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 of this book, but the entire book. Virtualization has radically changed the layout and operation of the 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 it 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. Thus 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 various versions of Hyper-V.
- Differentiate between the types of cloud services and when each type is best utilized.

The Evolution of the Datacenter

Many books are available that go into a great amount 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 that 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

As recent as 10 years ago, datacenters 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). Although 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. Likewise, some processor architectures would dictate which OS had to be used. The servers themselves may have been freestanding, but as technology 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 only to Intel processors, but it 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 simply to 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 they had 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 available capabilities. In the rest of this book, I focus primarily 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. 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, variations exist 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 it 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 not to do this doubling, but this does not mean that I turn off hyperthreading. Hyperthreading may sometimes help, but it certainly won't hurt.

Is There a Big and a Little Thread with Hyperthreading?

Hyperthreading enables two streams of execution on a single processor core, and you often hear numbers such as a 15 percent performance improvement. This leads to the belief that there is the main thread on the core and then a little "mini-me" thread that has a smaller capability. This is not true. With hyperthreading, a single core has some components duplicated, enabling two sets of logical state per core. Typically, during a thread of execution, the core is not fully utilized for various reasons, such as when a particular instruction stream uses only specific types of ALU (Arithmetic Logic Unit), leaving others unused, and more commonly when a cache miss occurs that causes the thread execution to stall while data is fetched. With hyperthreading and the two sets of logical state, if one thread is stalled because of a cache miss, the chances are good that the other thread can execute. This, therefore, keeps the core better utilized and improves the overall performance, and this is where the 15 percent performance gain comes from. Notice that both threads are equal and which one does more work just depends on how busy they are kept, the type of computations, the frequency of cache misses, and so on.

Earlier versions of Windows supported different processor architectures, including MIPS, Alpha, PowerPC, 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 whether you had a uniprocessor or multiprocessor system. However, given the negligible performance savings on modern, faster processors that were 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, with fewer variations. 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 consideration 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 4GB limit exists for 32-bit operating systems).

Storage

Storage falls into one of two buckets: internal or external. If the storage is internal (direct-attached storage, or DAS), 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 filesystem 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 filesystem, 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 that 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 SMB device that leverages the existing IP network connectivity.

The line between types of storage is also blurring greatly, especially with modern hyperconverged systems that contain both compute and the storage for workloads. Windows Server 2016 includes Storage Spaces Direct (S2D), which enables direct-attached storage in cluster nodes to be aggregated together and utilized as cluster storage. This is commonly referred to as a *VSAN technology* in the industry. When combined with other Windows Server storage features, using direct-attached storage no longer means compromising features and performance.

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). Windows Server also has other technologies that are 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 multipath 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 disks themselves have different characteristics, such as size and 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 solutions, 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.

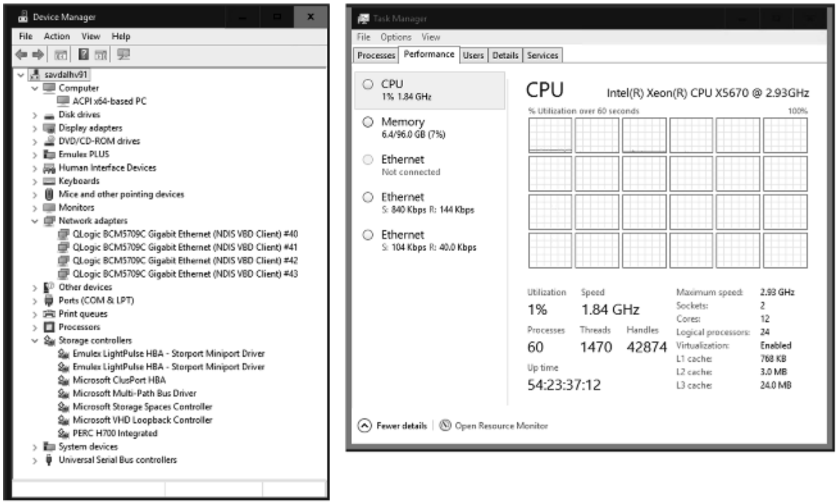
The storage also contains the actual operating system (which can be local or on a remote SAN using boot-from-SAN capabilities).

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 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 connecting to storage controllers, the operating system requires a driver specific to the network adapter to connect to the network. 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 10Gbps, but faster speeds are available. As with IOPS for storage, the higher the network speed, the more data that 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 resources and how they can vary, and where specific drivers and configurations may be required. This is critical to understand because many benefits of virtualization derive directly from the complexity and variation in all of the resources available to a server. Figure 1.1 shows the Device Manager output from a server. Notice all of the very specific types of network and storage hardware.



```
savdalhv91
  Computer
    ACPI x64-based PC
  Disk drives
  Display adapters
  DVD/CD-ROM drives
  Emulex PLUS
  Human Interface Devices
  Keyboards
  Mice and other pointing devices
  Monitors
  Network adapters
    QLogic BCM5709C Gigabit Ethernet (NDIS VBD Client) #40
    QLogic BCM5709C Gigabit Ethernet (NDIS VBD Client) #41
    QLogic BCM5709C Gigabit Ethernet (NDIS VBD Client) #42
    QLogic BCM5709C Gigabit Ethernet (NDIS VBD Client) #43
  Other devices
  Ports (COM & LPT)
  Print queues
  Processors
  Storage controllers
    Emulex LightPulse HBA - Storport Miniport Driver
    Emulex LightPulse HBA - Storport Miniport Driver
    Microsoft ClusPort HBA
    Microsoft Multi-Path Bus Driver
    Microsoft Storage Spaces Controller
    Microsoft VHD Loopback Controller
    PERC H700 Integrated
  System devices
  Universal Serial Bus controllers
```

```
○ CPU                  CPU      Intel(R) Xeon(R) CPU X5670 @ 2.93GHz
  1% 1.84 GHz
○ Memory
  6.4/96.0 GB (7%)
○ Ethernet
  Not connected
○ Ethernet
  S: 840 Kbps R: 144 Kbps
○ Ethernet
  S: 104 Kbps R: 40.0 Kbps

Utilization  Speed                  Maximum speed:       2.93 GHz
1%           1.84 GHz               Sockets:             2
                                    Cores:               12
Processes    Threads    Handles     Logical processors:  24
60           1470       42874       Virtualization:      Enabled
                                    L1 cache:            768 KB
Up time                             L2 cache:            3.0 MB
54:23:37:12                         L3 cache:            24.0 MB
```

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

All of these resources are 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 into specific hardware vendors.

Virtualization abstracts the physical hardware from that of the created virtual machines. At a 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 filesystem of the virtualization host or on remote storage. Figure 1.2 shows a high-level view of a virtualized environment.

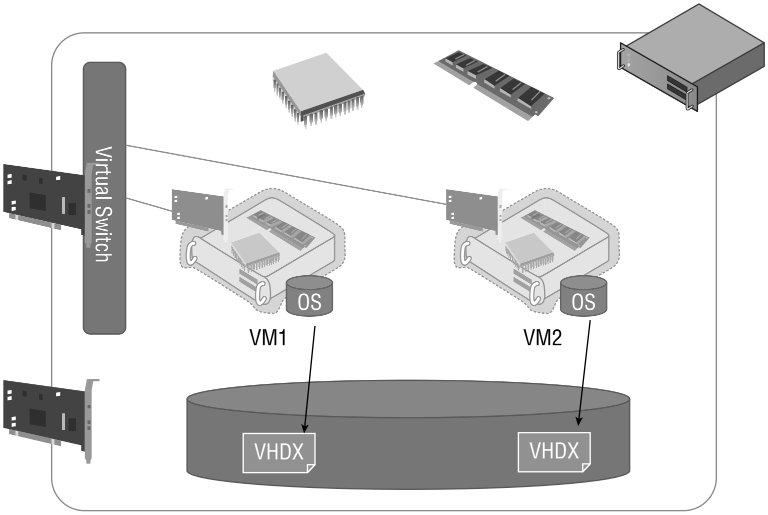
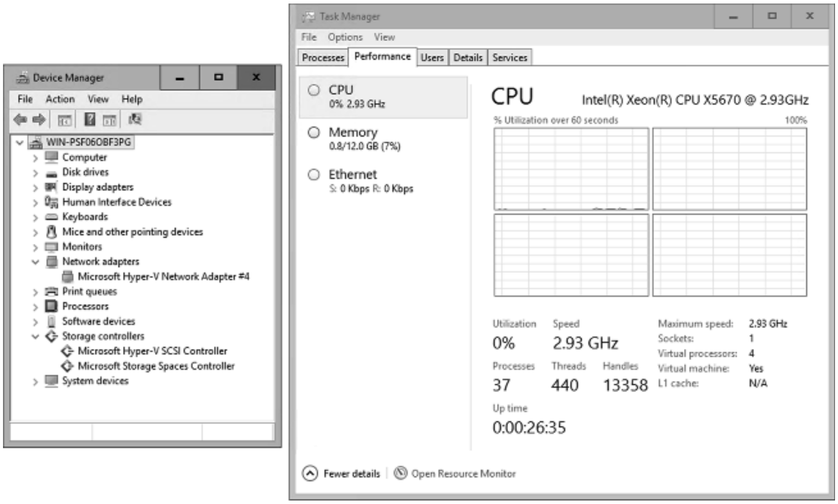


FIGURE 1.2
A high-level view of a virtualization host and resources assigned to virtual machines

Within the virtual machine, an operating system is installed such as Windows Server 2016, Windows Server 2008, Windows 10, 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 understand virtualized hardware directly. 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 hardware that is completely abstracted from the physical hardware.

Figure 1.3 shows a virtual machine (VM) that is running on the physical server shown in Figure 1.1. Notice the huge difference in what is visible. All of the same capabilities are available—the processor capability, memory (I assigned the VM only 12GB of memory, but up to 1TB 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



WIN-PSF06OBF3PG
- Computer
- Disk drives
- Display adapters
- Human Interface Devices
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
  - Microsoft Hyper-V Network Adapter #4
- Print queues
- Processors
- Software devices
- Storage controllers
  - Microsoft Hyper-V SCSI Controller
  - Microsoft Storage Spaces Controller
- System devices

CPU
0% 2.93 GHz

Memory
0.8/12.0 GB (7%)

Ethernet
S: 0 Kbps R: 0 Kbps

CPU Intel(R) Xeon(R) CPU X5670 @ 2.93GHz

% Utilization over 60 seconds 100%

Utilization 0%
Speed 2.93 GHz
Processes 37
Threads 440
Handles 13358
Up time 0:00:26:35

Maximum speed:	2.93 GHz
Sockets:	1
Virtual processors:	4
Virtual machine:	Yes
L1 cache:	N/A

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, provided those servers are running the same version of the hypervisor and 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 required, especially processor resources; 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 comes in only certain configurations, even if it was known that the processor requirement would not be high, it was not possible to 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 somehow to combine all of the resources and share them 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 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 are covered in future chapters, but resources such as those for processors and networks can be shared between virtual machines concurrently; resources like memory and storage can be segregated between virtual machines, but they 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 that is able seemingly to run many applications all at the same time. You may be streaming a movie with Internet Explorer, 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, the OS is time-slicing turns on the processor and giving each application a few milliseconds of time each cycle. With each application taking its turn on the processor very quickly, it appears as if all applications are 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 are 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 various 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 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 ultimately cost savings.

Many organizations have long struggled with a nontechnical aspect of their datacenters, and that is licensing. I cover licensing in detail later in this chapter, but when you have thousands of individual servers, each running a single operating system, it can be hard to track all of the licenses and hard to know exactly what version you need based on the capabilities required. Most important, it costs a lot of money. With virtualization, there are ways to license the virtualization hosts themselves and allow an unlimited number of virtual machines, making licensing of the OS and management software far more cost-effective.

Another challenge with a single operating system per physical server is all the islands of resources that you have to manage. Every server has its own local storage, and somehow you have to protect all of that data. Utilizing centralized storage such as a SAN for every physical server is possible but typically cost prohibitive. It's not practical to purchase Fibre Channel HBAs (cards that enable connectivity to Fibre Channel switches), Fibre Channel switches to accommodate all of the servers, and all of the cabling. Take those same servers and reduce the number of physical servers even tenfold using virtualization, and suddenly connecting everything to centralized storage is far more realistic and cost effective. The same applies to regular networking. Implementing 10Gbps networking in a datacenter for 100 servers is far more possible than it is for one with 1,000 servers.

On the opposite side of the scale from consolidation and centralization is the challenge of isolating workloads. Consider a branch location that for cost purposes has only a single server to host services for the local workers. Because there is only a single server, all roles have to run on a single OS instance without virtualization, which can lead to many complications in configuration and supportability. With virtualization, that same server can host numerous virtual machines, with each workload running in its own virtual machine, such as a virtual machine running a domain controller and DNS, another running file services, and another running a line-of-business (LOB) service. This allows services to be deployed and isolated to standard best practices. Additionally, many remote offices will deploy two virtualization servers with some kind of external storage enclosure that can be connected to both servers, or with Windows Server 2016, another option would be to deploy four servers with internal storage and leverage Storage Spaces Direct for clustered storage. This enables virtual machines to be moved between the servers, allowing high availability, which brings us to the next benefit of virtualization.

Physically deployed services that require high availability must have some native high-availability technology. With virtualization, it's still preferred to leverage the service's native high-availability capabilities, but virtualization adds options and can provide solutions where no native capability exists in the virtualized service. Virtualization can enable virtual machines to move between physical hosts with no downtime using Live Migration, and it can even provide disaster-recovery capabilities using technologies such as Hyper-V Replica. Virtualization also allows simpler backup and recovery processes by allowing backups to be taken of the entire virtual machine.

Consider the process of deploying a new service on a physical server. That server configuration has to be specified, ordered, delivered, and installed in the datacenter. Then the OS has to be installed and the actual service configured. That entire process may take a long time, which lengthens the time it takes to provision new services. Those delays may affect an organization's ability to respond to changes in the market and react to customer requirements. In a virtual environment, the provisioning of a new service consists of the creation of a new virtual machine for that service; with the right automation processes in place, that could take minutes from start to finish, instead of weeks. Because resources are pooled together in a virtual infrastructure, it is common always to run with sufficient spare capacity available to allow for new services to be provisioned as needed, and as the amount of free resources drops below a certain threshold, new hardware is purchased and added to the virtual infrastructure ready for additional services. Additionally, because the deployment of a new virtual machine does not require any physical infrastructure changes, the whole process can be completely automated, which helps in the speed of provisioning. By removing many manual steps, the chances of human error are removed, and with a high level of consistency between deployed environments comes a simplified supportability process.

Finally, I want to touch on using public cloud services such as Microsoft Azure Infrastructure as a Service (IaaS), which allows virtual machines to be hosted on servers accessed over the Internet. When using virtualization on premises in your datacenter, and in this case specifically Hyper-V, you have full compatibility between on and off premises, making it easy to move services.

There are other benefits that are specific to virtualization, such as simplified networking infrastructure using network virtualization, greater quality-of-service (QoS) controls, metering, and more. However, the benefits previously mentioned are generally considered the biggest wins of virtualization. To summarize, here are the key benefits of virtualization:

- Abstraction from the underlying hardware, allowing full mobility of virtual machines
- High utilization of resources
- Scalability and elasticity
- Energy, datacenter space, and cost reduction
- Simplification and cost reduction for licensing
- Consolidation and centralization of storage and other resources
- Isolation of services
- Additional high-availability options and simpler backup/recovery
- Speed of service provisioning and automation
- Compatibility with public cloud

Ultimately, what these benefits mean to the organization is either saving money or enabling money to be made faster.

History of Hyper-V

So far in this chapter, I have not used the word *Hyper-V* very much. I have focused on the challenges of traditional datacenters and the benefits of virtualization. I now want to start looking at the changes to the various versions of Hyper-V at a high level since its introduction. This is important because it will not only enable you to understand the features available in your Hyper-V deployments if you are not yet running Windows Server 2016 Hyper-V, but also show the great advancements made with each new version. All of the features I talk about are covered in further detail throughout this book, so don't worry if the following discussion isn't detailed enough. I provide a high-level explanation of what they are in this part of the chapter.

I'll start with the first version of Hyper-V, which was introduced as an add-on after the Windows Server 2008 release. Hyper-V was not an update to Microsoft Virtual Server, which was a virtualization solution Microsoft acquired as part of the Connectix acquisition. Microsoft Virtual Server was not well adopted in many organizations as a virtualization solution because it was a type 2 hypervisor, whereas Hyper-V is a type 1 hypervisor. There are numerous definitions, but I think of them quite simply as follows:

Type 2 Hypervisors A type 2 hypervisor runs on a host operating system. The host operating system manages the underlying hardware; the type 2 hypervisor makes requests to the host operating system for resources and to perform actions. Because a type 2 hypervisor runs on top of a host OS, access to all of the processor rings of operating systems running in the virtual machine is limited, which generally means slower performance and less capability.

Type 1 Hypervisors A type 1 hypervisor runs directly on the bare metal of the server and directly controls and allocates resources to virtual machines. Many type 1 hypervisors take advantage of a Ring –1, which is present on processors that support hardware virtualization to run the hypervisor itself. This then allows virtual machines still to be able to access Ring 0 (kernel mode) of the processor directly for their computations, giving the best performance while still allowing the hypervisor management of the resource. All modern datacenter hypervisors are type 1 hypervisors.

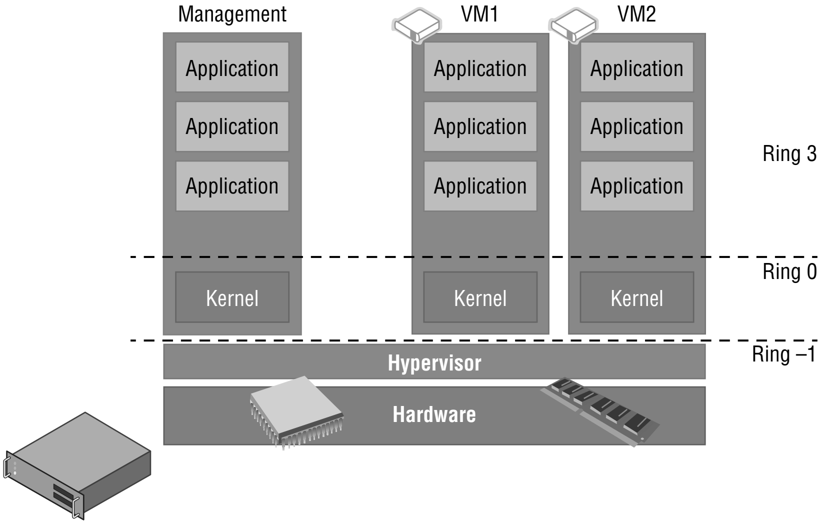
It is important at this stage to realize that Hyper-V is absolutely a type 1 hypervisor. Often people think that Hyper-V is a type 2 hypervisor because of the sequence of actions for installation:

1. Install Windows Server on the physical host.
2. Enable the Hyper-V role.
3. Configure and manage virtual machines through the Windows Server instance installed on the physical host.

Someone might look at this sequence of actions and how Hyper-V is managed and come to the conclusion that the Hyper-V hypervisor is running on top of Windows Server; that is not the case at all. When the Hyper-V role is enabled on Windows Server, changes are made to the boot configuration database to configure the hypervisor to load first, and then the Windows Server operating systems runs on *top* of that hypervisor, effectively becoming a pseudo virtual machine itself. Run the command `bcdedit /enum` on a Hyper-V host, and it shows that the hypervisor launch type is set to automatically launch.

The Windows Server operating system becomes the management partition for the Hyper-V solution. The hypervisor itself is quite compact and needs to be as light as possible, so it's focused on interacting with compute and memory resources and controlling access for virtual machines to avoid introducing latencies in performance. The management partition works for the hypervisor, and it is tasked with various items, such as hosting worker processes to communicate with virtual machines, hosting drivers for storage and network adapter interactions, and more. However, all of the virtual machines are running directly on the hypervisor and not on the host operating system that was installed. This is best shown by looking at the Hyper-V architecture in Figure 1.4, which clearly shows the hypervisor running in Ring –1 and both the management partition and all the virtual machines running side by side on the hypervisor. The management partition does have some additional privileges, capabilities, and hardware access beyond that of a regular virtual machine, but it is still running on the hypervisor.

Figure 1.4
Hyper-V architecture



What Is a Partition?

In the discussion of the history of Hyper-V, I referred to a management partition. The hypervisor runs directly on the hardware and assigns different amounts of resources to each virtual environment. These virtual environments can also be referred to as *partitions*, because they are partitions of the underlying resource. Because the management partition is not a true virtual machine (because not all of its resources are virtualized) and it has privileged access, it is referred to as the *management partition* or the *parent partition*. Although it can be confusing, it's also common to see the management partition referred to as the *host* because it is the OS closest to the hardware and is directly installed on the server. Sometimes virtual machines are referred to as *child partitions* or *guest partitions*.

Windows Server 2008 Hyper-V Features

The initial version of Hyper-V provided a solid foundation for virtualization and a fairly limited set of additional capabilities. As with all versions of Hyper-V, the processors must support hardware-assisted virtualization (AMD-V or Intel VT) and also Data Execution Prevention (DEP). Although Hyper-V is available only on 64-bit versions of Windows Server, it is possible to run both 32-bit and 64-bit operating systems. The initial version of Hyper-V included the following key capabilities:

- Up to 64GB of memory per VM
- Symmetric multiprocessing (SMP) VMs (up to four virtual CPUs [vCPUs] each). However, the exact number differed depending on the guest operating system. For example, four vCPUs were supported on Windows Server 2008 SP2 guests, but only two were on Windows Server 2003 SP2. The full list is available at:

  `http://technet.microsoft.com/en-us/library/cc794868(v=ws.10).aspx`

- Virtual Hard Disk (VHD) format for virtualized storage up to 2TB in size with multiple VHDs supported for each VM on either a virtual IDE controller or a virtual SCSI controller. VMs had to be booted from a VHD attached to a virtual IDE controller, but data VHDs could be connected to a virtual SCSI controller with higher performance through the virtual SCSI controller. Only 4 devices could be connected to the IDE controller (2 to each of the 2 IDE controllers), while each of the 4 virtual SCSI controllers supported up to 64 devices, each allowing up to 256 VHDs attached via the virtual SCSI.
- Leveraged failover clustering for high availability
- Ability to move virtual machines between hosts in a cluster with minimal downtime using quick migration. Quick migration worked by pausing the virtual machine and saving the device, processor, and memory content to a file on the cluster storage. It then moved that storage to another host in the cluster, reading the device, processor, and memory content into a newly staged virtual machine on the target and starting it. Depending on the amount of memory in the virtual machine, this may have meant minutes of downtime and the definite disconnect of any TCP connections. This was one of the biggest weaknesses of the Windows Server 2008 Hyper-V solution.
- Supported VSS (Volume Shadow copy Service) live backup of virtual machines. This allowed a backup to be taken of a virtual machine from the host operating system. The VSS request for the backup was then communicated to the virtual machine's guest operating system through the Hyper-V Integration Services to ensure that the application data in the VM was in an application-consistent state and suitable for a backup.
- The ability to create VM snapshots, which are point-in-time captures of a virtual machine's complete state (including memory and disk). This allowed a VM to be rolled back to any of these snapshots. The use of the term *snapshots* was confusing, because the term is also used in the backup VSS nomenclature, but in this case it's referring to snapshots used in the backup process, which are different from VM snapshots. In Windows Server 2012 R2, VM snapshots are now called *checkpoints* to help remove this confusion.
- Pass-through disk access for VMs was possible even though not generally recommended. It was sometimes required if VMs needed access to single volumes greater than 2TB in size (which was the VHD limit).
- Integration services available for supported guest operating systems, allowing capabilities such as heartbeat, mouse/keyboard interaction, backup services, time synchronization, and shutdown
- Multiple virtual networks could be created with support for 10Gbps and VLANs.

Windows Server 2008 R2 Changes

While Windows Server 2008 Hyper-V offered a solid foundation and a reliable solution for a v1, several limitations stopped Hyper-V from being seriously considered in many environments, among them the ability to move virtual machines between hosts in a cluster with no downtime. There were two challenges for Hyper-V to enable this:

- The VM had to be paused to enable the memory, processor, and device state to be saved to disk.

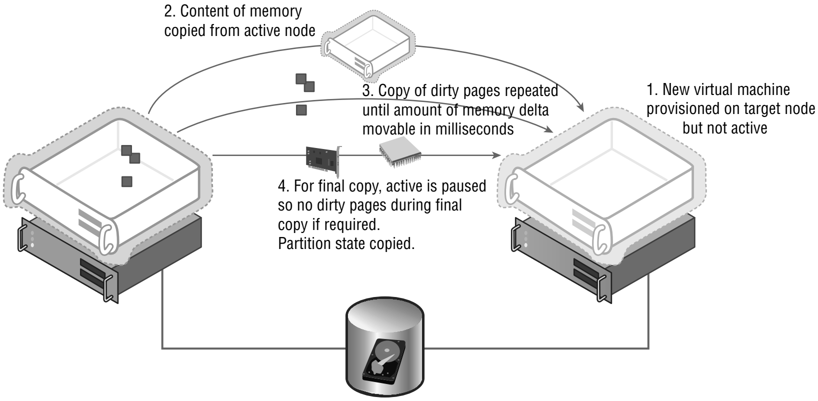
- NTFS is not a shared filesystem and can be mounted by only one OS at a time, which means that when a virtual machine moves between hosts in a cluster, the logical unit number, or LUN (which is a block of storage from a SAN), must be dismounted from the source host and mounted on the target host. This takes time.

Windows Server 2008 R2 solved both of these challenges. First, a new technology called Live Migration was introduced. Live Migration enabled the memory of a virtual machine and the virtual machine's state to be replicated to another host while the virtual machine was still running and then switched over to the new host with no downtime. I cover this in detail in Chapter 7, "Failover Clustering and Migration Technologies," but the technology worked at a high level using the following steps:

1. A container VM was created on the target host using the existing VM's configuration.
2. The memory of the VM was copied from the source to the target VM.
3. Because the VM was still running while the memory was copied, some of the memory content changed. Those dirty pages were copied over again. This process repeated numerous iterations, with the number of dirty pages shrinking by a magnitude each iteration, so the time to copy the dirty pages shrank greatly.
4. Once the number of dirty pages was very small, the VM was paused and the remaining memory pages were copied over along with the processor and device state.
5. The VM was resumed on the target Hyper-V host.
6. A reverse unsolicited ARP was sent over the network, notifying routing devices that the VM's IP address was moved.

The whole process can be seen in Figure 1.5. You may be concerned about Step 4, the VM being paused for a copy of the final few pages of dirty memory. This is common across all hypervisors and is necessary; however, only milliseconds of time are involved, so it's too small to notice and well below the TCP connection time-out, which means no connections to the server would be lost.

FIGURE 1.5
A high-level view of the Live Migration process


2. Content of memory copied from active node
3. Copy of dirty pages repeated until amount of memory delta movable in milliseconds
1. New virtual machine provisioned on target node but not active
4. For final copy, active is paused so no dirty pages during final copy if required.
Partition state copied.

Live Migration solved the problem of pausing the virtual machine to copy its memory between hosts. It did not, however, solve the problem that NTFS couldn't be shared, so the LUN containing the VM had to be dismounted and mounted, which took time. A second new technology solved this problem: *Cluster Shared Volumes*, or CSV.

CSV allows an NTFS-formatted LUN to be available simultaneously to all hosts in the cluster. Every host can read and write to the CSV volume, which removes the need to dismount and mount the LUN as VMs move between hosts. This also solved the problem of having to have one LUN for every VM to enable each VM to be moved independently of other VMs. (The LUN had to move when the VM moved, which meant that if other VMs were stored on the same LUN, those VMs would also have to move.) With CSV, many VMs could be stored on a single CSV volume, with VMs running throughout all hosts in the cluster. Behind the scenes, CSV still leverages NTFS, but it controls the writing of Metadata to the volume to a single host for each CSV volume to avoid any risk of NTFS corruption. This is also explained in detail in Chapter 7.

With Live Migration and CSV technologies working in unison, the ability to move a virtual machine between hosts in a cluster with no downtime was now possible and removed a major obstacle to the adoption of Hyper-V. Windows Server 2008 R2 included other enhancements:

- A processor compatibility mode that allowed a virtual machine to be migrated between different versions of the same processor family. When a guest OS started within a virtual machine, it would commonly query the processor to find out all of the instruction sets available, as would some applications, and those instruction sets would possibly be used. If a virtual machine was then moved to another host with a different processor version that did not support that instruction set, the application/OS would crash when it tried to use it. Download Coreinfo from:

  `http://technet.microsoft.com/en-us/sysinternals/cc835722.aspx`

  and execute it with the `-f` switch. This will show which instruction sets are supported on your processor. When the processor compatibility feature was enabled for a virtual machine, the high-level instruction sets were masked from the VM so it did not use them, allowing the VM to be moved between different versions of the processor.

- Hot-add of storage to the SCSI bus. This enabled additional VHDs to be added to a virtual machine without shutting it down.

- Network performance improvements, including support for jumbo frames, Virtual Machine Queues (VMQs), and allowing the use of NIC Teaming implemented by network drivers

- If the processor supported it, Second Level Address Translation (SLAT), which allowed the processor to own the mapping of virtual memory to physical memory, therefore reducing overhead on the hypervisor. SLAT is used by Hyper-V when available.

Windows Server 2008 R2 Service Pack 1

It's not common for a service pack to bring new features, but Windows Server 2008 R2 had one key feature missing, and this was the ability to change dynamically the amount of memory available to a virtual machine. SP1 for Windows Server 2008 R2 added the Dynamic Memory feature, which was different from how other hypervisors handled memory optimization.

Dynamic Memory worked by configuring a starting amount of memory and a maximum amount of memory. Hyper-V would then monitor the actual amount of memory being used within the virtual machine by processes via the integration services. If the amount of available memory dropped below a certain buffer threshold, additional memory was added to the virtual machine if it was physically available. If a virtual machine no longer needed all of its memory, some was reclaimed for use with other virtual machines. This enabled Hyper-V to achieve great optimization of VM memory and maximize the number of virtual machines that could run on a host.

The other new technology in Service Pack 1 was RemoteFX, based on technologies obtained through the Calista Technologies acquisition. The RemoteFX technology was focused on Virtual Desktop Infrastructure (VDI) deployments running on Hyper-V and making the VDI experience as rich as possible no matter the capabilities of the client device. RemoteFX consisted of three technologies to offer this rich capability:

- The first was the ability to virtualize a GPU (Graphical Processing Unit) in the Hyper-V server and then assign virtual GPUs to virtual machines. This works in a similar way to how CPUs are carved up between virtual machines. Once a virtual machine was assigned a vGPU, the OS within that VM could perform native DirectX processing using the GPU, allowing graphically rich applications to run, such as videoconferencing, Silverlight and Flash applications, and any DirectX application. As a demonstration, I installed Halo 2 in a RemoteFX-enabled virtual machine and played it over the network; you can see this at `http://youtu.be/CYiLGxfZRTA`. Without RemoteFX, some types of media playback would depend on the capability of the client machine, and certainly any application that required DirectX would not run. The key item is that all the graphical rendering is on the Hyper-V host's GPU and not on the local client.
- The second technology was related to the rich graphical capability and was an updated codec that was used to compress and uncompress the screen updates over the network.
- The final technology enabled USB device redirection at a port level. Typically, with Remote Desktop Protocol (RDP), certain types of devices could be used in remote sessions, such as a keyboard, a mouse, a printer, and some devices with an inbox such as a scanner. However, many other types of devices and multifunction devices would not work. RemoteFX USB redirection enabled any USB device to be used in a remote session by redirecting at a USB port level all USB request blocks (URBs).

Note that the last two components of RemoteFX, the codec and USB redirection, are not Hyper-V features but rather updates to RDP. I cover them because they are part of the RemoteFX feature family and complete the remote client experience.

The combination of Dynamic Memory and RemoteFX made Hyper-V a powerful platform for VDI solutions, and Dynamic Memory on its own was useful for most server virtual machines as well.

Windows Server 2012 Hyper-V Changes

Windows Server 2012 put Hyper-V to the top of the list of the true top hypervisors by closing nearly every gap it had with other hypervisors and leapfrogging the competition in many areas. This entire book focuses on many of the changes in Windows Server 2012, but here I call out some of the biggest improvements and new features.

One of the key reasons for the huge advancement of Hyper-V in Windows Server 2012 was not only the big focus on virtualization (to enable Hyper-V to compete and win against the competition) but also the success of Microsoft's public cloud service, Azure. I briefly cover the types of cloud services later in this chapter and in far more detail later in the book, but for now, realize that Azure is one of the largest public cloud services that exists. It powers many of Microsoft's cloud offerings and runs on Windows Server 2012 Hyper-V. All of the knowledge Microsoft gained operating Azure and the enhancements it needed went into Windows Server 2012, and the engineering teams are now cloud-first focused, creating and enhancing technologies that are then made available as part of new Windows Server versions. This is one of the reasons the release cadence of Windows Server has changed to an annual release cycle. Combining the development for the public and private cloud solutions makes Hyper-V a much stronger solution, which is good news for organizations using Hyper-V.

Scalability

The first grouping of changes relates to scalability, which previously was one of the weakest areas. Windows Server 2008 R2 did not change the scalability of virtual machines from Windows Server 2008 (although there were some modest improvements to the Hyper-V host limits). Windows Server 2012 made some big changes, as shown in Table 1.1.

Table 1.1: Scalability Changes from Windows Server 2008 R2 to Windows Server 2012

Attribute	Windows 2008 R2	Windows 2012	Improvement
Logical processors on hardware	64	320 (640 without Hyper-V role)	> 5x
LP:VP ratio	8:1 (12:1 for Windows 7 VDI)	No limit	
Physical memory	1TB	4TB	4x
Virtual processors per host	512	2,048	4x
Virtual processors per virtual machine	4	64 (includes NUMA awareness)	16x
Memory per virtual machine	64GB	1TB	16x
Active virtual machines per host	384	1,024	2.5x
Maximum cluster nodes	16	64	4x
Maximum cluster virtual machines	1,000	8,000	8x
Maximum VHD size	2TB	64TB (with VHDX)	32x

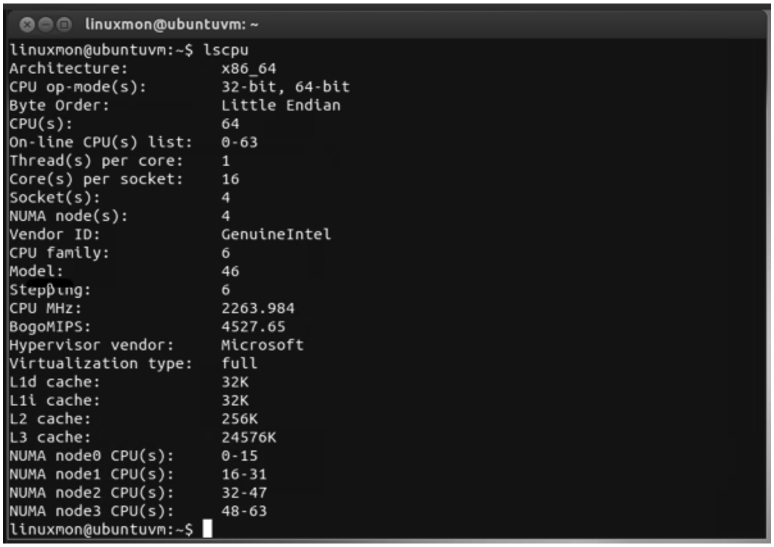
Some of the new scalability limits may seem ridiculously large: 64TB virtual hard disks, 1TB of memory in a single VM, and even 64 vCPUs in a single VM. But the point now is that almost any workload can be virtualized with Windows Server 2012 Hyper-V. To illustrate this capability to virtualize almost any workload, Microsoft released a statement that more than 99 percent of the world's SQL Server deployments could now run on Windows Server 2012 Hyper-V. One aspect that is important to the 64TB VHDX scalability is that it removes most scenarios of having to use pass-through storage, which maps a virtual machine directly to raw storage. The goal of virtualization is to abstract the virtual machine environment from the physical hardware. Directly mapping a virtual machine to physical storage breaks this abstraction and stops some features of Hyper-V from being used, such as checkpoints, Live Migration, and Hyper-V Replica. In all my years of consulting, I have never seen an NTFS volume 64TB in size. In fact, the biggest I have heard of is 14TB, but a 64TB limit means that VHDX scalability would not limit the storage workloads that could be virtualized.

Why Most Volumes Are Less Than 2TB

In most environments, it's fairly uncommon to see NTFS volumes greater than 2TB. One reason is that master boot record (MBR) partitioning had a limit of 2TB. The newer GUID Partition Table (GPT) removed this limitation, but volumes still stayed at around 2TB. Another reason concerns the unit of recoverability. Any set of data is typically restricted to the amount of data that can be restored in the required time frame. Legacy backup/restore solutions that were tape based could limit how large data sets would be, but modern backup/restore solutions that are primarily disk-based remove this type of limit.

The number one reason for limits on volumes is a corruption occurring on the NTFS volume. If a corruption occurs, the ChkDsk process must be run, which takes the volume offline while the entire disk is scanned and problems are repaired. Depending on the disk subsystem and its size, this process could take hours or even days. The larger the volume, the longer ChkDsk will take to run and the longer the volume would be offline. Companies would limit the size of volumes to minimize the potential time a volume would be offline if ChkDsk had to be run. In Windows Server 2012, ChkDsk has been rearchitected to no longer take the volume offline during the search for errors. Instead, it has to take the disk offline only to actually fix the problems discovered during an online scan. The maximum possible offline time for a volume is now 8 seconds, no matter how large the volume. With this change, we can expect to see larger NTFS volumes as organizations adopt Windows Server 2012 and above.

Also important to note about scalability is that only very large virtual machines can be created with tens of virtual processors, but the non-uniform memory access (NUMA) topology is passed to the virtual machine, enabling the most optimal levels of performance. This scalability applies to both Windows guest operating systems and Linux, as Figure 1.6 shows with a 64 vCPU Linux virtual machine. Also note in the figure the awareness of the NUMA nodes. This was another investment area in Windows Server 2012: making Linux a first-class guest operating system. Nearly every feature of Hyper-V worked equally for Windows guests and Linux guests.



```
linuxmon@ubuntuvm: ~
linuxmon@ubuntuvm:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                64
On-line CPU(s) list:   0-63
Thread(s) per core:    1
Core(s) per socket:    16
Socket(s):             4
NUMA node(s):          4
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 46
Stepβιng:              6
CPU MHz:               2263.984
BogoMIPS:              4527.65
Hypervisor vendor:     Microsoft
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              24576K
NUMA node0 CPU(s):     0-15
NUMA node1 CPU(s):     16-31
NUMA node2 CPU(s):     32-47
NUMA node3 CPU(s):     48-63
linuxmon@ubuntuvm:~$
```

FIGURE 1.6
Linux virtual machine running on Windows Server 2012 Hyper-V with 64 vCPUs

MOBILITY AND AVAILABILITY

As virtual machines became more scalable, the workloads that could be virtualized increased exponentially, which makes keeping the virtual machines available even more important. Windows Server 2012 made great advancements to the mobility and resiliency of virtual machines. Windows Server 2008 R2 had introduced Live Migration as a means to move virtual machines between nodes in a cluster that had shared storage. Windows Server 2012 took this to the next level by allowing multiple concurrent Live Migrations, which it would autoscale based on available bandwidth and would queue until they could be performed based on network bandwidth availability.

A big shift for Hyper-V architecture options was support of SMB 3 for the storage of virtual machines. This allows Hyper-V virtual machines to be run from SMB 3 file shares, enabling a new file-based storage option. This change made it possible for Windows Server 2012 file-share clusters to be used as the shared storage for Hyper-V environments in addition to any NAS or SAN solutions that support SMB 3. By using SMB 3 as the storage for virtual machines, an additional type of Live Migration was enabled, SMB Live Migration, which enabled virtual machines to be moved between *any* two Windows Server 2012 Hyper-V hosts, even if they were not part of a cluster. The Live Migration and SMB Live Migration processes remained similar, except that the handles and locks to the files on the SMB share are transferred between hosts as part of the SMB Live Migration process.

Storage Live Migration was introduced with Windows Server 2012 Hyper-V. It allows all of the storage-related items of a virtual machine to be moved between supported storage mediums with no downtime to the virtual machine. This included the virtual machine's configuration files, checkpoint data, smart paging files, and virtual hard disks. Any and all of these can be moved with no interruption to the virtual machine's availability. While this was an important feature to have because it was available in other virtualization solutions, its use must be accompanied with extreme caution. Consider the amount of I/O required to move the storage of a virtual machine, both reading from the source and writing to the target. If a storage subsystem is having performance issues, which is a reason to want to move the virtual machine, then performing a storage migration would add substantial I/O load and would likely worsen the situation in the short term. It is, however, an important feature to have and enables the true "Wow" mobility feature of Windows Server 2012, Shared Nothing Live Migration.

The ability to move a virtual machine without any constraints is the utopian goal of any virtualization solution: to be able to move a virtual machine between any hosts in the datacenter and between different storage subsystems without any downtime using only a 1Gbps network connection. Windows Server 2012 delivers this in Windows Server 2012 with Shared Nothing Live Migration. Shared Nothing Live Migration allows a virtual machine to be moved between stand-alone hosts, from a cluster to a stand-alone, from a stand-alone to a cluster, or from cluster to cluster without any interruption to virtual machine communication. A Storage Live Migration is performed first if required to move the storage of the virtual machine to the destination. Then it is synchronized while the memory of the virtual machine is copied, and synchronized again before the virtual machine is flipped and started on the destination. Being able to move virtual machines anywhere in the datacenter with no downtime is a useful capability, but the same cautions related to Storage Live Migrations apply—understand the impacts of moving virtual machines.

Mobility is important for moving virtual machines in planned scenarios to enable hardware and software maintenance on hosts without affecting the availability of virtual workloads. Beyond that, though, is making services available in unplanned events such as power outages, host crashes, and natural disasters. Windows Server 2012 greatly improved Failover Clustering, which is the backbone of Hyper-V high availability. However, what many customers asked for was a disaster-recovery (DR) feature that would allow an asynchronous replication of virtual machines from one datacenter to another. Hyper-V Replica provides this capability exactly, allowing the virtualized storage of a virtual machine to be replicated to a DR location Hyper-V server every 5 minutes in addition to providing numerous failover options, including the ability to test failover without impacting production replication. I cover high availability and disaster recovery in great detail later in the book, and I don't consider Hyper-V Replica the answer to all DR situations. Hyper-V Replica, which provides asynchronous replication between a primary VM and a replica VM, is one available tool that works well in specific scenarios.

Why Is Asynchronous Replication a Good Thing for Disaster Recovery?

Typically, synchronous is best for any kind of replication. With synchronous replication, a change made to the primary store is not committed until it is also written to the secondary store. For the best assurance of data integrity and to ensure no loss, this is a good thing. However, synchronous replication has a substantial cost. The connectivity required for synchronous replication needs to be resilient and fast enough, with a low enough latency to ensure that the performance of the primary workload is not negatively affected. For the replication of a virtual machine across datacenters, only the highest levels of connectivity would enable the storage replication without affecting the primary workload. Although these solutions are possible, they are typically part of SAN solutions, which are usually costly. With asynchronous replication, the primary workload is not affected, and the changes are replicated to the secondary store as quickly as possible or on a fixed interval. This achieves a good level of protection without requiring very fast, low-latency network connections, but it is not real-time replication. In the event of an unplanned failover to the DR site, a few minutes of data may be lost, but in a true disaster, a few minutes of state loss is typically accepted. Asynchronous brings disaster recovery to all workloads rather than just the tier 1 services that can utilize SAN-level synchronous replication.

Other Capabilities

Windows Server 2012 Hyper-V introduced many other capabilities that greatly change virtual environments:

- Virtual Fibre Channel support that allows virtual machines to communicate directly with Fibre Channel–connected SANs, which is a necessity for guest clustering scenarios that need shared storage and cannot use iSCSI
- Network virtualizing that enables complete abstraction of the network viewed by virtual machines from the physical network fabric, enabling complete isolation between virtual environments and also enabling environments to span multiple datacenters without having to modify IP configuration
- SR-IOV(Single Root I/O Virtualization) and dynamic VMQ for the highest level of virtual machine network performance
- Improvements to Dynamic Memory

When I created presentations for Windows Server 2012 Hyper-V, I used a single slide that showcased the majority of the new Hyper-V features (Figure 1.7) and, as noted, all of the new capabilities, none of which affected the ability to live-migrate virtual machines. These technologies are all covered throughout this book.

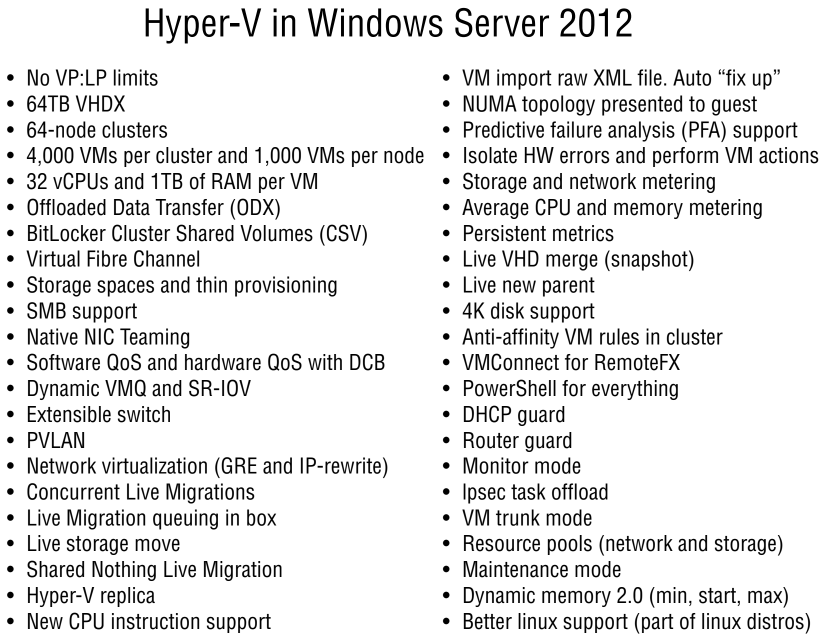


Figure 1.7
The major new features of Windows Server 2012 Hyper-V

Hyper-V in Windows Server 2012

- No VP:LP limits
- 64TB VHDX
- 64-node clusters
- 4,000 VMs per cluster and 1,000 VMs per node
- 32 vCPUs and 1TB of RAM per VM
- Offloaded Data Transfer (ODX)
- BitLocker Cluster Shared Volumes (CSV)
- Virtual Fibre Channel
- Storage spaces and thin provisioning
- SMB support
- Native NIC Teaming
- Software QoS and hardware QoS with DCB
- Dynamic VMQ and SR-IOV
- Extensible switch
- PVLAN
- Network virtualization (GRE and IP-rewrite)
- Concurrent Live Migrations
- Live Migration queuing in box
- Live storage move
- Shared Nothing Live Migration
- Hyper-V replica
- New CPU instruction support
- VM import raw XML file. Auto “fix up”
- NUMA topology presented to guest
- Predictive failure analysis (PFA) support
- Isolate HW errors and perform VM actions
- Storage and network metering
- Average CPU and memory metering
- Persistent metrics
- Live VHD merge (snapshot)
- Live new parent
- 4K disk support
- Anti-affinity VM rules in cluster
- VMConnect for RemoteFX
- PowerShell for everything
- DHCP guard
- Router guard
- Monitor mode
- Ipsec task offload
- VM trunk mode
- Resource pools (network and storage)
- Maintenance mode
- Dynamic memory 2.0 (min, start, max)
- Better linux support (part of linux distros)

I have focused on the changes to Hyper-V so far. However, many other changes in Windows Server 2012 enabled Windows Server 2012 to be an even better foundation for many Hyper-V services, such as changes to Failover Clustering, the new SMB 3 protocol, configuration levels that enable a server to be switched between server core and server via a GUI without having to reinstall, native NIC Teaming, Server Manager, PowerShell v3, and much more. In addition, I cover the non-Hyper-V features of Windows Server throughout this book where appropriate and where they bring value to a virtual experience.

Windows Server 2012 R2

I look at Windows Server 2012 Hyper-V as a whole new generation of Hyper-V from the previous versions. It took Hyper-V to new levels of scalability and functionality and made it a true enterprise hypervisor, bringing in major new technologies such as Hyper-V Replica, Network Virtualization, SMB 3 usage, and Live Migration. I look at Windows Server 2012 R2 as the continued advancement of the Hyper-V technology, refining many of the capabilities based on the feedback of enterprises that deployed Windows Server 2012 Hyper-V. Many organizations will welcome the 2012 R2 enhancements.

No scalability changes were made in Windows Server 2012 R2. I think most people would agree that the scalability of Windows Server 2012 meets today's and tomorrow's requirements. The focus was on improving the utilization of environments and fully embracing the technologies that companies were utilizing.

Generation 2 Virtual Machine

The format of virtual machines has not really changed since the first version of Hyper-V. Ten years ago, virtual machines required a lot of emulated hardware, because operating systems didn't natively understand virtualization. This is no longer true today. Nearly all modern operating systems understand virtualization and the synthetic types of resources available, making the emulated hardware previously required for compatibility not required.

Windows Server 2012 R2 introduces a new type of virtual machine, a generation 2 virtual machine, which removes all of the legacy emulated hardware previously present and shifts to a UEFI-based (User Extensible Firmware Interface) virtual machine exclusively using synthetic SCSI (allowing virtual machines to now boot from the synthetic SCSI) and network adapters (including PXE boot from a synthetic network adapter). Generation 1 virtual machines are still available, and there is no real performance improvement of a generation 1 vs. generation 2 virtual machine after the OS is installed and running, but a generation 2 virtual machine will install and boot faster.

Storage Enhancements

One feature that did not make Windows Server 2012 Hyper-V was the capability to dynamically resize a VHDX attached to a running machine. For some organizations, just adding VHD/VHDX files to a running virtual machine was not sufficient. 2012 R2 Hyper-V supports the dynamic resizing of VHDX files attached to the virtual machine's SCSI controller. This dynamic resizing supports both increasing the size and reducing the size, provided sufficient unpartitioned space exists within the VHDX file.

VHDX files can be shared among multiple virtual machines in 2012 R2 Hyper-V; and these shared VHDX files, which are hosted on Cluster Shared Volumes or a scale-out file server, are seen to the virtual machines as shared SAS storage and can be used as shared storage within the virtual machine for guest clustering scenarios. This removes the previous requirement to use iSCSI or virtual Fibre Channel to enable shared storage within virtual machines for guest clustering purposes.

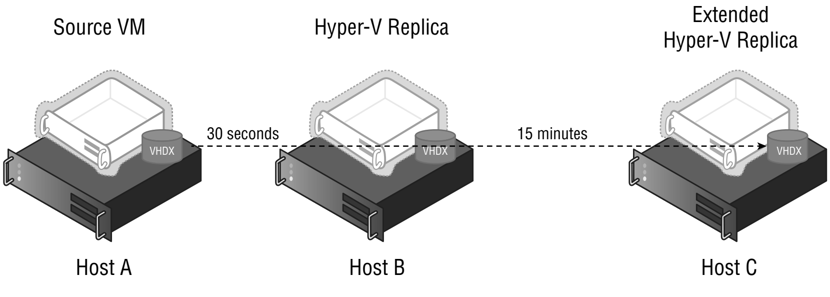
Resource metering was introduced in 2012 Hyper-V for processor, memory, and network, but not storage (other than the amount of storage used). In Windows Server 2012 R2, the resource metering is expanded to give more detail on the I/O profiles of storage, including average IOPS and data read and written. 2012 R2 also allows QoS to be used with storage to restrict the maximum IOPS of each individual virtual hard disk and also can alert administrators if the IOPS drops below a certain threshold.

Mobility and Availability

Live Migration in Windows Server 2012 may seem to be the perfect solution, covering all scenarios, but in 2012 R2, it has been made more efficient. The Windows 2012 Live Migration method of copying memory over the networks specified for Live Migration is still available in 2012 R2. However, the default now utilizes compression, which reduces the amount of data sent over the network, thus reducing Live Migration durations potentially by a factor of five at the expense of some extra CPU cycles to both compress the memory at the source and decompress at the target. Another option is to utilize SMB Direct as the transport. This may not seem like a good option initially, but the goal is to use it if the network adapters support remote direct memory access (RDMA), which allows it to be used; SMB Direct will be faster than even compressed Live Migration, but it uses almost no CPU. Windows Server 2012 R2 also allows Live Migration from Windows Server 2012, which allows organizations to migrate from 2012 to 2012 R2 without downtime for virtual machines.

Hyper-V Replica is also enhanced to allow different choices for the frequency of the asynchronous replication of the storage changes. The 5-minute frequency from Windows 2012 Hyper-V is still available, but additional options of 30 seconds and 15 minutes are also now offered (see Figure 1.8). Extended Hyper-V Replica can be configured, allowing a replica to be created of a replica. Note that the extended replica is sourced from the existing replica and not from the original virtual machine. This is a useful capability for organizations using Hyper-V Replica within a datacenter who also want an additional replica in a separate datacenter for true DR.

Figure 1.8
Extended Hyper-V Replica allows different replication intervals between the replicas



Other Capabilities

One major new feature in Windows Server 2012 R2 is the inclusion of a network virtualization gateway, which is critical to allowing different virtual networks to communicate and also to be able to communicate with the physical network fabric. Prior to 2012 R2, a hardware gateway was required, and there really were not many of them.

In 2012 R2, it's possible to export virtual machines and virtual machine checkpoints while they are running, enabling a simple cloning process that can be useful, especially in development and testing environments.

More capabilities were added for Linux virtual machines, including Dynamic Memory, live backup offering file consistency, and Hyper-V Replica IP reconfiguration during failover.

Activation can be a pain point in virtual environments. In Windows Server 2012 R2 Hyper-V, if the Hyper-V host is running Datacenter edition and is activated, then any Windows Server 2012 R2 virtual machine (Essentials, Standard, or Datacenter) on the server will automatically activate. No need for KMS (Key Management Service) or Active Directory–Based Activation (ADBA). If the VM leaves the host, it will deactivate. The only required action is to use the Automatic Virtual Machine Activation key in the guest OS, which can be found at the following location:

```
http://technet.microsoft.com/en-us/library/dn303421.aspx
```

Windows Server 2016

Windows Server 2016 continues the evolution of Windows and Hyper-V with a key theme of the cloud fabric that drives a number of innovations, including how applications are hosted and security. Providing a platform that can host the latest "born-in-the-cloud applications" while being used on premises, in Azure, and by hosting partners is central to the Windows Server 2016 release.

Windows Server 2016 makes some major changes to the scalability of the hypervisor host and the VMs as shown in Table 1.2.

Table 1.2: Windows Server 2016 Scalability Improvements

Resource Maximum	Windows Server 2012/2012 R2	Windows Server 2016
Physical (host) Memory	4 TB	24 TB (6x improvement)
Physical (host) Logical Processor	320	512
VM Memory	1 TB	12 TB (12x improvement)
VM vCPUs	64 vCPUs	240 vCPUs (3.75x improvement)

```
https://blogs.technet.microsoft.com/windowsserver/2016/08/25/windows-server-
scalability-and-more/
```

Containers and Nested Virtualization

Containers provide a sandbox for creating applications; these containers can contain the application, configuration, and details of dependencies such as libraries and runtimes. This enables simple and consistent deployment of applications, isolation from other applications, centralized management and storage, in addition to granular resource control.

Containers have been available in Linux distributions for a while and have gained adoption with Docker, which offered a standardized management solution, container technology, and library. Windows Server 2016 brings container technology to Windows for Windows applications in two types: Windows Containers and Hyper-V Containers that, while utilizing the same container technology, enable a deployment time choice to be made as to the level of isolation required for the application: user mode or kernel mode isolation. Management can be performed using PowerShell or Docker.

Enabling the kernel-mode isolation capability via Hyper-V Containers requires creating virtual machines that previously would have been impossible if the container host OS was a virtual machine, as creating a VM within a VM (nested virtualization) was not possible. Windows Server 2016 enables nested virtualization for Hyper-V Containers and general nested virtualization needs.

Shielded VMs

Shielded VMs provide protection for the data and state of the VM against inspection, theft, and tampering from administrator privileges. Shielded VMs work for generation 2 VMs that provide the necessary Secure Boot, UEFI firmware, and virtual TPM (Trusted Platform Module) (vTPM) 2 support required. While the Hyper-V hosts must be running Windows Server 2016, the guest operating system in the VM can be Windows Server 2012 or above and, shortly after the Windows Server 2016 release, Linux guest VMs.

A new Host Guardian Service instance is deployed in the environment, which will store the keys required to run shielded VMs for authorized Hyper-V hosts if they can prove that they're healthy through various types of attestation. A shielded VM provides the following benefits:

- BitLocker-encrypted disks
- A hardened VM worker process (VMWP) that helps prevent inspection and tampering
- Automatically encrypted Live Migration traffic as well as encryption of its runtime state file, saved state, checkpoints, and even Hyper-V Replica files
- No console access in addition to blocking PowerShell Direct, Guest File Copy Integration Components, and other services that provide possible paths from a user or process with administrative privileges to the VM

Other Capabilities

Windows Server 2016 provides two distinct groups of new capabilities for Hyper-V: those that are part of the Hyper-V role and those that the Hyper-V role will benefit from. Both are equally important in many scenarios, but there is a definite theme of enabling Windows Server 2016 and Hyper-V to be the definitive platform for the cloud on-premises, in hosting partners, and in Microsoft's own Azure public cloud for Windows and Linux workloads. Customers will have a choice of how to deploy their services without having to change how they write their applications and complete hybrid options.

When first considering the Hyper-V role specifically, there is a new VM hardware version available, version 7, that enables the new features discussed in the rest of this section. A version 7 virtual machine can be used only on a Windows Server 2016 host and uses a new binary VMCX configuration file instead of the old XML-based configuration that was prone to corruption. Generation 2 VMs can now have memory and network adapters hot-added and removed, providing more flexibility in VM resource configuration. Virtual TPMs are also now available for generation 2, hardware version 7 VMs, enabling high-security features such as shielded VMs and BitLocker. Linux VMs can now use the Secure Boot feature initially introduced for Windows VMs in Windows Server 2012 R2.

When looking at the rest of Windows Server 2016, many of the features, while usable by many technologies, certainly have Hyper-V as the focus role that will benefit from the technology. The new Nano Server deployment option for Windows Server, which features a completely refactored architecture that is a fraction of the size of a Server Core deployment, is the recommended option for cloud fabric servers, Hyper-V servers, and file servers, in addition to born-in-the-cloud application servers. Nano Servers are quick to deploy, require less patching and rebooting, and have no real local interface, but they can be managed remotely in rich ways. Windows Server 2016 has new builds released at a far greater pace than the regular once-every-two-years frequency to which we have grown accustomed. To enable easy adoption of new builds, rolling upgrades will be supported that allow a mix of Windows Server 2016 builds in a single cluster, and this functionality also extends to Windows Server 2012 R2, allowing organizations to add Windows Server 2016 nodes in their existing Windows Server 2012 R2 clusters. Major new storage technologies enable new types of replication and new ways to use direct-attached storage in cluster nodes.

Licensing of Hyper-V

The most painful aspect of most virtual environments is understanding the licensing of the hypervisor, the operating systems running in the virtual machines, and the management software. I don’t want to go into great detail about licensing in this book because, despite new licensing agreements, special combinations of licensing still exist through agreements with programs such as Server and Cloud Enrollment (SCE) and the legacy Enrollment for Core Infrastructure (ECI). For most organizations, the licensing is simple with Windows Server 2012 and above; however, changes in Windows Server 2016 are important to understand.

One Operating System (Well Two, but Really One) with Windows Server 2012 and 2012 R2

Prior to Windows Server 2012, numerous versions of Windows Server existed—Web, Standard, Enterprise, and Datacenter—and each version had different capabilities and different limits and were licensed differently. That all goes away in Windows Server 2012 and above; for medium and large companies, there are only two versions of Windows Server: Windows Server 2012 R2 Standard and Windows Server 2012 R2 Datacenter. Both versions are *exactly* the same:

- They have the same limits, both supporting 64 processor sockets, 640 logical processors (320 with Hyper-V role enabled), and 4TB of memory.
- Both have the same roles and features; for example, even Standard has Failover Clustering.
- They are essentially bit-for-bit the same operating system, other than that each shows different versions in the About menu option and different background wallpaper.
- Both are licensed in two-socket increments, and all sockets in the server must be licensed. If a server has four sockets, then two licenses of either Standard or Datacenter must be purchased.

The difference between Standard and Datacenter is in operating system environments (OSEs), or virtual instances for each license. This is the number of virtual machines running Windows Server that are included as part of your license: Standard allows two virtual instances per license, and Datacenter allows unlimited instances. From a virtualization environment perspective, this is a big difference. For each Standard license, I can run two virtual machines running Windows Server, while with Datacenter, I can run an unlimited number of virtual machines. Standard edition is now targeted at physically deployed operating system instances or very light virtualization, while Datacenter is targeted at virtualization hosts.

It is possible to stack licenses—for example, buying three Standard licenses for a server would allow me to run six virtual machines running Windows Server (each Standard license allows two "slots," with each "slot" supporting a Windows Server virtual machine), which would be cheaper than buying a Datacenter license. However, complications will occur if you want to move virtual machines between hosts.

Consider Figure 1.9, which shows two Hyper-V hosts in a remote office that needs only six virtual machines. The option shown in the example is using three copies of Windows Server Standard on one server and a single copy on the other server, and this is allowed. However, suppose you want to move the virtual machines to the other server, as shown in Figure 1.10, to perform maintenance on the first server. This can be done, but it requires moving two of the Windows Server Standard licenses between physical hosts. License mobility allows the movement of licenses only every 90 days, which means that you could move the virtual machines and the licenses, but you would not be able to move the virtual machines back for 90 days.

FIGURE 1.9
Using stacked Standard licenses for virtual machines

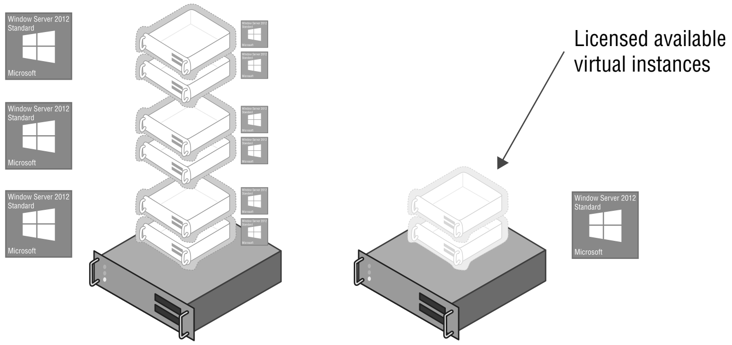
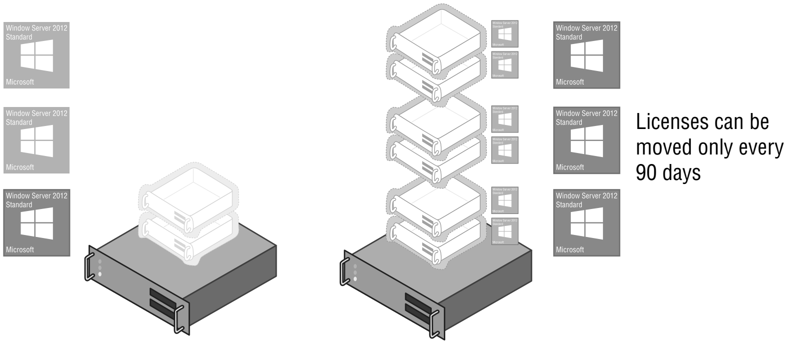

Window Server 2012 Standard
Microsoft
Licensed available virtual instances

FIGURE 1.10
Moving Standard licenses to enable licensed virtual machine migrations


Window Server 2012 Standard
Microsoft
Licenses can be moved only every 90 days

To allow free movement of the virtual machines, the high watermark of virtual machines ever present on the hosts would need to be used to calculate the required number of licenses, which would therefore be three copies of Standard on both servers, as shown in Figure 1.11. Now consider having 8, 10, or 20 virtual machines and having clusters of 16 or even 64 hosts. The unlimited number of virtual machines that accompanies the Datacenter edition makes much more sense, as shown in Figure 1.12. Using Datacenter enables highly dense deployments of virtual machines without you needing to worry about the licensing of the virtual machines.

Figure 1.11
Required Standard licensing to enable virtual machine mobility

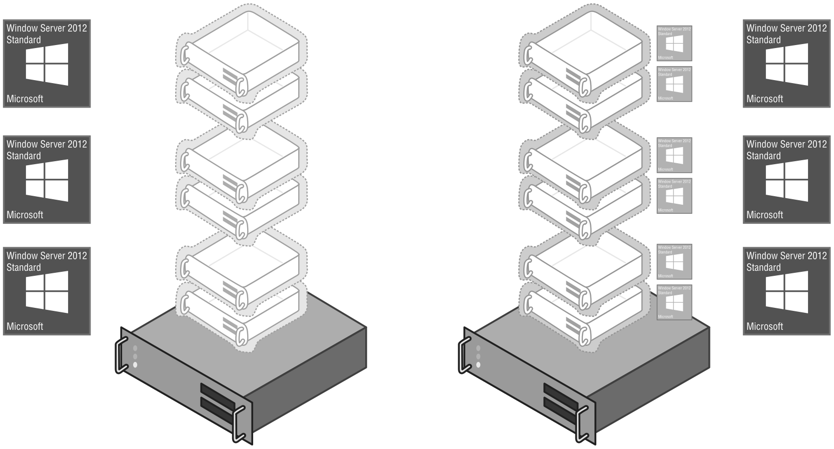
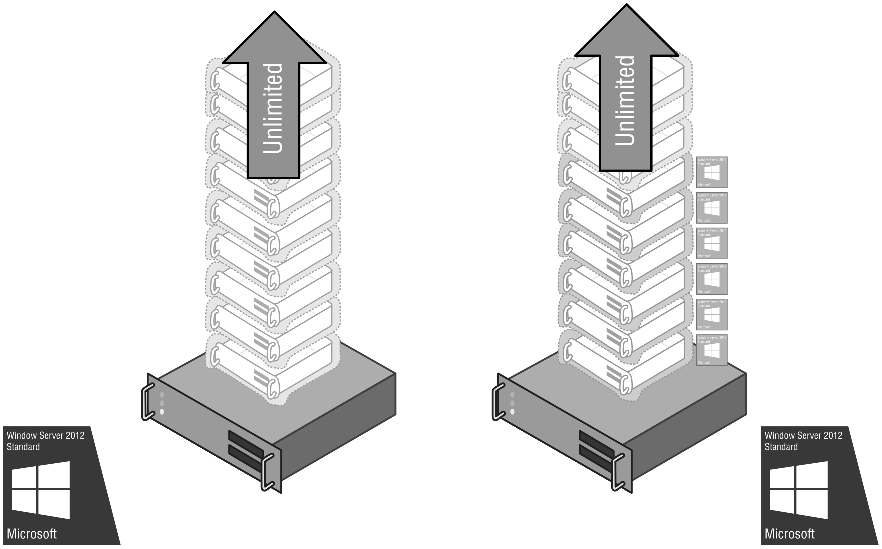


Figure 1.12
Using Datacenter to enable an unlimited number of virtual machines on the hosts for full mobility



It's important to realize that the use of Standard or Datacenter is not related to Hyper-V specifically, but rather the licensing of the operating systems running inside the virtual machines, and the same would apply to any hypervisor, such as XenServer or ESX.

This is an important point. Standard vs. Datacenter relates to the number of virtual instances running the Windows Server operating system. If you need to run something other than Windows Server (for example, Linux virtual machines or Windows Client virtual machines such as for a VDI environment), then these virtual instances do not apply and you need to license those operating systems to whatever licensing scheme is required. There is no limit to the number of virtual machines that you can run on Windows Server Standard Hyper-V, and it would be possible to have hundreds of virtual machines running Linux or Windows Client without the need to use Datacenter or have multiple Standard licenses.

In fact, there is another option if a virtual environment needs to run Linux or Windows Client exclusively, and no virtual instance rights for Windows Server are required. Microsoft makes available Microsoft Hyper-V Server, which is a free download from Microsoft that is designed for environments that don't wish to run Windows Server virtual machines and don't need the virtual instance rights included with the Standard or Datacenter edition, making it perfect for Linux and VDI environments. Microsoft Hyper-V Server is updated with each version of Windows Server, making the version that's currently available Microsoft Hyper-V Server 2012 R2, and it has all of the same capabilities of the version of Hyper-V that is available in Windows Server, but only the Hyper-V role is included. It cannot be a file server or a domain controller or be used for any other role, nor can the graphical interface or server management tools be installed; it runs in the Server Core configuration level.

Windows Server 2016 Changes to Licensing and Versions

While the virtual OSE rights of Standard and Datacenter remain the same for Windows Server 2016, there are two major changes:

- There are differences in features between the Standard and Datacenter SKUs.
- Windows Server has moved to per core licensing instead of per socket.

Standard vs. Datacenter

The introduction of changes in functionality between Standard and Datacenter may concern some readers that the technology they currently enjoy in the Standard edition will be missing in Windows Server 2016. However, that is not the case. No functionality is being removed from the Standard SKU of Windows Server 2016, but rather some of the new features in the 2016 version will be available only in the Datacenter SKU, specifically:

- Enterprise storage features, specifically Storage Spaces Direct and Storage Replica
- New network virtualization stack inspired and consistent with Azure
- Shielded VMs

Other features, such as Nano Server, containers, clustering changes, and everything else unless otherwise stated will be common to both the Standard and Datacenter SKU.

Windows Server 2016 Licensing

The per socket licensing of Windows Server 2012 (at least two-sockets are licensed for any node and purchased in two-socket increments) struggles in two major ways for modern deployments:

- Modern processors have an increasing number of cores, with the new many-core processors featuring more than 50 cores per socket. This would result in staggering numbers of VMs running on hosts with a single datacenter license, which does not make business sense for Microsoft.
- Cloud providers such as Azure and other hosters operate services based on vCPUs assigned to VMs where no actual physical sockets are visible, which makes any licensing based on sockets incompatible. A move to per socket licensing enables consistent and simple licensing across hybrid environments.

SQL Server 2012 made the switch to per core licensing, and this continues with Windows Server 2016 and System Center 2016. Both Standard and Datacenter are sold in two-core pack licenses with the following rules:

- Every socket must be licensed for at least eight cores (four two-core packs).
- Every server must be licensed for at least sixteen cores (eight two-core packs).
- Every core must be licensed.

If you compare this to the 2012 model, it is consistent; every server had to be licensed for at least two sockets, and most servers had processors with eight cores or less. Therefore, provided your servers have processors with eight cores or less, your licensing costs for Windows Server 2016 will be the same as with Windows Server 2012 R2. If you have processors with more, you should work with your Microsoft account representative, as there may be options to make the transition seamless. For customers with licensing and enterprise agreements, there will be grants of eight two-core packs for each existing two-socket Windows Server 2012 R2 license. If processors have more than eight cores, then the deployment may be under-licensed and additional two-core license packs may need to be purchased.

For Datacenter, an unlimited number of OS instances running Windows Server continue to be granted. However, the stacking of Standard changes. For Windows Server 2016, two OS instances running Windows Server are granted if all cores are licensed, but this is different. In Windows Server 2012, if you had a four-processor server, you would buy two copies of Standard (two processors each) to cover all sockets, and each came with two OS instance rights, giving four in total. Additional two-socket licenses could be purchased to get two more OS instances. For Windows Server 2016, if you have a four-socket server with eight cores each (or fewer cores—every socket still has to be licensed for eight cores, remember), you would need to buy sixteen two-core licenses (the financial equivalent of two old licenses) but you have covered the cores only once and you get two Standard OS instances for Windows Server, half the number of OS instance rights. If you wanted to stack Standard to get another two OS instance rights, you would have to license *every* core again, buying another sixteen two-core licenses. This is a major change; however, while this sounds daunting, very few organizations stack Windows Server Standard on servers with more than two sockets. Nevertheless, if you are one of those organizations, you should start conversations with your Microsoft account representative now. Stacking on systems with sixteen cores or less will work the same as Windows Server 2012.

Microsoft has a good licensing document that I recommend reading:

```
http://download.microsoft.com/download/7/2/9/
7290EA05-DC56-4BED-9400-138C5701F174/
WS2016LicensingDatasheet.pdf
```

Table 1.3 shows the number of two-core packs required, based on the number of sockets and the cores per socket in a system. It also indicates that extra licensing for Windows Server 2016 may be required (denoted with an exclamation point), if you have two sockets or more with more than eight cores per socket.

Table 1.3: Licensing Cost Changes for Windows Server 2016 vs. Windows Server 2012 R2

		Physical cores per processor				
		2	**4**	**6**	**8**	**10**
Procs per server	**1**	8	8	8	8	8
	2	8	8	8	8	10 !
	4	16	16	16	16	20 !

Microsoft 2016 Licensing Datasheet

Choosing the Version of Hyper-V

Given the information in the previous section, determining which version of Hyper-V is required is a fairly simple decision. While it is technically possible to mix Standard and Datacenter in a single cluster, this makes tracking licensing complex. I use the following criteria to decide which version of Hyper-V I need in a virtual environment:

- If the virtual machines will all be running non–Windows Server operating systems, use the free Microsoft Hyper-V Server.
- If the environment will be running only a few virtual machines with no plans to expand and with limited mobility required, then the Standard edition of Windows Server can be used. However, with the new stacking changes in Windows Server 2016 Standard, it is likely to be used only where very low numbers of virtual machines are needed with 16 cores or less, unless you are willing to incur additional costs above the current Windows Server 2012 R2 costs.
- If there will be more than a few virtual machines with future growth possible and full mobility of virtual machines required, use the Datacenter edition of Windows Server.

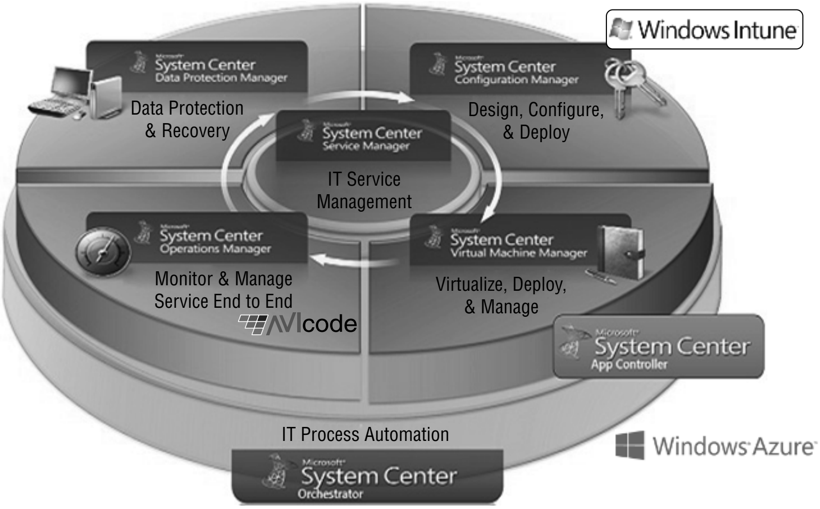
The Role of System Center with Hyper-V

The capabilities of the Hyper-V features that I described previously in this chapter are impressive, but it's important to realize that this is just for virtualization. Yes, Hyper-V is powerful and can enable almost any required scenario, but virtualization is the foundation and not the complete solution.

A production environment of any kind needs management services, and virtualization adds requirements to those management capabilities. For Windows Server and Hyper-V, the management solution is System Center. While it is possible to deploy Hyper-V without System Center in a small, limited capacity, it is required for any enterprise deployment. System Center comprises various components, and each is separately deployed and offers its own discrete capabilities. Moreover, while deployment of the entire System Center product offers numerous benefits, some organizations will deploy only certain components. For organizations wanting to deploy a true cloud with consistent capabilities with Azure, they can deploy Microsoft Azure Stack, which takes the Azure code and brings it on-premises running on top of Hyper-V.

Azure Stack is delivered in a prescriptive way with specific requirements and configurations and delivered at a rapid pace. System Center enables more flexible configurations, as you can manage all aspects of the environment, but with that comes more complexity. System Center will continue to run in a long-term servicing model with releases every couple of years. Chapter 9, "Implementing the Private Cloud, SCVMM, and Microsoft Azure Stack," details how System Center and Microsoft Azure Stack are leveraged. I briefly introduce all the components of System Center here because they will be discussed and used in the chapters preceding Chapter 9. Figure 1.13 shows the full System Center product.

FIGURE 1.13 Components of System Center


Windows Intune
Microsoft System Center Data Protection Manager
Data Protection & Recovery
Microsoft System Center Configuration Manager
Design, Configure, & Deploy
Microsoft System Center Service Manager
IT Service Management
Microsoft System Center Operations Manager
Monitor & Manage Service End to End
AVIcode
Microsoft System Center Virtual Machine Manager
Virtualize, Deploy, & Manage
Microsoft System Center App Controller
IT Process Automation
Microsoft System Center Orchestrator
Windows Azure

System Center is licensed in exactly the same way as Windows Server. It can be purchased in the Standard or Datacenter edition. The versions are identical except for the number of virtual instance rights: two for Standard and unlimited for Datacenter. It is licensed in two-socket increments for Windows Server 2012/2012 R2 and two-core increments for the 2016 version, which makes it easy to know how many and of what type of System Center licenses you need for your Windows Server environment. Typically, it will match your Windows Server licenses, and there are combination licenses available, such as ECI, which licenses Windows Server and System Center together.

It is important to note that System Center has been updating functionality throughout the Windows Server 2012/2012 R2 waves via update rollups (URs), which contain not only fixed but also new features. An example is a UR with added support for Azure IaaS in the SCVMM console.

Many organizations also extend System Center to the cloud with Operations Management Suite (OMS). OMS provides capabilities for on-premises and cloud-hosted systems that can integrate with System Center. OMS runs in the cloud without local infrastructure requirements, making it available to manage any workload, anywhere, and it can take advantage of the huge compute capabilities of the cloud. Some examples of capability include the following:

- Insights into your environment including log analysis from Windows and Linux systems and trend analysis of systems helping plan resources. This data can be gained through an OMS agent or through connectivity to Operations Manager.
- Automation allowing PowerShell to execute in the cloud
- Backup and recovery solutions
- Security and auditing through network, host, and audit inspection in addition to threat analysis

Long-term and even day-to-day management capabilities may be available in OMS. However, today OMS is best utilized with System Center; System Center manages the day-to-day, while OMS focuses on activities related to analytics and trend analysis.

System Center Configuration Manager

Moving through the products shown in Figure 1.13, I'll start with System Center Configuration Manager (SCCM). SCCM provides capabilities to deploy operating systems, applications, and OS/software updates to servers and desktops. Detailed hardware and software inventory and asset intelligence features are key aspects of SCCM, enabling great insight into an entire organization's IT infrastructure. SCCM 2012 introduces management of mobile devices such as iOS and Android through ActiveSync integration with Exchange and a user-focused management model. One key feature of SCCM for servers is settings management, which allows a configuration of desired settings to be defined (such as OS and application settings) and then applied to a group of servers (or desktops). This can be useful for compliance requirements.

Configuration Manager is closely aligned with the Windows client OS. As Windows 10 has shifted to being delivered at a frequent interval (approximately every four months in the new Windows as a Service paradigm), so too must Configuration Manager, in order to enable new Windows 10 functionality to be managed.

Configuration Manager has shifted to a naming convention of <year><month> to denote the version. For example, Configuration Manager 1511 represents the version released in November 2015 to coincide with the Windows 10 1511 release. Post Windows Server 2012 R2, Configuration Manager has added native support for mobile devices such as iOS and Android, where integration with Microsoft Intune is not possible or desired, in addition to new service plans that help manage the deployment of new branches of Windows 10 to groups of machines as they are released.

System Center Virtual Machine Manager and App Controller

Next in the circle of products in Figure 1.13, you see System Center Virtual Machine Manager (SCVMM). It gets a lot of attention in this book, but essentially it's the virtualization-specific management functionality across multiple hypervisors and gives insight and management into storage and network fabric resources. SCVMM allows the creation and deployment of virtual machine templates and even multitier services. It also lights up several Hyper-V features, such as network virtualization. App Controller provides a rich Silverlight web-based self-service interface for management of private and public cloud resources that, while useful, is removed in the 2016 version in favor of the Azure Pack interface.

SCVMM 2016 adds support for new Windows Server 2016 features such as Nano Server deployment and management, the new network virtualization stack, shielded VMs, and guardian host management in addition to simplifying the management of virtual environments.

System Center Operations Manager

System Center Operations Manager (SCOM) provides a rich monitoring solution for Microsoft and non-Microsoft operating systems and applications and also for hardware. Any monitoring solution can tell you when something is broken, and yes, SCOM does that. But its real power is in its proactive nature and best practice adherence functionality. SCOM management packs are units of knowledge about a specific application or component. For example, there is an Exchange management pack and a Domain Name System (DNS) for Windows Server management pack. The Microsoft mandate is that any Microsoft product should have a management pack that is written by the product team responsible for the application or operating system component. All of the knowledge of those developers, the people who create best practice documents, is incorporated into these management packs, which you can then just deploy to your environment. Operations Manager will raise alerts when potential problems are detected or when best practices are not being followed. Often customers object that when first implemented, Operations Manager floods them with alerts. This could be for various reasons (perhaps the environment has a lot of problems that should be fixed), but often Operations Manager will be tuned to ignore configurations that perhaps are not best practice but are nevertheless accepted by the organization.

Many third parties provide management packs for their applications and hardware devices. When I think about "it's all about the application" as a key tenant of the private cloud, the Operations Manager's ability to monitor from the hardware, storage, and network all the way through the OS to the application is huge, but it goes even further in Operations Manager 2012.

System Center Operations Manager 2012 introduced several changes, but two huge ones were around network monitoring and custom application monitoring. First, Microsoft licensed technology from EMC called SMARTS, which enables a rich discovery and monitoring of network devices. With the network discovery and monitoring functionality, Operations Manager can identify the relationship between network devices and services to understand, for example, that port 3 on this switch connects to server A. Then, if a switch problem occurs, Operations Manager will know the affected servers. CPU and memory information, among other types of information, is available for supported network devices.

The other big change was the acquisition by Microsoft of AVIcode, which is now Application Performance Monitoring (APM) in Operations Manager 2012. APM provides monitoring of custom applications without any changes needed by the application. APM currently supports .NET applications and Java Enterprise Edition (JEE).

Like SCVMM, Operations Manager 2016 investments include supporting all of the new Windows Server 2016 features but also extending monitoring support for LAMP stack, Azure, Office 365, and more. Additionally, Operations Manager has focused significant effort on easing the workload for administrators in understanding what management packs (MPs) are needed and if new versions are available. This now surfaces as Updates and Recommendations in the Operations Management console that will advise on new MPs and updates to MPs that will bring benefit to the environment. Additionally, the amount of "alert noise" (large numbers of alerts that muddy the data being viewed and therefore obstruct the viewing of alerts that you really care about) has been reduced, with more intuitive tuning via tune management packs.

System Center Data Protection Manager

System Center Data Protection Manager (DPM) is Microsoft's best-of-breed backup, continuous data protection, and recovery solution for key Microsoft workloads, including SharePoint, SQL Server, Dynamics, Exchange, Hyper-V, file services, and desktops. DPM allows granular recovery of information within the supported options for the product, including end-user self-recovery in certain scenarios. DPM can be useful in the private cloud, in the protection of the environment. DPM can back up and protect the Hyper-V servers, the SQL databases that are used by most of the System Center 2012 components, the management servers running the System Center infrastructure, and all of the virtual machines running on Hyper-V that are created.

DPM supports backing up at the Hyper-V server level, and that backup request will be passed by Hyper-V to the virtual machines. That allows the virtual machines to ensure that information on disk is in a backup-ready state so when the virtual machine is backed up, the integrity and usability of that backup can be ensured.

I do want to be clear; just because you can back up at the Hyper-V level does not mean that you should back up only at the Hyper-V level. If you want granular restoration capabilities of applications like SharePoint, SQL Server, and Exchange, you need to have the DPM agent installed within the virtual machine and actually be backing up from the VM directly, to enable DPM to have the knowledge of the application configuration and data.

System Center 2016 DPM adds support for the backup for VMware VMs in addition to better leveraging modern storage capabilities such as Storage Spaces Direct and even protect-shielded VMs. ReFS (Resilient File System) is utilized to streamline the creation of recovery points by utilizing ReFS cloning, therefore greatly increasing the number of sources that can be protected per DPM server and reducing the amount of storage required.

System Center Service Manager

I'll spend more time on System Center Service Manager (SCSM) in a later chapter, but think of it as the configuration management database (CMDB) for the entire infrastructure, which is another ITIL key capability. Service Manager is shown in the center of the rest of the System Center components for a good reason. It has connectors into all of the surrounding components, receiving feeds of information that it consolidates into a single view of everything related to an asset (such as a computer or person), giving a single point of truth for the entire organization.

Service Manager has capabilities commonly associated with a help desk solution, such as logging incidents, problems, and change requests, but it also handles change management and release management in addition to providing a powerful workflow engine to enable your organization's processes such as approvals to be replicated in Service Manager.

The key item that I focus on later is the service catalog, which provides the organization with the ability to request services, including services for software and virtual infrastructures. Organizations often have a help desk solution already in place, but realize that Service Manager is far more than a ticketing system. It can be implemented and integrated with another ticketing solution, all the while leveraged for its other powerful capabilities and CMDB functionality.

A welcome change in Service Manager 2016 is a new HTML5-based self-service portal that was previously a huge pain point for using Service Manager. Service Manager also integrates tightly with OMS.

System Center Orchestrator

System Center Orchestrator is the result of an acquisition of a product called Opalis, which has been renamed System Center Orchestrator as part of System Center 2012. Orchestrator provides two key capabilities that, as with Service Manager, I dive into in more detail in a later chapter.

First, Opalis was acquired because it had connectivity to many of the major datacenter applications and systems that exist, which with the acquisition now includes the Microsoft solutions. Integration packs exist for many systems and provide activities that are specific to the integration pack target, but Orchestrator can talk to targets that don't have integration packs, using many types of communication, including WMI, SSH, PowerShell, SNMP, and many more.

Second, Opalis had powerful runbook automation capabilities that leveraged all of this connectivity. Runbooks that were typically manually actioned by IT administrators and business users can be migrated to Orchestrator using an easy-to-use flowchart-type interface and can be completely automated. One shortcoming of Orchestrator is the limited capabilities of integration packs that would often result in having to use .NET activities to use PowerShell commands to complete functionality.

A PowerShell-based alternative is also provided, Orchestrator Service Management Automation (SMA), which enables standard PowerShell modules to be used as part of runbooks instead of the Orchestrator proprietary integration packs. It is Orchestrator SMA that is the future of Orchestrator and provides the greatest consistency with Azure Automation.

It is because of these capabilities that Orchestrator is shown as the foundation of the System Center product. All of the other System Center components can leverage Orchestrator for action requests made to other systems and complex processes. Orchestrator can talk to the rest of System Center, enabling automation of processes that use many components of System Center and other systems through a single runbook.

Clouds and Services

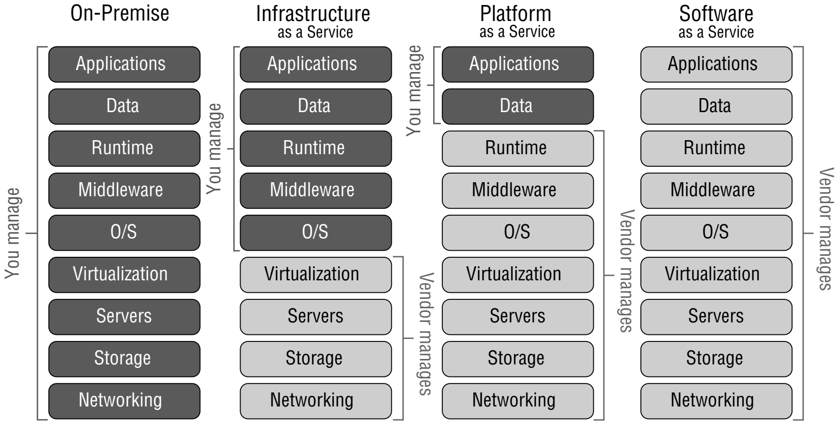
This book's primary focus is on Hyper-V, but the big technology investment area today is around various types of clouds and various types of capabilities offered "as a Service." I focus on several of these throughout the book, but in this section I provide a high-level summary of the types of clouds and "as a Service" offerings commonly seen, so they will make sense as I discuss their principles and use in later chapters.

There are two primary types of clouds: private and public. Virtualization focuses on services related to compute, such as creating, configuring, and running the virtual machines, but it does not focus on the storage or network fabrics that are major pieces of the datacenter. Virtualization does not help abstract the underlying resources from how they may be provisioned, and quotas to create resources are allocated to business units and users. Virtualization does not provide self-service capabilities and workflows to the clients. Cloud services enable this by providing rich management technologies that build on the virtualization foundation and enable intuitive, scalable, and controlled services that can be offered beyond just the IT team. With cloud services, different resources from the datacenter can be grouped together and offered to different groups of users with well-defined capabilities and capacity. There are many more benefits, and I go into more detail throughout this book.

Cloud services that are offered using an organization's internal resources are known as *private clouds*. Cloud services that are offered external to the organization, such as from a hosting partner or even solutions such as Microsoft Windows Azure, are called *public clouds*.

Within these clouds, different types of services can be offered, and typically these are seen from public cloud providers. There is, however, a movement of these types of services being offered in an organization's private cloud to its various business units, especially IaaS. There are three primary types of services: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). For each type, the responsibilities of the nine major layers of management vary between the vendor of the service and the client (you). Figure 1.14 shows the three types of service and a complete on-premises solution.

FIGURE 1.14
The key types of management and how they are owned for the types of cloud service



IaaS can be thought of as a virtual machine in the cloud. The provider has a virtual environment, and you purchase virtual machine instances. You then manage the operating system, the patching, the data, and the applications within. Examples of IaaS are Amazon Elastic Compute Cloud (EC2) and Azure IaaS, which give organizations the ability to run operating systems inside cloud-based virtual environments.

PaaS provides a framework in which custom applications can be run. Organizations need to focus only on writing the very best application within the guidelines of the platform capabilities, and everything else is handled. There are no worries about patching operating systems, updating frameworks, backing up SQL databases, or configuring high availability. The organization just writes the application and pays for the resources used. Azure is a classic example of a PaaS.

SaaS is the ultimate in low maintenance. The complete solution is provided by the vendor. The organization has nothing to write or maintain, other than configuring who in the organization should be allowed to use the software. A commercial example of SaaS is Hotmail, a messaging service on the Internet. An enterprise example is Office 365, which provides cloud-hosted Exchange, SharePoint, and Lync services all accessed over the Internet with no application or operating system management for the organization.

Ideally, for the lowest management overhead, SaaS should be used, then PaaS if SaaS is not available, and then IaaS if PaaS is not an option. SaaS is gaining a great deal of traction with services such as Office 365, but PaaS adoption is fairly slow. The primary obstacle for PaaS is that applications have to be written within specific guidelines to be able to operate in PaaS environments. Many organizations have many custom applications that cannot be modified or don't have the budget to change the application, which is why IaaS is so popular. With IaaS, an existing virtual machine on-premises can fairly painlessly be moved to the IaaS solution. In the long term, I think PaaS will become the standard for custom applications, but it will take a long time, and I think IaaS can help serve as the ramp to adopting PaaS.

Consider a multitiered service that has a web tier, an application tier, and a SQL database tier. Initially, all of these tiers would run as IaaS virtual machines. The organization may then be able to convert the web tier from IIS (Internet Information Services) running in an IaaS VM and use the Azure web role, which is part of PaaS. Next the organization may be able to move from SQL running in an IaaS VM to using SQL Azure. Finally, the organization could rewrite the application tier to directly leverage Azure PaaS. It's a gradual process, but the reduced overhead and increased functionality and resiliency at the end state is worth it.

As will be explored in this book, a key Microsoft differentiator is its hybrid capability to enable organizations to have a complete choice when deploying services, without having to change how they architect and create applications. When using Microsoft Azure Stack on-premises, an organization can write an application on the Azure Resource Manager (ARM) model and deploy it on-premises, to the public cloud, or to a hosting partner that leverages Microsoft Azure Stack. If a JSON (Java Script Object Notation) template is created to deploy a service to ARM, it can be deployed on-premises, to a hosting partner, or to Azure without modification. Typically, organizations will not pick one, but will utilize all of the options in the scenario where a particular type of hosting makes the most sense.

The Bottom Line

Articulate the key value propositions of virtualization. Virtualization solves the numerous pain points and limitations of physical server deployments today. Primary benefits of virtualization include consolidation of resources, which increases resource utilization and provides OS abstraction from hardware, allowing OS mobility; financial savings through less server hardware, less datacenter space, and simpler licensing; faster provisioning of environments; and additional backup and recovery options.

Master It How does virtualization help in service isolation in branch office situations?

Understand the differences in functionality between the different versions of Hyper-V. Windows Server 2008 introduced the foundational Hyper-V capabilities, and the major new features in 2008 R2 were Live Migration and Cluster Shared Volumes (CSV). Windows 2008 R2 SP1 introduced Dynamic Memory and RemoteFX. Windows Server 2012 introduced new levels of scalability and mobility with features such as Shared Nothing Live Migration, Storage Live Migration, and Hyper-V Replica in addition to new networking and storage capabilities. Windows 2012 R2 Hyper-V enhances many of the 2012 features with generation 2 virtual machines, Live Migration compression and SMB support, new Hyper-V Replica replication granularity, and Hyper-V Replica Extended replication. Windows Server 2016 builds on this with shielded VMs providing new levels of security for virtual environments, containers for new ways to deploy and manage applications, and other features and management enhancements.

Master It What is the largest virtual machine that can be created on Windows Server 2012 Hyper-V, and does this change for Windows Server 2016 Hyper-V?

Master It What features were enabled for Linux virtual machines in Windows Server 2016 Hyper-V?

Differentiate between the types of cloud service and when each type is best utilized. There are three primary types of cloud services: software as a Service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). SaaS provides a complete software solution that is entirely managed by the providing vendor, such as a hosted mail solution. PaaS provides a platform on which custom-written applications can run, and it should be used for new custom applications when possible because it minimizes maintenance by the client. IaaS allows virtual machines to be run on a provided service, but the entire OS and application must be managed by the client. IaaS is suitable where PaaS or SaaS cannot be used and in development/test environments.