

Introducing Windows Server 2012 R2

Preview
Release

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright 2013 © Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Control Number (PCN): 2013945003
ISBN: 978-0-7356-8293-1

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Karen Szall

Project Editor: Valerie Woolley

Editorial Production: Christian Holdener, S4Carlisle Publishing Services

Copyeditor: Andrew Jones

Contents

| | |
|--|------------|
| <i>Introduction</i> | <i>vii</i> |
| Chapter 1 Cloud OS | 1 |
| The big picture..... | 1 |
| Journey to the Cloud OS..... | 2 |
| Let's begin! | 3 |
| Chapter 2 Hyper-V | 5 |
| Previous enhancements to Hyper-V | 5 |
| Generation 2 virtual machines..... | 7 |
| Automatic VM activation..... | 10 |
| Remote access over VMBus..... | 11 |
| Cross-version live migration..... | 12 |
| Faster live migration | 13 |
| Online VHDX resize..... | 15 |
| Live export..... | 16 |
| More robust Linux support | 18 |
| Hyper-V Replica enhancements..... | 19 |
| There's more! | 22 |
| Chapter 3 Storage | 25 |
| Previous enhancements to storage | 25 |
| Microsoft's vision for storage..... | 27 |
| Building the solution using Windows Server 2012 R2 | 27 |
| Enabling the solution using System Center 2012 R2..... | 29 |
| Storage Management API..... | 30 |
| iSCSI Target Server enhancements | 31 |
| SMB 3.0 enhancements..... | 33 |
| Data deduplication enhancements..... | 35 |

| | |
|---|-----------|
| Storage Spaces enhancements..... | 36 |
| Storage Spaces in Windows Server 2012..... | 38 |
| Storage Spaces in Windows Server 2012 R2..... | 39 |
| Storage QoS..... | 40 |
| There's more!..... | 41 |
| Chapter 4 Failover Clustering | 43 |
| Previous enhancements to Failover Clustering..... | 43 |
| Guest clustering using shared virtual disks | 44 |
| Hosting highly available workloads..... | 44 |
| Separating virtual resources from physical infrastructure | 45 |
| Understanding shared virtual disks | 47 |
| Using shared virtual disks | 49 |
| CSV and SoFS enhancements | 51 |
| Changes to heartbeat threshold..... | 53 |
| Detecting the health of virtual machines..... | 55 |
| Other enhancements to Failover Clustering..... | 56 |
| Chapter 5 Networking | 57 |
| Previous enhancements to networking | 57 |
| Virtual RSS..... | 60 |
| Windows NIC Teaming enhancements..... | 62 |
| NIC Teaming in Windows Server 2012..... | 63 |
| NIC Teaming in Windows Server 2012 R2 | 63 |
| IPAM enhancements | 64 |
| IPAM in Windows Server 2012 | 66 |
| IPAM in Windows Server 2012 R2..... | 66 |
| Hyper-V Network Virtualization enhancements..... | 71 |
| How Hyper-V Network Virtualization works..... | 72 |
| Hyper-V Network Virtualization enhancements in Windows Server 2012 R2 | 74 |
| There's more!..... | 78 |

Chapter 6 Other enhancements 79

IIS 8.5..... 79

RDS enhancements 85

Group Policy enhancements 90

Workplace Join..... 95

Coming soon! 96

Appendix 97

Introduction

This book is intended to provide you with a “first look” at the new features and enhancements coming in Windows Server 2012 R2. Because the book is based on the Preview release of this platform, it’s possible some information might change between now and release-to-manufacturing (RTM) later this year. However, since Windows Server 2012 R2 is such an important (and exciting!) new version of the Windows Server platform, we wanted to get this information into your hands as soon as possible. Later, as RTM approaches, we’ll be publishing an updated and expanded edition of this book that looks at more features and goes into greater depth than we have time or space to do here.

The intended audience for this book is IT pros who deploy, manage, and maintain Windows Server workloads in data center, private cloud, and hosting provider environments. We assume that you are at least somewhat familiar with the features and capabilities of the current platform—Windows Server 2012. If you are not familiar with all the new features and enhancements Microsoft introduced previously in Windows Server 2012, we recommend that you first read *Introducing Windows Server 2012 RTM Edition* (Microsoft Press, 2012). This e-book is available as a free download from Microsoft in three formats:

- PDF from <http://go.microsoft.com/fwlink/?Linkid=251464>
- EPUB from <http://go.microsoft.com/fwlink/?Linkid=251572>
- MOBI from <http://go.microsoft.com/fwlink/?Linkid=251573>

You can also order a Print On Demand (POD) copy of this title from O’Reilly Media at <http://shop.oreilly.com/product/0790145372536.do>.

Acknowledgments

The following individuals at Microsoft have freely contributed their time and expertise in helping ensure that the content of this book is as accurate as possible:

- Aanand Ramachandran
- Adam Carter
- Ben Armstrong
- Bryan Matthew
- CJ Williams
- Elden Christensen
- Erez Benari
- Gabriel Silva
- Jeff Woolsey

- John Savill
 - Jose Barreto
 - Matthew John
 - Raghavendran Gururajan
 - Shivam Garg
 - Symon Perriman
 - Vijay Tandra Sistla
 - Vijay Tewari
- We apologize if anyone was forgotten!

Errata & book support

We've made every effort to ensure the accuracy of this content and its companion content. Any errors that have been reported since this content was published are listed on our Microsoft Press site at oreilly.com:

<http://aka.ms/IntroWinServ2012R2Preview/errata>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at mspinput@microsoft.com.

Please note that product support for Microsoft software is not offered through the addresses above.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://aka.ms/tellpress>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: <http://twitter.com/MicrosoftPress>.

Cloud OS

This chapter introduces the Preview release of Windows Server 2012 R2 that is at the heart of Microsoft's revolutionary new Cloud OS platform. The chapter describes five key areas Microsoft focused on when developing Windows Server 2012 R2 and sets the stage for the discussion of the new features and enhancements in Windows Server 2012 R2 that follow in the remaining chapters of this book.

The big picture

Information Technology (IT) is in the midst of a time of rapid change. More and more businesses are seeing cloud computing as a viable option for hosting their applications, services, and data. Some businesses have already implemented private clouds within their own data centers or have begun utilizing cloud services offered by hosting providers. Other businesses are in the process of evaluating the possible benefits they can reap from cloud availability, scalability, mobility, and agility. And for various reasons, some businesses are still skeptical of whether cloud computing is right for them.

But clearly Microsoft isn't skeptical. In fact, Microsoft is fully committed to the cloud as the computing paradigm of the future. Nowhere is this more obvious than in what's coming with this upcoming release of Windows Server. Microsoft firmly believes that cloud computing isn't a trend but rather a golden opportunity for businesses. Why is that?

Because businesses need to become agile in order to survive in today's competitive landscape. And to have an agile business, you need to build your applications and services on a highly available and elastic development platform. They need a uniform model for application lifecycle management with common frameworks across their physical infrastructure, virtual infrastructure, and the cloud. They need a highly scalable, secure identity solution they can use for managing their computing, networking, and storage assets both on-premises and in the cloud. They need to be able to process, store, and transfer huge amounts of data and perform analytics quickly and easily. And they need to be able to do all this in a cost-effective manner.

In other words, what they need is a cloud-optimized business. And that's what Microsoft intends to deliver with their current product release cycle. Because for the first time in their history, Microsoft has synchronized the development cycles of three major platforms:

- **Windows Server** A proven, enterprise-class platform that forms the foundation for building cloud solutions.

- **System Center** An integrated platform that provides a common management experience across private, hosted, and public clouds.
- **Windows Azure** An open and flexible cloud platform for building, deploying, and managing applications and workloads hosted on a global network of Microsoft-managed data centers.

Together, these three platforms comprise Microsoft’s vision for a Cloud OS, as shown in Figure 1-1. This book only focuses on one portion of this Cloud OS, namely Windows Server 2012 R2. It’s a key portion however, because it forms the foundation for businesses to be able to run their applications in private clouds, with service providers, or in the Windows Azure public cloud.

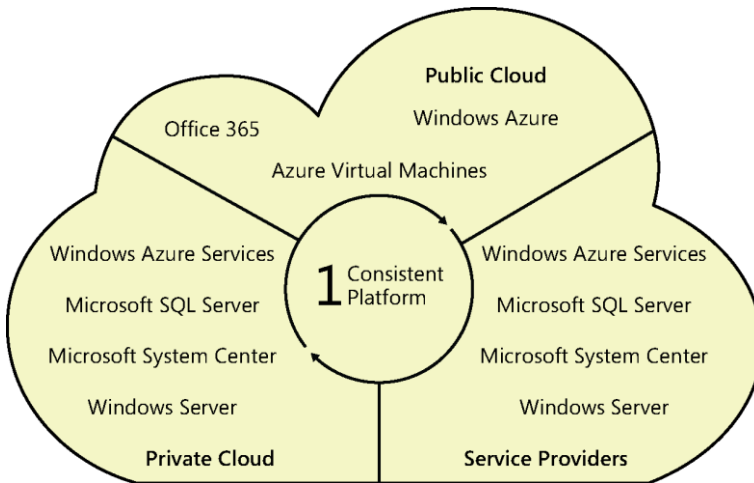


FIGURE 1-1 Microsoft thinks about the cloud in three parts.

Journey to the Cloud OS

To better understand Microsoft’s vision for a Cloud OS, start by thinking about how IT has traditionally managed server workloads. In the early days of Windows Server, you deployed and managed lots of physical servers on-premises. Each server had to be individually managed, and this meant performing tasks like configuring storage for them, configuring networking, tuning performance, and so on. Lots of servers meant lots of tasks to perform, and while scripting could automate many of these tasks, such solutions were typically inflexible and difficult to maintain.

Then along came virtualization, and suddenly you saw you could save money by retiring physical servers after migrating their workloads onto virtualization hosts. But the management paradigm stayed the same, for instead of managing lots of physical servers, you were now managing lots of virtual machines. But proliferation is proliferation whether it’s in the physical

or virtual realm, and managing thousands of individual virtual machines can be just as challenging as managing physical machines.

Then the concept of cloud computing arrived—with its promises of rapid elasticity, resource pooling, and on-demand self-service. Now, if a business wants to maintain control over its IT resources, it can implement a private cloud solution on-premises using Windows Server and System Center. If scalability is the issue, the business can opt for running its applications, services, or virtual machines in Windows Azure. And if reach and customization are important, the business can use the services of a cloud hosting service provider. Each of these approaches are equally valid, and it's up to the business to decide which to choose based on their needs and constraints.

From Microsoft's perspective, these three approaches (private cloud, service providers, and Windows Azure) are really one and comprise one consistent platform: the Cloud OS. Windows Server forms the foundation; System Center provides the management capability; and Windows Azure delivers the solutions. In other words, cloud is not just something that happens out there; it happens wherever and whenever you need it to optimize your business. That's what Microsoft means by cloud.

For example, do you need Active Directory? You can deploy it on-premises using Windows Server. But Active Directory is already waiting for you in Windows Azure. And with Windows Server 2012 R2 you can even virtualize domain controllers and host them in a service provider's cloud. The choice is yours.

Microsoft wants you to have the choice to implement the cloud computing model that best meets the needs of your business. And the Cloud OS—Windows Server, System Center, and Windows Azure—delivers that kind of choice to customers. Windows Server 2012 R2 is the foundation for all this, and that's what this book is about.

Let's begin!

In the chapters that follow we're going to examine what's new and enhanced in this Preview release of Windows Server 2012 R2. Because virtualization is at the heart of how cloud computing works, we're going to start by looking at Hyper-V first. Let's begin!

Hyper-V

Hyper-V virtualization represents the foundation of Microsoft's vision for the cloud operating system. Storage and networking are the walls that help support your virtualization infrastructure. Then, on top of everything, comes management and automation. Together, these different functionalities enable a wide range of cloud solutions that can meet the needs of any business.

But the bottom line is that virtualization is at the core of everything in most IT shops nowadays. For example, when IT wants to deploy a new workload (for example a Microsoft SQL Server machine) the common way of doing this (and it's really a best practice) is to virtualize the workload first instead of deploying the workload on a physical server. As a second example, when IT wants to deploy a new desktop image, the common practice is to create the image in a Hyper-V virtual environment before deploying it onto physical desktop computers.

Windows Server 2012 R2 builds upon the improvements added earlier in Hyper-V on Windows Server 2012 and adds new features and functionality that can deliver greater gains in performance, availability, and agility. This chapter examines what's new in this latest version of Hyper-V but first we'll briefly review what was previously introduced in Hyper-V on Windows Server 2012.

Previous enhancements to Hyper-V

A lot of powerful new features and capabilities were added to Hyper-V in the previous version of Windows Server, and space doesn't allow us to go into detail concerning each of them. As a quick summary however, the following enhancements could be characterized as some of the more significant improvements in the platform:

- **Increased scalability and resiliency** Hyper-V hosts running Windows Server 2012 supported up to 320 logical processors and 4 terabytes (TB) of memory, and virtual machines running on these hosts could be configured with 64 virtual processors and 1 TB of memory.
- **New live migration options** Beginning with Windows Server 2012 you could perform a live migration in a nonclustered environment, and could perform multiple live migrations simultaneously utilizing higher networks bandwidths.

- **Storage migration** Hyper-V in Windows Server 2012 allowed you to move the virtual hard disks used by a virtual machine to different physical storage while the virtual machine remained running.
- **Virtual machines on file shares** Hyper-V in Windows Server 2012 supported using Server Message Block 3.0 (SMB 3.0) file shares as storage for virtual machines. This meant you could store your virtual machine files on a cost-efficient Scale-Out File Server running Windows Server 2012 instead of buying an expensive storage area network (SAN) for this purpose.
- **Extensible virtual switch** Hyper-V on Windows Server 2012 included a new extensible virtual switch that provided an open framework to allow third parties to add new functionality such as packet monitoring, forwarding, and filtering into the virtual switch.
- **Windows PowerShell module** Hyper-V in Windows Server 2012 included a Windows PowerShell module for Hyper-V that provided more than 160 cmdlets for automating Hyper-V management tasks.
- **VHDX format** Hyper-V in Windows Server 2012 included a new virtual hard disk format called VHDX that supported up to 64 TB of storage. The VHDX format also provided built-in protection from corruption stemming from power failures and resisted performance degradation when using some large-sector physical disks.
- **Reclaiming snapshot storage** With Hyper-V in Windows Server 2012, when a virtual machine snapshot was deleted, the storage space that the snapshot consumed before being deleted was made available while the virtual machine was running. This meant that you no longer needed to shut down, turn off, or put the virtual machine into a saved state to recover the storage space. And even more importantly for production environments, differencing disks are now merged with the parent while the virtual machine is running.
- **Improved virtual machine import** The virtual machine import process in Hyper-V in Windows Server 2012 improved to help resolve configuration problems that might otherwise prevent a virtual machine from being imported. In addition, you could import a virtual machine by copying its files manually instead of having to export the virtual machine first.
- **Dynamic Memory improvements** Dynamic Memory was improved in Hyper-V in Windows Server 2012 to include support for configuring minimum memory. In addition, Smart Paging, a new memory management mechanism, was introduced to provide a reliable restart experience for virtual machines configured with less minimum memory than startup memory.

- **Single-root I/O virtualization (SR-IOV)** Hyper-V in Windows Server 2012 allowed you to assign network adapters that supported SR-IOV directly to virtual machines running on the host. SR-IOV maximized network throughput while minimizing network latency and CPU overhead needed for processing network traffic.
- **Virtual Fibre Channel** Hyper-V in Windows Server 2012 allowed you to connect directly to Fibre Channel storage from within the guest operating system that runs in a virtual machine. This allowed you to virtualize workloads and applications that require direct access to Fibre Channel–based storage. It also made guest clustering (clustering directly within the guest operating system) possible when using Fibre Channel–based storage.
- **Hyper-V Replica** Hyper-V in Windows Server 2012 allowed you to replicate virtual machines between storage systems, clusters, and data centers in two sites to provide business continuity and disaster recovery.

Now that we’ve reviewed the Hyper-V improvements introduced previously in Windows Server 2012, let’s move on and examine some of the new capabilities added to Hyper-V in Windows Server 2012 R2.

Generation 2 virtual machines

One of the key ways that Windows Server 2012 R2 advances the Hyper-V virtualization platform is in its support for a new generation of virtual machines. Microsoft refers to these as “Generation 2” virtual machines, and they have the key following characteristics:

- **UEFI-based** Beginning with Windows 8 and Windows Server 2012, Microsoft Windows now supports the Secure Boot feature of the Unified Extensible Firmware Interface (UEFI). This means that UEFI is now part of the Windows 8 and Windows Server 2012 boot architecture, and it replaces the Basic Input/Output System (BIOS) firmware interface used by previous versions of Windows for initiating the boot process. Generation 2 virtual machines comply with the UEFI Secure Boot standard and enable virtual machines to use Secure Boot.
- **Legacy free** In previous versions of Hyper-V, virtual machines used a standard set of emulated hardware devices to ensure compatibility running all versions of Windows. These emulated devices include an AMI BIOS, Intel 440BX chipset motherboard, S3 Trio graphics display adapter, Intel/DEC 21140 network adapter, and so on. With Generation 2 virtual machines, many of these emulated devices have now been removed and replaced with synthetic drivers and software-based devices as summarized in Table 2-1.
- **SCSI boot** Virtual machines in previous versions of Hyper-V needed to boot from integrated development environment (IDE) disks (virtual disks attached to the virtual machine using the IDE controller). Beginning with Windows Server 2012 R2, however,

Generation 2 virtual machines can now boot directly from SCSI disks (virtual disks attached to the virtual machine using the SCSI controller). In fact, Generation 2 virtual machines don't even have an IDE controller!

- **Faster deployment** Network-based installation of a guest operating system onto a Generation 2 virtual machine is significantly faster than for the previous generation of Hyper-V virtual machines for two reasons. First, the Legacy Network Adapter device is no longer required (or even supported) by Generation 2 virtual machines. And second, the SCSI controller performs much better than the legacy IDE controller in the previous generation of virtual machines. The result is that installing a supported guest operating system in a Generation 2 virtual machine takes only about half the time as installing the same guest operating system in a previous generation virtual machine.

TABLE 2-1 Hardware Device Changes in Generation 2 Virtual Machines

| LEGACY DEVICES REMOVED | REPLACEMENT DEVICES | ENHANCEMENTS |
|---|-----------------------------|--|
| IDE controller | Virtual SCSI controller | Boot from VHDX (64 TB max size, online resize) |
| IDE CD-ROM | Virtual SCSI CD-ROM | Hot add/remove |
| Legacy BIOS | UEFI firmware | Secure Boot |
| Legacy NIC | Synthetic NIC | Network boot with IPv4 & IPv6 |
| Floppy & DMA Controller | No floppy support | |
| UART (COM Ports) | Optional UART for debugging | Faster and more reliable |
| i8042 keyboard controller | Software-based input | No emulation – reduced resources |
| PS/2 keyboard | Software-based keyboard | No emulation – reduced resources |
| PS/2 mouse | Software-based mouse | No emulation – reduced resources |
| S3 video | Software-based video | No emulation – reduced resources |
| PCI Bus | VMBus | |
| Programmable Interrupt Controller (PIC) | No longer required | |
| Programmable Interrupt Timer (PIT) | No longer required | |
| Super I/O device | No longer required | |

Because of all these hardware changes, Generation 2 virtual machines only support the following versions of Windows as guest operating systems:

- 64-bit versions of Windows 8 and Windows Server 2012
- 64-bit versions of Windows 8.1 and Windows Server 2012 R2

As Figure 2-1 shows, when you create a new virtual machine in Windows Server 2012 R2 using Hyper-V Manager, you now have the option of choosing whether to create a first-generation virtual machine or a Generation 2 virtual machine. You can also specify which type of virtual machine to be created by using the *new -Generation* parameter of the New-VM Windows PowerShell cmdlet in Windows Server 2012 R2.

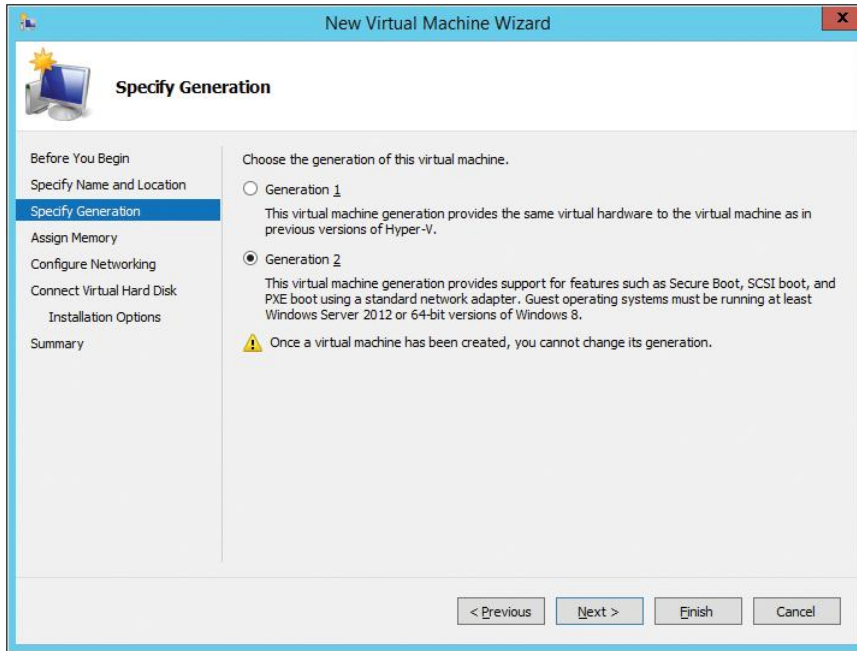


FIGURE 2-1 A Generation 2 virtual machine can be created using the New Virtual Machine wizard.

Once the Generation 2 virtual machine has Windows Server 2012 R2 installed as the guest operating system, opening Device Manager shows the various synthetic and software-based devices attached to the VMBus. Note that unlike first-generation virtual machines, there is no PCI-to-ISA bridge running in ISA mode, no PS/2 keyboard, no PS/2 mouse, no COM ports, and so on. Figure 2-2 compares Device Manager for Generation 1 virtual machines (left) with Device Manager for Generation 2 virtual machines (right).

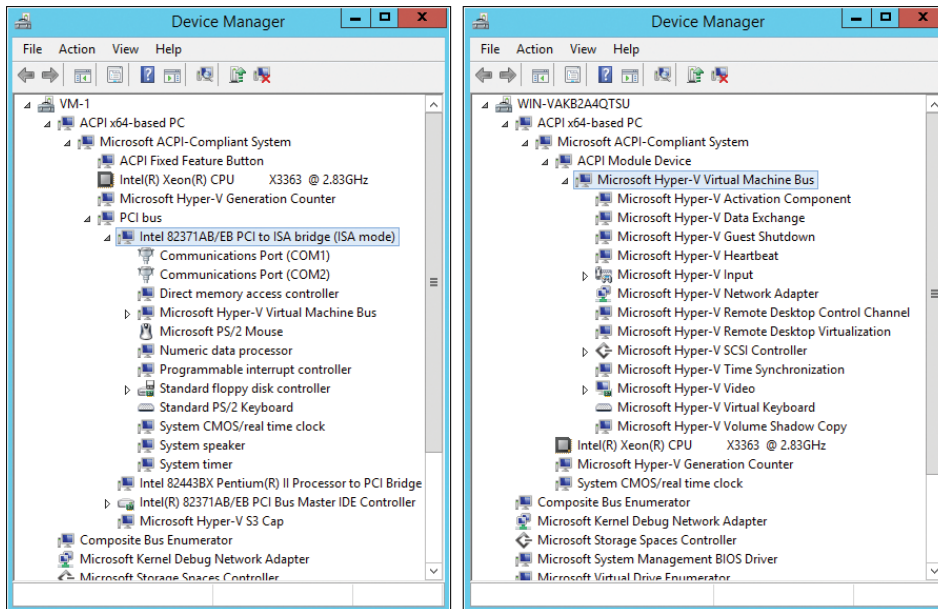


FIGURE 2-2 A comparison of what Device Manager shows for Generation 1 (left) and 2 (right) virtual machines.

Automatic VM activation

Starting way back with Windows Server 2003 R2 with Service Pack 2, the Datacenter edition of Windows Server has provided unlimited virtualization rights for servers to allow organizations to deploy as many virtual machines as they need in their environments. But until now this benefit has come with the cost of the administrative overhead of deploying a key management infrastructure for licensing and activating these virtual machines.

Beginning with Windows Server 2012 R2, when a new virtual machine with Windows Server 2012 R2 installed as the guest operating system boots up for the first time on a Hyper-V host running a Datacenter edition of Windows Server, the new virtual machine checks to see whether the host machine has been activated, and if it is activated then the virtual machine automatically activates itself as well. This new automatic activation capability removes a major customer pain point by greatly reducing the time and effort needed by large enterprises and hosts for managing licensing and activation of large numbers of virtual machines in their environment.

Regardless of whether your Hyper-V hosts are OEM machines or are running a volume-licensed version of Windows Server activated using Key Management Service (KMS) or Multiple Activation Key (MAK), if the host machine is running Datacenter edition and is activated, then all virtual machines running Windows Server 2012 R2 as a guest operating system are

automatically activated. And this is also completely secure with respect to your existing key management infrastructure since no keys are used to activate the virtual machines. So if you should copy or move one of your virtual machines to someone else's environment, for example, as part of demonstration purposes, your keys won't be exposed. Of course, the other environment must also be using hosts running an activated copy of a Datacenter edition of Windows Server.

Remote access over VMBus

Virtual Machine Connection (VM Connect) is a tool that you use to connect to a virtual machine running on a Hyper-V host. VM Connect is installed on the host when you add the Hyper-V role to your server. Specifically, if the server is running Windows Server 2012, then the VM Connect is installed with the Hyper-V role provided that either the server with a GUI installation option has been selected or the Minimal Server Interface option has been configured. (VM Connect is not available on Windows Server Core installations of Windows Server.)

The purpose of VM Connect is to enable Hyper-V administrators to directly interact with the guest operating system in a virtual machine from the local console of the host. While management of most virtual machines is typically performed remotely using either Remote Desktop Connection (RDC) or Windows PowerShell, there are times when you might need to work with a virtual machine directly on the host, for example when the virtual network adapter of a virtual machine stops functioning. In such cases, you can use Hyper-V Manager on the host and to connect to the virtual machine and open its desktop within the VM Connect window to configure or troubleshoot the virtual machine and its guest operating system even if the virtual machine has no connectivity with your network.

The way that VM Connect works in Windows Server 2012 and earlier is to present you with a bitmap image of the desktop of a virtual machine's guest operating system, which is generated by an emulated video card in the virtual machine. This bitmap image is updated in real time so you can see configuration changes as they happen. VM Connect also provides you with emulated keyboard and mouse devices in the virtual machine so you can directly control the desktop of the guest operating system. Because VM Connect in Windows Server 2012 and earlier uses bitmap images, certain limitations exist in how you can use VM Connect to interact with the guest operating system. For example, you can copy and paste text between the host machine's desktop and the desktop of the guest operating system, but you can't copy/paste images or files between them.

Beginning with Windows Server 2012 R2, however, VM Connect no longer connects you to the guest operating system using an emulated video card, keyboard, and mouse in the virtual machine. Instead, VM Connect uses Remote Desktop Services (RDS) in the guest operating

system of the virtual machine to provide the full RDS experience when you use it to connect to the virtual machine (see Figure 2-3). The result is an enhanced experience that enables you to:

- Copy/paste files between the desktop of the host and the desktop of the guest operating system.
- Redirect audio on the virtual machine to the host.
- Enable the guest operating system to use smart cards attached to the host.
- Enable the guest operating system to access any USB device attached to the host.

All of this is possible even if the virtual machine is not connected to the network. And you can do it with hosts you are managing remotely using Hyper-V Manager or Windows PowerShell. You don't have to be logged on interactively to the host to experience all this new VM Connect functionality.

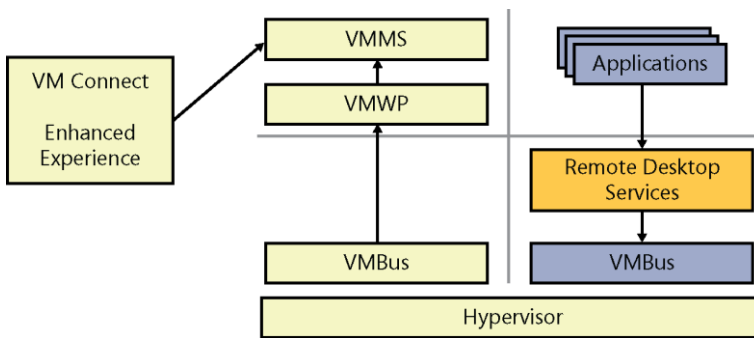


FIGURE 2-3 VM Connect now connects to the virtual machine using RDS in the guest operating system.

Cross-version live migration

Windows Server 2012 R2 also includes several significant improvements to live migration that can benefit organizations deploying private cloud solutions built with Windows Server and System Center. Live migration was introduced in Windows Server 2008 R2 to provide a high-availability solution for virtual machines running on Hyper-V hosts. Live migration uses the Failover Clustering feature to allow running virtual machines to be moved between cluster nodes without perceived downtime or loss of network connection. Live migration provides the benefit of increased agility by allowing you to move running virtual machines to the best host for improving performance, achieving better scaling, or ensuring optimal workload consolidation. Live migration also helps increase productivity and reduce cost by allowing you to service your host machines without interruption or downtime for your virtualized workloads.

The ability to perform cross-version live migration between Hyper-V hosts running Windows Server 2012 and Windows Server 2012 R2 is new in Windows Server 2012 R2. Cross-version live

migration can be performed using any of the live migration options supported by Windows Server 2012 including:

- Live migration on a failover cluster
- Live migration between failover clusters
- Live migration using a Scale-Out File Server that supports SMB 3.0
- Shared nothing live migration where no shared storage is used for the virtual machines

For organizations that have already begun deploying a private cloud solution based on Windows Server 2012, cross-version live migration means that you will be able to upgrade your private cloud solution from Windows Server 2012 to Windows Server 2012 R2 with zero downtime for the workloads running on your virtual machines. If you can tolerate a brief downtime window for your workloads, you can even choose to perform an in-place upgrade of your existing Hyper-V hosts from Windows Server 2012 to Windows Server 2012 R2.

And unlike previous versions of Windows Server, you don't have to perform a lot of preparatory actions before performing an in-place upgrade of your Hyper-V hosts. For example, you previously had to do things like turn off the virtual machines running on the host, and you also had to delete any snapshots and saved states of the virtual machines. When performing an in-place upgrade of a Hyper-V host from Windows Server 2012 to Windows Server 2012 R2, however, none of these preparatory steps are required and all of the virtual machine's snapshots and saved states are preserved.

Faster live migration

Live migration is also significantly faster in Windows Server 2012 R2 Hyper-V for two reasons. First, a new ability to compress live migration traffic can be used to reduce the amount of data that needs to be sent over the network during a live migration. This live migration compression capability is enabled by default for Hyper-V in Windows Server 2012 R2 and can often halve the time it takes to perform a live migration in a Windows Server 2012 R2 Hyper-V environment, depending on the processing resources available on the host machines for performing the compression operation.

Second, live migration can be faster in Windows Server 2012 R2 Hyper-V because of the ability to use network adapters that have Remote Direct Memory Access (RDMA) together with the SMB Direct and SMB Multichannel features of SMB 3.0. RDMA is a networking technology that enables high-throughput, low-latency communication that minimizes CPU usage on the computers using this technology. RDMA is an industry standard protocol defined in RFC 5040 that works by moving data directly between the memory of the computers involved, bypassing the operating systems on these machines. Examples of types of RDMA-capable network

adapter hardware include Infiniband (IB), Internet Wide Area RDMA Protocol (iWARP), and RDMA over Converged Ethernet (RoCE).

SMB Direct, which is short for SMB over Remote Direct Memory Access (SMB over RDMA), is a feature of SMB 3.0 that supports the use of RDMA-capable network adapters. By using SMB Direct for example, a Hyper-V host is able to access data on a remote SMB 3.0 file server (called a Scale-Out File Server) as quickly and easily as if the data was on local storage on the Hyper-V host. SMB Direct is available only on the Windows Server platform and was introduced in Windows Server 2012. SMB Direct requires that the SMB client and SMB server both support SMB 3.0.

SMB Multichannel is another feature of SMB 3.0 that enables the aggregation of network bandwidth and provides network fault tolerance whenever multiple paths are available between an SMB 3.0 client and an SMB 3.0 server. SMB Multichannel thus enables server applications to take full advantage of all available network bandwidth and be resilient to a network failure. SMB Multichannel is also the feature that is responsible for detecting the RDMA capabilities of network adapters to enable the use of SMB Direct. Once SMB Multichannel has determined that a network adapter is RDMA-capable, it creates multiple RDMA connections (two per interface) for that session. SMB Multichannel is also available only on the Windows Server platform and was introduced in Windows Server 2012, and it requires that the SMB client and SMB server both support SMB 3.0.

When a live migration is performed with virtual machines running on Hyper-V hosts that have RDMA-capable network adapters, SMB Direct and SMB Multichannel enable multiple network interfaces to be used for performing the live migration. This not only results in significantly faster live migrations but also results in less use of processing resources on the hosts as well. This is different from live migration compression, which utilizes available processor resources on the host to reduce the network load involved in transferring the compressed virtual machine memory across the network.

When would you use live migration compression? A typical scenario would be when the primary constraining factor limiting the speed of live migration is your network bandwidth but your Hyper-V hosts are not under heavy load as regards processing cycles. When would you use live migration using SMB Direct and SMB Multichannel? A scenario here would be when the primary constraining factor is high processor utilization on your host machines while you have lots of bandwidth available on your network. In general, if the network you are using for performing your live migration is 10 GbE or slower, you probably want to use the compression approach. If your network is faster than 10 GbE, then you should probably be using RDMA-capable network adapters so you can take advantage of the SMB Direct and SMB Multichannel capabilities of Windows Server 2012 and later.

Online VHDX resize

Another new capability of Hyper-V in Windows Server 2012 R2 is the ability to increase or decrease the size of a virtual hard disk attached to a virtual machine while that virtual machine is still running on the host. This means that if the workload running on a virtual machine should require more space, you can expand the virtual hard disk without interrupting any applications accessing the workload. And if you want to reallocate storage space from one virtual machine to another, you can shrink the virtual hard disk attached to the first virtual machine (provided that there is sufficient unpartitioned space on the disk) to free up space for expanding the disk on the second machine.

Online resizing of virtual hard disks requires that these disks be using the newer VHDX virtual hard disk format first introduced in Windows Server 2012. VHDX was designed to address the growing technological demands of today's enterprises and provides greater storage capacity, built-in data protection, and support for large-sector hard disk drives. In addition, online resizing requires that the virtual disk be attached to the virtual machine's SCSI bus.

For example, the following steps use Hyper-V Manager to expand the size of a running virtual machine:

1. In Hyper-V Manager, right-click the virtual machine and select Settings.
2. In the Settings dialog for the virtual machine, click the Hard Drive node under SCSI Controller for the virtual hard disk you want to expand, then click the Edit button to launch the Edit Virtual Hard Disk Wizard.
3. Select the Expand option on the Choose Action page, click Next, type the new size you want the virtual hard disk to have (see Figure 2-4), and then click Next followed by Finish.

Once you've expanded a virtual hard disk, the option to shrink it will be displayed next time you use the Edit Virtual Hard Disk Wizard.

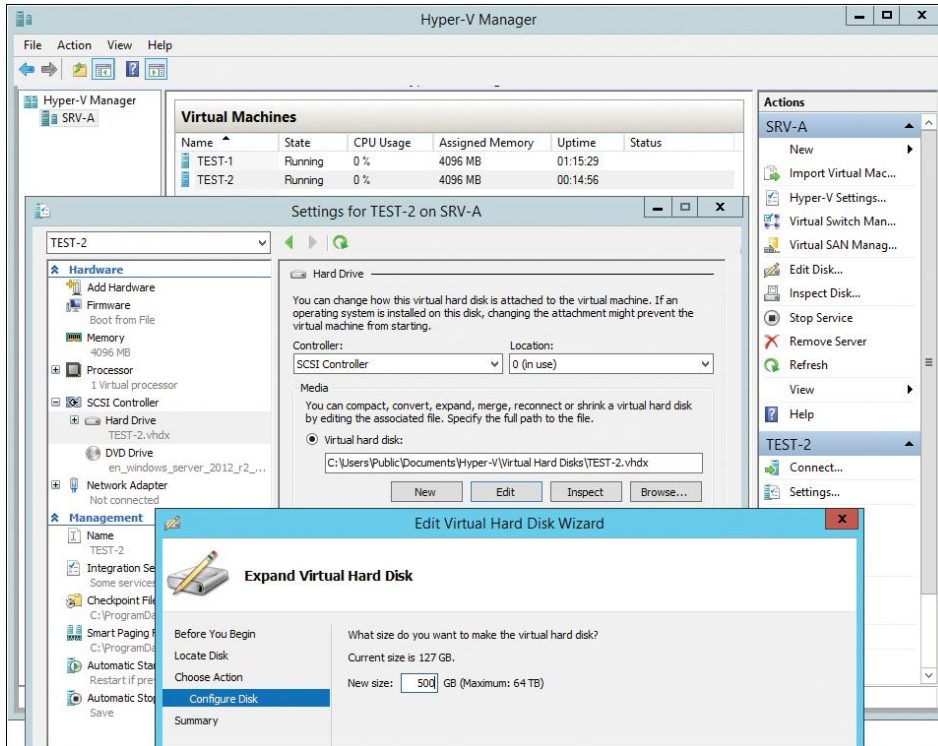


FIGURE 2-4 Virtual hard disks of running virtual machines can now be resized on Windows Server 2012 R2.

Live export

Not only can you now resize virtual hard disks attached to virtual machines while the virtual machines are running, you can also export a running virtual machine. You couldn't do this with virtual machines on Windows Server 2012 or earlier versions of Windows Server. However, with Windows Server 2012 R2 you can use Hyper-V Manager to export a complete copy of a running virtual machine or to export any snapshot of a running virtual machine. And you can use Virtual Machine Manager 2012 R2 to clone a running virtual machine, which basically involves exporting and then importing a virtual machine to create a new virtual machine that is based on the existing virtual machine. And you can even export snapshots while a virtual machine is running.

One scenario where live export can be helpful is when a running virtual machine in your environment begins to exhibit some instability but is still performing its expected workload. Previously, you had to choose between the lesser of two evils:

- Stop the virtual machine or take it offline and try to troubleshoot the problem. Unfortunately while the virtual machine is stopped or offline, its workload will no longer be available to users, and this can result in loss of either business or productivity.

- Let the virtual machine continue to run and hope it doesn't fail. This approach allows the virtual machine's workload to continue to be available, but instability often ends up with the application or guest operating system crashing, which means a probable interruption in workload will occur. Once again, this is likely to result in loss of either business or productivity.

With live export, however, you can now clone a copy of your unstable virtual machine without shutting the virtual machine down (see Figure 2-5). You can then let your production virtual machine continue to run while you perform troubleshooting steps on the cloned workload to try and see if you can resolve the issue causing the instability. Once you determine how to fix the problem by working with the cloned virtual machine, you might be able to repair your production virtual machine without needing to reboot the guest operating system or restart its running applications, depending on the issue causing the instability.

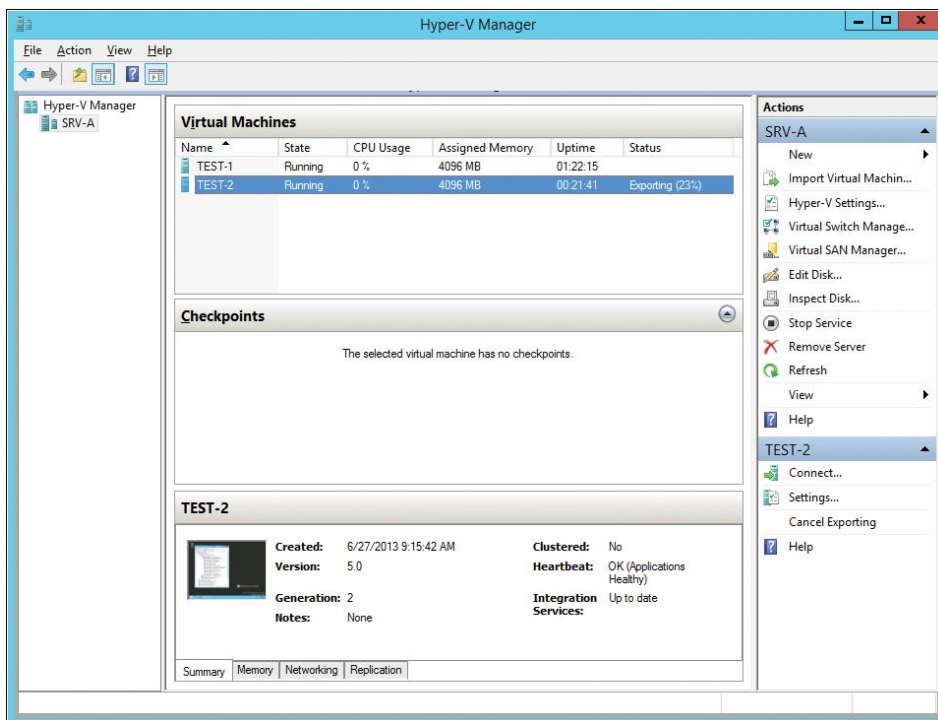


FIGURE 2-5 Running virtual machines can now be exported or cloned on Windows Server 2012 R2.

More robust Linux support

Hyper-V has supported installing and running various Linux distros or “flavors” in virtual machines for some time now. Linux guest support in Hyper-V is especially desired by hosting providers who often like to provide their customers with a wide range of platform options for running their web applications and services. Linux (and UNIX) support in Hyper-V is also important in the enterprise space where heterogeneous IT solutions are generally the norm.

Because of the needs of these customer segments, Microsoft envisions Hyper-V virtualization as “cross-platform from the metal up” and supports a wide range of Linux distros, as shown in Table 2-2, by providing Linux Integration Services (LIS) for specific versions of popular distros. Microsoft also includes robust Linux/UNIX capabilities across the entire System Center family of products and also in Windows Azure offerings as well. Linux/UNIX support is an integral part of all of these platforms and is not merely an extension of Windows-centric functionality.

TABLE 2-2 Current Availability of Linux Integration Services for Hyper-V in Windows Server 2012 R2

| DISTRO | VERSION | LIS AVAILABILITY |
|------------------------------|---------------------|---------------------------------------|
| Red Hat Enterprise Linux | 5.7, 5.8, 6.0-6.3 | Download LIS from Microsoft |
| | 5.9, 6.4 | LIS built-in and certified by Red Hat |
| SUSE Linux Enterprise Server | 11 SP2 | LIS built-in |
| CentOS | 5.7, 5.8, 6.0-6.3 | Download LIS from Microsoft |
| | 5.9, 6.4 | LIS built-in |
| Ubuntu Server | 12.04, 12.10, 13.04 | LIS built-in |
| Debian | 7.0 | LIS built-in |

As part of Microsoft’s continuing commitment to making Hyper-V the best all-around virtual platform for hosting providers, Linux support for Hyper-V in Windows Server 2012 R2 has now been enhanced in the following ways:

- **Improved video** A virtualized video driver is now included for Linux virtual machines to provide an enhanced video experience with better mouse support.
- **Dynamic Memory** Dynamic Memory is now fully supported for Linux virtual machines, including both hot-add and remove functionality. This means you can now run Windows and Linux virtual machines side-by-side on the same host machine while using Dynamic Memory to ensure fair allocation of memory resources to each virtual machine on the host.
- **Online VHDX resize** Virtual hard disks attached to Linux virtual machines can be resized while the virtual machine is running.

- **Online backup** You can now back up running Linux virtual machines to Windows Azure using the Windows Azure Online Backup capabilities of the in-box Windows Server Backup utility, System Center Data Protection Manager, or any third-party backup solution that supports backing up Hyper-V virtual machines.

Hyper-V Replica enhancements

In the short time that Windows Server 2012 has been released, Hyper-V Replica has proven to be one of its most popular features. Hyper-V Replica provides asynchronous replication of virtual machines between two Hyper-V hosts. It's easy to configure and doesn't need either shared storage or any particular storage hardware. Any server workload that you can virtualize on Hyper-V can be replicated using this capability, and replication is encrypted during transmission and works over any IP-based network.

You can use Hyper-V Replica with standalone Hyper-V hosts, failover clusters of hosts, or a mixture of these environments. The host machines can either be physically colocated or widely separated geographically. And they don't need to be in the same domain or even domain-joined at all.

Hyper-V Replica is an ideal technology for organizations that want to add support for disaster recovery to their Hyper-V environment to ensure business continuity. For example, you could use it to provide disaster recovery support for the branch offices by replicating their virtual machines to hosts at the head office. Another possible scenario would be to have a hosting provider set up a Replica server at their data center to receive replication data from a number of Hyper-V hosts running virtualized workloads on the premises of customers.

In Hyper-V Replica in Windows Server 2012 R2, greater control over the frequency at which data is replicated between hosts is a new feature. In Windows Server 2012, the replication frequency was fixed at every five minutes. Some customers provided feedback that this was not frequent enough for their environment, while others requested the option of performing replication less frequently. So now, as Figure 2-6 shows, there are two new replication frequencies you can choose from besides the default one of five minutes when you enable replication for a server:

- **30 seconds** Choosing this option means that the host in the replica site will never be more than a minute behind the host in the primary site. This option was provided in Windows Server 2012 R2 so that Hyper-V Replica could be used as an alternative to more expensive SAN solutions that have a similar low latency. Organizations that simply need to replicate data as quickly as possible, for example between two data centers in the same metropolitan area, might choose this option.

- **15 minutes** This option was provided especially for organizations that wanted to replicate data over networks that had very high latency or low reliability, for example over a satellite link. To ensure that replication would tolerate network outages and succeed in such scenarios, a long replication window like this can now be chosen when you enable replication on a host in Windows Server 2012 R2, and choosing this option means that the host in the replica site will never be more than an hour behind the host in the primary site.

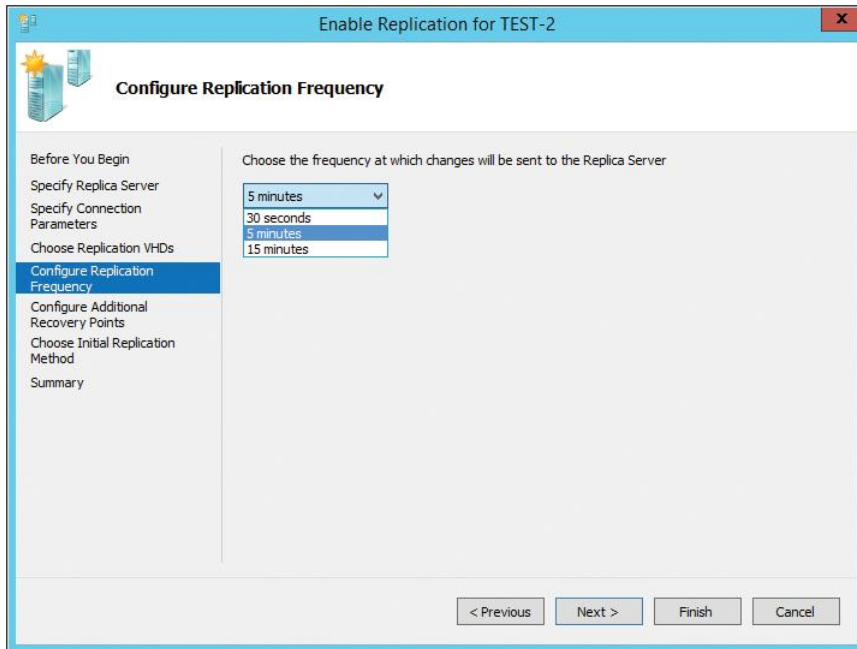


FIGURE 2-6 Hyper-V Replica in Windows Server 2012 R2 now supports three possible replication frequencies.

A second new capability for Hyper-V Replica in Windows Server 2012 R2 is the introduction of extended replication. This allows a chain of replication to be configured between hosts so that, for example, HOSTA automatically replicates to HOSTB, which automatically replicates to HOSTC. As Figure 2-7 shows, you configure extended replication when you enable replication on a host.

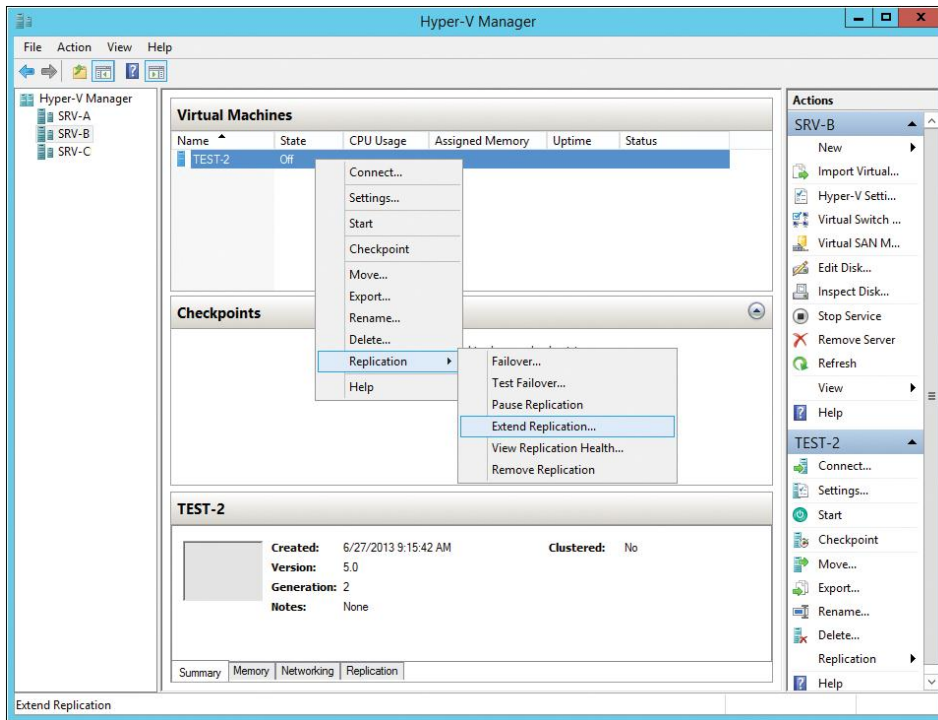


FIGURE 2-7 Hyper-V Replica in Windows Server 2012 R2 now supports extended replication.

One possible use for this new feature would be enterprises that want to do replication both on-premises and to a local hoster. With extended replication, enterprises can do a first-hop replication on-premises and then a second hop offsite, like this:

On-premises host A → On-premises host B → Hosting provider

Another usage scenario might be for hosting providers that provide Hyper-V Replica services to their customers and also want to replicate customer virtual machines to the hoster's backup data center. Extended replication in this scenario would thus be:

Customer site → Primary data center → Secondary data center

These enhancements to Hyper-V Replica in Windows Server 2012 R2 don't just represent new features added to the platform in response to customer requests; they also represent the next steps in Microsoft's vision of offering cloud-scale disaster recoverability solutions based on the Windows Server platform, System Center, and Windows Azure. As Figure 2-8 shows, another key part of this vision is Windows Azure Hyper-V Recovery Manager, a Windows Azure service that provides orchestration of the replication of private clouds managed using System Center Virtual Machine Manager 2012 R2.

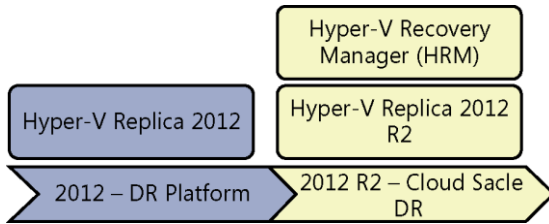


FIGURE 2-8 Hyper-V Replica and Hyper-V Recovery Manager (HRM) are part of Microsoft’s vision for cloud-scale disaster recoverability.

For example, by using Hyper-V Recovery Manager together with VMM 2012 R2, you can replicate your organization’s primary data center to your disaster recovery site as shown in Figure 2-9. Using Hyper-V Recovery Manager, you can enhance your organization’s disaster recovery preparedness by performing failovers of selected virtual machine workloads in your environment to replicate them to your backup site. And the best thing about it is that you can do this at a fraction of the cost of using traditional SAN replication.

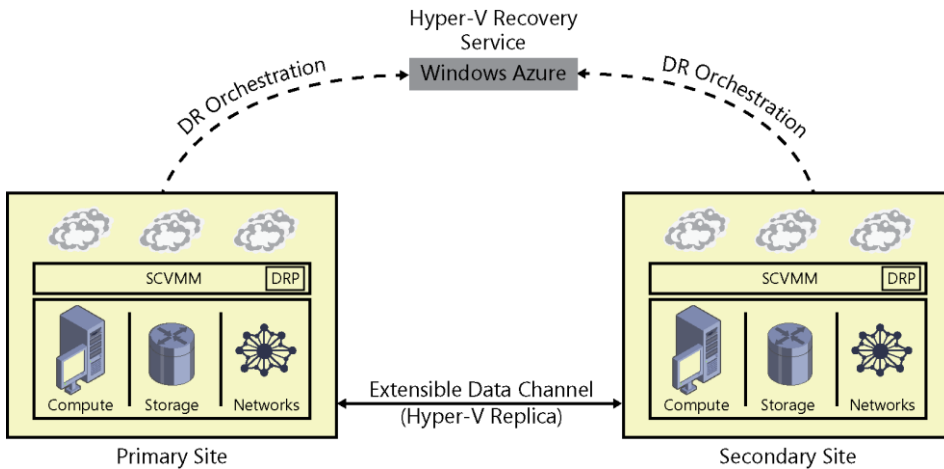


FIGURE 2-9 Hyper-V Recovery Manager lets you replicate your organization’s primary data center to your disaster recovery site using Windows Azure.

There’s more!

Hyper-V in Windows Server 2012 R2 also includes support for Quality of Service (QoS) management for virtual hard disks, which allows you to monitor and restrict the Input/Output Operations Per Second (IOPS) for a virtual hard disk attached to a virtual machine. We’ll talk about that in Chapter 3, “Storage,” since it fits well into the theme of that chapter.

Hyper-V in Windows Server 2012 R2 also now allows guest clustering using shared VHDX files. This new capability is going to be a game changer, especially for hosters who want to maintain separation between their own storage infrastructure and that of their tenants. Since this is related to the topic of clustering, we'll defer discussion of this one until we get to Chapter 4, "Failover Clustering" later in this book.

There are still other enhancements to Hyper-V in Windows Server 2012 R2, but since this is only a "First Look" book we'll leave these until we get more chances to play with the product so we can revise and expand this book when Windows Server 2012 R2 is released to manufacturing and becomes generally available to customers.

Storage

Storage is a key part of any IT infrastructure. For many organizations, storage is also a major cost center that consumes a large portion of the budget of the IT department. Maximizing the performance and efficiency of storage while helping to reduce costs was a major goal of Windows Server 2012, and the numerous new features and capabilities introduced in that platform now provide enterprises with new ways of squeezing the most out of shrinking IT budgets when it comes to storage.

Windows Server 2012 R2 takes these previous enhancements a step further and enables organizations to envision how storage infrastructure can be built and managed. This chapter examines the new storage features in Windows Server 2012 R2, focusing mainly on improvements to file- and block-based storage in the platform. The next chapter will build on this one by examining how failover clustering has been enhanced in Windows Server 2012 R2 and how Hyper-V virtualization can benefit from the improvements to storage and failover clustering. But first we'll briefly review some of the storage enhancements that were previously introduced in Windows Server 2012.

Previous enhancements to storage

A number of new storage capabilities were introduced in the previous version of Windows Server, and space doesn't allow us to describe each of them in detail. But as a quick summary, the following might be characterized as some of the more significant storage improvements in the platform:

Storage Spaces Storage Spaces provided storage virtualization capabilities that allow you to group industry-standard disks (such as Serial ATA or Serial Attached SCSI disks) into storage pools. You could then create virtual disks called "storage spaces" or "spaces" from the available capacity in the pools and provision resilient storage volumes as they were needed. This allowed you to make more efficient use of disk capacity, provision storage quickly and easily without impacting users, delegate the administration of storage, and provide cost-effective storage for business-critical applications that make use of low-cost, commodity-based just-a-bunch-of-disks (JBODs).

SMB 3.0 A new version of the Server Message Block (SMB) protocol, a network file sharing protocol that enables applications on a computer to read and write to files and to request services from server programs on a network, was introduced in Windows Server 2012. These improvements included SMB Direct, SMB Multichannel, SMP Transparent Failover, and other enhancements that enabled new scenarios such as storing Hyper-V virtual machine files

and Microsoft SQL Server database files on file shares on an SMB 3.0 file server (called a Scale-out File Server) instead of having to store these files on local storage, Direct Attached Storage (DAS), or a Storage Area Network (SAN) array.

ReFS The new Resilient File System (ReFs) introduced in Windows Server 2012 provided enhanced integrity, availability, scalability, and error protection for file-based data storage. ReFS supported volume sizes up to 18 exabytes in size and could be especially useful on file servers storing large amounts of data or running disk-intensive applications that require high levels of performance. ReFS in Windows Server 2012 did not support certain NTFS features however, such as disk quotas.

Data deduplication Data deduplication allowed more data to be stored in less space without compromising the integrity or fidelity of the data stored on the volume. It accomplished this by segmenting files into small, variable-sized chunks, identifying any duplicate chunks present, and maintaining only a single copy of each unique chunk of data. One scenario where this was useful was when data was transferred over the WAN to a branch office using the BranchCache feature of Windows Server 2012.

iSCSI Target Server The iSCSI Target Server provides block storage to servers and applications on the network using the Internet SCSI (iSCSI) standard. When combined with other availability technologies in Windows Server 2012, iSCSI Target Server provided continuously available storage that previously required organizations to purchase expensive, high-end SAN arrays.

ODX Offloaded Data Transfer (ODX) functionality in Windows Server 2012 enabled ODX-capable storage arrays to bypass the host computer and directly transfer data within or between compatible storage devices. The result was to minimize latency, maximize array throughput, and reduce resource usage such as CPU and network consumption on the host computer. For example, by using ODX-capable storage arrays accessed via iSCSI, Fibre Channel, or SMB 3.0 file shares, virtual machines stored on the array could be imported and exported much more rapidly than they could without ODX capability being present.

Chkdsk Windows Server 2012 introduced a new Chkdsk model that allowed organizations to confidently deploy large, multiterabyte NTFS file system volumes without worrying about their availability being compromised should file system corruption be detected on them. The new version of Chkdsk ran automatically in the background and actively monitored the health state of the file system volume. Should file system corruption be detected, NTFS now instantaneously self-healed most issues online without requiring Chkdsk to run offline. This means that the amount of time needed for running Chkdsk on multiterabyte data volumes can be reduced from hours to only a few seconds, plus in many scenarios you won't even need to take the disk offline and run Chkdsk on it at all.

Storage management improvements Beginning with Windows Server 2012, you could now use the File and Storage Services role in Server Manager to remotely manage multiple file servers running Windows Server 2012, including their storage pools, volumes, shares, and iSCSI virtual disks, all from a single user interface. You could also use the new Windows PowerShell cmdlets in Windows Server 2012 to automate the same storage management tasks.

Now that we've reviewed the storage improvements introduced previously in Windows Server 2012, let's move on and look at some of the new storage capabilities and enhancements added in Windows Server 2012 R2.

Microsoft's vision for storage

As you can see from the previous section, Windows Server 2012 introduced a lot of new storage features and capabilities to the Windows Server platform. Together with System Center 2012 SP1, Windows Server 2012 provided organizations with a cost-effective solution for building and deploying private clouds using file-based storage access comprised of low-cost commodity storage accessed over a standard Ethernet network.

While the R2 release of Windows Server 2012 adds a number of incremental improvements to both file- and block-based storage and to how storage is managed on the Windows Server platform, it also represents something more. Microsoft's vision and goals with respect to storage for this new release are threefold:

- To greatly reduce the capital and operational storage and available costs for organizations deploying Infrastructure-as-a-Service (IaaS) services for private clouds, hosted clouds, and cloud service providers.
- To disaggregate compute and storage resources so they can be independently managed and scaled at each layer of cloud infrastructure.
- To allow enterprises to build software-defined storage solutions using inexpensive, industry-standard servers, networks, and shared JBOD storage.

With this focus in mind, the incremental improvements to storage capabilities in Windows Server 2012 R2 are designed to specifically target the above three goals.

Building the solution using Windows Server 2012 R2

To understand how Windows Server 2012 R2 can be used to implement the above vision for cloud computing, let's look at an example. Figure 3-1 shows the compute, networking, and storage components of a simple private cloud solution built using the Windows Server platform. You can think of this solution as having four layers as follows:

- **Compute layer** At the top are several Hyper-V hosts joined together in a failover cluster. These hosts use commodity server hardware to provide cost-efficient scale-out capabilities. For example, if the solution needs more processing power to run more workloads running in virtual machines, you can add another commodity server to the Hyper-V cluster. Utilizing the scale-out approach like this is often a more cost-effective solution for organizations than using a scale-up solution that involves only two expensive high-end host machines, where you need to add another processor to each host if you want to run more workloads.

- **Network layer** A low-cost industry-standard Ethernet network is used to connect the Hyper-V cluster that provides compute resources for the solution with the Scale-out File Servers (SoFS) that provide virtualized storage resources for the cloud. This kind of approach can be a lot more cost-effective for many organizations than utilizing a proprietary SAN for their storage layer. That's because you don't need to install expensive host bus adapters (HBAs) in the Hyper-V hosts to enable them to connect to storage volumes (logical unit numbers or LUNs) provisioned on the SAN. This approach is only possible, however, because of new capabilities in version 3.0 of the SMB protocol that was first introduced in Windows Server 2012.
- **Virtualized storage layer** The virtual machines running on the clustered Hyper-V hosts have their virtual machine files (virtual hard disks, configuration files, snapshots, and so on) stored on SoFS. SoFS was first introduced in Windows Server 2012 and represented clustered file servers that allow you to store server application data, such as Hyper-V virtual machine files or SQL Server database files, on file shares while maintaining a similar level of reliability, availability, manageability, and high performance as using a SAN for storing such files. With a SoFS, all file shares are online on all nodes simultaneously in an active-active cluster configuration. Again, this kind of approach can be much more cost-effective for organizations than using a SAN for storing the virtual machine files for a Hyper-V cluster. And again, using a scale-out approach instead of scale-up can be a more cost-effective solution, and as we'll see in the next chapter, Windows Server 2012 R2 increases the scale-out capabilities of the SoFS role service.

To enable the virtualization of storage resources, Storage Spaces can be used on SoFS. This allows the physical storage resources for the solution to be pooled together to provide resiliency in case of failure of a storage device. Storage Spaces was also introduced in Windows Server 2012 and provides two types of resiliency—mirroring and parity. Storage devices can be selectively reserved as hot spares so they can automatically replace devices that fail, thus ensuring that the integrity of data is preserved in the event of a power interruption or hardware failure. The storage pools you create using Storage Spaces can be used to provision virtual disks (virtualized storage) on which you can create new volumes and shares for your solution.

- **Physical storage layer** Depending on the performance needs of the workloads running in the solution, the virtual machine files for the Hyper-V clusters can be stored on different types of storage devices. Supported drives include Serial ATA (SATA) and Serial Attached SCSI (SAS) disks, which can be either hard disk drives (HDDs) or solid-state drives (SSDs). These disks can either be internal to the SoFS, directly connected as DAS, or within JBOD enclosures.

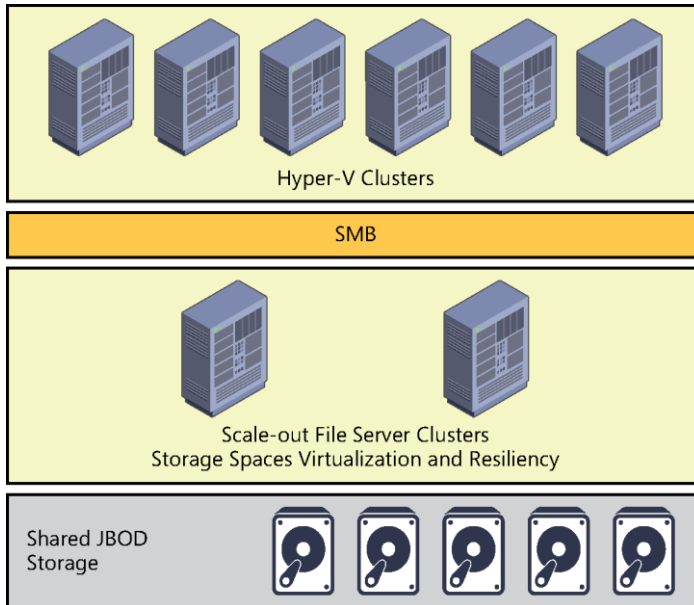


FIGURE 3-1 Microsoft's vision for storage can be implemented by using Windows Server 2012 Hyper-V, SoFS clusters, and shared JBOD storage.

While this kind of cloud solution is already possible using Windows Server 2012, the enhancements to virtualization, storage, and networking in Windows Server 2012 R2 now make it possible to optimize such solutions to achieve enterprise-quality levels of performance, reliability, and availability. That's why Windows Server 2012 R2 is being positioned by Microsoft as a cloud-optimized server operating system.

Enabling the solution using System Center 2012 R2

Windows Server 2012 R2 only represents the foundation for a cloud solution. To manage the cloud solution and the workloads running on it, you also need System Center, and in particular you need System Center Virtual Machine Manager (VMM). What truly makes this new release of Windows Server a cloud-optimized operating system is that it represents the first time that Microsoft has synchronized the product release cycles of Windows Server and System Center. The goal of doing this is to ensure that Microsoft can deliver to both its enterprise and service provider customers a completely integrated solution for building and deploying both private and hosted clouds.

To achieve this goal, the R2 release of System Center 2012 (particularly in VMM 2012 R2) also includes numerous enhancements, particularly in the areas of storage performance, provisioning, and management. For example, System Center 2012 R2 now supports:

- Faster enumerations through its Storage Management Initiative - Specification (SMI-S) storage service

- Real-time updates for out-of-band changes using Common Information Model (CIM) indications
- Fibre Channel fabric discovery and zone provisioning
- Support for Hyper-V Virtual Fibre Channel
- ODX optimized virtual machine deployments
- Rapid provisioning using differencing disks

Although the focus of this book is on Windows Server 2012 R2 and its new features and enhancements, System Center 2012 R2 (and particularly VMM) should really be considered the default platform going forward for managing a cloud solution built using Windows Server 2012 R2 as its foundation.

Storage Management API

One of the key storage management improvements introduced in Windows Server 2012 and System Center 2012 was the Storage Management application programming interface (SM-API). SM-API is a Windows Management Infrastructure (WMI)-based programming interface that provides a standards-based way of managing storage on the Windows Server platform and it supersedes the Virtual Disk Service (VDS) API used in previous versions of Windows Server. And in VMM 2012, the new Windows Standards-Based Storage Management service—which utilizes SM-API—replaces the Microsoft Storage Management Service used in previous versions of VMM.

Figure 3-2 shows how SM-API can be used to manage different kinds of storage providers, arrays, and devices by using Server Manager, VMM, or a third-party storage management tool. (You can also use Windows PowerShell to manage storage on Windows Server 2012.) Some examples of different types of storage providers and arrays you can manage using SM-API include:

- Older symmetric multiprocessing (SMP) based internal storage subsystems
- Newer SMI-S-based internal storage subsystems
- SMI-S-based network attached storage (NAS) devices
- SMI-S-compliant Fibre Channel switches on SAN arrays using CIM pass through
- JBODs that are compatible with Storage Spaces using CIM pass through

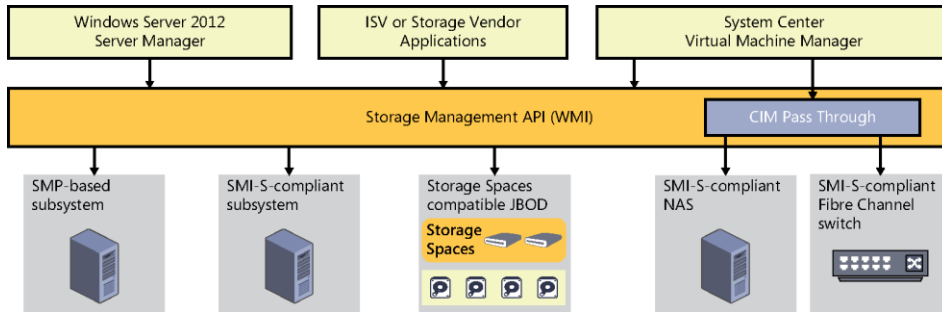


FIGURE 3-2 SM-API can be used to manage a wide range of storage providers and arrays.

While SM-API isn't new in this R2 release of Windows Server 2012, it has been enhanced in several ways, and especially in the area of performance. Specifically, the new version of SM-API includes:

- A new architecture that performs enumerations of storage resources 10 times faster than previously
- The addition of remoting and cluster-awareness when managing Storage Spaces
- Support for new Storage Spaces features like write-back caching and storage tiering which are described later in this chapter
- The ability to use VMM to manage Storage Spaces and SoFS using SM-API

iSCSI Target Server enhancements

Internet Small Computer System Interface (iSCSI) is an industry-standard protocol that allows sharing of block-level storage over a TCP/IP network. Block-level storage is typically used in SANs and is supported by the iSCSI, Fibre Channel, and SAS connection mechanisms. File-level storage involves using network shares on volumes that have been formatted using a file system like NTFS or ReFS.

iSCSI is designed to transmit and receive Small Computer System Interface (SCSI) commands and data encapsulated as TCP packets. This enables servers to utilize storage on an iSCSI-based storage device, such as an iSCSI SAN, even when the servers and SAN are in different locations.

Fibre Channel SANs can be prohibitively expensive for a small or mid-sized business because they require specialized connection hardware such as HBAs and cabling. By contrast, iSCSI needs no specialized connection hardware or special-purpose cabling because it can use a standard Ethernet network for connecting servers with the storage array. This means that iSCSI storage can be deployed using an organization's existing network infrastructure, which helps keep the cost of the iSCSI approach low.

Beginning with Windows Server 2012, a built-in role service (iSCSI Target Server) and client component (iSCSI Initiator) are included and can be used to implement an iSCSI-based storage solution without the need of purchasing a third-party iSCSI SAN. Using these new features, organizations can deploy iSCSI storage without the need of purchasing any additional storage hardware or software.

Some of the usage scenarios for iSCSI storage include:

- Deploying diskless servers that boot from iSCSI virtual disks over the network
- Providing block storage to applications that require or can benefit from it
- Creating iSCSI storage test environments where you can validate applications before deploying them onto a third-party iSCSI SAN

Because Microsoft iSCSI technologies are based on industry standards, you can also deploy Windows Server–based iSCSI storage solutions together with third-party solutions.

One of the key components of the iSCSI implementation in Windows Server 2012 is iSCSI Target Server, a role service under the File and Storage Services role. In a typical iSCSI storage scenario, an iSCSI initiator (a service running on the server consuming the storage) establishes a session (consisting of one or more TCP connections) with an iSCSI target (an object on the target server that allows an iSCSI initiator to establish a connection with the target server) in order to access an iSCSI virtual disk (storage backed by a virtual hard disk file) on the iSCSI Target Server (the server or device, such as a SAN, that shares storage so that users or applications running on a different server can consume the storage). Figure 3-3 shows how these different iSCSI components work together on the Windows Server platform.

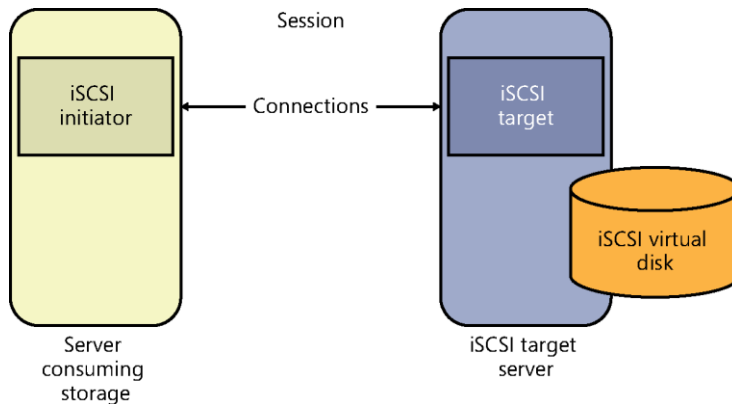


FIGURE 3-3 The basic iSCSI components work together on Windows Server.

In the R2 release of Windows Server 2012, the iSCSI Target Server role service has been enhanced in a couple of important ways:

- VHDX support is now included, which enables the provisioning of larger LUNs up to 64 TB in size. VHDX support also means you can now expand or shrink iSCSI LUNs while they are online (see the section “Online VHDX resize” in Chapter 2 “Hyper-V”)

and dynamically grow them for greater scalability and flexibility. And VHDX is now the default virtual disk format when creating new iSCSI LUNs.

- iSCSI Target Server can now be fully managed using SMS-S. This means that you can now perform end-to-end management of your iSCSI storage system using VMM.

SMB 3.0 enhancements

SMB 3.0 is at the core of the new SoFS functionality introduced previously in Windows Server 2012 and enables file-based storage solutions (file servers) to provide network storage for servers that have similar performance to expensive, proprietary SAN solutions. Using SMB 3.0 and the Windows Server 2012 platform, organizations can build low-cost scale-out storage fabrics that can meet the needs of a broad range of private and hosted cloud solutions.

Some of the key features of SMB 3.0 include:

- **SMB Scale Out** Allows you to create file shares using Cluster Shared Volumes (CSV) to provide simultaneous access to data files, with direct I/O, through all nodes in a file server cluster. This results in improved utilization of network bandwidth, load balancing of SMB 3.0 clients, and optimized performance for server applications.
- **SMB Transparent Failover** Allows you to perform hardware or software maintenance of nodes in a clustered SMB 3.0 file server (an SoFS) without interruption to server applications storing data on the file shares. If a hardware or software failure occurs on a cluster node, SMB 3.0 clients can transparently reconnect to another cluster node without interruption as well.
- **SMB Direct** Supports using network adapters that have Remote Direct Memory Access (RDMA) capability, which allows them to function at full speed with very low latency and very low CPU utilization. When used with workloads, such as Hyper-V or SQL Server, the result is that a remote SMB 3.0 file server can resemble local storage in its performance.
- **SMB Multichannel** Provides aggregation of network bandwidth and network fault tolerance when multiple paths are available between the SMB 3.0 client and the SMB 3.0 file server. This results in server applications taking full advantage of all available network bandwidth and being more resilient to network failure.
- **SMB Encryption** Provides end-to-end encryption of SMB data to protect data from eavesdropping without the need for configuring Internet Protocol security (IPsec), deploying specialized hardware, or utilizing WAN accelerators. Encryption can be configured on a per share basis or for the entire SMB 3.0 file server.

With the release of version 3.0 of SMB, the SMB protocol has become more than just a network file sharing protocol used for copying files over a network. Some of the additional uses for SMB now include:

- A protocol transport for CSV that enables I/O forwarding between the nodes of a failover cluster.

- A storage protocol that enables Hyper-V hosts to access and run virtual machine files stored on file servers on the network.
- A protocol transport for performing live migrations of virtual machines between clustered and nonclustered Hyper-V hosts.

A number of improvements have been made to SMB 3.0 in Windows Server 2012 R2. For example, the performance of SMB Direct has been enhanced to provide a 50 percent improvement for small IO workloads when used with RDMA-capable network adapters. For example, 8KB data transfers have now increased from about 300K I/O operations per second (IOPS) to about 450K IOPS per interface.

A second improvement is the increased efficiency and density of hosting workloads with small I/Os, for example when running an online transaction processing (OLTP) database workload inside a virtual machine. SMB Direct in Windows Server 2012 R2 also includes optimizations for using 40 Gbps Ethernet and 56 Gbps InfiniBand for network transport.

SMB connections can also now be managed per share on SoFS instead of per file server as in Windows Server 2012. But we'll defer further discussion of this until we reexamine SoFS in Chapter 4, "Failover Clustering."

Another new feature of SMB 3.0 in Windows Server 2012 R2 is SMB Bandwidth Management. Because SMB now has so many different functions in a network and storage infrastructure built using Windows Server 2012, it now represents a common infrastructure component in many environments. That means it's important to be able to control how much bandwidth SMB uses when it's performing many different tasks within an infrastructure. As Figure 3-4 shows, you can now configure bandwidth limits for different categories of SMB usage. For example, the figure shows that three categories of bandwidth limits have been configured to ensure optimal performance of the various infrastructure components present in this infrastructure:

- **Default** A limit of 100 MB/s has been configured for Hyper-V host 1 to use SMB when performing file copies from the file server used for library storage by VMM 2012 R2.
- **VirtualMachine** No limit has been set for the amount of bandwidth that Hyper-V host 1 can utilize when using SMB to access virtual machine files stored on the SoFS.
- **LiveMigration** A limit of 500 MB/s has been set for SMB to use when performing live migrations of virtual machines from Hyper-V host 1 to Hyper-V host 2.

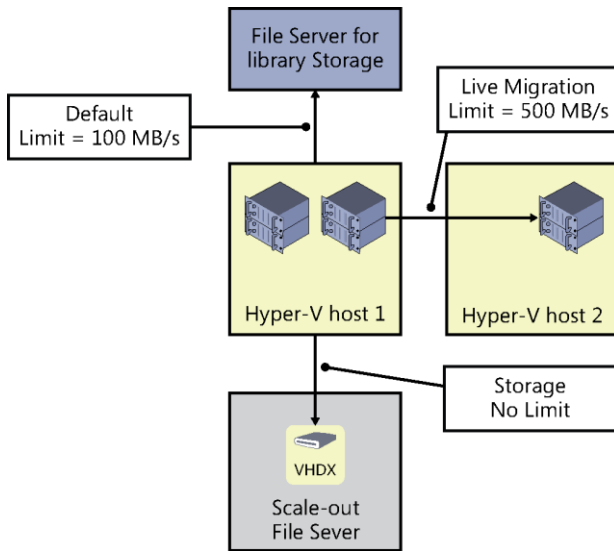


FIGURE 3-4 SMB 3.0 now supports bandwidth management.

Data deduplication enhancements

Data deduplication was introduced in Windows Server 2012 to help enterprises cope with exponentially increasing growth of data storage in their environments. Data deduplication allows Windows Server 2012 to store more data in less physical space to optimize the capacity of their storage fabric. Data deduplication is highly scalable, resource efficient, and nonintrusive in Windows Server 2012, and can run on multiple volumes simultaneously without affecting other workloads running on the server. Checksums, consistency, and identity validation are used to ensure data integrity, and redundant copies of file system metadata are maintained to ensure data is recoverable in the event of corruption.

Windows Server 2012 R2 includes several important improvements to the way data deduplication works. For example, in the previous version deduplication could only be used with files that are closed, such as virtual machine files stored in the VMM library. With this new release, however, deduplication can now be used even with open virtual hard disk files (both VHD and VHDX).

Deduplication in Windows Server 2012 was also incompatible with CSVs. This meant that deduplication couldn't be used to optimize storage of virtual machine files stored on SoFS. This limitation has now been removed in the R2 release of Windows Server 2012 with support for deduplication of data stored on CSVs used by SoFS. For example, Figure 3-5 shows a failover cluster of Hyper-V hosts with the virtual machine files being stored on CSVs used by a two-node SoFS. In order for this scenario to work, SMB 3.0 must be used as the network storage protocol. While SoFS running Windows Server 2012 can provide this functionality, using SoFS running Windows Server 2012 R2 enables deduplication to be turned on for the

CSVs, which enables space savings as high as 90 percent on the CSVs. This type of space savings can be especially beneficial for virtual disk infrastructure (VDI) environments running on Windows Server 2012 R2 Hyper-V hosts.

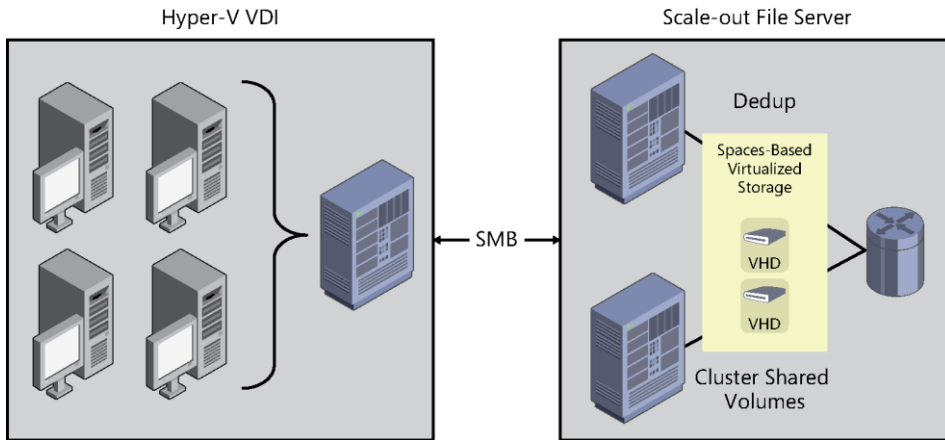


FIGURE 3-5 Windows Server 2012 R2 now supports data deduplication on CSV volumes.

Other improvements to data deduplication in Windows Server 2012 R2 include performance enhancements resulting from faster read/write of optimized files and improved optimization speed. Deduplication is supported only for data volumes, however, and not for boot or system volumes. In addition, ReFS volumes do not support using deduplication.

Storage Spaces enhancements

Until Windows Server 2012 was released, implementing storage virtualization required purchasing proprietary third-party SAN solutions that were expensive and required using their own set of management tools. Such solutions also required special training to implement, manage, and maintain them effectively. Storage Spaces, first introduced in Windows Server 2012, was designed to make storage virtualization affordable even for small businesses. Storage Spaces is simple to deploy and manage, and it can provide businesses with shared storage that can grow on demand to meet an organization's changing needs.

Some of the benefits of using Storage Spaces include:

- **Increased scalability** Additional physical storage can easily be added and used to meet increasing business demands.
- **Increased flexibility** New storage pools can be created and existing ones expanded as the need arises.
- **Increased efficiency** Unused storage capacity can be reclaimed to enable more efficient use of existing physical storage resources.

- **Increased elasticity** Storage capacity can be preallocated by using thin provisioning to meet growing demand even when the underlying physical storage is insufficient.
- **Lower cost** Low-cost, commodity-based storage devices can be used to save IT departments money that can be better allocated elsewhere.

To understand how Storage Spaces might be used for private cloud solutions, Figure 3-6 compares a traditional SAN-based storage solution with one built using Storage Spaces in Windows Server 2012. On the left side is a cluster of Hyper-V hosts whose virtual machine files are stored in LUNs on the SAN. These LUNs are backed by enterprise-class SAS disks (which can be HDDs or SSDs) mounted in disk shelves in the SAN chassis. Establishing connectivity between the Hyper-V host cluster and the SAN requires installing Fibre Channel or iSCSI HBAs in these hosts (depending on type of SAN involved) and data is transferred between the SAN and the Hyper-V host cluster using either Fibre Channel or iSCSI as a block-level storage protocol. Proprietary technology is required for this solution in the form of HBAs, cabling, and the SAN chassis.

By comparison, using Storage Spaces on the right requires no use of proprietary technology. Instead, all of the components of this solution can use off-the-shelf commodity-based server hardware. The SAN chassis is replaced with a cluster of file servers running Windows Server 2012 on enterprise-level server system hardware and rack-based JBOD enclosures containing the same kind of enterprise-class SAS disks (HDDs or SSDs) that might be used in the traditional SAN approach. Connectivity between the Hyper-V host cluster requires only a standard Ethernet network (typically 10 GbE) and high-performance network interface cards installed in the Hyper-V hosts and uses SMB 3.0 as a file-based storage protocol. Everything is commodity hardware here, and there's no vendor lock-in as there is with the traditional SAN approach.

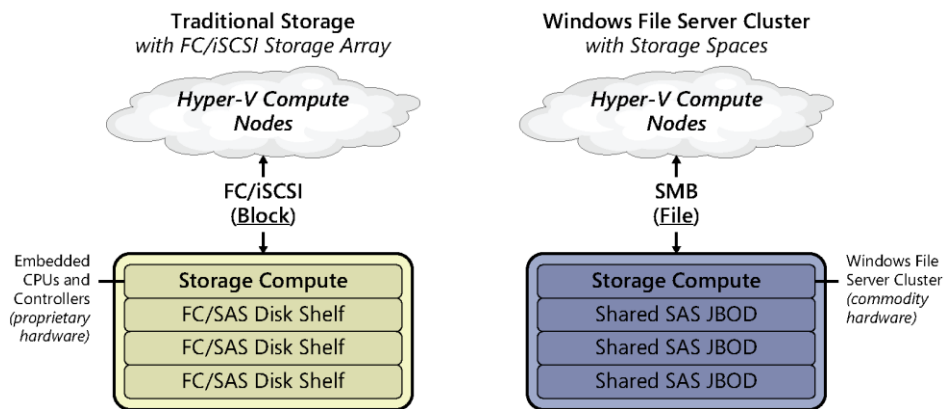


FIGURE 3-6 A comparison of traditional SAN storage with one based on Storage Spaces.

Storage Spaces in Windows Server 2012

The reaction when Storage Spaces was announced was somewhat qualified, especially by large enterprises that run workloads requiring the highest levels of performance involving millions of IOPS and massive throughput. The reason for this was because it was initially assumed that the performance of a virtualized storage solution based on Storage Spaces would fall short of what a typical SAN array can deliver. However, Microsoft soon proved its critics wrong with a demonstration performed at TechEd 2012, where a three-node high-performance server cluster was connected to a 24-bay JBOD filled with enterprise-grade SSDs. When Storage Spaces was used to present storage from the JBOD to the cluster nodes, the performance resulted in an aggregate sequential throughput of 12 GB/s and 1.45 million IOPS. When a second JBOD was added to the environment, the IOPS were increased to 2.7 million! Clearly, Storage Spaces is an enterprise-ready storage virtualization technology and its usage scenarios are not limited only to smaller deployments.

The challenge, however, with the Windows Server 2012 version of Storage Spaces is deciding whether you want to optimize performance or storage capacity when building your storage virtualization solution. For example, if you use Storage Spaces to create storage pools backed by low-cost, large-capacity commodity HDDs, you get a capacity-optimized storage solution but the performance might not be at the level that some of your applications require. This is typically because large-capacity HDDs are optimized for sequential data access while many server applications perform best with random data access. On the other hand, if you create pools using more expensive SSDs, you can easily achieve the kind of random I/O performance your applications require but you probably won't have enough room in your budget to meet your capacity requirements for storage.

The logical solution is to use a mix of low-cost, large-capacity commodity HDDs together with expensive, high-performance enterprise-class SSDs. Building a Storage Spaces solution along these lines can provide you with the best of both worlds, and deliver high levels of IOPS at a relatively low cost compared to using a SAN. This means that there are three ways you can build a virtualized storage solution using Storage Spaces in Windows Server 2012:

- **Capacity-optimized approach** Uses only low-cost, large-capacity commodity HDDs to provide high capacity while minimizing cost per terabyte
- **Performance-optimized approach** Uses only expensive, high-performance enterprise-class SSDs to provide extreme performance, high throughput, and the largest number of IOPS per dollar
- **Balanced approach** Uses a mix of HDDs and SSDs to achieve good performance and reasonable capacity at an acceptable cost

Unfortunately, there's a problem with the balanced approach. This is because while most enterprise workloads have a relatively large data set, the majority of data in this working set is often cold (seldom-accessed) data. Only a minority of data is typically in active use at a given time, and this hot data can be considered the working set for such workloads. Naturally, this working set also changes over time for the typical server workload. Since the working set is small, it would seem natural to place the hot data (the working set) on high-performance SSDs while

keeping the majority of the data (which is cold data) on high-capacity HDDs. But the working set changes over time, so how do you seamlessly ensure that hot data is placed on SSDs and cold data on HDDs when you use Storage Spaces to create pools containing both SSDs and HDDs?

The answer is you couldn't do that with Storage Spaces—until now.

Storage Spaces in Windows Server 2012 R2

As Figure 3-7 shows, the Windows Server 2012 R2 version of Storage Spaces now allows you to create a tiered storage solution that transparently delivers an appropriate balance between capacity and performance that can meet the needs of enterprise workloads. The result is that the workload's most frequently accessed data (the working set) will automatically be stored on the SSD tier while the rest of the workload's data is stored on the HDD tier.

How does Storage Spaces accomplish this? By having the file system actively measure the activity of the workload in the background and then automatically and transparently move data to the appropriate tier (SSD or HDD) depending on how hot or cold the data is determined to be. Storage Spaces in Windows Server 2012 R2 can thus ensure that the workload's hot data is always stored on the SSD tier to take advantage of the high performance of this tier, and it's cold data on the HDD tier to make use of the high capacity of this tier. If a portion of the data for a particular file becomes hotter (is accessed more frequently), then it gets moved from the HDD tier to the SSD tier. And if the portion of data becomes cooler (is accessed less frequently), then it gets moved from the SSD tier to the HDD tier.

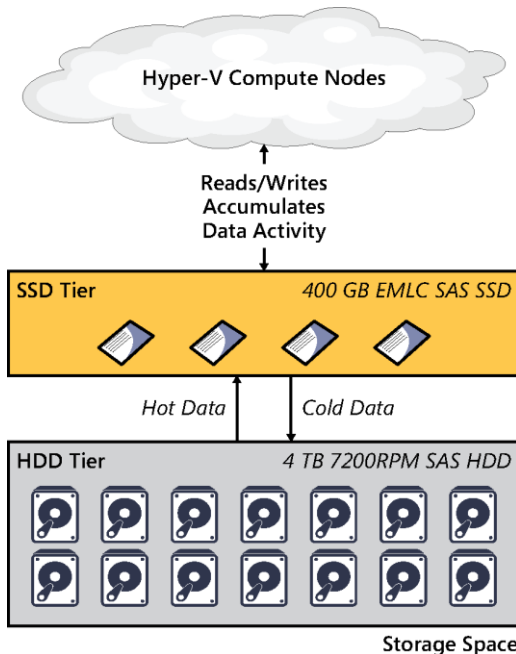


FIGURE 3-7 Storage Spaces in Windows Server 2012 R2 now supports data tiering.

This seamless movement of data between tiers is configured by default to happen daily in 1 MB chunks, but you also have the option of configuring the scheduled task for this operation to run as frequently as you want. Data moves between tiers in the background and has minimal impact on the performance of the storage space. If needed, you can use Windows PowerShell to assign certain files to a specific tier, thereby overriding the automatic placement of data based on heat. For example, the parent virtual hard disk file for a collection of pooled virtual machines in a VDI environment might be assigned to the SSD tier to ensure the file always remains pinned to this tier. The result of doing this can be to achieve significant improvements in the boot times of the hundreds or thousands of virtual desktops derived from this parent virtual hard disk.

Tiered Storage Spaces is an exciting new capability in Windows Server 2012 R2, and when this book is revised and enlarged in the RTM timeframe, we'll take a closer look at how it can be implemented. But there's one additional new feature of Storage Spaces that we'll briefly describe here, and it's called *write-back caching*. While the goal of tiering is to balance capacity against performance, the purpose of write-back caching is to smooth out short-term bursts of random writes. Write-back caching integrates seamlessly into tiered volumes and is enabled by default. The write-back cache is located on the SSD tier of a storage space and services smaller, random writes; larger, sequential writes are serviced by the HDD tier. You can also enable write-back caching on nontiered volumes.

Storage QoS

Storage Quality of Service (QoS) is another new feature of file-based storage introduced in Windows Server 2012 R2. Storage QoS is enabled at the VHDX layer and allows you to limit the maximum IOPS allowed to a virtual disk. It can also allow you to set triggers to send notifications when a specified minimum IOPS is not met for a virtual disk. Possible usage scenarios for this feature include:

- Configuring different service-level agreements (SLAs) for different types of storage operations within your infrastructure. For example, a hoster might use this feature to configure Bronze, Silver, and Gold SLAs for storage performance available for different classes of tenants. You can even set alerts that trigger when virtual machines are not getting enough IOPS for storage access.
- Restricting the disk throughput for overactive or disruptive virtual machines within your environment that is saturating the storage array. Hosting providers will especially love this capability since it means they won't have to worry about one tenant consuming excessive storage fabric resources at the expense of other tenants.

As Figure 3-8 shows, Storage QoS can even be configured while the virtual machine is running. This allows organizations to have a lot of flexibility in how they manage access to the storage fabric from workloads running in their cloud environments.

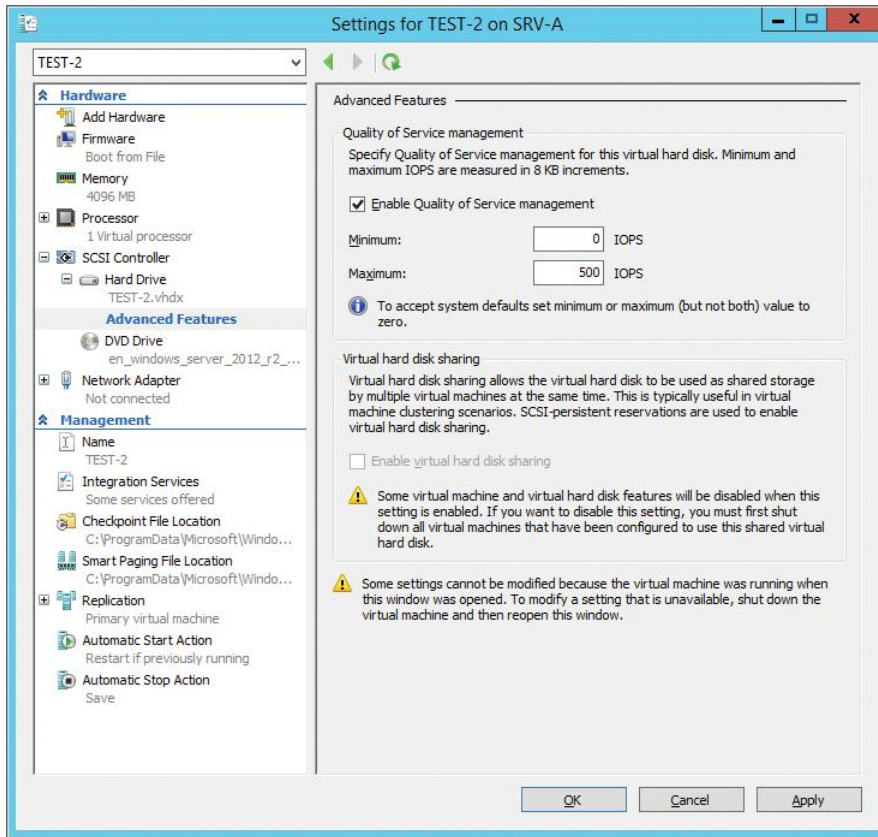


FIGURE 3-8 You can configure Storage QoS for a virtual machine using Hyper-V Manager.

There's more!

Windows Server 2012 R2 also includes other storage enhancements. For example, enhancements to CSV in failover clustering now results in a more highly optimized rebalancing of SoFS traffic. But since this is related to the topic of clustering, we'll defer discussion of this one until we get to Chapter 4, "Failover Clustering," later in this book. Storage Spaces can also be deployed together with a SoFS cluster, and we'll defer a deeper look at this topic to the next chapter as well.

Failover Clustering

Both virtualization and storage can only take you so far unless you also add high availability into the mix. Failover Clustering, a key feature of the Windows Server platform, is designed to do just that by providing high availability and scalability to many types of server workloads including Hyper-V hosts, file servers, and different server applications such as Microsoft SQL Server and Microsoft Exchange Server that can run on both physical servers and virtual machines.

While Windows Server 2012 included a number of important enhancements to the Failover Clustering feature, Windows Server 2012 R2 adds even more. This chapter continues the discussion of what's new in Windows Server 2012 R2 by describing several of the key improvements to Failover Clustering functionality in the new platform. Unfortunately in a short book like this we won't be able to cover all of the Failover Clustering improvements in Windows Server 2012 R2, but when the book is revised and expanded around RTM it will include information about some other Failover Clustering improvements in the new platform.

But first let's start by reviewing the Failover Clustering enhancements that were previously introduced in Windows Server 2012.

Previous enhancements to Failover Clustering

Some of the many ways that Failover Clustering was enhanced in Windows Server 2012 include:

- **Improved scalability** Compared with Failover Clustering in Windows Server 2008 R2, the number of cluster nodes supported increased from 16 to 64 in Windows Server 2012. The number of clustered roles or virtual machines also increased from 1,000 to 8,000 (up to 1,024 per node) in the new platform. This increased scalability enabled new scenarios and efficiencies to help IT departments deliver more for the dollar.
- **Cluster Shared Volumes enhancements** Cluster Shared Volumes (CSV) were introduced in Windows Server 2008 R2 to provide common storage for clustered virtual machines. CSV was enhanced in Windows Server 2012 and enabled to provide storage for additional clustered roles such as the new Scale-out File Server (SoFS) feature, which can provide continuously available and scalable file-based (SMB 3.0) server storage for Hyper-V and applications such as SQL Server. CSV could also be integrated with the new Storage Spaces feature of Windows Server 2012 to enable

scale-out access to data by virtualizing cluster storage on groups of inexpensive disks (JBODs). CSV in Windows Server 2012 was also integrated with new SMB 3.0 features like SMB Multichannel and SMB Direct, which allow CSV traffic to stream across multiple networks in the cluster and leverage network adapters that support Remote Direct Memory Access (RDMA). Other CSV improvements in Windows Server 2012 included support for BitLocker Drive Encryption, removal of external authentication dependencies, and improved file backup.

- **Updating failover cluster nodes** Cluster-Aware Updating (CAU) was introduced in Windows Server 2012 to enable software updates to be applied automatically to the host operating system or other system components on the nodes of a failover cluster while maintaining availability during the update process. CAU reduced maintenance time by automating what was previously a very repetitive task.
- **Quorum improvements** New features of the cluster quorum feature in Windows Server 2012 included simplified quorum configuration, support for specifying which cluster nodes had votes in determining quorum, and dynamic quorum, which provides administrator the ability to automatically manage the quorum vote assignment for a node based on the state of the node.
- **Other enhancements** Some of the many other enhancements to Failover Clustering in Windows Server 2012 included simplified migration of the configuration settings of clustered roles, more robust integration with Active Directory Domain Services, improved cluster validation tests, improved Windows PowerShell support, Node Maintenance Mode, clustered tasks, new clustered roles like iSCSI Target, guest clusters using virtual Fibre Channel, and more. Many of the Hyper-V enhancements in Windows Server 2012 are also relevant to Failover Clustering, for example virtual machine prioritization, pre-emption to shut down low-priority virtual machines, virtual machine health monitoring, Hyper-V Replica Broker, and so on.

Guest clustering using shared virtual disks

As we mentioned briefly at the end of Chapter 2, Hyper-V in Windows Server 2012 R2 now allows guest clustering using shared VHDX files. This new exciting capability will be especially appreciated by hosters who want to maintain separation between their own storage infrastructure and that of their tenants. Why is that?

Hosting highly available workloads

Consider your typical hoster for a moment. A hoster provides its customer with services that allow them to run their virtual machines in the cloud instead of on location at their company premises. These virtual machines are pretty important to the customers too, since they are typically running server workloads—like SQL Server—that are critical to the operation of the

customer's business. In fact, they're so important to the customer that they want the hoster to make sure these workloads are highly available. And since the hoster says they have the infrastructure to do this and wants the customer's business, they agree.

Now let's say that until today the customer has been running their virtualized workloads on-premises. To ensure high availability for their workloads, they've been using two types of failover clustering, namely guest host clustering and guest clustering. First, the customer has been using host clustering, which means running the Failover Clustering feature in the parent partition of two or more Hyper-V hosts. (To understand host clustering, think of making a single virtual machine highly available.) Using host clustering helps you ensure continuous availability in the event of a hardware failure on a host, when you need to apply software updates to the parent partition resulting in a reboot being required, and similar scenarios.

Second, the customer has been using guest clustering, which means running the Failover Clustering feature in the guest operating system of two or more virtual machines. (To understand guest clustering, think of multiple virtual machines in a failover cluster.) Using guest clustering helps you proactively monitor application health and mobility within the guest operating system and protect against application failures, guest operating system problems, host and guest networking issues, and other problem scenarios.

By combining both types of failover clustering like this, the customer has the best of both worlds. In other words, having a cluster of clusters (guest clustering on top of host clustering) gives you the highest level of availability for your virtualized workloads.

But now let's say that the customer wants to move their virtualized workloads into a hosted cloud. So the customer asks the hoster to provide them with high availability similar to what they've been using on-premises. The hoster agrees but wants to maintain complete separation between the tenant's virtual workloads and the hoster's own physical storage infrastructure.

Separating virtual resources from physical infrastructure

The new guest clustering using shared virtual disks capability of Windows Server 2012 R2 now makes such a scenario possible for hosters. As you'll see in a moment, what this new capability enables you to do is to keep your physical infrastructure layer (computer, storage, and network) and your virtualized resources (tenant virtual machines and the workloads running on them) separate from one another.

This approach can benefit hosters since it allows them to maintain strict control over the physical infrastructure of their cloud while providing great flexibility in how they deliver virtual resources to customers. Specifically, it allows them to provision virtual machines, services, and applications to customers (together with the virtual compute, storage, and network resources needed) while keeping the underlying physical resources opaque to them.

For example, when a customer spins up a new virtual machine, the customer doesn't care which Hyper-V host it's running on, which physical network that host is sitting on, or which logical unit number (LUN) its virtual machine files are stored on. All the customer cares about

is that they get the necessary virtualized compute, storage, and network resources they need to run their workload with the performance they desire. The hoster should be able to reallocate, reconfigure, and upgrade their physical infrastructure without interrupting customer workloads or even the customers being aware of it.

With guest clustering in the previous versions of Windows Server, maintaining strict separation of tenant virtual machines and the hoster's physical infrastructure just wasn't possible. That's because in order to implement guest clustering you had to present a LUN to your virtual machines so they could use it as shared storage for the Failover Clustering feature running in the guest operating system of the virtual machines. In Windows Server 2008 R2, you would generally do this by having iSCSI initiators running in the guest operating system to enable connectivity with an iSCSI-based storage device. (You could also use Fibre Channel over Ethernet, or FCoE, for this purpose as well, but the point is that you were still restricted to using NICs to transport the data.) Typically you would host the LUN on an iSCSI SAN, but you could also download and install the free Microsoft iSCSI Target Software from the Microsoft Download Center, install it on a server running Windows Server 2012 R2, and use that server as shared storage for the guest cluster.

Windows Server 2012 made guest clustering easier in two ways. First, the iSCSI Software Target is now an in-box feature integrated into the Failover Clustering feature, and this makes it easier to implement guest clustering using shared iSCSI storage. Second, Windows Server 2012 also includes an in-box Hyper-V Virtual Fibre Channel adapter that allows you to connect directly from within the guest operating system of a virtual machine to LUNs on a Fibre Channel SAN. This means that in Windows Server 2012 you have three choices for the shared storage you'll need if you want to implement guest clustering, namely iSCSI storage, Fibre Channel storage, or Hyper-V over SMB 3.0.

But the problem with guest clustering in Windows Server 2012 is that it still requires that something in your virtual infrastructure (the guest operating system of the clustered virtual machines) needs to be able to directly connect to something in your physical infrastructure (a LUN on your iSCSI or Fibre Channel SAN). What this effectively does when guest clustering is implemented is to open a hole between the physical infrastructure and the clustered virtual machines as shown in Figure 4-1.

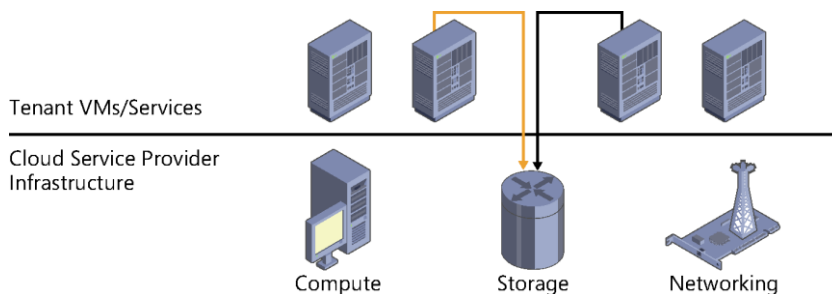


FIGURE 4-1 Hosters like to keep their customers' virtual machines and their own supporting infrastructure separate, but until now guest clustering has required establishing a connection between them.

What's the problem with opening such a hole? For hosters, it basically ties their hands because they can't make changes to their physical storage infrastructure without having it potentially impact the virtualized workloads of the customers that they're hosting. Because of this, hosters usually decline to implement guest clustering because it limits their ability to separate their physical infrastructure from the tenant's workloads. Customers that want to move workloads into the cloud and have high availability ensured by both host and guest clustering are unable to do this, and this makes them unhappy because they are unable to migrate their workloads into the hoster's cloud to simplify the operations and management of their workloads. So hosters who want to offer both host and guest clustering to customers can't do this, and they lose out on potential business opportunities.

But now with the new guest clustering using shared VHDX capability of Windows Server 2012 R2, hosters can provide guest clustering to customers without the need of providing direct access by the clustered virtual machines to an iSCSI target or a LUN on a Fibre Channel storage array. This means you can now implement guest clustering for tenant virtual machines running in a hoster's cloud while maintaining complete separation between the hoster's physical storage infrastructure and the virtualized storage resources consumed by the virtual machines (see Figure 4-2). This is an important issue for most hosters as they usually utilize separate networks for providing tenant virtual machine connectivity and storage infrastructure connectivity and, for security and reliability reasons, they want to keep these networks completely separate.

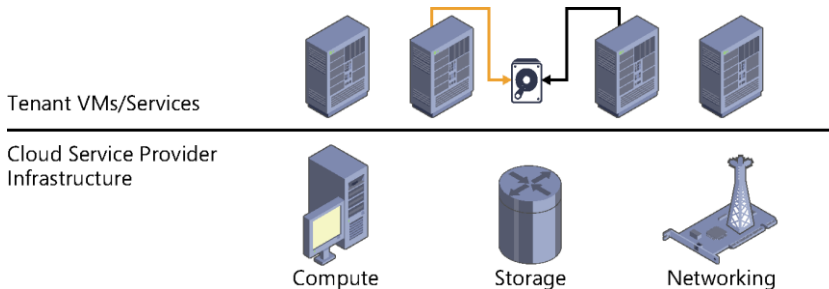


FIGURE 4-2 Guest clustering using shared virtual disks in Windows Server 2012 R2 enables hosters to keep their customers' virtual machines and their own supporting infrastructure separate.

Understanding shared virtual disks

The key to being able to implement guest clustering for tenant virtual machines running in a hoster's cloud is a new capability in Windows Server 2012 R2 that allows a VHDX file (the new virtual hard disk format introduced earlier in Windows Server 2012) to be shared by multiple virtual machines running on one or more Hyper-V hosts. To the hosts, these shared virtual disks look like simple VHDX files attached to multiple virtual machines (each virtual machine already has at least one other virtual hard disk for the guest operating system). To the virtual machines themselves, however, the shared virtual disks appear to be (and behave as if they are) virtual Serial Attached SCSI (SAS) disks that can be used as shared storage for a failover cluster. The

virtual machines in a guest cluster can share one or more virtual SAS disks depending on how the cluster is configured. Figure 4-3 shows how the hosts and virtual machines view the shared VHDX files in a guest cluster.

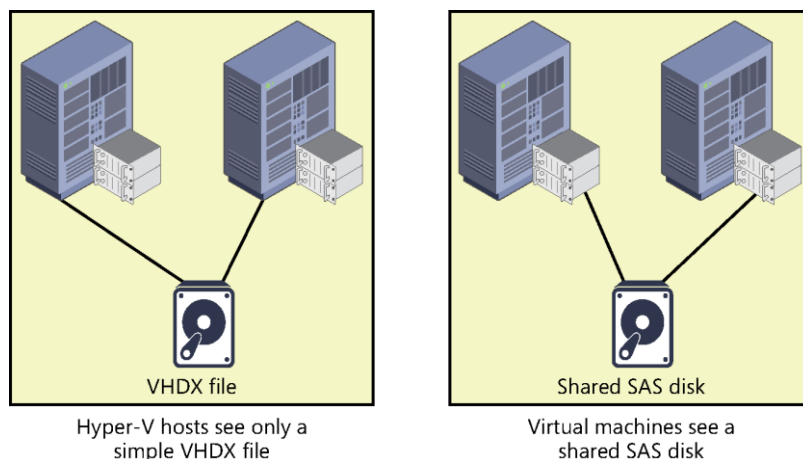
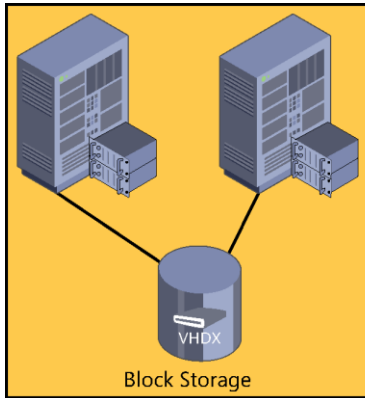


FIGURE 4-3 Hyper-V hosts see shared virtual disks as simple VHDX files, but the guest cluster of virtual machines running on these hosts sees the virtual disks as shared SDS disks.

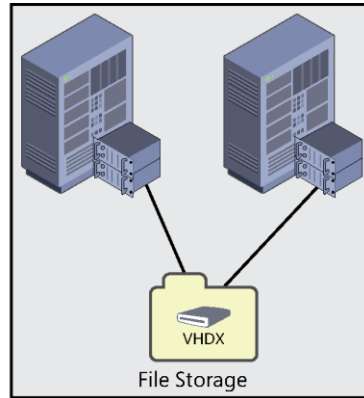
As in Windows Server 2012, the actual physical storage backing the shared storage volume(s) used by the failover cluster can be implemented using one of two approaches as shown in Figure 4-4:

- Using CSV disks for block storage, for example by placing the shared VHDX file on a CSV disk provisioned from a virtual disk using Storage Spaces
- Using a SoFS for file storage, which allows the shared VHDX file to be stored on an SMB 3.0 shared folder

Both of these approaches allow the use of low-cost commodity storage instead of more expensive SAN solutions for the shared storage used by the guest cluster. They also allow you to deliver guest clustering using exactly the same infrastructure you use to deliver standalone virtual machines. In other words, you don't need to have specialized storage such as an iSCSI or Fibre Channel SAN, or a Windows Server system with the Microsoft iSCSI Target Software installed in order to implement guest clustering.



Cluster Shared Volumes (CSV)
for block storage



Scale-out File Server
for file storage

FIGURE 4-4 Guest clusters using shared virtual disks can use either CSV disks for block storage or SoFS for file storage.

Using shared virtual disks

Implementing guest clustering using shared virtual disks on Windows Server 2012 R2 is easy:

1. Create a new VHDX file on the volume you will use for shared storage for the cluster.
2. Open the Settings dialog for a virtual machine in Hyper-V Manager.
3. Click the SCSI Controller option under the Hardware category, select Hard Drive, and click Add to add a new hard drive to the controller.
4. Browse to select the VHDX file you created earlier, then expand the new hard drive under the SCSI Controller to expose the Advanced Features option.
5. Click the Advanced Features option and select the Enable Virtual Hard Disk Sharing checkbox as shown in Figure 4-5, and then click OK (or Apply) to apply all the changes.
6. Repeat steps 2 through 5 of the above procedure for each virtual machine in the guest cluster.

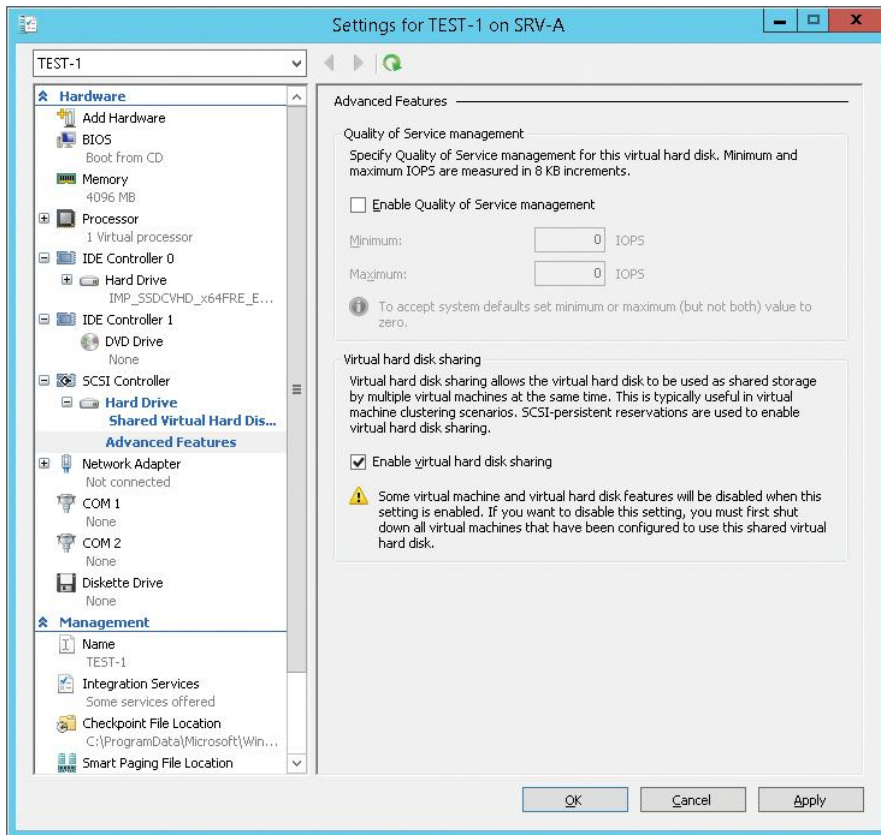


FIGURE 4-5 Use these settings to share a virtual hard disk.

A few things to consider:

- The shared virtual disk must be a data disk; guest operating system disks cannot be shared.
- The shared virtual disk must use the newer VHDX format; it cannot use the older VHD format.
- The shared virtual disk must be attached to the virtual machine's SCSI controller; it cannot be attached to the IDE controller.
- When performing the above procedure, don't click Apply until you have selected the Enable Virtual Hard Disk Sharing checkbox. If you do this, you will have to remove the disk from the controller and reselect it in order to share it.

Of course, all this can be done using Windows PowerShell as well.

CSV and SoFS enhancements

At the end of Chapter 3, “Storage,” we mentioned that enhancements to CSV in Failover Clustering now result in a more highly optimized rebalancing of how the SoFS feature works, but that since this related to the topic of clustering we’d defer discussing it until this present chapter. This is a fairly significant improvement for SoFS, so let’s look at it now together with some other improvements in how CSV now works in Windows Server 2012 R2.

Recall that SoFS is a feature introduced in Windows Server 2012 that allows you to use the Failover Clustering feature to deploy active-active clusters of file servers that can store server application data, such as Hyper-V virtual machine files or SQL Server database files, using file shares instead of using LUNs on a SAN. The key of course is that a SoFS allows you to achieve a similar level of reliability, availability, manageability, and performance as that of a SAN. And since a SoFS can use Storage Spaces, another feature of Windows Server 2012 that allows you to use low-cost commodity disks to create pools of storage from which you can provision resilient volumes, the approach can often be much more cost-effective than using a SAN. That’s obviously good news for organizations whose IT budgets are constrained, which probably includes everyone nowadays.

Figure 4-6 illustrates how a SoFS can redirect I/O over SMB to the optimal node. A virtual machine running on a Hyper-V host wants to access a file in a shared folder named Share2 on a two-node SoFS. The nodes of the SoFS are named File Server 1 and File Server 2. When the virtual machine attempts to connect to the share, it might connect to either Share2 on File Server 1 or to Share2 on File Server 2. Let’s say that it establishes an SMB connection to Share2 on File Server 1 as shown by the dashed line on the left. Unfortunately, this is not an optimal connection because the file it wants to access is actually located on storage attached to File Server 2. The SoFS detects this situation, however, and it automatically and seamlessly transitions the SMB connection from Share2 on File Server 1 to Share2 on File Server 2 as shown by the dashed arrow at the bottom. From this point until the end of the SMB session, to provide optimal I/O and throughput, the SMB connection between the virtual machine and Share2 on File Server 2 can use direct I/O as shown by the solid line.

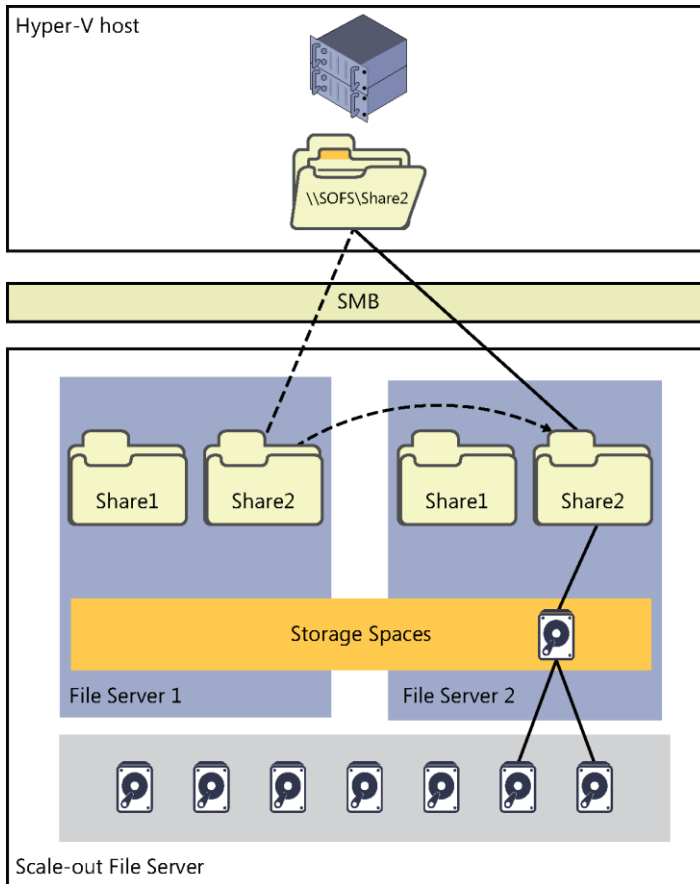


FIGURE 4-6 A SoFS can redirect I/O over SMB to the optimal node.

To ensure that a SoFS can deliver SAN-quality throughput and IOPS, Microsoft has made a couple of optimizations in how a SoFS works in Windows Server 2012 R2. We'll begin by considering things from the perspective of CSV. Recall that CSV allow multiple nodes in a failover cluster to simultaneously have read-write access to the same LUN (disk) provisioned as an NTFS volume. This enables clustered roles to fail over quickly from one node to another node without requiring a change in drive ownership or dismounting/remounting a volume. CSV also help simplify managing a large number of LUNs in a large failover cluster. CSV work by providing a general-purpose, clustered file system layered on top of NTFS. Examples of CSV applications can include clustered virtual hard disk (VHD or VHDX) files for clustered Hyper-V virtual machines and scale-out file shares to store application data for the SoFS role.

Consider a failover cluster with a bunch of nodes with a bunch CSV disks, and each CSV disk is backed by a shared LUN exposed through Storage Spaces on a SoFS. Applications running on any node have simultaneous read/write access to the shared LUN even if it is only mounted

on one of the cluster nodes. The coordinator node, one of the cluster nodes, handles all synchronization of metadata for file system I/O on the shared LUN and is the only node where NTFS is mounted. An example of such metadata would be the file system changes that occur when a virtual machine running on a Hyper-V cluster is turned off. Such metadata changes are routed over the SMB path to the coordinator node. By contrast, file system I/O that is not metadata is sent directly down the stack to the storage to provide direct I/O performance.

In Windows Server 2012, the Failover Clustering feature handles orchestration of file system changes on CSV disks by performing orchestration separately for each LUN. This meant that if you had eight cluster nodes using four LUNs for shared storage, you had to manually spread the CSV disks across the cluster. Also, Failover Clustering in Windows Server 2012 had no built-in mechanism for ensuring that they stayed spread out. In fact, all of the CSV disks could be owned by a single node.

Failover Clustering in Windows Server 2012 R2, however, now includes a mechanism for fairly distributing the ownership of CSV disks across all cluster nodes based on the number of CSV disks each node owns. Rebalancing ownership of CSV disks happens automatically whenever a new node is added to the cluster, a node is restarted, or a failover occurs.

Another enhancement in Windows Server 2012 R2 is when a workload, such as a virtual machine running on a Hyper-V host, attempts to establish an SMB 3.0 connection with a SoFS, the SoFS will try to determine whether the workload has an optimized I/O path it can use to access the CSV disk using direct I/O. One of the key improvements with Failover Clustering in Windows Server 2012 was that CSV disks could perform more operations in direct I/O mode than occurred in Windows Server 2008 R2. Direct I/O can be used with storage that has Fibre Channel, iSCSI or SAS connectivity, and involves writing directly to a CSV disk. Drivers for devices that can transfer large amounts of data at a time can use direct I/O for those transfers, and using direct I/O for such transfers improves a driver's performance, both by reducing its interrupt overhead and by eliminating the memory allocation and copying operations inherent in buffered I/O.

Changes to heartbeat threshold

While the goal of failover clustering is to deliver high availability for server workloads, beneath the hood failover clustering is simply a health detection model. Each and every second a heartbeat connection is tested between nodes in the cluster. If no heartbeat is heard from a node one second, nothing happens. No heartbeat for two seconds? Nothing happens. Three seconds? Nothing. Four seconds? Nothing. Five seconds?

Five seconds is the default heartbeat threshold for all cluster roles in the Windows Server 2012 version of Failover Clustering. That means if the coordinator node doesn't hear from one of the other cluster nodes in five seconds, it assumes that the other node has failed or is

partitioned, so it takes corrective action to remedy the situation and ensure continued availability of the workloads currently running on the failed node.

As an example, let's say you have a single-subnet failover cluster of Hyper-V hosts with virtual machines running on them. VM-A is currently running on HOST-1 and everything is working fine until an unexpected network interruption occurs on the network connecting the nodes. A transient issue with the network switch could be the cause, perhaps because someone tripped over the switch's power cable in the server room and plugged it back in immediately but the switch took a few seconds to reboot. Whatever the cause of the network interruption, Failover Clustering decides that since the heartbeat threshold had been exceeded for HOST-1, that node must be down, so it assumes that VM-A has become unavailable even though clients are still able to access the workload running on the virtual machine. Since HOST-1 has been determined to have failed, Failover Clustering begins taking remedial action on that node, which terminates any active client connections to the workload. Meanwhile, it starts booting up VM-A on a different node so clients will be able to access the workload again.

What has actually happened here, unfortunately, is that Failover Clustering could be said to have triggered a false failover. Of course, that's not really true—it's the network interruption that caused the problem. The best action in this case would be to ensure that all network cables are physically secured. But what if your network experiences more mysterious transient failures? And what if you, as cluster administrator, can't identify or address these network failures because another section of the IT department oversees the network infrastructure and they aren't cooperative or helpful? For one reason or another, transient network interruptions are sometimes unavoidable for some customers. Yet network interruptions that exceed the heartbeat threshold can cause a cluster to fail over when such action is neither necessary nor desired. What's to be done?

To address this issue, the Windows Server team made a change in Failover Clustering in Windows Server 2012 R2. Instead of the five-second heartbeat threshold used for all clustered roles in Windows Server 2012, the heartbeat threshold has been increased, but only for the Hyper-V clustered role. For Hyper-V cluster nodes on the same subnet, the threshold is now 10 seconds. And for Hyper-V cluster nodes on different subnets, the threshold is now 20 seconds. This reduction of cluster sensitivity to health problems was specifically made to enable Hyper-V clusters to provide increased resiliency to packet loss on unreliable networks. However, increasing the heartbeat threshold like this can have the negative result of greater downtime when a real failure does happen, so administrators also have the option of configuring a lower threshold if they want to. But you shouldn't raise the threshold any higher than 20 seconds or it can cause TCP sessions to unexpectedly terminate. And remember, increasing the heartbeat threshold doesn't fix the underlying network problems you're experiencing—it only masks them.

Detecting the health of virtual machines

Interruptions in the physical network infrastructure of a failover cluster aren't the only types of network problems that can cause problems for clusters. Network disconnections can also occur at the virtual machine level on the clustered Hyper-V hosts. If this happens in Windows Server 2012, the virtual machine continues to run on the cluster node even though the workload on the virtual machine is no longer available to clients.

A new setting called Protected Network in Windows Server 2012 R2 can resolve this kind of problem by automatically moving the virtual machine to a different cluster node if the virtual machine's network becomes disconnected. To enable this capability for a virtual machine running on a Hyper-V host cluster, simply do the following:

1. Open the Settings dialog for a virtual machine in Hyper-V Manager.
2. Expand the Network Adapter node under Hardware, and select the Advanced Features option.
3. Select the Protected Network checkbox, as shown in Figure 4-7.

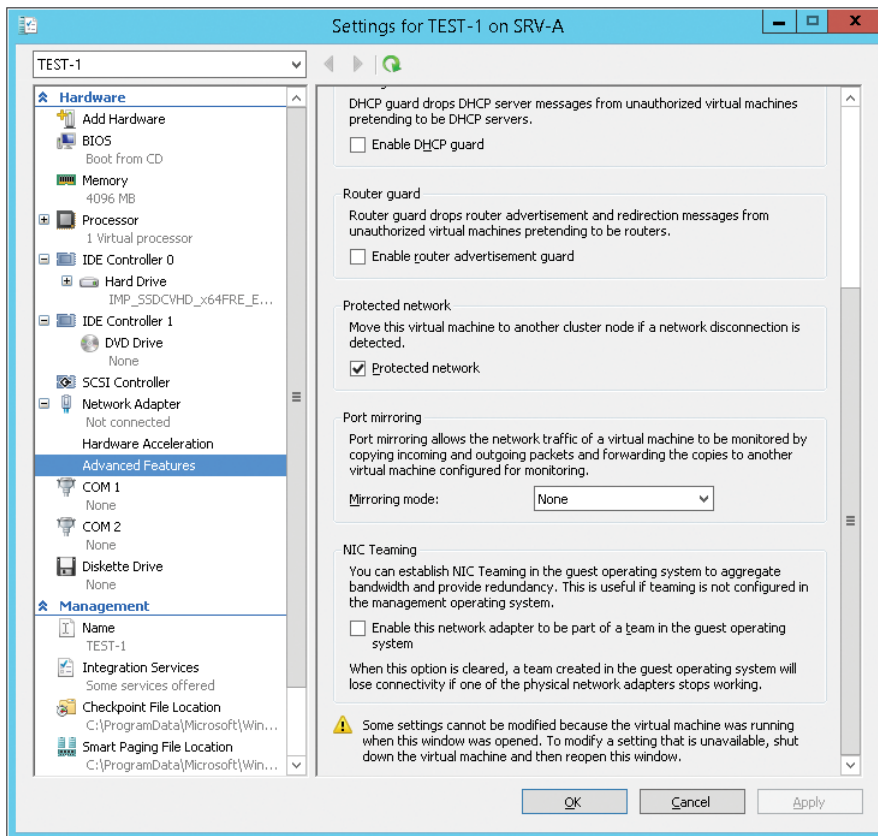


FIGURE 4-7 You can now enable the Protected Network setting.

Other enhancements to Failover Clustering

There are other enhancements to Failover Clustering and we'll just briefly mention them here since this book only represents a first look at the Preview release of Windows Server 2012 R2. (When the book is revised and expanded around the RTM timeframe, we'll dig a bit deeper into these additional enhancements.) For example:

- CSV now supports the Resilient File System (ReFS) and deduplication.
- CSV now supports parity spaces and the new features of Storage Spaces in Windows Server 2012 R2 such as tiered spaces and write-back caching.
- The state of a CSV disk can now be viewed on a per-node basis, for example to see whether a particular CSV disk is available or whether its I/O is direct or redirected.
- A higher percentage of total physical memory can now be allocated to the CSV cache.
- Failover clusters can now be deployed without creating any computer objects in Active Directory Domain Services.
- Improvements have been made that make it easier to implement dynamic quorum.
- The assigned and current quorum votes for nodes can now be viewed in Failover Cluster Manager.
- A new cluster dashboard is now included in Failover Cluster Manager that lets you quickly view the health of all of your clusters.

There's more still, but that's all we'll examine for now.

Networking

Currently, IT is all about the cloud, and the foundation of cloud computing is infrastructure. If you're an enterprise that's going to build and manage a private cloud, then you'll be dealing with three main kinds of infrastructure: compute, storage, and network. And if you're a hoster creating a cloud for selling services to customers, then you'll be working with the same three building blocks but on a much larger scale and with additional features that are essential for hosting environments such as multitenant isolation, IP address management, and Network Virtualization.

In Chapter 2, "Hyper-V," we looked at Hyper-V improvements in Windows Server 2012 R2. Hyper-V hosts provide the compute infrastructure needed for running virtualized workloads in a cloud infrastructure. In Chapter 3, "Storage," we examined storage improvements in the new platform. Storage Spaces and the Scale-out File Server (SoFS) are two storage technologies that can enable new scenarios and help lower costs when deploying the storage infrastructure for a cloud solution. In Chapter 4, "Failover Clustering," we looked at how the Failover Clustering feature has been enhanced in the platform. Failover clustering enables your compute and storage resources to be highly available, which is essential for today's always-on businesses.

In this chapter, we'll now examine the networking improvements in Windows Server 2012 R2. Networking is the underlying glue that holds your infrastructure together, makes possible the delivery of services, makes remote management a reality, and more.

But first let's begin by reviewing the networking enhancements introduced earlier in Windows Server 2012.

Previous enhancements to networking

Some of the many networking improvements introduced previously in Windows Server 2012 included the following:

- **Dynamic VMQ** Virtual Machine Queue (VMQ) allows a host's network adapter to pass DMA packets directly into the memory stacks of individual virtual machines. The net effect of doing this is to allow the host's single network adapter to appear to the virtual machines as multiple NICs, which then allows each virtual machine to have its own dedicated NIC. Windows Server 2012 improved this by introducing Dynamic VMQ, which dynamically distributed incoming network traffic processing to host processors, based on processor use and network load. The earlier implementation, which was also called Static VMQ, was removed in Windows Server 2012.

- **Receive Side Scaling** Receive Side Scaling (RSS) allows network adapters to distribute kernel-mode network processing across multiple processor cores in multicore systems. Such distribution of processing enables support of higher network traffic loads than are possible if only a single core is used. RSS was enhanced in Windows Server 2012 to support systems with up to 64 processors, improved scalability across Non-Uniform Memory Access (NUMA) nodes, improved management and diagnostics, and automatic load balancing capabilities for non-TCP traffic such as UDP unicast, multicast, and IP-forwarded traffic.
- **Windows NIC Teaming** Also known as load balancing and failover (LBFO), Windows NIC Teaming enables multiple network interface cards (NICs) on a server to be grouped together into a team. This has two purposes: to help ensure availability by providing traffic failover in the event of a network component failure and to enable aggregation of network bandwidth across multiple NICs. Previously, implementing NIC teaming required using third-party solutions from independent hardware vendors (IHVs). Beginning with Windows Server 2012, however, NIC teaming was now an in-box solution that worked across different NIC hardware types and manufacturers.
- **Quality of Service enhancements** Quality of Service (QoS) refers to technologies used for managing network traffic in ways that can meet service level agreements (SLAs) and/or enhance user experiences in a cost-effective manner. For example, by using QoS to prioritize different types of network traffic, you can ensure that mission-critical applications and services are delivered according to SLAs and to optimize user productivity. Windows Server 2012 introduced a number of new QoS capabilities including Hyper-V QoS, which allows you to specify upper and lower bounds for network bandwidth used by a virtual machine, and new Group Policy settings to implement policy-based QoS by tagging packets with an 802.1p value to prioritize different kinds of network traffic.
- **Data Center Bridging** Data Center Bridging (DCB) is an IEEE standard that allows for hardware-based bandwidth allocation for specific types of network traffic, which means that DCB is yet another QoS technology. DCB-capable network adapter hardware can be useful in cloud environments where it can enable storage, data, management, and other kinds of traffic all to be carried on the same underlying physical network in a way that guarantees each type of traffic its fair share of bandwidth. Windows Server 2012 supported DCB, provided that you had both DCB-capable Ethernet NICs and DCB-capable Ethernet switches on your network.
- **Dynamic Host Configuration Protocol enhancements** Dynamic Host Configuration Protocol (DHCP) functionality was enhanced in several ways in Windows Server 2012. DHCP Server Failover was introduced as a new approach for ensuring DHCP availability by enabling two DHCP servers to replicate lease information between them. That way, one of the DHCP servers could assume responsibility for providing addresses to all the clients on a subnet when the other DHCP server became unavailable. Policy-based assignment allowed a DHCP server to

evaluate DHCP requests against policies you defined for a specific scope and in a defined processing order.

- **Domain Name System enhancements** Domain Name System (DNS) functionality was also enhanced in several ways in Windows Server 2012. The DNS Server component included improved support for DNS Security Extensions (DNSSEC) including support for DNS dynamic updates in DNSSEC signed zones, automated trust anchor distribution through Active Directory, automated trust anchor rollover, and support for updated DNSSEC standards. The DNS Client component included improved support for Network basic input/output system (NETBIOS) and Link-local multicast name resolution (LLMNR), binding order optimization, and asynchronous DNS caching.
- **IP Address Management** IP Address Management (IPAM) is a new built-in framework introduced in Windows Server 2012 for discovering, monitoring, auditing, and managing the IP address space used on a corporate network. IPAM provided a central and integrated experience for managing IP addresses that could replace manual, work-intensive tools such as spreadsheets and custom scripts that can be tedious, unreliable, and scale poorly.
- **Network virtualization** Network virtualization was introduced in Windows Server 2012 as a way for organizations to keep their own internal IP addresses when moving their servers into a hoster's cloud. Network virtualization works by allowing you to assign two different IP addresses to each virtual machine running on a Windows Server 2012 Hyper-V host: the customer address, which is the IP address that the server had when it resided on the customer's premises before it was migrated into the cloud; and the provider address, which is the IP address assigned by the cloud provider to the server once the server has been migrated to the provider's datacenter. Network virtualization thus lets the cloud provider run multiple virtual networks on top of a single physical network in much the same way as server virtualization lets you run multiple virtual servers on a single physical server. Network virtualization also isolates each virtual network from every other virtual network, with the result that each virtual network has the illusion that it is a separate physical network. This means that two or more virtual networks can have the exact same addressing scheme, yet the networks will be fully isolated from one another and each will function as if it is the only network with that scheme.
- **BranchCache enhancements** BranchCache allows organizations to increase the network responsiveness of centralized applications that are being accessed from remote offices, with the result that branch office users have an experience similar to being directly connected to the central office. BranchCache was first introduced in Windows Server 2008 R2 and was enhanced in Windows Server 2012 with improved performance and reduced bandwidth usage, default encryption of cached content, new tools that allowed you to preload cachable content onto your hosted cache servers even before the content was first requested by clients, single instance storage and downloading of duplicated content, and tighter integration with the File Server role.

There were also other networking technologies introduced or improved in Windows Server 2012 that closely relate to other infrastructure components like compute and storage. For example, there was the Hyper-V Extensible Switch, which added new virtual networking functionality to the Hyper-V server role. There was version 3.0 of the Server Message Block (SMB) file-sharing protocol, which enabled new network storage scenarios such as the SoFS. There was single-root I/O virtualization (SR-IOV), which enabled a network adapter to divide access to its resources across various PCIe hardware functions and reduced processing overhead on the host, which can make the network performance of a virtual machine nearly as good as that of a physical computer. And there was a lot more new networking features and capabilities introduced previously in Windows Server 2012.

Many of the above networking features and technologies have now been improved even more in Windows Server 2012 R2. Let's examine some of these enhancements now, and then when this book is revised and expanded around RTM, we'll look at some others and dig a bit deeper.

Virtual RSS

Today, enterprise-grade NICs can be extremely fast—so fast in fact that a single processor core of the server won't be able to make full use of the NIC's throughput capability. RSS in Windows Server 2012 helps you get around that by allowing kernel-mode network processing to be spread across multiple cores in a multicore server system. In Windows Server 2012, however, virtual machines were limited to using only one virtual processor for processing network traffic. As a result, virtual machines running on Hyper-V hosts were unable to make use of RSS to utilize the highest possible network traffic loads. To compound the problem, VMQ would affinitize all traffic destined for a virtual machine to one core inside the host for access control list (ACL) and vmSwitch extension processing.

With Windows Server 2012 R2, however, this is no longer a limitation. That's because a new feature called virtual RSS (vRSS) maximizes network utilization for a virtual machine by spreading the processing of the traffic across multiple virtual processors inside the virtual machine and also inside the host. This is demonstrated by Figure 5-1, which shows a virtual machine that has four virtual processors assigned to it. The physical NIC can now spread traffic among available cores inside the host, while the virtual NIC distributes the processing load across the virtual processors inside the virtual machine.

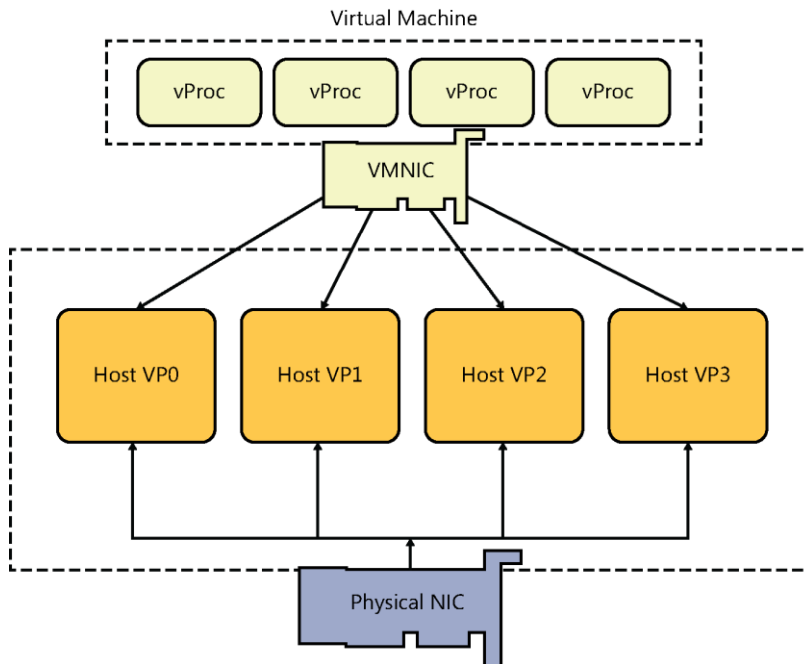


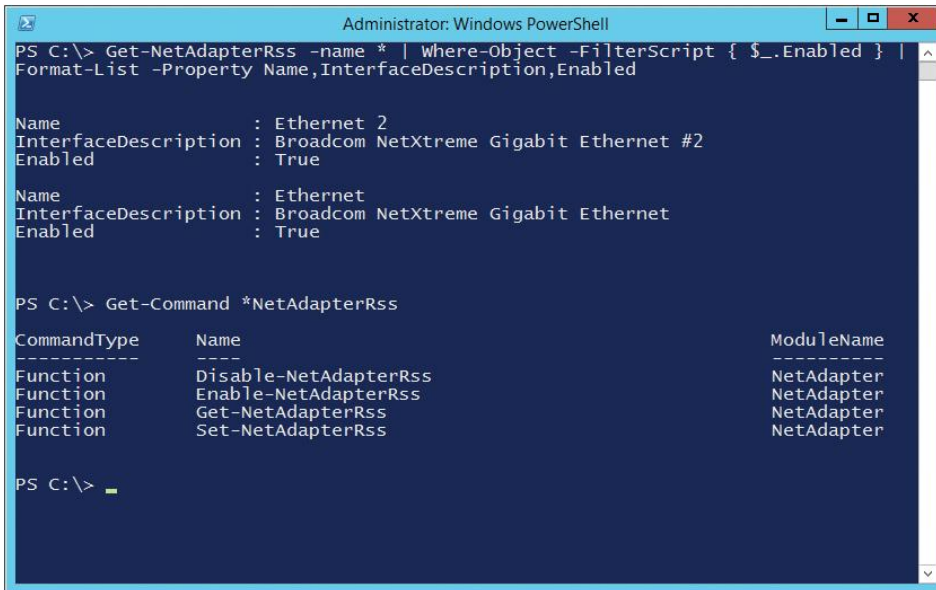
FIGURE 5-1 Virtual Receive Side Scaling (vRSS) is available in Windows Server 2012 R2.

The result of using vRSS is that it is now possible to virtualize network-intensive physical workloads that were traditionally run on bare-metal machines. A typical usage scenario might be a Hyper-V host that has a small number or only one virtual machine running on it, but the applications running in that virtual machine generate a large amount of network traffic. For example, vRSS can be especially useful for virtual network appliances, virtual gateways, file servers, and similar network-intensive applications, as you'll now be able to virtualize them without any network throughput degradation.

A nice thing about vRSS is that it will work on existing network hardware that is VMQ-capable. This means you don't have to upgrade your hardware in order to take advantage of this new capability.

RSS (and vRSS) is disabled by default in Windows Server 2012 and should only be enabled on network-intensive virtual machines. This is because extra processing is required for vRSS to spread the incoming network traffic inside the host. In other words, enabling vRSS trades CPU cycles for network throughput.

You can configure and manage vRSS by using Windows PowerShell commands and scripts. Figure 5-2 shows how to use the `Get-NetAdapterRss` cmdlet to get all RSS-capable network adapters on a system that have RSS enabled and display their names, interface description, and state. The figure also shows how to use the `Get-Command` cmdlet to display a list of all cmdlets available for managing RSS and vRSS.



```
Administrator: Windows PowerShell
PS C:\> Get-NetAdapterRss -name * | Where-Object -FilterScript { $_.Enabled } |
Format-List -Property Name,InterfaceDescription,Enabled

Name                : Ethernet 2
InterfaceDescription : Broadcom NetXtreme Gigabit Ethernet #2
Enabled              : True

Name                : Ethernet
InterfaceDescription : Broadcom NetXtreme Gigabit Ethernet
Enabled              : True

PS C:\> Get-Command *NetAdapterRss

CommandType      Name                                     ModuleName
-----
Function         Disable-NetAdapterRss                   NetAdapter
Function         Enable-NetAdapterRss                   NetAdapter
Function         Get-NetAdapterRss                     NetAdapter
Function         Set-NetAdapterRss                     NetAdapter

PS C:\>
```

FIGURE 5-2 You can use Windows PowerShell to manage vRSS.

Windows NIC Teaming enhancements

NIC teaming involves linking together two or more network adapters in a server. This can provide two types of benefits for enterprise networks. First, it allows you to aggregate the throughput from multiple network adapters. For example, let's say you have a server system that has two 1 gigabit network adapters configured as a team. The result is that the total throughput of the team is $1 + 1 = 2$ gigabits, so teaming network adapters together basically gives your server a bigger "pipe" for sending and receiving traffic over the network the server is connected to.

The second benefit of NIC teaming is that it helps ensure continuous availability of the server's connection to the network by providing fault tolerance. For example, let's say that one of the NICs in the above team fails. If this happens, the throughput drops from 2 gigabits to 1 gigabit, and while such a 50 percent drop in network traffic handling capability could affect the performance of applications running on the server, the good thing is that the server still has some connectivity with the network. Without NIC teaming, failure of a single NIC would have caused the throughput to drop from 1 gigabit to zero, which is probably much worse from a business point of view.

Before Windows Server 2012, if you wanted to make use of NIC teaming, then you had to use third-party NIC teaming software from your network adapter vendor. With the release of Windows Server 2012, however, NIC teaming became a built-in feature called Windows NIC Teaming that makes it possible to team together even commodity network adapters to aggregate throughput and enable fault tolerance.

NIC Teaming in Windows Server 2012

When you configure NIC Teaming in Windows Server 2012, you have two choices you can select for the load-balancing mode:

- **Address Hash** Selecting this mode causes Windows to load balance network traffic between the physical network adapters that have been teamed together. This is the default load-balancing mode when you create a new team and should be used for most servers including Hyper-V hosts.
- **Hyper-V Port** Selecting this mode causes Windows to load balance network traffic by virtual machine. This mode has the limitation in that each virtual machine running on the host can use only a single NIC in the team for sending and receiving traffic. As a result, this mode should only be selected when the virtual machines running on the host have multiple virtual network adapters. If you use this mode when a virtual machine has only one virtual network adapter and the physical NIC being used by that adapter fails, that virtual machine will lose all connectivity with the network even though the host machine still has connectivity because of the fault tolerance of the NIC team.

Another limitation of how NIC Teaming is implemented in Windows Server 2012 is the way that the default load-balancing mode works. The hashing algorithm used by Windows NIC Teaming first creates a hash based on the contents of the network packet. Then it assigns any packets having this hash value to one of the network adapters in the team. The result is that all packets belonging to the same Transmission Control Protocol (TCP) flow (or stream) are handled by the same network adapter. This type of load balancing can prove to be ineffective in certain scenarios.

For example, say you have a virtual machine that has a single virtual network adapter running a Hyper-V host that has two network adapters configured as a team with Address Hash mode configured. The virtual machine's workload includes several transactional applications, and the TCP flows for these applications are distributed between the teamed network adapters on the host. The applications are currently utilizing most of the available throughput of the team.

Now let's say that you need to copy a very large file to the virtual machine, for example a 150 GB Windows Media file. Address Hash mode will cause the entire TCP flow for this file copy operation to be handled by just one of the NICs in the team. The result is that flows for any other applications that are currently being handled by that particular NIC might become starved for bandwidth, impacting the performance of the applications while the file copy completes.

NIC Teaming in Windows Server 2012 R2

To address these issues, the implementation of NIC Teaming in Windows Server 2012 R2 introduces a new mode—Dynamic—that basically combines the capabilities of the Address Hash and Hyper-V Port modes. Dynamic mode breaks up TCP flows into "flowlets" and then

distributes the flowlets across the different NICs in the team. As Figure 5-3 shows, Dynamic mode is now the default load-balancing mode in Windows Server 2012 R2.

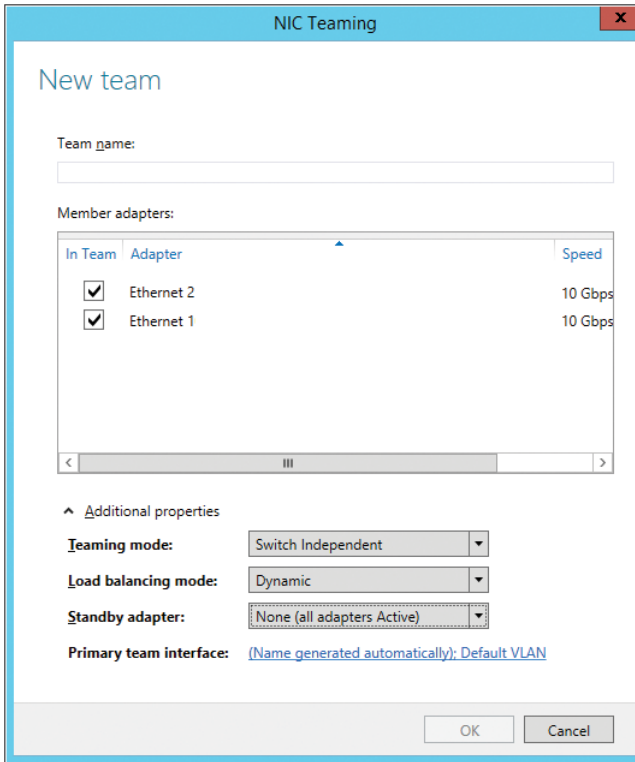


FIGURE 5-3 Dynamic mode is now the default load-balancing mode for NIC Teaming in Windows Server 2012 R2.

IPAM enhancements

Enterprise networks today tend to be a lot more complex than they were 20 or even 10 years ago. With the advent of virtualization technologies like Hyper-V, network administrators have to deal with both physical networks and virtual networks. Nodes on both kinds of networks need IP addresses in order to be reachable, and these IP addresses can be assigned statically (manually), dynamically using DHCP, dynamically using Automatic Private IP Addressing (APIPA), or more typically, using some combination of these three methods. DNS is also needed in order to provide name resolution services so servers can have easy-to-remember friendly names like SRV-A instead of hard-to-memorize IP addresses like 169.254.115.33.

DHCP simplifies the task of allocating IP addresses to clients on a network (and to servers by using DHCP reservations), but large organizations can have multiple DHCP servers with each server having dozens or more scopes and each scope having its own special set of

options. Similarly, DNS simplifies management of fully-qualified domain names (FQDNs) for both servers and clients, but large organizations can have multiple DNS servers with each one authoritative over dozens of zones and each zone containing thousands of resource records.

How does one manage all this? On the Windows Server platform, you can use the DHCP Server snap-in to manage all the DHCP servers in your organization, but the snap-in really isn't efficient when it comes to managing large numbers of DHCP servers. Similarly, you can use the DNS Server snap-in to manage all your DNS servers, but again this snap-in isn't efficient for managing large numbers of DNS servers.

For example, on large networks neither the DHCP snap-in nor DNS snap-in are very useful for helping you get quick answers to any of the following questions:

- Which DHCP servers in my organization manage which blocks of the IP address space?
- Which IP addresses are actually being used at each site where my company has a physical presence?
- Which IP addresses have been assigned to virtual network adapters of virtual machines running on Hyper-V hosts?
- How can I modify a particular scope option for a certain number of scopes residing on several different DHCP servers?
- How can I determine which subnets of an IP address range being managed by a certain DHCP server are not being used?
- How can I determine how many free IP addresses are available for leasing for certain scopes on certain DHCP servers?
- How can I find all scopes that have 95 percent or more of their address pool leased out to clients?
- How can I track all the IP addresses that have been assigned over the last 12 months to a certain server on my network?
- How can I find DNS servers that don't have a certain server option configured?

Administrators of large enterprises often want answers like these—and want them quickly. However, it can be difficult for them to keep track of their IP addressing schemes, DHCP server configurations, and DNS server configurations. Cloud hosting providers can have even greater difficulties keeping track of such information because their environments include both physical and virtual networks, and because of the address space reuse that often happens in multitenant cloud environments.

In the past, most large enterprises and hosters have relied on either spreadsheets, custom software developed in-house, or third-party commercial programs for keeping track of IP addressing schemes, DHCP server configurations, and DNS server configurations. Beginning with Windows Server 2012, however, Microsoft introduced an in-box solution for performing these kinds of tasks. That in-box solution is IPAM.

IPAM in Windows Server 2012

IPAM is an integrated set of tools that helps you plan, deploy, manage, and monitor your IP address infrastructure. IPAM includes functionality for performing address space management in multiserver environments and includes monitoring and network auditing capabilities. IPAM can automatically discover the IP address infrastructure of servers on your network and allows you to manage them from a central interface that is integrated into Server Manager. IPAM is an agentless technology that works together with the Active Directory Domain Services (AD DS), DHCP Server, DNS Server, and Network Policy Server (NPS) roles of the Windows Server platform.

Some of the capabilities of IPAM in Windows Server 2012 include:

- Integrates management of IP addresses, domain names, and device identities
- Organizes, assigns, monitors, and manages static and dynamic IPv4 and IPv6 addresses
- Automatic discovery of domain controllers, DHCP and DNS servers, and dynamic IP addresses in use
- Provides custom IP address space display, reporting, and management
- Tightly integrates with Microsoft DNS and DHCP servers
- Centralizes configuration and update of Microsoft DHCP and DNS servers
- Monitors and manages specific scenario-based DHCP and DNS services
- Tracks and audits changes and provides real-time view of status
- Audits server configuration changes and tracks IP address use

Although IPAM in Windows Server 2012 provided enough functionality for some organizations, other organizations (especially hosters) wanted something more. That something more has now arrived with the enhanced IPAM included in Windows Server 2012 R2.

IPAM in Windows Server 2012 R2

IPAM in Windows Server 2012 R2 represents a significant step forward in simplifying the management of addressing and DNS/DHCP server management both for large enterprises and especially for cloud hosting providers. Let's examine some of these improvements now.

First and perhaps most importantly, IPAM in Windows Server 2012 R2 now allows you to manage your virtual address space in addition to its physical address space. This means that enterprises that have a mixture of physical and virtual environments now have an in-box unified IP address management solution. Figure 5-4 shows the new Virtualized IP Address Space option in the IPAM page of Server Manager and some of the available configuration options.

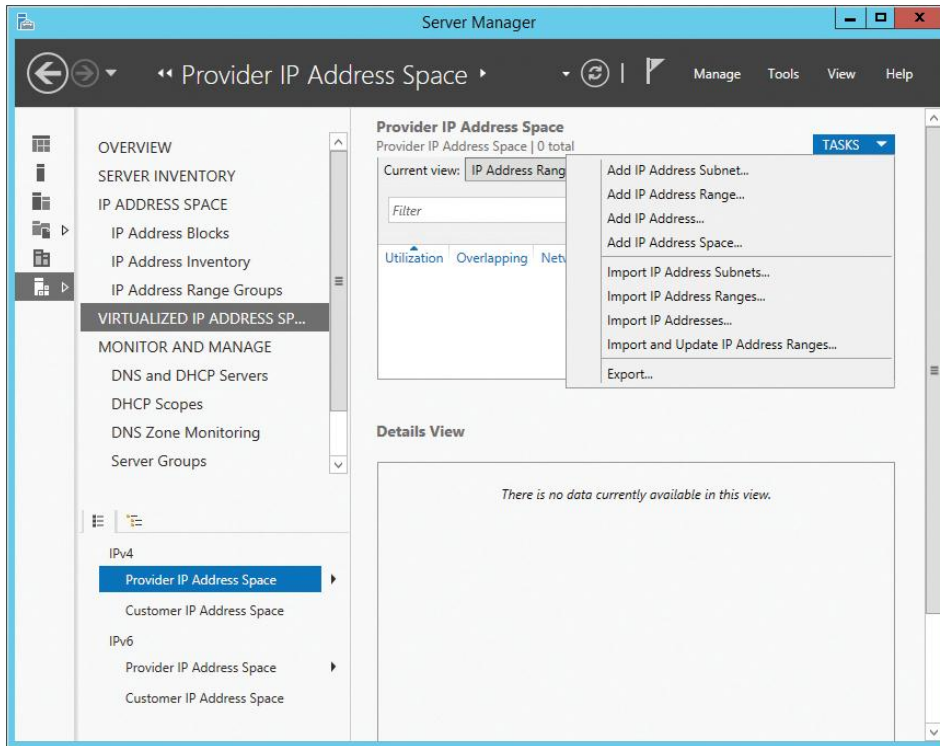


FIGURE 5-4 You can now manage the Virtualized IP Address Space option for IPAM using Server Manager.

This enhancement should be especially useful to very large enterprises and to hosters whose networks often consist of multiple datacenters in different geographical locations. Such networks typically consist of an underlying fabric layer of infrastructure resources with a multitenant virtual network layered on top.

For hosters that have built their cloud using the Windows Server platform, IPAM in Windows Server 2012 R2 offers the additional advantage of tight integration between IPAM and System Center Virtual Machine Manager (VMM) 2012 R2. Because IPAM basically knows everything about your physical network, such as its subnets, virtual LANs (VLANs), and pools, IPAM can make all this information available to VMM for the rapid provisioning of new virtual networks.

Another very important enhancement to IPAM in Windows Server 2012 R2 is the introduction of role-based access control. Role-based access control provides the organization with a mechanism for specifying access to objects and capabilities in terms of the organizational structure. Instead of assigning certain rights and permissions to a user or group of users, role-based access control allows you to assign a certain role instead. The role aligns with the kinds of tasks a certain type of user needs to be able to perform, and assigning the role to a user automatically grants that user all the necessary rights and permissions they need to do their job.

For example, let's say your organization is very large and you have a fairly large IT staff to manage your network. You want to assign the job of being able to manage the IP address

space for your network to three individuals, and you want to give the job of managing network infrastructure servers (DHCP and DNS) to two other people. Role-based access control in IPAM now makes doing these things simple.

Role-based access control in IPAM is also highly granular. For example, let's say that you want to allow a junior administrator to be able to manage only a certain DHCP scope on a certain DHCP server. Role-based access control in IPAM allows you to grant such access to the user while preventing her from having any other IP address management capabilities in your organization.

Role-based access control also includes the ability to delegate administrative privileges. This means that a senior administrator can delegate to junior administrators the ability to perform certain kinds of address management tasks in your organization.

To understand how role-based access control is integrated into the capabilities of IPAM, Figure 5-5 shows the basic architecture for how IPAM works. At the top is an administrator whose computer is running Windows 8.1 and has the Remote Server Administration Tools (RSAT) for Windows 8.1 installed. The administrator opens Server Manager on his computer and selects the IPAM page of Server Manager, as shown previously in Figure 5-4. The tasks available for the administrator to perform will depend on the specific role assigned to the administrator—for example, whether he is a network administrator, fabric administrator, system administrator, forensics investigator, and so on.

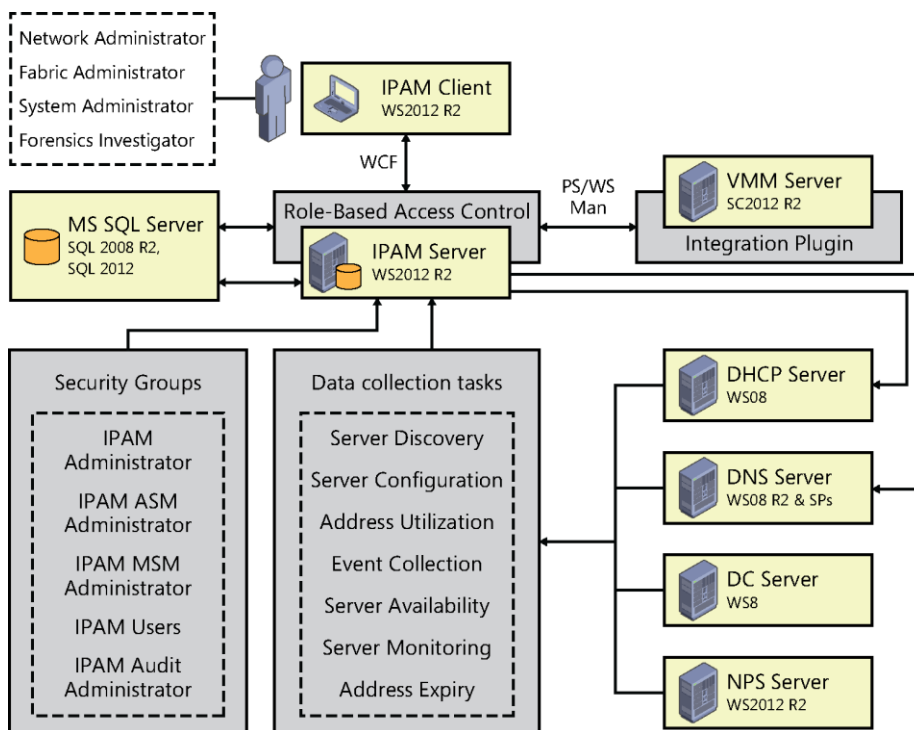


FIGURE 5-5 An example of how role-based access control is integrated into the capabilities of IPAM in Windows Server 2012 R2.

The IPAM client (in Server Manager) communicates with the IPAM server using Windows Communication Foundation (WCF)—a framework for building service-oriented applications. All communications between the IPAM client and IPAM server must go through the role-based access control component, which controls what tasks the administrator is able to perform.

The IPAM server performs various data-collection tasks including performing server discovery and monitoring, determining the configuration and availability of servers, performing event collection, determining address utilization, checking for expired addresses, and so on. To perform these different tasks, IPAM communicates with the DHCP servers, DNS servers, NPS servers, and domain controllers on your network.

As described earlier, the IPAM server can also communicate with one or more VMM servers. Such communications are made possible by using an IPAM integration plug-in included in VMM 2012 R2.

While role-based access control should be the way to use IPAM going forward, IPAM in Windows Server 2012 R2 still includes the various IPAM security groups that were used in the previous version of IPAM for granting administrative privileges to users and groups.

The IPAM server stores the information it collects about your organization's network in a database. In the previous version of IPAM in Windows Server 2012, addressing information that was collected could only be stored in the Windows Internal Database (WID). IPAM in Windows Server 2012 R2, however, can now store data in a Microsoft SQL Server database running either on the local IPAM server or on an external server. This means that you can now ensure that the IPAM database is highly available, back the database up more easily, perform custom queries against it using T-SQL, and so on. As Figure 5-6 shows, when you provision a Windows Server 2012 R2 system as an IPAM server, you have the option of using either the WID or a SQL server for storing the data that IPAM collects for your network. As Figure 5-5 illustrates, the SQL Server database for IPAM must be on SQL Server 2008 R2 or SQL Server 2012.

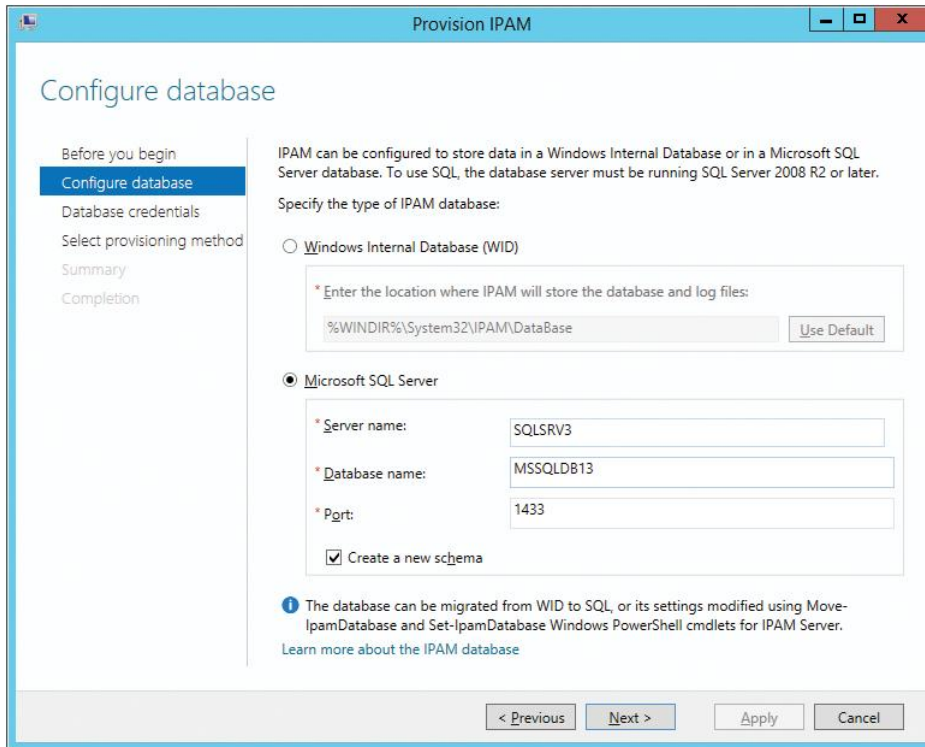


FIGURE 5-6 IPAM in Windows Server 2012 R2 can now store data in a Microsoft SQL Server database.

IPAM in Windows Server 2012 R2 also includes extensive capabilities for allowing you to monitor and manage the activity of the DHCP and DNS servers on your network. For example, IPAM allows you to monitor such things as:

- Server availability
- DHCP Scope utilization
- DNS Zone replication health
- DHCP Failover health

IPAM also enables you to enable/disable features, activate/deactivate entities, allow/deny actions, and so on relating to DNS and DHCP servers.

In terms of management capability, IPAM in Windows Server 2012 R2 makes it easy for administrators to manage from a central user interface and for a whole organization such things as:

- DHCP servers
- DHCP scopes
- DHCP properties
- DHCP options
- DHCP filters

- DHCP failover relationships
- DHCP policies
- DHCP classes
- DHCP reservations
- DNS resource records

Just imagine how much easier this is compared to using the DHCP or DNS snap-ins for performing such tasks!

IPAM in Windows Server 2012 R2 is also fully integrated with Windows PowerShell. In fact, there is 100 percent parity between what you can do using the IPAM page in Server Manager and the IPAM cmdlets in Windows PowerShell. The IPAM Windows PowerShell provider also facilitates integration with other platforms such as System Center Configuration Manager and the Microsoft Assessment and Planning (MAP) Toolkit. Such integration can simplify and speed the network discovery of the IP address inventory of your network. And you can also leverage an Internet Control Message Protocol (ICMP)-based discovery module for performing network discovery as well.

IPAM in Windows Server 2012 R2 also integrates with Active Directory Domain Services (AD DS). Specifically, IPAM enables synchronization of Active Directory Sites and Subnets information from Active Directory to IPAM. This too makes it quick and easy for IPAM to determine the subnet structure of your organization's network.

IPAM in Windows Server 2012 R2 is clearly a cost-effective, scalable, and customizable solution for unified management of physical and virtual network IP address spaces, and DHCP and DNS services in both enterprise and hoster environments.

Hyper-V Network Virtualization enhancements

Hyper-V Network Virtualization was introduced in Windows Server 2012 as a key part of Microsoft's vision for software-defined networking (SDN). Traditional enterprise networks typically had many different physical networking devices such as Ethernet switches, routers, virtual private networking (VPN) gateways, hardware firewalls, and other kinds of network appliances. And of course they needed lots of wires to connect all the network hosts (servers and appliances) together.

The modern data center network is no longer like that. Instead of dozens of network devices and hundreds or thousands of servers, the modern data center might consist of only a dozen or so very powerful virtualization host systems, a handful of 10 GbE switches, and a couple of perimeter firewalls. Instead of thousands of individual physical servers, you now have thousands of virtual machines running on relatively few Hyper-V hosts. One reason this change is possible is because of server virtualization and consolidation, which enables organizations to virtualize workloads that previously needed to run on physical server systems. Another reason it's possible, however, is because of SDN technologies like network virtualization that allow you to consolidate multiple physical network devices onto a single physical networking device in a

similar fashion to how server virtualization lets you consolidate multiple physical servers onto a single physical virtualization host. The basic idea of network virtualization is nothing new, and Hyper-V Network Virtualization is simply Microsoft's implementation of the concept of network virtualization.

How Hyper-V Network Virtualization works

The basic idea behind network virtualization is that it allows multiple virtual machine networks to overlay a cloud hosting provider's underlying physical network (see Figure 5-7). Each virtual machine network, which can be composed of one or more virtual subnets, is independent of all other virtual machine networks and also of the hoster's underlying physical network. In other words, the exact physical location of an IP subnet on the hoster's physical network is decoupled from the virtual network topology of each customer's network.

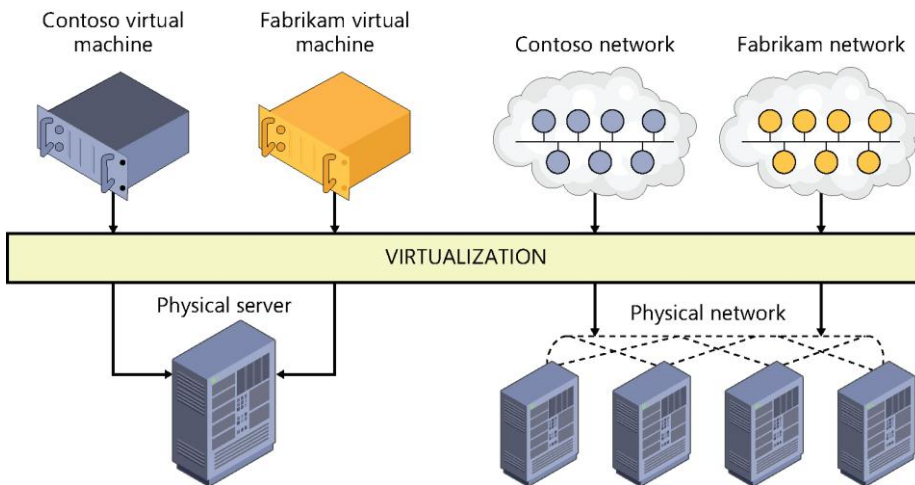


FIGURE 5-7 Hyper-V Network Virtualization allows multiple virtual machine networks to overlay a cloud hosting provider's underlying physical network.

The benefit of this decoupling is that customers can easily move physical server workloads to a hoster's cloud while preserving the IP addresses and network topology of their workloads. For example, let's say that your organization has three physical servers residing on-premises and having private IP addresses of 192.168.33.45, 192.168.33.46, and 192.168.33.47. You want to virtualize these servers by converting them into virtual machines, which you will move to the hoster's cloud. Your physical servers are currently using 192.168.0.0/16 as their address space, while the hoster uses 10.0.0.0/24 for their physical network.

If the hoster is using Hyper-V hosts running Windows Server 2012, your servers can keep their existing IP addresses in the 192.168.0.0/16 address space when their workloads are moved into the hoster's cloud. This means that your existing clients, which are used to accessing physical servers located on the 192.168.0.0/16 subnet, will still be able to do so with no modifications needed to your routing infrastructure, management platform, or network

security policies. All that is required is that a gateway be used to connect the physical networks where your clients and the resources they need reside with the virtual network on which your virtualized workloads are running on the hoster's physical network.

The ability of Hyper-V Network Virtualization to preserve your network infrastructure (addressing and subnet scheme) allows existing services to continue to work while being unaware of the physical location of the subnets. The way this works is that network virtualization enables you to assign two different IP addresses to each virtual machine running on a Windows Server 2012 Hyper-V host. These two addresses are:

- **Customer Address (CA)** This is the IP address that the server had when it resided on the customer's premises before it was migrated into the cloud. In the above example, this might be the 192.168.33.45 address for a particular server that the customer wants to move to the cloud.
- **Provider Address (PA)** This is the IP address assigned by the cloud provider to the server once the server has been migrated to the provider's data center. In the above example, this might be 10.44.2.133, or some other address in the 10.0.0.0/24 address space.

The CA for each virtual machine is mapped to the PA for the underlying physical host on which the virtual machine is running. Virtual machines communicate over the network by sending and receiving packets in the CA space. The virtual machine's packets are then encapsulated into new packets that have a PA as source and destination address so they can be routed over the hoster's physical network. The standards-based Network Virtualization Generic Routing Encapsulation (NVGRE) protocol is used by Hyper-V Network Virtualization in Windows Server 2012 to encapsulate the virtual machine's packet inside a new packet. The header of this new packet has the appropriate source and destination PA, in addition to the virtual subnet ID, which is stored in the Key field of the GRE header. The virtual subnet ID in the GRE header allows hosts to identify the customer virtual machine for any given packet even though the PAs and the CAs on the packets may overlap. In other words, Hyper-V Network Virtualization keeps track of CA-to-PA mappings to enable hosts to differentiate packets for virtual machines of different customers. All virtual machines on the same host can therefore share a single PA, which helps increase network scalability because you only need as few as a single IP address per host, which lowers the burden on switches in the hoster's network.

The result is that Hyper-V Network Virtualization allows the hoster to run multiple customer virtual networks on top of a single underlying physical network in much the same way as server virtualization lets you run multiple virtual servers on a single physical server. Network virtualization isolates each virtual network from every other virtual network so that each virtual network has the illusion that it is a completely separate network. Multiple customers can even use the exact same addressing scheme for their virtual networks; customer networks will be fully isolated from one another and will function as if each network is the only one present with that particular addressing scheme.

Hyper-V Network Virtualization also makes it easier for large enterprise to move server workloads between multiple data centers where overlapping addressing schemes exist between these data centers. Hyper-V Network Virtualization thus provides increased virtual machine mobility across data centers, hosting provider clouds, and Windows Azure.

Hyper-V Network Virtualization enhancements in Windows Server 2012 R2

While Windows Server 2012 provided the base functionality for implementing network virtualization, Windows Server 2012 R2 includes some new features and enhancements that not only make network virtualization easier to implement and manage but also provide customers with a more comprehensive and integrated SDN solution. This section will briefly examine some of these enhancements, and later when this book is revised and expanded around RTM we'll go into greater depth concerning some of them.

One key enhancement with Hyper-V Network Virtualization in Windows Server 2012 R2 is that it can now dynamically learn the IP addresses on the virtual machine networks. This improvement provides several new benefits such as increasing the high availability options available when deploying a network virtualization solution. For example, with Hyper-V Network Virtualization in Windows Server 2012 R2 you can now use guest clustering inside a virtual network, something you couldn't do before using Hyper-V Network Virtualization in Windows Server 2012. Another benefit is that you can now deploy domain controllers, DNS servers, and DHCP servers as virtual machines on your virtual network running on top of your cloud hoster's physical network infrastructure.

Hyper-V Network Virtualization in Windows Server 2012 R2 also includes several performance enhancements over how this technology worked in Windows Server 2012. For example, you now have the option to be able to load balance NVGRE traffic across multiple NICs. This means that customers who move their workloads into a hoster's cloud now have new options for load-balancing network traffic and providing failover capability to ensure the availability of the virtual network at all times.

Another performance improvement with Hyper-V Network Virtualization in Windows Server 2012 R2 involves work Microsoft is doing with network hardware vendor partners to help bring new network adapter hardware to market that has the capability of offloading NVGRE processing from the host system's processor to the network adapter. These NVGRE Task Offload Enabled NICs will soon be available and should provide significantly better performance for Hyper-V Network Virtualization solutions over current network adapters that don't let you offload NVGRE processing from the host system's processors to the adapter. Emulex is one such network adapter hardware vendor, and you can read their analysis of the performance gains that can be achieved using NVGRE Task Offload Enabled NICs on their blog at <http://o-www.emulex.com/blogs/labs/2013/06/03/benefits-network-virtualization-offload-technologies-optimize-performance-nvgre/>.

The architecture of the Hyper-V Extensible Switch has also been modified in Windows Server 2012 R2 to provide new functionality for customers that implement Hyper-V Network Virtualization solutions. Microsoft introduced the Hyper-V Extensible Switch in Windows Server 2012 to provide new capabilities for tenant isolation, traffic shaping, protection against malicious virtual machines, and hassle-free troubleshooting. The Hyper-V Extensible Switch was also designed to allow third parties to develop plug-in extensions to emulate the full capabilities of hardware-based switches and support more complex virtual environments and solutions. It does this by allowing custom Network Driver Interface Specification (NDIS) filter drivers (called *extensions*) to be added to the driver stack of the virtual switch. This means that networking independent software vendors (ISVs) can create extensions that can be installed in the virtual switch to perform different actions on network packets being processed by the switch.

The Hyper-V Extensible Switch supports three kinds of extensions:

- **Capturing extensions** These can capture packets to monitor network traffic but cannot modify or drop packets.
- **Filtering extensions** These are like capturing extensions but also can inspect and drop packets.
- **Forwarding extensions** These allow you to modify packet routing and enable integration with your physical network infrastructure.

The Hyper-V Extensible Switch also lets you use the built-in `Wfpplwfs.sys` filtering extension of the Windows Filtering Platform (WFP) to intercept packets as they travel along the data path. Networking ISVs can use this functionality to develop applications that can perform packet inspection on a virtual network.

In Windows Server 2012, however, the Hyper-V Extensible Switch was layered above Hyper-V Network Virtualization functionality as shown on the left in Figure 5-8. This meant that capturing, filtering, or forwarding extensions installed in the switch could only see the CA packets, that is, the packets that the virtual machine sees on its virtual network. The extensions could not see or manipulate the PA packets, that is, the packets that the Hyper-V host sees on the hosting provider's underlying physical network.

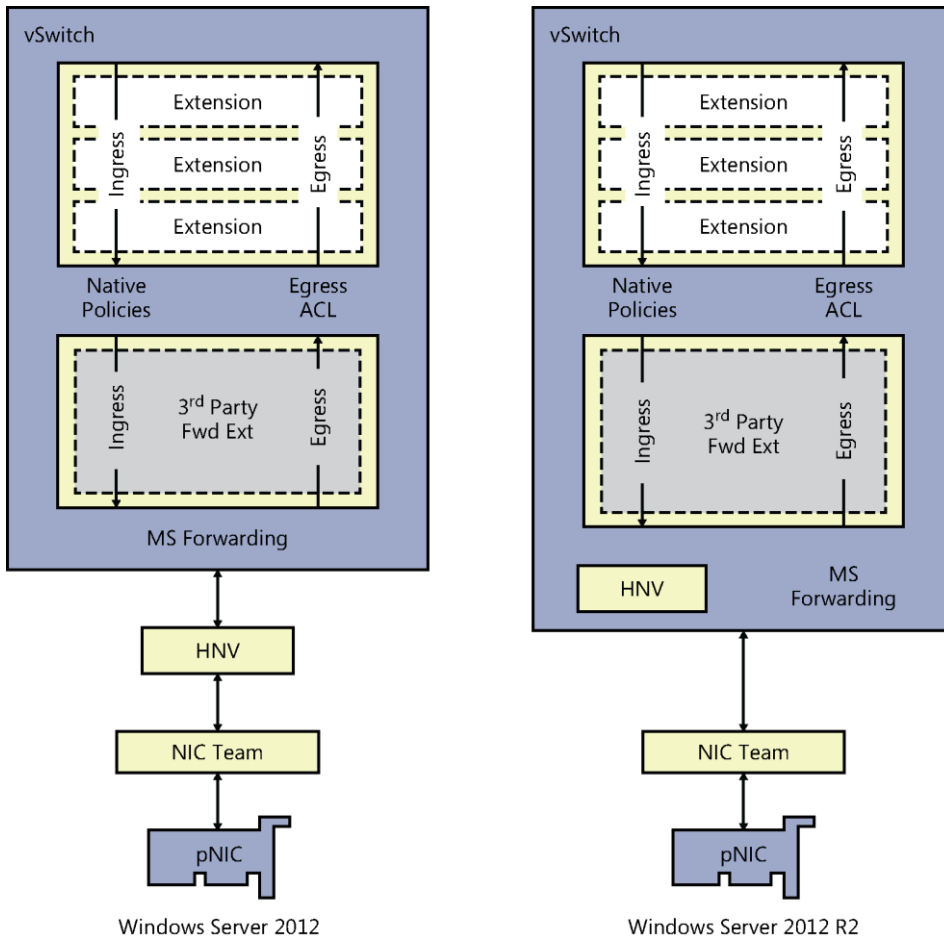


FIGURE 5-8 A comparison of the architecture of the Hyper-V Extensible Switch in Windows Server 2012 and Windows Server 2012 R2.

However, in Windows Server 2012 R2 network virtualization functionality now resides in the Hyper-V Extensible Switch, as shown on the right in Figure 5-8. This means that third-party extensions can now process network packets using either CA or PA addresses as desired. This enables new types of scenarios, such as hybrid forwarding, whereby Hyper-V Network Virtualization forwards the network virtualization traffic while a third-party extension forwards non-network virtualization traffic. Another possibility might be a networking ISV developing a firewall application that drops certain kinds of packets on the customer's network and other types of packets on the provider's network. It also allows third-party ISVs to use the Hyper-V Extensible Switch to implement their own network virtualization solutions using Hyper-V instead of needing to use the in-box Hyper-V Network Virtualization approach from Microsoft.

Finally, if you wanted to implement a network virtualization solution using Hyper-V in Windows Server 2012, you needed to use third-party gateway products to do this. That's because Windows Server 2012 doesn't include an in-box gateway, and a gateway is needed to provide connectivity between virtual machines running on the virtual network and resources on physical networks at local or remote sites. The result is that Hyper-V Network Virtualization by itself in Windows Server 2012 creates virtual subnets that are separated from the rest of the network the way islands are separated from the mainland.

Beginning with Windows Server 2012 R2 however, Hyper-V Network Virtualization now includes an in-box component called Windows Server Gateway (WSG), a virtual machine-based software router that allows cloud hosters to route data center and cloud network traffic between virtual and physical networks including the Internet. WSG can be used to route network traffic between physical and virtual networks at the same physical location or at multiple different physical locations. This allows organizations to implement new kinds of scenarios. For example, if your organization has both a physical network and a virtual network at the same location, you can now deploy a Hyper-V host configured with a WSG virtual machine and use it to route traffic between your virtual and physical networks. As a second example, if your virtual networks reside in a hoster's cloud built using Hyper-V in Windows Server 2012 R2, the hoster can now deploy a WSG that allows you to create a VPN connection between your VPN server and the hoster's WSG to allow your users to connect to your organization's virtual resources in the cloud using the VPN connection.

WSG is fully integrated with Hyper-V Network Virtualization in Windows Server 2012 R2 and allows routing of network traffic even when multiple customers are running tenant networks in the same datacenter. To deploy WSG, you should use a dedicated Hyper-V host that runs only WSG and no other virtual machines. You can also deploy WSG on a failover cluster of Hyper-V hosts to create a highly available WSG solution to provide failover protection against network outages or hardware failure.

As Figure 5-9 shows, the WSG of Hyper-V Network Virtualization in Windows Server 2012 R2 can be used to implement three types of gateway solutions:

- Implementing a multitenant-aware VPN for site-to-site connectivity, for example, to allow an enterprise to span a single virtual network across multiple data centers in different geographical locations.
- Performing multitenant-aware network address translation (NAT) for Internet access from a virtual network.
- Providing a forwarding gateway for in-data center physical machine access from a virtual network.

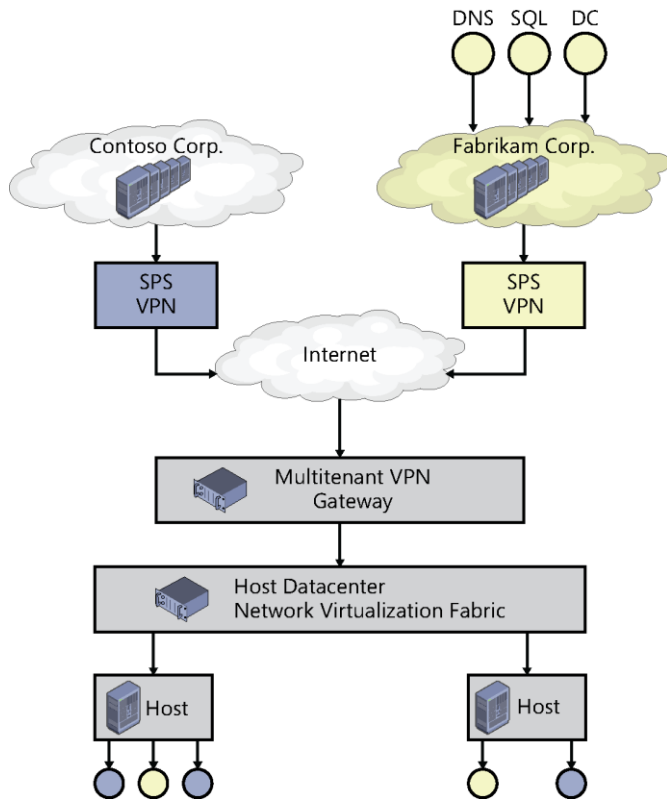


FIGURE 5-9 Windows Server Gateway implements three types of solutions.

There's more!

Because this book is short and is based on the preview release of Windows Server 2012 R2 and we want to get this information out to you, we don't have the time or space to cover all the networking improvements in Windows Server 2012 R2. When we revise this book around RTM, however, we'll include expanded coverage of these and other improvements, and at greater depth as well. Meanwhile, let's now move on to the final chapter and examine some other miscellaneous improvements in Windows Server 2012 R2.

Other enhancements

There are a lot more new features and enhancements in Windows Server 2012 R2 than the ones we've covered in the previous chapters. But since time and space are limited in this short book based on the Preview release of the platform, we're only going to look at what's new in the following features in this penultimate chapter:

- Internet Information Services (IIS)
- Remote Desktop Services (RDS)
- Group Policy
- Workplace Join

IIS 8.5

Microsoft positioned Windows Server 2012 as a highly scalable and elastic platform for cloud computing. IIS 8.0 in Windows Server 2012 introduced a number of new capabilities that were beneficial to a wide spectrum of customers ranging from an enterprise hosting line of business (LoB) applications or a cloud hosting provider managing a multitenant public cloud. Some of the significant improvements made to IIS 8 in Windows Server 2012 included:

- **NUMA-aware scalability** Non-Uniform Memory Architecture (NUMA) was designed to overcome the scalability limits of the traditional symmetric multiprocessing (SMP) architecture, where all memory access happens on the same shared memory bus. SMP works well when you have a small number of CPUs, but it doesn't when you have dozens of them competing for access to the shared bus. NUMA alleviated such bottlenecks by limiting how many CPUs could be on any one memory bus and connecting them with a high-speed interconnection. To utilize these capabilities, IIS in Windows Server 2012 introduced NUMA-aware scalability, which worked by intelligently affinitizing worker processes to NUMA nodes. On NUMA-aware hardware, IIS will try to assign each worker process in a web garden to a different NUMA node to achieve optimal performance.
- **Server Name Indication** Before Windows Server 2012, you could use host headers in IIS to support hosting multiple HTTP websites using only a single shared IP address. Moreover, if you wanted these websites to use HTTPS, then you had a problem because you couldn't use host headers with HTTPS since IIS didn't support that. Instead, you had to assign multiple IP addresses to your web server and bind a different IP address to each HTTPS site, which incurred a lot of management

overhead for IIS administrators. However, IIS 8 in Windows Server 2012 supported Server Name Indication (SNI), which allowed a virtual domain name to be used to identify the network end point of an SSL/TSL connection. This meant IIS could host multiple HTTPS websites, each with their own SSL certificate, bound to the same shared IP address. SNI thus provided increased scalability for web servers hosting multiple SSL sites, and it helped cloud hosting providers to better conserve the dwindling resources of their pool of available IP addresses.

- **SSL central store** Before Windows Server 2012, managing SSL certificates on servers in IIS web farms was time consuming because the certificates had to be imported into every server in the farm, which made scaling out a farm by deploying additional servers a complex task. Replicating certificates across IIS servers in a farm was further complicated by the need to manually ensure that certificate versions were in sync with each another. IIS in Windows Server 2012 solved this problem by introducing a central store for storing SSL certificates on a file share on the network instead of in the certificate store of each host.
- **CPU throttling** IIS in Windows Server 2012 allowed you to configure an application pool to throttle the CPU usage so that it could not consume more CPU cycles than a user-specified threshold. You could also configure IIS to throttle an application pool when the system was under load, and this allowed your application pool to consume more resources than your specified level when the system was idle because the Windows kernel would only throttle the worker process and all child processes when the system came under load.
- **Application Initialization** When users try to open a website in their web browser and then have to wait for several seconds or longer for the site to respond, they get frustrated. Before Windows Server 2012, the delay that occurred when a web application was first accessed was because the application needed to be loaded into memory before IIS could process the user's request and return a response. With the introduction of Application Initialization in IIS 8, however, application pools could now be prestarted instead of having to wait for a first request to arrive for a web application in the pool. Administrators could choose which applications should be preloaded on their IIS servers, and IIS could even be configured to return a static "splash page" while the application was being initialized so the user felt the website being accessed was responding instead of failing to respond.
- **Dynamic IP address filtering** Before Windows Server 2012, IIS could use static IP filtering to block requests from specific clients. This functionality was of limited usefulness however, since it meant that you had to first discover the IP address of the offending client and then manually configure IIS to block that address. Also, IIS offered no choice as to what action it would take when it blocked a client; an HTTP 403.6 status message was always returned to the offending client. IIS 8, however, introduced a new capability called dynamic IP address filtering, which allowed you to configure an IIS server to block access for any IP address that exceeded a specified

number of concurrent requests or exceeded a specified number of requests within a given period of time. You could also configure how IIS responded when it blocked an IP address; for example, by aborting the request instead of returning HTTP 403.6 responses to the client.

By comparison, the improvements made to IIS 8.5 in Windows Server 2012 R2 are targeted at a different audience, namely, website administrators who are highly focused on scalability and manageability. Such improvements are increasingly important in today's world. In fact, both Microsoft's public website and the Windows Azure platform are deeply tied to IIS.

One of the key scalability goals Microsoft has for IIS 8.5 is enabling cloud-hosting providers to host more sites on a single IIS server. In Windows Server 2012 and earlier, when IIS starts up on a host the Windows Activation Service (WAS) loads the entire configuration for IIS. This configuration can be very large if several thousand sites are being hosted on the server, which is typical in many hoster environments. Because of this, loading the IIS configuration can consume a lot of memory on the server, which can impact the performance of other options such as initializing worker processes for web applications.

To address this issue, the WAS component has been redesigned in IIS to better handle large configurations and improve the memory efficiency of the IIS startup process. In addition, the HTTP protocol stack (Http.sys) now uses a single catch-all request queue and binding to be used by WAS for initializing the worker processes associated with each of the sites running on the IIS server. The result of this change is that when WAS starts up, it no longer has to create thousands of request queues and bindings in Http.sys—one for each of the thousands of worker processes associated with each of the thousands of sites running on the server. The WAS component can now examine a client request in this queue and determine which site and worker process should handle that request. WAS then creates a queue for the request, registers the binding, and spins up the worker process for the site to start the site.

This new functionality is called Dynamic Site Activation, and it addresses the issue of being "config-bound" which hosters who run large numbers of sites on IIS servers can experience. By default, if an IIS 8.5 server is hosting 100 or more sites, this new WAS functionality is used. If the server is hosting fewer than 100 sites, however, the old method of creating separate queues and bindings for each site is still used since such a scenario has only a relatively small configuration that can load quite easily without undue memory being needed. This functionality can be tuned by modifying the *dynamicRegistrationThreshold* parameter using the IIS Configuration Editor, as shown in Figure 6-1. Note that you must restart WAS after changing this parameter for the change to take effect.

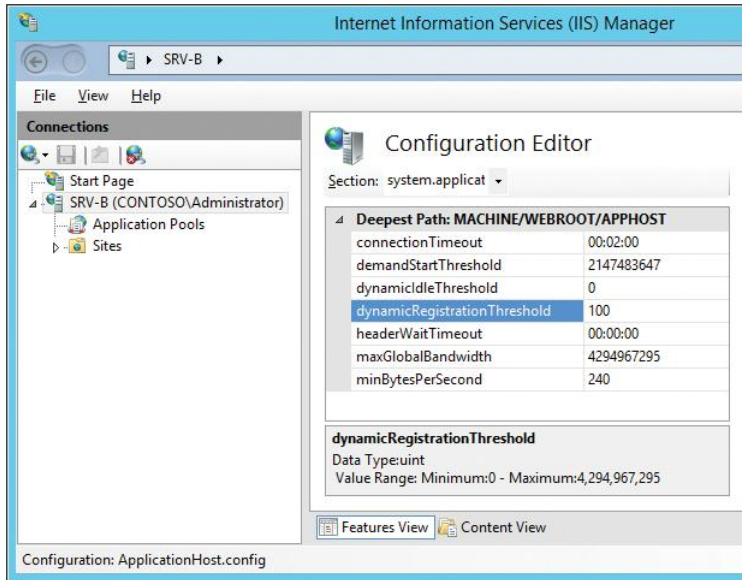


FIGURE 6-1 The Configuration Editor configures Dynamic Site Activation.

A second issue, however, that can be experienced on IIS servers hosting large numbers of sites has to do with cold requests. A cold request is a request that comes for a worker process that has not yet been started. The cold request has to wait for IIS to initialize, for the appropriate framework (for example .NET or PHP) to initialize, for the associated site to initialize, and so on. As a result, the response time to cold requests with previous versions of IIS sometimes left something to be desired.

This issue was partly addressed with IIS 8 in Windows Server 2012 where static sites responded much more quickly (typically a few hundred milliseconds) to cold requests than for IIS 7 in Windows Server 2008 R2. However, there was only a small performance improvement in IIS 8 for dynamic sites with managed code, with such sites typically taking several seconds to start in response to cold requests, largely because of the time needed to load the application framework needed by the site. From the perspective of the hoster's customers however, taking several seconds to launch their LoB web application can be viewed negatively as unacceptable and poor performance.

To improve the start time for dynamic sites in response to cold requests, a module called Application Initialization was included in IIS 8 in Windows Server 2012 that allowed the administrator to preload the application framework needed for a dynamic site so that the site could respond to a cold request in a few hundred milliseconds instead of several seconds. However, this module is not useful for hosters because such preloaded application frameworks consume additional memory for each site configured to use them. Since the IIS servers of a hoster are typically hosting thousands of sites and are usually memory-bound (assume at least 100 MB needed per dynamic site) such preloading application frameworks for all sites hosted on a server just wasn't feasible. IIS can partly address this problem by killing idle worker

processes after a default of 20 minutes of inactivity, and WAS can halve this time interval if it determines that memory pressure has reached 80 percent on the server.

IIS 8.5 in Windows Server 2012 R2 now takes a different perspective on how to address the problem of cold request delays for dynamic sites. A new feature called Idle Worker Process Page-out that saves memory by allowing an idle worker process to be paged to disk so that the worker process is removed from memory. The page-out feature can be made to perform even better by utilizing a solid state disk (SSD) for the paging file of an IIS server. The result is that memory pressure can now be greatly reduced for servers hosting thousands of dynamic sites and the response time to cold requests for these sites can be significantly improved.

Idle Worker Process Page-out is not enabled by default in IIS 8.5. Instead, worker processes simply terminate idle worker processes. You can enable Idle Worker Process Page-out by opening the Advanced Settings dialog for an application pool in IIS Manager and changing the value of the Idle Time-out Action setting from Terminate to Suspend, as shown in Figure 6-2. You can also enable Idle Worker Process Page-out at the server level, and while doing this won't affect the configuration of existing sites, any new sites that are created will inherit this setting.

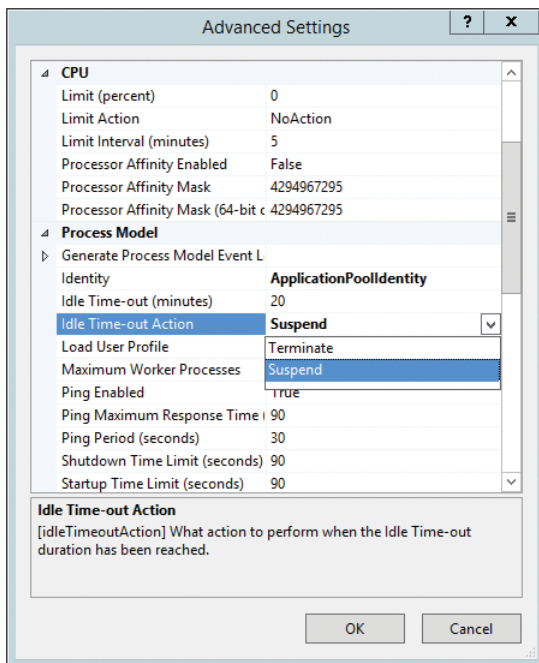


FIGURE 6-2 You can enable Idle Worker Process Page-out through Advanced Settings.

Another set of enhancements made in IIS 8.5 is in the area of logging. IIS logging can be used for monitoring and tracking IIS activity and for troubleshooting and debugging IIS when problems occur. One frequent request from customers who use IIS is for being able to log more kinds of IIS activities. To address such requests, IIS 8.5 now allows administrators to specify that additional custom fields be logged by IIS in addition to the standard W3C logging

fields. This is illustrated by Figure 6-3, which shows the list of additional sources that can be logged for Request Header. Additional custom fields can also be added for Response Header and for Server Variables. In addition to the prepopulated fields for the variables and headers, the administrator can manually specify custom sources.

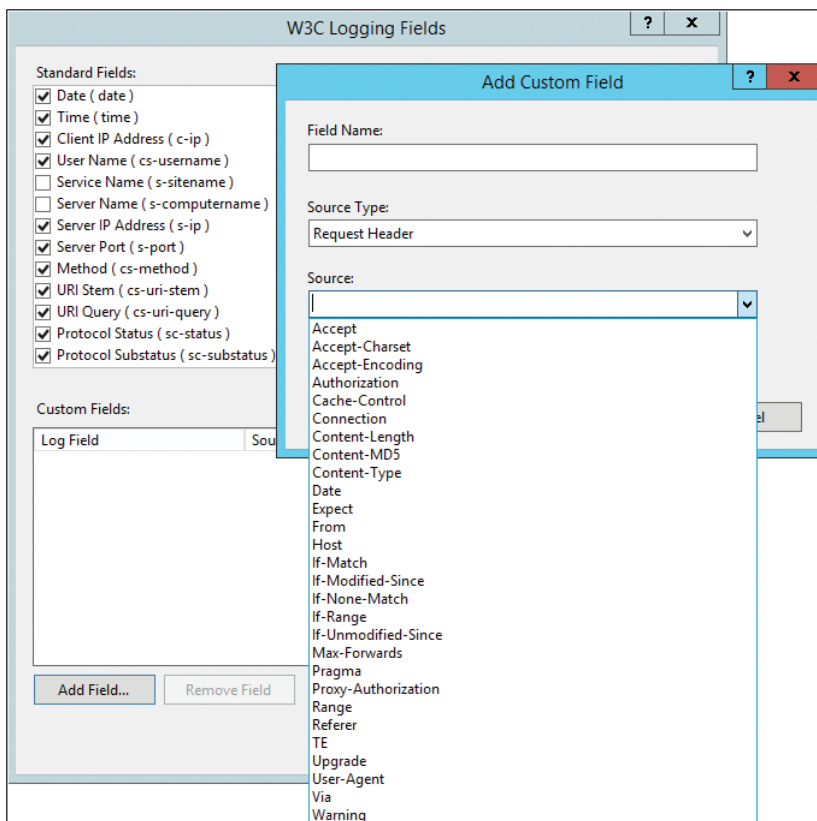


FIGURE 6-3 You can add custom fields for IIS logging.

A common scenario where this could come in handy is when the web server is front-ended by networking equipment such as a proxy or load balancer. In such a situation, the IIS logs cannot accurately track client IP addresses because the IP captured by Http.sys is the proxy or load balancer's IP. Network hardware that proxy HTTP requests usually have the ability to store the original client IP in the custom header X-FORWARDED-FOR, and so the administrator can record it as a custom Response Header and thus record the original IP of the clients.

Note that the custom logging is limited to a total of 64 KB, and if the administrator defines custom fields which exceed that, they will be truncated. This means that care must be taken when deciding which fields to collect. Note also that when enhanced logging is enabled, IIS creates log files with _x appended to their name to indicate that these files are enhanced logging fields.

RDS enhancements

RDS, which was once called Terminal Services, is a server role that provides capabilities that enable users to access session-based desktops, virtual machine-based desktops, or remote applications running in the data center from both within a corporate network and from the Internet. RDS can deliver a rich-fidelity desktop or application experience and allows remote users to securely connect from either managed or unmanaged devices. RDS is also the foundation for delivering the Microsoft Virtual Desktop Infrastructure (VDI) solution on the Windows Server platform.

A big change introduced in Windows Server 2012 was how RDS could be easily deployed in different kinds of scenarios. When you added the RDS role in Windows Server 2012, you first had a choice of choosing two deployment options:

- **Standard** This option provides you with more flexibility concerning how you deploy different RDS role services to different servers and is intended for production environments.
- **Quick Start** This option deploys all the RDS role services required on a single computer using mostly the default options and is intended mainly for test environments.

Once you've selected the appropriate deployment option for your environment, you're next choice is which type of RDS scenario you want to implement:

- **Virtual machine-based desktop deployment** Lets remote users connect to virtual desktops running on a Remote Desktop Virtualization Host to access applications installed on these virtual desktops (and also RemoteApp programs if session virtualization is also deployed).
- **Session-based desktop deployment** Lets remote users connect to sessions running on a Remote Desktop Session Host to access session-based desktop and RemoteApp programs.

That's not all, though as there were a number of other significant enhancements to RDS in Windows Server 2012 such as:

- **RemoteFX enhancements** RemoteFX was first introduced in Windows Server 2008 R2 as a way of delivering a full Windows experience over the RDP across a wide variety of client devices. RemoteFX uses host-side rendering to enable graphics to be rendered on the host instead of the client by utilizing the capabilities of a RemoteFX-capable graphics processing unit (GPU) on the host. RemoteFX vGPU uses GPU virtualization to expose a virtual graphics device to a virtual machine running on a RemoteFX-capable host so that multiple virtual desktops could share the single GPU on the host. RemoteFX functionality was enhanced for RDS in Windows Server in a lot of ways with support included for multitouch gestures and manipulations in remote sessions; improved multimonitor support; dynamic adaptation to changing network

conditions by using multiple codecs to optimize how content is delivered; optimization of performance when sending traffic over a wide area network (WAN) by choosing between TCP or UDP (called RemoteFX for WAN); integration throughout the RDS role services instead of being installed as its own separate role service; and more.

- **Enhanced USB Redirect** USB redirection from within RDS was first introduced in Windows 7 Service Pack 1 and Windows Server 2008 R2 Service Pack 1 to support VDI scenarios. USB Redirect occurs at the port protocol level and enables redirection of a wide variety of different types of universal serial bus (USB) devices, including printers, scanners, webcams, Voice over Internet Protocol (VoIP) headsets, and biometric devices. RDS in Windows Server 2012 enhanced this capability with support for USB Redirect for Remote Desktop Session Host (RD Session Host) to enable new kinds of scenarios for businesses that implement session virtualization solutions. In addition, USB Redirect for Remote Desktop Virtualization Host (RD Virtualization Host) no longer requires installing the RemoteFX 3D Video Adapter on the virtual machine in order to work.
- **User Profile Disks** Before Windows Server 2012, preserving user state information for sessions and virtual desktops required using Windows roaming technologies like roaming user profiles (RUP) and Folder Redirection (FR). But implementing RUP and FR added more complexity to deployments. RDS in Windows Server 2012 simplifies session-based and VDI deployments with the introduction of User Profile Disks, which store user data and settings for sessions and virtual desktops in a separate VHD file that can be stored on a network share.

RDS in Windows Server 2012 R2 isn't a radical change from what it was in Windows Server 2012, but it does have some exciting new features and enhancements that can make managing remote sessions easier, provide a richer user experience, and allow you to get the best possible value from VDI.

For example, in previous versions of RDS, when you opened a RemoteApp program and dragged it around on the desktop, only an outline of the program's window was displayed. And if you hovered the mouse pointer over the taskbar icon of a RemoteApp program, no thumbnail preview was displayed, only the generic icon associated with the program. In Windows Server 2012 R2 however, dragging around a RemoteApp program now drags the entire contents of the program's window, and hovering over the taskbar icon of the RemoteApp program displays a live thumbnail preview of the program, as shown in Figure 6-4.

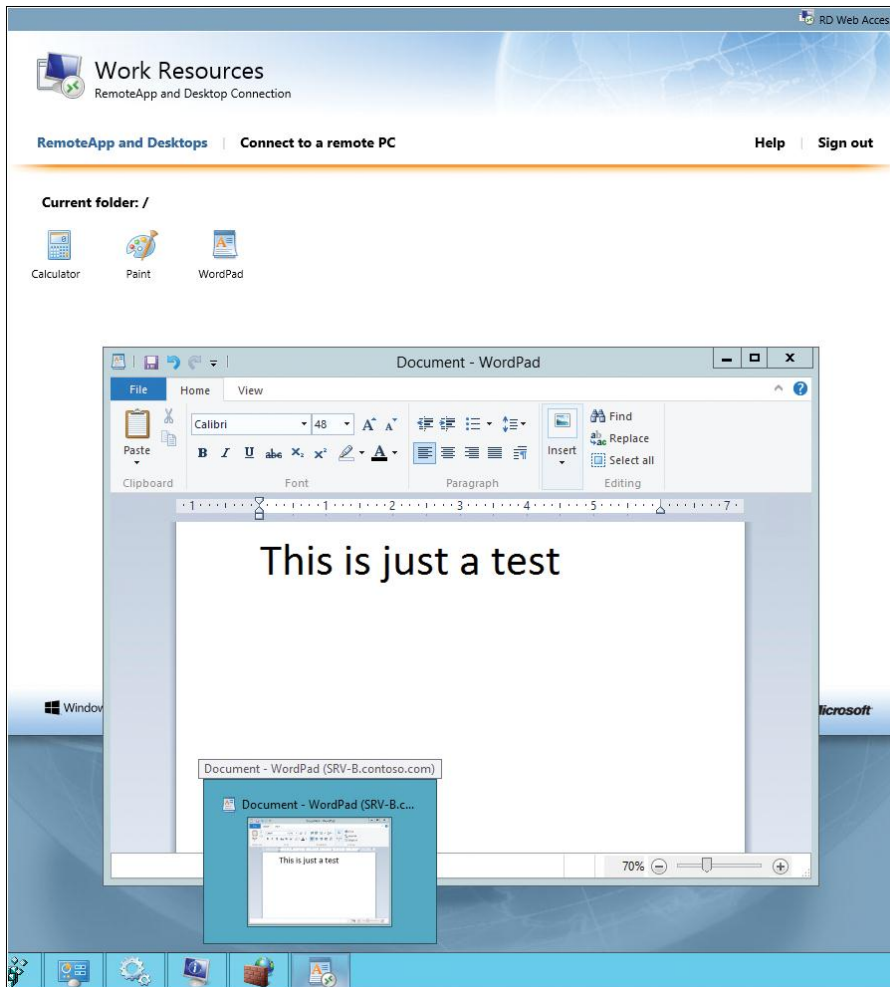


FIGURE 6-4 Hovering over the taskbar icon of a RemoteApp program now displays a live thumbnail preview of the program.

RDS in Windows Server 2012 R2 also includes a number of improvements in the area of display control. For example, RDS now allows you to change the display resolution in a remote desktop session and have the session window resize itself properly so you won't have to use the scroll bars to get to the taskbar and charms. This also works in multimonitor scenarios as well. And on tablet devices, a remote desktop session will now automatically rotate as you rotate the device. RDS in Windows Server 2012 R2 also includes DirectX 11.1 support and improved codec performance. Microsoft continues to improve the graphics and video performance over remote desktop sessions and in VDI scenarios with numerous improvements to DirectX and codecs.

Another new feature of RDS in Windows Server 2012 R2 is called Quick Reconnect. In previous versions of Windows Server, reconnecting to a remote desktop session over a slow or

unreliable network connection could sometimes take up to a minute. With RDS in Windows Server 2012 R2, however, reconnecting to disconnected sessions will typically take no more than five seconds even over slow connections. The user will also no longer be stuck seeing that uninformative grey screen when the remote desktop client is trying to reconnect to a previously disconnected session and instead will see a more informative message.

Finally, and perhaps most excitingly, a frequent request from Microsoft customers who use RDS is to add the ability for administrators to shadow the sessions of remote users. Such shadowing capability is now included in Windows Server 2012 R2. To see how this works, Figure 6-5 shows the connections in Server Manager to a Remote Desktop Session Host running Windows Server 2012 R2. In previous versions of RDS, if you right-clicked a remote session you only got three options: Disconnect, Send Message, and Log Off. But in RDS in Windows Server 2012 R2 there is now a fourth context menu option called Shadow, which enables administrators to shadow the session of a connected remote user.

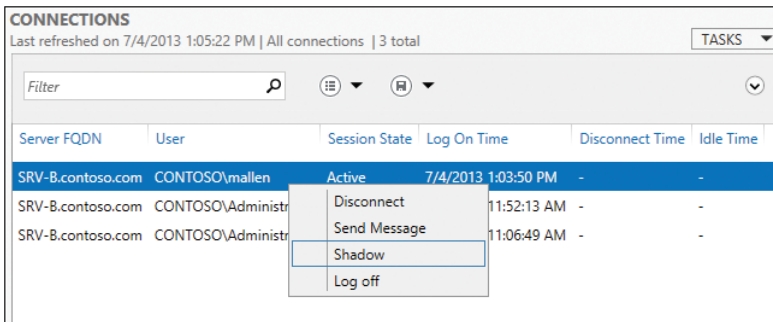


FIGURE 6-5 The new Shadow option for a remote session.

Selecting the Shadow option opens the Shadow dialog shown in Figure 6-6. This dialog offers administrators two options for shadowing the remote session:

- **View** This option allows the administrator only to view what the user is doing in his session.
- **Control** This option allows the administrator to view the user's session and also take control of the session and perform actions on the user's desktop.

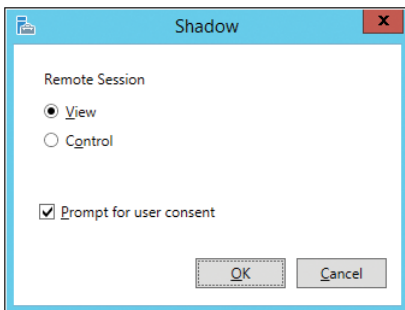


FIGURE 6-6 You can view or control remote sessions.

By default, the Prompt For User Consent checkbox is selected in the Shadow dialog. This means that when the administrator clicks OK to close the dialog, the user's screen dims and a message bar is displayed across their screen, as shown in Figure 6-7.

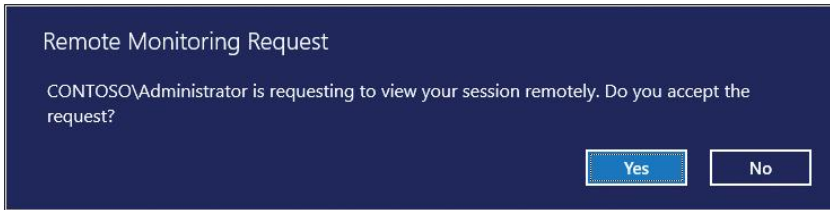


FIGURE 6-7 The user must accept the prompt to allow shadowing of his session.

Let's pause for a few notes at this point:

- By default, Group Policy is configured to require user consent to shadow requests. In many organizations this might be the correct approach to ensure the privacy of employees. Be sure to consult your legal and human resources departments before changing this policy setting.
- To shadow a remote user, the user doesn't need to be connected to a full remote desktop session. Administrators can even shadow individual RemoteApp programs as remote users work with them.
- Shadowing functionality is also built into the Remote Desktop Client client (Mstsc.exe). By including the /shadow parameter and specifying the remote session when you use mstsc to launch the Remote Desktop Client from the command line, an administrator can use the client to view or control the remote user's session.

To continue now with our walkthrough of shadowing, if the user clicks No in the message bar on their screen, the administrator will not be able to shadow the user's session. If she clicks Yes, the message bar disappears and the user's screen is no longer dimmed, and they can continue with their work as usual. Meanwhile, a new window appears on the RDS administrator's desktop showing the appearance of the shadowed user's desktop in real time, as shown in Figure 6-8.

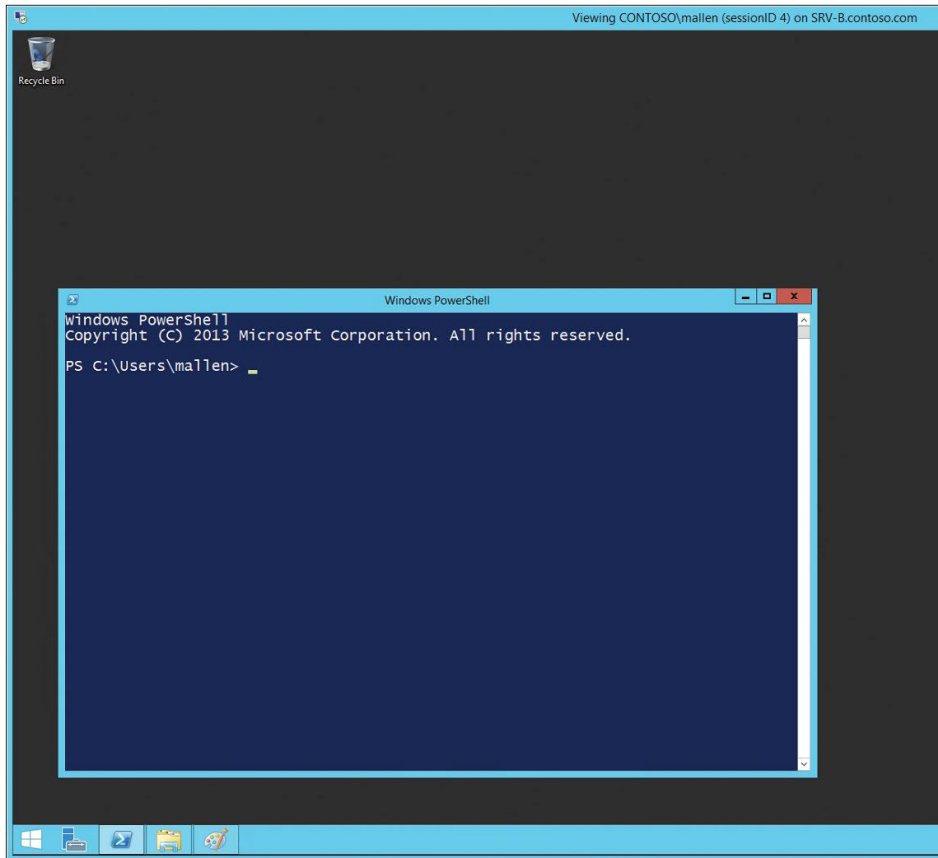


FIGURE 6-8 What shadowing a user's session might look like.

Once the administrator is finished shadowing the session, he can simply close the shadow window.

Group Policy enhancements

Group Policy received a lot of new functionalities in Windows Server 2012 and Windows 8. Some of the more important enhancements included:

- **Remote Group Policy Update** Windows Server 2012 enabled the use of the Group Policy Management Console (GPMC) to refresh the computer and user Group Policy settings (including security settings) on all remote computers in an organizational unit (OU). It did this by scheduling a task that ran an update.exe on the remote computers. You could also use the Invoke-GPUupdate Windows PowerShell cmdlet to automate the remote Group Policy update. These functionalities helped when you needed to push out a new Group Policy setting immediately to users or computers in your environment.

- **Group Policy infrastructure status** You could use the GPMC to view the status of Active Directory and SYSVOL replication for all Group Policy Objects (GPOs) or for a selected GPO. This included viewing the security descriptors, GPO version details, and number of GPOs listed in Active Directory and SYSVOL for each domain controller. This functionality was helpful for monitoring and diagnosing replication issues related to Group Policy at the domain level.
- **Fast Startup** Computers running Windows 8 and later are configured by default to use Fast Startup, a new boot mode that is a hybrid of traditional cold boot and resuming from hibernate. Fast Startup closes all user sessions on shutdown but hibernates the kernel session instead of closing it. This enabled Windows 8 computers to shut down and start up more quickly than in previous versions of Windows. Although Fast Startup can help speed up Group Policy processing, some policy settings or scripts processed during startup or shutdown might not be applied when Fast Startup is enabled.
- **New Group Policy starter GPOs** Windows Server 2012 included two new starter GPOs that can make configuring Group Policy firewall port requirements a lot easier. These starter GPOs are named Group Policy Reporting Firewall Ports and Group Policy Remote Update Firewall Ports. You could import these starter GPOs when you created a new GPO for this purpose.
- **Local Group Policy support for Windows RT** Group Policy can be used to manage devices that run Windows RT. By default, the Group Policy Client service was disabled on Windows RT devices and must be enabled and configured to start automatically using the Services snap-in.

Additionally, there were lots of new policy settings and preferences added to Group Policy in Windows Server 2012 including settings and preferences for Internet Explorer 10 and many other Windows components and capabilities.

As far as enhancements in Windows Server 2012 R2 go, this time around it's more of a fine-tuning of Group Policy capabilities rather than a lot of new functionality. For example, one of the key enhancements in Windows Server 2012 R2 is policy caching, which can significantly reduce the amount of time it takes to process Group Policy on a client. Policy caching works by having the client download the policy from a domain controller and save a copy of the policy to a local store on the client. Then when the next Group Policy processing occurs, the client can apply the policy cached in the local store instead of having to download the policy again from the network.

By speeding up Group Policy processing, policy caching can shorten boot times for clients. This can be especially helpful in scenarios where the network connection experiences latency or is connecting from off-premises over the Internet, for example in a DirectAccess scenario. Note that policy caching only works when Group Policy is running in synchronous mode.

Policy caching is disabled by default in Windows Server 2012 R2. To enable policy caching, configure the following policy setting named Configure Group Policy Caching as shown in Figure 6-9. This policy setting can be found under:

Computer Configuration\Policies\Administrative Templates\System\Group Policy

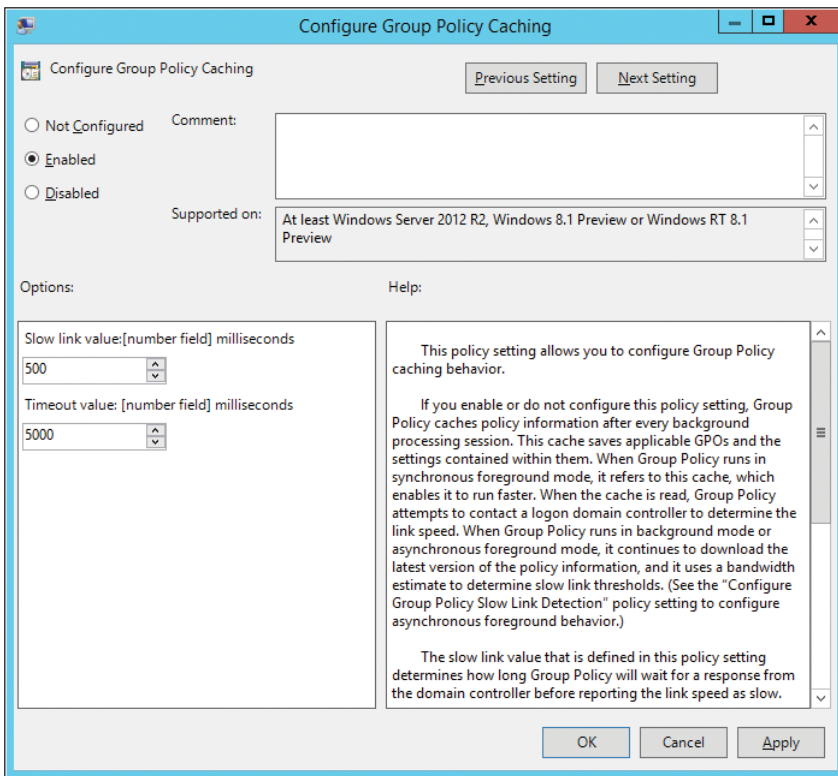


FIGURE 6-9 The new Configure Group Policy Caching policy setting.

Another important area where Group Policy has been enhanced in Windows Server 2012 R2 is in regards to Internet Protocol version 6 (IPv6) support. An increasingly important part of the job of the Windows Server administrator is to prepare the organization's network for migration to IPv6. The reasons for this include the exponential growth of the Internet, the proliferation of mobile devices that need to be able to connect to the corporate network, and the exhaustion of the IPv4 address space. Windows Server 2012 included a number of new IPv6 capabilities such as built-in support for NAT64/DNS64 when implementing DirectAccess, new Windows PowerShell cmdlets that supersede the Netsh.exe command-line utility of earlier Windows versions, and improved Internet connectivity by marking well-known IPv6 Internet resources Windows can't reach as unreachable so that in the future only IPv4 is used for connecting to them.

With Windows Server 2012 R2 however, IPv6 has now made more inroads into Group Policy functionality on the platform. For example, you can now specify an IPv6 address when using item-level targeting with Group Policy Preferences (GPP). This allows you to configure the scope of individual preference items so they apply only to computers that have a specific IPv6 address range. For example, here's how you might use this capability to use a GPO to apply a certain power plan to computers running Windows 7 or later whose IPv6 addresses fall within the address range 2001:DB8:3FA9:/48:

1. Open the GPO in the Group Policy Management Editor and expand Computer Configuration | Preferences | Control Panel Settings | Power Options.
2. Right-click Power Options and select New | Power Plan (at least Windows 7).
3. In the New Power Plan Properties dialog, click the Common tab, select the Item-level Targeting checkbox, and click Targeting.
4. In the Targeting Editor, click New Item | IP Address Range.
5. Select the Use IPv6 checkbox, as shown in Figure 6-10, and specify the IPv6 address and prefix length.

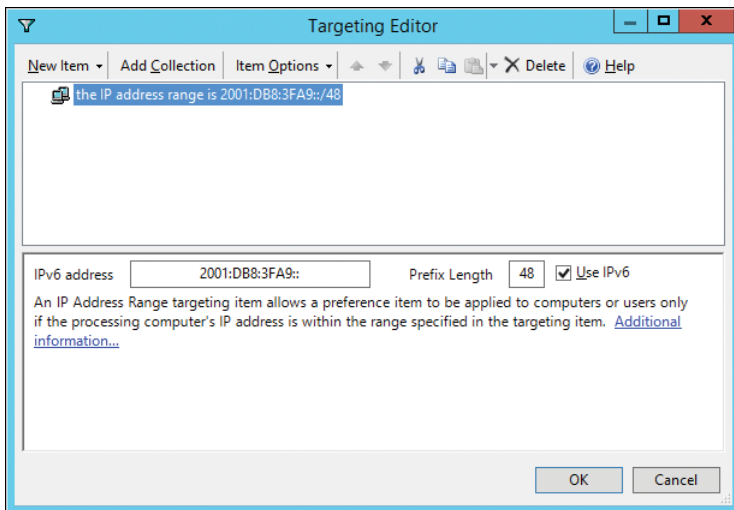


FIGURE 6-10 Item-level targeting now supports IPv6 addresses.

Support for IPv6 has been introduced into other areas of Group Policy as well. For example, when you create a new TCP/IP Printer preference item, you can specify an IPv6 address of the network printer, as shown in Figure 6-11.

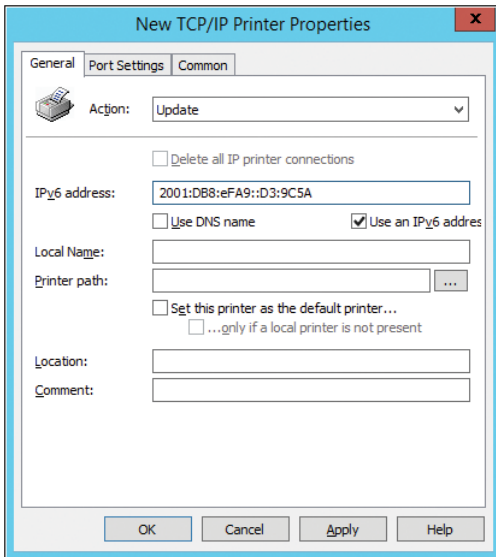


FIGURE 6-11 You can now specify an IPv6 address for a network printer.

Another example of IPv6 support in Group Policy can be found when you create a new VPN Connection preference item, as shown in Figure 6-12.

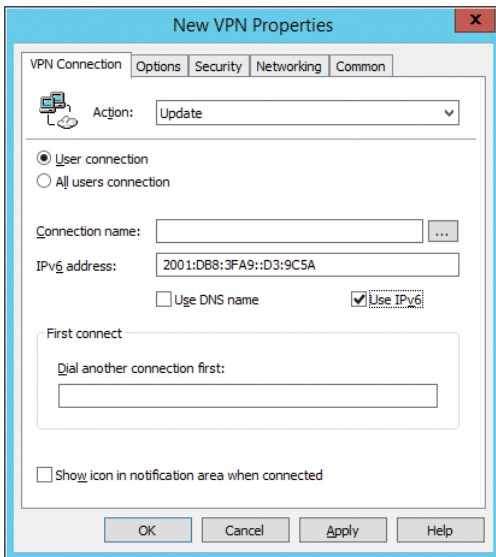


FIGURE 6-12 You can now specify an IPv6 address when creating a new VPN connection.

Other enhancements to Group Policy in Windows Server 2012 R2 include some new policy settings, but we'll leave this topic for now and revisit it in greater depth when this book is revised around RTM.

Workplace Join

One of the key Active Directory enhancements in Windows Server 2012 R2 is called Workplace Join. Basically, what this means is that companies can now provide a Single Sign On (SSO) experience for all workplace-joined devices, which at present includes Windows and iOS devices.

The way it works is that users join their personal devices to their workplace by making their devices known to the company's Active Directory. The goal of doing this is to allow users to be able to easily and securely access resources and services on the company network so they can perform their job better.

Workplace Join associates the device with the user and enables a better user experience by providing a seamless second factor authentication. Users sign in once from any application running on their workplace-joined devices. They are not prompted for credentials by every company application when they are using their workplace-joined devices. This provides both the user and the company with some major benefits in the areas of usability and security, namely:

- Usability is enhanced because the user no longer has to repeatedly enter their credentials into their device to access resources on the company network.
- Security is enhanced because the risks involved in saving passwords with each application on the user's device are avoided.

When a user's device is workplace-joined, the attributes of the device are stored in Active Directory and can be retrieved to issue security tokens for the applications running on the user's device. This enables the company to grant the appropriate rights and permissions for the user's application to securely access company resources and services.

Workplace Join uses the new Device Registration Service (DRS) included in the Active Directory Federation Services (AD FS) role in Windows Server 2012 R2. DFS provisions a device object in Active Directory for each workplace-joined device. DFS also configures a certificate on the user's device to represent the identity of the device. And, by deploying DRS together with the Web Application Proxy, organizations can remotely join user devices to their workplace over an Internet connection.

Workplace Join enables new scenarios for organizations that want to take advantage of bring-your-own-device (BYOD) to enhance user productivity. As Figure 6-13 shows, Workplace Join represents a middle-of-the-road approach to device manageability and security for enterprises.

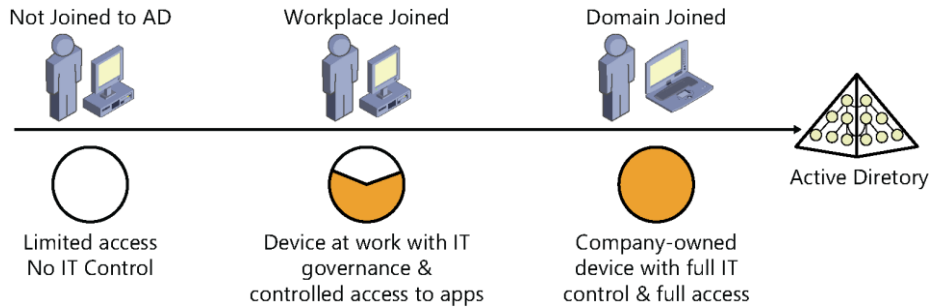


FIGURE 6-13 Workplace Joined represents a middle-of-the-road approach to device manageability and security for enterprises.

Coming soon!

Because this book has been based on the Preview release of Windows Server 2012 R2, and because we've kept the book short so we could get it out to you as quickly as possible, we haven't been able to cover all the exciting new features and improvements in Windows Server 2012 R2. But as RTM approaches, we'll be revising and expanding this title to include discussion of additional new features, and we'll also be going into greater depth on some of these features. So stay tuned for the RTM edition of this book!

Meanwhile, turn now to the Appendix where we've listed some online resources you can check out for additional information concerning what's new in Windows Server 2012 R2.

Additional resources

The following additional resources can be used to learn more about Windows Server 2012 R2:

- Download Windows Server 2012 R2 Preview from the TechNet Evaluation Center at <http://technet.microsoft.com/en-us/evalcenter/dn205286.aspx>
- Windows Server 2012 R2 on the Microsoft Server and Cloud Platform at <http://www.microsoft.com/en-us/server-cloud/windows-server/windows-server-2012-r2.aspx>
- What's New in Windows Server 2012 R2 in the TechNet Library at <http://technet.microsoft.com/en-us/library/dn250019.aspx>
- See also the Windows Server Blog on TechNet at <http://blogs.technet.com/b/windowsserver/>
- You can find videos and slide decks about Windows Server 2012 R2 from Microsoft TechEd North America 2013 on Channel 9 at <http://channel9.msdn.com/Events/TechEd/NorthAmerica/2013>
- Post your questions about Windows Server 2012 R2 to the Windows Server forums on TechNet at <http://social.technet.microsoft.com/Forums/windowsserver/en-US/home?category=windowsserver>

About the author



MITCH TULLOCH is a well-known expert on Windows administration, deployment, and virtualization. He has published hundreds of articles on a wide variety of technology sites and has written more than two dozen books, including the Windows 7 Resource Kit (Microsoft Press, 2009), for which he was lead author; Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter (Microsoft Press, 2010); and Introducing Windows Server 2012 (Microsoft Press, 2012), a free e-book that has been downloaded over a quarter of a million times.

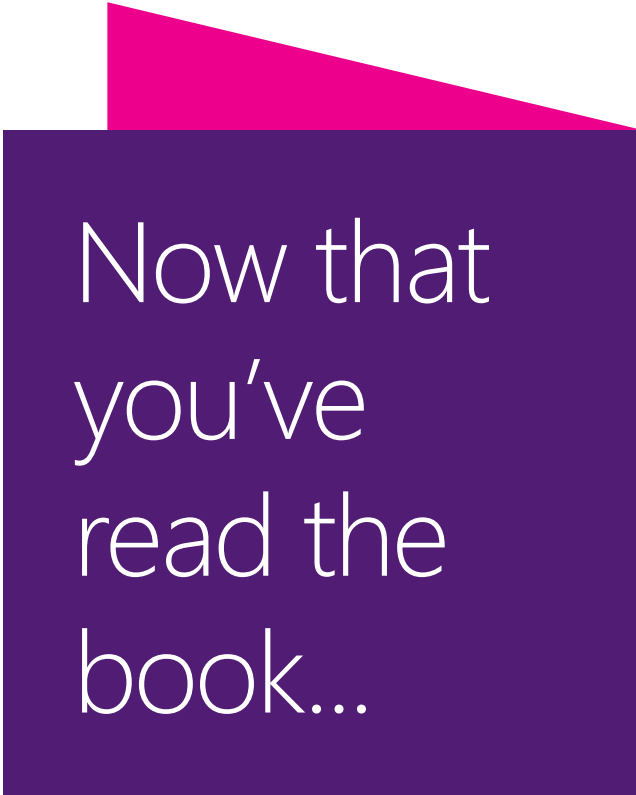
Mitch is also Senior Editor of WServerNews (<http://www.wservernews.com>), the world's largest newsletter focused on system admin and security issues for Windows servers. Published weekly, WServerNews helps keep system administrators up to date on new server and security-related issues, third-party tools, updates, upgrades, Windows compatibility matters, and related issues. With more than 100,000 subscribers worldwide, WServerNews is the largest Windows Server-focused newsletter in the world.

Mitch has been repeatedly awarded Most Valuable Professional (MVP) status by Microsoft for his outstanding contributions to supporting the global IT community. He is an eight-time MVP in the technology area of Windows Server Setup/Deployment.

Mitch also runs an IT content development business based in Winnipeg, Canada, which produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at <http://www.mtit.com>.

You can also follow Mitch on Twitter at <http://twitter.com/mitchtulloch> or friend him on Facebook at <http://www.facebook.com/mitchtulloch>.



Now that
you've
read the
book...

Tell us what you think!

Was it useful?

Did it teach you what you wanted to learn?

Was there room for improvement?

Let us know at <http://aka.ms/tellpress>

Your feedback goes directly to the staff at Microsoft Press,
and we read every one of your responses. Thanks in advance!

