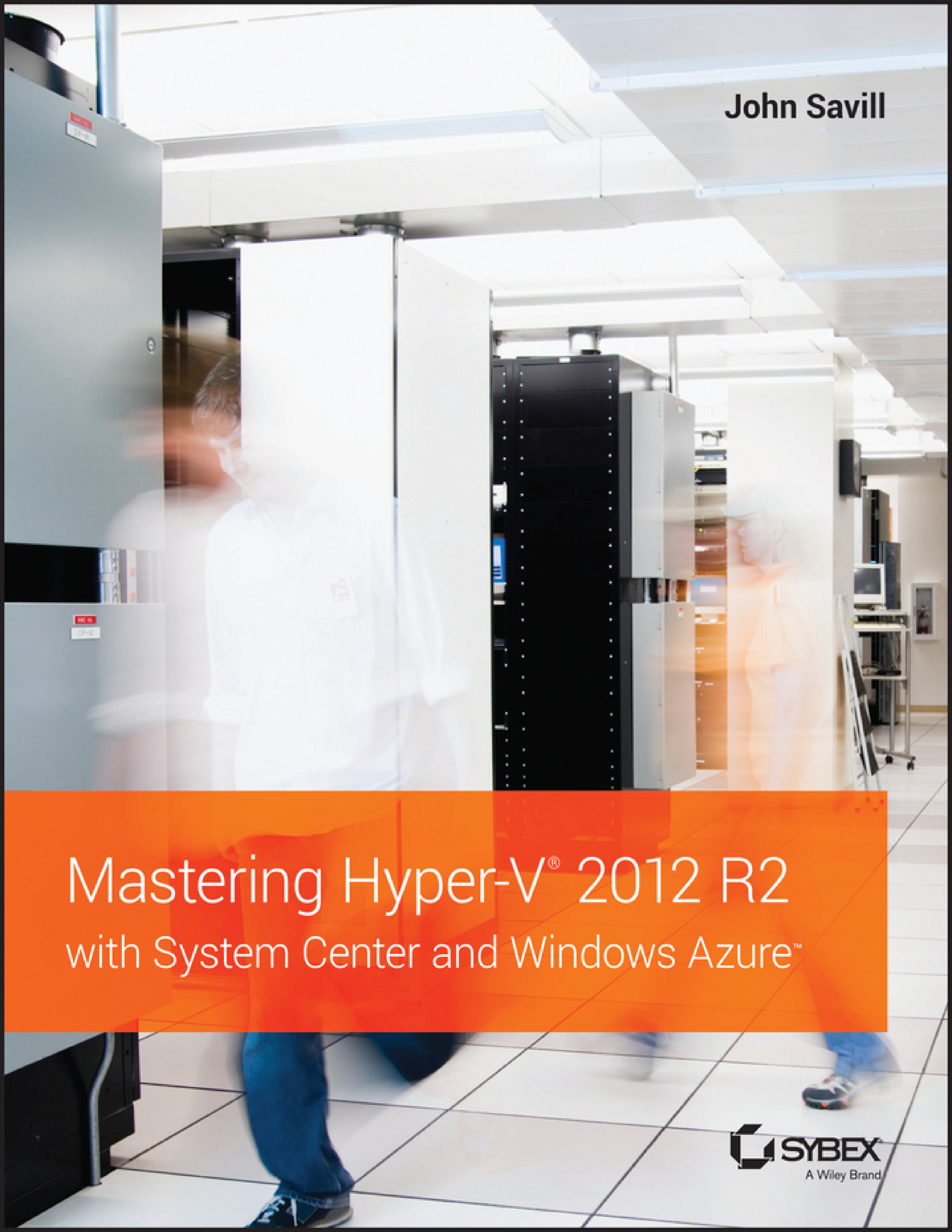

John Savill

Mastering Hyper-V® 2012 R2 with System Center and Windows Azure™

SYBEX
A Wiley Brand

Acquisitions Editor: Mariann Barsolo

Development Editor: Kim Beaudet

Technical Editor: Sean Deuby

Production Editor: Rebecca Anderson

Copy Editors: Judy Flynn and Kim Wimpsett

Editorial Manager: Pete Gaughan

Vice President and Executive Group Publisher: Richard Swadley

Associate Publisher: Chris Webb

Book Designers: Maureen Forys, Happenstance Type-O-Rama; Judy Fung

Proofreader: Rebecca Rider

Indexer: Robert Swanson

Project Coordinator, Cover: Todd Klemme

Cover Designer: Wiley

Cover Image: ©Getty Images, Inc./ColorBlind Images

Published simultaneously in Canada

ISBN: 978-1-118-82818-2

ISBN: 978-1-118-82815-1 (ebk.)

ISBN: 978-1-118-82833-5 (ebk.)

For general information on our other products and services or to obtain technical support, please contact our Customer Care Department within the U.S. at (877) 762-2974, outside the U.S. at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013958305

Dear Reader,

Thank you for choosing *Mastering Hyper-V 2012 R2 with System Center and Windows Azure*. This book is part of a family of premium-quality Sybex books, all of which are written by outstanding authors who combine practical experience with a gift for teaching.

Sybex was founded in 1976. More than 30 years later, we're still committed to producing consistently exceptional books. With each of our titles, we're working hard to set a new standard for the industry. From the paper we print on, to the authors we work with, our goal is to bring you the best books available.

I hope you see all that reflected in these pages. I'd be very interested to hear your comments and get your feedback on how we're doing. Feel free to let me know what you think about this or any other Sybex book by sending me an email at contactus@wiley.com. If you think you've found a technical error in this book, please visit http://sybex.custhelp.com. Customer feedback is critical to our efforts at Sybex.

Best regards,

Chris Webb
Associate Publisher, Sybex

For my wife, Julie, and my children, Kevin, Abby, and Ben.
My everythings.

Acknowledgments

I could not have written this book without the help and support of many people. First, I need to thank my wife, Julie, for putting up with me for the last six months being busier than usual and for picking up the slack as always—and for always supporting the crazy things I want to do. My children, Kevin, Abby, and Ben, always make all the work worthwhile and can turn the worst, most tiring day into a good one with a smile and a laugh. Thanks to my parents for raising me to have the mindset and work ethic that enables me to accomplish the many things I do while maintaining some sense of humor.

Of course the book wouldn't be possible at all without the Wiley team: Mariann Barsolo, the acquisitions editor; the developmental editor, Kim Beaudet; the production editor, Rebecca Anderson; the copyeditors, Judy Flynn and Kim Wimpsett; and the proofreader, Rebecca Rider. Thanks also to my technical editor and friend, Sean Deuby.

Many people have helped me over the years with encouragement and technical knowledge, and this book is the sum of that. The following people helped with specific aspects of this book, and I want to thank them and give them the credit they deserve for helping make this book as good as possible (and if I've missed anyone, I'm truly sorry): Aashish Ramdas, Ben Armstrong, Charley Wen, Corey Sanders, Don Stanwyck, Elden Christensen, Gabriel Silva, Gavriella Schuster, Jake Oshins, Jeff Woolsey, John Howard, Jose Barreto, Kevin Holman, Kevin Saye, Matt McSpirit, Michael Gray, Michael Leworthy, Mike Schutz, Patrick Lang, Paul Kimbel, Scott Willwerth, Stephen Stair, Steve Linehan, Steven Ekren, and Vijay Tandra Sistla.

About the Author

John Savill is a technical specialist who focuses on Microsoft core infrastructure technologies including Windows, Hyper-V, System Center, and anything that does something cool. He has been working with Microsoft technologies for 20 years and is the creator of the highly popular NTFAQ.com website and a senior contributing editor for *Windows IT Pro* magazine. He has written five previous books covering Windows and advanced Active Directory architecture. When he is not writing books, he regularly writes magazine articles and white papers; creates a large number of technology videos, which are available on his YouTube channel, http://www.youtube.com/ntfaqguy; and regularly presents online and at industry leading events, including TechEd and Windows Connections. When he was writing this book, he had just completed running his annual online John Savill Master Class, which was even bigger and more successful than last year, and he is busy creating a John Savill Hyper-V Master Class, which will include two days of in-depth Hyper-V goodness.

Outside of technology, John enjoys teaching and training in martial arts (including Krav Maga and Jiu-Jitsu), spending time with his family, and participating in any kind of event that involves running in mud, crawling under electrified barbed wire, running from zombies, and generally pushing limits. While writing this book, John was training for the January 2014 Walt Disney World Dopey Challenge, which consists of running a 5K on Thursday, a 10K on Friday, a half marathon on Saturday, and then a full marathon on Sunday. The logic behind the name is that you would have to be dopey to do it, but after completing the Goofy Challenge in 2013—which consisted of the half-marathon and marathon portions—it seemed silly not take it a step further with the new Dopey event that was unveiled for 2014. As John's friend and technical editor Sean says, he does it for the bling ☺.

John tries to update his blog at www.savilltech.com/blog with the latest news of what he is working on.

Introduction

The book you are holding is the result of 20 years of experience in the IT world and over 15 years of virtualization experience that started with VMware and includes Virtual PC and now Hyper-V. My goal for this book is simple: to help you become knowledgeable and effective when it comes to architecting and managing a Hyper-V–based virtual environment. This means understanding how Hyper-V works and its capabilities, but it also means knowing when to leverage other technologies to provide the most complete and optimal solution. That means leveraging System Center and Windows Azure, which I also cover because they relate to Hyper-V. I also dive into some key technologies of Windows Server where they bring benefit to Hyper-V.

Hyper-V is now a mature and widely adopted virtualization solution. It is one of only two x86 server virtualization solutions in Gartner's leader quadrant, and in addition to being used by many of the largest companies in the world, it powers Windows Azure, which is one of the largest cloud services in the world.

Hyper-V is a role of Windows Server, and if you are a Windows administrator, you will find Hyper-V management fairly intuitive, but there are still many key areas that require attention. I have structured this book to cover the key principles of virtualization and the resources you will manage with Hyper-V before I actually cover installing and configuring Hyper-V itself and then move on to advanced topics such as high availability, replication, private cloud, and more.

I am a strong believer in learning by doing, and I therefore highly encourage you to try out all the technologies and principles I cover in this book. You don't need a huge lab environment, and for most of the topics, you could use a single machine with Windows Server installed on it and 8 GB of memory to enable a few virtual machines to run concurrently. Ideally though, having at least two servers will help with the replication and high availability concepts. Sometimes in this book you'll see step-by-step instructions to guide you through a process, sometimes I will link to an external source that already has a good step-by-step guide, and sometimes I will link to videos I have posted to ensure maximum understanding.

I have created an application that is available in the Windows Store, Mastering Hyper-V. It provides easy access to the external links, videos, and code samples I use in this book. As you read each chapter check out the application to find related content. The application can be downloaded from `http://www.savilltech.com/mhv`. Using the Windows Store allows me to also update it over time as required. Please get this application as I will use it to add additional videos based on reader feedback that are not referenced in the main text and include additional information where required.

Who Should Read This Book

I am making certain assumptions regarding the reader:

- You have basic Windows Server knowledge and can install Windows Server.
- You have basic knowledge of what PowerShell is.
- You have access to a Hyper-V server to enable test implementation of the many covered technologies.

This book is intended for anyone who wants to learn Hyper-V. If you have a basic knowledge of virtualization or a competing technology such as VMware, that will help but is not a requirement. I start off with a foundational understanding of each technology and then build on that to cover more advanced topics and configurations. If you are an architect, a consultant, an administrator, or really anyone who just wants better knowledge of Hyper-V, this book is for you.

There are many times I go into advanced topics that may seem over your head. In those cases, don't worry. Focus on the preceding elements you understand, and implement and test them to solidify your understanding. Then when you feel comfortable, come back to the more advanced topics. They will seem far simpler once your understanding of the foundational principles are solidified.

What's Inside

Here is a glance at what's in each chapter.

> **Chapter 1, “Introduction to Virtualization and Microsoft Solutions,**” focuses on the core value proposition of virtualization and how the datacenter has evolved. It covers the key changes and capabilities of Hyper-V in addition to the role System Center plays in a Hyper-V environment. I will cover the types of cloud services available and how Hyper-V forms the foundation of private cloud solutions.

> **Chapter 2, “Virtual Machine Resource Fundamentals,**” covers the core resources of a virtual machine, specifically architecture (generation 1 and generation 2 virtual machines), processor, and memory. You will learn about advanced configurations to enable

many types of operating system support along with best practices for resource planning.

Chapter 3, "Virtual Networking," covers one of the most complicated aspects of virtualization, especially when using the new network virtualization capabilities in Hyper-V. This chapter covers the key networking concepts, how to architect virtual networks, and how to configure them. I'll also cover networking using System Center Virtual Machine Manager (SCVMM) and how to design and implement network virtualization.

Chapter 4, "Storage Configurations," covers the storage options for Hyper-V environments, including the VHD and VHDX formats plus capabilities in Windows Server 2012 R2 that help manage direct attached storage. You will learn about storage technologies for virtual machines such as iSCSI, Virtual Fibre Channel, and shared VHDX; their relative advantages; and also the storage migration and resize functions.

Chapter 5, "Managing Hyper-V," walks through the installation of and best practices for managing Hyper-V. The basics of configuring virtual machines, installing operating systems, and using the Hyper-V Integration Services are all covered. Strategies for migrating from other hypervisors, physical servers, and other versions of Hyper-V are explored.

Chapter 6, "Maintaining a Hyper-V Environment," focuses on the tasks required to keep Hyper-V healthy after you've installed it, which includes patching, malware protection, backup, and monitoring. Key actions such as taking checkpoints of virtual machines, setting up service templates, and performance tuning are covered.

Chapter 7, “Failover Clustering and Migration Technologies,” covers making Hyper-V highly available using failover clustering and will include a deep dive into exactly what makes a cluster tick, specifically when running Hyper-V. Key migration technologies such as Live Migration, Shared Nothing Live Migration, and Storage Migration are explored in addition to configurations related to mobility outside of a cluster and placement optimization for virtual machines.

Chapter 8, “Hyper-V Replica and Cloud Orchestration,” shifts from high availability to a requirement of many organizations today, providing disaster recovery protection in the event of losing an entire site. This chapter looks at the options for disaster recovery, including leveraging Hyper-V Replica and orchestrating failovers with Windows Azure in the event of a disaster.

Chapter 9, “Implementing the Private Cloud and SCVMM,” shows the many benefits of the Microsoft stack to organizations beyond just virtualization. This chapter explores the key benefits and what a private cloud using Microsoft technologies actually looks like. Key components and functional areas, including the actual end user experience and how you can leverage all of System Center for different levels of private cloud capability, are all covered.

Chapter 10, “Remote Desktop Services,” shifts the focus to another type of virtualization, virtualizing the end user experience, which is a critical capability for most organizations. Virtual desktop infrastructure is becoming a bigger component of the user environment. This chapter looks at the different types of desktop virtualization available with Remote Desktop Services with a focus on capabilities that are enabled by Hyper-V, such as advanced graphical capabilities with RemoteFX.

Chapter 11, “Windows Azure IaaS and Storage,” explores the capabilities of one of the biggest public cloud services in the world, which is powered by Hyper-V. This chapter will cover the fundamentals of Windows Azure and how to create virtual machines in Windows Azure. The chapter will also cover the networking options available both within Windows Azure and to connect to your on-premises network. I will examine the migration of virtual machines and how to leverage Windows Azure Storage. Ways to provide a seamless management experience will be explored.

Chapter 12, “Bringing It All Together with a Best-of-Breed Cloud Solution,” brings together all the different technologies and options to help architect a best-of-breed virtualization and cloud solution.

Chapter 13, “The Hyper-V Decoder Ring for the VMware Administrator,” focuses on converting skills for VMware to their Hyper-V equivalent. This chapter also focuses on migration approaches and ways to translate skills.

NOTE

Don't forget to download the companion Windows Store application, Mastering Hyper-V, from `http://www.savilltech.com/mhv`.

The Mastering Series

The Mastering series from Sybex provides outstanding instruction for readers with intermediate and advanced skills, in the form of top-notch training and development for those already working in their field and clear, serious education for those aspiring to become pros. Every Mastering book includes the following elements:

- Skill-based instruction, with chapters organized around real tasks rather than abstract concepts or subjects
- Self-review test questions, so you can be certain you're equipped to do the job right

How to Contact the Author

I welcome feedback from you about this book or about books you'd like to see from me in the future. You can reach me by writing to john@savilltech.com. For more information about my work, visit my website at `www.savilltech.com`.

Sybex strives to keep you supplied with the latest tools and information you need for your work. Please check the Sybex website at `www.sybex.com/go/masteringhyperv2012r2`, where we'll post additional content and updates that supplement this book should the need arise.

Chapter 1
Introduction to Virtualization and Microsoft Solutions

This chapter lays the foundation for the core fabric concepts and technologies discussed throughout not just this first part but also for the entire book. Virtualization has radically changed the layout and operation of a datacenter, and this datacenter evolution and its benefits are explored.

Microsoft's solution for virtualization is its Hyper-V technology, which is a core part of Windows Server and is also available in the form of a free stand-alone hypervisor. The virtualization layer is only part of the solution. Management is just as critical, and in today's world, the public cloud is also a consideration, and so a seamless management story with compatibility between your on- and off-premises resources provides the model implementation.

In this chapter, you will learn to

- Articulate the key value propositions of virtualization
- Understand the differences in functionality between the different versions of Hyper-V
- Differentiate between the types of cloud services and when each type is best utilized

The Evolution of the Datacenter

There are many texts available that go into large amounts of detail about the history of datacenters, but that is not the goal of the following sections. Instead, I am going to take you through the key changes I have seen in my 20 years of working in and consulting about datacenter infrastructure. This brief look at the evolution of datacenters will help you understand the challenges of the past, why virtualization has become such a key component of every modern datacenter, and why there is still room for improvement.

One Box, One Operating System

Datacenters as recent as 10 years ago were all architected in a similar way. These huge rooms with very expensive cabling and air conditioning were home to hundreds if not thousands of servers. Some of these servers were mainframes, but the majority were regular servers (although today the difference between a mainframe and a powerful regular server is blurring), and while the processor architecture running in these servers may have been different—for example, some were x86 based, some Alpha, some MIPS, some SPARC—each server ran an operating system (OS) such as Windows, Linux, or OpenVMS. Some OSs supported different processor architectures while others were limited to a specific architecture, and likewise some processor architectures would dictate which OS had to be used. The servers themselves may have been freestanding, and as technology has advanced, servers got smaller and became rack mountable, enabling greater compression of the datacenter.

Understanding x86

Often the term *x86* is used when talking about processor architecture, but its use has been generalized beyond just the original Intel processors that built on the 8086. *x86* does not refer to only Intel processors but is used more generally to refer to 32-bit operating systems running on any processor leveraging x86 instruction sets, including processors from AMD. *x64* represents a 64-bit instruction set extension processor (primarily from Intel and AMD), although you may also see amd64 to denote 64-bit. What can be confusing is that a 64-bit processor is still technically x86, and it has become more common today to simply use *x86* to identify anything based on x86 architecture, which could be 32-bit or 64-bit from other types of processor architecture. Therefore, if you see *x86* within this book or in other media, it does not mean 32-bit only.

Even with all this variation in types of server and operating systems, there was something in common. Each server ran a single OS, and that OS interacted directly with the hardware in the server and had to use hardware-specific drivers to utilize the capabilities available. In the rest of this book, I'm going to primarily focus on x86 Windows; however, many of the challenges and solutions apply to other OSs as well.

Every server comprises a number of resources, including processor, memory, network, and storage (although some modern servers do not have local storage such as blade systems and instead rely completely on external storage subsystems). The amount of each resource can vary drastically, as shown in the following sections.

Processor

A server can have one or more processors, and it's common to see servers with two, four, or eight processors (although it is certainly possible to have servers with more). Modern processors use a core architecture that allows a single processor to have multiple cores. Each core consists of a discrete central processing unit (CPU) and L1 cache (very fast memory used for temporary storage of information related to computations) able to perform its own computations, and those multiple cores can then share a common L2 cache (bigger but not as fast as L1) and bus interface. This allows a single physical processor to perform multiple parallel computations and actually act like many separate processors. The first multicore processors had two cores (dual-core) and this continues to increase, with eight-core (octo-core) processors available and a new “many-core” generation on the horizon, which will have tens of processor cores. It is common to see a physical processor referred to as a socket and each processor core referred to as a logical processor. For example, a dual-socket system with quad-core processors would have eight logical processors (four on each physical processor, and there are two processors). In addition to the number of sockets and cores, there are variations in the speed of the processors and the exact instruction sets supported. (It is because of limitations in the continued increase of clock speed that moving to multicore became the best way to improve overall computational performance, especially as modern operating systems are multithreaded and can take advantage of parallel computation.) Some processors also support hyperthreading, which is a means to split certain parts of a processor core into two parallel computational streams to avoid wasted processing. Hyperthreading does not double computational capability but generally gives a 10 to 15 percent performance boost. Typically with hyperthreading, this would therefore double the number of logical processors in a system. However, for virtualization, I prefer to not do this doubling, but this does not mean I turn off hyperthreading. Hyperthreading may sometimes help, but it certainly won't hurt.

Previous versions of Windows actually supported different processor architectures, including MIPS, Alpha, and PowerPC in early versions of Windows and more recently Itanium. However, as of Windows Server 2012, the only supported processor architecture is x86 and specifically only 64-bit from Windows Server 2008 R2 and above (there are still 32-bit versions of the Windows 8/8.1 client operating system).

Prior to Windows Server 2008, there were separate versions of the hardware abstraction layer (HAL) depending on if you had a uniprocessor or multiprocessor system. However, given the negligible performance savings on modern, faster processors that was specific to the uniprocessor HAL on single-processor systems (synchronization code for multiple processors was not present in the uniprocessor HAL), this was removed, enabling a single unified HAL that eases some of the pain caused by moving from uni- to multiprocessor systems.

Memory

The memory resource is generally far simpler and not really a huge variation. Some memory supports error-correcting code (ECC), which provides resiliency against the most common types of internal corruption, and memory has different speeds. However, for most environments, the memory considerations is simply how much there is! Generally, the more memory, the better, and with only 64-bit versions of Windows Server, there are no longer considerations around the maximum amount of memory that can be used by an operating system (a 4 GB limit exists for 32-bit operating systems).

Storage

Storage will fall into one of two buckets. The storage is internal (direct-attached storage, or DAS), which means the disks are local to the server, and attached via a technology such as SCSI, SATA, or SAS (even if the storage is in an external storage enclosure but is connected via one of these means, it is still considered direct-attached). Alternatively, the storage is external, such as storage that is hosted on another server or on a storage area network (SAN) or on network-attached storage (NAS). Various protocols may be used for external storage access that offer either file-level or block-level access to the storage.

File-level access enables the requesting server to access files on the server, but this is offered over a protocol that hides the underlying file system and actual blocks of the file on disk. Examples of file-level protocols are Server Message Block (SMB) and Network File System (NFS), typically offered by NAS devices.

Block-level access enables the requesting server to see the blocks on the disk and effectively mount the disk, format the mounted disk with a file system, and then directly manipulate blocks on the disk. Block-level access is typically offered by SANs using protocols such as iSCSI (which leverages the TCP/IP network) and Fibre Channel (which requires dedicated hardware and cabling). Typically, block-level protocols have offered higher performance, and the SANs providing the block-level storage offer advanced features, which means SANs are typically preferred over NAS devices for enterprise storage. However, there is a big price difference between a SAN and potentially the dedicated storage hardware and cabling (referred to as storage fabric), and an NFS device that leverages the existing IP network connectivity.

The hardware for connectivity to storage can vary greatly for both internal storage such as SCSI controllers and external storage such as the host bus adapters (HBAs), which provide the connectivity from a server to a Fibre Channel switch (which then connects to the SAN). Very specific drivers are required for the exact model of storage adapter, and often the driver version must correlate to a firmware version of the storage adapter.

In all components of an environment, protection from a single point of failure is desirable. For internal storage, it is common to group multiple physical disks together into arrays that can provide protection from data loss due to a single disk failure, a Redundant Array of Independent Disks (RAID), although Windows Server also has other technologies that will be covered in later chapters, including Storage Spaces. For external storage, it is possible to group multiple network adapters together into a team for IP-based storage access. For example, SMB, NFS, and iSCSI can be used to provide resiliency from a single network adapter failure, and for non-IP-based storage connectivity, it is common for a host to have at least two storage adapters, which are in turn each connected to a different storage switch (removing single points of failure). Those storage adapters are effectively joined using Multi-Path I/O (MPIO), which provides protection from a single storage adapter or storage switch failure. Both the network and storage resiliency configurations are very specific and can be complex.

Finally, the actual disks themselves have different characteristics, such as size and also their speed. The higher availability of SSD storage and its increase in size and reduced cost is making it a realistic component of modern datacenter storage solutions. This is especially true in tiered solution, which allow a mix of fast and slower disks with the most used and important data moved to the faster disks. Disk speed is commonly measured in input/output operations per second, or IOPS (pronounced “eye-ops”). The higher the IOPS, the faster the storage.

The storage also contains the actual operating system (which can be local or on a remote SAN using boot-from-SAN capabilities).

Networking

Compute, memory, and storage enable a server to perform work, but in today's environments, that work often relies on work done by other servers. In addition, access to that work from clients and the communication between computers is enabled through the network. To participate in an IP network, each machine has to have at least one IP address, which can be statically assigned or automatically assigned. To enable this IP communication, a server has at least one network adapter, and that network adapter has one or more ports that connect to the network fabric, which is typically Ethernet. As is true when it is connecting to storage controllers, the operating system requires a driver specific to the network adapter to connect to it. In high-availability network configurations, multiple network adapters are teamed together, which can be done in many cases through the driver functionality or in Windows Server 2012 using the native Windows NIC Teaming feature. Typical networking speeds in datacenters are 1 gigabit per second (Gbps) and 10 Gbps, but faster speeds are available. Like IOPS with storage, the higher the network speed, the more data you can transfer and the better the network performs.

How Virtualization Has Changed the Way Companies Work and Its Key Values

I spend quite a lot of time talking about the resources and how they can vary, and where specific drivers and configurations may be required. This is critical to understand because many of the benefits of virtualization derive directly from the complexity and variation in all the resources available to a server. Figure 1.1 shows the Device Manager output from a server. Notice all the very specific types of network and storage hardware.

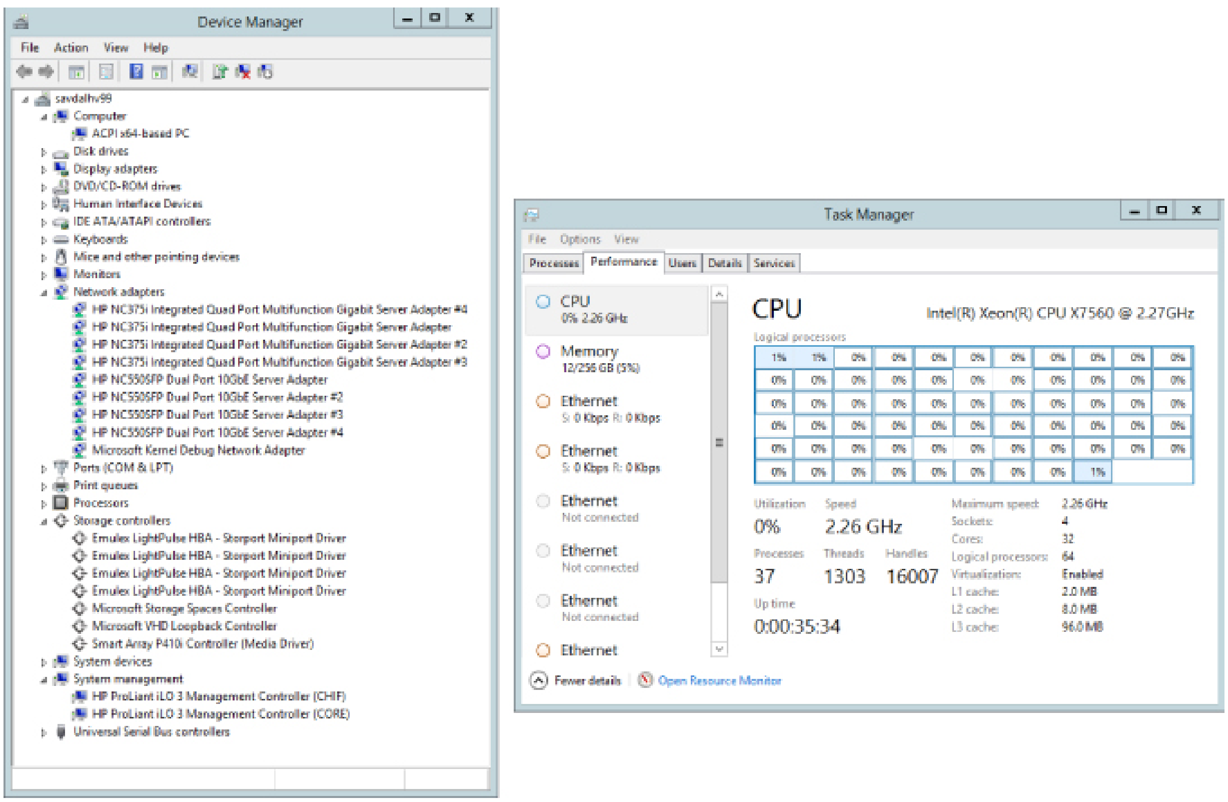

Figure 1.1 The Device Manager view of a typical physical server with Task Manager showing some of its available resources

All these resources are very specific to the deployed operating system and are not easy to change in normal physical server deployments. If the boot disk from a server is placed in a different server with a different motherboard, network, or storage, there is a strong possibility the server will not boot, and it certainly will lose configuration settings and may not be able to use the hardware in the new server. The same applies to trying to restore a backup of a server to different hardware. This tight bonding between the operating system and the hardware can be a major pain point for organizations when they are considering resiliency from hardware failure but also for their disaster recovery planning. It's necessary to have near identical hardware in the disaster recovery location, and organizations start to find themselves locked in to specific hardware vendors.

Virtualization abstracts the physical hardware from that of the created virtual machines. At a very high level, virtualization allows virtual machines to be created. The virtual machines are assigned specific amounts of resources such as CPU and memory in addition to being given access to different networks via virtual switches; they are also assigned storage through virtual hard disks, which are just files on the local file system of the virtualization host or on remote storage. Figure 1.2 shows a high-level view of how a virtualized environment looks.

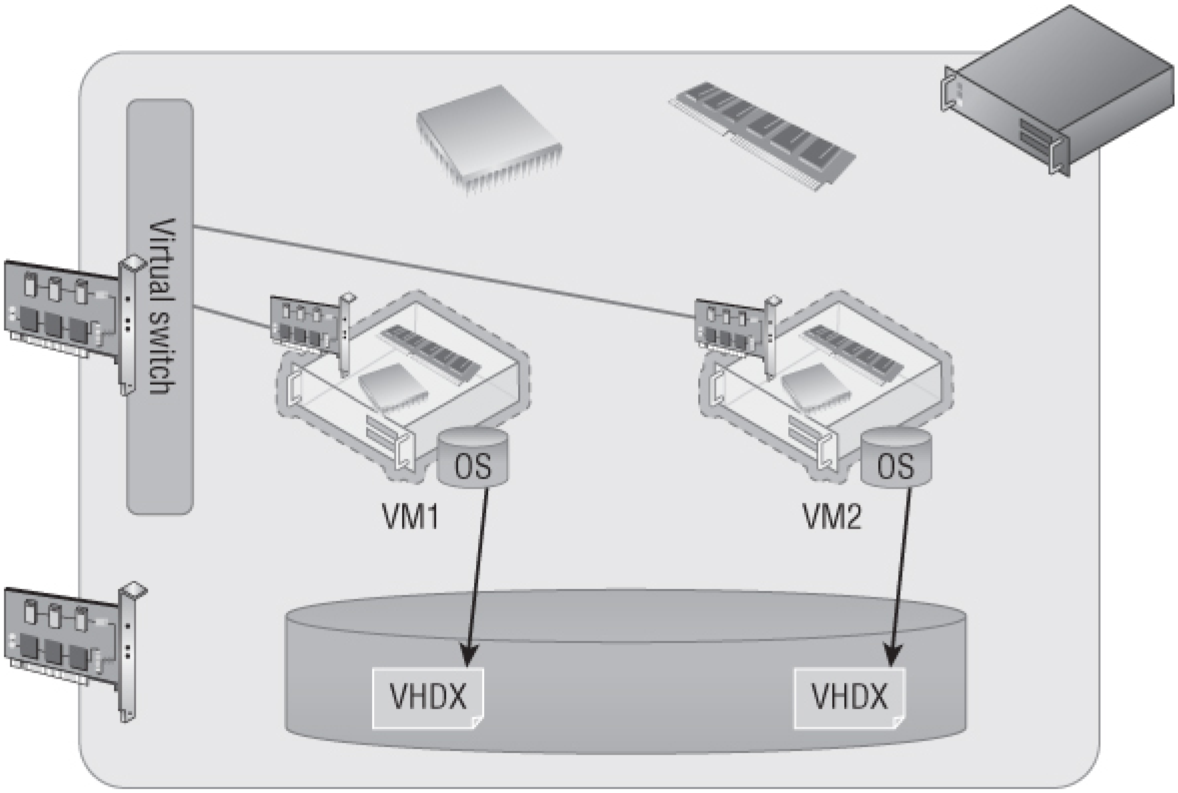

Figure 1.2 A high-level view of a virtualization host and resources assigned to virtual machines

Within the virtual machine, an operating system is installed such as Windows Server 2012 R2, Windows Server 2008, Windows 8, or a Linux distribution. No special process is needed to install the operating system into a virtual machine, and it's not even necessary for the operating system to support virtualization. However, most modern operating systems are virtualization-aware today and are considered “enlightened” to be able to directly understand virtualized hardware. The operating system installed in the virtual machine, commonly referred to as the guest operating system, does not see the physical hardware of the server but rather a set of virtualized sets of hardware that is completely abstracted from the physical hardware. Figure 1.3 shows a virtual machine that is running on the physical server shown in Figure 1.1. Notice the huge difference in what is visible. All the same capabilities are available—the processor capability, memory (I only assigned the VM 212 GB of memory but up to 1 TB can be assigned), storage, and networks—but it is all through abstracted, virtual hardware that is completely independent of the physical server on which the virtual machine is running.

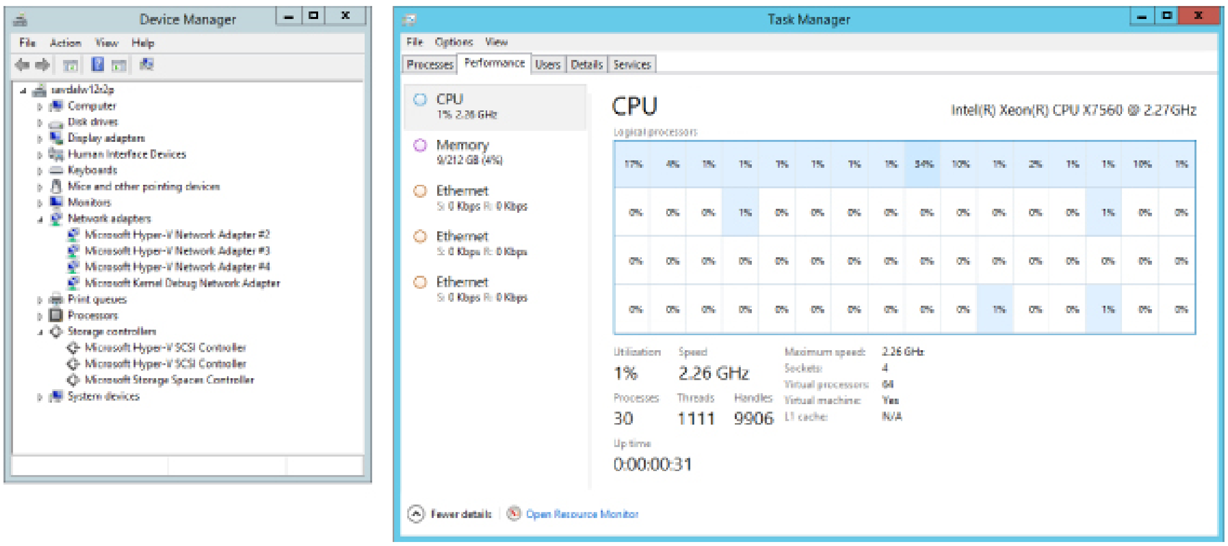

Figure 1.3 A virtual machine running on a physical server

This means that with virtualization, all virtualized operating system environments and their workloads become highly mobile between servers. A virtual machine can be moved between any two servers, provide that those servers are running the same version of the hypervisor and that they have enough resource capacity. This enables organizations to be more flexible with their server hardware, especially in those disaster recovery environments that now allow any hardware to be used in the disaster recovery location as long as it runs the same hypervisor. When a backup needs to be performed, it can be performed at the hypervisor level and then at restoration provided the new server is running the same hypervisor version. As long as this is the case, the virtual machine backup can be restored and used without additional reconfiguration or manual repair.

The next major pain point with physical servers is sizing them—deciding how much memory they need, how many processors, how much storage (although the use of SANs has removed some of the challenge of calculating the amount of local storage required), how many network connections, and what levels of redundancy. I spent many years as a consultant, and when I was specifying hardware, it always had to be based on the busiest possible time for the server. It was also based on its expected load many years from the time of purchase because organizations wanted to ensure that a server would not need to be replaced in six months as its workload increased. This meant servers would be purchased that had far more resources than were actually required, especially the processor resources, where it was typical to see a server running at 5 percent processor utilization with maybe a peak of 15 percent at its busiest times. This was a huge waste of resources and not optimal resource utilization. However, because each OS instance ran on its own box and often server-class hardware only comes in certain configurations, even if it was known that the processor requirement would not be high, it was not possible procure lower-specification hardware. This same overprocurement of hardware applied to the other resources as well, such as memory, storage, and even network resources.

In most environments, different services need processor resources and memory at different times, so being able to somehow combine all the resources and share between operating system instances (and even modify the amounts allocated as needed) is key, and this is exactly what virtualization provides. In a virtual environment, the virtualization host has all of the resources, and these resources are then allocated to virtual machines. However, some resources such as processor and network resources can actually be shared between multiple virtual machines, allowing for a much greater utilization of the available resource and avoiding the utilization waste. A single server that previously ran a single OS instance with a 10 percent processor usage average could run 10 virtualized OS instances in virtual machines with most likely only additional memory being required in the server and higher IOPS storage. The details of resource sharing will be covered in future chapters, but resources such as those for processors and networks can actually be shared between virtual machines concurrently; resources like memory and storage can be segregated between virtual machines but cannot actually be shared because you cannot store different pieces of information in the same physical storage block.

The best analogy is to consider your Windows desktop that is running a single OS and likely has a single processor but is able to seemingly run many applications all at the same time. You may be using Internet Explorer to stream a movie, sending email with Outlook, and editing a document in Word. All of these applications seem to be running at the same time, but a processor core can perform only one computation at a time (ignoring multicores and hyperthreading). In reality, though, what is happening is that the OS is time-slicing turns on the processor and giving each application a few milliseconds of time each cycle, and with each application taking its turn on the processor very quickly, it appears as if all of the applications are actually running at the same time. A similar concept applies to network traffic, except this time there is a finite bandwidth size and the combined network usage has to stay within that limit. Many applications can send/receive data over a shared network connection up to the maximum speed of the network. Imagine a funnel. I could be pouring Coke, Pepsi, and Dr Pepper down the funnel and all would pour at the same time up to the size of the funnel. Those desktop applications are also assigned their own individual amounts of memory and disk storage. This is exactly the same for virtualization except instead of the OS dividing up resource allocation, it's the hypervisor allocating resources to each virtual machine that is running but uses the same mechanisms.

Building on the previous benefit of higher utilization is one of scalability and elasticity. A physical server has a fixed set of resources that are not easily changed, which is why physical deployments are traditionally overprovisioned and architected for the busiest possible time. With a virtual environment, virtual machine resources can be dynamically changed to meet the changing needs of the workload. This dynamic nature can be enabled in a number of ways. For resources such as processor and network, the OS will use only what it needs, which allows the virtual machine to be assigned a large amount of processor and network resources because those resources can be shared. So while one OS is not using the resource, others can. When it comes to resources that are divided up, such as memory and storage, it's possible to add them to and remove them from a running virtual machine as needed. This type of elasticity is not possible in traditional physical deployments, and with virtualization hosts generally architected to have far more resources than in a physical OS deployment, the scalability, or maximum resource that can be assigned to a virtualized OS, is much larger.

The consolidation of operating system instances onto a smaller number of more powerful servers exposes a number of additional virtualization benefits. With a reduced number of servers that are more powerful but more highly utilized, organizations see reduced datacenter space requirements, which leads to energy savings and also ultimately cost savings.