Hyper-V Best Practices

Chapter No 1
"Accelerate Hyper-V Deployment"
In this package, you will find:

The author’s biography
A preview chapter from the book, Chapter no.1 "Accelerate Hyper-V Deployment"
A synopsis of the book’s content
Information on where to buy this book

About the Author

10 years ago, Hyper-V MVP Benedict Berger started working with Microsoft Virtualization Solutions. Especially with the first beta version of Windows Server "Longhorn", there were many new features he wanted to try out immediately. "I was particularly interested in Hyper-V, but at that time, there was hardly any experience on this topic, especially in German-speaking areas. Therefore, I had to work out many solutions on my own, and since then, I have been sharing the knowledge I gained with much more intensity and reach. I founded the German Hyper-V Community together with two friends a couple of years ago to promote this topic in Germany even further," he explains.

He currently works as a solution consultant for Elanity Network Partner GmbH. Elanity is an IT-service provider that specializes in cloud enablement in Germany. In this dream job, he helps his customers to adjust their IT solutions to their own business processes and to deploy "real" dynamic private and public clouds instead of just virtualizing some systems. So, the tools of his daily work are Windows Server with Hyper-V, System Center, and Microsoft Azure. Before joining Elanity, he worked for a German financial service provider and the global car giant Volkswagen in similar functions.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
He is a regular speaker at global conferences such as System Center Universe DACH, E2EVC Virtualization Conference and Microsoft TechEd as well as local conferences such as the TechNet Conference in Berlin, in which more than 500 experts participated.

He blogs on the German Virtualization Blog (http://blogs.technet.com/b/germanvirtualizationblog/) and on his personal blog (http://blog.benedict-berger.de). He is also the creator of the PDT GUI, a graphical user interface for the PowerShell Deployment Toolkit.

Acknowledgments

First of all, I have to thank my wife, Carina, for being there for me, every time, and making this book possible.

I thank the whole community for contributing best practices to Hyper-V and evolving the experience around a great virtualization solution. A special thanks goes to the folks at Microsoft and my fellow Hyper-V MVPs who have spent a lot of time each day to make this solution a perfect one.

A huge thank you goes to my reviewers for contributing a lot of time to ensure the quality of this book, especially Benjamin Armstrong, Lead Program Manager of Hyper-V, who is so busy creating the next version of Hyper-V but still took his time to improve this book. However, every reviewer, including Andreas Baumgarten, Carsten Rachfahl, Shannon Fritz, and Carlo Mancini did a great job ensuring this book's quality!

I thank my employer, Elanity Network Partner GmbH, and my bosses, Peter Schröder and Andreas Waltje, for allowing me the time and resources to support the Hyper-V community and making projects such as this book real.

I also thank my great team of coworkers at Elanity, especially Natascha Merker, who actively supported me through writing this book. Kamil Kosek has to be mentioned too for his great PowerShell skills, ensuring a high degree of automation.

Last but not least, I thank the great team at Packt Publishing for creating this book, accepting me as an author, and offering great support through the process of writing.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Hyper-V Best Practices

Hyper-V Server and Windows Server 2012 R2 with Hyper-V provide best-in-class virtualization capabilities. Hyper-V with the second generation VMs have security enhancements, less time to boot the OS, speed up OS installations, and automatic activation of VMs. There are many more enhancements in R2, which will give the developers reasons to rejoice. This book will equip you with the real-world Hyper-V configurations and best practices to take full advantage of its virtualization capabilities.

What This Book Covers

Chapter 1, Accelerate Hyper-V Deployment, starts with pointing out the ideal installation method of Hyper-V hosts and then goes on to introduce automatic VM models.

Chapter 2, High Availability Scenarios, covers the creation of a Hyper-V Failover Cluster and configuration of its best practices.

Chapter 3, Backup and Disaster Recovery, starts with the backup methods of Hyper-V hosts and VMs, and continues with Hyper-V Replica as a disaster recovery component in Hyper-V.

Chapter 4, Storage Best Practices, details the different storage scenarios possible with Windows Server 2012 R2 and Hyper-V.

Chapter 5, Network Best Practices, covers the different network virtualization scenarios possible with Windows Server 2012 R2 and Hyper-V.

Chapter 6, Hyper-V Performance Tuning, explains the fundamentals of performance management and tuning in Hyper-V.

Chapter 7, Management with System Center, describes the management of Windows Server with Hyper-V through various System Center components.

Chapter 8, Migration to Hyper-V 2012 R2, covers the migration to Hyper-V 2012 R2 from other hypervisors or earlier Hyper-V versions.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Accelerate Hyper-V Deployment

"Based on my slogan 'Keep It Smart and Simple (K.I.S.S.),' the planning phase is essential for a successful implementation of a Hyper-V environment."

Andreas Baumgarten – MVP System Center Cloud and Datacenter Management

This chapter provides an overview of how to automate the installation of the best practice Hyper-V host and its first running virtual machines. You will learn how to create unattended installations of Hyper-V with minimal effort. All the examples shown in this chapter are proven real-world best practice.

This chapter includes the following topics:

• Planning the Hyper-V host
• Unattended installation of Hyper-V through XML files
• Rapid deployment of virtual machines

Why Hyper-V projects fail

Before you start deploying your first production Hyper-V host, make sure that you have completed a detailed planning phase. I have been called in to many Hyper-V projects to assist in repairing what a "specialist" has implemented. Most of the time, I start to correct the design because the biggest failures happen there, but are only discovered later during implementation. I remember many projects where I was called in to assist with installations and configurations during the implementation phases, because these were the project phases where a real expert was needed.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Accelerate Hyper-V Deployment

Based on experience—this is wrong. Most critical to a successful design phase are two reasons: its rare existence and somebody with technological and organizational experience with Hyper-V. If you don't have this, look out for a Microsoft Partner with a Gold Competency called Management and Virtualization on Microsoft Pinpoint (http://pinpoint.microsoft.com) and take a quick look at the reviews done by customers for successful Hyper-V projects. If you think it's expensive to hire a professional, wait until you hire an amateur. Having an expert in on the design phase is the best way to accelerate your Hyper-V project.

Before you start your first deployment in production, make sure that you have defined the aim of the project and its smart criteria, and have done a thorough analysis of the current state. After this, you should be able to plan the necessary steps to reach the target state, including a pilot phase.

Planning your environment

Besides the organizational skill needed for a successful Hyper-V project, there are some helpful tools that can help in most cases with the technical details. How many hosts will I need for my Hyper-V setup? How many CPUs and how much RAM is needed? What bandwidth is needed on my network? To answer these questions, I commonly use the free Solution Accelerator, Microsoft Assessment and Planning Toolkit (MAP Toolkit) by Microsoft (downloadable at the shortlink, http://bit.ly/1lzt2mJ). The MAP Toolkit is shown in the following screenshot:

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
The easy-to-use MAP Toolkit does a full inventory of your existing environment, including performance counters over time. After running the wizards in your existing infrastructure, you will get an overview, a detailed report of the existing hardware and software infrastructure, and—most importantly—a measure of how these are used in your data center as of today, including used CPU cycles, memory, Storage I/O, and network usage. MAP even includes a planning wizard to plan the hardware requirements of your future Hyper-V hosts based on your current workloads and hardware configurations.

After having a basic understanding of the current usage and future needs of your hardware, it's time to choose the appropriate servers to run Hyper-V and its virtual machines. The good news is that all major server vendors have hardware in their portfolio that performs well for this task, so choose whatever vendor you like; there is just one thing you absolutely need to make sure of. The chosen hardware should be on the Windows Server Catalog and be certified for Windows Server 2012 R2 (shortlink http://bit.ly/1gII6h7). This way, you are making sure that your hardware has undergone extensive testing for Windows Server 2012 R2 with Hyper-V. You will be able to open a support call at Microsoft in case you ever run into problems using this hardware with Hyper-V. If you are going to use an older version of Hyper-V (which you should avoid, but licenses might force you to), select the corresponding host OS on the hardware catalog. Make sure that your host setup includes the necessary adapters to comply with your chosen storage (refer to Chapter 4, Storage Best Practices) and network designs (refer to Chapter 5, Network Best Practices).

Which CPU vendor you choose won't make a huge difference; just make sure that you stick to one, because mixed CPU vendors won't allow you to use live migration between Hyper-V hosts. Be sure that the chosen CPU models have support for server virtualization (Intel VT/AMD-V) and Data Execution Prevention (XD/NX) enabled. I strongly recommend that you use hyperthreading-enabled CPUs for server virtualization with active Second Level Address Translation (SLAT). Both are hardware-accelerated CPU features that add more performance to Hyper-V. For best performance, make sure to buy CPU models from the newest certified Enterprise Server line of the vendor of your choice. Due to the current licensing of Windows Server 2012 R2 Datacenter and several other products, I recommend that you choose CPUs with as many cores per socket as financially reasonable. As of early 2014, the sweet spot was around eight cores per socket for ideal value for money.

To choose the right RAM for your Hyper-V hosts, make sure that it supports Error Checking and Correction (ECC) RAM and choose modules large enough to fit with the amount designed into your hosts. As RAM is very inexpensive these days, you should choose the bigger modules in case of any doubts to ensure growth in future.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
For your storage and networking options, see the corresponding chapters of this book. However, to host the Hyper-V management partition, I strongly recommend that you use two local SSDs or HDDs in Raid1 and not share the disks with virtual machines or other data. I have experienced the best results with these local hard drives and have found some problems with remote boot scenarios due to the higher complexity of boot-from-SAN setups—which is also a possible, and supported, but not preferred scenario. You don't need high-performance disks for the OS; all I/O performance should be added to the VM storage.

Another important topic is to choose fewer bigger boxes over many small Hyper-V hosts. This enables more efficient management. However, while needing a failover resource in a cluster, a Hyper-V cluster should consist of at least three nodes; otherwise 50 percent of your hardware is reserved for failover scenarios.

Refer to Chapter 6, Hyper-V Performance Tuning, it includes advanced hardware sizing guidelines for performance tuning.

Preparing your host systems

Many "Prepare Your System" chapters start with telling you to update all your hardware and software components to the latest releases. This chapter doesn't make an exception to this rule. In no other technical area have I seen so many successful fixed environments due to firmware and driver updates. Windows Server with Hyper-V has undergone a very rapid development cycle with many releases in a short timeframe. Most hardware vendors released firmware and drivers with greatly shortened testing periods and were forced to release several updates due to firmware and driver updates to their products. Before you start setting up your Hyper-V host, update BIOS, RAID Controller, and the Network Interface Card (NIC) firmware to their latest release. Use the home page of the server-vendor, not the vendor of the individual components, for reference to the latest certified releases. Use only downloads from the individual components' vendor if you see those problems you encounter fixed by the corresponding release notes.

Other than this, you only need your Hyper-V installation media, the Windows 8.1 ADK (shortlink http://bit.ly/1jAfero), and a USB drive to prepare for rapid Hyper-V installations. Download either the full version of Windows Server 2012 R2 with Hyper-V from your Volume Licence Portal or the 180-day Evaluation version of Hyper-V (shortlink http://bit.ly/1hIRE XL). In fact, it does not make any difference whether you use the Evaluation edition or the full version media—they are interchangeable—the only difference will be made by the product key you enter. All Hyper-V features are also supported by the free editions of Hyper-V Server 2012 R2 (shortlink http://bit.ly/1oL1lbM); all the screenshots and configurations you see in this book are created using the full version of Windows Server 2012 R2 with Hyper-V and could vary slightly from the free edition. Hyper-V is very easy to install.
To familiarize yourself with Hyper-V, just insert the installation media, select it as the boot device, and click through the various options in the setup wizard. If this will be the only Hyper-V host you will ever install, this will be a great installation experience. Most of the time, you will not stick to just one host, and to speed up things, we will mainly use unattended installations of the Hyper-V hosts from now on. The unattended setup uses configurations saved in a precreated unattended.xml file, which can be either slipstreamed into the installation media or saved on a USB drive so that it's available to the host during installation. This enables a standardized and very rapid Hyper-V deployment with a onetime preparation.

Creating unattended installation files

To create an unattended.xml file, you can either start from scratch with a simple text editor or use a GUI. To leverage the second option, start the setup of the Windows Assessment and Deployment Kit (ADK) you downloaded earlier. At the setup prompt, select only Deployment Tools, as shown in the following screenshot. After the completion of the installation, start Windows System Image Manager from the Start screen:

![Windows Assessment and Deployment Kit for Windows 8.1](image)

Select the features you want to change

Click a feature name for more information.

- Deployment Tools
- Windows Preinstallation Environment (Windows PE)
- User State Migration Tool (USMT)
- Volume Activation Management Tool (VAMT)
- Windows Performance Toolkit
- Windows Assessment Services
- Microsoft SQL Server 2012 Express

Deployment Tools

Tools to customize and manage Windows images and to automate installation.

Includes:

- Deployment Image Servicing and Management (DISM) tool. To use DISM cmdlets, PowerShell 3.0 must also be installed.
- OEM Activation 3.0 Tool.
- Windows System Image Manager (SIM).
- osCmdLine, BCDBoot, DISMapi, WIMCAP, and other tools and interfaces.

Estimated disk space required: 0 bytes
Disk space available: 64.5 GB

Windows ADK 8.1

For More Information:

www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Accelerate Hyper-V Deployment

After the tool is fully loaded, select the File menu, open the Select an Image wizard, and browse to the Install.wim file or your installation media in the source's subdirectory. Select the Windows Server 2012 R2 SERVERSTANDARD edition for your first unattended installation file and allow the creation of a new catalog file. If you receive a warning message stating that you are unable to write the catalog file, open Windows Explorer, navigate to the Install.wim file, open its properties, and uncheck the read only checkbox. If you have your installation media sitting on a physical read-only media, copy Install.wim to a local hard drive first. Select the Server Standard Edition with GUI:

Select Windows Edition

After the catalog creation is completed, select the File menu again, create a New Answer File, and save it as unattended.xml to a place of your choice.

Windows System Image Manager will then create the basic XML structure of your unattended file, as shown in the following screenshot:

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Opening this XML file in Internet Explorer will show you the actual file contents. Every Windows Server 2012 R2 setup will check for an existing unattended.xml file at the start of every available drive letter, but will only work if the XML structure is correct. We will now continue to fill this `unattended.xml` file with contents specific to the Hyper-V setup to allow a Zero-Touch installation of your Hyper-V hosts.

### Adding basic components

Start by adding the most basic components by expanding the **Components** tree under the **Windows Image** section in the left-hand side corner of the tool. Let's now add language and locale information:

1. First, add the `amd64_Microsoft-Windows-International-Core-WinPE` components to `Pass1` and fill it with the basic language options. The generated XML part with all the mandatory parameters will look like the following code:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <SetupUILanguage>
    <UILanguage>en-US</UILanguage>
  </SetupUILanguage>
  <InputLocale>en-US</InputLocale>
  <UILanguage>en-US</UILanguage>
  <SystemLocale>en-US</SystemLocale>
  <UserLocale>en-US</UserLocale>
</component>
```

If you prefer language settings other than US English, make sure that the language components are included in the installation media and refer to the correct locale IDs, which can be found on Microsoft MSDN (shortlink [http://bit.ly/1gMNu2B](http://bit.ly/1gMNu2B)).

2. Next, add `amd64_Microsoft-Windows-Setup_neutral` to `Pass1` to configure some basic OS configurations such as Disk Layout. A generated sample XML part for a BIOS-based system is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

For More Information:

This configuration will make sure that there are clean partitions that follow Microsoft's default deployment model. The small partition at the start of the disk is created to support Bitlocker. Microsoft's full disk encryption can be used with Hyper-V hosts and can also be activated later. The use of Bitlocker is only recommended in high-security environments.
3. If your host does not have BIOS anymore and uses an UEFI-based setup routine, the XML file will be edited to include the following code as well:

```xml
  <Disk wcm:action="add">
    <CreatePartitions>
      <CreatePartition wcm:action="add">
        <Order>2</Order>
        <Size>100</Size>
        <Type>EFI</Type>
      </CreatePartition>
      <CreatePartition wcm:action="add">
        <Order>3</Order>
        <Extend>false</Extend>
        <Type>MSR</Type>
        <Size>128</Size>
      </CreatePartition>
      <CreatePartition wcm:action="add">
        <Order>4</Order>
        <Extend>true</Extend>
        <Type>Primary</Type>
      </CreatePartition>
      <CreatePartition wcm:action="add">
        <Order>1</Order>
        <Size>350</Size>
        <Type>Primary</Type>
      </CreatePartition>
    </CreatePartitions>
    <ModifyPartitions>
      <ModifyPartition wcm:action="add">
        <Active>false</Active>
        <Format>NTFS</Format>
        <Label>Bitlocker</Label>
        <Order>1</Order>
        <PartitionID>1</PartitionID>
      </ModifyPartition>
    </ModifyPartitions>
  </Disk></DiskConfiguration>
</component>
```

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Which edition to install

In the earlier versions of Windows Server with Hyper-V, there were huge differences among the different available editions. Some features and hardware requirements were only available on Enterprise and Datacenter editions of Windows Server. Almost all the features and hardware specifications are available even with the Standard edition of Windows Server 2012 R2, so there is no need to use an Enterprise edition (which was dropped by Microsoft earlier). The main difference between the Standard and Datacenter editions relies on the virtualization rights. Each Windows Server Standard edition allows you to run two guest Operating System Environments (OSEs) with Windows Server editions, and a Datacenter edition allows you to run an unlimited number of Windows Server VMs on this particular licensed Hyper-V host. There is only one technical difference between the two editions: on a Datacenter edition, all Windows Server guest VMs will be automatically activated, when provided with a corresponding key during setup. There is no need for a MAK or KMS-based OS activation anymore.

Basically, it's not very important which edition you choose for the host install, so stick to Windows Server 2012 R2 Standard. If you want to leverage Automatic Virtual Machine Activation (AVMA), install a Datacenter edition on the host. It is easy to upgrade a Standard edition later to a Datacenter edition, but there is no downgrade option.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
If you are not sure which edition you are using, open a PowerShell window with administrative privileges and run the following command:

```
get-windowsedition -online
```

To find out which editions are available for upgrade, run the following command:

```
Get-WindowsEdition -online -target
```

Finally, to upgrade to the target edition, run the following command:

```
Set-WindowsEdition -online -edition ServerDatacenter
```

While it's suitable to install a Datacenter edition on a Hyper-V host, you should never do this inside a virtual machine. Installing Standard editions into virtual machines always gives you an easy way to move this VM at a later time to a non-Datacenter Hyper-V host.

The next step to build our unattended installation is to set up the installation target and edition. Navigate to the `ImageInstall` string under the `Microsoft-Windows-Setup` node and add the following code:

```
<ImageInstall><OSImage><InstallFrom><MetaData wcm:action="add"><Key>/Image/Name</Key><Value>Windows Server 2012 R2 SERVERSTANDARD</Value></MetaData></InstallFrom><InstallTo><DiskID>0</DiskID><PartitionID>2</PartitionID></InstallTo></OSImage></ImageInstall>
```

If you have chosen the UEFI-based setup, choose `PartitionID 4` according to your disk setup. This will make sure that you install the Standard edition of Windows Server 2012 R2 to the correct partition.

As the last step in Pass1, we will fill out the `UserData` tree under the `Microsoft-Windows-Setup` node and edit the following code:

```
<UserData><ProductKey><WillShowUI>OnError</WillShowUI></ProductKey><AcceptEula>true</AcceptEula><FullName>YourName</FullName><Organization>YourOrg</Organization></UserData>
```

Fill in `Name` and `Org Data` with anything you like; however, these fields are mandatory. The `product key` field is optional. If you intend to use a 180-day trial version of Windows Server or are leveraging KMS Server activation capabilities, do not enter a product key. If you are using MAK-based OS activations, enter your product key. You can also install a MAK product key at a later time by opening a PowerShell window with administrative privileges and running the following command:

```
slmgr -upk (this uninstalls the current product key) and
slmgr -ipk <key> (including dashes)
```

For More Information:

www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
To GUI or not to GUI

After adding the basic parameters, it’s now time to add some comfort to our Zero-Touch installation.

In Windows System Image Manager, add amd64_Microsoft-Windows-Shell-Setup_neutral to Pass4 and Pass7.

Edit the XML file to set your time zone settings (run `tzutil /l` in a Shell to get a list of all the valid time zones) and your local administrator password. Don’t worry about entering a password into Windows System Image Manager; it will encrypt the password while saving the file. The following code shows how to set the regional and user information:

```xml
<settings pass="specialize"><component language="neutral"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
versionScope="nonSxS" publicKeyToken="31bf3856ad364e35"
processorArchitecture="amd64" name="Microsoft-Windows-Shell-Setup"><TimeZone>W. Europe Standard Time</TimeZone></component>
</settings>
<settings pass="oobeSystem"><component language="neutral"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
versionScope="nonSxS" publicKeyToken="31bf3856ad364e35"
processorArchitecture="amd64" name="Microsoft-Windows-Shell-Setup">
<UserAccounts><AdministratorPassword><Value>UABAAHMAcwB3ADAAcgBkAABgAABMAAbgBpAAMAcBByAGEAdABvAHIAbhAAMIcwB3AG8AcgBkAA==</Value><PlainText>false</PlainText></AdministratorPassword></UserAccounts></component>
</settings>

To allow a rapid deployment of hosts, I have not entered a computer name at this stage, so the setup will generate a random computer name for each node installed. If you want to enter a computer name, add the following code to your XML-specialized section:

```xml
<ComputerName>Hyper-V01</ComputerName>
```

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.

For More Information:

www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Another optional feature was selected right at the beginning of our XML creation: the GUI. By selecting the Windows Server Standard edition and not the Standard Core edition, we have included the complete GUI of Windows Server in our setup. Unlike previous versions of Windows Server with Hyper-V, the GUI is now a feature that can be activated and deactivated at a later stage. Please note that the GUI is not available on the free Hyper-V Server 2012 R2. The full GUI installation process offers the same great user experience we know for many versions of Windows Server and Windows Client operating systems, but Server Core is the recommended installation method by Microsoft for Windows Server 2012 R2 and Hyper-V. The Core installation option offers a reduced attack surface with less patching efforts and fewer reboots. It even comes with a smaller resource footprint than its Full GUI equivalent. However, offering only a PowerShell Window as the single point of local administration discouraged many system administrators in the past, so Core setups aren't found often. Don't forget that all administrative APIs are active on a Core Server, so you can connect with your MMC consoles from other servers or clients without the need to use the PowerShell modules. With Windows Server 2012 R2, you have even more choices; Microsoft added the MinShell installation mode where, in addition to PowerShell, all administrative consoles, known from the Remote Server Administrations Tools (RSAT), are available locally, but security critical elements such as Internet Explorer still aren't presented locally. Following customer feedback, MinShell is getting more attention than the older Core setup. Both Core and MinShell do offer another huge advantage: they don’t allow the installation of programs and configurations that require a GUI, though these systems are typically very clean in terms of software onboard, adding to a seamless operation in a production environment. The following best practice model was established by our customers to manage the GUI:

1. Install the Server with Full GUI. Make yourself familiar with the operating system and its roles.
2. After finishing the initial configuration, uninstall the GUI and run the server in the MinShell or Core mode.
3. To switch from Full Server GUI to MinShell later on, open a PowerShell Window with administrative privileges and run

   Uninstall-WindowsFeature Server-Gui-Shell –Restart

   for MinShell and Get-WindowsFeature *gui* |
   Uninstall-WindowsFeature –Restart for Server Core.

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
4. To reinstall MinShell, run the following command:

   `Install-WindowsFeature Server-Gui-Mgmt-Infra -Restart`

5. To reinstall Full Server GUI, run the following command:

   `Install-WindowsFeature Server-Gui-Mgmt-Infra, Server-Gui-Shell -Restart`

In every case, a reboot of the host will occur. Look at the Windows Server Core PowerShell shown in the following screenshot:

![Server Core without GUI elements](image)

**Hyper-V hosts in Active Directory domains**

The basic operating system setup will now already be based on a Zero-Touch installation, but we want to achieve more than this and will include some additional options.

For More Information:

www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
Add the `amd64_Microsoft-Windows-TerminalServices-LocalSessionManager` component to Pass4 and configure it to enable Remote Desktop Access to the server:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <fDenyTSConnections>false</fDenyTSConnections>
</component>
```

To reach the Server via RDP via its designated IP address, we will also set the basic network settings. Keep in mind that based on your converged network setup for Hyper-V, these might be overwritten at a later step (Chapter 5, Network Best Practices).

Add the `amd64_Microsoft-Windows-TCPIP` component to Pass4 and configure a static IP Address—in this case, based on the name of the interface. This is also possible using the MAC address. Configure the network as shown in the following code:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Interfaces>
    <Interface wcm:action="add">
      <Ipv4Settings>
        <DhcpEnabled>false</DhcpEnabled>
        <Metric>10</Metric>
        <RouterDiscoveryEnabled>true</RouterDiscoveryEnabled>
      </Ipv4Settings>
      <UnicastIpAddresses>
        <IpAddress wcm:action="add" wcm:keyValue="1">192.168.1.41/24</IpAddress>
      </UnicastIpAddresses>
      <Identifier>Local Area Connection</Identifier>
    </Interface>
  </Interfaces>
</component>
```

For More Information:

Whether Hyper-V hosts should be added to an Active Directory domain is a topic that is often discussed. Having seen a lot of Hyper-V environments, either domain-joined or workgroup-joined, my answer to this is a strong yes. Windows Server 2012 R2 Servers can boot up even clusters when domain-joined without an Active Directory domain controller available, so this chicken-or-egg problem from earlier Hyper-V versions is not a problem any more. Hyper-V will run without an Active Directory domain; however, very basic capabilities such as live migration won't be available on workgroup environments. Huge Hyper-V installations or high-security companies even leverage their own management domain to place their Hyper-V hosts into an Active Directory domain.

There is little security consideration standing against a huge management benefit, through credential management, group policies, and so on, so you should domain-join all Hyper-V hosts to your existing Active Directory domain. If your Hyper-V hosts will be placed in high-security environments, join them to a dedicated management domain (within a separated Active Directory forest) and not to your production domain.

Add the `amd64_Microsoft-Windows-UnattendedJoin` component to `Pass4` and configure it to join an existing Active Directory domain:

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <Identification>
    <Credentials>
      <Domain>Elanity.local</Domain>
      <Password>Hannover96</Password>
      <Username>joindomain</Username>
    </Credentials>
    <JoinDomain>Elanity.de</JoinDomain>
    <MachineObjectOU>OU=Hyper-V,DC=Elanity,DC=local</MachineObjectOU>
  </Identification>
</component>
```

A typical configuration that is seen in this step is the disabling of the Windows Firewall. In my opinion, this is a bad practice. The Windows Firewall is a great layer of security and should be configured to your needs, but not disabled. For a central Firewall configuration, we'll use Group Policy settings, so we don't need to include any configuration in our `unattended.xml`.

For More Information:

Activating Hyper-V features

After our operating system is prepared to host Hyper-V, it's time to activate the Hyper-V components. Add the following product packages and their roles and features to your unattended.xml file:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<servicing>
  <package action="configure">
    <assemblyIdentity language="" publicKeyToken="31bf3856ad364e35" processorArchitecture="amd64" name="Microsoft-Windows-ServerStandardEdition" version="6.3.9600.16384" />
    <selection name="Microsoft-Hyper-V-Common-Drivers-Package" state="true" />
    <selection name="Microsoft-Hyper-V-Guest-Integration-Drivers-Package" state="true" />
    <selection name="Microsoft-Hyper-V-Server-Drivers-Package" state="true" />
    <selection name="Microsoft-Hyper-V-ServerEdition-Package" state="true" />
  </package>
  <package action="configure">
    <assemblyIdentity language="" publicKeyToken="31bf3856ad364e35" processorArchitecture="amd64" name="Microsoft-Windows-ServerCore-Package" version="6.3.9600.16384" />
    <selection name="Microsoft-Hyper-V" state="true" />
    <selection name="Microsoft-Hyper-V-Offline" state="true" />
    <selection name="Microsoft-Hyper-V-Online" state="true" />
    <selection name="VmHostAgent" state="true" />
    <selection name="AdminUI" state="true" />
    <selection name="ServerManager-Core-RSAT" state="true" />
    <selection name="ServerManager-Core-RSAT-Feature-Tools" state="true" />
    <selection name="ServerManager-Core-RSAT-Role-Tools" state="true" />
  </package>
</servicing>
```

For More Information:

www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
After adding these Hyper-V components, the creation of our `unattended.xml` file is completed. You can download the complete sample XML file (http://bit.ly/1xBIGb2). Place the file in the root folder on the USB drive and boot the Server system from your installation media. You will now experience a fully Zero-Touch Hyper-V installation. In Chapter 2, High Availability Scenarios, you will learn how to advance this even further into a Zero-Touch cluster installation.

![Unattended.XML file for automatic Hyper-V setup](image)

**Post-installation tasks**

Be sure to remove the USB drive with the unattended setup file prior to moving the host to production. A host reboot could otherwise force a reinstallation, including a wipe of all hard drives, due to the trigger of another unattended installation.

Run Windows Update to make sure that you have installed all the available updates. Are there any Windows updates you should not install on Hyper-V hosts? Yes, drivers should not be installed over a Windows Update unless support tells you to do so. However, besides this, install every available Windows update in all of your Hyper-V hosts. Check out the update list for Hyper-V to check whether there are any Hotfixes available that are recommended for installation but not yet available through Windows Update (shortlink http://bit.ly/1kx0yYS).

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
The Hyper-V role is already enabled, and we are ready to create virtual machines. To ensure network connectivity and safe operations of our VMs, we will configure some additional parameters after the installation.

First of all, we need some basic network connectivity for our virtual machines. If you have a second Network Interface Card (NIC) available in your host, run the following command in an elevated PowerShell session:

```
New-VMSwitch -Name external -NetAdapterName "Local Area Connection 2"
```

If you have only one NIC, run the following command:

```
New-VMSwitch -Name external -NetAdapterName "Local Area Connection" -AllowManagementOS $true
```

Now, your virtual machines can use an external Hyper-V switch named "external" to communicate over the network.

Ever wondered about the many errors your RDP-mapped printer can create on a Hyper-V host? I could not believe this for a long time, but recently, I have seen a blue-screened Hyper-V Server due to improper printing drivers. Do you need to print from a Hyper-V host? Absolutely not! So, make sure that you disable RDP Printer Mapping through a Group Policy (or Local Policy).

Navigate to **Computer Configuration** | **Policies** | **Administrative Templates** | **Windows Components** | **Remote Desktop Services** | **Remote Desktop Session Host** | **Printer Redirection** | **Do not allow client printer redirection** | **Enable** in a Group Policy.

Hyper-V uses some default paths to store virtual machine configuration and its hard disks. I find this very interesting but definitely not suitable for a production environment. Make sure that you change the default paths, if possible to a non-system drive, by running the following commands in an elevated PowerShell window:

```
Set-VMHOST -computername localhost -virtualharddiskpath 'D:\VMS'
Set-VMHOST -computername localhost -virtualmachinepath 'D:\VMS'
```

I have not seen any issues with placing VM configuration files and virtual hard disks into the same folder structure. You have everything your VM configuration depends on in one place.

Another important post-installation task is to follow the rule: do not install other roles on Hyper-V hosts. A Hyper-V host is a Hyper-V host and nothing else. Move all the other services into virtual machines that run on the Hyper-V host.

For More Information:

Moreover, also keep the following points in mind:

- Do not install any features other than Failover Clustering and **Multipath I/O (MPIO)** on a Hyper-V host
  
  There are exceptions in an SMB3 scenario where you also want to install **Datacenter Bridging (DCB)** and SMB bandwidth limits

- Limit software installations to an absolute minimum, that is, backup and monitoring agents

## Antivirus on a Hyper-V host

Another great topic for discussion is whether you should install an antivirus client on a Hyper-V host or not. Many companies have compliance rules stating that on every Server or every Windows machine, an AV client needs to be installed. If there is a rule like this in place, follow it and install an AV agent on your Hyper-V hosts. Make sure that you also implement the long list of files, which contain all the Hyper-V configuration files and virtual machine data, you have to exclude from your scans.

I have seen antivirus engines on Hyper-V hosts doing bad things such as breaking a virtual hard disk, deleting an essential system file, or just producing a very intense amount of storage I/Os. Excluding all relevant files and folders regarding Hyper-V and its VMs, there is nothing left worth scanning on a Hyper-V host. If you are not bound by a compliance policy, I highly recommend that you do not install antivirus products on Hyper-V.

There are some approaches for Hyper-V-aware antivirus products; however, I have not seen one flawless working solution as of today, so you should protect your VMs from malware from inside the VM by installing your AV agents into the virtual machines.

## Setting the Pagefile

One of the most frequent configuration tips around Hyper-V hosts is to manually configure the Pagefile. The values described are sometimes quite creative.

After doing many tests with Hyper-V hosts with all different kinds of RAM configurations and deep technology-oriented exchanges with Microsoft Product Teams, including the Hyper-V Product Team itself, on how Pagefile management is working in Windows Server 2012 R2, there is only one recommendation I have today: leave it alone.
The Windows Pagefile is by default managed by Windows. If you have followed all other best practices described up to this point and, most importantly, you did not install other services on the Hyper-V host itself (management OS), you are all set. There is no way you can reach the same or even a better efficiency in PageFile management by manually altering this automatic configuration. I have not seen a single Hyper-V installation on Windows Server 2012 R2 as of now that had problems with automatic Pagefile management.

Again, this only affects the Hyper-V host and not the Pagefile configuration of the virtual machines.

There are some other valuable post-installation tasks for performance management in Chapter 6, Hyper-V Performance Tuning. You can manage the Pagefile as shown in the following screenshot:

![Pagefile configuration](image)

## Creating virtual machines

You are all set, and it's time to create some virtual machines. To do a rapid deployment of virtual machines, we will rely on PowerShell.

Creating a new virtual machine with PowerShell is easy; just open an elevated PowerShell prompt, and run the following command:

```
New-VM
```
Without any additional parameters, this will create a new virtual machine with the default parameters. To create a new Generation 2 VM, run the following command:

```
New-VM -Generation 2
```

To create a new virtual machine with a specified name, a custom path to store the VM files, and a memory configuration, run the following command:

```
New-VM -Name VM01 -Path C:\VM01 -MemoryStartupBytes 1024MB
```

Your newly created virtual machine doesn't have a hard disk yet. Create a new VHDX file by running the following command:

```
New-VHD -Path C:\vms\vm01\vm01_c.vhdx -SizeBytes 60GB -Dynamic
```

The new VHD cmdlet

The created VHDX is not yet attached to a virtual machine. Do this by running the following command:

```
Add-VMHardDiskDrive -VMName VM01 -Path C:\vms\vm01\vm01_c.vhdx
```

To add a network adapter to our virtual machine, run the following command:

```
Add-VMNetworkAdapter -vmname "VM01" -switchname "external"
```

Then, start the VM by running the following command:

```
Start-VM -Name VM01
```

For More Information:

You will recognize that the virtual machine now has all the basic hardware parameters but fails to boot due to a missing operating system. There are multiple ways to create an operating system for a standard VM. The most granular way to achieve this is using Virtual Machine Manager templates (see Chapter 7, Management with System Center, for details), but there are great capabilities already included in Windows Server 2012 R2. The approach that is seen most often is to manually install the first virtual machine and include everything you want in each of your virtual machines, such as operating system, updates, and backup agents. Then, sysprep the virtual machine by executing `sysprep.exe` present at `C:\Windows\System32\sysprep\` with the `Generalize` and `OOBE` options and shut down the virtual machine. Copy it to a template folder and mark this as read only. With Windows Server 2012 R2, you can even copy and export running virtual machines.

If you need a new virtual machine, just copy the Template folder, rename it to your machine name, and a preinstalled operating system with all your previous created configurations are still available. If you even want to save this first manual installation, there is a free tool for you. In Microsoft’s TechNet Gallery, you will find a tool called Convert-WindowsImage that creates a fully functional VHDX template just from the operating system’s installation media for you (download it from the shortlink at `http://bit.ly/1odCE1X`):

For More Information:
www.packtpub.com/virtualization-and-cloud/hyper-v-best-practices
It is even possible to run your `unattended.xml` file created with Windows System Image Manager directly through this tool, out of the box.

If you don't like patching all your images and archived VMs manually, you can use a solution to update these VHD/VHDx files offline with `Apply-WindowsUpdate.ps1` — just another Gem from the TechNet Gallery (download this from the shortlink at `http://bit.ly/1o4sczI`).

As you have seen in this chapter, I have mainly used Generation 2 VMs. If your guest operating systems are Windows Server 2012 and higher, this should be your default option. Generation 2 VMs allow faster booting, better stability, and smaller attack surface through a greatly reduced set of legacy hardware.

**Summary**

With the tools from this chapter and the configuration files you have already created up to now, you will be able to deploy new Hyper-V hosts and VMs faster and in a more reliable way than ever before. Besides this, you learned valuable best practices to plan and configure your single Hyper-V host. Continue with *Chapter 2, High Availability Scenarios*, to create high-available solutions based on your current setup to leverage additional capabilities of Hyper-V and virtualization.
Where to buy this book

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.