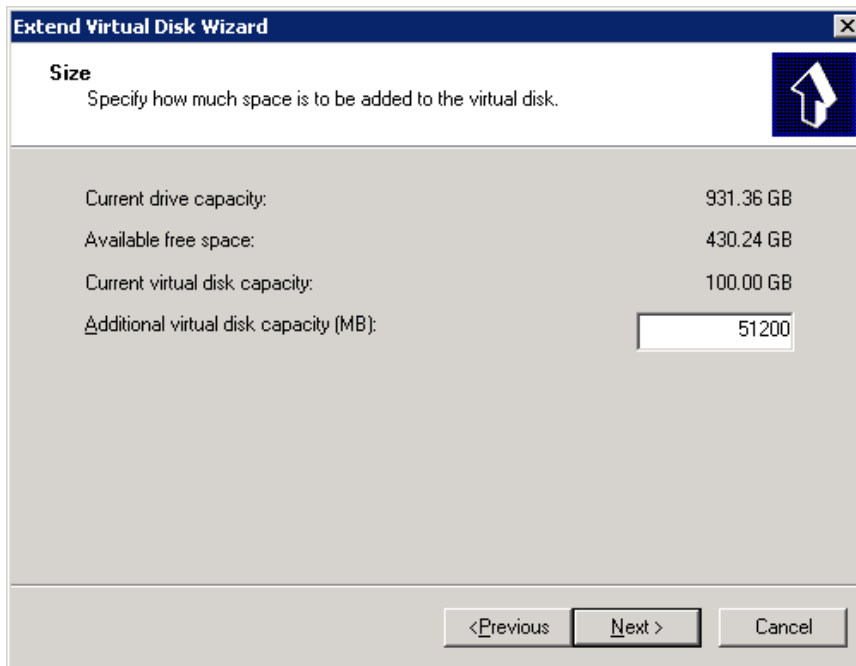
- Add the new CSV in Cluster Shared Volumes.

The newly provisioned storage is now available on your cluster for VM placement.

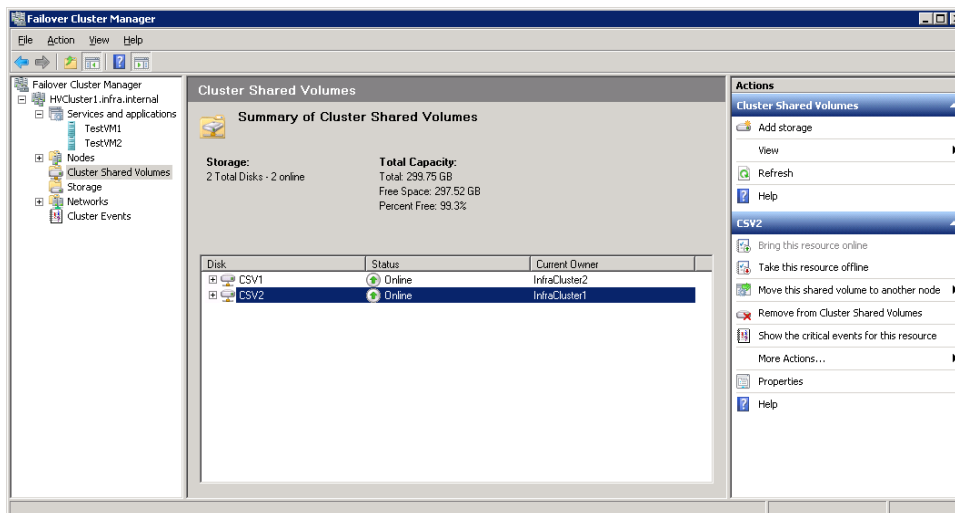
Extending a CSV

There will come a time when you want to add some space to an existing CSV. You can do this without any down time.

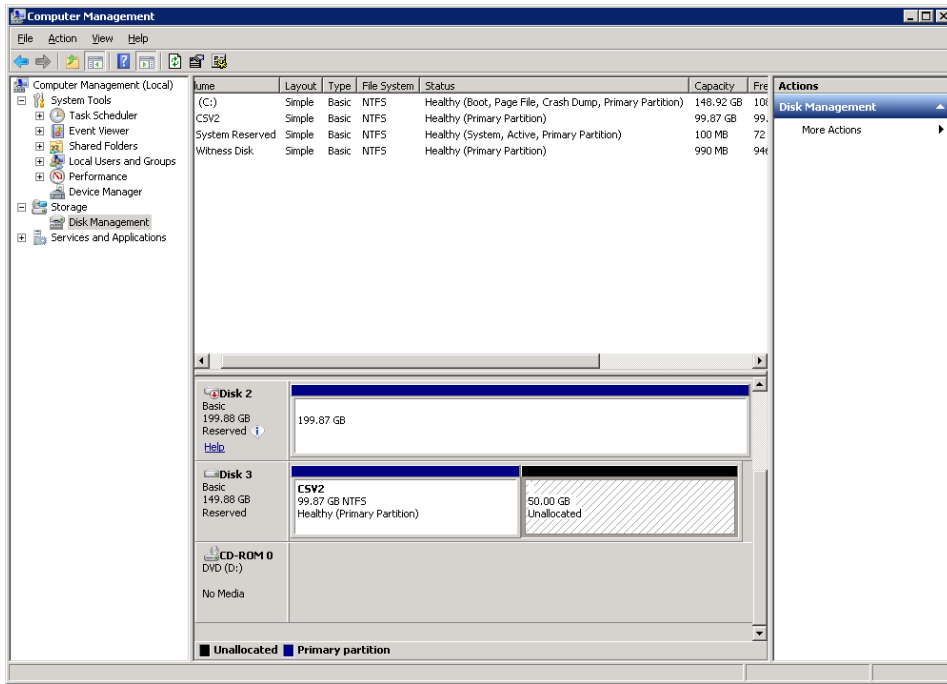
- Log into the storage server and browse into Devices in the iSCSI target administration console.
- Find the required device, right-click it, and select Extend Virtual Disk.
- You will enter the required additional capacity in the Extend Virtual Disk Wizard. You do not enter the required new size of the device. And remember the figure is in MB, not GB. In this example we are adding 50GB to a 100 GB device to result in a 150 GB device.



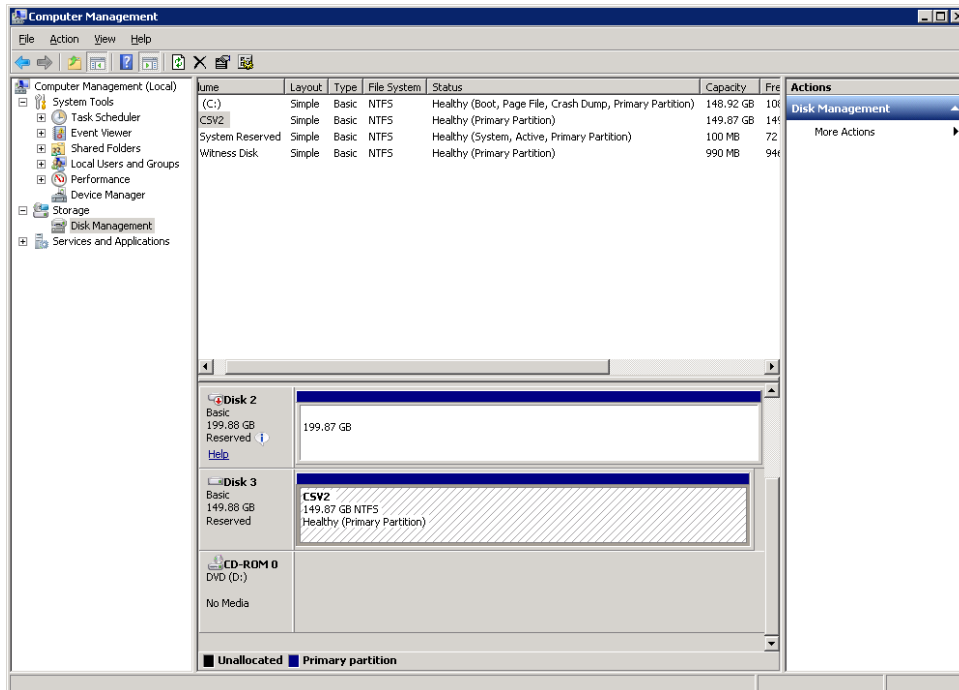
4. Use Failover Cluster Manager to identify the CSV coordinator (current owner). That will be the machine that shows the CSV volume as being online.



5. Find the volume in Disk Management for the CSV. Select the file system and extend it.



- The CSV is now extended to include the additional 50 GB of disk space from the storage server.



Summary

You've seen how you can build a small production environment or a test/training lab using the free Microsoft iSCSI Software Target. Using this, you don't need to splash out on third party software solutions or even purchase an expensive SAN solution. It is a very nice way to build a Hyper-V cluster on shoestring budget.

You get what you pay for in the storage world. There are of course some downsides. Unfortunately, the Microsoft iSCSI Software Target does not support iSCSI initiators with MPIO on clustered hosts. That means we cannot have network fault tolerance in the SAN. Some purists will not like that at all. We also don't get fancy features like block level replication between different sites, like you can get some software solutions and hardware appliances.

But let's face it, not every production environment needs those features. There are some who operate on a very low level, and without a cheap storage solution, they would otherwise be running non-clustered Hyper-V hosts. The Microsoft iSCSI Software Target, even without MPIO, is something that will probably give them with higher levels of uptime. And as for those of us who regularly need a demo/training lab, the Microsoft iSCSI Software Target gives us a "SAN" solution we can quickly deploy to even a PC.

Credits

1. Hans Vredevoort, MVP: for contacting the storage folks in Microsoft to find out that the Microsoft iSCSI Software Target v3.3 does actually support MPIO from initiators that are members of a cluster.

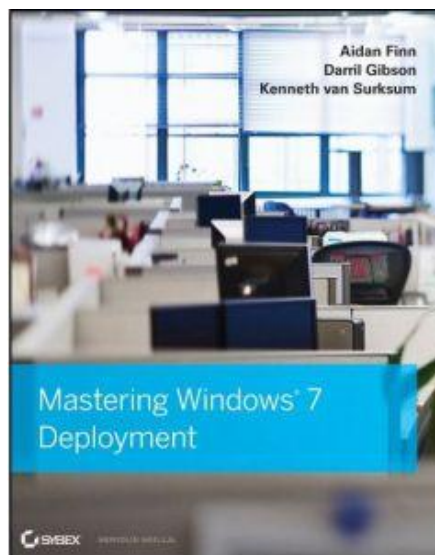
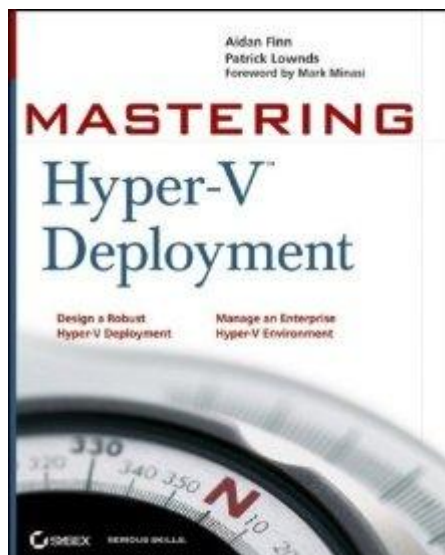
About The Author

Aidan Finn, MVP, has been working in IT since 1996, focusing on server/desktop management and virtualisation, including Hyper-V and System Center. He has worked as an administrator, contractor, and consultant with the likes of Amdahl/DMR, Barclays, Aer Lingus, and DVG.

Aidan blogs on <http://www.aidanfinn.com> and is one of the people behind the [Windows User Group](#) in Ireland. Aidan was a Microsoft [Most Valuable Professional](#) (MVP) on System Center Configuration Manager in 2008 and was switched the Virtual Machine expertise in 2009. Aidan has worked closely with Microsoft, including online interviews, launch events, and road shows.

When Aidan isn't at work, or at home in Ireland, he's out and about with camera in hand trying to be a [photographer](#). Aidan is the lead author of [Mastering Hyper-V Deployment](#) (Sybex, 2010). He is one of the contributing authors of [Mastering Windows Server 2008 R2](#) (Sybex, 2009) and of [Mastering Windows 7 Deployment](#) (Sybex, 2011) .

Aidan is on Twitter as [@joe_elway](#).



This could be where your company promotes its products

Contact me via my blog for more information.