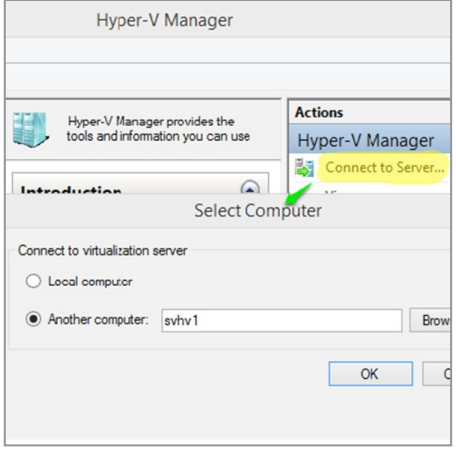
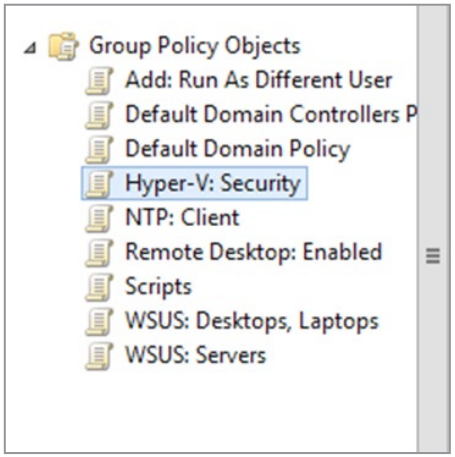




EXPLORED: 7 KEY AREAS OF HYPER-V

A detailed guide to help improve the core areas of your Hyper-V environment.

Brought to you by **Altaro Software**, developers of Altaro VM Backup

Compiled and written by **Eric Siron**

ALTARO VMBACKUP

Table of contents

Introduction

Explored: 7 Key Areas of Hyper-V

A detailed guide to help improve the core areas of your Hyper-V environment.

Hyper-V, Microsoft's computer virtualization technology, has matured into an enterprise-ready platform. Unfortunately, newcomers often find themselves stranded in a sea of unfamiliar terms and concepts. The good news is that their questions are common. This eBook gathers together information on some of the most common stumbling blocks to help you chart a clear path to a successful deployment.

This eBook is written for Hyper-V Server 2012 and 2012 R2.

Most material in this work has previously appeared on http://www.altaro.com/hyper-v/. It has been revised and expanded for this eBook. Visit our site for more great content, including free scripts to help you manage your Hyper-V environment.

CHAPTER 1:
Seven Keys to Hyper-V Security

For most small institutions, securing Hyper-V is often just a matter of just letting domain admins control the hypervisor. If that's not enough, there are a number of ways you can harden your Hyper-V deployment beyond the basics. If you are interested at all in taking security beyond the defaults, you'll want to plan your approach before you even begin putting your systems together.

Manage Access to Virtual Machine Functions

In the past, AzMan (Authorization Manager) was the tool of choice for managing specific virtual machine functions (Shut Down, etc.). AzMan was deprecated in 2012 and no longer works for Hyper-V Server 2012 R2. The MMC console and the XML file for Hyper-V are still there, but they won't control Hyper-V Server 2012 R2. The replacement is System Center Virtual Manager (VMM), which installs its own WMI path and has its own control mechanisms. Unfortunately, there is no longer any free, built-in way to manage control of virtual machines like this.

The "new" method is called **simplified authorization**. This fancy-sounding term actually just means that there is a new **Hyper-V Administrators** group created on each computer with the Hyper-V role enabled. Members of this group can control most anything related to Hyper-V (storage locations outside the default can still be an issue) but otherwise have no special powers on the Hyper-V host.

For most organizations, this is likely to be of limited use. In small organizations, it's normal that all administrators are full administrators; there's not a huge amount of distinction between who can control what. In

larger organizations, the Hyper-V administrator is probably responsible for the management operating system as well and is probably in the local administrators group anyway. If you're mixing roles and applications on your Hyper-V host, then this might come in handy. However, you're not supposed to be running anything else on the host other than Hyper-V-related software. If you do go against the recommendation and add other roles, then this group might be of some value.

Remember that you still control access to any given guest operating system just as you would if it were a physical machine. Users with local administrative access can still perform reboots, software installations, etc. They will not be able to turn them on, snapshot them, change virtual hardware, or anything of that nature without some level of administrative access on the host.

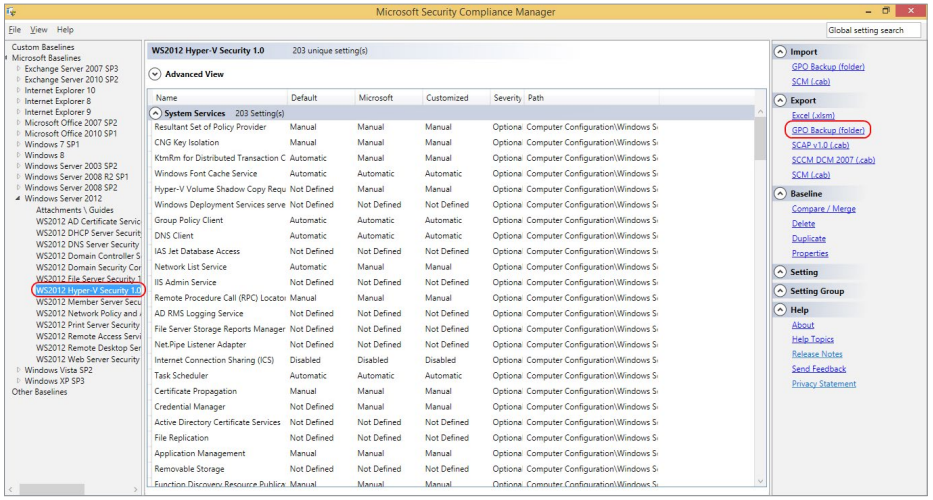
Group Policy

Group Policy is a great way to manage your systems, and is one of the greatest draws to using Active Directory domain membership. If you've decided not to join your Hyper-V hosts to your domain, you can still do most of this in local policy on one system, then export it, then import the exported policy on each unjoined Hyper-V host.
It is highly recommend that you be extremely judicious when using any setting under *Computer Configuration\Windows Settings\Security Settings\User Rights Assignment*. This isn't just a suggestion for Hyper-V; this is for your domain. This is because, in Group Policy, when there is a parent-child conflict, the OU closest to the object (child) takes precedence. In Group Policy security lists, entries are exclusive. So, once one of these security policies is enabled, only the accounts that appear on that list will be granted the related permissions. Any other accounts, such as those defined locally on the individual computers, will be excluded. This can prevent authorized accounts from logging on, or worse.

One common way to run afoul of this issue is to attempt to harden Hyper-V through Group Policy by manipulating these lists by following regular Windows procedures. The problem with this approach is that there is a local special account on systems with Hyper-V enabled called **Virtual Machines**. This account is not visible at the domain level, so it cannot be added to domain group policy without some wizardry. So, people following a hardening guide will go in and tinker with **Create symbolic links**, and suddenly find they have lost the ability to Live Migrate or build new VMs or do all sorts of things. They'll get Access Denied errors and spend a lot of time playing with ICACLS on their virtual machine storage folders, all to no avail. The lesson here is, if you absolutely must set these security policies at the domain level, make sure that you don't follow generic Windows hardening guides on a Hyper-V system.

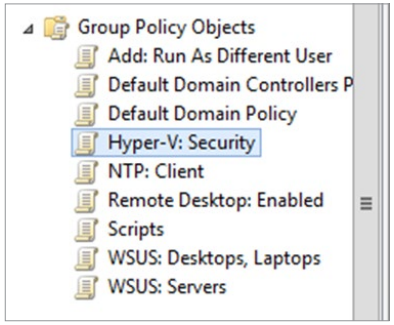
Fortunately for you, there is a Group Policy hardening tool with settings just for Hyper-V Server. Download Microsoft Security Compliance Manager. This tool is quite powerful and has many uses, so consider this just a basic introduction. Install the application on something other than your Hyper-V hosts. You could use your management workstation, but it does want to use a local SQL database. If you're uncomfortable with having that on your desktop/ laptop, find a more suitable location.

Once you've got it installed, expand *Windows Server 2012* on the left (it hasn't yet been updated to 2012 R2, but settings from the earlier version are fine). Underneath that, click on *WS2012 Hyper-V Security 1.0*. You'll be presented with a list of all the settings Microsoft thinks you should use to harden Hyper-V. These apply equally well whether you are running Hyper-V Server or Windows Server with Hyper-V as a role. You could pick and choose what you like, or you can use the export features at the right to save a GPO backup which can then be imported using Group Policy Management Console or Import-GPO. If your Hyper-V hosts aren't domain-joined, the included LocalGPO tool can be used, although you'll need to research that on your own (in the help files) as instructions are not included here. This is shown in the following screenshot.



Importing into GPMC is pretty straightforward, but in order for it to work as expected, your **Hyper-V hosts need to be in their own OU**. Do **not** allow them to inherit from another OU with hardening settings, especially one with the regular Windows Server hardening settings. If you've made any of your Hyper-V systems into a domain controller, sorry!

First, create a new Group Policy Object by right-clicking on the *Group Policy Objects* folder and clicking *New*. Give it a descriptive name. You should end up with something like the following screenshot:



Don't make any changes to it; they'll be lost. Then, right-click on it and choose **Import Settings**. This will take you to a wizard that's really easy to figure out. When prompted, point it to the folder you exported from SCM. Remember that it wants the folder that actually contains the **manifest.xml** file, not the GUID sub-folder. The only screen that might give you pause is the *Migrating References* screen. Just leave it on *Copying them identically from the source* and keep going. After a bit, you should receive a *Success* notification and you can close the wizard. Then just link the GPO to the OU for your Hyper-V hosts and continue on. If you want to hurry things up a bit, you can run *gpupdate* on the systems.

This section was specifically about applying group policy to your hosts. If you want to apply GPOs specifically to your virtual machines, Ben Armstrong wrote an article about using WMI to accomplish that task.

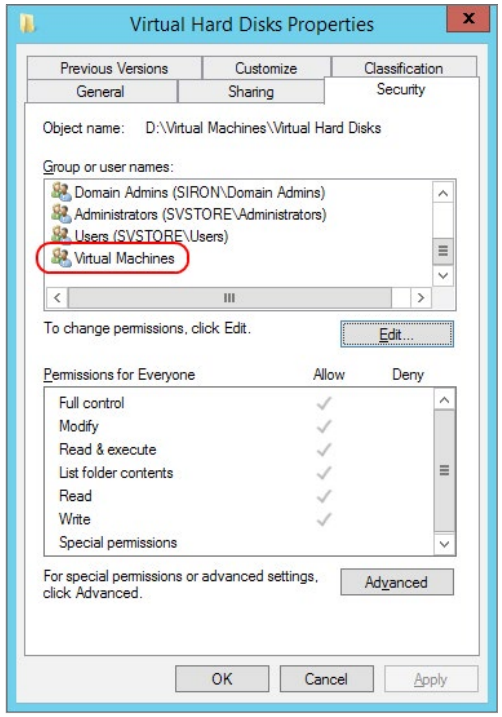
File, Folder, and Share Security

With Hyper-V storing almost everything in the traditional file and folder format, many administrators are led into a false sense of familiarity. So, some jump into hardening things at that level and run straight into unforeseen consequences. Sometimes, this shows up when using the tools. They'll try to perform some function in Hyper-V Manager and receive an "Access denied" message. Their first response is, "But I'm a domain administrator!" Remember that Hyper-V Server is an always-on server, not a user-mode application. You may be running the interface as a user, but it contacts the background server with your request, and the server carries them out. Some other servers, such as IIS, can use a security model that includes **impersonation**, where the server attempts to carry out requests by a user by pretending to be that user. Hyper-V Server doesn't operate on that security model. Instead, it carries out its functions in the context of the management operating system's Local System account. When it tries to talk to other computers, that means that it is trying to authenticate using the management operating system's computer account. In a domain, that computer account exists in Active Directory and can be used in inter-computer security operations. In a workgroup, you have to use something like **CredSSP**. For some uses, you also have the option to partially disable security checks by adding an entry to *WinRM TrustedHosts* (which means, "blindly, absolutely, and unquestioningly trust any computer that uses a name that appears in this list"). The possible uses for CredSSP and TrustedHosts are limited, which is why many things require domain membership.

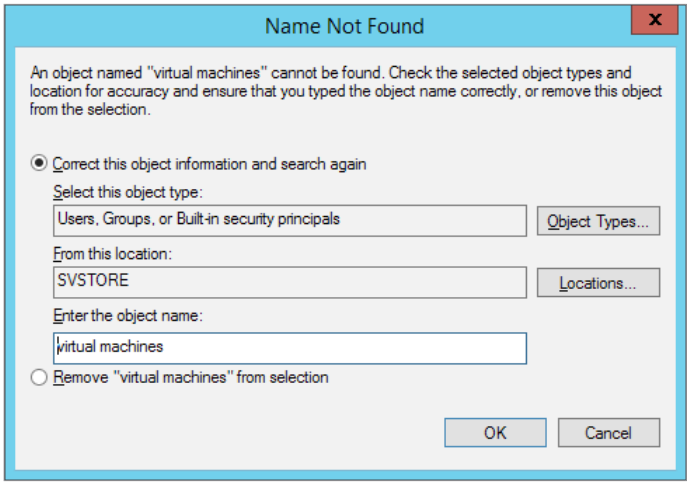
For a Hyper-V system that only operates locally, NTFS permissions are your concern. The big thing is to create a folder or use the default location and let Hyper-V manage the security right from the start. Don't come back later and

start turning screws. This also applies to storage on block-level remote storage, specifically iSCSI and Fibre LUNs. If you absolutely must change permissions, stay away from modifying inheritance patterns. You could easily wind up stripping required privileges away from an account you weren't even aware of. As a prime example, the virtual machine object needs control over its own files.

Remember that **Virtual Machines** account from the last section? Well look here:



That account was automatically added by Hyper-V. So, any manual tinkering with NTFS permissions and inheritance could potentially result in Hyper-V not being able to manage files. If this account is removed, it's not a simple matter of using the interface to restore. If you try manually adding the account, this is what you will see:



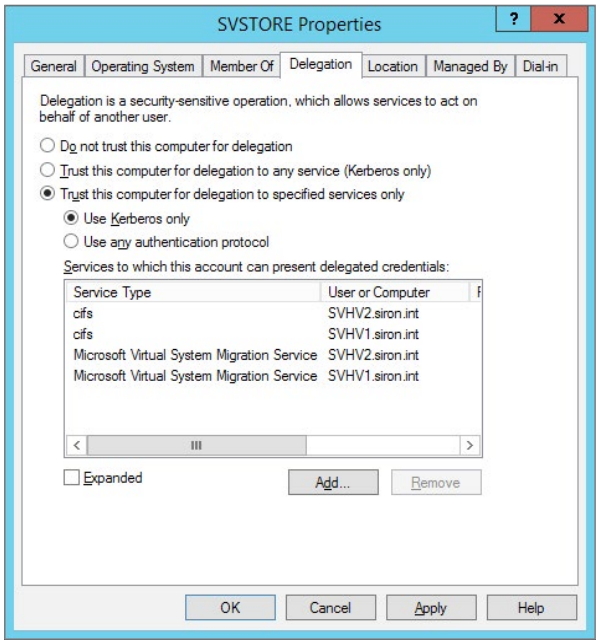
You can't browse for it, either. It can be re-added quickly, if a bit cryptically. Open PowerShell and enter the following, substituting your folder name as necessary:

```
$sid = Get-Acl -Path "C:\ClusterStorage\VMData1\
Virtual Hard Disks"

$sid.SetSecurityDescriptorSddlForm( ($sid.Sddl +
"(A;;FA;;;S-1-5-83-0)(A;OICIIO;0x101f01ff;;;S-1-5-83-0)") )

Set-Acl -Path "C:\ClusterStorage\VMData1\Virtual Hard
Disks" -AclObject $sid
```

Credit for above script to Nathan Storms (http://architectevangelist.wordpress.com/2011/02/20/hyper-v-virtual-machines-security-group-issue/). Be aware that the referenced blog post demonstrates resetting security on a root folder, which is not advised in a modern Windows environment.

For virtual machines hosted on SMB, there's a bit more to think about. First, domain membership is not optional. All involved physical machines must be domain members. Beyond that, you add share permissions and protocol access restrictions on top of the NTFS permissions. Of the two, share permissions are probably the easiest. The computer account(s) of the Hyper-V system(s) that will host virtual machines on the share need to have Full Control on this share. That's enough to get your SMB 3-based VM hosting working.

If you've got those VMs on an SMB 3 share, then you've opened the door to having VMs that can move between Hyper-V Servers. First, there's the traditional failover cluster. For that, you don't really have to do anything else (assuming the cluster already exists). If you want to migrate SMB-hosted VMs in Shared Nothing fashion, there might be a bit more work to do. If you will be using a remote machine to initiate a Shared Nothing Live Migration (you almost certainly will at some point), you need to enable delegation. What delegation means is that you can use your credentials from computer A to tell computer B to perform a function on computer C. Sometimes, computer C is actually computer A, but the basic issue is that your credentials are being used in a remote location. Because this can be a pretty severe security risk, it is advised that you not just open the floodgates on such delegation. Instead, use **constrained delegation**. This is configured on the Active Directory computer object for the machine to be controlled, and delegation is extended to the computers that might be used to control it. The following screenshot shows a computer object with two other machines granted delegation:



The CIFS entry controls SMB access. CIFS stands for **Common Internet File System**, which was the first iteration of the technology that eventually became SMB. The two terms are not interchangeable in most other contexts. The *Microsoft Virtual System Migration Service* should be self-explanatory. Be aware that this delegation is only necessary if you've set the migration model to use Kerberos delegation instead of CredSSP. You could also opt to set *Trust this computer for delegation to any service (Kerberos only)*, but doing so opens the gates a lot wider than is necessary.

The Network

Network access is your first line of defense against Bad Things coming from attackers. If you have the hardware, the expertise, and the budget, network security is best done in the networking hardware. If you don't, then you still have the Windows Firewall. This software is much maligned, which is sad because it's a whole lot better than nothing, and a lot less troublesome than many third-party software firewalls. It does sometimes get in the way (because that's what firewalls are for), but that doesn't mean you should just jump straight to turning it off. That's like saying, "My scarf made it harder to breathe, so I stopped wearing clothes."

The 2012 R2 series has been tuned to allow you to perform a great many management tasks without ever touching the firewall at all. If it's been your go-to practice to disable the firewall and you aren't confident in your hardware-level network security, you might consider revisiting this practice. If you do find an activity being blocked by the firewall, **selectively** open it.

The Windows Firewall does not interfere with guest traffic in any way, shape, or form. The adapter for the Hyper-V virtual switch is completely unbound from anything that the Windows Firewall has access to. Packets will pass through it without ever being inspected by the management operating system's firewall. Making changes to the firewall in Hyper-V to restrict or free traffic in the guests is a wasted effort.

There are, however, extensions to the Hyper-V switch available that do allow for packet processing at this level. These are beyond the scope of this eBook.

You can also get a measure of protection for the host through network isolation. Usually, this will be by employing VLANs and placing the Hyper-V host in its/their own or in a VLAN that's restricted to infrastructure systems. If you haven't got networking equipment that understands VLANs, you can still place certain systems in their own IP subnet(s). Without VLANs, they'll still

be reachable by broadcast and non-TCP/IP discovery methods, but anything is better than nothing. Of course, you'll also need a router any time you have disparate subnets. The big takeaway from this paragraph should be that the Hyper-V host does not need to be on the same IP subnet or VLAN as any of its guests or any other system.

The Guests

Hypervisor design is built around the idea of isolation. Just like modern operating systems isolate application processes so that one (theoretically) can't wreck another, hypervisors are designed to isolate guest operating systems. Historically, these have been called "partitions", although you don't see that terminology often in the x86/x64 world (you will, however, see it in some Hyper-V Event Logs). Every time you create a virtual machine, Hyper-V defines a partition for it. The management operating system also lives inside a partition. Like all technologies, this partitioning has its limitations. Regardless of design goals and processes, these guests are, in fact, accessing the same resource pool. There's always a danger that someone will figure out how to trample one partition from another. If you're using Intel chips, that's already happened.

What you need to do then, is secure your guests. You can treat them like isolated sandboxes when you're dealing with known quantities, like beta software from your (least) favorite vendor. You cannot treat them the same way if you're working with completely unknown software. For instance, back in the day, a company I worked for used a Windows 98 computer that only had a modem connection and a floppy drive for external access. We'd put little things on it to see if they were infected. It seems like a virtual machine would be a perfect corollary… except for the risk outlined in that article.

Securing a guest is mostly like securing a physical machine. Anyone who has console access might as well have full administrative powers, because you really just need an Internet search engine to figure out how to get into an operating system from its console. It needs its network connections protected, and any relevant antimalware software installed.

Antimalware

Installing antimalware on your hosts isn't as easy a decision as installing it on your guests. Your host should be pretty much isolated from user activity anyway; their traffic passes over the Hyper-V switch while your host's traffic moves over the management adapter. If you're using a fully converged design in which the management adapter is on the virtual switch, you still have a good degree of separation. There are currently no known compromises of the Hyper-V switch. The hard drive data

in the guests is also similarly isolated from the host. This means that a virus lurking in a VHDX is not a meaningful threat to the host. The same should go for guest memory. However, because of the break-out attack linked in #5, you can't just assume the natural isolation of the hypervisor will be sufficient.

If you're going to run antimalware, be aware that it is a threat to the proper operation of Hyper-V. Most of them seem to dislike the XML files that define your virtual machines. If antimalware strikes them, your virtual machines will just disappear. You need to make sure that you've got your exclusions configured properly. For Hyper-V alone, this wiki article lists the critical exclusions to make. Unfortunately, it's not quite the whole story for cluster-joined systems. This KB article is of some help. Pay very special attention to what it says about the shared disk model. For users of McAfee VirusScan Enterprise, this very blog post will likely be of great help. His exclusion list does exceeds what Microsoft recommends, but will eliminate problems with McAfee. You'll also notice that he talks about a "low risk process". McAfee, like many other vendors, doesn't necessarily not scan something just because you've set an exclusion. A lot of times, "exclusion" means something a little bit more like, "scan it, but don't tell me about any problems you find". Compare your vendor's method for actually excluding files and processes, and get these items added.

Patches and Hotfixes

I know, you're a little gunshy about patching after the serial system killers that came out of Microsoft in 2013. They're really going to have to work to earn all of our trust back. But, that doesn't mean you should just stop patching, either. Keep an eye out and keep as up-to-date as is sensible. The community at large usually knows within a couple of days. Just use the search engine of your choice for any given KB article number and you'll find out pretty quickly how deployment is going.

Thomas Maurer has made keeping up with patch easy by publishing a simple page that links to all the Hyper-V and Failover Cluster patches and hotfixes for the last three versions.

Summary

Security in Hyper-V is a many-faceted and complex thing. No one can give you a single magic bullet solution. This list is by no means all-inclusive; I didn't talk about common sense things like "don't write your password down or e-mail it to anyone". Do your research, do your due diligence, and keep your systems safe.

CHAPTER 2:
Hyper-V Manager – An Introduction

The very first graphical tool you'll use to manage your Hyper-V R2 infrastructure is Hyper-V Manager. It is simplistic enough to learn quickly but powerful enough to be the primary management application for small deployments. Even deployments that utilize System Center Virtual Machine Manager (SCVMM) will find uses for this tool.

How to Acquire Hyper-V Manager

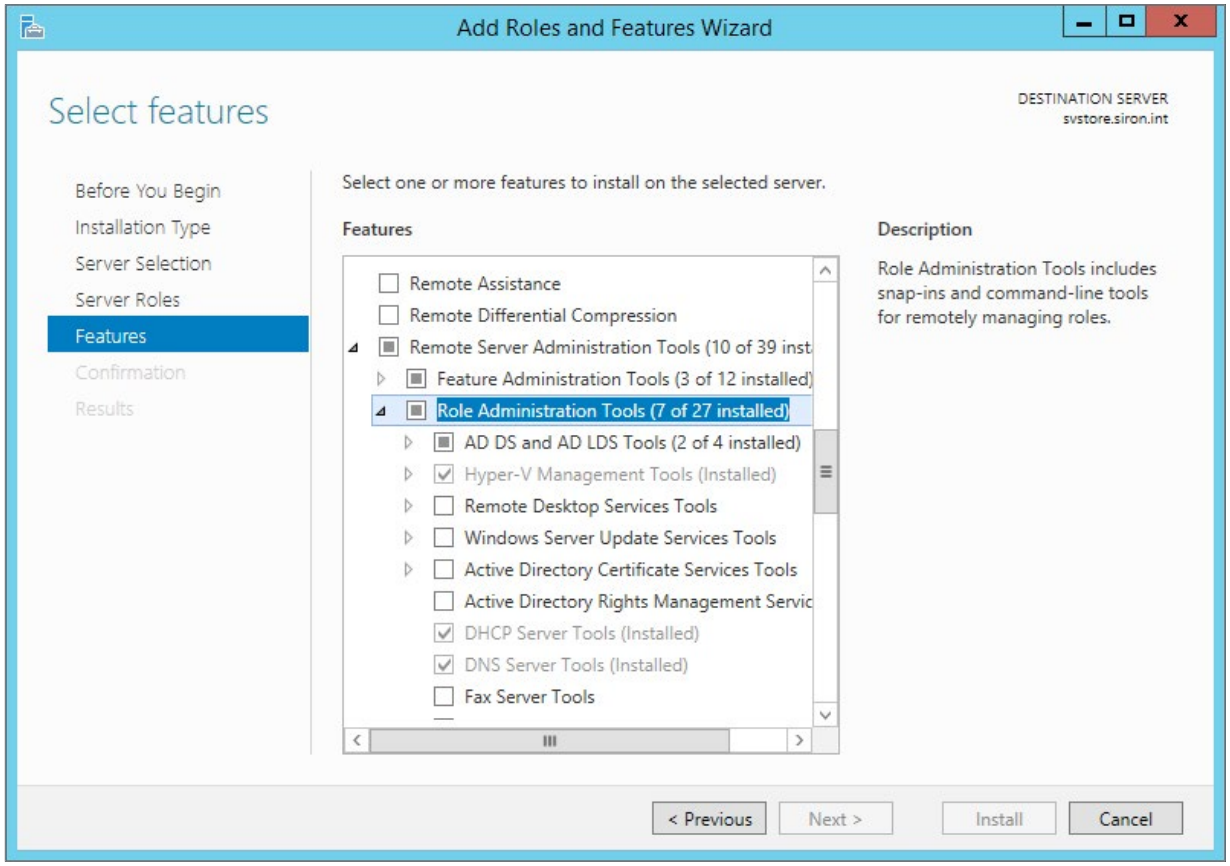
Hyper-V Manager is completely free… with a license of Windows. The second caveat is that you'll get the best results if you match your Windows version with the version of Hyper-V that you want to manage. Windows 8 (Professional or Enterprise editions) and Windows Server 2012 can manage Hyper-V Server 2012. Windows 8.1 and Windows Server 2012 R2 can manage Hyper-V Server 2012 R2. The lower versions have the ability to manage the later version of Hyper-V, but will be unable to access any of R2's new features. Windows 8.1 and Windows Server 2012 R2 can manage Hyper-V Server 2012, but none of the new features of R2 will work. Windows 7 and Windows Server 2008 R2, or any earlier versions of these operating systems, cannot manage the newer versions of Hyper-V Server using free GUI tools.

Enabling Hyper-V Manager

Hyper-V Manager is already built in to your operating system. All you need to do is make it available. The method you follow depends on the operating system you're using.

Windows Server 2012 or later

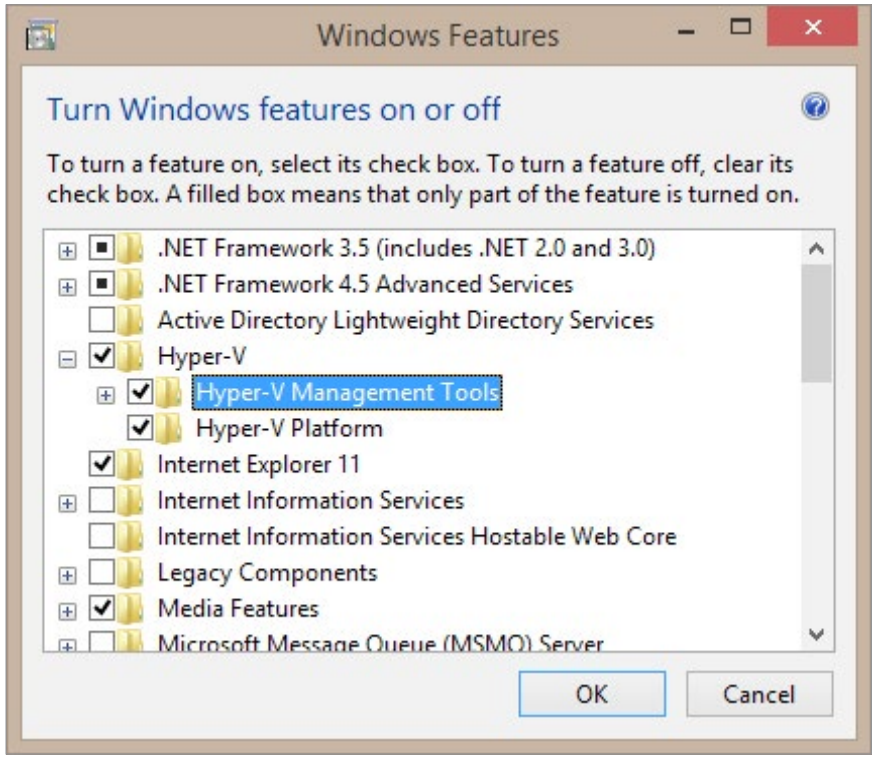
1. In Server Manager, go to *Add Roles and Features*. Alternatively, in Control Panel, go to *Turn Windows features on or off*.
2. In the wizard, skip to the *Features* page.
3. Expand *Remote Server Administration Tools*, then expand *Role Administration Tools*.
4. Check *Hyper-V Management Tools*. This is shown in the following screen shot:



5. Proceed through the rest of the wizard as normal.

Windows 8 Professional/Enterprise or later

1. Open Control Panel and click *Turn Windows features on or off*. If you're having difficulty finding Control Panel, open the Start screen and start typing "Windows features". The automated search should find it quickly (on Windows 8 you might have to click *Settings*).
2. Expand *Hyper-V. Check Hyper-V Management Tools* as shown (it is not necessary to install the Hyper-V Platform):

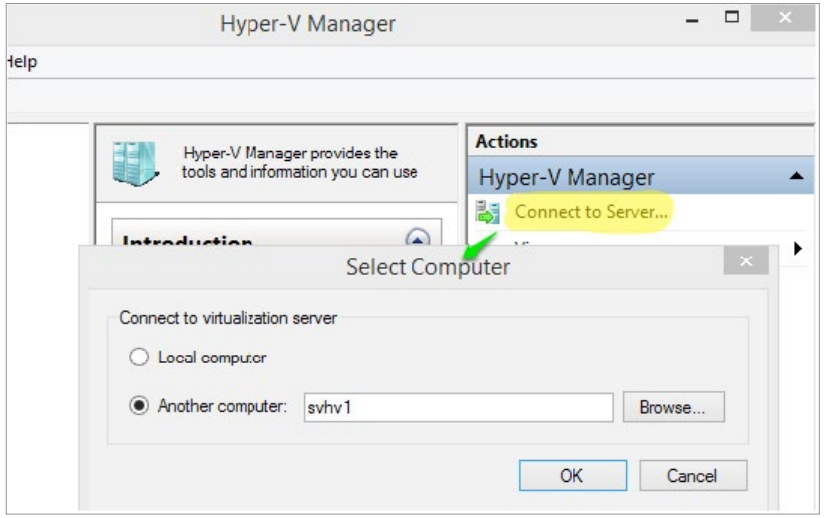


3. Click *OK*.

Regardless of your Windows version, once you've got the tool installed, you'll find it under the "Administrative Tools" menu selection on the Start Menu.

Interface Quick Tour

When you open the application for the first time, it provides some basic information in the center pane. Of course, you're using the tool to manage a Hyper-V server, so let's dive into that. As with most things Microsoft, there are multiple ways to begin. If you right-click the *Hyper-V Manager* item in the left pane, you'll get a context menu with *Connect to Server…* as an option. With that item highlighted, you can find the same item under the *Action* menu. Enter the name or IP of a Hyper-V host and click OK to connect to it.



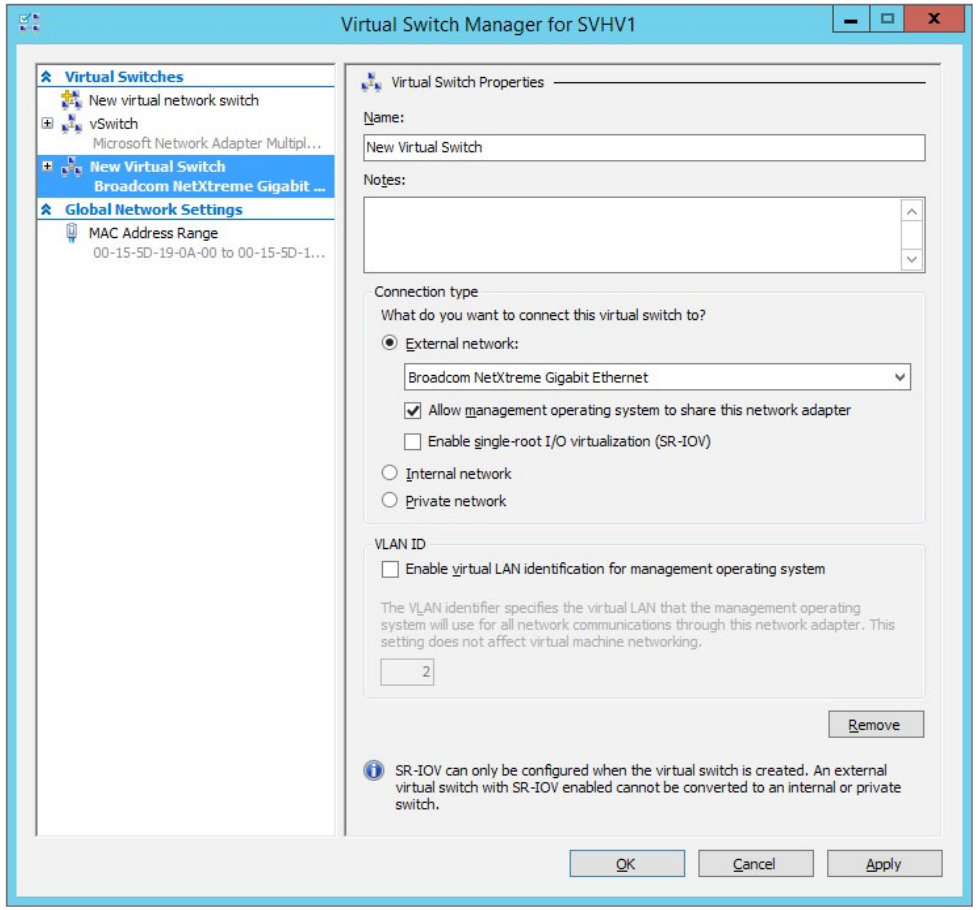
This will add the server to the console and select it so that, unless you highlight something else, context menus will operate on it.

Most components of the interface are simple and self-explanatory. If this is your first time in the program, click around and familiarize yourself with the layout and what the menu items do.

Virtual Switch Manager

In previous versions, this was one of the more important components of Hyper-V Manager. In 2012 and later, PowerShell has largely superseded this graphical tool, as it can't manage or display all the possible features of the virtual switch. However, this is a good starting place for those who are new to Hyper-V. Just be aware that some features, notably Quality of Service, are not configurable in Hyper-V Manager. A later section in this eBook will deal specifically with configuring the virtual switch in PowerShell. Also, if you are interested in using a team of network cards to host your virtual switch, you might want to read the next section before creating your switch.

The purpose of this dialog is to create one or more virtual switches. These virtual switches function much like physical layer-2 switches. When you create a virtual machine and give it a virtual network card, you'll be asked to connect it to a virtual network; that is much like plugging a physical card it into a physical switch. The primary difference is that you don't have to worry about port numbers.



One confusing thing about this dialog is the way the *OK* and *Apply* buttons work. You can configure multiple virtual networks without clicking either button; just make whatever changes you want to one and then either select another or click on "New virtual network" to begin working on a new one.

Starting at the top of the *Virtual Switch Properties*, the first thing you encounter is the *Name* field. This isn't all that important in a single-node deployment with only one NIC, but for any other deployment this matters. In a cluster, only identically named virtual networks can participate in any migration. You'll face the same restriction using Shared Nothing Live Migration. For instance, the above screenshot shows a switch named "vSwitch". For a high-availability virtual machine on this node to be LiveMigrated to another node, the destination node must contain a virtual network with the same name(s) as the one in use by the virtual machine. If there is only one node but multiple virtual switches, the names will guide how you load-balance the virtual machines.

The notes field is just that; use as you wish.

The connection type is very important. Only the external type can be used to connect virtual machines to a physical network and is the type you will most commonly use. It will bond to a physical network card in the host. Unfortunately, the drop-down box isn't exactly easy-to-use for determining which NIC is being assigned. On the Hyper-V host, run "IPCONFIG /ALL" at a command prompt and use the *Description* field for guidance. If you check the box for "Allow management operating system to share this network adapter", then Hyper-V will create a virtual network card for the management operating system and attach it to this virtual switch. With the more recent versions, it's recommended that you do this using PowerShell, as the GUI only allows you to create a single interface.

If you do select the option to share the physical NIC, you'll then be able to apply a VLAN Identifier for it. If you're plugging into a switch that uses VLANs, then the management operating system will be connected to the VLAN specified by this identifier. As indicated, it has no effect on any of the virtual machines.

Internal and private networks are not required for a successful deployment and are therefore beyond the scope of this discussion.

Virtual Machine Management

With a host added, you can now manage its virtual machines. If you haven't got any, you can use the *New Virtual Machine* wizard to create one. This process is very straightforward and will not be covered in-depth in this article. Just be aware that it sets a lot of default items, so you'll probably want to go back after creation and adjust some settings prior to using the created virtual machine. With a host highlighted on the left, the center pane serves as a heads-up display for all the host's virtual machines and their status. With a virtual machine highlighted, the right pane contains possible actions for both the host and the VM. These are mostly self-explanatory and you can quickly investigate the items. One that does deserve special mention is the *Reset* action. It is like hitting the Reset button on a physical computer and will **not** initiate a graceful shutdown. In that respect, this is also the difference between *Turn Off* and *Shut Down*; the latter is graceful, the former is not. Only use *Reset* and *Turn Off* when there are no graceful shutdown options.

The item that you'll probably spend most of your time in is the Settings dialog for a virtual machine. Again, most of this is straightforward so there's little benefit in exhaustive coverage. There are some things to note:

- Most settings cannot be changed while the virtual machine is turned on or in a saved state.
- BIOS tab: This is where you establish the boot order and whether or not NumLock is activated at boot. Note that regardless of what you set here, the state of the NumLock key isn't always perfectly translated when using Hyper-V's remote connection tools. This is especially notable with Windows Server 2003 virtual machines.
- Memory tab: You cannot configure dynamic memory when creating a virtual machine, so you'll need to access those settings here.
- Hard drives: Do not move the VM's boot drive to a SCSI controller or it will not start. Do not place a VHD containing a page file on the SCSI chain or it will never be used.
- Network adapters: Only use a Legacy adapter if you need network-boot capabilities or if the guest operating system does not support Integration Components.
- Integration services: For the most part, defaults here are fine, but there is a lot of information available on these components. They are beyond what a simple introductory text can cover, so you're advised to spend some time researching them.

Snapshots/Checkpoints

Snapshots and checkpoints are the same thing, but Microsoft is gradually phasing out the **Snapshot** term in favor of **Checkpoint** to avoid confusion with the process by the same named employed by Volume Shadow Copy Services. If you tell Hyper-V Manager to take a checkpoint of a virtual machine, it happens instantly without verification. You can use Hyper-V Manager to fully manage checkpoints. Be aware that this runs the risk of consuming large quantities of disk space, so research checkpoints if you are new to Hyper-V.

Differences between Hyper-V Manager and System Center Virtual Machine Manager

Organizations with more than a few virtual machines on a single host will probably want to utilize SCVMM. It adds a great deal of functionality beyond Hyper-V Manager, but it does not replace it. Here are the major differences where the two products have overlapping functionality:

- Hyper-V Manager adds no software to your Hyper-V deployment. SCVMM will install an agent on your host(s).
- Hyper-V Manager has no paid licensing requirements at all.
- Hyper-V Manager maintains almost real-time updates of what's happening on your host while SCVMM is delayed by several minutes. So, if a virtual machine is in a blue

screen reboot loop, you'll need Hyper-V Manager to successfully stop it.

- Hyper-V Manager processes the Hyper-V host's configuration by direct communication. SCVMM leverages a database to track configurations (and a great many other things), so it is a much "heavier" program. As such, it requires a Windows Server that can run its management component and a SQL Server Express instance.
- Hyper-V Manager's VM connection tool allows you to connect to a virtual machine even if it's off. SCVMM's does not.
- Hyper-V Manager's VM connection tool gives you a specific option to insert the integration services installation CD into a running VM so you can install it manually. SCVMM only gives you an option to install the integration services to a powered-off system, although it handles the entire process for you.
- SCVMM allows you to configure ranges of VLANs that virtual switches are allowed to trunk. Hyper-V Manager can't manage that at all. In an installation that has never had SCVMM, the virtual switches will trunk all VLANs.
- SCVMM cannot track the progress of a snapshot merge. Hyper-V Manager can.
- Deleting a VM in Hyper-V Manager does not delete its VHDs. Deleting it in SCVMM does.

Hyper-V Cluster Integration

Hyper-V Manager is aware of failover clusters, but it has no functionality to manage them (use Failover Cluster Manager or SCVMM for that). It cannot move virtual machines from one node to another, but if you use another tool to migrate a VM (Live or otherwise), it will indicate that it is moving. The most important thing about Hyper-V Manager's handling of clusters is that cannot create a virtual machine in High Availability mode. Failover Cluster Manager can convert existing virtual machines to High Availability mode and it can create virtual machines in High Availability mode (as can SCVMM).

Failover Cluster Manager

This section was specifically about Hyper-V Manager, but Failover Cluster Manager is a related tool that will come in handy if you're clustering Hyper-V. If you're using Windows Server 2012 or 2012 R2, the tool is built in and you just need to enable it. In the Add Roles or Features wizard, just look under the Features section of Remote Server Administration tools instead of the Roles section where you found Hyper-V Manager. If you're using Windows 8 or 8.1, you'll next need to download and install Remote Server Administration Tools from Microsoft's download site at http://www.microsoft.com/download. Make sure to get the download specific for your version, as they are different.

Windows 8:
http://www.microsoft.com/en-us/download/details.aspx?id=28972

Windows 8.1:
http://www.microsoft.com/en-us/download/details.aspx?id=39296

Summary

Spend some time becoming acquainted with Hyper-V Manager. While not as encompassing or as potent as PowerShell, it has the functionality to easily manage your Hyper-V environment for most day-to-day tasks. If you're new to Hyper-V, it is the easiest way to become acquainted with the hypervisor.

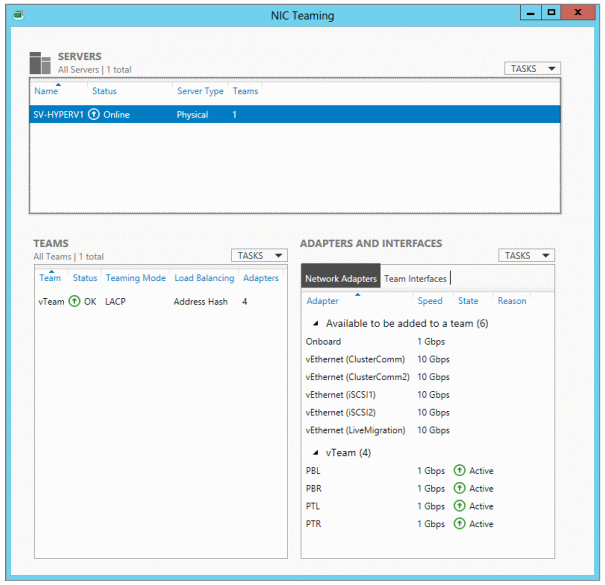
CHAPTER 3:
Set Up Native Network Teams for Hyper-V

Network adapter teaming is not a new concept by any means, but the introduction of native teaming in the Windows Server 2012 product marks the first time that Microsoft has openly supported it. Now, instead of using fickle teaming software from manufacturers that make your Hyper-V system unsupportable, you can create a network team right inside the management operating system and use it to carry your virtual machines' traffic. It is also the technology that made Converged Fabrics feasible for a Hyper-V host.

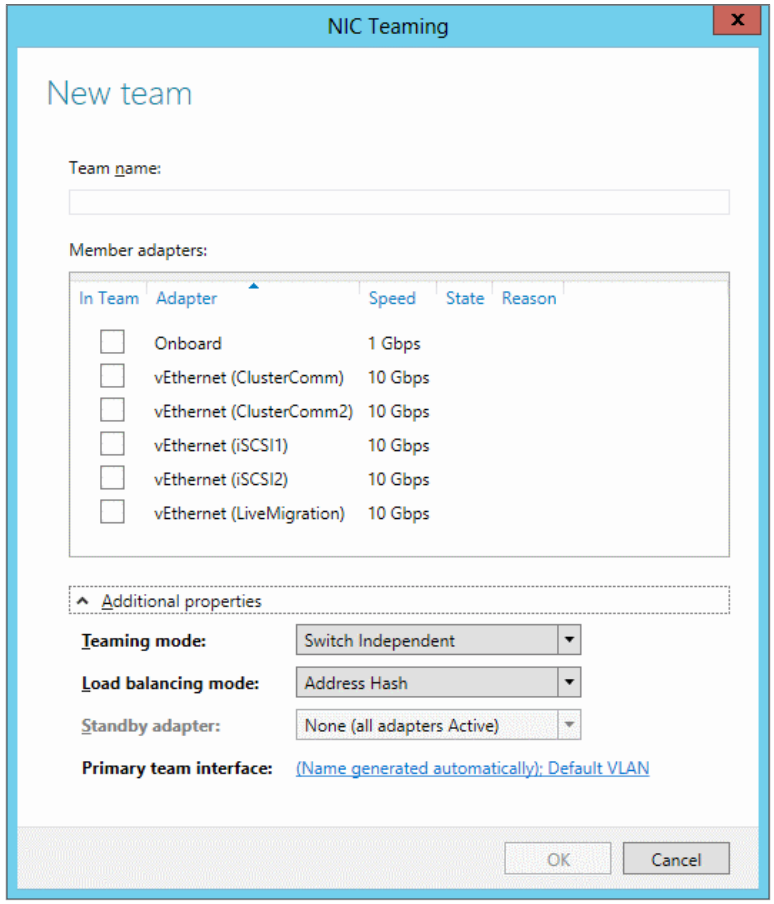
This section will focus on configuring the network team. Unlike the virtual switch, it's not as critical that you use PowerShell for a team, since nothing is permanent. However, PowerShell is faster and, in some ways, easier.

The GUI Way

In Server Manager, switch to the *Local Server* tab. There's a *Teaming* entry on the left. It will have a status of *Enabled* if you have a team and *Disabled* if you don't. Whichever it says, click it. You should get something akin to the following:



To create a new team, click on the *Tasks* drop-down box in the *Teams* section in the lower left and choose *New Team*. You'll be greeted with the following window:



This is pretty straightforward. Name it and pick the adapters you want to be part of it. Expand "Additional Properties" for the other options. Pick your teaming and load-balancing mode. When the team is created, an adapter will be created on it. If necessary, you can pick a VLAN for that adapter to be a member of. If you do so, it will only receive packets with an 802.1q tag for that VLAN. If no other adapters are created, then all other traffic on that team is discarded. If you leave the adapter at default, it gets all traffic. This is the setting you want for an adapter that will host a Hyper-V virtual switch.

The right side of the main screen deals with the team's adapters, both the physical adapters it sits on and the adapters that it hosts, called "Team Interfaces". You can use the Tasks drop-down on this section to create additional adapters on the team. Do not do this in any situation in which you will use the Hyper-V switch on the team. More on this later.

The PowerShell Way

All the magic happens with New-NetLbfoTeam. This cmdlt only has a few parameters. Let's go through them:

LoadBalancingAlgorithm

You can choose between TransportPorts, IPAddresses, MacAddresses, or HyperVPorts. The descriptions in the

Microsoft document are accurate but will be expanded on in the section after this.

Name

If you guessed, "Name of the team", give yourself a gold star.

TeamMembers

This is a comma-separated list of the names of the adapters that you want to put into the team. You can find adapter names with Get-NetAdapter.

TeamNicName

No, not a nickname, a NIC name, as in "network interface card". When the team is created, an adapter (team interface) is created on it. If you want, it can have its own name. Otherwise, you can skip this parameter and it will get the same name as the team. If you're going to create a Hyper-V switch on the team, this is the adapter name you'll pass to New-VMSwitch.

TeamingMode

Your choices are Dynamic (2012 R2 only), LACP, Static, and SwitchIndependent. The descriptions in the document are pretty good but will be expanded on later.

Related Cmdlets

There are a number of supporting cast members for your team. They are all documented in one convenient location. Of particular interest are Get-NetLbfoTeam, which is a quick way to see the status of a team, Get-NetLbfoTeamMember, which shows you the status of an individual member, and Add-NetLbfoTeamMember and Remove-NetLbfoTeamMember, who pretty much speak for themselves. You can use Set-NetLbfoTeam to modify the team and Set-NetLbfoTeamMember to set a member online or offline.

Notes on the Windows Team

- There is no functional difference between LACP and Static teaming. They both require that the switch you connect to be set to the same mode and you cannot connect to multiple physical switches in the same LACP or Static team. There are some switches that allow a stacked configuration in which multiple switches join to become the same logical switch, and these can usually accept a LACP or Static team that spans physical members.
- A static team trusts that the administrator knows what s/he's doing. If a member adapter sees a connection on the other end, it marks that link as up and participating in the team. If there's a misconfiguration, the static team won't know. You'll know because you'll have lots of

communications problems. A LACP team is (effectively) a static team with integrity checks. The ports on each side negotiate with each other and will stop participating if anything is wrong on the other end.

- Not all switches support LACP equally. You may need to change your load-balancing algorithm to get some to work, and some may never work.
- There is a reason for the apparent discrepancy in the load-balancing names in the GUI and in PowerShell, although it's up to you if you think it's a good one. The Address Hash mode in the GUI is essentially the same thing as the TransportPorts parameter of New-NetLbfoTeam. They both rely on a hash built from the source/destination ports and source/destination IPs. However, not all communications has all that information. Packets can be tracked by IP if they don't have port information, and then by MAC address if they don't have IPs. Higher levels will automatically fall back to lower levels. You can force the maximum level in PowerShell by using one of the other two modes. Once done, it will show in the drop-down box in the GUI.
- The load-balancing mode you set only applies to outbound traffic. Inbound traffic load-balancing is determined by the connected switch.
- In switch independent mode, all inbound traffic will cross the primary adapter. This is because a MAC address can only be registered on a single port. LACP and static teams are treated as singular ports and so MAC addresses are registered on the team, not the member physical ports. You cannot choose the primary adapter on the Hyper-V side.
- If using HyperVPorts load-balancing, virtual adapters are bound to specific physical adapters and do not move unless the port fails. This is the one exception to the rule of all MAC addresses being bound to a single port in switch independent mode.
- Standby adapters are only available in switch independent mode.
- You cannot use SR-IOV with teaming.
- HyperVTransports mode is pretty much useless if you aren't using virtual adapters on a Hyper-V switch. As in, 100% of your outbound traffic will use the primary adapter. Inbound traffic will still be balanced by the connected switch, but only if the virtual switch is in a switch dependent mode.
- Don't use multiple team interfaces at the same time as you use a Hyper-V virtual switch. The behavior is unpredictable at best.
- There is no QoS balancing for multiple team interfaces. Use the Hyper-V virtual switch with a single team interface.
- There can be only one team interface per VLAN.
- For all load-balancing methods except HyperVPorts,

vNICs will try to register a VMQ on every single adapter in the team. This can deplete your available queues in a hurry. Use HyperVPorts if VMQ is important and your physical adapters don't have enough queues to satisfy the evenly-spread load.

- Many guides insist that you should default to the HyperVPorts mode if you are using a Hyper-V switch. Unless you're using the switch independent mode, this is an unjustified oversimplification. The closer you are to a 1:1 ratio of vNICs to pNICs, the better the AddressHash mode will perform. There is, unfortunately, no magic number that will tell you which to use. Higher vNIC counts with higher traffic loads are usually better served by HyperVPorts mode.

Link Aggregation and Bandwidth

A very important thing to understand with network teaming is that even though the links are aggregated, bandwidth isn't simply multiplied by the number of members. The total available bandwidth is (roughly) equivalent to the total, but most individual communications will use only a single physical path. For example, a file copy from 192.168.25.30 to 192.168.25.40 will only use one adapter on the sending side and one adapter on the receiving side. However, if you place a virtual switch on a network team, it can direct the traffic for connected virtual machines across different physical paths, resulting in a much better utilization of hardware than would be realized using a single adapter.

Summary

This is just a brief introduction to teaming in the 2012 and R2 releases of Windows and Hyper-V Server. For a more detailed explanation of the options and how they all interact, Microsoft has published a more comprehensive manual.

Download Microsoft's teaming guide for Windows Server 2012: http://www.microsoft.com/en-us/download/details.aspx?id=30160

Download Microsoft's teaming guide for Windows Server 2012 R2: http://www.microsoft.com/en-us/download/details.aspx?id=40319

Teaming provides a powerful way to provide greater networking resources to virtual machines at little cost.

CHAPTER 4:
A Quick Guide to Hyper-V's Virtual Switch

As with the previous section on native network teaming, this section will be primarily devoted to working with the switch and won't talk much about its concepts. In short, the primary purpose of the Hyper-V virtual switch is to allow your virtual machines to communicate with other machines.

This section will only cover using PowerShell. The GUI method was described above in the introduction to Hyper-V Manager, and as explained there, it doesn't allow you to access all the options. If you're installing Hyper-V as a role through Server Manager, it doesn't either. PowerShell gives you far more control. More than a few of the people who create a switch with the GUI will wind up deleting their switch and making another one because the GUI didn't even let them see the options that they wanted.

It all starts with New-VMSwitch. Read the full documentation, if you like. The following adds some notes to Microsoft's rather Spartan explanations.

AllowManagementOS
Set this to zero or $false. All this does is create a virtual adapter for you, which you then have to go back, locate, rename and change all the settings on. You can make your own virtual adapter with Add-VMNetworkAdapter that begins life with most of the settings that you want.

ComputerName
"One or more hosts", as in, use this one line to create the same switch with the same settings on all the hosts in that new cluster of yours. Try that in the GUI. Leave this parameter off for the local system.

EnableIov
Another setting that's inaccessible in the Server Manager wizard and can never be changed later. Of course, if your hardware doesn't support it, setting this can end badly. Remember that if you create a virtual switch on top of a network team, SR-IOV is not available.

MinimumBandwidthMode
"Default Value: weight" is an error. The default is Absolute. This cannot be changed after the fact. So, if you used a GUI or trusted the documentation, you'll have to delete and recreate the switch to set QoS mode to Weight.

Name
The name for the virtual switch. Remember to choose a name you can type easily. Auto-complete doesn't work for switch names.

Notes
This is a free-form text field for anything you like.

SwitchType
This is only used if you want to make the switch internal or private. The External type is automatically chosen with the NetAdapterName parameter. As with the AllowManagementOS parameter, it doesn't mean exactly what it says. If you choose Private and then use Add-VMNetworkAdapter to make a vNIC in the management OS on it, it becomes Internal. If you use Remove-VMNetworkAdapter on all its vNICs in the management OS, it becomes Private. There is no converting to or from the External type.

NetAdapterInterfaceDescription
This parameter indicates the physical adapter you want to create the switch on by its description. This isn't incredibly useful because the description is probably a lot longer and more difficult to type than the name, but it does provide consistency with the way the GUI does things. As the documentation indicates, you can see the description in Get-NetAdapter. Don't use this with Private or Internal switch type because you can't and because if you could, it wouldn't be a Private or External switch type.

NetAdapterName
The name of the adapter to create a virtual switch on. You can get a list of the adapters with Get-NetAdapter. This can be a physical adapter or the virtual adapter that is created when you team adapters together. It cannot be a virtual adapter on a virtual switch. As with NetAdapterInterfaceDescription, don't use this with the Internal or Private switch types for exactly the same reasons. It's not recommended that you use this with the NetAdapterInterfaceDescription parameter either.

Confirm
This is used in case you want the system to ask you if you're sure that you want to create the switch. It could be used in cases where a script is created to be run by other admins.

What You Get

Either you get a brand new virtual switch, or you get an error. The most likely causes for errors is because you tried to create an internal/private switch with an adapter or because you picked an adapter it doesn't like.

If all went well, you've got the virtual switch you've always wanted. Try Get-VMSwitch to see your handiwork. Now, you can start creating virtual machines and attaching their virtual adapters to your switch. They'll connect by switch name. What you can also do is create virtual adapters in the management operating system. This is something you must use PowerShell for, because the GUI only lets you create a single virtual adapter for the management OS. The PowerShell way is Add-VMNetworkAdapter. Notice that the verb is "Add" instead of the typical "New".

The Fine Print

Some things that bear repeating and some other things you might be interested in:

- SR-IOV can only be enabled when the switch is created. It must then be set on the vNICs. Your physical adapter determines how many SR-IOV vNICs you can have. Check its documentation or ad sheet for Virtual Functions. One Virtual Function equals one SR-IOV vNIC.
- If you Live Migrate a VM with an SR-IOV adapter to a host that doesn't have any available Virtual Functions, the Live Migration fails. The reason is that it's like trying to replace the vNIC on the fly without interruption.
- You can use an adapter team for a virtual switch or you can enable SR-IOV. You can't do both. The reason is that the virtual machine talks directly to the Virtual Function, which is right on the hardware, which is difficult and maybe impossible to properly perform on a team. You can team in the guest if it's running Windows Server 2012 or later.
- You can't team vNICs in the management OS, but you can in guest OSs if they support it (2012+ for Windows guests).
- VMQ (virtual machine queues) have limited slots like SR-IOV, also determined by the adapter hardware. Check with your hardware vendor. Unlike SR-IOV, nothing will fail if there aren't any queues left and you don't need to set anything on the switch.
- You will likely encounter problems if VMQ is enabled but the hardware doesn't support it.

Summary

The virtual switch in Hyper-V is often a conceptual hurdle for those new to virtualization. Fortunately, working with it is fairly simple, even for those not familiar with PowerShell. You now know enough to get started. Feel free to continue your research using other resources, such as our blog.

CHAPTER 5:
Hyper-V Virtual CPUs

Capacity planning is a fundamental activity when building a virtual environment. CPUs are a particular sticking point, as there's not much guidance. Application vendors often make the problem worse with unclear requirements. They'll often insist that you dedicate an entire CPU core to the virtual machine running their application.

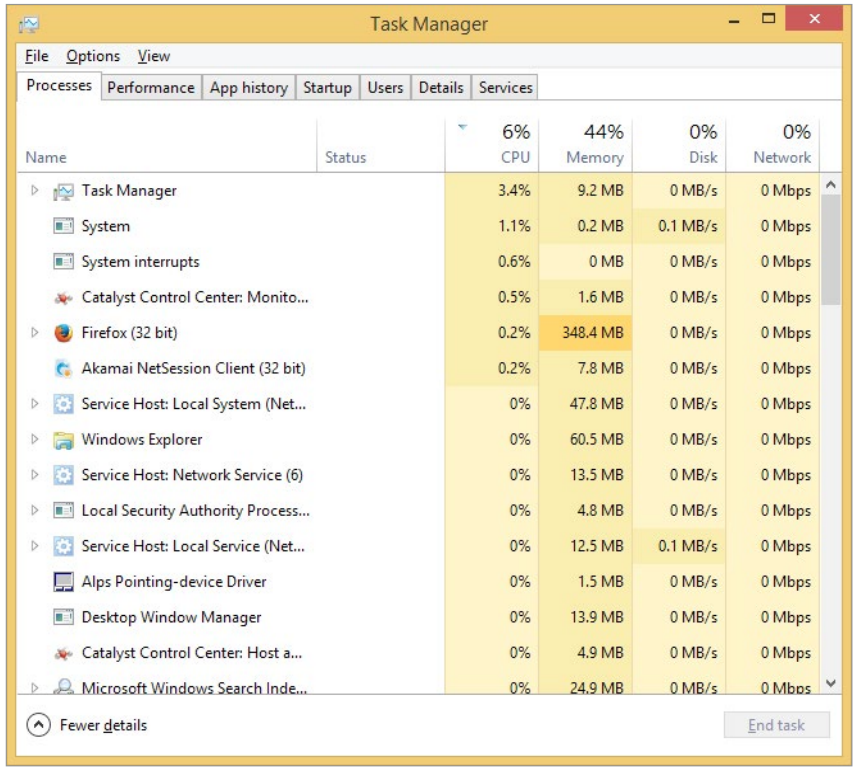
While there's really no good rule to tell you how to plan for CPU usage, understanding how virtual CPUs **(vCPUs)** work in Hyper-V will aid you in designing an environment that suits your needs.

Physical Processors are Never Assigned to Specific Virtual Machines

This is the most important note. Assigning 2 vCPUs to a system does not mean that Hyper-V plucks two cores out of the physical pool and permanently marries them to your virtual machine. You can't actually assign a physical core to a VM at all. So, does this mean that vendor request to dedicate a core just can't be met? Well, not exactly. More on that later.

Start by Understanding Operating System Processor Scheduling

Let's kick this off by looking at how CPUs are used in regular Windows. Here's a shot of a typical Task Manager screen:



Nothing fancy, right? Looks familiar, right?

Now, back when computers never, or almost never, came in multi-CPU multi-core boxes, we all knew that computers couldn't really multitask. They had one CPU and one core, so there was only one possible thread of execution. But aside from the fancy graphical updates, Task Manager then looked pretty much like Task Manager now. You had a long list of running processes, all of them with a metric indicating what percentage of the CPUs time they were using.

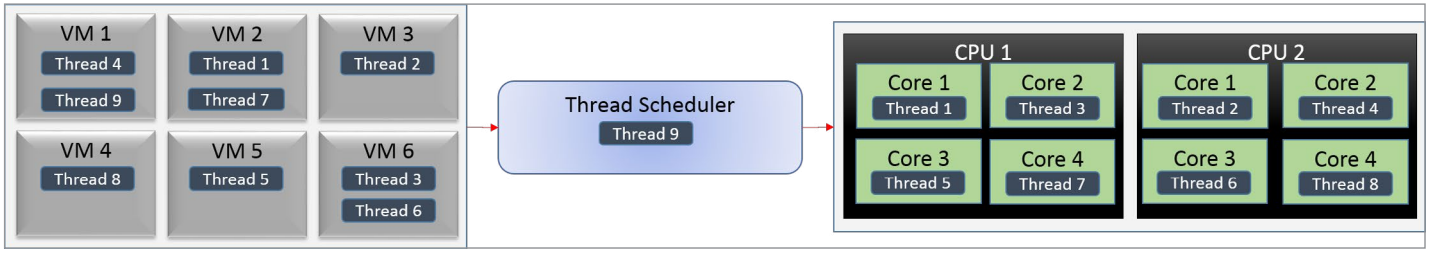
Then, as in now, each line item you see is a **process** (or, new in the recent Task Manager versions, a process group). A process might consist of one or many **threads**. A thread is nothing more than a sequence of CPU instructions (key word: sequence).

What happens is that (in Windows, this started in 95 and NT) the operating system would stop a running thread, preserve its state, and then start another thread. After a bit of time, it would repeat those operations for the next thread. Remember that this is **pre-emptive**, meaning that it is the operating system that decides when a new thread will run. The thread can beg for more, and you can set priorities that affect where a process goes in line, but the OS is in charge of thread scheduling.

The only difference today is that you have multiple cores and/or multiple CPUs in practically every system (as well as hyper-threading in Intel processors), so Windows can actually multi-task now.

Taking These Concepts to the Hypervisor

Because of its role as a thread manager, Windows can be called a "supervisor" (very old terminology that you really never see anymore): a system that manages processes that are made up of threads. Hyper-V is a hypervisor: a system that manages supervisors that manage processes that are made up of threads. Pretty easy to understand, right? Task Manager doesn't work the same way for Hyper-V, but the same thing is going on. There is a list of partitions, and inside those partitions are processes and threads. The thread scheduler works pretty much the same way. What follows is a basic visualization of thread scheduling:



Of course, there are always going to be a lot more than just nine threads going at any given time. They'll be queued up in the thread scheduler.

What about Processor Affinity?

You probably know that you can affinitize threads in Windows so that they always run on a particular core or set of cores. There's no way to do that in Hyper-V with vCPUs. Doing so would be of questionable value anyway; dedicating a thread to a core is not the same thing as dedicating a core to a thread, which is what many people really want to try to do. You can't prevent a core from running other threads in the Windows world.

How Does Thread Scheduling Work?

The simplest answer is that Hyper-V makes the decision at the hypervisor level, but it doesn't really let the guests have any input. Guest operating systems decide which of their threads they wish to operate. The above image is necessarily an oversimplification, as it's not simple first-in-first-out. NUMA plays a role, for instance. Really understanding this topic requires a fairly deep dive into some complex ideas, and that level of depth is not really necessary for most administrators.

The first thing that matters is that (affinity aside) you never know where any given thread is going to actually execute. A thread that was paused to yield CPU time to another thread may very well be assigned to another core when it is resumed. Did you ever wonder why an application consumes right at 50% of a dual core system and each core looks like it's running at 50% usage? That behavior indicates a single-threaded application. Each time it is scheduled, it consumes 100% of the core that it's on. The next time it's scheduled, it goes to the other core and consumes 100% there. When the performance is aggregated for Task Manager, that's an even 50% utilization for the app. Since the cores are handing the thread off at each scheduling event and are mostly idle while the other core is running that app, they amount to 50% utilization for the measured time period. If you could reduce the period of measurement to capture individual time slices, you'd actually see the cores spiking to 100% and dropping to 0% (or whatever the other threads are using) in an alternating pattern.

What we're really concerned with is the number of vCPUs assigned to a system and priority.

What Does the Number of vCPUs I Select Actually Mean?

You should first notice that you can't assign more vCPUs to a virtual machine than you have physical cores in your host.

```
PS C:\> Set-VM -Name svscvmm -ComputerName svhv2 -ProcessorCount 4
Set-VM : 'svscvmm' failed to modify device 'Processor'. (Virtual machine ID 8C009901-8A81-42AD-BFF0-BA0DB12CD857)
Cannot assign the specified number of processors for virtual machine 'svscvmm' is out of range. The range is 1 through
2. (Virtual machine ID 8C009901-8A81-42AD-BFF0-BA0DB12CD857)
A parameter that is not valid was passed to the operation.
At line:1 char:1
+ Set-VM -Name svscvmm -ComputerName svhv2 -ProcessorCount 4
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : InvalidArgument: (Microsoft.HyperV.PowerShell.VMTask:VMTask) [Set-VM], VirtualizationOpe
   rationFailedException
    + FullyQualifiedErrorId : InvalidParameter,Microsoft.HyperV.PowerShell.Commands.SetVM
```

So, a virtual machine's CPU count means the maximum number of threads that it is allowed to operate on physical cores at any given time. The virtual machine shown above can't have more than two vCPUs because the host only has two cores. Therefore, there is nowhere for a third thread to be scheduled. But, if the host had 24 cores and this VM was left at 2 vCPUs, then it would only ever send a maximum of two threads up to the hypervisor for scheduling. Other threads would be kept in the guest's thread scheduler (the supervisor), waiting their turn.

But Can't You Assign More Total vCPUs to all VMs than Physical Cores?

Absolutely. Not only can you, but you're almost definitely going to. It's no different than the fact that you might have 40+ processes "running" on a dual core laptop. It can't actually run more than two threads at a time, but it's always going to have far more than two threads scheduled. Windows has been doing this for a very long time now, and Windows is so good at it (usually) that most people don't even pause to consider just what's going on. Your VMs (supervisors) will bubble up threads to run and Hyper-V (hypervisor) will schedule them the way (mostly) that Windows has been scheduling them ever since it outgrew cooperative scheduling in Windows 3.x.

What's The Proper Ratio of vCPU to pCPU/Cores?

This is the question that's on everyone's mind. In the generic sense, this question has no answer.

In the past, the recommendation was 1:1. Some people still say that today. And you know, you can do it. It's wasteful, but you can do it. You could also run a desktop configuration on a quad 16 core server and never have any contention. But, you probably wouldn't see much performance difference. Why? Because almost all threads sit idle almost all the time. If something needs 0% CPU time, what does giving it its own core do? Nothing, that's what.

Later, the answer was upgraded to 8 vCPUs per 1 physical core. OK, sure, good.

Then it became 12.

And then the recommendations went away.

They went away because they really didn't make any sense. Somewhere along the way, they were born from aggregated observations and testing, but really, think about it. You know that mostly, operating threads will be evenly distributed

across whatever hardware is available. So then, the amount of physical CPUs needed doesn't depend on how many virtual CPUs there are. It's entirely dependent on what the operating threads need. And, even if you've got a bunch of heavy threads going, that doesn't mean their systems will die as they get pre-empted by other heavy threads. It really is going to depend on how many other heavy threads they wait for.

Here's the dirty little secret about CPUs: Every single time a thread runs, no matter what it is, it drives the CPU at 100% (power-throttling changes the clock speed, not workload saturation). The CPU is a binary device; it's either processing or it isn't. The 100% or 20% or 50% or whatever number you see in Task Manager is completely dependent on a time measurement. If you see it at 100%, it means that the CPU was completely active across the measured span of time. 20% means it was running a process 1/5th of the time and 4/5th of the time it was idle. What this means is that a single thread can't actually consume 100% of the CPU the way people think it can, because Windows/Hyper-V will pre-empt it when it's another thread's turn.

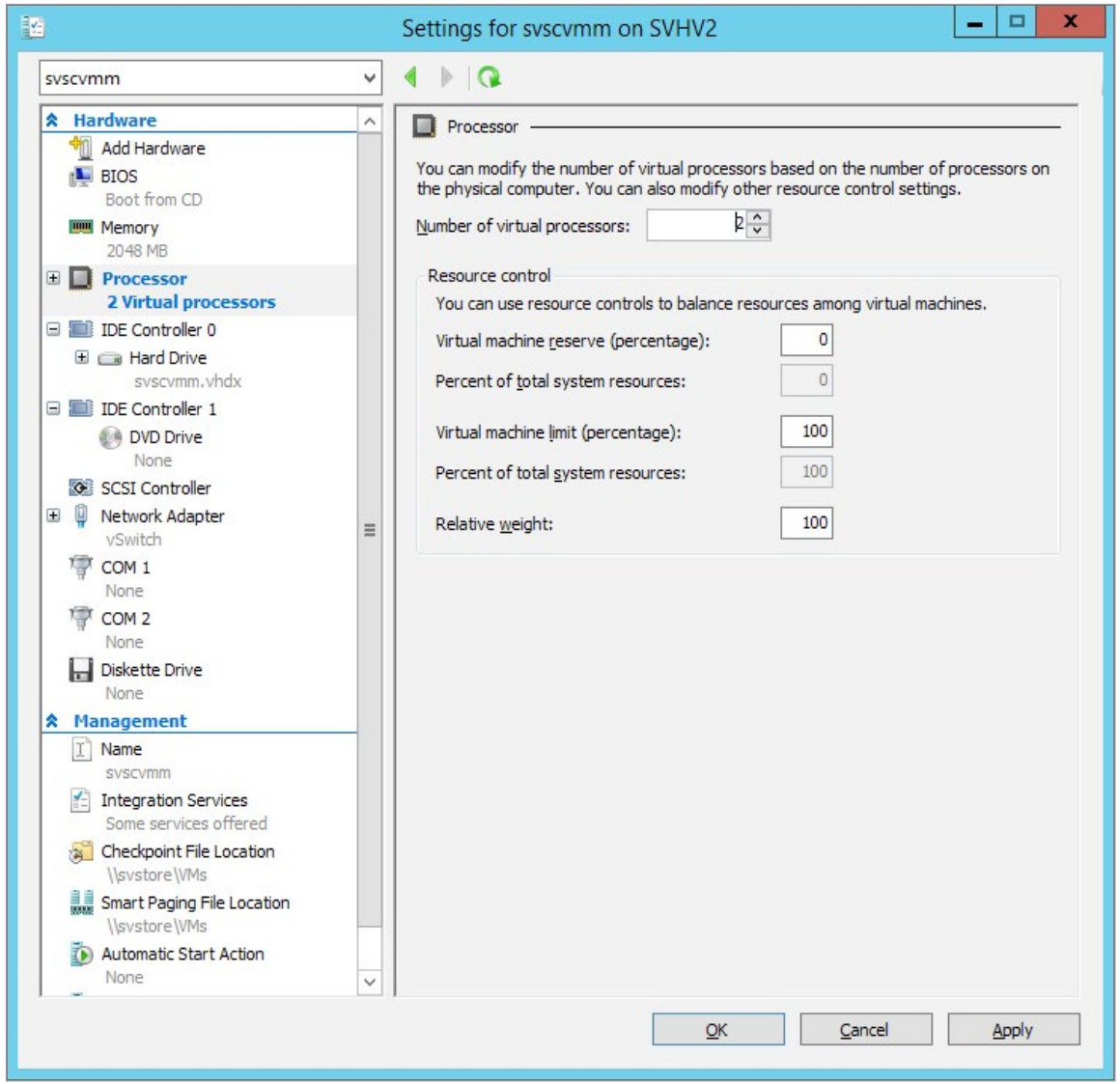
You can actually have multiple "100%" CPU threads running on the same system. The problem is that a normally responsive system expects some idle time, meaning that some threads will simply let their time slice go by, freeing it up so other threads get CPU access more quickly. When you have multiple threads always queuing for active CPU time, the overall system becomes less responsive because the other threads have to wait longer for their turns. Using additional cores will address this concern as it spreads the workload out.

What this means is, if you really want to know how many physical cores you need, then you need to know what your actual workload is going to be. If you don't know, then go with the 8:1 or 12:1, because you'll probably fine.

What about Reserve and Weighting (Priority)?

Don't tinker with CPU settings unless you really understand what's going on. Let the thread scheduler do its job. Just like setting CPU priorities on threads in Windows can get initiates into trouble in a hurry, fiddling with hypervisor vCPU settings can throw a wrench into the operations.

Let's look at the config screen:



The first group of boxes is the reserve. The first box represents the percentage that I want to set, and its actual meaning depends on how many vCPUs I've given the VM. In this case, we have a 2 vCPU system on a dual core host, so the two boxes will be the same. If the reserve is set to 10 percent, that's 10 percent of the total physical resources. If it's dropped down to 1 vCPU, then 10 percent reserve becomes 5 percent physical. The second box, which is grayed out, will be calculated for you as you adjust the first box.

The reserve is a hard minimum… sort of. If the total of all reserve settings of all virtual machines on a given host exceeds 100%, then at least one virtual machine isn't going to start. But, if a VM's reserve is 0%, then it doesn't count toward the 100% at all. But, if a VM with a 20% reserve is sitting idle, then other processes are allowed to use up to 100% of the available processor power… until such time as the VM with the reserve starts up.

Then, once the CPU capacity is available, the reserved VM will be able to dominate up to 20% of the total computing power. Because time slices are so short, it's effectively like it always has 20% available, but it does have to wait like everyone else.

So, that vendor that wants a dedicated CPU? If you really want to honor their wishes, this is how you do it. You enter whatever number in the top box that makes the second box the equivalent processor power of however many pCPUs/cores they think they need. If they want one whole CPU and you have a quad core host, then make the second box show 25%.

The next two boxes are the limit. Now that you understand the reserve, you can understand the limit. It's a resource cap. It keeps a greedy VM's hands out of the cookie jar. The final box is the weight. As indicated, this is relative. Every VM set to 100 (the default) has the same pull with the scheduler, but they're all beneath all the VMs that have 200, so on and so forth. If you're going to tinker, this is safer than fiddling with reserves because you can't ever prevent a VM from starting by changing relative weights. What the weight means is that when a bunch of VMs present threads to the hypervisor thread scheduler at once, the higher weighted VMs go first. That's it, that's all.

But What About Hyper-Threading?

Hyper-Threading is an Intel-specific technology that lets a single core process two separate instructions in parallel (called pipelines). Neat, right? One problem: the pipelines run in lockstep. If the instruction in pipeline one finishes before the thread in pipeline two, pipeline one sits and does nothing. But, that second pipeline shows up as another core. So the question is, do you count it? The official response is: No. Hyper-Threading should not be counted toward physical cores when considering hypervisor processing capabilities. In practice, you'll have more leeway. It's not quite as good as another actual core, but it's not useless either. Your mileage may vary.

Summary

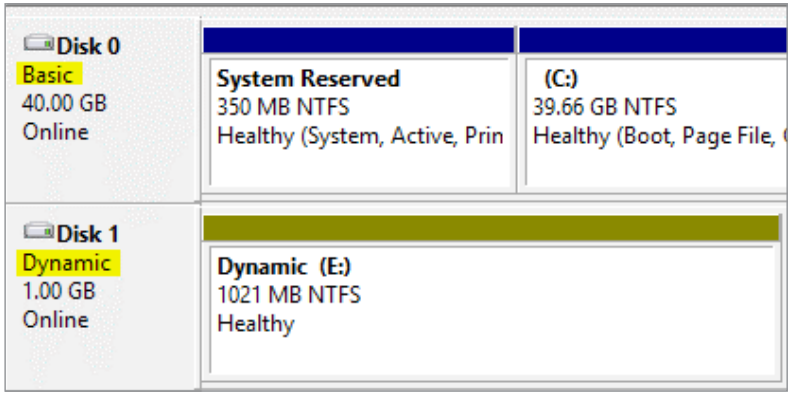
Virtual CPUs in Hyper-V aren't overly difficult to understand, but they're often difficult to plan against. If you really don't know what to do, start a VM with 2 vCPUs and increase them as necessary. If you later find that your VMs are all in contention, the best thing to do is usually to scale out your deployment with more Hyper-V hosts.

CHAPTER 6:
Proper Use of Hyper-V Dynamic Disks

In many cases, Hyper-V administrators will simply use a fixed disk for all of their virtual machines. They're easy to understand and they don't require any special monitoring. The space they start with is the space they always occupy. The problem with this approach is that it wastes a lot of disk space. Part of the reason virtualization is employed is because it has the power to consolidate hardware resources. The purpose of this section is to explain where Hyper-V's dynamic disk can be successfully employed and rebuttals to the common excuses that people give for not using them.

Terminology Clarification

The proper label for this technology is "dynamically expanding". Usually, this is shortened to just "dynamic VHD" or "dynamic disk". However, the term "dynamic disk" also refers to a logical volume type in Windows operating systems, as opposed to the basic disk. These dynamic disks are normally used for Windows-controlled software RAID and by special applications such as Microsoft Data Protection Manager. The only relevance this disk type has to Hyper-V is that host-initiated VSS backups cannot backup VHDs using the dynamic volume type without taking their owner offline.



The rest of this article will use the term "dynamic VHD" and variants of "dynamically expanding virtual disks" interchangeably. Also, no distinction will be made between VHD and VHDX unless it's important.

What Dynamically Expanding Disks Are

The idea behind dynamically expanding disks is very simple to understand. Virtual disks, at their core, are just files. When you create a new virtual disk, you assign it a size. The fixed disk type pre-allocates all that space as a VHD file of approximately the same size. The dynamic disk VHD is initially created as a very tiny file which grows as it is written to. They can be shrunk, but shrink operations must be scripted or performed manually. However, if a file is deleted inside a dynamic VHD, its blocks will be reused rather than further expanding the file. **It cannot grow beyond its set maximum.**

FUD-Busting

The first order of business is to answer the copious amounts of libel leveled against dynamic disks.

"They'll Eat All Your Disk Space and Your VMs Will Die!" FUD

"The cat, having sat upon a hot stove lid, will not sit upon a hot stove lid again. But he won't sit upon a cold stove lid, either." – Mark Twain

This FUD is usually accompanied by a terrifying anecdote about a situation in which a physical storage location for VHDs filled up and all the VMs were paused. This is a risk you take with dynamic VHDs, but blaming the outcome on dynamic disks is inappropriate. The real problems were that the storage allocation was not properly planned and the storage location was not properly monitored. If you, or your organization, have weaknesses in either of these areas, then, by all means, used fixed VHDs.

"Performance with Dynamic Disks is TERRIBLE!" FUD

The performance differences are present, but trivial for most common loads. The real issue is that some individuals have a very unhealthy obsession with performance. If your primary driver is performance, then virtualization is not an optimal solution to begin with.

This FUD is usually reinforced by benchmarks that make a really big deal about the millisecond or two that separates fixed from dynamic. Benchmarks are useful for comparing two different hardware components or tracking performance variance across time. So, if you want to know if the latest AMD chip can process data as quickly as the latest Intel chip, benchmarks are appropriate. If you want to know if fixed disks are faster than dynamic disks, then benchmarks are appropriate. If you want to find evidence that your hardware doesn't run as well as it did when you first bought it and you have benchmark results from earlier, benchmarks are appropriate. Proclaiming that because product A's benchmark is superior to product B's that product B is absolutely useless, shouldn't be used, or anything similar is appropriate only for unscrupulous salesmen/marketing types and blind-worship fanboys. Benchmarks rarely, if ever, give you more than a very general idea of what will happen in production. The benchmark differences between fixed and dynamic do not translate well into generalized production metrics.

"Dynamic Disks Cause Fragmentation and will Make your VMs Useless!" FUD

The benchmarks that were the be-all-end-all of the previous FUD tend to disappear pretty quickly when this FUD is trotted out. That's because the results of the benchmarks don't come anywhere near supporting the premise of the FUD. I won't go so far as to say that fragmentation is a non-issue, but it's certainly one of the most overblown issues in a hypervisor setting. With multiple virtual machines accessing the same storage location, disk access is naturally scattered regardless of fragmentation. With modern multi-spindle systems, the effects of fragmentation are typically minimized anyway. Proponents of this FUD will have a horror story at hand to back it up, usually involving a high-I/O load system that truly isn't appropriate for a dynamic VHD. However, if I have one anecdote and I add another anecdote, what I have is anecdotes, not data.

Something to remember about fragmentation is that drive access on a de-duplicated volume works in a very similar pattern to a fragmented drive, and in many situations is worse. With support for active VHDs in de-duplicated storage locations coming in 2012 R2, keep in mind that it will be inconsistent for someone to claim that fragmentation is a major concern while de-duplication is not.

More on fragmentation at the end of this post.

"So-and-so Says to Never Use Dynamic Disks and S/He's an Expert" FUD

Experts are human and humans are fallible. Not all the experts agree on this topic, and sometimes these quotes aren't considered in context. As an example, for some

loads, only fixed VHD will do. For some message delivery methods, an expert may not have the time or space to properly explain dynamic VHDs, so taking the safe route of recommending fixed disks may seem like the most appropriate thing to do. Some audiences don't have the foundational knowledge or ability to grasp the risks of dynamic VHD, so again, recommending fixed disks may be the safest route. No matter who is speaking or what the subject matter is, advice that includes the terms "always" or "never" should automatically be suspect.

How Dynamic VHDs Operate in the Real World

When you put away the anecdotes and the benchmarks and all the other toys of the FUDdy-duddies and look at a real deployment of dynamic VHDs, what you'll see is they hold up very well when used sensibly. You can make some predictions on size and you must watch drive space usage. The various bits of any given dynamic VHD are generally distributed in a fashion that doesn't cause meaningful fragmentation. Consumers of the applications and services on dynamic VHDs don't see any delays or issues.

Using Dynamic VHDs Sensibly

"Cookies are a sometimes food." – Cookie Monster

You probably don't want to use dynamic VHDs all the time. Applications and services that perform heavy disk I/O are not appropriate. Some vendors will publish best practices recommendations that indicate what they prefer. As a general rule, databases are poor candidates for dynamic disks. Applications that use large amounts of disk space for scratch storage (also usually databases) are better suited for fixed disks. Even with those as guidelines, there are production databases working well on dynamic disks, if they are low-load systems with small amounts of data and only a few users. Even some medium-sized databases will work on dynamic disks. The underlying hardware has more impact than the VHD format.

Two places that are appropriate for dynamic VHDs are those that contain operating systems and those that will contain randomly accessed but slow-growth data, such as general-purpose file servers.

Operating System VHDs

Arguably, the best purpose of a dynamic VHD is virtual disks that hold only operating systems. A prime example is Windows Server 2012 itself. Once in production, it will rarely, if ever, use more than 22-24 GB (unless you have lots of RAM assigned and it needs a large page file). However, if you install it on a 24 GB VHD, you have two problems. The first is that VSS won't have the 20% free space it wants and that can lead to problems. The other is that Microsoft won't support Windows Server 2012 running on anything less

than 32 GB. So, consider a system that holds 100 virtual machines with the "fixed only" philosophy. That's 800 GB of completely dead space. Disk space is certainly getting cheaper, but any organization that considers 800 GB to be cheap enough to just throw away probably also has a lot more than just 100 virtual machines. A very basic driver of virtualization and thin-provisioning is to reduce this waste. Adopting an "always fixed, all the time" attitude defeats one of the major purposes of virtualizing in the first place.

File Server VHDs

File servers are another perfect application of dynamic VHDs. Access is more or less random anyway, so fragmentation is usually a non-issue. No one expects blinding performance out of a file server (at least, no one who has an ounce of reason), so any performance hits are automatically trivial. Also, it's always been normal to create gigantic empty fields of storage space for file servers and then allow them to fill gradually. With even sloppy monitoring, this should be a controllable issue.

Making Fragmentation Go Away

If you're genuinely concerned about fragmentation of dynamic VHDs, you might be happy to know that there's a solution that doesn't preclude their use. Just use Storage Live Migration to empty out a LUN to one or more temporary locations, then use Storage Live Migration to move its virtual machines back. This will eliminate fragmentation at the LUN level.

Summary

Despite fear-mongering, dynamic VHDs are nothing to be afraid of. Use them wisely, and you'll regain potentially budget-saving amounts of space without making meaningful compromises.

CHAPTER 7:
Connecting Hyper-V to Storage

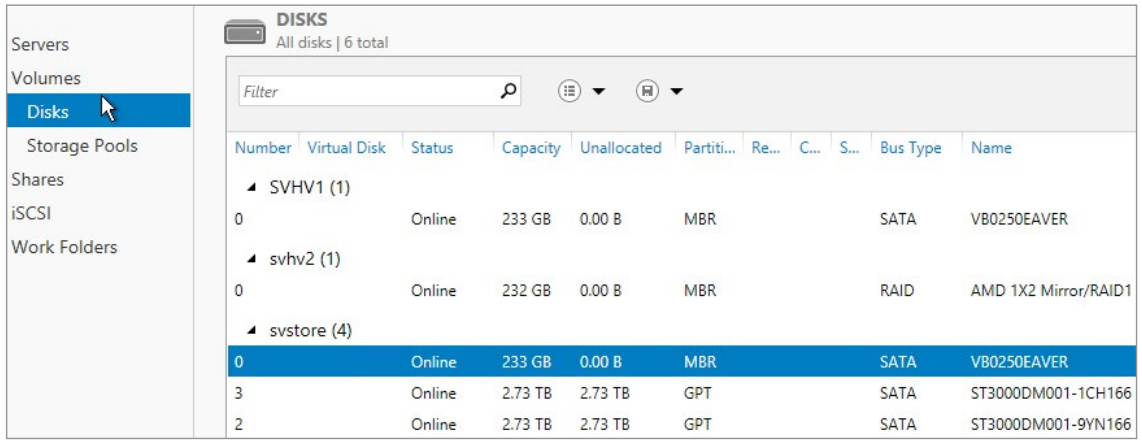
The Altaro Hyper-V blog contains a long and detailed series on storage in Hyper-V, starting from the basics and working up through performance testing. This section presents the most practical portion, which deals with connecting a Hyper-V host to storage.

This section works only with GUI methods, although there are PowerShell equivalents to most of them. Even if you're using Hyper-V Server or Windows Server in core mode, you can connect remotely from a system running a GUI edition of Windows Server 2012 or 2012 R2 and perform these procedures.

Internal/Direct-Attached Disks

In this section, you'll see how to connect to and use local disks. However, once you've connected to a LUN on Fibre Channel or iSCSI storage, it is treated the same way.

You have the option of using the Disk Management MMC console, but this is the 2012 era, and Server Manager is the current tool. Server Manager can't control CD/DVD drives or MPIO. The systems used to generate this content are from our sub-$2000 cluster document. Two machines are running Hyper-V Server. The third is running a GUI version of Windows Server and hosts the storage for the cluster. In the following screenshot, you're looking at the *Disks* tab on the *File and Storage Services* section of Server Manager on that Windows Server. It is also connected to both of the nodes (using item 2, *Add other servers to manage* on the welcome screen of Server Manager). Server Manager also comes in the RSAT tools for Windows desktops. As explained in the section on Hyper-V Manager, you must use Windows 8.1 to manage 2012 R2 servers.



This build is modified somewhat from the original document. SVHV1 is still exactly as described in the document. SVHV2 has another 250GB drive and is running in RAID 1 using the BIOS. SVSTORE has the drives from the document, as well as a pair of 3 TB drives (the 2TB drive didn't make it into the screenshot).

The 3TB drives are completely fresh, with no data. They'll be built into a mirror for the virtual machines. This will give them superior read performance and, more importantly, protect them from drive failure. If this system were intended to host virtual machines locally, it would be preferable to have them in RAID-1. That's because any software mirroring takes processing power, and you really shouldn't be stealing CPU time from your VMs. Unfortunately, the N40L's RAID processor just can't handle drives of this size.

So, this system will use Storage Spaces to create a mirror. Before we get into that, we'll set up a single disk.

Prepare a Local Disk for Usage

This section will use a local Storage Spaces virtual disk for its example. These are example the same steps you'd use for a single internal disk, another type of virtual disk (provided by an internal hardware RAID system), or an iSCSI or Fibre Channel disk.

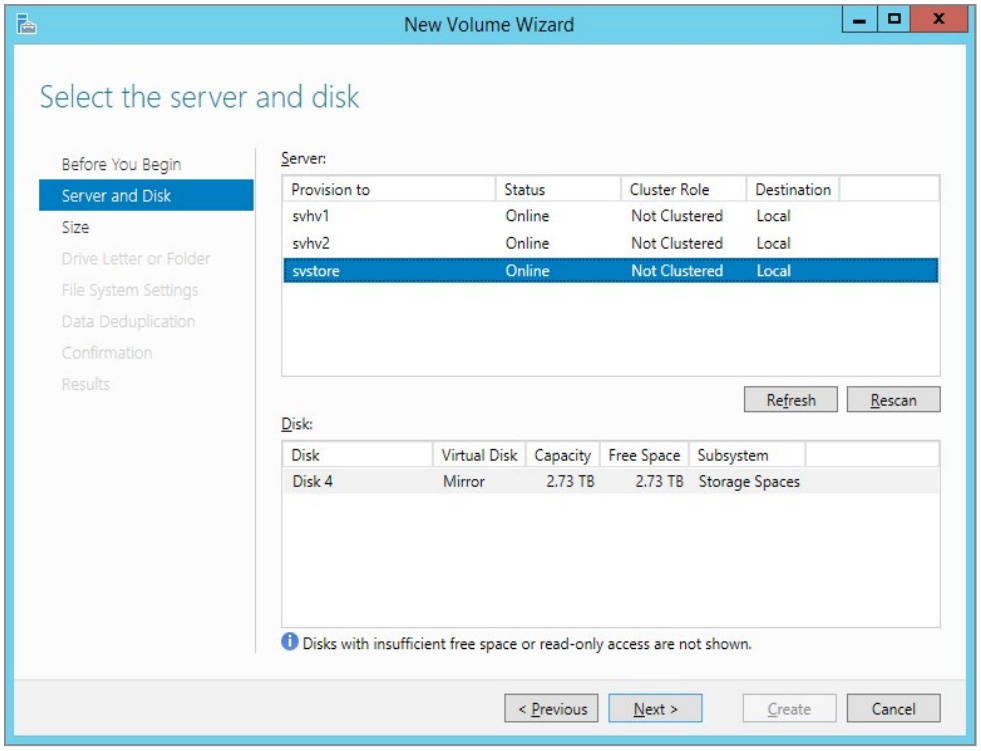
This is only one of the many ways to do this. The way most of us are used to doing it is through Disk Management, and that process has not changed. We'll use Server Manager to demonstrate 2012's new features.

In the *File* and *Storage Services* section of Server Manager, go to the Disks sub-section under Volumes. In the *Disk* list at the top, find your new disk. If it's freshly installed, it's probably *Offline*. If it's online, it may show an *Unknown* partition.

To get started, right-click on the disk to work with and click *Bring Online*. You'll get a warning message that lets you know that onlining a disk that is currently being used by another server might cause data loss. Acknowledge the message (I'm assuming, for this section, that the disk isn't being used by another server). From this point, you'd traditionally Initialize the disk, then create a volume.

With Server Manager, you can now get it all done in a single wizard. Right-click on your freshly onlined disk and choose *New Volume*. This will kick off the *New Volume* wizard. The first screen is informational, so click *Next*.

The second screen has you choose the server and disk to work with. Since you specifically started by right-clicking on a disk, you should already have the correct disk selected:

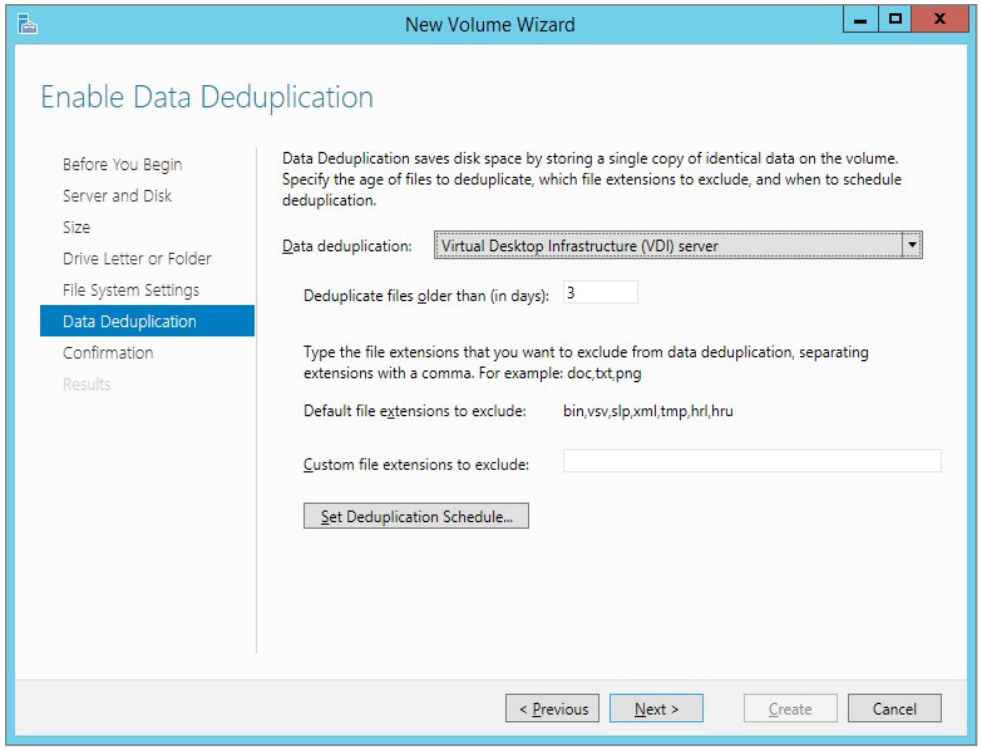


Once you click *Next*, you'll get a pop-up dialog. This occurs because we did not take the Initialize step. It's just telling us that it's going to initialize the disk as a GPT. This shouldn't be a problem, since our system is already bootable and Windows Server 2012+ has no issues with GPT. If you prefer the MBR method for any reason, you'll have to use something other than Server Manager to initialize the disk. Click OK to proceed or Cancel to find another way.

The next few screens are modernized versions of Disk Management's screens: capacity of the new volume, drive letter/mount selection, and format/label/allocation unit specification.

Depending on the options you have available/installed, the next screen will be for Deduplication. I don't want to spend a lot of time on this, but the short introduction is that for Hyper-V, this is most appropriate for VDI. You **can** use it for hosting servers, but you're sort of on your own if performance suffers or you don't get the space savings boost you expect. Remember that Microsoft's deduplication does not occur inline in real time. It occurs on a set schedule.

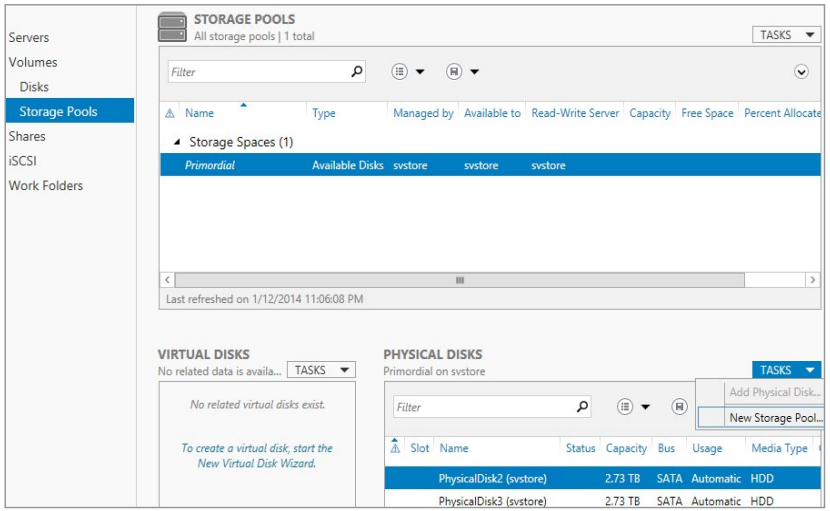
After this, it's the *Confirmation* and *Results* screens, where you can watch your disk get created. Unlike Disk Management, Server Manager's wizard only performs quick formats, so this shouldn't take long regardless of disk size. Once it's done, your disk is ready to use.



Prepare a Storage Spaces Volume for Usage

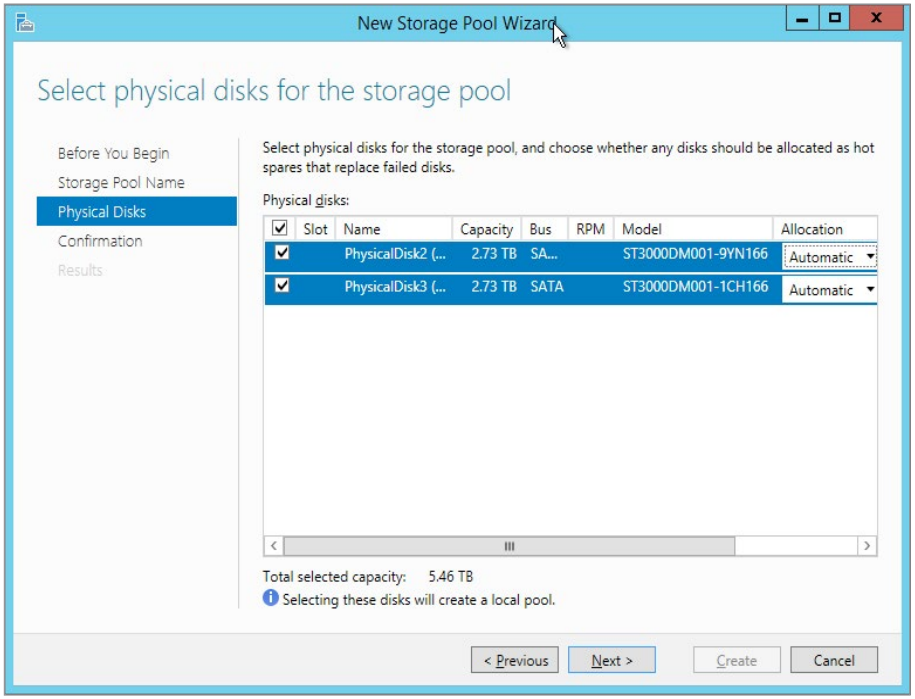
The disk(s) that you want to use can be online or offline (right-click option on the *Disks* tab), but they must have at least some space that's not already claimed by a volume or partition. For this demo, I'm going to use completely clean disks.

To get started, go to the *Storage Pools* section in File and Storage Services. In the lower right, under *Physical Disks*, make sure that the disks you want to use appear. You can use the *Tasks* menu on the *Storage Pools* section at the top to *Refresh* or *Rescan Disks* if some are missing. To get started, open either of these menus and click *New Storage Pool*.



On the first screen of this wizard (not shown), you give the new Storage Space a name and, optionally, a description. For the shown system, we just used "SVSTORE Space". Click Next when you're happy with the name (or at least accepting of it).

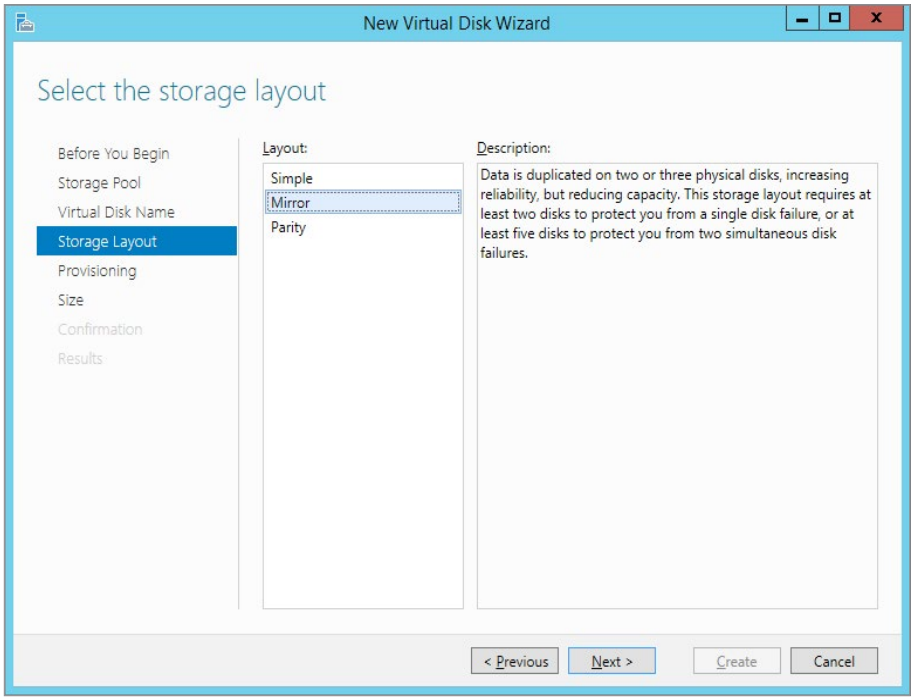
On the second screen, you select the disks you want to be part of the new Storage Space. On the right, each disks has a drop-down for *Automatic, Manual, or Hot Spare*. *Automatic* allows Storage Spaces to figure out the best way to use the disks *Manual* allows you to specify the interleave size during the creation of a virtual disk (that part comes later). *Hot Spare* does just what it says, making the disk available as a hot spare. If you're not familiar with this term, a hot spare disk sits empty until an array disk fails. The data from that disk is copied to the hot spare and it then comes online in place of the failed disk. Usually, hot spares are used in a parity system or as a +1 for mirrors or RAID-10 configurations. I selected Automatic for both my drives. Click *Next* once you've set the selections as desired.



Your hard work will be rewarded with a summary screen. If you click it, click *Create*. If you don't, click *Back* or *Cancel*. These directions will assume you went with *Create*. In that case, you'll get a screen with a few progress bars. Once they're done, hopefully none of them turn red. These directions will also assume they stayed blue. Once the process completes, you can *Close*. Before you do that, you might want to check the box for *Create a virtual disk when this box closes*. If you don't, then you'll have to right-click on the new storage pool you just created and select *Create Virtual Disk…*, which is an entire extra click.

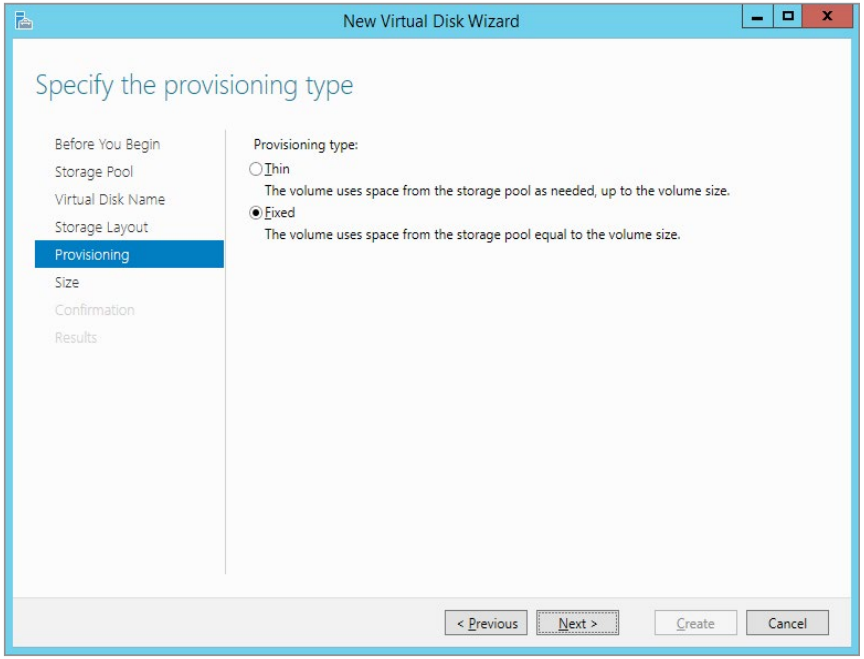
The first two screens of the virtual disk wizard are pretty easy. The first is just explanatory welcome text and the second has you choose your pool. In the demonstration system, there is only one. On the third screen, you have to give your new virtual disk a name and, optionally, a description. The demo system uses "Mirror". You'll notice there's a checkbox to tier the storage. It will be grayed if you have only spinning disks; you need both spins and solids in the same batch for tiering to work. Click *Next* when this page is good enough.

It's on the fourth screen that you get your first real choice: *Simple, Mirror, and Parity*. These are effectively RAID-0, RAID-1, and RAID-5/6, respectively. There are some differences between these and the industry-standard versions, but that's a discussion for another time.

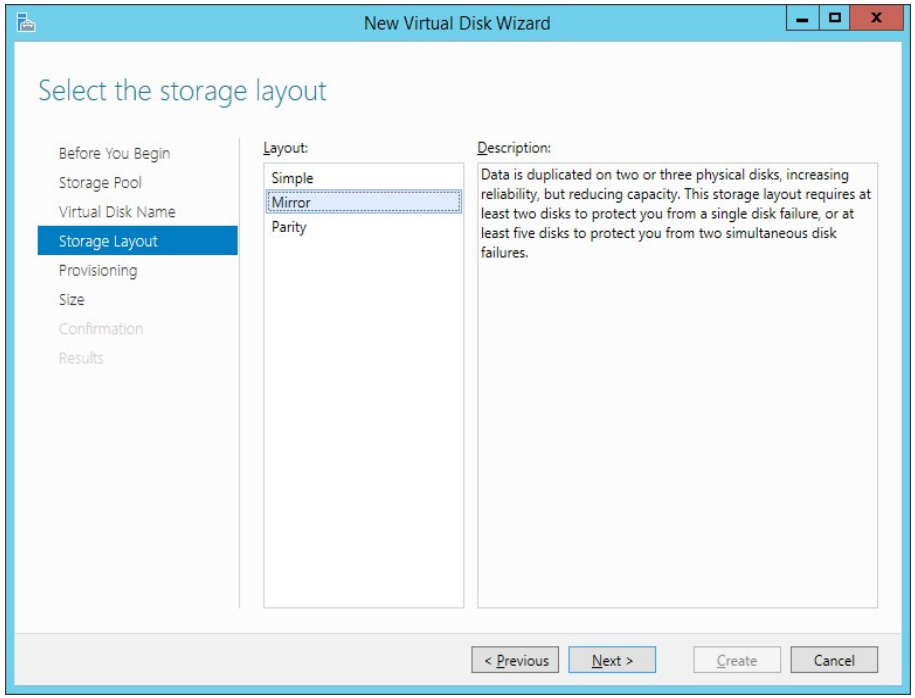


The next screen has you choose between thick or fixed (thin) provisioning. The upside of thin provisioning that things like volume creation can be a lot faster and snapshots can be a lot smaller. The downside is fragmentation and other performance hits when the space is expanded. In larger storage systems with many spindles, the cost of these operations is usually lost in other latencies and is irrelevant to all but the most demanding systems (or neurotic administrators). In a two-spindle system, the hit is more noticeable… although the load is usually a lot lighter, so it

still probably doesn't really matter. The demo system uses thick provisioning because the storage will never have any other purpose.



The final screen you can make changes on is the size screen. The demo system will only use a single Space, but feel free to split yours up as you see fit. You might want to have separate Spaces so you can have some for Hyper-V storage and others for regular file shares or SOFS storage.



After this, there's nothing left but to review the Confirmation screen and watch the progress bars on the Results page. Well… actually, there is more. I would definitely check the *Create a volume when this wizard closes* checkbox before clicking Finish. Otherwise, you'll have to jump over to the *Disks or Volumes* tab to start the wizard, and that's a lot more than just one extra click. It's at least three.

We won't go through the New Volume wizard again. Read the **Prepare a Single Disk for Usage** section if necessary.

Fibre Channel

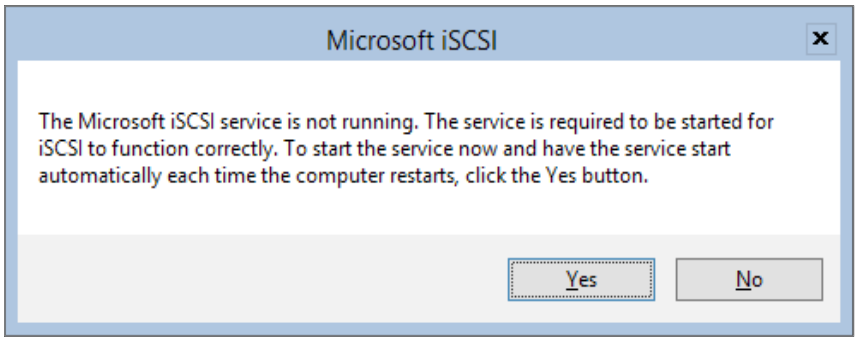
We'd love to give you a nice how-to guide on FibreChannel connections. Unfortunately, it's really vendor-driven. The basic process involves loading drivers and software for your host bus adapter (HBA), then masking and connecting World Wide Names. Work with your storage vendor.

iSCSI

iSCSI works in a client/server configuration, but it uses the terms **initiator** instead of "client" and **target** instead of "server". Perform any necessary configuration on the target. As with Fibre Channel, configuration of the target is determined by the storage vendor. The target needs to be set up first.

Before you get started configuring Windows or Hyper-V Server as an initiator, there are a few things you need to sort out. Don't team NICs to be used in iSCSI. You can use vNICs that are connected to a Hyper-V virtual switch that is hosted to a team, but dedicated physical NICs are faster. The initiator and target should be in the same subnet(s) if at all possible. Routed iSCSI traffic can be noticeably slower.

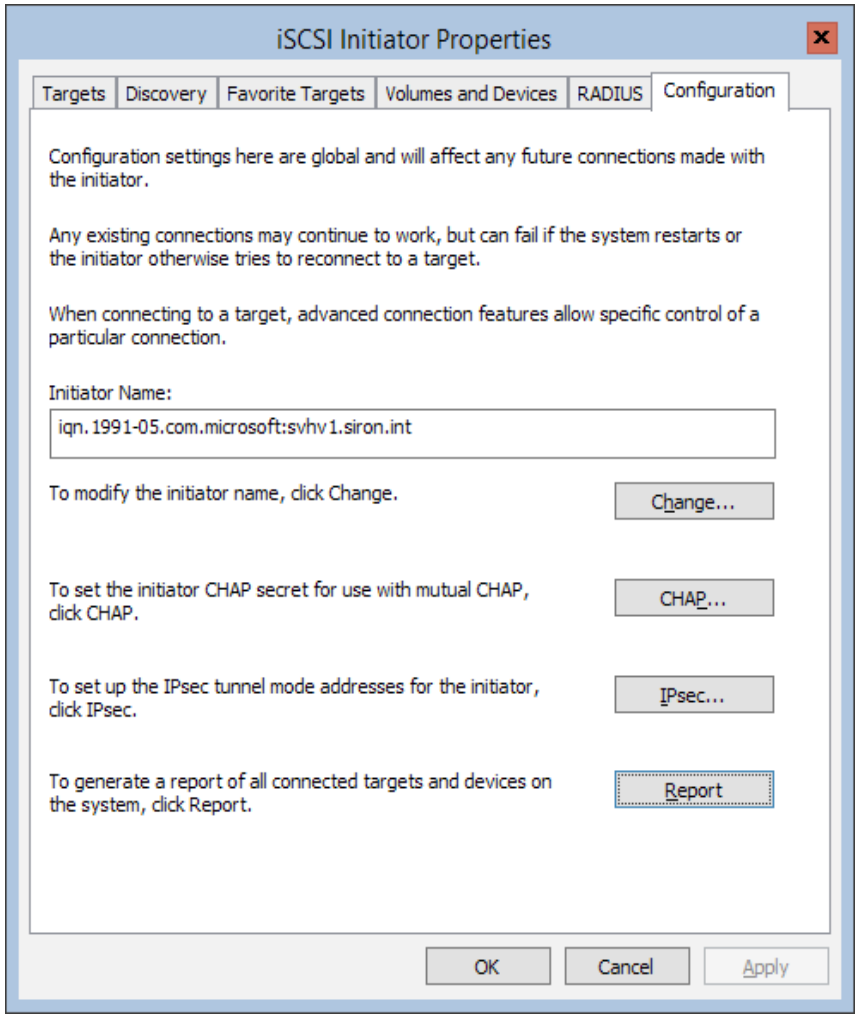
There is a connection GUI for iSCSI on all modern versions of Windows and Hyper-V Server. It exists whether or not the GUI is installed. At any command or Run prompt, just type **iscsicpl.exe.** If you've never run it before, you'll be presented with the following dialog:



Click *Yes* to proceed. Next, you'll be presented with the iSCSI Initiator Properties dialog. This is a busy dialog, and there is a lot going on. Start on the *Configuration* tab, because it contains valuable information that usually gets skipped in iSCSI discussions.

iSCSI Initiator Configuration Tab and iSCSI Security

Let's take a look at the Configuration tab:



The vital data here is the Initiator Name. When this computer requests to connect to a target, this is the name it's going to present. You'll notice you can change this. Hopefully, you're asking, "Then, is it possible for me to lie and pretend I'm an entirely different computer?" The answer is yes, absolutely you can. All Microsoft initiators, by default, start with **iqn.1991-05.com.microsoft:** and then the FQDN of the computer. You could change the initiator in an attempt to confuse a potential attacker, but this is an approach called "security by obscurity" which is rightly derided because it's only going to fool the laziest and most incompetent attackers. However, most targets do allow you to restrict access to specific initiators and IP addresses. IP addresses are a bit tougher to spoof than initiator names, since the network behaves poorly and has blatantly obvious problems when two systems try to use the same IP address at the same time, but this is still not a good security measure.

If iSCSI security is really a concern, then you have three choices. The first is CHAP (challenge-handshake authentication protocol). This is a simple security method that involves a pre-shared key. The CHAP button you see on the dialog above is used only for **mutual** or **reverse CHAP**. For this, the *target* will present a password to the **initiator**, because it's just as easy to spoof a target as to spoof an initiator. Here is where you set that password. It is important to understand that the **only** security this provides is at the

moment of connection. If CHAP fails, no connection is made. If CHAP succeeds, everything sends back and forth in the clear without further verification, at least until the next connection attempt.

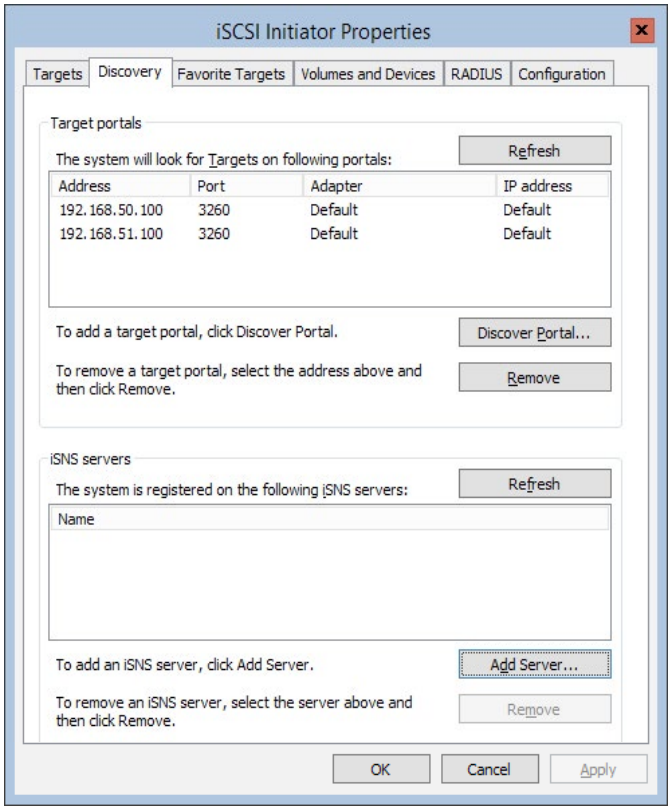
The second iSCSI security method is IPSec. What IPSec gives you is a completely encrypted communications chain, but at a high overhead cost, and that's before you get into the administrative nightmares. We're not going to talk about IPSec any further except to say that this is where you configure it on the initiator side. If you're thinking about IPSec, it's highly recommended you also consider iSNS as it might reduce your headaches a little. iSNS will be briefly revisited in the next section.

The third method is to isolate iSCSI onto its own network. We've already talked about using a dedicated subnet for performance reasons, but it's also an easy way to get some security for the traffic. If you don't put any gateway devices in your iSCSI network, then an attacker will be forced to compromise your network at the layer-2 level, which is difficult, to say the least. You can employ VLANs to add a bit more security. The best approach is to use dedicated switches and switch ports.

The *Reports* button needs a GUI edition of Windows Server to run.

iSCSI Initiator Discovery Tab

Discovery should be self-explanatory, but there is, unfortunately, a lot of confusion with it. Discovery merely finds out what is available on a given target. You provide the target IP address (or name, but of course, it will be resolved to an address), and Discovery goes to see what disks are there. Just because an initiator can discover disks doesn't mean that it can connect to them! Here is a shot of the Discovery tab with some portals added:



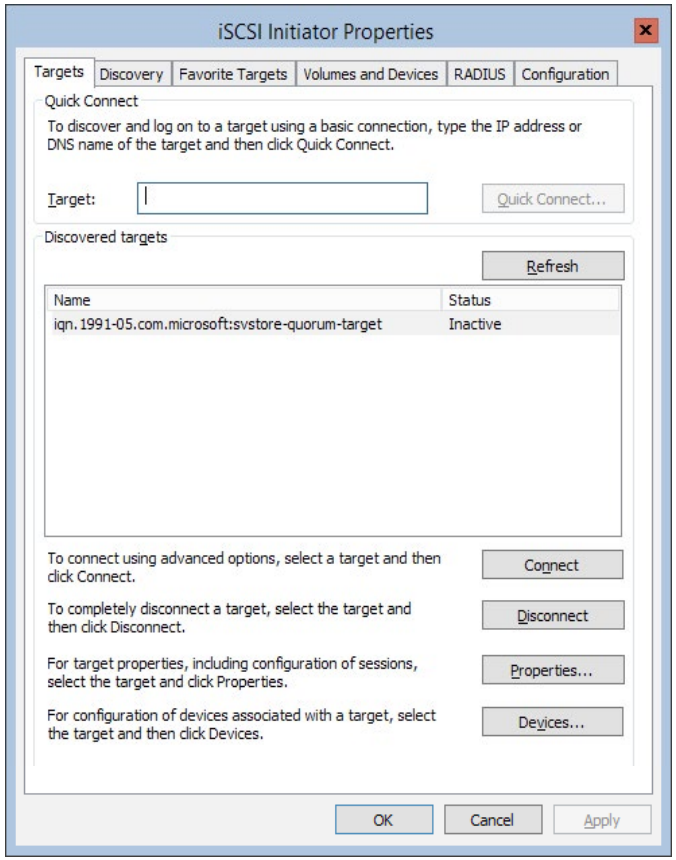
Use the *Discover Portal* button to enter the portals you want to connect to. When you click, you'll be given a small dialog that will allow you to enter the IP or DNS name of the target as well as the port to connect on. It is recommended that you use the IP, or your iSCSI infrastructure will be dependent upon DNS. It is also recommended that you not change the target's port number without a compelling reasons (security by obscurity is not a compelling reason). You can click the *Advanced* button on this small dialog to configure more security options. This dialog will be shown and discussed in the next section, as it is the same dialog you see when connecting to a disk. However, the context is very important. This page is **just** for connecting to the discovery portal. Settings here are completely distinct from disks. Most initiators do not secure their portals against discovery attempts, so setting things here might actually cause unnecessary problems. The only thing I set here is ensuring that the source addresses for each portal is on the same subnet as the target, but that's not strictly necessary. Once you've entered information for a portal, it will be displayed in the dialog as shown above. Please be aware that you can completely mangle the information and it will still be accepted, although it's not going to be useful.

Unfortunately, there is no way to modify the settings for an existing portal item. You have to remove it. This will pop up a warning dialog about the item being on the Favorites tab. Deal with that before trying to re-add a portal, or you might be in for some confusion. Look to the relevant section below for more information.

The lower portion of this dialog is for iSNS settings. It requires an iSNS server, obviously. What you can do with such a thing is preconfigure just about everything, including IPSec settings, and manage them from a central location. This is not common usage, but its feature list is fairly compelling if you're looking at a complicated iSCSI deployment.

iSCSI Initiator Targets Tab

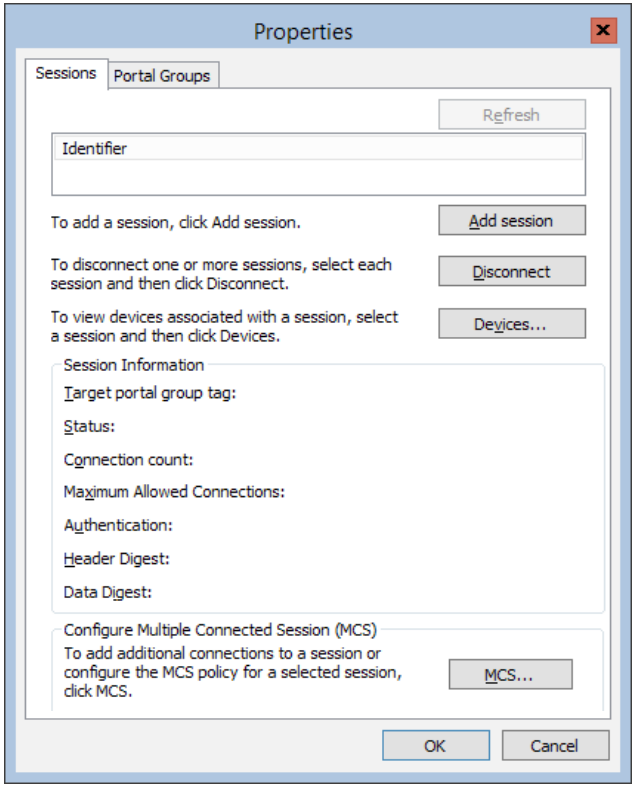
The Targets tab is the first you see when you start the applet. Here is what one looks like after a target has been added (and maybe with a click of the *Refresh* button):



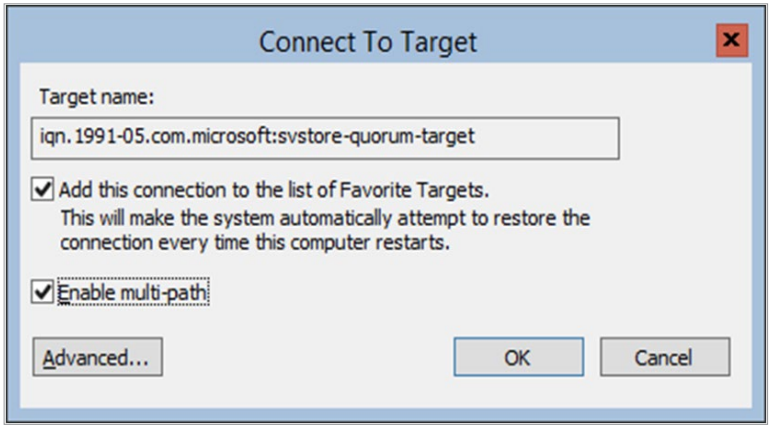
The *Quick Connect* button is for attaching to a new portal and target, not one that's already in the *Discovered targets* box. It doesn't allow for any modifications to security or the port, so you might skip using it in favor of the *Properties* dialog.

Target Tab's Properties Sub-Dialog

By highlighting an item on the Discovered targets list with a status of *Inactive*, you might be able to use the *Connect* button to jump right onto it. However, you won't want to do this if there are any special needs, such as multiple source and destination IP paths. This is because you're only going to get one shot to create connections. Some initiator software may intervene and build up all the necessary IP paths, so this may not matter. For most of us, we'll use the **Properties** button. You'll get this box:

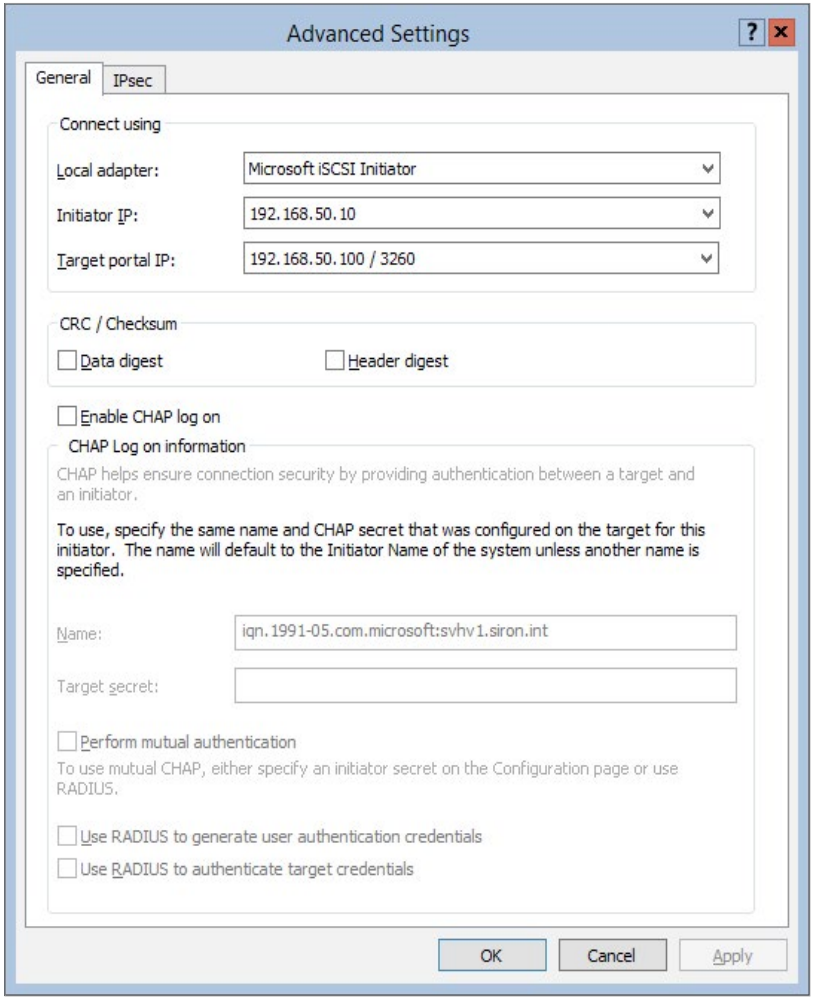


If there are any active connections to the disk, the list box at the top will contain entries. Your ultimate goal is to have one session per path to storage. For the demo system, the initiator has IPs 192.168.50.10 and 192.168.51.10, while the target has IPs 192.168.50.100 and 192.168.51.100. So, this requires two, one on 192.168.50.0/24 and the other on 192.168.51.0/24. We start by clicking *Add session*, and we get the following (this is the same dialog you get when clicking *Connect* on the Targets tab):



In this screen capture, *Enable multi-path* has already been selected, as this system uses two paths. But, we want to define the paths manually, because Windows usually doesn't figure out the connections.

Click the *Advanced* button to set that up. You'll be presented with the *Advanced Settings* dialog (which is the same dialog you get when adjusting Advanced Settings on the discover portal):



These options have been set in accordance with the first connection. Don't just arbitrarily select either *Data digest* and/or *Header digest* because they sound important. If the target isn't configured to use them in the exact same pattern you select here, your connections won't work at all. What they do is verify data validity at the expense of a bit of overhead. The boxes indicate whether you want to calculate a checksum on the data portion of the iSCSI TCP/IP packet or its header, or, of course, both.

The lower portion of this screen is for CHAP security. You need to make sure that this is configured the same way that your target is. The *Target secret* refers to the password that the initiator is expecting **for this disk, NOT** the discover portal. As mentioned elsewhere, mutual authentication is the password that the target will attempt to send back to the initiator and is set on the Configuration tab. Your target may refer to this is Reverse CHAP. The lower two boxes are for RADIUS authentication. The topic of RADIUS with iSCSI is beyond this basic documentation, but the basic approach is to ensure that both your target and initiator can contact the same RADIUS system.

Once you have this screen the way you like, click *OK* twice. You'll be back at the *Properties* screen, and if all is well, you'll have an entry in the list box with an undecipherable identifier made up of two gigantic hexadecimal numbers.

Now, go back through the *Add session* dialog and repeat the earlier steps to set up a connection from initiator IP 192.168.51.10 to target IP 192.168.51.100.

The Devices button allows you to look at some connection details for this disk. Much of it will probably be pretty cryptic. However, there is an MPIO button that, as you will see, can be quite useful.

At the very bottom of the Properties dialog is the section on MCS (Multiple Connected Session). This is not commonly used as MPIO has much broader support. MPIO will be discussed on the next section. For now, let's go to the next tab in the iSCSI dialog.

iSCSI Initiator Favorite Targets Tab

There is no screen capture of this tab because there's not much to it. When the system reboots, it will try to connect to every iSCSI disk listed here. If you're using multiple paths, then each path will be listed twice. The *Details* button will show how you've configured the connection. Connections you've made from anywhere else in this applet will automatically be placed here. If you don't ever want to connect to a disk again (or if you want to reconfigure it), remove it from here.

iSCSI Initiator Volumes and Devices Tab

There is no apparent practical use of this tab is. Maybe some apps benefit from it. Hyper-V Server will function just fine if you never use it or if you fill it all in. Use the *Auto Configure* button if you like, but there will be no obvious effect.

iSCSI Initiator RADIUS Tab

This is for informing the initiator about RADIUS servers it should use if it's going to participate in RADIUS authentication to targets. As previously stated, RADIUS is beyond the scope of this discussion; if you know how to set up RADIUS, this page will be easy for you.

iSCSI Additional Points

Once you have everything configured, you're connected. Windows will now treat the disk as though it were local. Just look in Disk Management or DISKPART or Get-Disk to see it. Read the Internal/Direct Attached Disks section above to continue.

The issue with iSCSI is it really doesn't have a good "reconnect" mechanism. If it loses connectivity, it's supposed to try to reconnect on its own. If you've manually disconnected or determined that a disconnect occurred, you must restart the **server** (not the service) or rebuilding the connection from scratch. If you have an iSCSI connection made that is idle for a very long time, it seems like it quietly drops the connection but thinks it's still active (this could be a target issue). When does this happen? With Cluster Shared Volumes, mostly. Be aware that if you have only one node accessing a CSV for a very long time, the other node(s) may disconnect from the disk underneath that CSV. If you're lucky, the CSV will just go into Redirected Access mode if you try to move a VM to it. Otherwise, the VM won't be able to talk to its disk. This condition seems to be very rare, but it's something to be aware of.

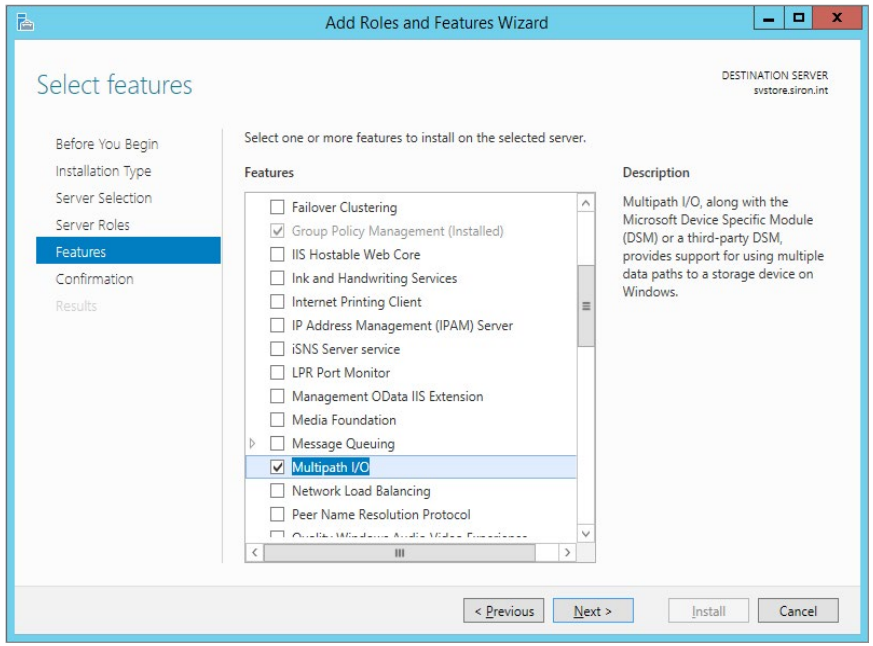
Multi-Path I/O (MPIO)

Multi-path I/O is a topology-neutral technology for establishing more than one connection from the same host to the same storage across unique pathways. It's mostly talked about in terms of iSCSI, but it's absolutely not limited to that. If you have two internal SAS controllers and your internal disks have two SAS connectors, you can use MPIO. If you have two controllers to connect to your external storage device and it has two or more connections, you can use MPIO for that.

If you have multiple connections established, but don't use MPIO, Disk Management will show the disk twice. One of them will report: "Offline (The disk is offline because it has a redundant path with another device)". If the disk is formatted, Server Manager's disk console will show it twice, with one of them having *Unknown* for Status and *Partition*.

Get-Disk is much more terrifying, giving one of the two disks a status of *Failed*. Only DISKPART is happy with it, but it will show zero-bytes of capacity on one disk.

MPIO is in all flavors of Windows Server and in Hyper-V Server. But, you have to enable it. If you're using the Add Roles and Features wizard, it's on the Features page:

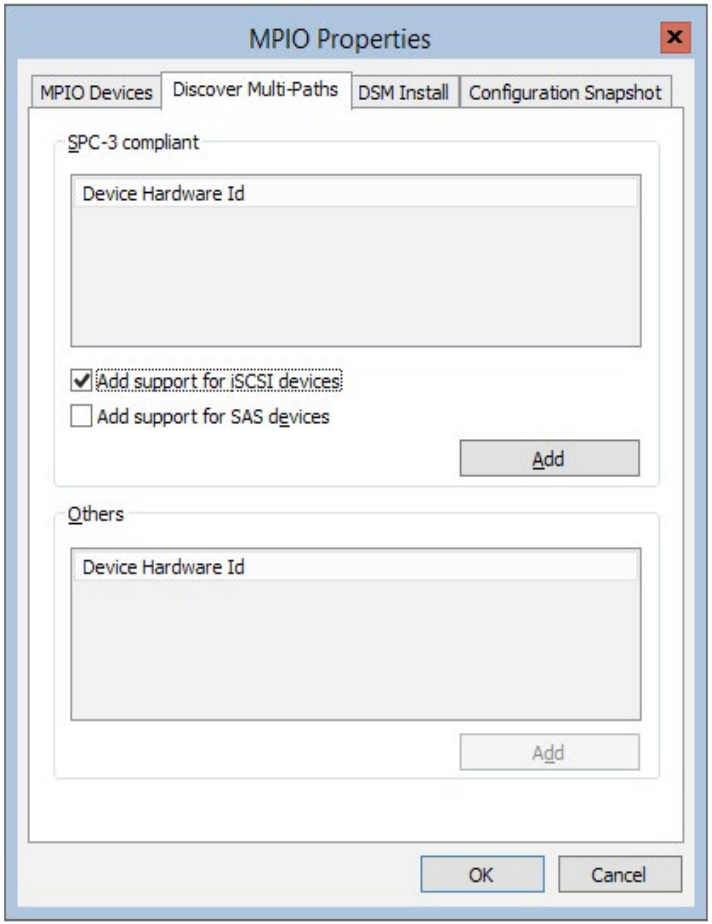


You do **not** need to restart after installing. You will need to restart after the first time assigning it to a disk system.

We'lll only going to talk about iSCSI disks here because that's the topic of this section. Be aware that many vendors have their own method of setting up MPIO, so **read the manual!** What follows is for a generic iSCSI target.

After you've installed MPIO, run MPIOCPL.EXE at any command or PowerShell or Run prompt. As with ISCSICPL.EXE, this will work whether you are on a GUI server or not. The first tab you'll see is *MPIO Devices*. It's pretty much useless at the beginning. It will show an item named *Vendor 8Product 16*. This is a meaningless placeholder for developers to see how it's done. What you want to do is flip over to the Discover Multi-Paths tab. Since we're talking about iSCSI, tick that checkbox. If you're here for SAS, tick that one.

Then, click the *Add* button:



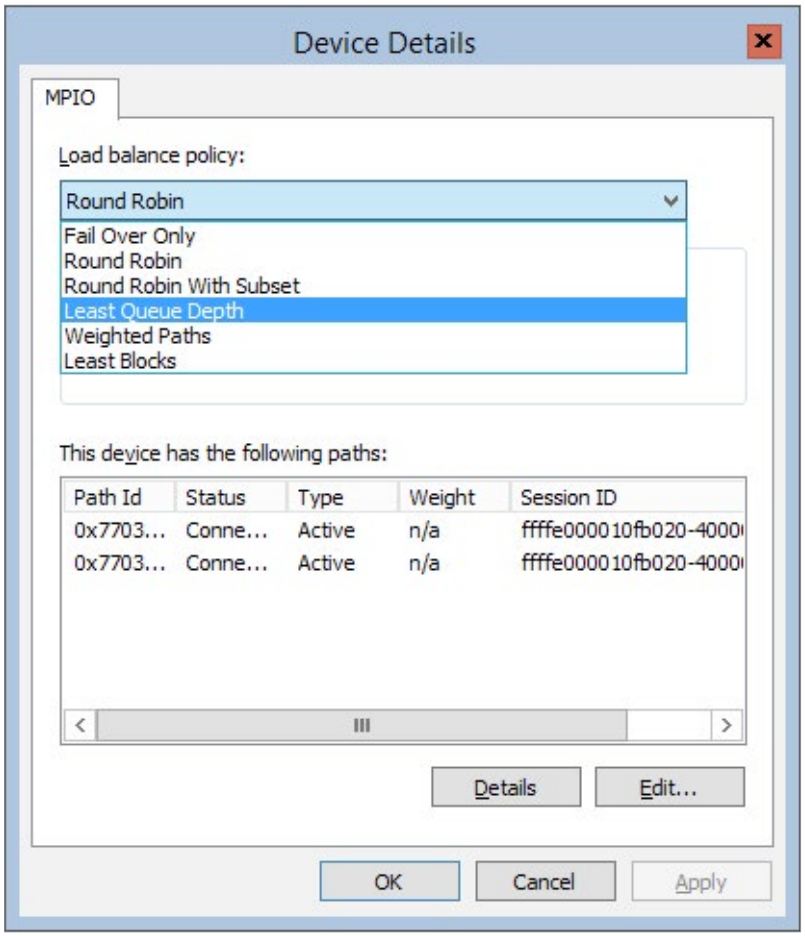
Now, you're going to have to reboot. When it comes back, everything will be fine.

The other tabs in the MPIO dialog box aren't really that useful unless your vendor documentation indicates that they are.

In case you're curious, no, you can't set up MPIO path discovery before you make the connections to storage, nor can you avoid the reboot. You could opt to make only one connection from storage, add MPIO support, then add another connection, if you wish. But, that doesn't really do anything for you.

By default, MPIO works in load-balanced configuration with round robin. That means it will send sequential requests down the other path. Because, just like any other I/O request, this can get imbalanced, you may wish to use a different load-balancing algorithm. On a local system, Disk Management allows you to modify MPIO settings on the property sheet of an MPIO disk.

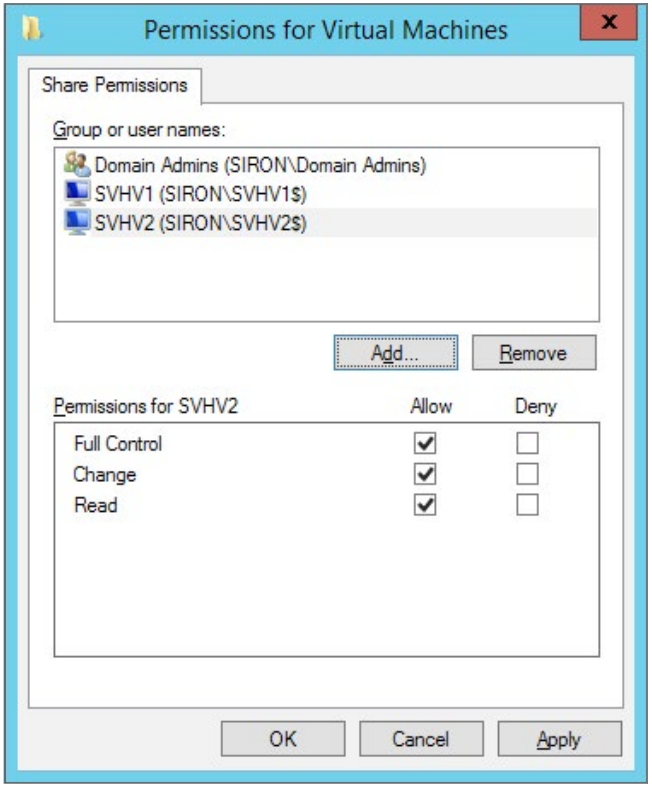
Unfortunately, Disk Management cannot set MPIO policy on a remote server. For a GUI-less or remote system, you can use PowerShell. If you must, MPCLAIM.EXE still works. For iSCSI disks, you can find an MPIO button on the Properties sheet for a disk in ISCSICPL.EXE (read the iSCSI sections above if you don't know how to find this).



When you select an option from the drop-down, this will show a details pane that explains the setting you've picked. Making a selection here configures the policy for all connections to that disk, but does not affect other connections to other disks.

SMB 3.0

This will be easy: as long as the share and NTFS permissions are configured properly, all you have to do is point Hyper-V to it as shown in the following screenshot.



The really nice thing is that this is, by default, shared storage. It works right away for standalone and clustered systems alike.

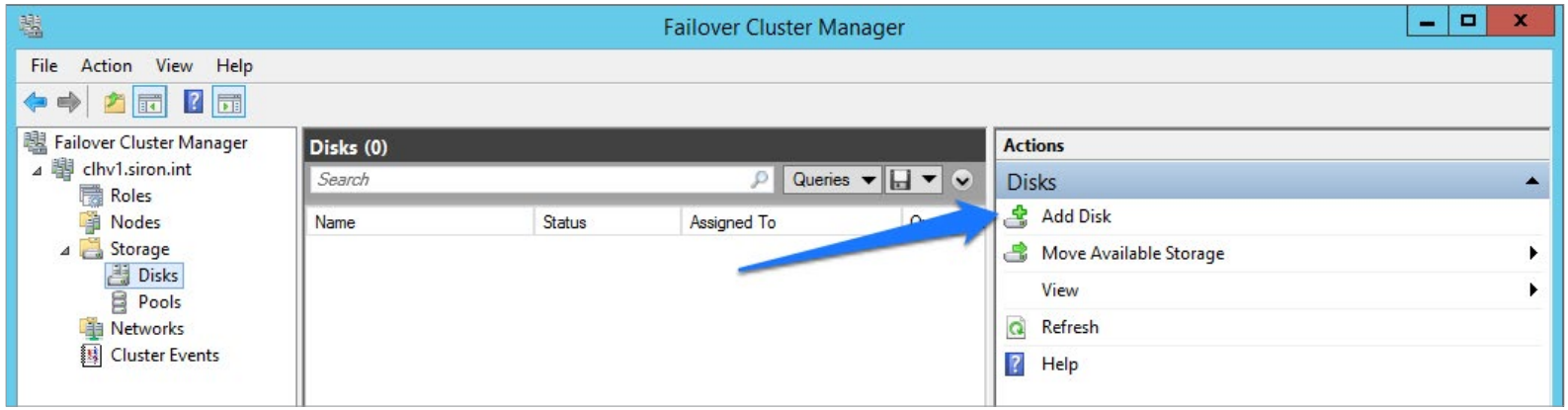
Storage for a Hyper-V Cluster

Aside from SMB 3 storage, there are two specific ways to use storage in your cluster. The first, the **Cluster Disk**, is the oldest, and the one that works for every single application that can be covered by Microsoft Failover Clustering. A cluster disk can only be owned by a single node at any given time, and only that node can access the disk's data. If another node, or any other system, tries to access it, the data will be corrupted. So, when a virtual machine on a cluster disk is Live Migrated to another node, disk ownership moves with it. For this reason, if you use a standard cluster disk for Hyper-V, only one virtual machine can be stored on it. This is because you can't Live Migrate two virtual machines in precise lockstep.

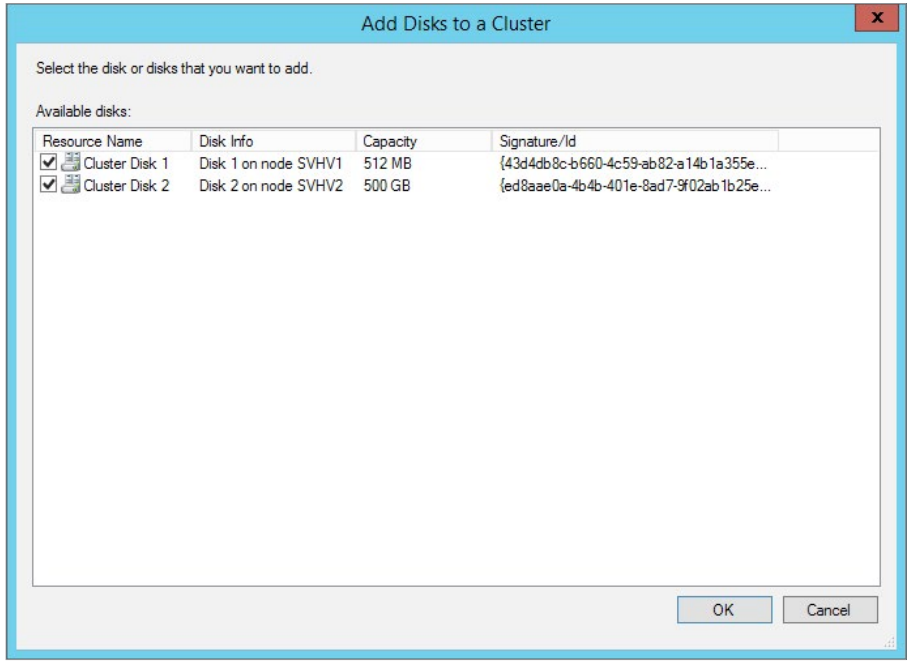
Standard Cluster Disks

The first step is to connect each node to storage. For direct-connected systems, this is probably just a matter of plugging it in. For iSCSI, read above. For Fibre Channel, read your manufacturer's instructions. For bringing it online and formatting it (as described in the **Prepare a Local Disk for Usage** section), do this on only one node (it doesn't matter which). Leave it Offline everywhere else.

Once the disk is connected to all nodes, access Failover Cluster Manager, expand the *Storage* node, and click *Disks*. In the right pane, click *Add Disk*:



A progress bar will show while storage is scanned on the nodes for disks that can be used in clustering: anything that's not local storage, is formatted as NTFS or ReFS, is visible from all nodes, and is online on one node.



Be aware that the *disk # on node #* header might be the easiest way to identify which disk you're actually looking at. The *Capacity* might be of assistance, as well. Select the disks you want in your cluster and click *OK*. The disk(s) will now be in the list of disks in Failover Cluster Manager. If you're not especially taken with the name *Cluster Disk 1*, you can double-click it and change the name in the Properties sheet. You might also want to flip through the other tabs of this dialog, although most people won't want to change it. You can now set about using the disk. Cluster disks are optimal for the disk witness in a quorum.

Cluster Shared Volumes

A Cluster Shared Volume (CSV) is only owned by one node at a time, but all nodes in a cluster have direct read/write access to CSVs, provided they have their own connection. If a node can't directly connect to the storage underlying the CSV, its I/O is redirected through the node that currently owns the CSV. This is called **Redirected Access Mode**; the owner node is called the **Coordinator Node**. Metadata operations (file open, close, rename, etc.) are always handled through the coordinator node and transmitted throughout the cluster over the cluster network(s). Redirected I/O also moves over the cluster network(s). Despite a common misperception (likely due to some confusing early documentation on the subject), there is no dedicated CSV network. The benefit of the CSV is that multiple VMs can be stored on the same CSV.

In order to create a CSV, first follow the steps to create a standard cluster disk. Then, still in the Disks node of Storage in Failover Cluster Manager, right-click on the

disk and click *Add* to *cluster shared volumes*. All done. As with the cluster disk, you can rename it by double-clicking it. You'll notice there are fewer options for a CSV than for a standard cluster disk. The cluster really handles the care and feeding of its CSVs.

Your CSV is accessible through a **Symbolic Link** (technically, it's a **Junction** if it refers to a folder, but that's probably just trivia). If you look inside the *ClusterStorage* folder on the system drive of any node, you'll find a "folder" named *Volume1*. This folder can be renamed, and you can put data in it. Don't let the familiar view fool you though, this is actually a direct connection to the root of the volume that was a cluster disk just a few moments ago. When you create new virtual machines, target them to this folder, they'll automatically be on highly available storage (but that doesn't mean the VM is highly available — a discussion for another topic).

Summary

You should now be able to connect to just about any type of storage, whether external or internal. Don't forget that there is also a rich PowerShell set for storage that can duplicate most everything you've seen in the GUI: http://technet.microsoft.com/en-us/library/hh848705.aspx.

About Altaro

Altaro Software (www.altaro.com) is a fast growing developer of easy to use backup solutions targeted towards SMBs and focused on Microsoft Hyper-V. Altaro take pride in their software and their high level of personal customer service and support, and it shows; Founded in 2009, Altaro already service over 15,000 satisfied customers worldwide and are a Gold Microsoft Partner for Application Development.

About Altaro VM Backup

Altaro VM Backup is an easy to use, yet powerful backup solution for Microsoft Hyper-V, which takes the guesswork out of backing up VMs and does all the complex Hyper-V backup configuration for the admin. This means best in class technology at the most competitive price on the market.

Demonstrating Altaro's dedication to Hyper-V, they were the first backup provider for Hyper-V to support Windows Server 2012 and 2012 R2 and also continues support Windows Server 2008 R2.

For more information on features and pricing, please visit:
http://www.altaro.com/vm-backup/

Don't take our word for it – Take it for a spin!

DOWNLOAD YOUR FREE COPY OF ALTARO VM BACKUP
and enjoy unlimited functionality for 30 days. After your 30-day trial expires you can continue using the product for up to 2 VMs for free, forever. No catch!

Follow Altaro

Like our eBook? ***There's more!***

Subscribe to our Hyper-V blog
http://www.altaro.com/hyper-v/
and receive best practices, tips, free Hyper-V PowerShell scripts and more here:
http://www.altaro.com/hyper-v/sign-up/

Follow Altaro at:

Share this resource!

Share now on: