

# All you need to know about Microsoft Windows Server 2016 Virtualization

## **Clint Wyckoff**

Global Technical Evangelist, Veeam Software  
Microsoft Cloud and Datacenter Management MVP,  
VMware vExpert, MCP, VMCE

**AVAILABILITY**  
for the Always-On Enterprise™

# Contents

<b>Introduction</b> .....	<b>5</b>
<b>History and Evolution of Windows Server Virtualization</b> .....	<b>6</b>
Windows Virtual PC & Microsoft Virtual Server .....	6
Windows Hyper-V: Server 2008 and 2008 R2 .....	7
Windows Hyper-V: Server 2012 and 2012 R2 .....	8
Summary .....	9
<b>What's New in Windows Server 2016 Virtualization</b> .....	<b>9</b>
<b>Nano Server</b> .....	<b>10</b>
What Does Nano Set Out to Fix? .....	11
Summary .....	12
<b>Windows Containers</b> .....	<b>12</b>
Windows Containers Architecture .....	13
Applications within Containers .....	15
Container Deployment and Image Creation .....	16
Container Management with PowerShell Direct .....	19
Docker and Windows Server 2016 Containers .....	20
Docker Hub .....	20
Docker and Windows Server 2016 Containers .....	23
Docker Container and the Docker CLI .....	25
Hyper-V Container .....	25
Hyper-V Container Deployment Example .....	26
Summary .....	27

<b>Top New Features of Windows Server 2016 Hyper-V .....</b>	<b>28</b>
Production Checkpoints .....	28
PowerShell Direct .....	32
Hyper-V Manager Enhancements .....	34
ReFS Fixed VHD Creation .....	36
Hyper-V Integration Services .....	37
VM Configuration File Format .....	38
Hypervisor Power Management — Connected Standby.....	40
RemoteFX vGPU and VDI.....	40
<b>Security Enhancements in Windows Server 2016 Virtualization .....</b>	<b>41</b>
Server Security Concepts.....	42
Virtual Secure Mode .....	42
Shielded VMs and Guarded Fabric Hosts.....	43
Summary .....	45
<b>Performance Isolation Techniques .....</b>	<b>46</b>
Storage Quality of Service (QoS).....	46
Single Instance Policy.....	47
Multi-Instance Policies .....	47
Storage QoS Management.....	48
Host Resource Protection .....	49
<b>Hyper-V Availability.....</b>	<b>50</b>
VM Compute and Storage Resiliency .....	50
Shared VHDX .....	51
Hyper-V Replica.....	52
Memory Management.....	53
Networking Enhancements .....	54

<b>Upgrading the Environment to Hyper-V 2016 .....</b>	<b>55</b>
Upgrading the VM Hardware Version .....	56
<b>Hyper-V Supports Linux .....</b>	<b>56</b>
<b>Appendix A.....</b>	<b>58</b>
Licensing in Windows Server 2016 .....	58
Installing Windows Server 2016 .....	60
Create New VM Using PowerShell .....	70
About the Author.....	71
External Reviewers.....	71
About Veeam Software .....	71

# Introduction

Windows Server 2016 is set to become generally available at some point during 2016. At the time of writing this eBook, Microsoft has yet to provide a definitive date, however, Microsoft has been releasing technical preview versions of their upcoming releases to allow IT Professionals the opportunity to learn the new technology as well as provide feedback. Technical previews offer a fantastic vehicle for teams to begin to test and learn the new technologies that are due to release in the next version of Windows Server. As organizations are moving at an extremely fast pace and continuing to virtualize more mission critical applications these technical previews become an invaluable asset.

The topic we will be discussing in this eBook is Windows Server 2016 Virtualization – also known as Hyper-V 2016. Components within Hyper-V are changed and/or added with each release Microsoft provides. Knowing this is important and key to understanding the increasing functionality and usability through documents such as this.

Many of the new features and functionality do require some basic usage of PowerShell. Throughout this eBook you will find them documented as examples allowing IT Professionals to leverage Hyper-V PowerShell scripts in their own environments. The goal of this eBook is to arm you with the necessary tools to successfully test and eventually manage a Windows Server 2016 Hyper-V environment.

# History and Evolution of Windows Server Virtualization

Before diving into what is new and upcoming within Windows Server 2016 Virtualization, let's start by giving you some history on Microsoft's hypervisor platform and how it has evolved over the years.

## Windows Virtual PC & Microsoft Virtual Server

Originally developed by Connectix (Connectix Virtual PC) and acquired by Microsoft, Virtual PC was designed in the late 1990s and initially released within Microsoft in February, 2003 with the intent of creating virtual machines on x86 desktop hardware.

Virtual PC for Windows provided Windows desktop customers with an additional tool for migrating to Windows XP or to Windows 2000 Professional, support for legacy applications, and enabled a range of other uses for application development, call centers, technical support, education and training.

Virtual Server addressed customer demand for an application migration solution based on virtualization and supported by Microsoft. In addition, it provided significant cost efficiencies by consolidating multiple Windows NT 4.0 servers and their applications onto a single Windows Server system.

Microsoft Virtual Server was designed as a web-based interface typically deployed through Internet Information Services (IIS). This web-based interface was the mechanism that IT used to manage virtual machines. Both Virtual PC and Virtual Server are called Type-2 Hypervisors. These virtualization platforms contained several limitations, and both have been deprecated and replaced by Hyper-V.

### Hypervisor Design: Two approaches

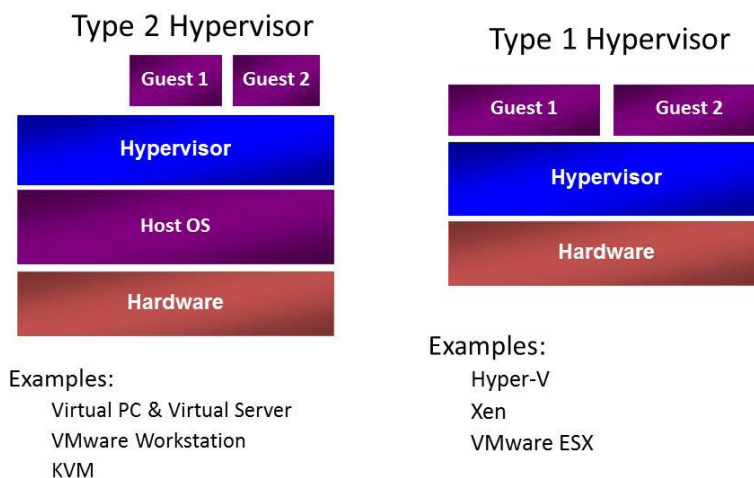


Figure 1: Courtesy Microsoft - Type 1 vs. Type 2 Hypervisors

There is often a lot of confusion around Type-1 and Type-2 hypervisors. The table below provides a detailed explanation:

Hypervisor Type	Description
Type-1, native or bare-metal hypervisors	These hypervisors run directly on the host's hardware to control the hardware and to manage guest operating systems. For this reason, they are sometimes called bare metal hypervisors. A guest operating system runs as a process on the host. The first hypervisors, which IBM developed in the 1960s, were native hypervisors.[2] These included the test software SIMMON and the CP/CMS operating system (the predecessor of IBM's z/VM). Modern equivalents include Microsoft Hyper-V, Oracle VM Server for SPARC, Oracle VM Server for x86, the Citrix XenServer, and VMware ESX/ESXi.
Type-2 or hosted hypervisors	These hypervisors run on a conventional operating system just as other computer programs do. Type-2 Hypervisors abstract guest operating systems from the host operating system. Microsoft Virtual PC, Microsoft Virtual Server, VMware Workstation, VMware Player, VirtualBox and QEMU are examples of Type-2 Hypervisors.

## Windows Hyper-V: Server 2008 and 2008 R2

Initially released within Server 2008, Hyper-V is Microsoft's first Type-1 Hypervisor. Microsoft has incrementally added new features and functionality to Hyper-V with each version of Windows Server. Unlike previous iterations of Microsoft hypervisors, Hyper-V creates a partition; an isolated computing environment from the parent Windows Server Operating System and the guest virtual machines (VMs). The underlying guest VMs have their hardware components virtualized, and depending on the VM configuration, each guest may only have a subset of the parent's processing and memory allocated. The guest VM hard disks are emulated as files that are contained in the Virtual Hard Disk (VHD) file format. These individual VHD files contain the guest operating system, applications and data.

Server 2008 R2 introduced new capabilities including Live Migration with Cluster Shared Volumes (CSV). Building Live Migration into Hyper-V provided the ability to move VMs' compute ownership from one node of a failover-cluster to another without any downtime or service interruption. Previously in Server 2008, the only option was to Quick Migrate, which required the VM to be placed into a saved state prior to moving the contents of the guest VM memory to another host.

In Windows Server 2008 and 2008 R2, Hyper-V was deployed as a role service inside of the Standard, Enterprise and Datacenter Editions. Choosing the correct version depended on how many VMs were required within the environment or if it required high availability. The high availability of Hyper-V is provided by Windows Failover Clustering (only available in Enterprise and Datacenter Editions).

In Windows Server 2008 and 2008 R2, Hyper-V was also deployable as a standalone variant called Hyper-V Server. This version was extremely popular with Managed Service Providers (MSP) as it did not require any underlying licenses of Windows Server to run it. So if a HSP only ran instances of Linux guest VMs, it would be free.

Edition	Features	Scalability	Virtual Operating Systems License
Standard	Limited	Limited	1 Windows Operating System
Enterprise	Unlimited	Unlimited	4 Windows Operating System
Datacenter	Unlimited	Unlimited	Unlimited Windows Operating System
Hyper-V Server 2008 and 2008 R2	Limited	Limited	0 Windows Operating System

## Windows Hyper-V: Server 2012 and 2012 R2

Windows Server 2012 and 2012 R2 brought several key enhancements and technologies to Hyper-V. For the first time Hyper-V could now be deployed and run in a desktop environment. Windows 8.1 allowed the Hyper-V role to be enabled, which allowed great flexibility and provided a fantastic way for users running labs to learn new technologies.

Hyper-V on Server 2012 and 2012 R2 introduced support for large-scale virtual machines. The new VHDX file format supports virtual hard disks of up to 64 TB in size. Guest VMs could now have 64 virtual processors and 1 TB of virtual RAM. Hyper-V hosts could contain 320 logical processors, 4 TB of memory and run 1024 VMs all on a single host. Also new in Server 2012 was the concept of Storage Migration, moving virtual hard disks that are being used by individual VMs from one physical storage device to another while the VM stays running.

Many new enhancements to storage were included in Windows Server 2012 and 2012 R2. These are listed below:

- SMB Multichannel and SMB Direct, when used with Remote Direct Memory Access network adapters.
  - RDMA supported network cards enhanced Live Migration performance by using fewer CPU cycles, providing low latency and increasing throughput by allowing the adapters to coordinate the transfer of large data chunks at near line speed.
- SMB shares, when used with Scale Out File Services role in Windows Server 2012 or 2012 R2, allows for an inexpensive way for IT Professionals to get the many benefits of shared storage for Hyper-V guest VMs without the expensive costs of an Enterprise SAN.

Within Windows Server 2012 and 2012 R2, Hyper-V is deployable in 2 variations: Standard and Datacenter. Both installations provide the exact same features and functionality. The only difference is the amount of Virtual Operating System Environment (VOSE) that are included with the single license and Datacenter supports Automatic Virtual Machine Activation on the host.



Edition	Features	Scalability	Virtual Operating Systems
Standard	Unlimited	Unlimited	2 Windows OS
Enterprise	Unlimited	Unlimited	Unlimited Windows OS
Hyper-V Server 2012 & 2012 R2	Unlimited	Unlimited	0 Windows OS

**Note:** When it comes to licensing, you should consult with your reseller of choice to ensure that you are in compliance with all End User Licensing Agreements.

## Summary

Looking back, we can easily see that Microsoft has been consistently evolving Hyper-V based on customer, user and partner feedback. Benefitting from their own hyper-scale cloud environment, Azure, has allowed Microsoft to learn from their own findings and tune their internal triumphs and challenges. Microsoft plans to make many of these new learnings generally available to the enterprise within Windows Server 2016.

# What's New in Windows Server 2016 Virtualization

As previously mentioned, the focus of this eBook is to take a deep dive into the technical components within Windows Server 2016 Virtualization. This knowledge of the upcoming Hyper-V release will be invaluable and empower you, the reader, with the key knowledge of Hyper-V to support it when released. At the time of writing this eBook, Technical Preview 4 (TP4) was used for all scenarios and screenshots. As newer TPs become available updates will be provided. Now with that said, who is ready to get their learn on?

# Nano Server

In previous versions (WS 2008, 2008R2, 2012, 2012R2) when deploying the operating system, you had to choose which version and which mode. The options included Windows Server with a Graphical User Interface (GUI) or Server Core as seen in the image below. Server Core was a minimalistic version of Windows Server that only allowed a very small subset of operations to be completed. To aid the configuration was SConfig, which is a minimal interface that simplified many operations used either via Remote Desktop to the Server Core or through the Console. Also available through Server Core was Command Prompt, Notepad, Windows Installer (Msiexec), Registry Editor, System Information and Task Manager. All other operations needed to be performed remotely through Server Manager, MMC Snap-Ins or Remote PowerShell. This minimalistic footprint of Server Core provides many benefits within Cloud Environments.

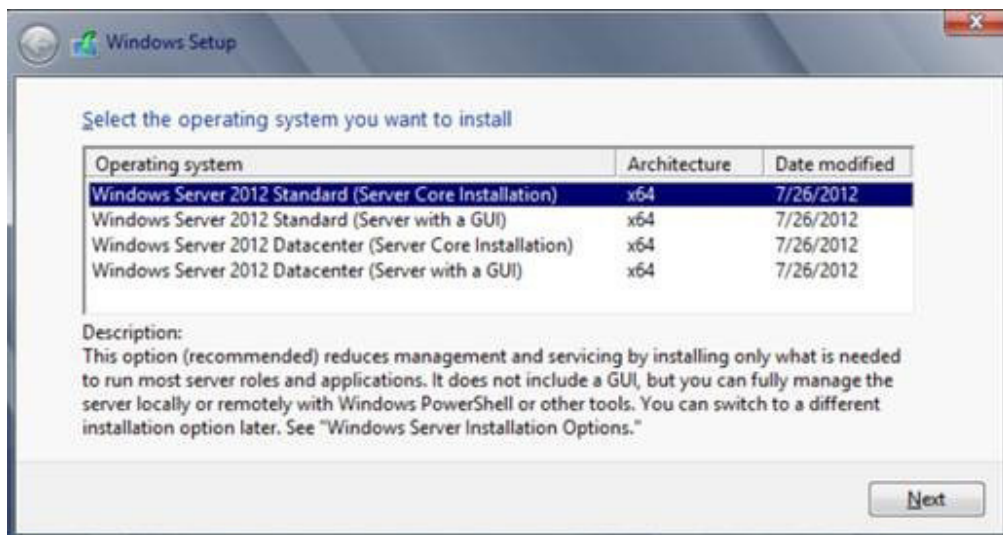


Figure 2: Windows Server 2012 Installation Options

Windows Server 2016 introduced a version that was even smaller than Server Core. This new version is called Nano Server, a headless 64-bit only, deployment option. Nano Server was created to serve as either a cloud fabric and infrastructure host (Hyper-V, Windows Failover Clustering, Networking and Storage) or as a deployment option for born-in-the-cloud applications such as ASP.NET v5 and Platform as a Service (PaaS) applications.

The key feature of Nano Server is the fact that it is truly headless. For example, you cannot remote desktop into a Nano Server, all operations must be done remotely. When deploying Nano Server, only the required packages for that instance are included. No unnecessary packages are included which reduces the attack surface and the footprint of the base image. Taking this approach not only speeds up deployment times, it also reduces the ongoing administrative effort when trying to manage Nano Server.

*Wait, so packages within the image? What's that mean?*

Nano Server by default contains zero binaries or metadata within the server, even drivers come as an add-on. This makes deploying Nano Server perfect for those that want to deploy ONLY what they need and keep the footprint as minimalistic as possible.

TP 4 Nano server is ideal for some key scenarios in your environment such as:

- Hyper-V host
- Storage host for Scale-Out File Servers
- DNS server
- Web server (IIS)
- A host for applications specifically designed for this
- Container host

## What Does Nano Set Out to Fix?

Nano Server brings several advantages to the current processes used within the datacenter today. The typical IT Professional is familiar with the dreaded patch Tuesday, the second Tuesday of the month. The day that Microsoft releases Patches, Hotfixes and Security Updates to the public. These updates often times require reboots. Reboots cause downtime and the potential of introducing new risk into the environment. Nano Server requires far less Security Updates, Patches and Hot Fixes – this results in less reboots! Fortunately, in this scenario, less patches do not equate to less security. Microsoft has done research in 2014 to list out the differences.

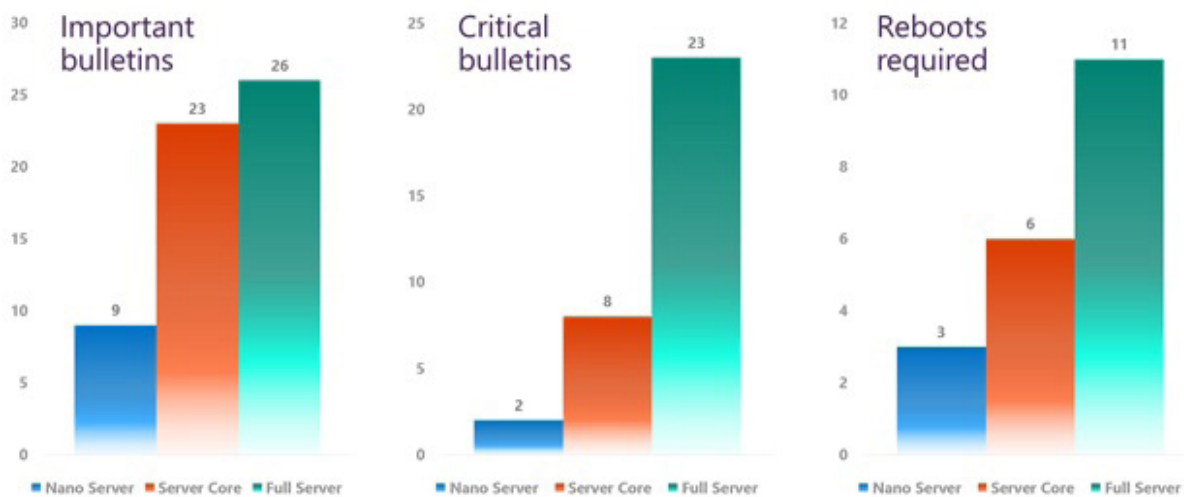


Figure 3: Patches & Reboots ©Microsoft

The figure above illustrates the differences from Nano Server to Server Core to Full Server. Notice that Full Server requires almost 3x more Important Updates and nearly 12x more Critical Updates than Nano Server. This is made possible because only the required components are deployed with Nano Server. Full Server requires many more resources and binaries to make the GUI experience possible resulting in a much larger attack footprint and potential for vulnerabilities. Also achieved within Nano Server is a much smaller disk and VHDx footprint, faster setup times and less internal processes.

## Summary

Nano Server presents many opportunities within the Modern Datacenter, and its possibilities are endless. We could spend all day writing about Nano Server, in fact Mike Ressler already has written an entire eBook, "All you need to know about Microsoft Windows Nano Server." This is a great read and it is strongly encouraged that you to read this eBook if you have not yet.

# Windows Containers

Through the course of IT history, there have been many great advancements in technology, the latest of which is Containers. This section will focus on Windows Server 2016 Containers. First, to level set and ensure that we are all on the same page, what seems like such a long while ago where IT Professionals were racking and stacking servers within the data center to install applications and operating systems on; this provided a 1:1 relationship. Then x86 virtualization came into the mix and at a high level Virtualization inserted an abstraction layer that separates the bare metal hardware that applications and servers used to reside on and the operating systems and applications being deployed. This provided many benefits that IT Organizations around the world are continuing to benefit from.

Containers take the foundation that server virtualization provides to the next level by allowing the kernel of the operating system to create multiple isolated user-space application instances, instead of one. The benefits gained from the Container approach is the ability to accelerate application deployment as well as reducing the efforts required to deploy apps. In the public cloud, this provides massive improvements that organizations of all shapes and sizes can benefit from. The ability to on-demand and at large scale stand-up and tear down environments provides much needed agility to the Developer Operations (DEVOPS) world. Hyper-V and the traditional virtualization we are familiar with in the modern data center is hardware virtualization; Containers is Operating System, Server Application, and Code virtualization.

In the end, the ultimate goal is to improve business productivity and have more scalable, better performing applications. Containers provide a great way for Developers to write and enhance their applications for the Cloud and continue to adopt the ‘write-once, run-anywhere’ mentality. This in turn enables the business to be more agile and respond faster to ever-increasing demands. IT Professionals can utilize the technology to help enable their Developers by providing standardized environments for all of the Development (DEV), Quality Assurance (QA), User Acceptance Testing (UAT) and Production (PROD) environments. Also, abstracting the hardware completely away from the applications and operating systems makes the underlying hardware infrastructure completely irrelevant. The common theme within Windows Server 2016 is optimization for the Cloud, whether that’s Public, Private or Hybrid. With the compute, storage and networking infrastructure layers optimally tuned and purpose-built to work with these next generation virtualization technologies, it’s possible to rapidly scale-up and scale-down environments based upon the changing needs of the business. Containers are a great example of the future of the Software Defined Data Center (SDDC).

## Windows Containers Architecture

As previously mentioned, Windows Containers provide isolated operating system environments, they run as an isolated processes within their parent OS. Windows Server 2016 Microsoft has embedded virtualization technologies within the Windows kernel that provides the ability to create multiple instances of the Windows application run-time. The image below is an illustration of the new Windows Container architecture for Windows Server 2016.



Figure 4: Architectural Layout of Containers within Windows Server 2016

For example, Application 1, Application 2 and Application 3 depicted in the image above represent the front-end of a Sales Ordering System. Each individual application environment believes that it is its own instance of Windows. During peak holiday season or large annual sales, the environment can quickly and easily be scaled to meet the demands.

Containers differ from the traditional VM that IT Professionals are used to deploying. VMs are completely segmented, virtualized instances of hardware and operating systems that run applications. Defined within them are virtual hard disks, unique operating systems, virtual memory and virtual CPUs. The image below illustrates that each application has its own dedicated installation of an operating system. Application 1 could be deployed on Linux and Application 2 could be deployed on Windows – they are 100% independent from each other. With Containers, the parent OS is shared so all application instances would need to support the OS of the parent. Windows Containers technology brings forth two distinct types of containers that we'll discuss: Windows Containers and Hyper-V Containers. Both types are deployed, managed and function in the same fashion. The key difference is that they differ in the level of isolation provided between containers.

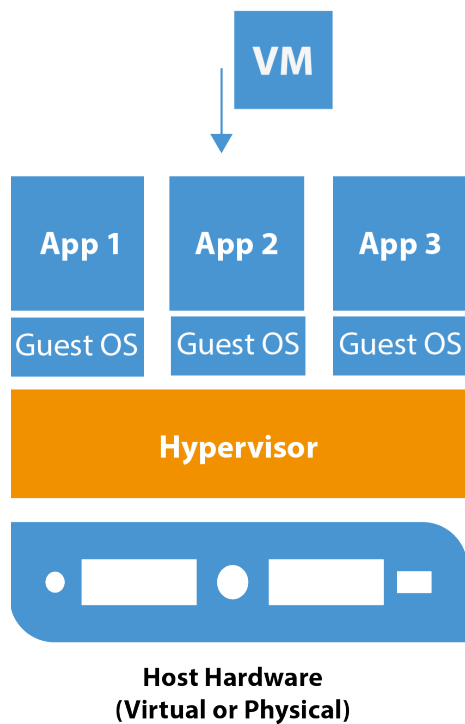


Figure 5: VMs Architecture that IT Professionals Deploy Today

## Applications within Containers

From a look, smell and feel perspective Containers operate much like traditional physical servers or virtual machines. VMs and Servers have operating systems and applications, just like containers, this is where the similarities end. Several key fundamentals make up a containerized application and we should begin with thinking about it in a layered approach.

- **Container Host**
  - Can be either a Virtual or Physical Windows Server 2016 Core or Nano server with the Container Feature enabled. Just like a Hyper-V Host, the Container Host will run multiple Windows Containers.
- **Container Image**
  - With a deployed Container all of the changes within the Container are captured in a sandbox layer. For example, if a Windows Server Core Container was deployed, then an IIS application is installed, these changes to the base are captured in the sandbox. Once the Container is stopped, those changes can be discarded or converted into a new Container image. This provides a highly scalable environment and ensures consistency.
- **Container OS Image**
  - This is first layer of the Container; the Container OS image cannot be changed. From this Container OS image, multiples of the same application can be deployed.
- **Sandbox**
  - With a deployed Container, all of the changes within the Container are captured in a sandbox layer.
- **Container Repository**
  - This is the location where Container OS Images are stored and deployed from. Container repositories are typically stored either in a shared location or on the Container Host itself.
- **Container Management Technology**
  - The management of Windows Containers can be from either PowerShell or Docker.

## Container Deployment and Image Creation

The Windows Server Container based OS image itself is an image that is provided and certified by Microsoft. All container based OS images are exactly the same regardless of the environment. With Windows Server 2016, the two container OS images available are Windows Server Core and Nano Server. The table below shows the options available within TP4. For an easy getting started guide, visit [msdn.microsoft.com/virtualization](https://msdn.microsoft.com/virtualization) and navigate to the Container Quick Start section.

Host Operating System	Windows Server Container	Hyper-V Container
Windows Server 2016 Full UI	Core OS Image	Nano OS Image
Windows Server 2016 Core	Core OS Image	Nano OS Image
Windows Server 2016 Nano	Nano OS Image	Nano OS Image

In the following scenario, outlined below are the steps required to deploy a virtualized container host that will run Hyper-V containers. As of Technical Preview 4, Windows 10 Build 10586 or later and Windows Server Technical Preview 4 or later are required to support nested virtualization.

- At least 4 GB RAM available for the virtualized Hyper-V host.
- Windows Server 2016 Technical Preview 4, or Windows 10 build 10565, on both the physical and the virtualized host.
- A processor with Intel VT-x (this feature is currently only available for Intel processors).
- The Container host VM will also need at least 2 virtual processors.

### Step 1: Change PowerShell Execution Policy

Check the Windows PowerShell Execution Policy to ensure that your machine is capable of PowerShell scripts. `Set-ExecutionPolicy` cmdlet can be used to assign either Restricted, AllSigned, RemoteSigned or Unrestricted.

#### Set-ExecutionPolicy Unrestricted

### Step 2: Download New-ContainerHost.ps1 from Microsoft

Download the `New-ContainerHost.ps1` configuration script from Microsoft. This command will automatically connect to Microsoft and download the configuration file that will be used to gather all of the necessary bits to deploy a new container host.

```
wget -uri https://aka.ms/tp4/New-ContainerHost -OutFile c:\Temp\New-ContainerHost.ps1
```



### Step 3: Create the Container Image for the Lab Environment

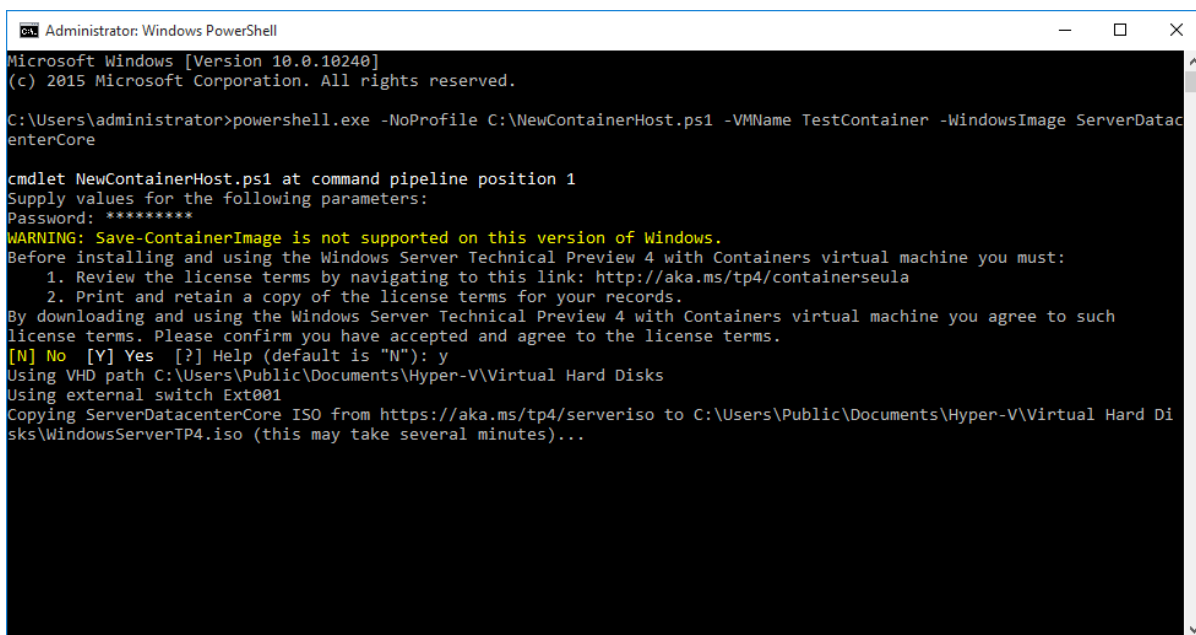
The configuration file that was obtained in Step 2 above contains all of the necessary information to download and deploy the container host as seen in the screen shot below. After this PowerShell script runs, the Windows Server 2016 Container OS is ready to use.

There are 3 Windows Images available and noted below in the command with the -WindowsImage switch. The three options are as follows:

- NanoServer
- ServerDatacenter
- ServerDatacenterCore

#### **From an Elevated Command Prompt:**

```
C:\WINDOWS\system32>powershell.exe -NoProfile c:\Temp\New-ContainerHost.ps1 -VmName VMName -WindowsImage ServerDataCenterCore -Hyperv
```



```
Administrator: Windows PowerShell
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\administrator>powershell.exe -NoProfile C:\NewContainerHost.ps1 -VMName TestContainer -WindowsImage ServerDatacenterCore

cmdlet NewContainerHost.ps1 at command pipeline position 1
Supply values for the following parameters:
Password: *****
WARNING: Save-ContainerImage is not supported on this version of Windows.
Before installing and using the Windows Server Technical Preview 4 with Containers virtual machine you must:
  1. Review the license terms by navigating to this link: http://aka.ms/tp4/containerseula
  2. Print and retain a copy of the license terms for your records.
By downloading and using the Windows Server Technical Preview 4 with Containers virtual machine you agree to such
license terms. Please confirm you have accepted and agree to the license terms.
[N] No [Y] Yes [?] Help (default is "N"): y
Using VHD path C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks
Using external switch Ext001
Copying ServerDatacenterCore ISO from https://aka.ms/tp4/serveriso to C:\Users\Public\Documents\Hyper-V\Virtual Hard Di
isks\WindowsServerTP4.iso (this may take several minutes)...
```

Figure 6: New-ContainerHost.ps1 run from Elevated Command Prompt

The command above will communicate with Microsoft and download the corresponding .iso file for whichever of the 3 server images you selected. This download only happens upon the first run of the command and will then deploy a .vhd into the environment. In this case **c:\Users\Public\Documents\Virtual Hard Disks** is the location for Virtual Hard Disks. Once this command completes successfully you will have a fully functional virtualized **Windows Server 2016 Core** Hyper-V Container host. This example has walked through the process to create a nested virtualization environment that will host Windows Server Containers.

#### Step 4: Deploy a Windows Based Container

Once Steps 1-3 have been successfully completed, the Windows-based Containers can now be deployed. An example of how to deploy a Windows Contained based on Windows Server 2016 Server Core is listed in the PowerShell code below.

```
$MyVeryFirstContainer = New-Container -Name "Container001"  
-ContainerImageName WindowsServerCore -SwitchName "Virtual Switch"
```

#By adding the -runtime switch and using HyperV will make this container a Hyper-V #container.

```
$MyVeryFirstContainer = New-Container -Name "Container001"  
-ContainerImageName WindowsServerCore -SwitchName "Virtual Switch"  
-runtime HyperV
```

The screen shot below shows a Windows-based Container that is running nested on an instance of Windows Server 2016.

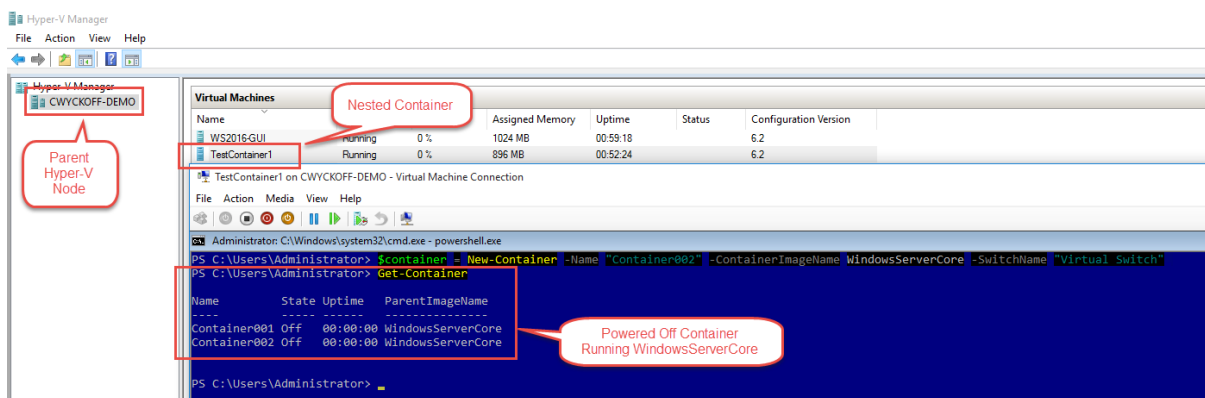


Figure 7: Windows Server Core Container OS Deployed Nested

## Container Management with PowerShell Direct

Once a Container has been deployed, it can be fully managed by using PowerShell or PowerShell Direct. The examples shown below will use PowerShell Direct.

To manage the container (Container001) that was deployed above, we can leverage PowerShell Direct from a management workstation. Simply `Enter-PSSession` into your container host and then `Enter-PSSession` into the Container, Container001. As an alternative, an IT Administrator could Remote Desktop or use Virtual Machine Connection on the Container Host.

The commands below provide a sample of the scenario described.

1. `Enter-PSSession` to enter PowerShell Direct to Container Host, TestContainer
2. Get the container ID
3. Use ContainerID to `Enter-PSSession` to Container001

```
#PowerShell Direct to Container Host and PowerShell Direct to #the ContainerImage
```

```
Enter-PSSession -VMName TestContainer1 -Credential chost001\administrator
```

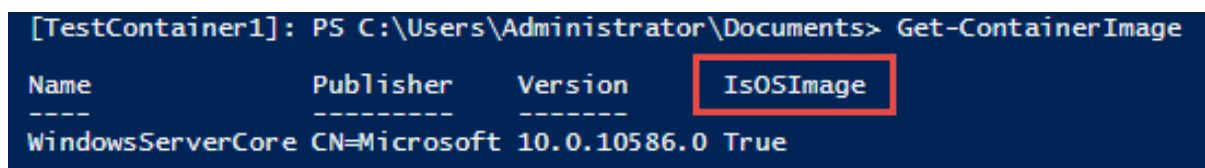
```
#Get-ContainerID to Enter-PSSession into Container001
```

```
Get-Container -Name Container001 | Select Name, ContainerID
```

```
Enter-PSSession -ContainerID 555a3d7e-560d-40a1-9b44-672024956962 -RunAsAdministrator
```

4. Display the Container Image by using `Get-ContainerImage`

**Note:** `Get-ContainerImage` will specify whether the ContainerImage is an OS Image or not.



```
[TestContainer1]: PS C:\Users\Administrator\Documents> Get-ContainerImage
```

Name	Publisher	Version	IsOSImage
WindowsServerCore	CN=Microsoft	10.0.10586.0	True

Figure 8: `Get-ContainerImage` will display whether the Image is an ISOImage or as an alternative an application image.

Now a Windows Container Image is running nested within our Hyper-V environment and is ready to have applications deployed. An Administrator could also add these container images to a domain and manage them like other Windows Servers.

WindowsServerCore and NanoServer are examples of base OS images that can have applications deployed inside. For a more comprehensive list of Container Applications that can be deployed as well as specific code samples, please reference Microsoft's GitHub Virtualization-Documentation portal at <https://github.com/Microsoft/Virtualization-Documentation>

## Docker and Windows Server 2016 Containers

Application virtualization separates the applications from the hardware (virtual or physical) by creating containerized instances of those individual applications. To manage Containers, it is important to understand the relationship of Microsoft Windows Containers and Docker. Docker, an open-source container management suite, provides everything that an application needs to run including the system library and code. Docker has also become a household name in the container ecosystem as they created a common toolset for Linux-based packaging and a deployment of applications to any Linux host. Docker containers ensure that the applications can run consistently, regardless of the environment, as long as the environment was Linux.

Within Windows Server 2016, Docker and Microsoft are working together to provide the same consistent experience across both the Linux and Windows ecosystem. Windows Server 2016 will extend functionality to run Docker on Windows.

## Docker Hub

### Step 2:

Download the [New-ContainerHost.ps1](#) configuration script from Microsoft. This command will automatically reach out to Microsoft and download the configuration file that will be used to gather all of the necessary bits to deploy a new container host.

```
wget -uri https://aka.ms/tp4/New-ContainerHost -OutFile c:\New-ContainerHost.ps1
```

### Step 3:

Create the container host within the environment. The configuration file that was obtained within Step 2 above contains all of the necessary information to download and deploy the container host (Figure 27). Afterwards, we are left with our newly created Windows Server 2016 Container OS ready

There are 3 Windows Images available and are noted below in the command with the `-WindowsImage` switch:

- NanoServer
- ServerDatacenter
- ServerDatacenterCore

### **From an Elevated Command Prompt:**

```
C:\WINDOWS\system32>powershell.exe -NoProfile c:\Temp\New-ContainerHost.ps1 -VmName VMName -WindowsImage NanoServer -Hyperv
```

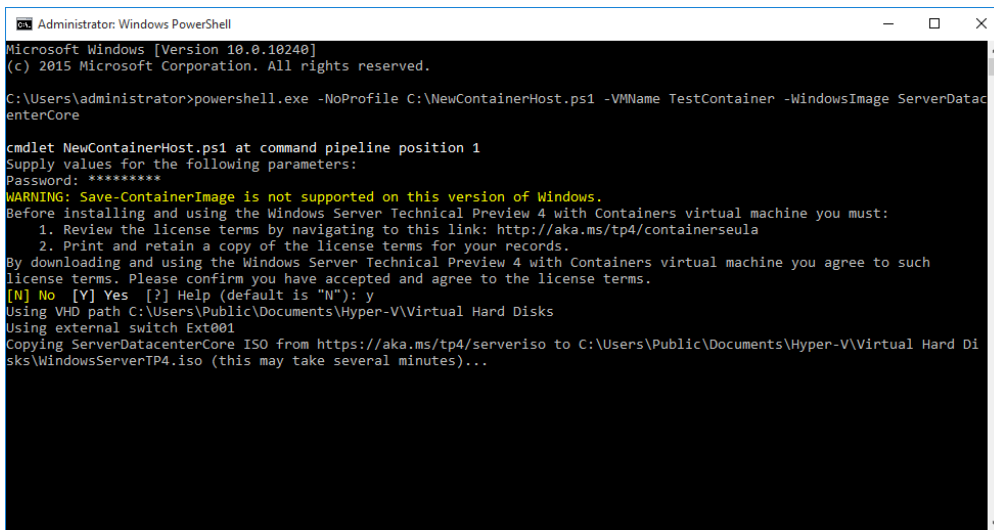


Figure 9: New-ContainerHost.ps1 run from Elevated Command Prompt

The command above will reach out to Microsoft and download the corresponding .iso file for whichever of the 3 server images you selected. This download only happens upon the first run of the command and will then deploy a .vhd into the environment. In this case, **c:\Users\Public\Documents\Virtual Hard Disks** is my location for Virtual Hard Disks on my Hyper-V host. Once this command completes successfully, we have a fully functional virtualized **Windows Server 2016 Core** Hyper-V Container host. This example illustrates the deployment within a nested virtualization environment.

#### Step 4:

With this, we are now ready to deploy our first Windows-based Container.

```
$MyVeryFirstContainer = New-Container -Name "Container001"  
-ContainerImageName WindowsServerCore -SwitchName "Virtual Switch"  
  
#By adding the -runtime switch and using HyperV will make this  
container a Hyper-V #container.  
  
$MyVeryFirstContainer = New-Container -Name "Container001"  
-ContainerImageName WindowsServerCore -SwitchName "Virtual Switch"  
-runtime HyperV
```

The command above will deploy a Windows Container utilizing the WindowsServerCore Image that was downloaded and deployed in Step 3 above (Figure 28).

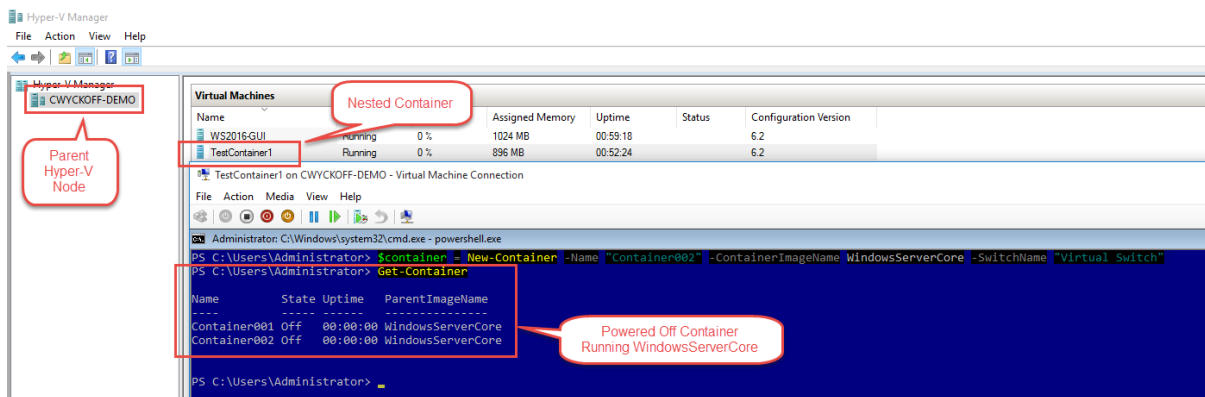


Figure 10: Windows Server Core Container OS Deployed Nested

To manage the container instance (Container001) that was deployed above, we can leverage PowerShell direct from our management workstation. Simply **Enter-PSSession** into your container host and then **Enter-PSSession** into the Container, Container001. As an alternative, an IT Administrator could Remote Desktop or use Virtual Machine Connection on the Container Host. The commands above provide a sample of the scenario described.

5. **Enter-PSSession** to enter PowerShell Direct to Container Host, TestContainer
6. Get the container ID

Use ContainerID to **Enter-PSSession** to Container001

```

#PowerShell Direct to Container Host and PowerShell Direct to #the ContainerImage
Enter-PSSession -VMName TestContainer1 -Credential chost001\administrator

#Get-ContainerID to Enter-PSSession into Container001
Get-Container -Name Container001 | Select Name, ContainerID

Enter-PSSession -ContainerID 555a3d7e-560d-40a1-9b44-672024956962 -RunAsAdministrator
    
```

7.

**Note:** `Get-ContainerImage` will specify whether the `ContainerImage` is an ISO

```
[TestContainer1]: PS C:\Users\Administrator\Documents> Get-ContainerImage
```

Name	Publisher	Version	IsOSImage
WindowsServerCore	CN=Microsoft	10.0.10586.0	True

Figure 11: `Get-ContainerImage` will display whether the Image is an ISOImage or as an alternative an application image.

As previously mentioned, now we have a Windows Container Image running nested within our Hyper-V environment ready to have applications deployed upon. Also, an Administrator could add these container images to the domain and manage them like other Windows hosts.

WindowsServerCore and NanoServer are base Image OS examples to deploy applications within. For a more comprehensive list of Container Applications that can be deployed as well as specific code samples, reference Microsoft's GitHub Virtualization-Documentation portal at <https://github.com/Microsoft/Virtualization-Documentation>

## Docker and Windows Server 2016 Containers

Application virtualization, as discussed within this section of the eBook previously, separates the applications from the hardware (virtual or physical) by creating containerized instances of those individual applications. Linux-based Docker has become a household name in the container ecosystem as the organization has created a common toolset, packaging model and deployment mechanism to allow the distribution of the applications on any Linux host. Docker, an open-source container management suite, provides everything that an application needs to run including the system library and code. Just like Windows Containers, Docker containers ensure that the apps can run the same everywhere, regardless of the environment, as long as the environment was Linux...that was then. Within Windows Server 2016, Docker and Microsoft are working together to provide the same consistent experience across both Linux and Windows operating systems, regardless if the workload runs on-premises or within the public cloud. Windows Server 2016 will allow the ability to run Docker on Windows.

IT Professionals and Developers that are familiar with Docker are likely familiar with the [Docker Hub](#). The Docker Hub is a collection of container applications that are consistently being contributed to within the source project. In summary, the partnership between Docker and Microsoft includes strategic investments from both parties to help develop the container ecosystem. The screenshot below is an example of the current Docker Hub interface.

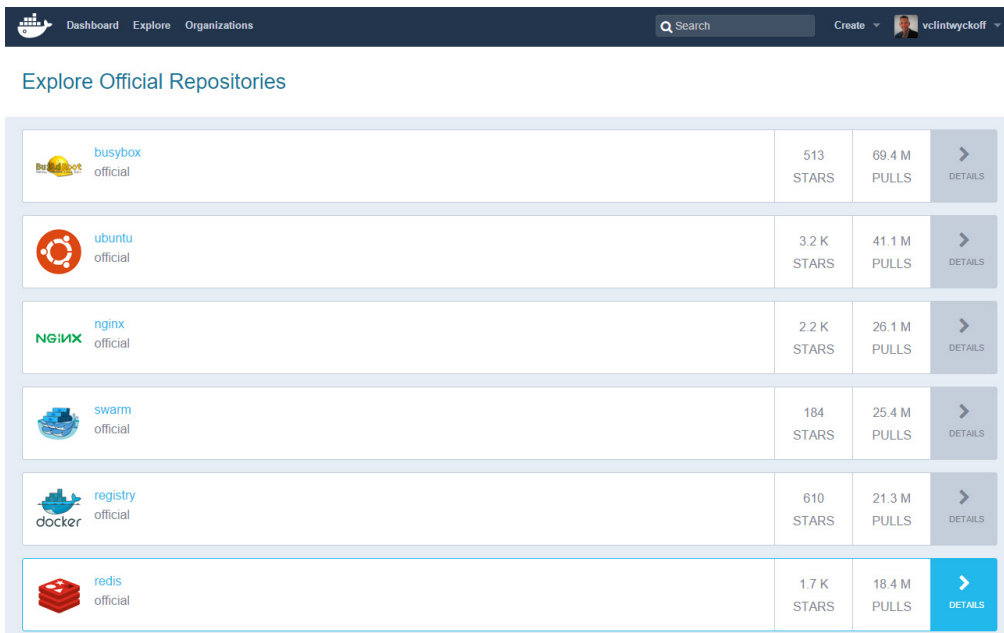


Figure 12: Docker Hub collection of open-source container applications

Windows Containers will be deployable through the Docker API libraries that developers are familiar with as well as the **Docker client**. One very important concept of Containers is that they leverage components from their Host. These variations are described in the table below:

Host Operating System	Supported Container	Management
Windows Server	Windows Container	Docker Client PowerShell PowerShell Direct
Linux	Docker	Docker Client

Since Windows Container leverages the container host kernel, they must run on a Windows-based Operating system. Likewise, Linux containers require Linux-based hosts. The strong partnership Microsoft has formed with Docker has allowed the Docker client to be able to manage both platforms.



## Docker Container and the Docker CLI

The Windows Container image contains the necessary command line interface libraries required to deploy Docker-based containers. For a step-by-step guide on how to create your first Docker container using the WindowsServerCore Container Image, visit [https://msdn.microsoft.com/en-us/virtualization/windowscontainers/quick\\_start/manage\\_docker](https://msdn.microsoft.com/en-us/virtualization/windowscontainers/quick_start/manage_docker). The screenshot below is an example of using the Docker CLI to manage a Windows Server Core Docker IIS Container.



Figure 13: Docker Container and Docker CLI

## Hyper-V Container

One of the key scenarios and deployment mechanisms of Windows Containers is the ability to run within multi-tenant cloud environments, i.e. Azure in the future. Cloud environments provide the ability to configure multi-tenant environments between applications and tenants. This requires a much deeper isolation between the container instances. Hyper-V containers are different from Windows Containers. These differences are listed in the table below:

Container Type	Security
Windows Container	Windows based security
Hyper-V Container	Kernel Isolation via a Type-2 Guest

If a Windows Container is configured using the `-runtime HyperV` switch, it will be configured as a Hyper-V Container instead of a Windows Container. This will provide a layer of complete isolation from all other containers on the host because it was converted to a full Hyper-V Guest instance. The PowerShell code below is an example of how to create a new Windows Server Core Hyper-V Container.

**#By adding the -runtime switch and using HyperV will make this container a Hyper-V #container.**

```
$MyVeryFirstContainer = New-Container -Name "Container001"  
-ContainerImageName WindowsServerCore -SwitchName "Virtual Switch"  
-runtime HyperV
```

## Hyper-V Container Deployment Example

Below is an example of when a Hyper-V container would be deployed versus a traditional Windows Container. As of TP4, this can only be deployed in a private cloud environment, as the Hyper-V Runtime Containers are not supported in Azure as of yet.

Imagine an environment that has an application set with multiple front end app servers along with a backend SQL database all running Windows Containers. All of the components within the stack are trusted amongst each other; they are within the same trust boundary. If we were to deploy another separate application with its dependencies on the same Container Host, we would be introducing what Microsoft refers to as "hostile multi-tenancy." In the trusted scenario, if malware or an intrusion was made, there would only be one application affected. By introducing a second, third or fourth... application on this Container Host, we have now introduced multiple trust boundaries. Code from application A could affect the performance of application B.

Hyper-V containers provide further isolation allowing each container to have its own dedicated copy of the Windows kernel with directly assigned memory, CPU and IO. In the end, they are isolated and the trust

boundaries are isolated. The caveat is if Hyper-V Containers are used you will get less density on the Host.

If a Windows Container was originally created without using the `-runtime HyperV` switch, it can be easily converted to and from a Hyper-V Container. The screen shot below shows an example of an existing Windows Container converted to a Hyper-V Container.

```
[TestContainer1]: PS C:\Users\Administrator\Documents> Get-Container | Select Name, RunTimeType
Name          RuntimeType
-----
Container001  Default
Container002  Default

[TestContainer1]: PS C:\Users\Administrator\Documents> Set-Container -Name Container001 -Runtime HyperV
[TestContainer1]: PS C:\Users\Administrator\Documents> Set-Container -Name Container002 -Runtime HyperV
[TestContainer1]: PS C:\Users\Administrator\Documents> Get-Container | Select Name, RunTimeType
Name          RuntimeType
-----
Container001  HyperV
Container002  HyperV
```

Figure 14: Set-Container -RunTime to convert to type HyperV

## Summary

The IT Industry in bridging the gaps between Development and IT Operations through DEVOPS. DEVOPS and the management of the Development process by using either Windows Containers or Docker is a great example of this new world. This will provide a consistent environment regardless of location along with great benefits for scalability. Microsoft is embracing the Open Stack Community with its tremendous investments in Windows Container technology. This investment will continue to close the gap between what used to be two distinctly different ecosystems.

# Top New Features of Windows Server 2016 Hyper-V

The release of Windows Server 2016 will introduce one of the largest code upgrades that Microsoft has ever released. To put this in context, this would be like moving from Windows NT 3.51 directly to Windows Server 2012 R2. With that, there have been a number of great new features that have been added to the Microsoft Hyper-V stack.

## Production Checkpoints

Checkpoints, also known as Snapshots in previous versions of Windows Server, are a mechanism for capturing a state of a virtual machine. Checkpoints allow a changed state to revert back to when the checkpoint was taken. When originally developed, Microsoft intended for Snapshots/Checkpoints to only be used for Development and Lab environments. It was common practice in many organizations to use these Snapshots/Checkpoints in Production to revert back to changes. For example, it has been well documented that sometimes hotfixes and patches can cause issues with production systems. Once discovered, organizations would simply revert a VM from a previous state to fix the issue. This was not supported and not recommended by Microsoft.

A major advancement in Windows Server 2016 is the release of Production Checkpoints.

Previous versions of Windows Server Hyper-V used .XML-based files to represent VM Memory and the state of VM Devices respectively at the time of the Checkpoint. So not to be confused with Production files, these Checkpoint-specific files must be stored within a separate Checkpoint File Location (Figure 3). New to Windows Server 2016, Microsoft has now deprecated the .XML file format and have since introduced .VMCX and .VMRS file formats. We will get into this deeper within the Virtual Machine Configuration File chapter of the eBook. The last portion of the checkpoint architecture is the differencing disk that's used. This differencing disk follows the .AVHD(x) file format and is stored in the same directory as the Production .VHD(X) file. While the Checkpoint is open, all writes that occur are captured within this differencing hard disk. At the time of replay, the VM is powered off, the blocks of data are merged to the production .VHD(X) and the VM is brought back online.

Let's take a look at this problem a bit deeper and use SQL Server as an example. With Standard Windows Server Checkpoints, all of the disk and memory state is captured, this includes in-flight transactions. So when you choose to apply this checkpoint, the application can have issues rolling back to this point in time. Production Checkpoints are fully supported for all Production applications as the technology now uses Windows Backup technologies. VSS is used inside the Windows guest operating system and System Freeze on Linux to appropriately place the application in a consistent state during the checkpoint process.

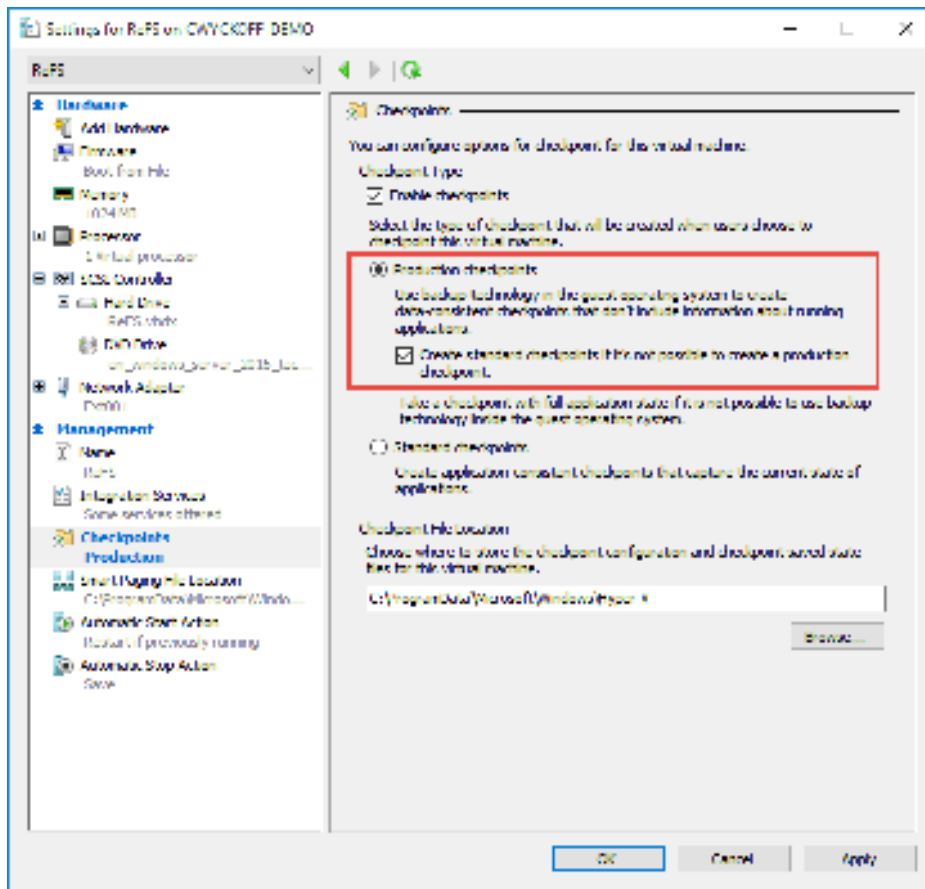


Figure 15: Checkpoint Settings of an individual Virtual Machine and Checkpoint File Location

Figure 3 continues to illustrate the settings available on an individual virtual machine. All VMs that are created on Windows 10 or Windows Server 2016 TP 4 have Production Checkpoints enabled by default, however, you can choose via checkbox to revert to standard checkpoints if production is not available.

To change between types of checkpoints:

1. Right click on the VM, choose Settings.
2. Within the Management pane, choose Checkpoints
3. Click either Production or Standard Checkpoints.

```
Set-VM -Name VM_Name -CheckpointType Disabled
Set-VM -Name VM_Name -CheckpointType Production
Set-VM -Name VM_Name -CheckpointType ProductionOnly
Set-VM -Name VM_Name -CheckpointType Standard
```

In Figure 4, below, the example leverages PowerShell to change the checkpoint type to standard and then initiate a checkpoint with the name, StandardCheckpoint.

```
Set-VM -Name VM_Name -CheckpointType Standard  
Get-VM -Name VM_Name |Checkpoint-VM -SnapshotName StandardCheckpoint
```

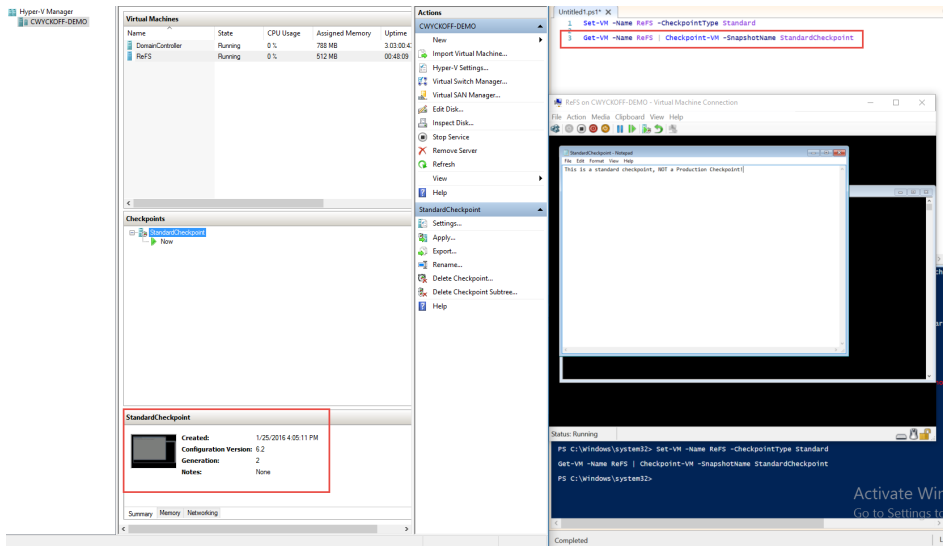


Figure 16: Standard Checkpoint using PowerShell

As previously mentioned, Standard checkpoints capture the memory and disk state of the virtual machine, so when reverted, the VM comes back up in exactly the same state as it was when the checkpoint was initiated. As seen below in Figure 5, upon applying checkpoint, StandardCheckpoint our VM comes directly back as it was before.

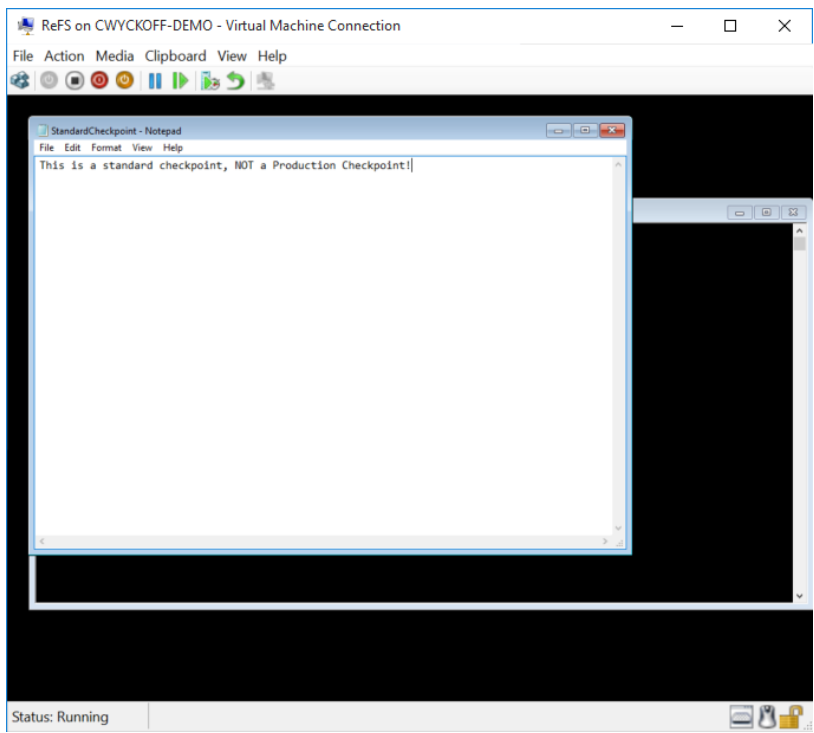


Figure 17: Standard checkpoint revert – Memory saved state

To enable Production checkpoints and replay this example, we can use the GUI within Hyper-V Manager or Powershell.

Within Hyper-V Manager, using the steps listed above, change the checkpoint type to Production and leave the checkbox un-checked — this way we are forcing Hyper-V to use Production checkpoints. Whenever you take a manual snapshot through Hyper-V Manager with Production Checkpoint enabled, you receive a confirmation that Production Checkpoints were used (Figure 6).

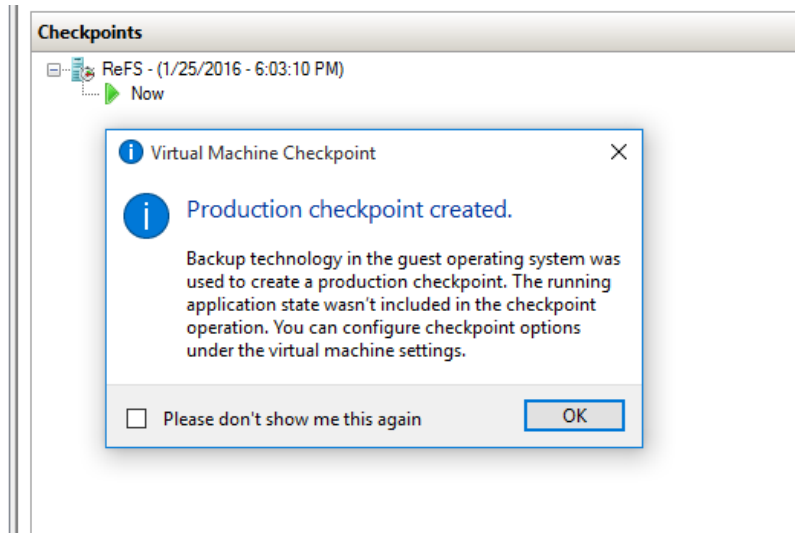


Figure 18: Production Checkpoint Confirmation Message

```
Set-VM -Name VM_Name -CheckpointType ProductionOnly
Get-VM -Name VM_Name | Checkpoint-VM -SnapshotName
ProductionCheckpoint
```

The key difference between Standard Checkpoints and Production Checkpoints is Volume Snapshot Service (VSS) is used for Windows VMs, and Linux-based VMs flush their file system buffers to create a file system consistent checkpoint. These are the same technologies that are used within image backup processes, making it possible to now checkpoint production workloads that include SQL Server, Exchange, Active Directory and SharePoint for example.

Figure 7, below, shows that whenever this Production Checkpoint example is applied, our VM is brought up in a clean state. Meaning the Guest Operating System feels and looks as though it was shut down properly. Keep in mind we are still within Technical Preview, after applying a Production type snapshot you MUST manually power the VM back on.

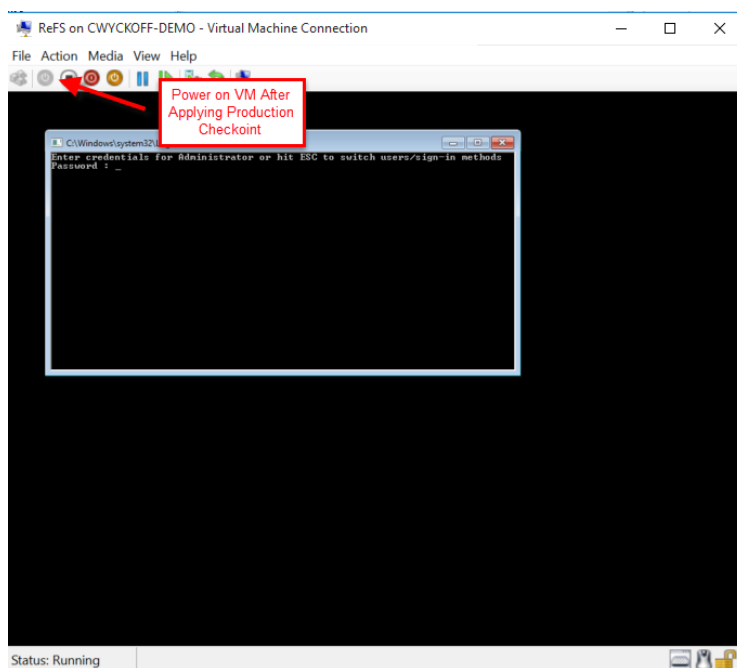


Figure 19: Post Production Checkpoint – Power on VM!

## PowerShell Direct

PowerShell is a great tool for remotely administering and managing virtual and physical machines. Physical machines do offer the ability of connecting to their DRAC, iLO or Remote KVM to perform actions when there is zero network connectivity.

PowerShell Direct gives IT Professionals the ability to run remote PowerShell commands against a guest Hyper-V VM without the IP network requirement. This feature is supported on Hyper-V hosts that are running Windows 10 or Windows Server 2016 Technical Preview 3. The guest VM must also be running Windows 10 or Windows Server 2016 Technical Preview 3 or greater in order to be managed.

PowerShell Direct utilizes the VMBus of the Hyper-V host to communicate with the Guest VM. Traditional PowerShell requires PSRemoting to be enabled and the VMs to have network connectivity. With PowerShell Direct, one could boot up a VM, connect to the VM, configure networking and add to the domain with ease.

Microsoft has introduced 2 new variables into PowerShell -VMName and -VMGuid. When connecting to the VMs, first log into the Hyper-V host or Windows 10 desktop. It is possible to use PSRemoting to connect to the parent host and within the PSRemote session then enter PowerShell Direct.

**Enter-PSSession** is an interactive session to the remote VM. Through this method, your connection remains sticky until you Exit the PowerShell session or close the PowerShell window.

```
Enter-PSSession -VMName VM_Name -Credential localhost\administrator
```

```
<Run your commands>
```

```
Exit-PSSession
```



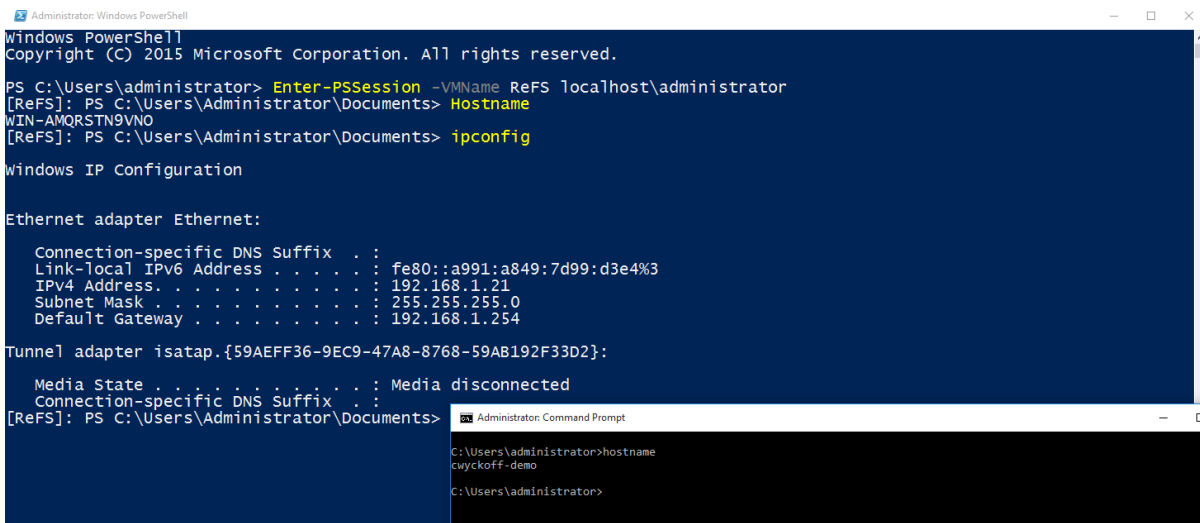


Figure 20: PowerShell Direct Connecting Using -VMName

Another method to execute commands within a remote VM is **Invoke-Command**. **Invoke-Command** uses PowerShell Direct and is the preferred connection method if executing an entire script. **Get-Credential** is used to store the credentials within the session, this is used when running multiple lines or commands within a single session.

**\$Credential = Get-Credential**

**Invoke-Command -VMName VM\_Name -Credential \$Credential -ScriptBlock { Get-Process }**

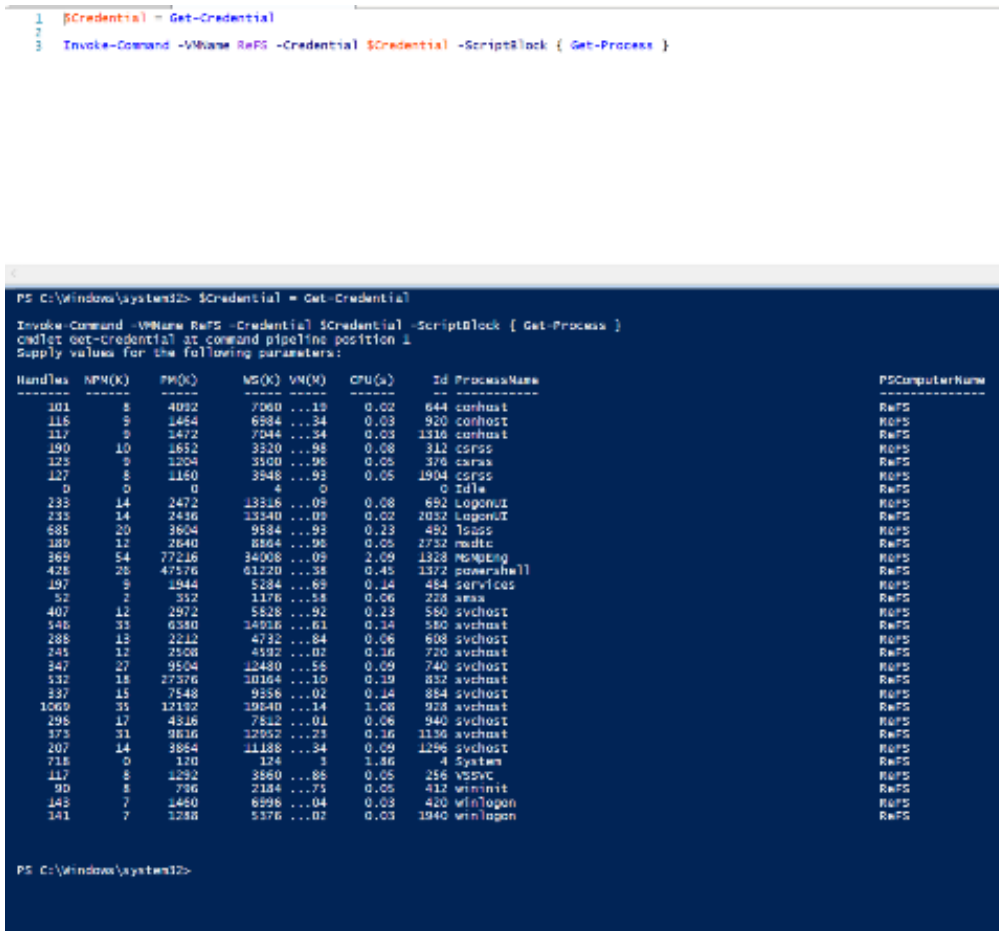


Figure 21: Invoke-Command method to run remote script block that lists out all processes on the VM

## Hyper-V Manager Enhancements

Hyper-V Administrators have come to know Hyper-V Manager very well over the years. It is one of the native Management Tools that Microsoft provides to manage standalone and a small number of remote Hyper-V nodes. Hyper-V Manager is included and available through Programs and Features such as Hyper-V Management Tools on any operating system that has Hyper-V as an installable feature. This includes Windows 8, 8.1 and 10. Windows Server 2016 offers many enhancements including Alternate Credential Support, the ability to manage previous versions of Hyper-V as well as an updated management protocol.

The image below displays how to utilize Hyper-V Manager to connect to a remote Hyper-V node. You can connect to remote Hyper-V nodes using Fully-Qualified-Domain-Name (FQDN) or IP Address using alternate credentials from what is being used locally. These new remote management and alternate credential capabilities utilize WinRM as opposed to WMI. When managing remote Hyper-V nodes, remote management must be enabled.

To enable WinRM from a PowerShell session, simply run:

```
Invoke-Command -ComputerName VM_Name -ScriptBlock { winrm quickconfig }
```

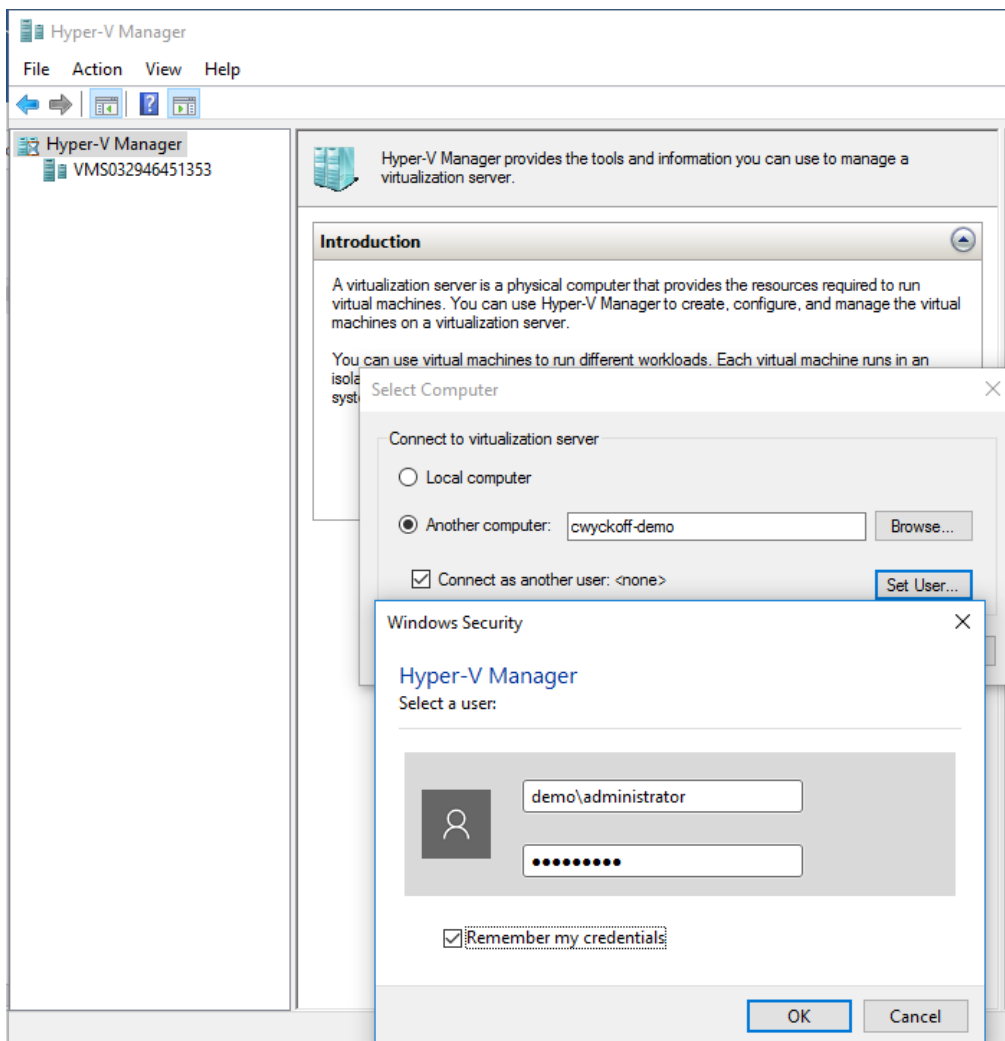


Figure 22: Remote Connection to Hyper-V Node with Alternate Credentials

Adding the ability to manage multiple versions of Hyper-V from a single interface is a much needed and wanted addition as well. From a single Windows 10 or Windows Server 2016 installation, you can manage computers running Hyper-V on Windows Server 2012, 2012R2, Windows 8 and Windows 8.1.

Lastly, is the updated management protocol where Hyper-V Manager has been updated to support WS-MAN protocol, which supports CredSSP, Kerberos or NTLM authentication. This is a great addition as now it is possible to manage Hyper-V nodes outside of the existing domain or maybe even in a secure DMZ environment. This added authentication protocol makes it possible to perform live migrations without having to enable constrained delegation within Active Directory.

As an Administrator on the Hyper-V host to be managed:

1. Enable PowerShell Remoting – [Enable-PSRemoting](#)
2. Add the managing computer to the TrustedHosts ACL from an elevated **Command Prompt**
  - a. **WSMan:\localhost\Client\TrustedHosts -value "<Computer.fqdn.com>"**
  - b. **WSMan:\localhost\Client\TrustedHosts -value \* -force**

3. Grant the managing computer permission to delegate explicit credentials
  - a. `Enable-WSManCredSSP -Role Client -DelegateComputer "<Computer.fqdn.com>"`
  - b. `Enable-WSManCredSSP -Role Client -DelegateComputer *`

## ReFS Fixed VHD Creation

Within Hyper-V when creating Virtual Hard Disks, there is the option to create a dynamic hard disk or a fixed size hard disk. Dynamic hard disks are thinly provisioned; this disk type only consumes the blocks of data that are required. For example, if a 40GB dynamic hard disk was created and was only using 11GB for the operating system, the VHD(X) would only use 11GB worth of space. On Generation 1 VMs, dynamic hard drives suffered around 25% performance loss over fixed disks. Generation 2 VMs have reduced this performance penalty drastically, making it feasible to provision dynamic virtual hard disks when running Generation 2 virtual hardware.

When provisioning fixed size VHD(X) drives, Hyper-V must write out zeros for the entire size of the NTFS formatted Windows disk. For instance, when creating a SQL Server and provisioning a 150GB VHD(X) for the data directory, Windows would write out 150GB worth of zeros. Resilient File System (ReFS) was introduced within Windows Server 2012 with the purpose and design of solving data integrity, availability and scalability issues. It's recommended by Microsoft to deploy VMs on Cluster Shared Volumes (CSV).

Drive Format	Command	Time to Complete
NTFS	<code>Measure-Command { New-VHD -Path C:\Temp\NTFS.vhdx -SizeBytes 30GB -Fixed }   fl TotalSeconds</code>	17.0601 seconds
ReFS	<code>Measure-Command { New-VHD -Path C:\Temp\REFS.vhdx -SizeBytes 30GB -Fixed }   fl TotalSeconds</code>	1.565 seconds

Ben Armstrong and the Hyper-V team have made great advancements in making these ReFS and VHD(X) operations much more efficient for virtual disk creation and the amount of IO it takes to merge VM Checkpoints. These enhancements to the Checkpoint merge process will allow more frequent backups which will ultimately reduce the Recovery Point Objectives (RPO) for the applications and data within VMs.

## Hyper-V Integration Services

Hyper-V Integration Services is a required software package that runs within the Guest VM and provides a set of drivers that the VM requires to run properly. Hyper-V Integration Services also improves the integration between the Hyper-V host and the Guest VM by providing the following services:

- Operating System Shutdown
- Time Synchronization
- Data Exchange
- Heartbeat
- Backup (Volume Shadow Service)
- Guest Services

Each of these services can be either enabled or disabled. By default, all services are enabled with the exception of Guest Services. The diagram below displays the VM Settings. To navigate to the VM Settings, right clicking on the VM and choosing Settings, then Integration Services under the Management area.

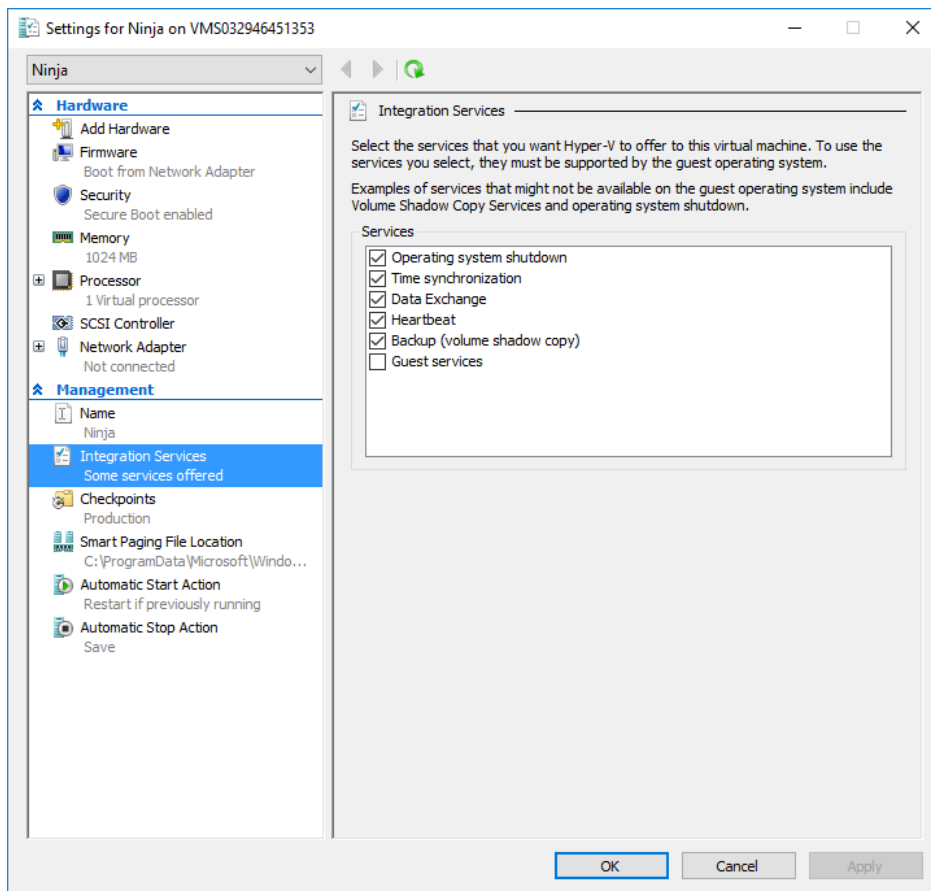


Figure 23: Hyper-V Integration Settings

The Integration Services provide many components to the Guest VMs. These services require ongoing maintenance and updates. On previous versions of Windows Server, the Integration Services were updated at the Hyper-V host level when Patches, Service Packs or Security Updates were rolled out. This update methodology causes version mismatches between the Hyper-V host and the Guest VMs, and places a large burden of keeping these services up to date manually through the host vmguest.iso or through a software distribution system.

With Windows Server 2016, the Hyper-V Integration Services updates will be delivered via Windows Updates. This provides a better update process for Administrators and ensures that these services are updated regularly. With the Integration Services being deployed through Windows Updates, the vmguest.iso has been deprecated and will no longer be included with Hyper-V.

Integration Services are not exclusive to Windows-based VMs — Linux distributions are also supported. There are many improvements in support for Linux in Windows Server 2016. This eBook contains a dedicated chapter focused on Microsoft and Linux.

## VM Configuration File Format

Each VM within the Hyper-V environment has a corresponding configuration file that holds all of the information about the individual VM. For example, the configuration file contains info about the vCPU and vRAM allocations, checkpoint policy and information that Hyper-V is managing and keeping track of as well. Before Windows Server 2016, this configuration file was an XML-based format. The XML format can lead to performance issues on larger deployments. In testing on Windows Server 2012 R2, Ben Armstrong and the Hyper-V team enabled Hyper-V Replica on 100 VMs with an RPO of 30 seconds. The constant updating of the each VM's XML-based configuration files took most of an entire CPU core on the Hyper-V host.

Windows Server 2016 introduces a binary format for tracking VM Configuration, .VMCX and .VMRS. This new file format serves the purpose of fixing two key areas of concern:

1. Performance
2. VM Configuration File Corruption

When the scenario above is compared to the new binary, non-XML-based file format, performance was decreased to around 19% of the single CPU core. This save performance can be used for running VMs since it is not being spent updating VM configuration files.

The second challenge Microsoft set to resolve was VM configuration file corruption. At large scale, it has been observed on a very infrequent basis that VM config. files can become corrupt. The new .VMCX and .VMRS file format brings forth a new change logging algorithm. As changes occur, they are first written to a log, which is then replayed into the actual configuration, and then the log is cleared. When corruption occurs, it is easy to repair the corrupted configuration file by systematically replaying the log.

VM configuration files have a non-standard naming convention. The VM configuration file name contains the characters that make up the VMID; otherwise known as the VMGuid. When executing PowerShell Direct, the option of using -VMName or -VMGuid is available. The sample PowerShell line below is executed on the Hyper-V host, and will retrieve the VMName and VMID.

```
Get-VM -Name HV001 | select VMName, VMID
```

The image below illustrates the output of the above PowerShell as well as the VM Configuration files stored on the Hyper-V host. By default, VM configuration files are stored in 'C:\ProgramData\Microsoft\Windows\Hyper-V'. This can be changed to an alternate location if desired.

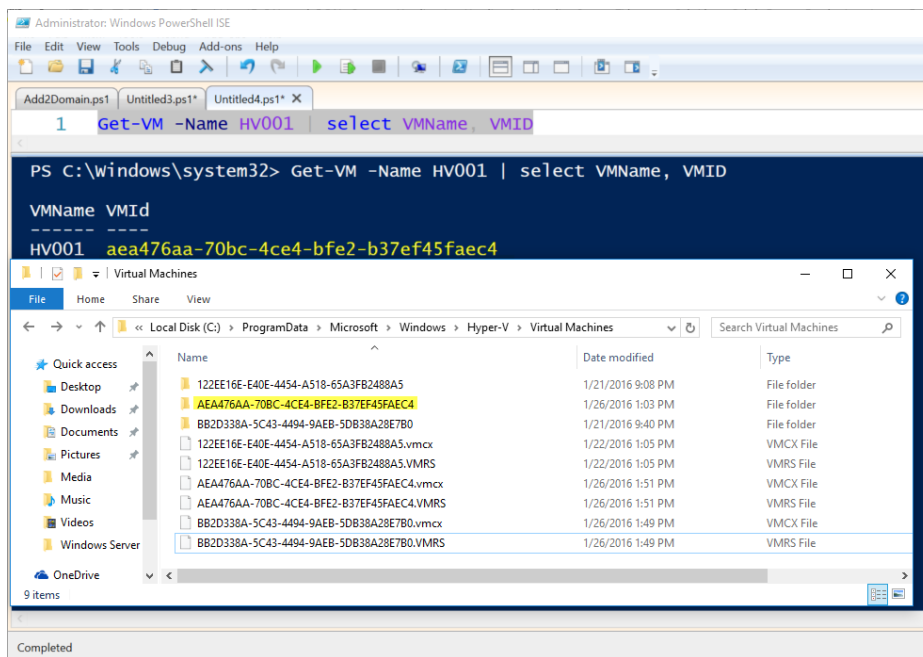


Figure 24: VMId Output from PowerShell and VM Configuration Files New in Hyper-V 2016

## Hypervisor Power Management — Connected Standby

With Windows 8, the Hyper-V role was an available option that was recommended for lab deployment purposes. These notebook style devices use the Always On / Always Connected power model which caused battery life issues. Windows Server 2016 and Windows 10 now fully support the Connected Standby state, resolving battery life issues whenever the Hyper-V role is enabled on notebook computers.

## RemoteFX vGPU and VDI

Virtual Desktop Infrastructure (VDI) running in Hyper-V as it relates to high-powered graphics, intensive workloads has been a challenge for Microsoft VDI customers. RemoteFX provides the ability to run 3D graphics within a VM where the VM leverages and utilizes physical hardware graphics cards within the Hyper-V host. In Windows Server 2016, Microsoft has made quite a few RemoteFX and vGPU improvements.

- OpenGL 4.4 and OpenCL 1.1 API
- RemoteFX on generation 2 VMs
- Larger dedicated vRAM and configurable amounts vRAM
- 4K Graphics Support

The steps required to enable RemoteFX have largely remained the same between Windows Server 2012 R2 and Windows Server 2016, however, it is recommended to visit [Microsoft TechNet](#) for the latest steps and updates required. You should also consult with the deployed graphics card to ensure that the card is supported on Windows Server 2016. The graphics card manufacturer can also provide documentation on the latest GPU supported drivers.

Veeam Vanguard and Microsoft MVP, [Didier Van Hoya](#), has a great [blog post](#) where he performed initial testing on Technical Preview 4 of Windows Server 2016 Hyper-V. If VDI with GPU is an area of interest, this article is worth checking out.



# Security Enhancements in Windows Server 2016 Virtualization

Looking back over the course of the previous few years, there has been significant increases in the amount of security breaches that have stemmed from hackers, malware and phishing attempts. The digital era of today and all line of business (LOB) applications have some type of online and/or internet facing presence. Regardless of which vertical the business operates within, security has become an extremely important aspect of the modern datacenter.

When it comes to VMs, Microsoft views Administrators of the Infrastructure as being one of the areas of exploitation. Some of the most common attacks are social engineered phishing attacks where administrator credentials are compromised. Insider attacks by the IT Administrator have been increasing as well.

To correct the situation, Microsoft views that IT needs to change the way that IT Security is viewed. Legacy models of thinking fall into the “protect the castle” mentality while the new thought process should realize and assume that a breach will occur. With this breach, how fast can IT be notified? How fast can IT respond to the breach? With IT shifting their thought process as it relates to security, they can begin to think more effectively about securing the IT environment and LOB applications.

Windows Server 2016 Virtualization aims to resolve these key challenges:

1. How is the environment protecting the Guest VMs from the Hyper-V Host and the credentials of the administrator of the host.
2. How do I know if I am deploying VMs to a host that has already been compromised?
3. If the environment has been compromised, how can IT protect individual virtual hard disks?

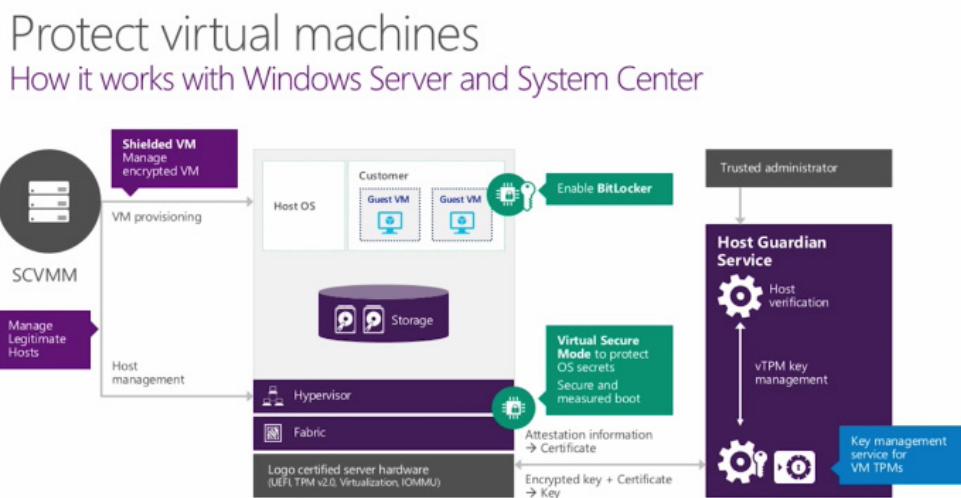


Figure 25: How Security Works with Windows Server and System Center Microsoft

## Server Security Concepts

Before diving into the individual features that solve these challenges, a few areas of the technology that make up these security enhancements will be defined. Hardware vendors have been designing and shipping PCs and Servers with Trusted Platform Module (TPM) chips installed on the motherboard. These PCs and Servers operate as the Hyper-V host. Introduced in Windows 7, Bitlocker is a hard disk encryption feature that scrambles or encrypts all of the data stored on the hard disk. Bitlocker leverages TPM to not only protect the hard disks when lost or stolen, but also validate the integrity of boot and system files. In the event an unsuccessful boot was made, access to the system will be prohibited. New to Windows Server 2016 Hyper-V is Virtual Trusted Platform Module (vTPM), which provides the same in-guest encryption as TPM but only for VMs.

## Virtual Secure Mode

Modern day servers and personal computers (PCs) have several different components within them: CPU, Devices and Memory. When the Windows Operating System is installed, access is granted to run privileged code on these pieces of hardware. When running Hyper-V on that same piece of bare-metal hardware, the installation of the Operating System with Hyper-V is what communicates with the memory, CPU and other devices within. Hyper-V controls access to memory within the system through Second Level Address Translation (SLAT), this restricts the parent OS' access to the privileged resource. New within Server 2016 and Windows 10 is Isolated User Mode (IUM). IUM separates the parent OS into two distinctly separate Hyper-V controlled operating environments, both with kernel mode and user mode. One runtime is a secure operating environment which is run in an isolated address space, separate from the normal Windows kernel. The separate address spaces are referenced in a hierarchical fashion through Virtual Trust Levels (VTL) where VTL 0 represents the traditional Windows kernel and VTL 1 represents the IUM runtime environment.

This new security feature was introduced in Windows 10 Hyper-V and is a crucial improvement for Windows Server as more and more workloads continue to be deployed in a hybrid-cloud (on-premises and off-premises) scenario. The IUM runtime environment is where all of the system components and devices are run from. Zero third-party code can be executed within this secure IUM environment and the code base inside is consistently being checked for any modification. If the Windows kernel is compromised, there is zero access inside the IUM.

For more details on Virtual Secure Mode, visit [channel9.msdn.com](http://channel9.msdn.com) for a great in-depth video by [David Hepkin](#) who is a member of the Windows Engineering Team.

## Shielded VMs and Guarded Fabric Hosts

In concept, Shielded VMs (Generation 2) should be protected from theft and tampering from both malware and a Hyper-V Administrator perspective. These Shielded VMs cannot be interacted with in any way, they are completely isolated. There is no console access provided, and keyboard and mouse interaction is not available.

Shielded VMs provide the ability of installing a Virtual Trusted Platform Module (vTPM) inside the VM along with the presence of either Bitlocker or a 3rd party full-disk encryption solution to ensure that only the designated owners can run the VM. It's important to understand is that a physical TPM is NOT required to utilize vTPM inside the VM with Windows Server 2016 TP4 (build 10586)

Shielded VMs and vTPM are distinctly different. With Shielded VMs, when the Administrator chooses to Live Migrate the VMs from one Hyper-V host to another, the traffic is encrypted over the wire. Also, when checkpoints are utilized, they are encrypted as well. Imagine a Service Provider (SP) scenario where an infrastructure is provided to run Hyper-V workloads. Currently this SP could interact with the console and send keystrokes as well as make kernel mode attacks. Secondly, this SP could power off the VM, double-click the VHD(X) to mount the virtual hard disk and gain access to the data within. Shielded VMs are protected against all of these scenarios. It is also, possible to convert a running virtual machine into a Shielded Virtual Machine, making it easy to move from traditional mode to Shielded. While, vTPM is simply running in-guest encryption that is leveraging the vTPM virtual device.

In this same SP example, Microsoft also provides Host Guardian Services (HGS). HGS is added to an environment through the Add Roles and Features. The HGS allows a tenant the ability to grant run permissions to the hosting provider. This allows the SP the ability to run their tenant's existing VMs, or the tenant can create new VMs directly on the IaaS provided.

Host Guardian Service is not exclusive to the SP use case; the Enterprise use case is valid as well. Any environment looking to provide a secure hardware environment for VMs and applications while knowing their data is protected from insider Administrator attacks as well as outside attempts.

When Shielded VMs are deployed on guarded hosts within the fabric, these hosts can provide host attestation. There are two modes available: Hardware Trusted and Active-Directory Admin Trusted.

Mode 1, hardware trusted attestation, provides the best security available and is the most complex. Hardware trusted mode does require TPM 2.0 hardware, which is a new hardware technology, as well as UEFI 2.3.1. The benefits of H/W attestation mode is the ability to register each Hyper-V host's TPM and establish baseline configuration item policies for each node.

## Attestation Workflow (hardware-trusted)

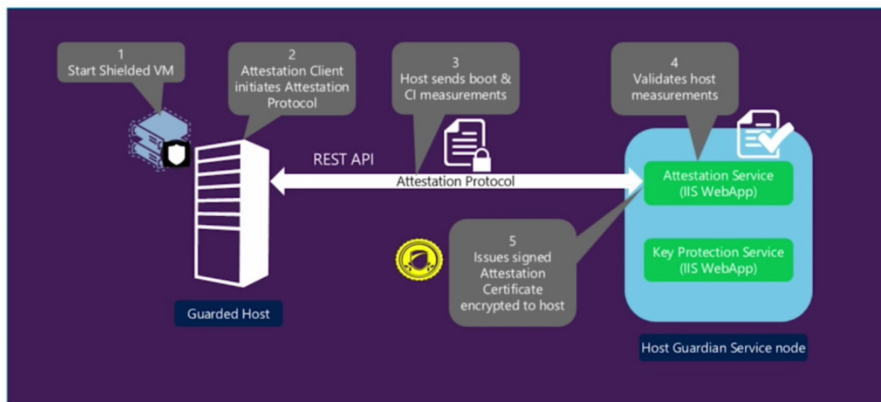


Figure 26: Attestation Workflow for Hardware Trusted Host Guardian Service ©Microsoft

Mode 2 is Active Directory-based (admin-trusted mode) and is easier to setup, however, it provides lower levels of assurance. This mode requires a separate Active Directory infrastructure for running the Host Guardian Service. The key difference between admin-trusted and hardware-trusted is the TPM presence within the hardware-trusted mode. With admin-trusted mode, the Guarded Host sends the Kerberos service ticket which proves the host is a member of the domain as well as resides within the necessary Security Group.

## Attestation Workflow (admin-trusted)

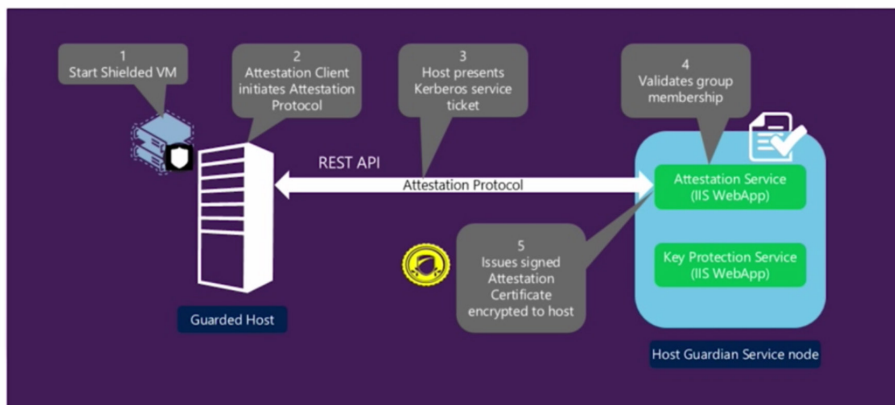


Figure 27: Attestation Workflow for Admin Trusted Host Guardian Service ©Microsoft

A typical deployment scenario would include a separate Active Directory Forest for the Host Guardian Services (HGS) along with a one-way trust to the domain where the Hyper-V hosts and VMs reside. This architecture is commonly referred to as the fabric infrastructure.

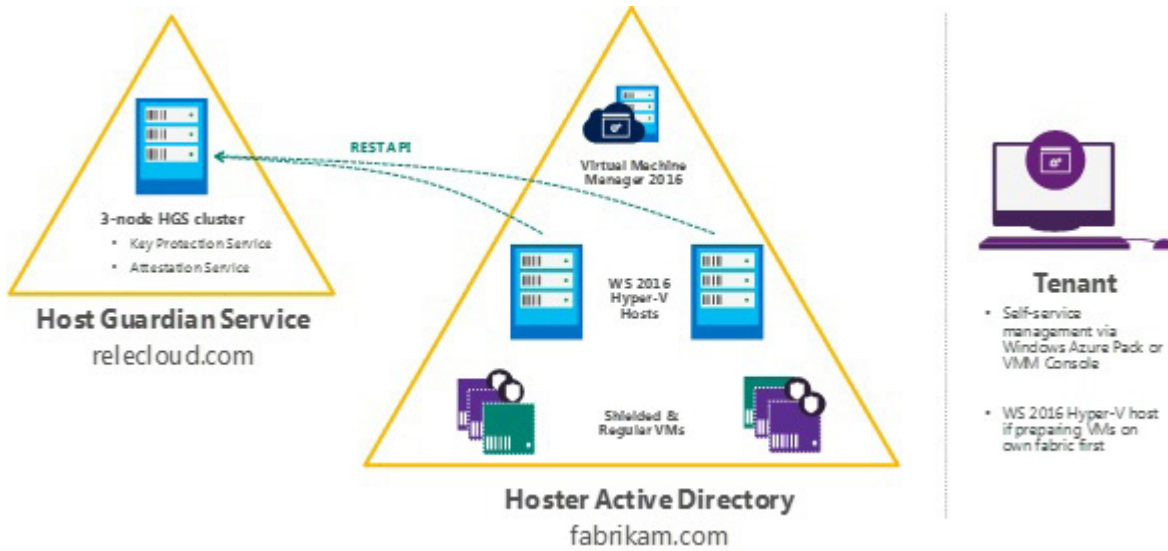


Figure 28: Logical Representation of the deployment topology

These key documents from Microsoft provide step-by-step instructions and great detail for deploying and administering TPM Attestation as well as Active Directory Attestation modes.

[Shielded VMs and Guarded Fabric Validation Guide for Windows Server 2016 \(TPM\)](#)

[Shielded VMs and Guarded Fabric Validation Guide for Windows Server 2016 \(Admin\)](#)

[Step-by-Step Guide: Deploy Shielded VMs Using TPM-trusted Attestation](#)

## Summary

Windows Server 2016 and, specifically, the virtualization pieces are making heavy advancements in the security of data regardless of geolocation. The security features within Windows Server and Hyper-V 2016 focus on securing not only the VMs and their parent hosts on-premises, but also ensure that the workloads being run off-premises are secure. It is safe to say that what is available today within Technical Preview 4 is only a small slice of what's going to be included when Windows Server 2016 does become Generally Available.

# Performance Isolation Techniques

One of the great benefits of the virtualized datacenter and the fabric that the VMs consume is the flexibility and dynamics it provides. Applications, network components, storage and compute nodes tend to misbehave from time to time. Within the datacenter, there is a phenomenon known as the “Noisy Neighbor.” Typically, the “Noisy Neighbor” is most visible within the shared storage infrastructure presenting unique challenges. These next set of features included in Hyper-V 2016 aim to solve these issues and make VM and application performance much more predictable.

## Storage Quality of Service (QoS)

Microsoft initially introduced Storage QoS in Windows Server 2012 R2. This initial iteration of Storage QoS allowed Hyper-V Administrators the ability to set minimum and maximum thresholds at a per-VHD(X) level as long as the VMs were running on the same Hyper-V node. Likely, the environment contains many Hyper-V hosts within clusters. These clusters require CSV disks present with running VMs. In the 2012 R2 scenario, when VM1 begins to have an input/output (IO) storm due to batch processing, the VM would begin to steal resources away from all of the other VMs on the CSV. The initial Storage QoS for Hyper-V was host exclusive, none of the other hosts were aware of the settings that were applied to the different VHD(X)s within the environment.

Windows Server 2016 will resolve this issue through a set of new VM contention and prevention techniques. Storage QoS within Server 2016 supports two different deployment models. First is Hyper-V using a Scale-Out File Server. Each Hyper-V host now contains a Rate Limiter, which will receive instructions from the brains of the operation, the Scale-Out File Server. The second is Hyper-V using Cluster Shared Volumes (CSV). It is here that the storage framework, Centralized Policy Manager, resides and tells each Hyper-V host which VMs get which storage policy applied and how much IO each VM is permitted. IO is relative and only makes sense to the individual applications that are generating the IO, for instance, SQL Server best practices recommend 64k while other applications may use 8k, 16k or 32k block sizes. Through Storage QoS, each block of data, regardless of size, is Normalized (Normalized IOPs) to a size of 8k. Any request smaller than 8k is normalized to one IO. If a 32k block request comes through, it would be normalized to 4 Normalized IOPs.

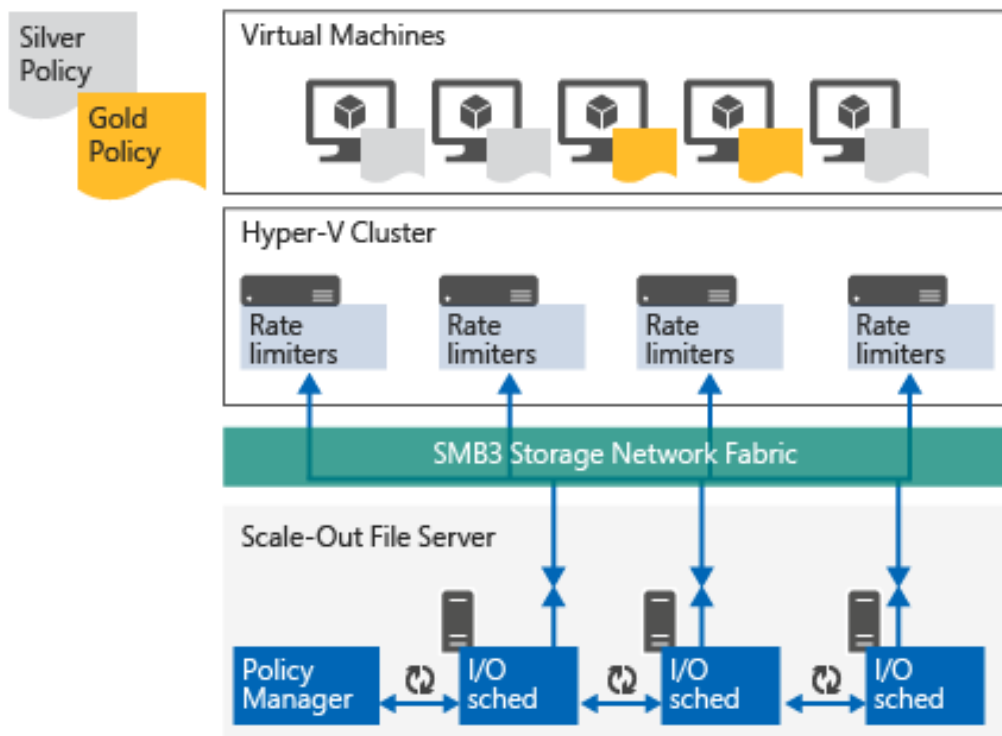


Figure 29: Using Storage QoS on a Scale-Out File Server ©Microsoft

There are two types of storage policies:

- Single Instance Policy
- Multi-Instance Policies

### Single Instance Policy

Single Instance policies combine minimum and maximum thresholds for a pool of VHD(X) files within the set policy. For example, when creating a Single Instance Policy with a minimum IO threshold of 500 IOPs and a maximum of 1,500 IOPs. This policy is then applied across a set of VMs. The result is that **COMBINED** these VMs will be guaranteed a minimum of 500 IOPs but **COMBINED** will not exceed 1,500. An overall limiting factor to bear in mind is the production storage that the VMs reside on, as it must be capable of keeping pace.

### Multi-Instance Policies

Multi-Instance Policies work similar to single instance policies with the minimum and maximum thresholds. The difference lies in that Multi-Instance Policies address each VM and their corresponding VHD(X) files separately. For example, when creating a Multi-Instance Policy with a minimum IO threshold of 500 IOPs and a maximum of 1500 IOPs, each VM is guaranteed at least 500 IOPs and each VM will never individually exceed 1500 IOPs.

## Storage QoS Management

To create and manage Storage QoS policies you should become familiar with basic PowerShell or utilize System Center Virtual Machine Manager (SCVMM). These tools are used to define and create policies at the Cluster level (SoFS or CSV) and then apply these said policies to the Hyper-V hosts.

The important item to note within Storage QoS is that *ALL* QoS policy creation is performed at the storage cluster level. The policy application is performed at the compute cluster or individual Hyper-V host level.

Below is a PowerShell example that creates a new Multi-Instance or Single instance storage QoS policy within the environment. The second piece of PowerShell is needed to gather the policy GUID which is used to apply the policy.

```
$PlatPolicy = New-StorageQosPolicy -Name Platinum -PolicyType  
SingleInstance -MinimumIops 500 -MaximumIops 1500  
$PlatPolicy = New-StorageQosPolicy -Name Platinum -PolicyType  
MultiInstance -MinimumIops 500 -MaximumIops 1500  
$PlatPolicy.PolicyID  
<GUID Format 12345678-1234-1234-1234-123456789abc'>
```

To apply the policy at a cluster level, the following PowerShell would be used to select the 'ReallyImportant' VM and apply the QoS policy that was created above. The `-QoSPolicyID` is the GUID that we gathered above with the `.PolicyID` reference.

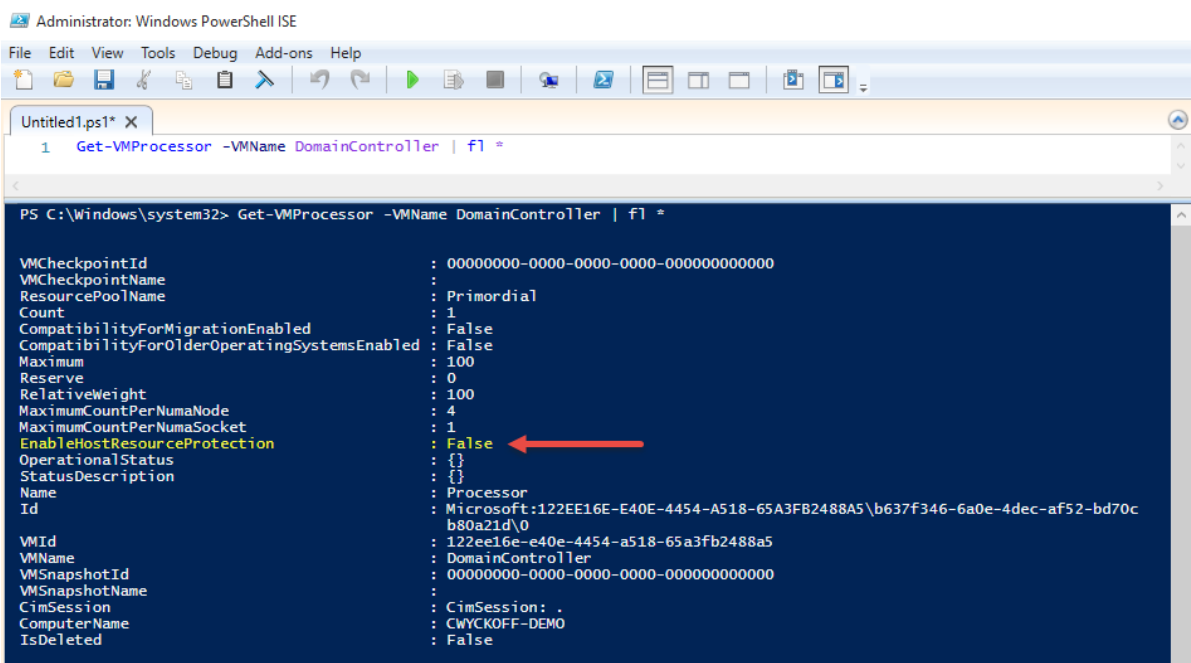
```
Get-ClusterGroup | Where-Object {$_.Name -is 'ReallyImportantVM'}  
| Get-VM | Get-VMHardDiskDrive | Set-VMHardDiskDrive -QoSPolicyID  
$GUIDFromBefore
```



## Host Resource Protection

Host resource protection is a technology that was initially built and designed for Microsoft's hyper-scale public cloud, Azure, and is now making its way into private cloud environments within Windows Server 2016. Host Resource protection is enabled by default whenever you install Windows Server 2016. Malware, ransomware and other malicious activities are becoming the norm both in public and private cloud environments. Host Resource Protection aims to identify abnormal patterns of access by leveraging its heuristics-based approach to dynamically detect malicious code. When an issue is identified, the VM's performance is throttled back as to not affect the performance of the other VMs that reside on the Hyper-V host.

Host Resource Protection can be disabled by issuing a simple PowerShell command.



```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* X
1 Get-VMProcessor -VMName DomainController | fl *
PS C:\Windows\system32> Get-VMProcessor -VMName DomainController | fl *
VMCheckpointId           : 00000000-0000-0000-0000-000000000000
VMCheckpointName         :
ResourcePoolName         : Primordial
Count                    : 1
CompatibilityForMigrationEnabled : False
CompatibilityForOlderOperatingSystemsEnabled : False
Maximum                  : 100
Reserve                  : 0
RelativeWeight           : 100
MaximumCountPerNumaNode  : 4
MaximumCountPerNumaSocket : 1
EnableHostResourceProtection : False
OperationalStatus       : {}
StatusDescription       : {}
Name                    : Processor
Id                      : Microsoft:122EE16E-E40E-4454-A518-65A3FB2488A5\b637f346-6a0e-4dec-af52-bd70c
b80a21d\0
VMId                    : 122ee16e-e40e-4454-a518-65a3fb2488a5
VMName                  : DomainController
VMSnapshotId           : 00000000-0000-0000-0000-000000000000
VMSnapshotName         :
CimSession              : CimSession: .
ComputerName            : CWYCKOFF-DEMO
IsDeleted               : False
```

Figure 30: Get-VMProcessor cmdlet used to view status of HostResourceProtection

**Note:** This eBook was written based on Windows Server 2016 TP 4. Within TP 4, Host Resource Protection is **DISABLED** by default– To enable, use the **Set-VMProcessor** cmdlet and reboot the Guest VM.

**Set-VMProcessor -VMName \$VM -EnableHostResourceProtection 1**

Above is the PowerShell required to enable Host Resource Protection on Windows Server 2016 TP4. You can run the Get-VMProcessor command to review and to ensure it has been applied correctly.

# Hyper-V Availability

Microsoft defines “Availability” as anything done within the Windows Server platform to keep your servers running; Anything that causes your VMs and Applications to be unavailable or powered off is unacceptable. Within the technology stack, availability encompasses a few areas: Failover Clustering, Compute Resiliency, Storage Resiliency and Replication. This chapter will dive into the new areas within Windows Server 2016 Hyper-V that will help ensure that the virtual infrastructure is ready to serve the applications and data the business requires.

## VM Compute and Storage Resiliency

Windows Failover Clustering introduces a unique set of challenges. The slightest interruption in network, storage or cluster communication can cause havoc for days. In Server 2012 R2 when communication errors occur, the nodes within the cluster begin to take resources offline and attempt to bring them back online as quickly as possible. In some cases, these attempts cause hours and, in rare instances, days’ worth of clean up for IT Professionals. In the event of a huge problem, this is the behavior we would want to have occur, this is NOT the behavior we desire whenever the Network goes bump in the night.

VM Compute Resiliency sets out to resolve these problems. VM Compute Resiliency will provide Clustered Hyper-V environments the ability to withstand minor service disruption(s) without aggressively failing over VMs and their services to other surviving Cluster nodes. The configurable default setting on Server 2016 is 4 minutes.

VM Storage Resiliency sets out to allow the minor bumps and bruises within the storage infrastructure without massive failover. Storage Resiliency also fixes the problems whenever storage does go offline for long periods of time by placing each VM in PausedCritical state. The VM state is updated and captured in memory prior to the VM even noticing the storage was offline. Hyper-V will hold these VMs and applications within the VMs in memory until the storage becomes available again.

The most disruptive situation in a Failover Cluster environment is commonly known as, “flapping cluster node(s).” This situation occurs whenever a node within a Hyper-V Cluster environment comes and goes online and offline many times within a short window. This WILL definitely wreak havoc on your environment as the cluster tries to keep pace with these transient errors. To resolve these transient issues, Microsoft introduced several new Failover Clustering States.

- **Unmonitored:** VM state that notes whenever a virtual machine is no longer being monitored by the Cluster Service
- **Isolated:**
- **Quarantine:** Host state that is noted whenever the node is no longer permitted to join the cluster (Default 2 Hours).
  - Node is quarantined when the node ungracefully leaves / joins the cluster 3x within an hour.
  - VMs running on the node are gracefully Live Migrated (Zero Downtime) from the node once it is quarantined.

If a node is in quarantine, it can be brought out manually by executing the `Start-ClusterNode` and by using the `-ClearQuarantine` flag.

```
Start-ClusterNode -ClearQuarantine
```

These settings are variable and allow the Administrator to tweak based on the environmental requirements. For more detailed information on the resiliency settings and the PowerShell required to alter the default settings, visit <https://blogs.msdn.microsoft.com/clustering/2015/06/03/virtual-machine-compute-resiliency-in-windows-server-2016/>

## Shared VHDX

Shared VHDX is the enabling technology that allowed customers and service providers the ability to run virtualized failover clusters. This technology was released in 2012 R2 initially and contained many limitations and known usability issues. Prior to Server 2012 R2, if a customer or service provider desired to run virtualized failover clusters, Guest iSCSI was required or, alternatively RAW, to map LUNs to the individual VMs within the cluster. This created a significant amount of manual management overhead as well as introduced many environmental limitations.

Windows Server 2016 is introducing the following abilities:

- Ability to do online resize of Shared VHDX virtual hard disk(s)
- Host based backup of guest running Shared VHDX
- Increased GUI usability
- Hyper-V replica shared VHDX

## Hyper-V Replica

Hyper-V Replica in Windows Server 2012 R2 provided a built-in way of replicating entire VMs from one location to another. The benefits of replication as opposed to traditional backup is that replication provides the best Recovery Time Objectives (RTO) for applications and services. Each VM can be configured to replicate the latest changes on a configurable frequency: 30 seconds, 5 minutes or 15 minutes. Software-based replication allows flexibility, it does not require like-for-like compute or storage hardware on the source or DR side of the network. For example, in the Production site, a customer may run an Enterprise SAN and have many VMs running on CSV block storage. At their DR site, they may target an SMB Share for file-based storage. In Server 2012 R2, to enable Hyper-V Replica on an individual VM and to initiate the wizard, simply right click on the VM and choose “Enable Replication.”

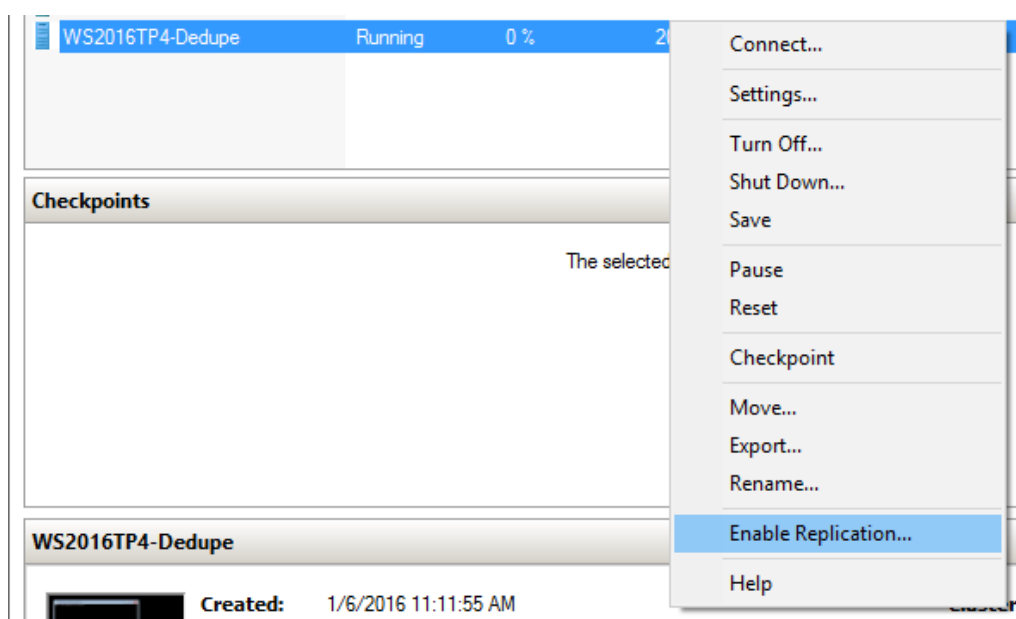


Figure 31: Enable Hyper-V Replication on Windows Server 2012 R2

Microsoft provides a tool, The Capacity Planner for Hyper-V Replica, which provides IT Professionals guidance on appropriate storage, network and compute configurations that can help ensure a successful deployment and configuration.

The link to download this free utility is here:

<https://www.microsoft.com/en-us/download/details.aspx?id=39057>

Windows Server 2016 Hyper-V introduces Hyper-V Replica with VMs using Shared VHDX hard disk configurations and the ability to replicate these VMs into Microsoft Azure using Azure Site Recovery. Microsoft is also adding the ability to Hot-Add VHDX's to a running VM. The second piece of new functionality coming to Hyper-V Replica fixes some of the shortcomings previously within 2012 R2 such as when a customer added a new VHDX to a Hyper-V replica VM many errors and failures were encountered. Replicating Set and Not Replicating set of virtual hard disks is a new concept that allows the hot-add of a VHDX to a VM to occur. This new VHDX is automatically added to the Not Replicating set with no errors occurring during the next replication cycle. Below is sample PowerShell that adds all of the virtual disks for a given VM to the set of Replicated Disks. During the next replication cycle, Hyper-V will initiate a full replica of that individual VHDX with zero errors.

```
#Replicates all of the disks for VM "VMName"
```

```
Set-VMReplication "VMName" -ReplicatedDisks (Get-VMHardDiskDrive  
"VMName")
```

There are performance concerns to keep in mind when enabling Hyper-V Replica on VMs and their applications. Hyper-V Replica utilizes a journaling mechanism to keep track of data changes within a VM, the file extension .hrl (Hyper-V Replica Log) is used. Whenever an IO occurs to a VM, the data is written twice, once to the VHD(X) and second to the .HRL file. When replicating large amounts of transactional servers, it does not take much IO to saturate an Enterprise SAN that is not performing optimally. Due to the logging mechanisms, this IO doubled making two individual writes for each block of data. This will have performance implications if not sized correctly, and customers should take advantage of the Hyper-V Replica Capacity Planning tool.

## Memory Management

Dynamic memory was first introduced in Hyper-V 2008 R2 SP1. Dynamic Memory monitors the VMs activity and will dynamically expand and contract the memory assigned to the VM based on the system requirements. In all previous releases of Windows Server, one had to power off a VM that did not have Dynamic Memory to adjust if required. Hyper-V 2016 now allows you to change the assigned and startup memory allocated to a VM in-flight without having to power the VM off. The image below displays Hyper-V Manager in Hyper-V 2016 and shows the amount of memory that a VM is assigned as well as the amount of memory a VM is demanding. If Dynamic Memory was not enabled for this VM, this can be altered by editing the settings of the VM.

```
#Change Dynamic Memory Off and Set Memory to 8GB
```

```
Get-VM -VMName "VMName" | Set-VMMemory -DynamicMemoryEnabled 0 |  
Set-VMMemory -StartupBytes 8096
```

WS2016-GUI			
<b>Startup Memory:</b>	1024 MB	<b>Assigned Memory:</b>	538 MB
<b>Dynamic Memory:</b>	Enabled	<b>Memory Demand:</b>	457 MB
<b>Minimum Memory:</b>	512 MB	<b>Memory Status:</b>	OK
<b>Maximum Memory:</b>	1048576 MB		

Summary | Memory | Networking

Figure 32: Hyper-V Manager – Memory Demand vs. Assigned Memory

Hot-Add applies the ability of dynamically expanding VM memory, similarly you can decrease memory on a VM. If you attempt to decrease the memory on a VM below the amount being requested, Hyper-V will display an error message. Hyper-V will attempt to decrease the memory as much as possible while not starving out the VM.

## Networking Enhancements

New within Server 2016: Administrators can now hot-add a vNIC to an individual VM without any system downtime regardless if the guest VM is running Windows or Linux. Also, new to the Server Operating System is the ability to name a vNIC within Hyper-V Manager or PowerShell and have that name be reflected in the guest operating system. By default, these features are enabled and are only available on Generation 2 VMs. This is configured via PowerShell. Below is an example:

### #Add NIC and Rename

```
$VM = Get-VM -Name "VMName"
```

```
Add-VMNetworkAdapter -VMName $VM -SwitchName "vSwitch 001" -Name  
"Fancy New NIC" -Passthru | Set-VMNetworkAdapter -DeviceNaming On
```

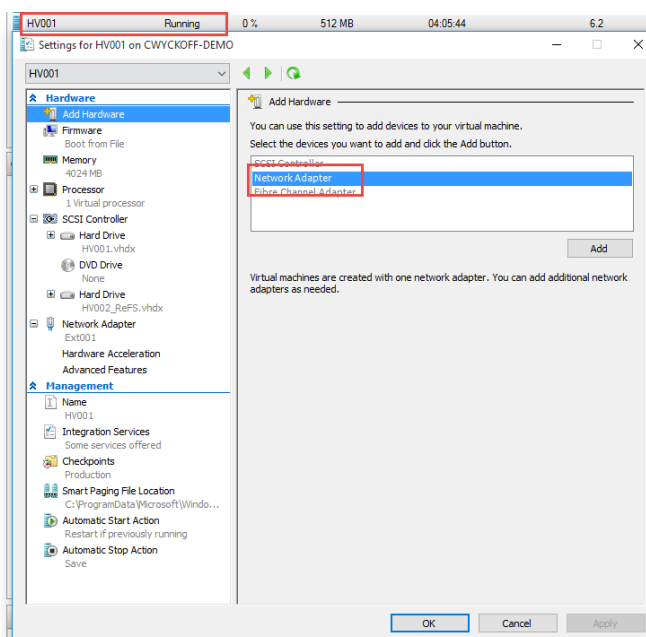


Figure 33: Hot-Add vNic to VM through Hyper-V Manager

# Upgrading the Environment to Hyper-V 2016

Upgrading versions of Windows Server has been unadvisable and is still considered to be frowned upon within the Windows Server community. In previous versions of Windows Server, it was required to build a separate environment with the new version of Hyper-V and Windows Server installed which was ready to receive VMs. Windows Server 2012 R2 allowed the ability to do unlike Cluster type (2012 or 2008R2 -> 2012R2) Live Migrations. This allowed IT Professionals the ability to upgrade the environment with zero downtime. The downside was that extra hardware would need to be utilized or purchased.

Windows Server 2016 introduces a new concept, Cluster OS rolling upgrades. This new upgrade feature within Server 2016 allows for Windows Server 2016 and Windows Server 2012 R2 cluster nodes to coexist within the cluster.

## **The steps to upgrade from Windows Server 2012 R2 Hyper-V OR Scale-Out-File-Server to Windows Server 2016:**

1. Drain roles off Node 1 within Microsoft Failover Cluster Manager or Virtual Machine Manager
2. Evict Node 1 from Cluster
3. Install Windows Server 2016 on evicted cluster node
4. Add evicted and newly upgraded Hyper-V server back to the cluster
5. Rinse and repeat until all of the nodes within the cluster have been upgraded
6. Upgrade cluster functional level

## Upgrading the VM Hardware Version

VM versioning has been a task that IT Professionals have dealt with whenever upgrading cluster versions. Previously when a VM was running a previous hardware version and was added to a 2012 R2 cluster, the VM version would have been upgraded automatically. Upgrading the VM hardware version is an irreversible process.

With Server 2016 and cluster version 2016 version 5, VMs (Windows Server 2012 R2) are fully supported in a 2016 cluster. Windows Server 2016 will not change the version of your VMs while providing full compatibility. This functionality provides the benefit of moving VMs from one cluster version to another with zero downtime to the workload and application. Support for the Hyper-V specific features, mentioned previously, are not available. When timing to upgrade the VM version is appropriate each VM is independent of the others. The hardware upgrade is accomplished by executing a single PowerShell command. Be advised that upgrading VM versions is an irreversible process and does require a VM reboot.

### Steps required to upgrade VM hardware version:

1. Power off VM
2. From an elevated PowerShell session on the Hyper-V host, (Can use PowerShell Remoting from external workstation) run the command:

```
a. Update-VMVersion "VMName"
```

## Hyper-V Supports Linux

Many Enterprise environments run both Windows-based workloads and applications alongside Linux-based workloads and applications. For years, the Microsoft Virtualization team has been contributing to the Linux communities and providing enhancements to ensure that when Linux workloads were deployed within a Hyper-V VM, all components work and operate as they normally would on another platform. Full support is dependent upon the Linux distribution and whether the specific vendor has adopted Hyper-V support.

For the Hyper-V Linux support matrix, visit: <https://technet.microsoft.com/en-us/library/dn531030.aspx>

Within Hyper-V, Linux VMs support both emulated devices as well as Hyper-V-specific virtual devices. The Linux Integration Services (LIS) or FreeBSD Integrations Services (BIS) are required to take advantage of the features of Hyper-V, i.e. Checkpoints, Backup Technology, Time Synchronization, etc. Many of the new releases of the Linux OS and FreeBSD have these integration services included in the operating system. The integration services for Linux provide the same levels of performance improvement and service enhancements as their Windows based counterparts. For legacy Linux workloads that do not have the Integration Services built-in there are LIS packages available for download through Microsoft that provide the device drivers required for the specific distribution that the workload is running.



Linux based workloads within Hyper-V provide support for the latest compute enhancements in relation to dynamic memory, 64 vCPUs within a single VM, online backup support, hot-add and online resize of VHD(X). Also new to Linux Guest VMs in Server 2016 is Linux Secure Boot. Secure Boot was a feature that became available with the introduction of Generation 2, Windows VMs on Server 2012.

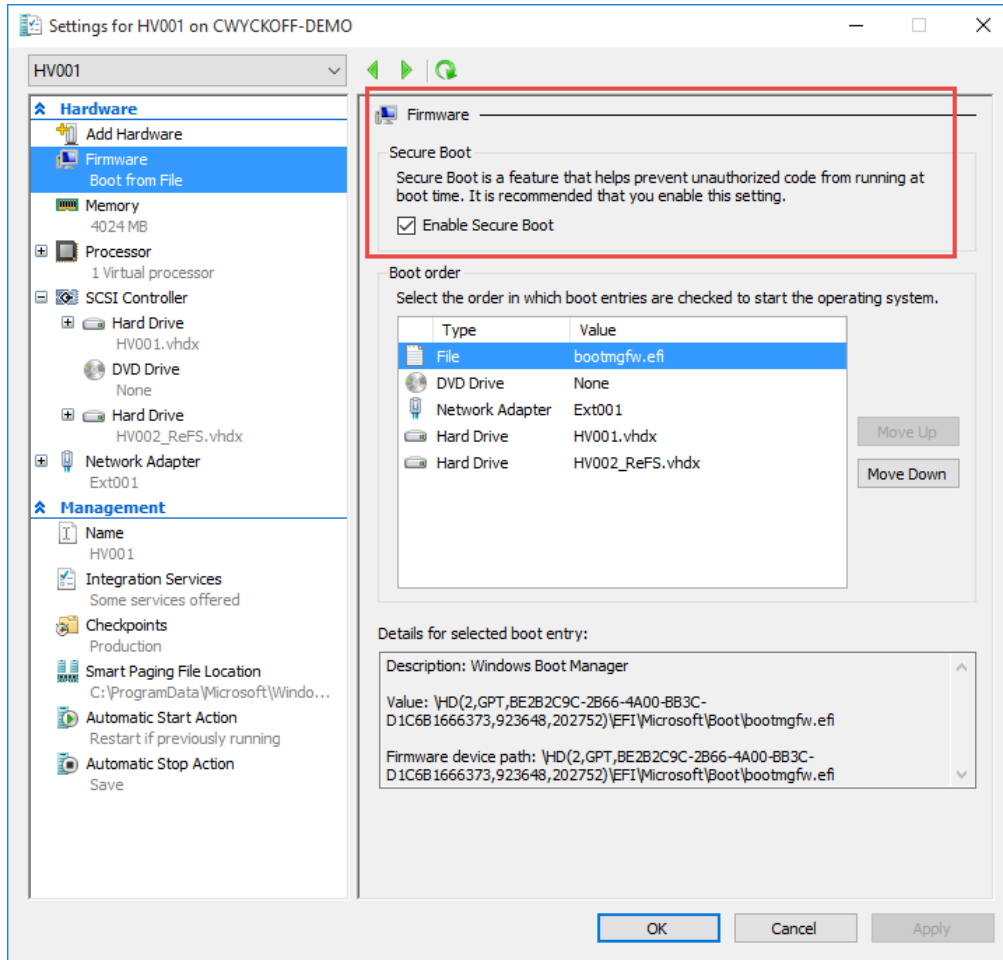


Figure 34: Secure Boot on Powered Off Windows VM

Secure Boot is activated on a VM by VM basis and provides a secure way through the UEFI BIOS to validate that only approved components are set to run on the Guest OS. Secure Boot is defined as part of the UEFI specification which is enabled by default on ALL Generation 2 VMs. The illustration above displays the Secure Boot option with the VM Settings. Microsoft recommends leaving Secure Boot enabled as it helps prevent unauthorized code from running at the time of boot. Secure Boot is fully supported and operational for Ubuntu 14.04 and later, SUSE Linux Enterprise Server 12 and later, Red Hat Enterprise Linux 7.0 and later, and CentOS 7.0 and later.

# Appendix A

## Licensing in Windows Server 2016

**Disclaimer: All of the licensing information contained within this eBook was retrieved from official documentation and is accurate as of 3/2016. This information is subject to change in the future. To ensure that your organization is licensed correctly, please contact Microsoft and/or your preferred Microsoft Partner.**

Windows Server 2016 Technical Preview 4 has been active for quite some time and has been used as a reference. Windows Server 2016 will be delivered in two separate editions (Note: Other editions may become available closer to general availability): Standard and Datacenter. Standard Edition is used within non-virtualized environments — i.e. Web Server, Application Server, SQL Server, etc. Datacenter Edition is to be used within highly virtualized environments where the use case is Hyper-V hosts. Within these editions, specific features and functionality are separated.

Windows Server 2016 Editions		
	Datacenter	Standard
Core functionality of Windows Server	•	•
OSEs / Hyper-V containers*	Unlimited	2
Windows Server containers	Unlimited	Unlimited
Nano Server	•	•
New storage features including Storage Spaces Direct and Storage Replica*	•	
New Shielded Virtual Machines and Host Guardian Service*	•	
New networking stack*	•	

Figure 1: Windows Server 2016 Editions, courtesy of Microsoft

In the table above, Datacenter Edition does permit an unlimited amount of Windows Guest OS Environments (OSE) to run and receive their licensing through the parent Hyper-V node. Standard Edition provides two Guest OSE to run and receive their licensing through the parent Hyper-V node. Windows Containers should be treated and licensed just as a VM. For example, when leveraging nested virtualization, which is new to Server 2016, the guest running these container instances would be licensed the same as an OSE. When running Linux workloads those particular guest operating systems require licensing through their specific distribution end-user license agreement (EULA).

The licensing change within Server 2016 is that Microsoft has moved away from the per-CPU Processor licensing model to a per-CPU Core licensing model. The second change to the server licensing is that there is no longer feature parity between Standard and Datacenter editions. As noted, this has changed from Server 2012 R2 where features were equal.

To license the physical cores within Server 2016 (Figure 2) it is required to license all of the cores within a physical server. The minimum number of cores to license equals 8 per processor or 16 per server while the minimum count of cores to license per physical server equals 16.

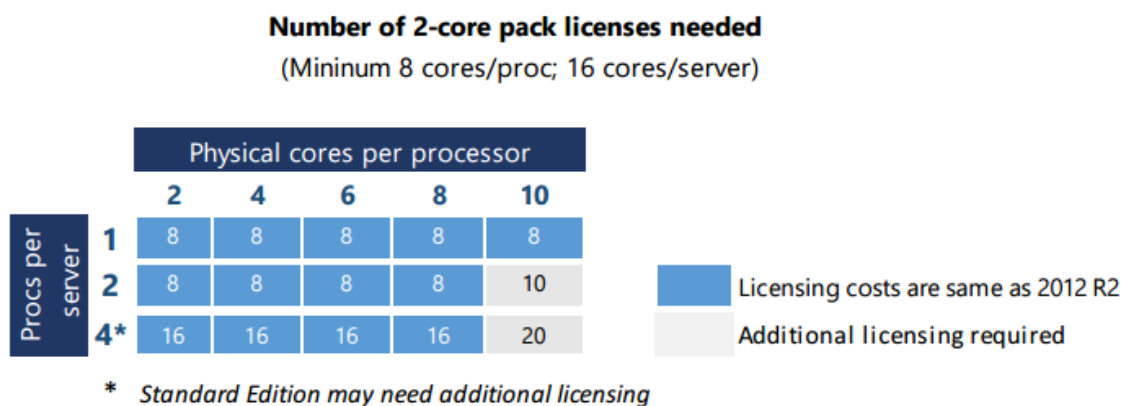


Figure 2: Server CPU Core Licensing, Courtesy Microsoft®

To learn more about Windows Server 2016, and be in the know of all the latest and greatest coming to Windows Server 2016, please visit the [Windows Server 2016 website](#).

[Windows Server 2016 Licensing Datasheet](#)

[Windows Server 2016 Licensing Frequently Asked Questions](#)

## Installing Windows Server 2016

Windows Server 2016 comes in two Editions, Standard and Datacenter. Both editions are deployable in full GUI mode or Core. Nano server is a component that is included within the installation media and built from the ISO of Windows Server. Installing Windows Server 2016 is an easy process that should not take much time at all regardless if being deployed on a physical piece of bare metal hardware or deployed within a VM. This section of the eBook will focus on outlining the steps required to deploy a Windows Server 2016 Core VM within a Hyper-V environment.

Before installing Windows Server 2016, one must first obtain the installation media which can be obtained through the TechNet Evaluation Center. If interested in learning more about Windows Server 2016, Technical Preview 4 is the most recent release available. When Windows Server 2016 does become Generally Available, which we anticipate to be later in 2016, one should utilize the normal methods of license procurement.

Once downloaded, mount the ISO image for Windows Server 2016.

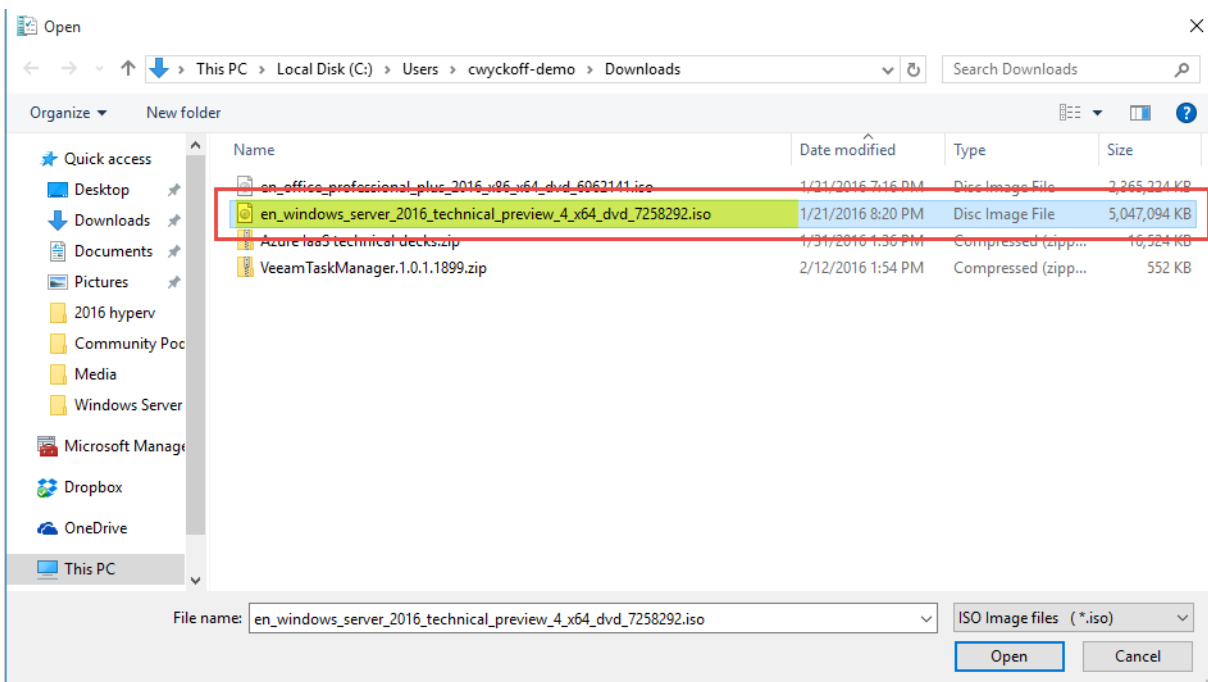


Figure 3: Windows Server TP4 ISO

With the ISO file, we are now ready to deploy our first Windows Server 2016 TP4 VM! If this ISO was being used to deploy Windows Server to a physical piece of hardware, there are many options for mounting ISO images available. Most modern servers have remote interface cards available. For example, DRAC or iLO allow for remote console and remote ISO mounting. This eliminates the need to burn a DVD or extract to a USB stick.

The [Windows USB / DVD Download Tool](#), a free utility, helps with creation of the bootable media. Another free open source tool that can help with the ISO image creation is [Rufus](#).

Navigate and launch Hyper-V Manager. This can be done via a remote workstation utilizing the alternate credential option within Windows 10 and Hyper-V 2016. To begin the New Virtual Machine wizard, right-click on the root name space in the left pane and choose, New Virtual Machine.

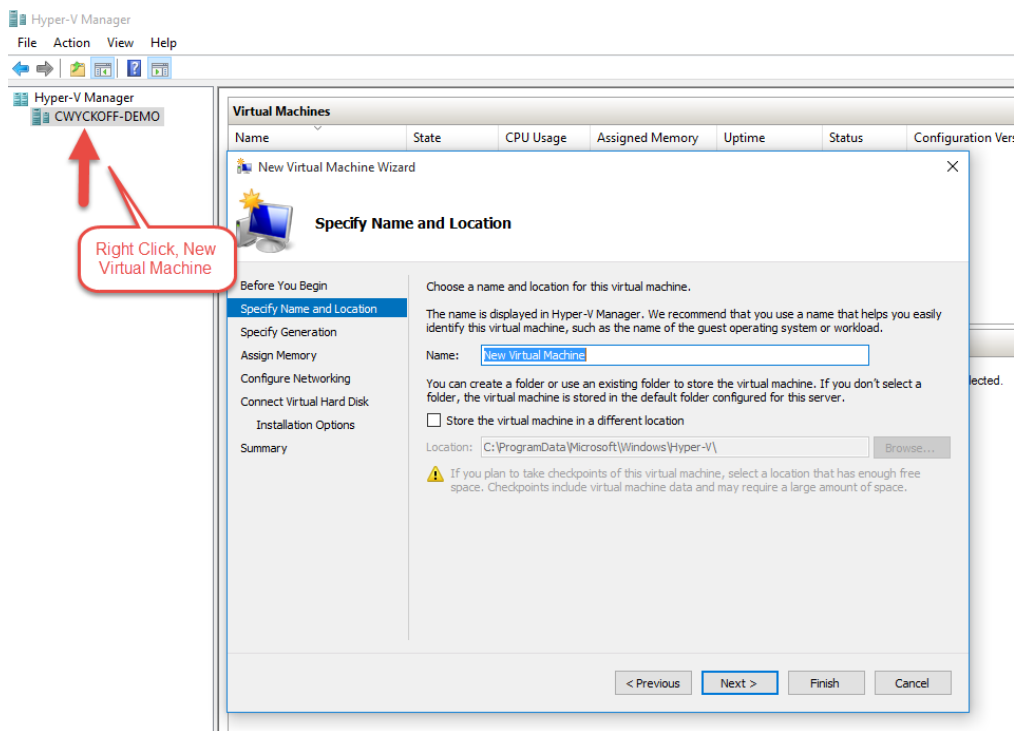


Figure 4: New Virtual Machine Wizard

After giving the VM a name, choose **Next**. The next step in the wizard is to choose the VM Generation, it is recommended to choose Generation 2 if you're deploying Windows Server 2012 or greater. When choosing a VM Generation, this is a one-time setting and the VM would require being rebuilt to alter the VM Generation.

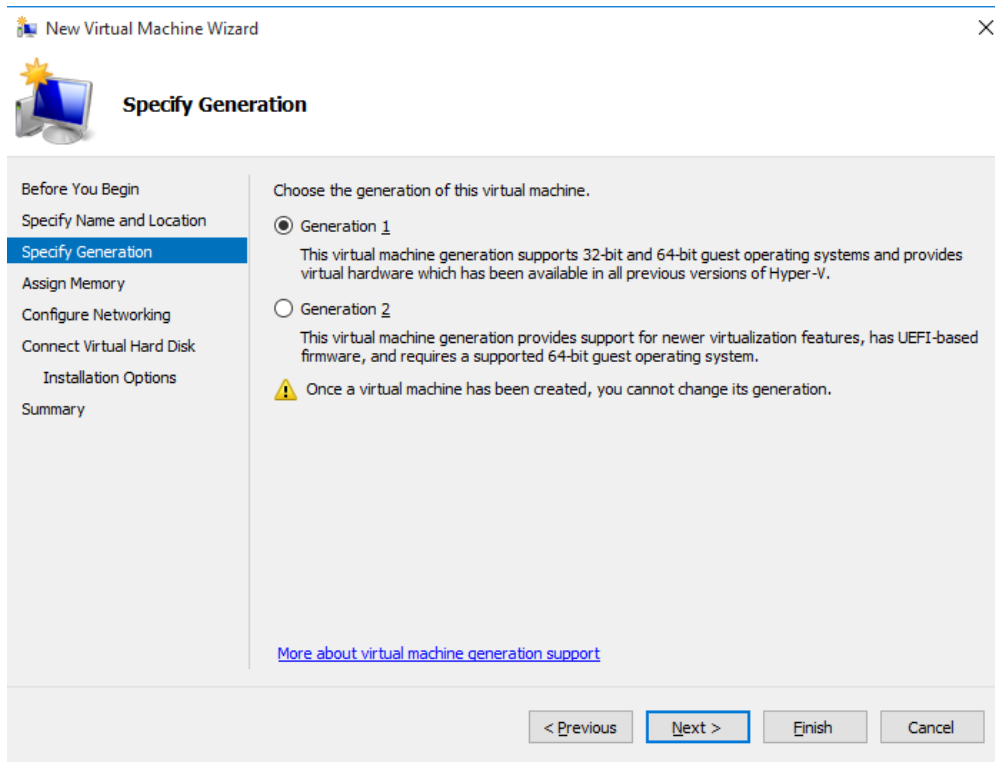


Figure 5: New Virtual Machine Wizard, VM Generation

After choosing the VM Generation appropriate for the workload being deployed, choose **Next**.

The image below displays the Assign Memory section of the New Virtual Machine Wizard. Within this section, assign the VM with Dynamic Memory and its dedicated Startup Amount or alternatively assign Static Memory to the VM. Certain applications behave differently when using Dynamic Memory, consult the documentation of the application being deployed to verify if the application supports Dynamic Memory. For example, SharePoint does NOT support being deployed on VMs with Dynamic Memory, it is also not recommended to use Dynamic Memory on SQL Server either. New to Server 2016 is the ability to alter the VM memory allocations with zero downtime; if you make an incorrect assignment here, it is easy to correct.

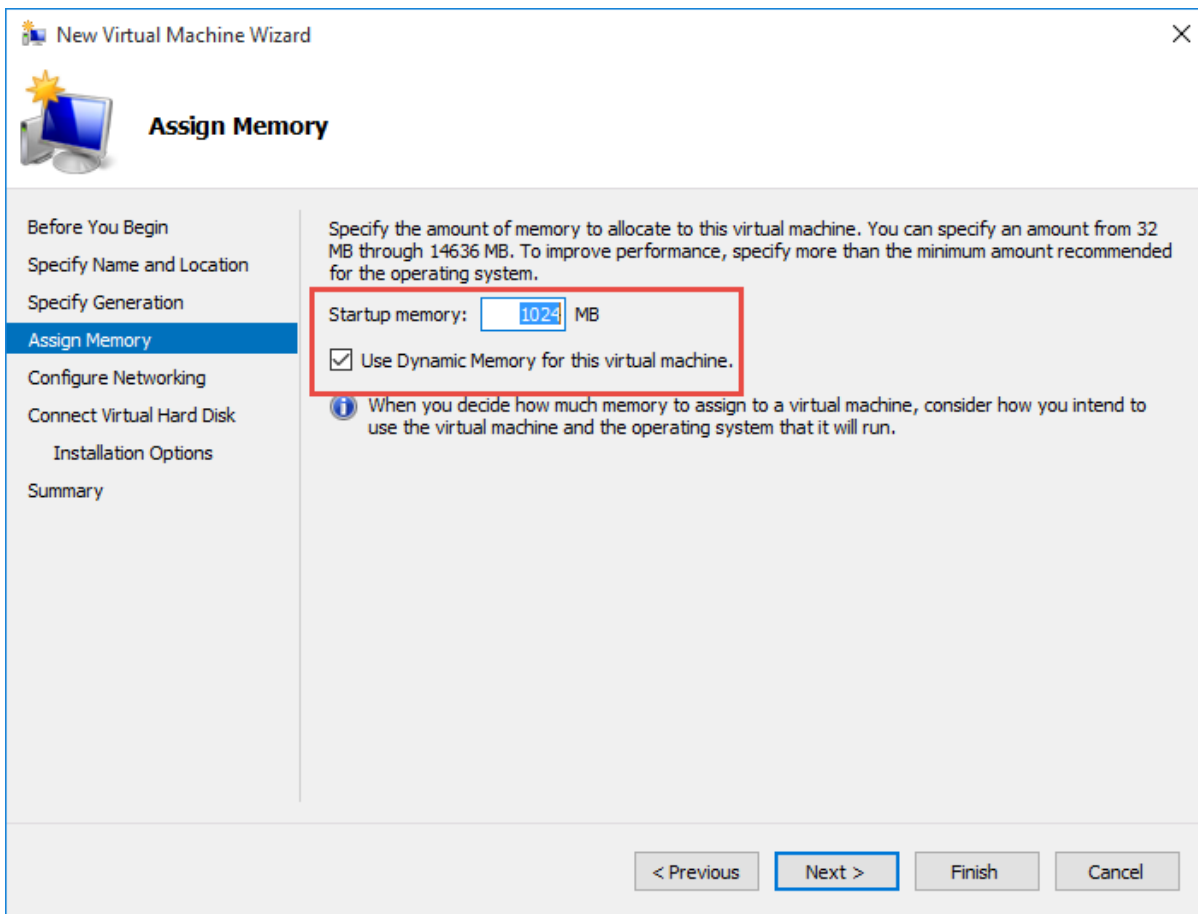


Figure 6: New Virtual Machine Wizard, Assign Memory

Choose **Next** when you've made the appropriate choice for your application and VM being deployed. Configure Networking is the next step in the New Virtual Machine Wizard. Within this section, the VM is provided its Virtual Switch. Within Hyper-V, there are four types of Virtual Switches: External, Internal, Private and NAT. The NAT vSwitch is an addition to Hyper-V 2016 that is currently only deployable through PowerShell. In the example below, Virtual Switch Ext001 is an External Switch.

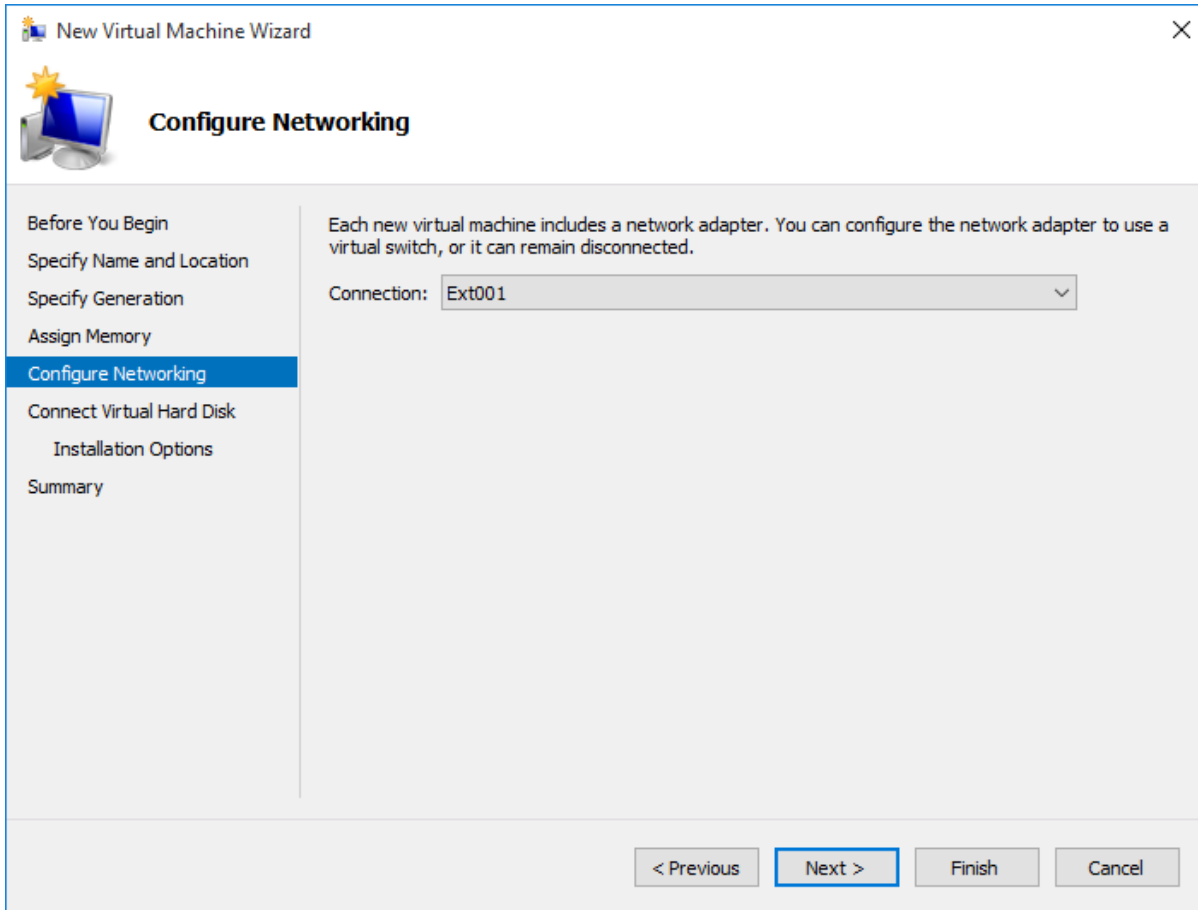


Figure 7: New Virtual Machine Wizard, Configure Networking

Choose **Next** to move forward. At this point in the New VM wizard, the Virtual Hard Disk settings are elected. Dynamically Expanding VHDX is selected by default. This option is used for most applications that are not highly transactional. Microsoft has made great advancements in the disk allocation process for dynamically expanding VHDX, however, it is my personal preference to use static disks for these types of applications. To assign a static disk, assign the virtual hard disk later. In our example below, the default is elected.

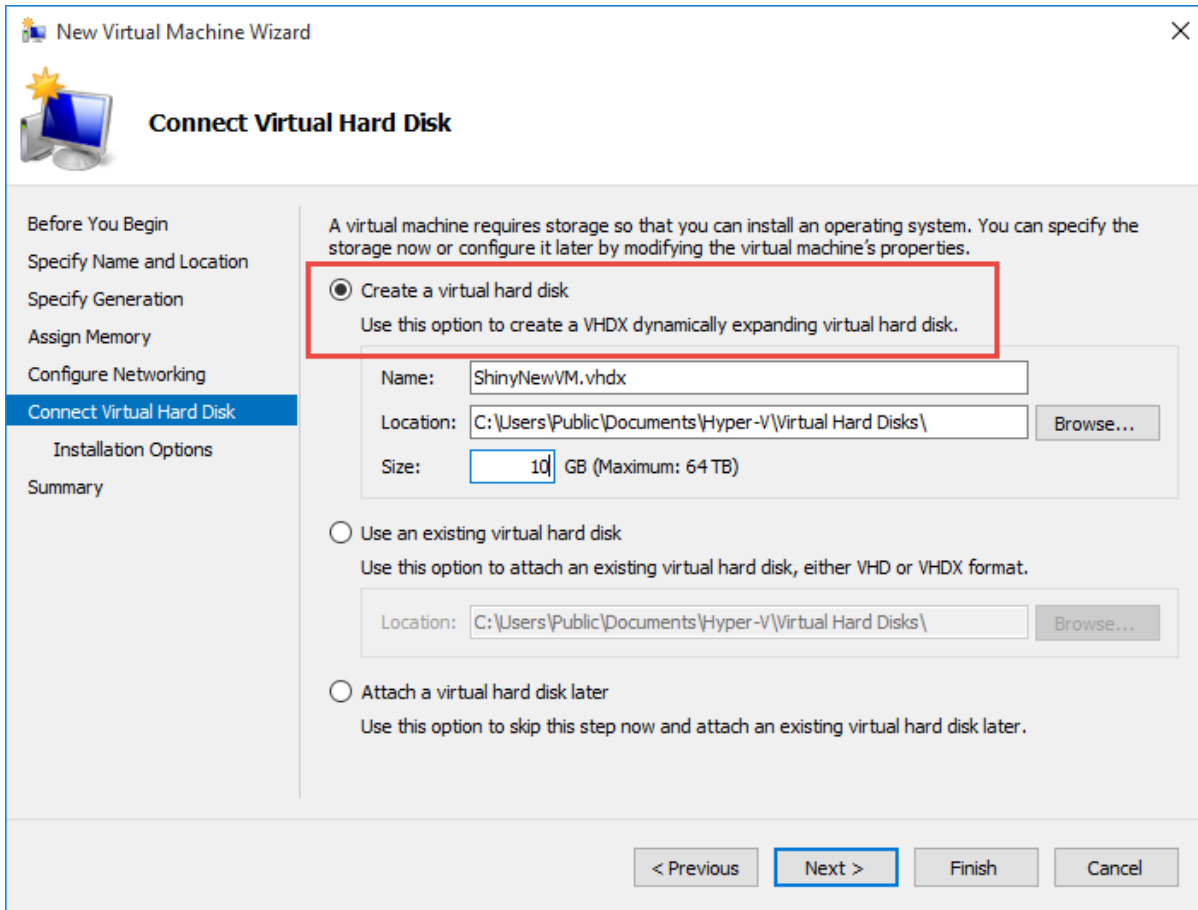


Figure 8: New Virtual Machine Wizard, Connect Virtual Hard Disk



Choose **Next**. In the Installation Options screen, browse out to the ISO that was previously downloaded and choose **Next**.

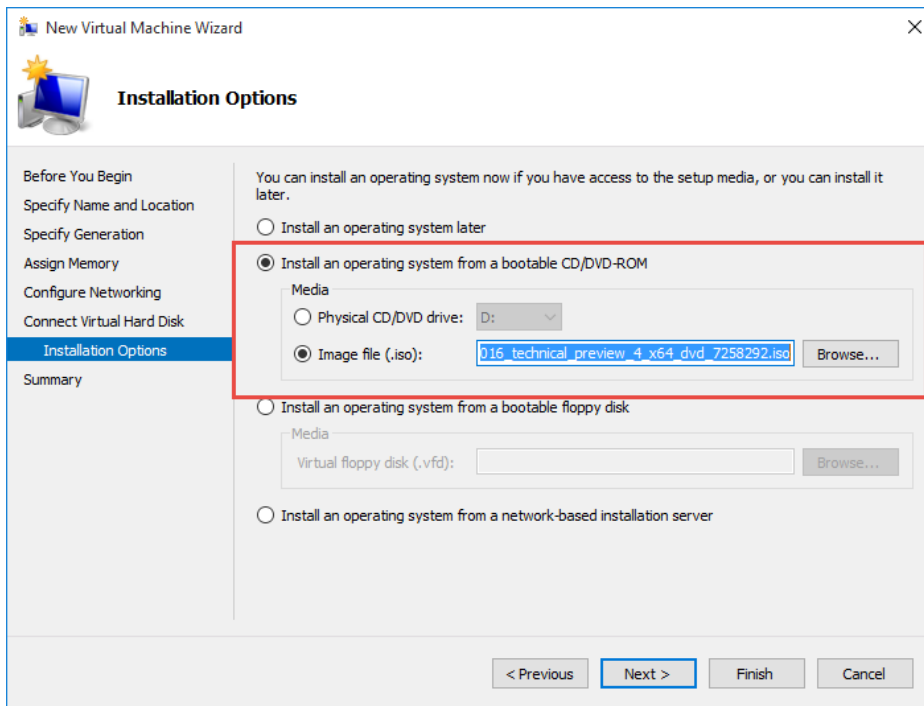


Figure 9: New Virtual Machine Wizard, Installation Options

The last step within the New Virtual Machine Wizard permits a final review of the settings. Choose previous to navigate back and make any last minute changes prior to finishing the wizard. When validation completes, choose **Finish**.

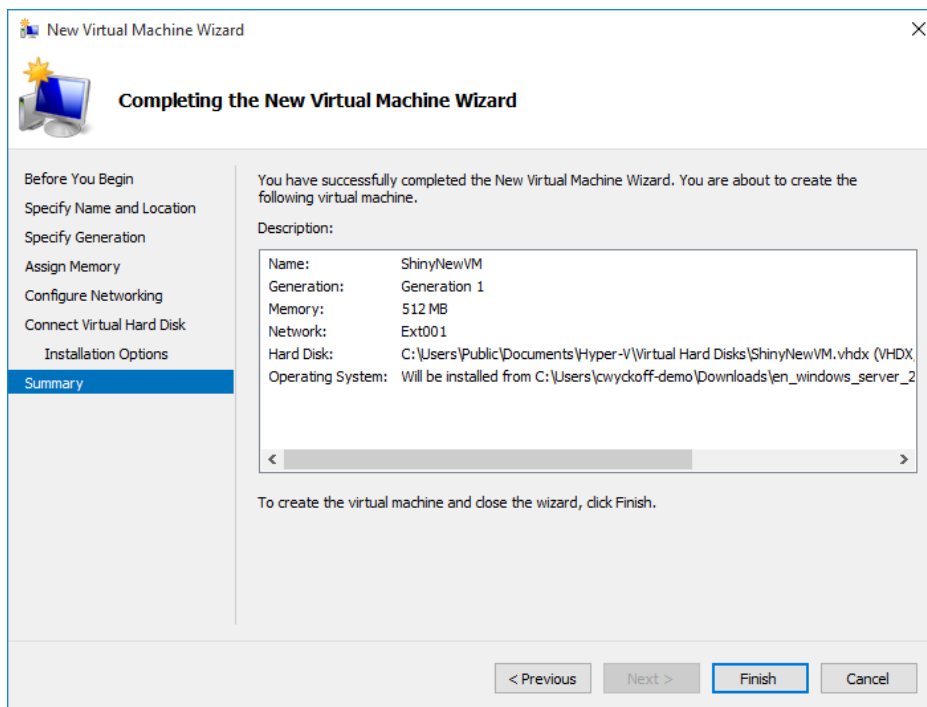


Figure 10: New Virtual Machine Wizard, Completing the New Virtual Machine Wizard

Upon completion, we are now ready to Power On the VM.

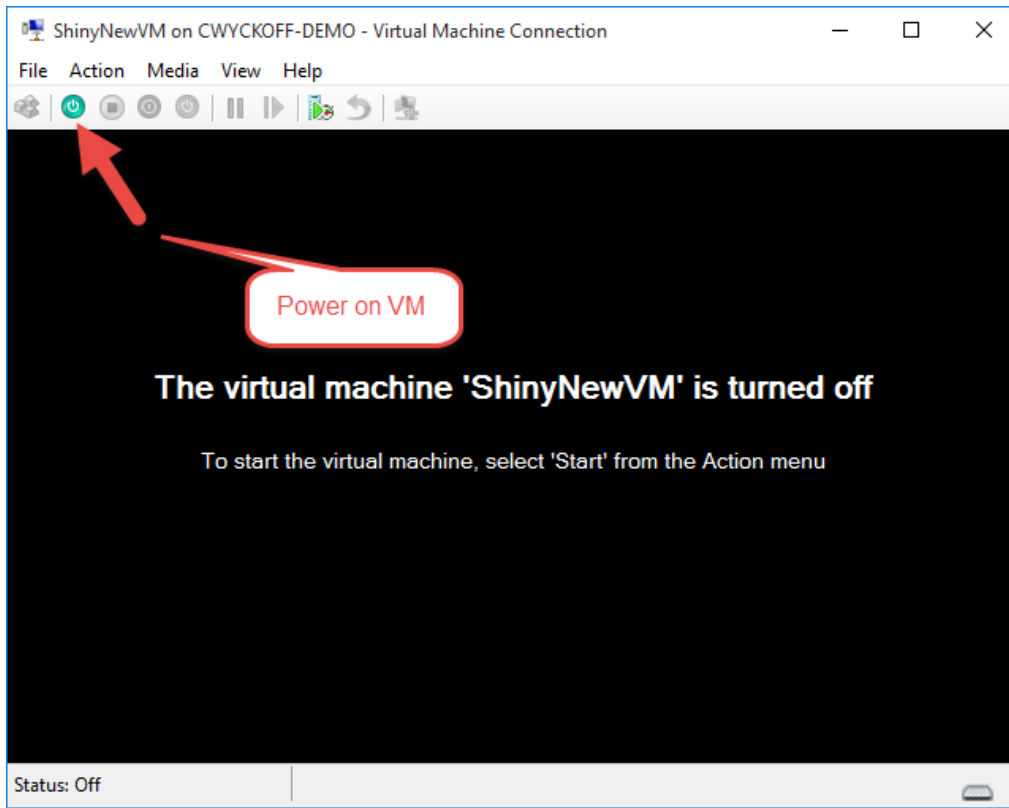


Figure 11: Virtual Machine Connection, Power On VM

When the VM is powered on for the first time, no operating system is present; the VM will boot to the ISO image that was assigned within the New Virtual Machine Wizard.

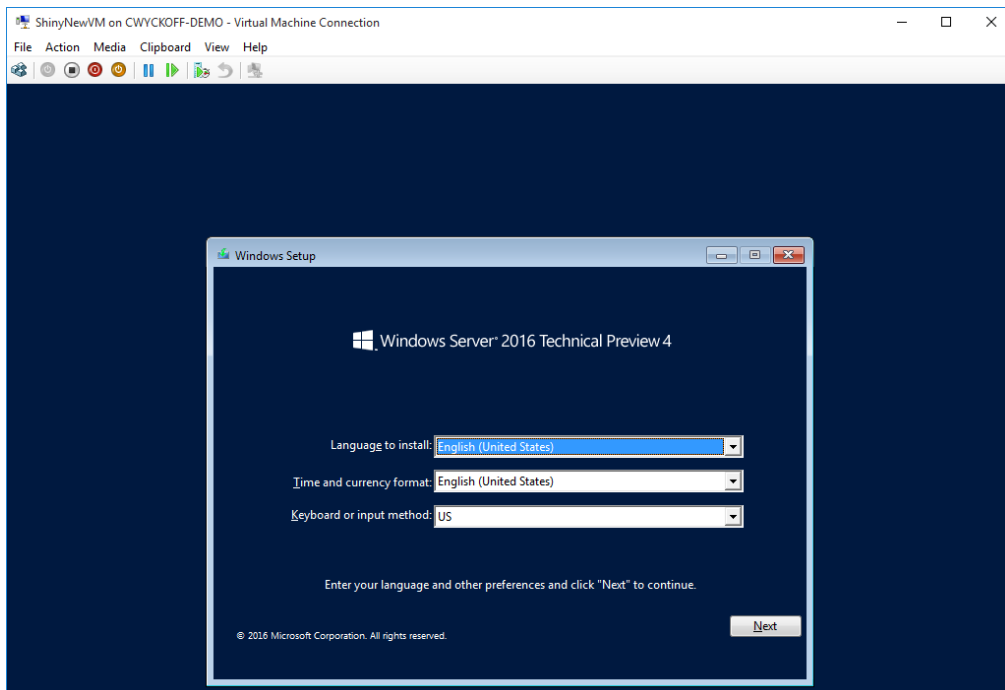


Figure 12: Windows Server 2016 TP4 Installation Wizard

With the VM powered on, the Windows Server 2016 Technical Preview 4 installation will begin. The first step in the process is language and keyboard layout selection. Upon completion, choose **Next**.

The next screen in the wizard is simple: choose **Install Now**.

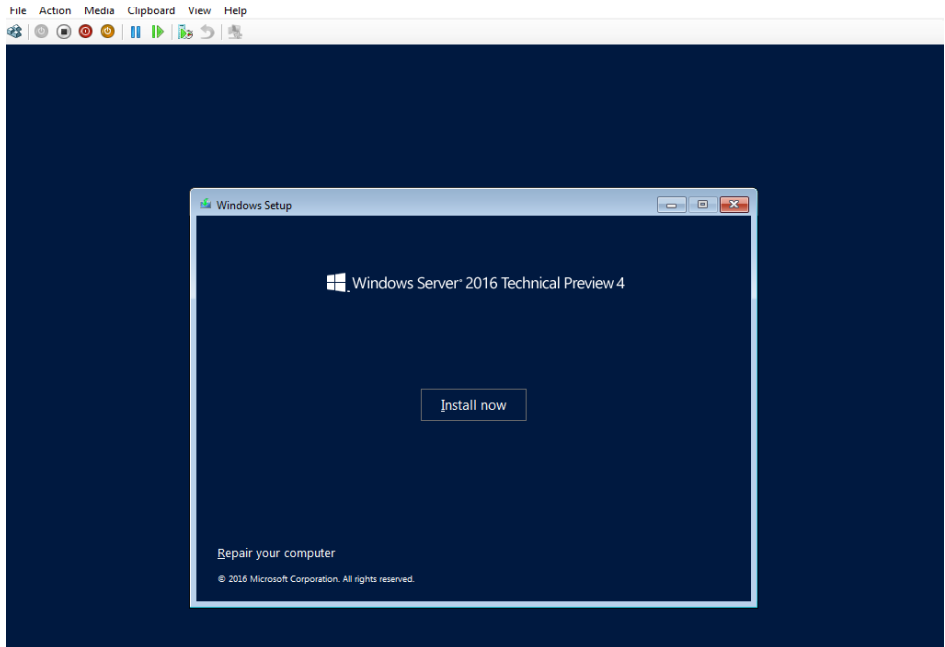


Figure 13: Windows Server 2016 TP4 Installation Wizard, Install Now

After choosing Install Now, you'll have to choose to install either the full GUI version of Windows Server or the Core version. Unless the application has specific requirements for GUI-based applications, I recommend installing the Core version of Windows Server.

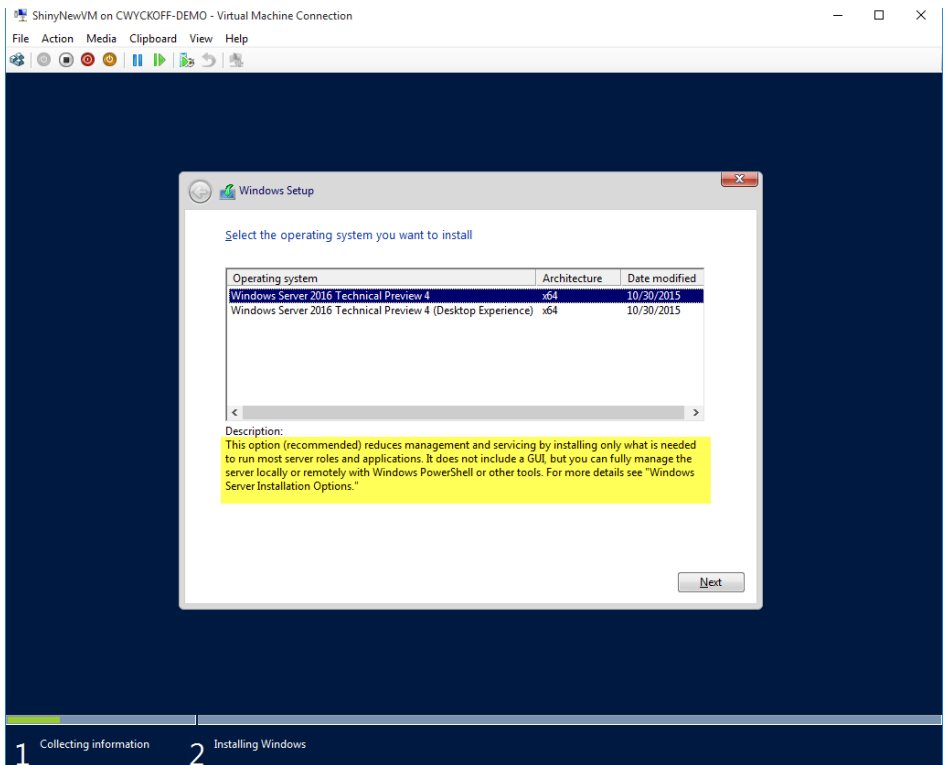


Figure 14: Windows Server 2016 TP4 Installation Wizard, Choose GUI or Core Installation

## All you need to know about Microsoft Windows Server 2016 Virtualization

For brand new Windows installations, I don't recommend upgrading from previous Windows versions. Clean Windows Server installations are considered to be an industry best practice. Choose the Custom option to setup the VHDX as appropriate and allocate the OS partition.

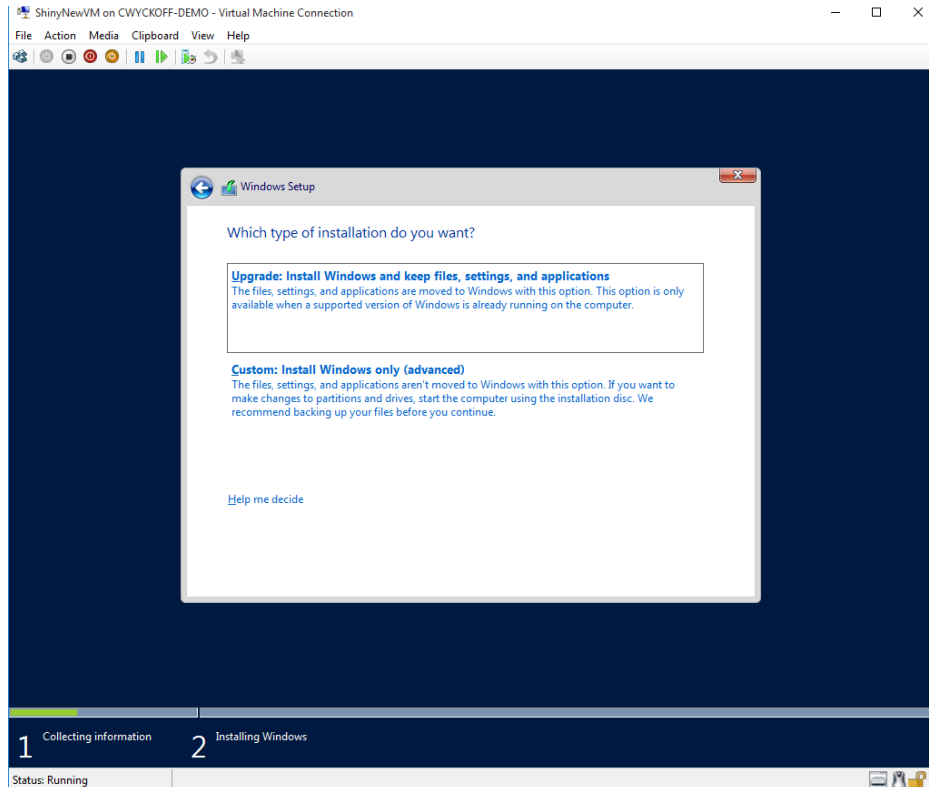


Figure 15: Windows Server 2016 TP4 Installation Wizard, Installation Options

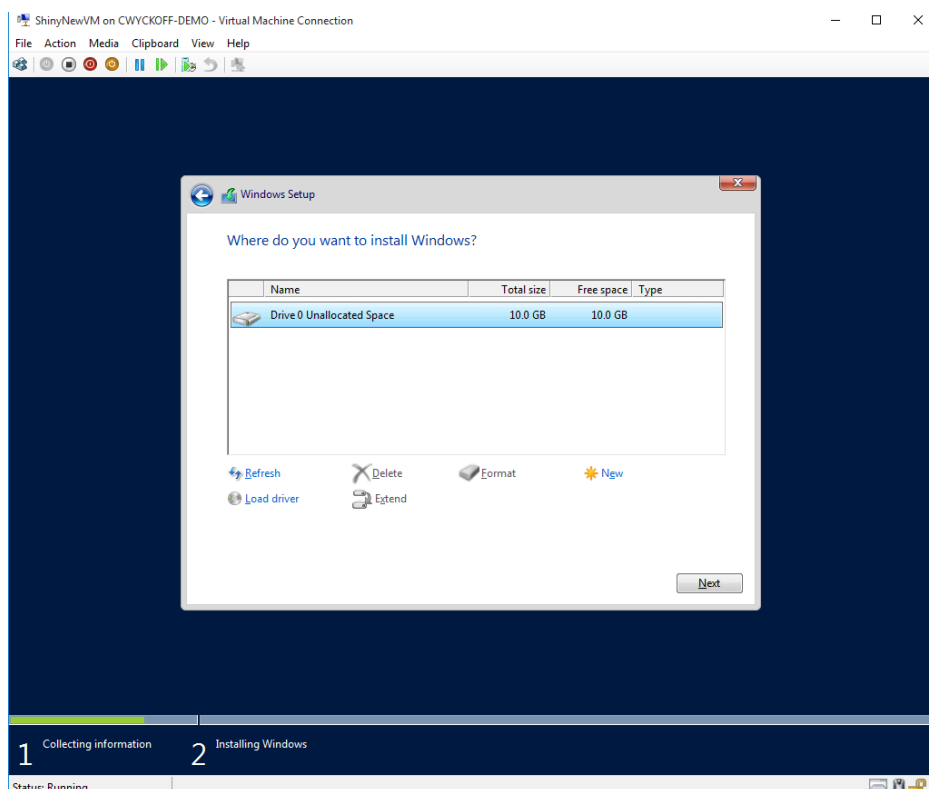


Figure 16: Windows Server 2016 TP4 Installation Wizard, Disk Allocation and Partition Setup

The last screen in the installation wizard is where the magic begins, and Windows Server 2016 TP4 is deployed.

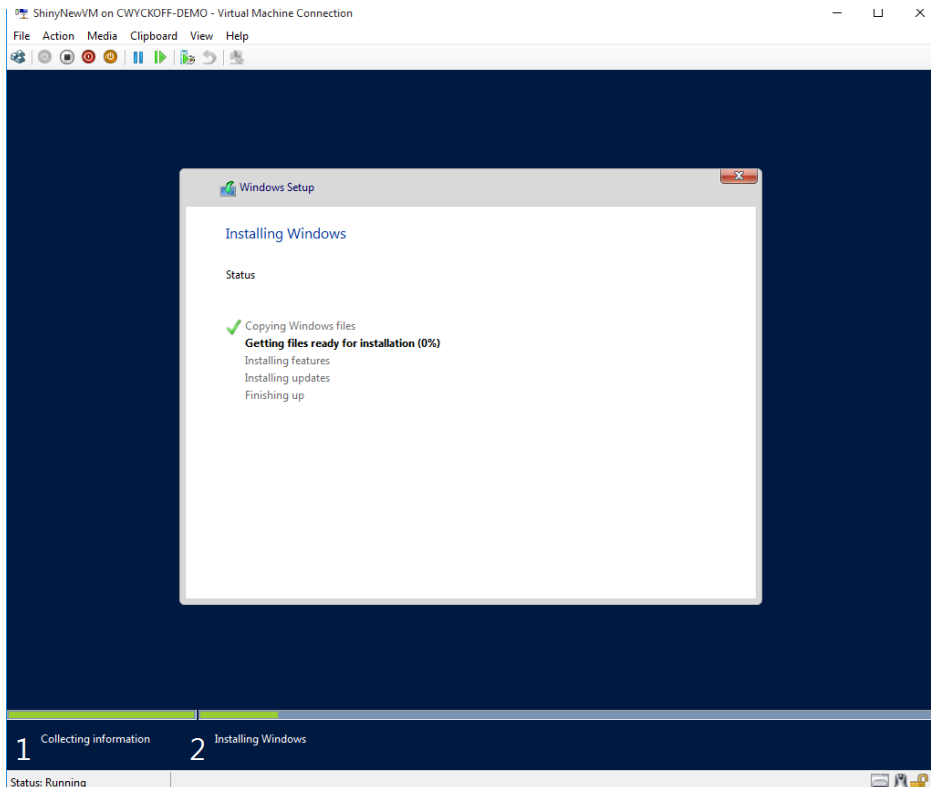


Figure 17: Windows Server 2016 TP4 Installation Wizard, Windows Installation

Upon completion, you'll be prompted to enter ctrl + alt + delete to login. The server has been deployed successfully and can now be setup for application deployments.

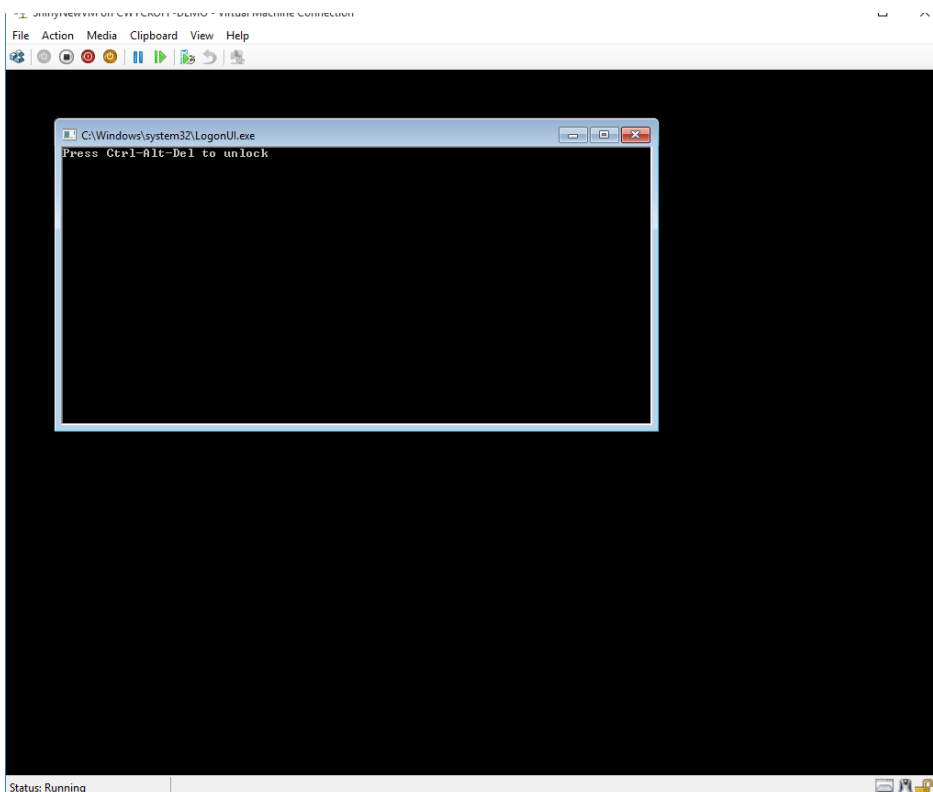


Figure 18: Press Ctrl + Alt + Del to unlock and setup Windows Server 2016 TP4 for Application Deployment.

## Create New VM Using PowerShell

The steps illustrated above outline how to deploy Windows Server 2016 using Hyper-V Manager, however, there is another option: PowerShell. Below is an example that would achieve the same exact result in a much quicker and repeatable fashion than what we did through the GUI. As an additional option, you can leverage an Unattend.xml to completely automate the entire build process.

```
$VMName = "VMName"

New-VM -Name $VMName -Generation 2 -SwitchName Ext001
-MemoryStartupBytes 2048MB -NewVHDPath "C:\Users\Public\Documents\
Hyper-V\Virtual hard disks\VMName.vhdx" -NewVHDSIZEBytes 10GB |
Set-VM -DynamicMemory -ProcessorCount 2

Add-VMdvdDrive -VMName $VMName -Path "C:\Users\cwyckoff-demo\
Downloads\en_windows_server_2016_technical_preview_4_x64_
dvd_7258292.iso"

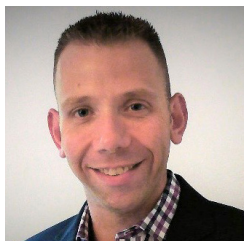
$dvd_drive = Get-VMdvdDrive -VMName $VMName

Set-VMFirmware -VMName $VMName -FirstBootDevice $dvd_drive

vmconnect.exe localhost $VMName

Start-VM -Name $VMName
```

## About the Author



**Clint Wyckoff** is a Product Strategy Evangelist at Veeam focused on all things Microsoft. He is an avid technologist and virtualization fanatic with over a decade of Enterprise Data Center Architecture experience. Clint is an energetic and engaging speaker that places a large emphasis on solving real-world IT challenges. Additionally, he has been awarded VMware vExpert designation for 2015 and is also a VMCE and MCP. Follow Clint on [Twitter @ClintWyckoff](#) or [@Veeam](#).

## External Reviewers



This eBook was externally reviewed for technical accuracy by Veeam Vanguard and Microsoft Cloud and Datacenter Management MVP, **Dave Kawula**. Dave is well-known in the community as an evangelist for Microsoft, Veeam, and 5nine technologies. Locating Dave is easy as he speaks at several conferences each year, including: TechEd (Ignite), MVPDays Community Roadshow, Tech Mentor, and VeeamON. Follow Dave on twitter [@DaveKawula](#).

## About Veeam Software

**Veeam**<sup>®</sup> recognizes the new challenges companies across the globe face in enabling the Always-On Business<sup>™</sup>, a business that must operate 24/7/365. To address this, Veeam has pioneered a new market of *Availability for the Always-On Enterprise*<sup>™</sup> by helping organizations meet recovery time and point objectives (RTPO<sup>™</sup>) of less than 15 minutes for all applications and data, through a fundamentally new kind of solution that delivers high-speed recovery, data loss avoidance, verified protection, leveraged data and complete visibility. **Veeam Availability Suite**<sup>™</sup>, which includes **Veeam Backup & Replication**<sup>™</sup>, leverages virtualization, storage, and cloud technologies that enable the modern data center to help organizations save time, mitigate risks, and dramatically reduce capital and operational costs.

Founded in 2006, Veeam currently has 37,000 ProPartners and more than 183,000 customers worldwide. Veeam's global headquarters are located in Baar, Switzerland, and the company has offices throughout the world. To learn more, visit <http://www.veeam.com>.



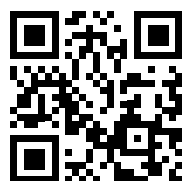
**AVAILABILITY**  
for the Always-On Enterprise™

---

# **NEW Veeam<sup>®</sup>** **Availability** **Suite<sup>™</sup> v9**

**RTPO<sup>™</sup> <15 minutes for**  
**ALL applications and data**

---



**Learn more and preview**  
**the upcoming v9 release**

**[vee.am/v9](https://vee.am/v9)**