VEEAM

All you need to know about Microsoft Windows Server 2016 Virtualization (Updated for Windows Server 2016 GA)

Clint Wyckoff

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

Contents

Introduction

Windows Server 2016 has been generally available since October 2016. This eBook has been updated to provide the latest and greatest additions within Window Server 2016 as it applies to virtualization; Hyper-V. This is the main topic we will be discussing in this eBook, Windows Server 2016 Virtualization — also known as Hyper-V 2016. Components within Hyper-V are updated or additional functionality is added with each release of Windows Server. Knowing this is important to understanding the increased functionality as well as the overall usability of Windows Server through documents such as this.

Many of the new features and functionalities do require some basic usage of PowerShell. Throughout this eBook you will find sample PowerShell scripts documented as examples allowing IT professionals to leverage Hyper-V PowerShell within their own environments. The mission 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.

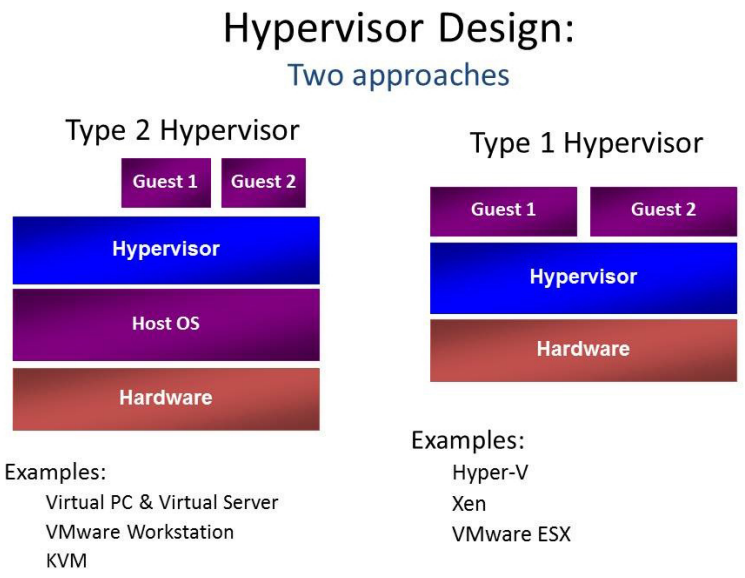

Figure 1: Type 1 vs. Type 2 Hypervisors ©Microsoft

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; a computing environment isolated 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 (MSPs) as it did not require any underlying licenses of Windows Server to run it. So, if an MSP only ran instances of Linux guest VMs, it would be free.

Edition	Features	Scalability	Virtual operating systems license
Standard	Limited	Limited	One Windows Operating System
Enterprise	Unlimited	Unlimited	Four Windows Operating System
Datacenter	Unlimited	Unlimited	Unlimited Windows Operating System
Hyper-V Server 2008 and 2008 R2	Limited	Limited	Zero 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 one TB of virtual RAM. Hyper-V hosts could contain 320 logical processors, four 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 two 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 is included with the single license and Datacenter supports Automatic Virtual Machine Activation on the host.

Edition	Features	Scalability	Virtual operating systems
Standard	Unlimited	Unlimited	Two Windows OS
Enterprise	Unlimited	Unlimited	Unlimited Windows OS
Hyper-V Server 2012 & 2012 R2	Unlimited	Unlimited	Zero 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 the product based on their own internal triumphs and challenges. Microsoft has made much of this new learning 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 key knowledge of Hyper-V and the ability to effectively and efficiently support a production Hyper-V environment. This eBook has been updated and now takes the General Availability (GA) of Windows Server 2016 into consideration and includes some of the findings that have been gained since it became officially available.

I hope you enjoy this eBook and the learnings it will provide!

Nano server

In previous versions (Windows Server 2008, 2008R2, 2012, 2012R2) when deploying the operating system, you had to choose which version and which mode to deploy. The options included Windows Server with a Graphical User Interface (GUI) or Windows 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.

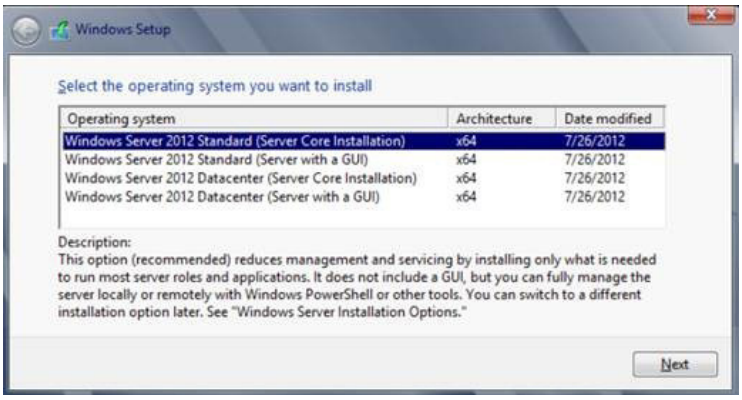

Figure 1: Windows Server 2012 Installation Options

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 is being updated to serve only Container-based environments. Previously, Nano Server was able to operate as a Hyper-V host, DNS Server and many other infrastructure related services. On June 15, 2017 Microsoft announced that the direction of Nano server is being updated to address one key scenario in your environment:

- Container Image host

Admittedly, Nano Server was not the easiest operating system to get running, as it required many steps to inject the packages required as Nano did not include many of the system binaries that are present in the GUI versions of Windows Server. Not to mention Nano Server was extremely difficult to get running on physical, bare-metal hardware. For infrastructure-related services, like Hyper-V for instance, the recommended deployment type for Windows Server is Server Core.

Summary

When announced, Nano Server presented many opportunities within the modern data center and it was thought that the possibilities would be endless. However, Microsoft has decided to remove the infrastructure-related capabilities and focus Nano Server on Containers –- making it the very best container image possible. For infrastructure-related roles and services, Microsoft recommends deploying Windows Server Core. Be sure to read the official Microsoft blog as well as others for more detailed information on Nano Server, Server Core and their intended use cases.

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 ensure that we are all on the same page it seems like such a long while ago 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 reap.

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 benefit 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 scale stand-up and tear down environments on-demand and at a large provides much needed agility to the Developer Operations (DEVOPS) world. The 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 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 on 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 and run as an isolated process within their parent OS. In Windows Server 2016, Microsoft has embedded virtualization technologies within the Windows kernel that provide 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.

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.

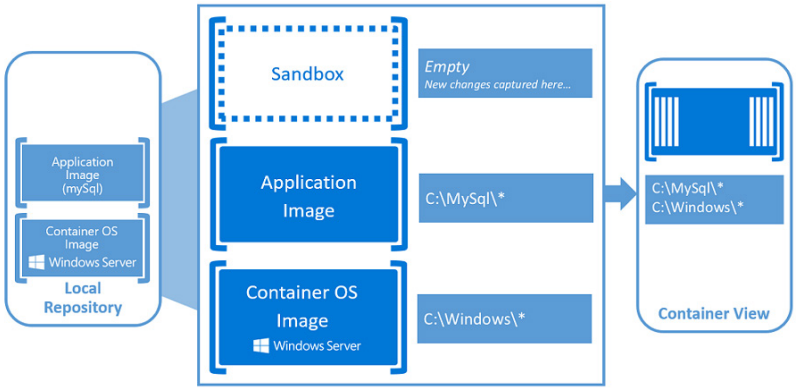

Figure 4: Container architecture within Windows Server 2016 ©Microsoft

***Note:** If you are running Window Server 2016 TP5, please update to Windows Server 2016 as critical updates are needed for the Windows Container to function.*

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 and 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 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 and 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 the 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 the changes within the Container are captured in a sandbox layer.
- **Dockerfile**
  - The dockerfile is used to automate the creation of container images.

Microsoft provides a fully documented step-by-step document that IT administrators or developers can utilize to begin testing or using Windows Server Containers and Docker within the environment.

Summary

The IT industry is bridging the gaps between development and IT operations through DEVOPS. DEVOPS and the management of the development process by using either Windows Containers or Dockers 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 perspective, the changes added to Windows Server 2016 are 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. The following chapter will walk you through many of the newest additions to Microsoft's virtualization product, Hyper-V.

Resilient File System (ReFS) v3.1

While a file system is not directly a feature of Hyper-V; Hyper-V is one of the biggest consumers of storage and the preferred file system for Hyper-V is Resilient File System (ReFS). ReFS was introduced into the Windows Server product in Windows Server 2012 with v1.0 and has been consistently improved upon as Microsoft evolves and updates their OS versions. If you have done any testing with technical preview versions of Windows Server, it will be obvious that the ReFS version has been increased from 3.0 to 3.1 within the GA version of Windows Server 2016.

ReFS is absolutely the file system of the future for Microsoft, just like NTFS was back in the Windows NT 3.1 days. ReFS was designed to overcome many of the shortcomings that NTFS had present. Many of the key characteristics of ReFS focus around making sure that the data stored on the file system is protected from many of the common errors that can cause data loss. For instance, the requirement for chkdsk is completely removed due to the automatic integrity checking and data "scrubbing" (data integrity scanning) that the file system inherently can perform on either a per-volume, per-directory or per-file basis. If corruption is detected, when configured with Storage Spaces or Storage Spaces Direct, ReFS will automatically attempt to recover from the errors without any interaction from the end-user. The corrupt data is rebuilt from either the mirrored copy or the party bit of data that is on the other hard-disks within the cluster. ReFS employs an allocation-on-write update methodology for its own files system metadata and all the metadata has 64-bit checksums which are stored independently.

Another key characteristic of ReFS is the fact that with File Integrity enabled, the file system acts more like a log-structured file system, gathering small random blocks of writes into large sequential writes for better efficiency. For example, if a server has applications that writes its data in 4k, 8k and 16k sizes with File Integrity enabled, ReFS will automatically group the writes together into larger sequential writes — resulting in large 32k and / or 64k writes making the file system much more efficient at handling larger sets of data, thus making it the optimal file system for virtualization and even backup targets.

ReFS supports many of the same features that its counterpart NTFS does, however there are several key features that also make ReFS the optimal file system for virtualization, for example Data Integrity Streams. Within Windows Server 2016 ReFS v3.1 now supports the cloning of like blocks of data within a file via the Block Clone API. This API allows the manipulation of pointer files to reference existing blocks of data on disk without ever physically writing out a new file. Software Companies, like Veaam®, can leverage these public facing APIs to interact with in many different ways.

ReFS is the file system of the future for Microsoft, so expect lots of development around this area and more and more products within the Microsoft technology stack to add full-support for the advanced capabilities it provides.

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 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 to changes. For example, it has been well documented that sometimes hotfixes and patches can cause issues with productions systems. Once discovered, organizations would simply revert a VM from a previous state to fix the issue. This was not supported or recommend 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 has 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.

When Hyper-V 2016 is deployed on the Resilient File System (ReFS) v3.1 within Windows Server 2016 the Checkpoint process can leverage the Block Clone API. Due to the nature of how snapshots were conducted within Server 2012R2 for instance, Microsoft never supported creating Checkpoints on any production system. ReFS makes this much more efficient as the existing blocks of data are never physically moved around; they're simply referenced via the metadata that ReFS employs.

Let's take a look at this problem a bit deeper and use SQL Server as an example. With Standard Windows Server Checkpoints, all the disk and memory state is captured, including 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.

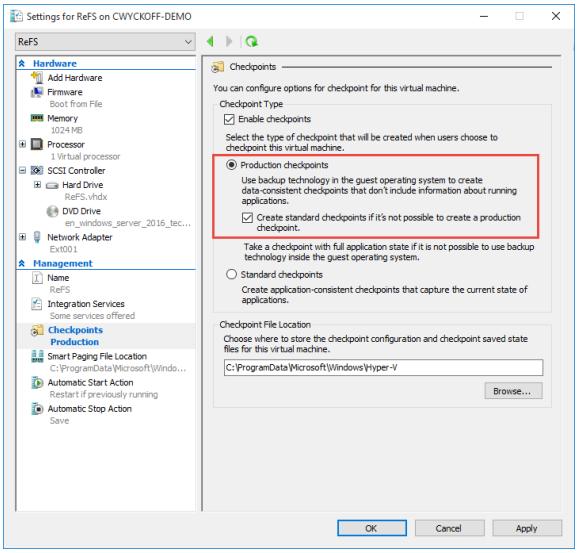

Figure 5: Checkpoint settings of an individual virtual machine and Checkpoint file location

Figure 5 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 6, 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
```

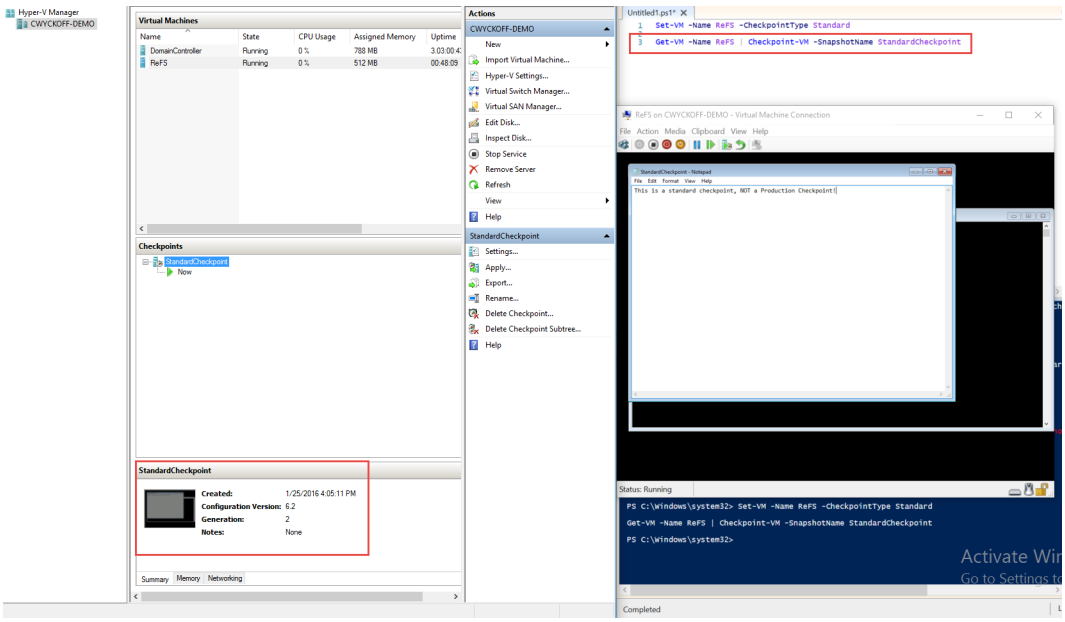

Figure 6: Standard Checkpoint using PowerShell

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 7, upon applying the checkpoint StandardCheckpoint, our VM comes back as it was before.

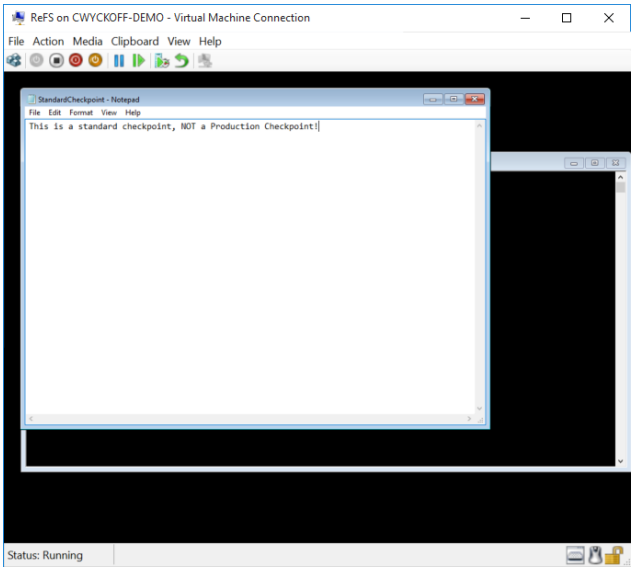

Figure 7: Standard checkpoint revert — memory saved state

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 changes the checkpoint type to production and leaves 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 8).

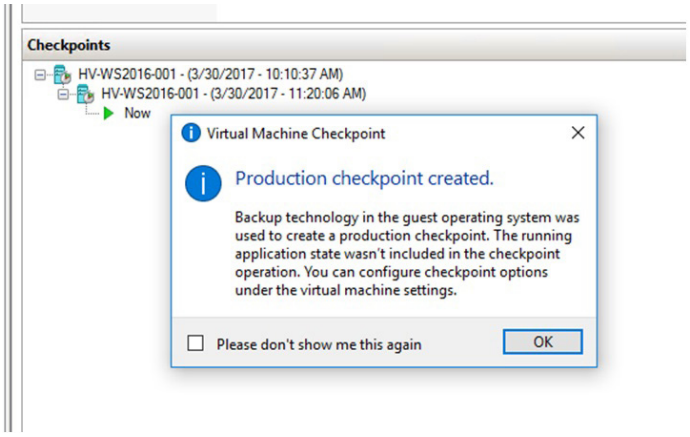

Figure 8: Production Checkpoint confirmation message

```
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 that 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 9, below, shows that whenever this production Checkpoint example is applied, our VM is brought up in a clean state. This means the guest operating system feels and looks as though it was shut down properly. Keep in mind we are still within Technical Preview and after applying a production type snapshot you MUST manually power the VM back on.

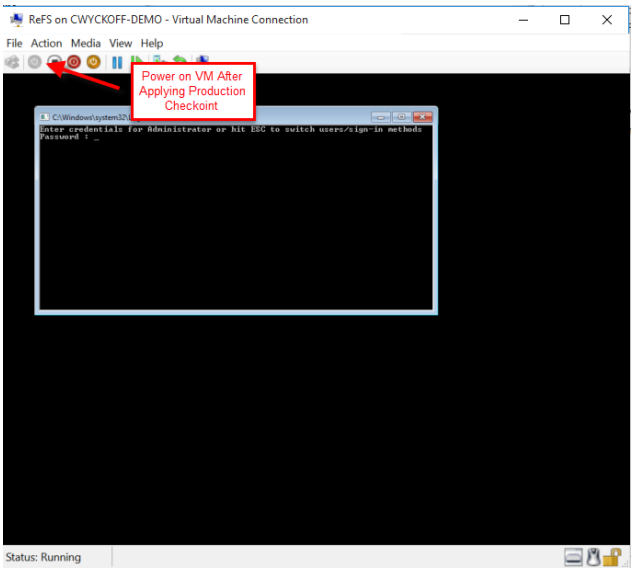

Figure 9: Post production Checkpoint — power on VM!

Making sure that applications and workloads are recoverable when things go bump in the night is very important. Modern backup solutions leverage snapshots and checkpoints to create point-in -time restore points. In Hyper-V 2016 these backup products leverage recovery Checkpoints. Recovery Checkpoints are application consistent exactly like production Checkpoints — the main difference is that recovery Checkpoints are initiated by the backup software. In image 10 below we can see that the backup software utilized the recovery Checkpoint.

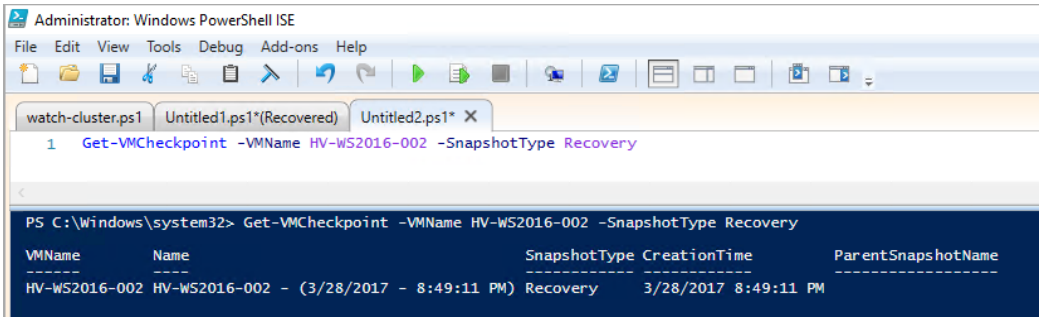

Image 10: Illustrates the recovery Checkpoint being initiated by backup software providers

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. The guest VM must also be running Windows 10 or Windows Server 2016 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 two 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
```

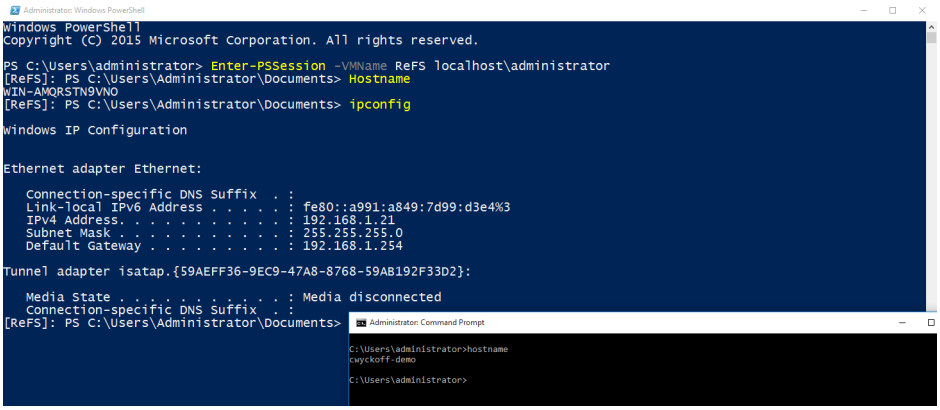

Figure 11: PowerShell Direct connecting using -VMName

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 }
```

```
$Credential = Get-Credential

Invoke-Command -VMName ReFS -Credential $Credential -ScriptBlock { Get-Process }
```

```
PS C:\Windows\system32> $Credential = Get-Credential

Invoke-Command -VMName ReFS -Credential $Credential -ScriptBlock { Get-Process }
cmdlet Get-Credential at command pipeline position 1
Supply values for the following parameters:

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName                                                  PSComputerName
-------  ------    -----      ----- -----   ------     -- -----------                                                  --------------
    101       8     4092       7060 ...19       0.02    644 conhost                                                      ReFS
    116       9     1464       6984 ...34       0.03    920 conhost                                                      ReFS
    117       9     1472       7044 ...34       0.03   1316 conhost                                                      ReFS
    190      10     1652       3320 ...98       0.08    312 csrss                                                        ReFS
    123       9     1204       3500 ...96       0.05    376 csrss                                                        ReFS
    127       8     1160       3948 ...93       0.05   1904 csrss                                                        ReFS
      0       0        0          4   0                   0 Idle                                                         ReFS
    233      14     2472      13316 ...09       0.08    692 LogonUI                                                      ReFS
    233      14     2436      13340 ...09       0.02   2032 LogonUI                                                      ReFS
    685      20     3604       9584 ...93       0.23    492 lsass                                                        ReFS
    189      12     2640       8864 ...96       0.05   2732 msdtc                                                        ReFS
    369      54    77216      34008 ...09       2.09   1328 MsMpEng                                                      ReFS
    428      26    47576      61220 ...38       0.45   1372 powershell                                                   ReFS
    197       9     1944       5284 ...69       0.14    484 services                                                     ReFS
     52       2      352       1176 ...58       0.06    228 smss                                                         ReFS
    407      12     2972       5828 ...92       0.23    560 svchost                                                      ReFS
    546      33     6380      14916 ...61       0.14    580 svchost                                                      ReFS
    288      13     2212       4732 ...84       0.06    608 svchost                                                      ReFS
    245      12     2508       4592 ...02       0.16    720 svchost                                                      ReFS
    347      27     9504      12480 ...56       0.09    740 svchost                                                      ReFS
    532      18    27376      10164 ...10       0.19    832 svchost                                                      ReFS
    337      15     7548       9356 ...02       0.14    884 svchost                                                      ReFS
   1069      35    12192      19640 ...14       1.08    928 svchost                                                      ReFS
    296      17     4316       7812 ...01       0.06    940 svchost                                                      ReFS
    373      31     9816      12952 ...23       0.16   1136 svchost                                                      ReFS
    207      14     3864      11188 ...34       0.09   1296 svchost                                                      ReFS
    718       0      120        124     3       1.86      4 System                                                       ReFS
    117       8     1292       3860 ...86       0.05    256 VSSVC                                                        ReFS
     90       8      796       2184 ...75       0.05    412 wininit                                                      ReFS
    143       7     1460       6996 ...04       0.03    420 winlogon                                                     ReFS
    141       7     1288       5376 ...02       0.03   1940 winlogon                                                     ReFS


PS C:\Windows\system32>
```

Figure 12: 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 a 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 }
```

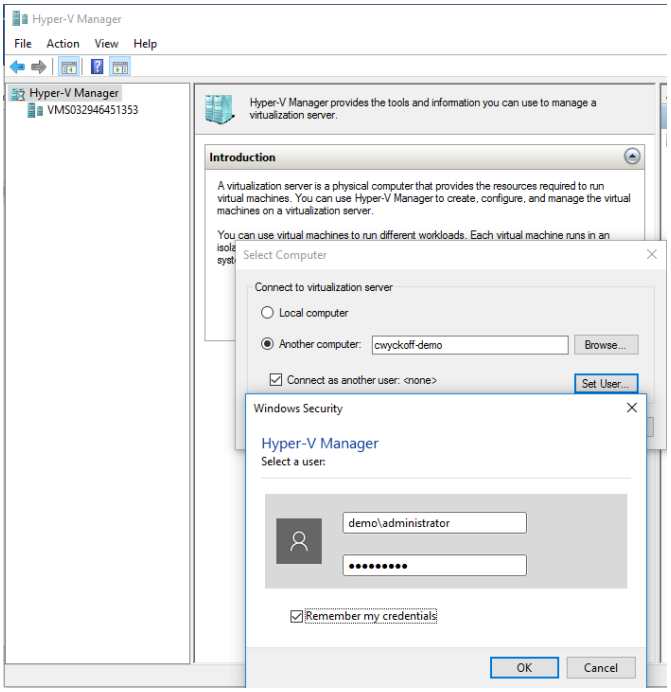

Figure 13: Remote connection to Hyper-V node with alternate credentials

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 *
```

Hyper-V 2016 has introduced the concept of production Checkpoints, these are easy for an IT administrator to execute via PowerShell, PowerShell Direct and remotely via application requests – ie. backup products. Backup products use recovery Checkpoints as these are application consistent (for supported OS versions) just like production Checkpoints. However, these are made by the VSS requestor making them machine generated. In Windows 2016 the main difference is that these Checkpoints are completely separated from the Host VSS snapshot. Figure 14 below illustrates how PowerShell can be leveraged to view the Checkpoint type.

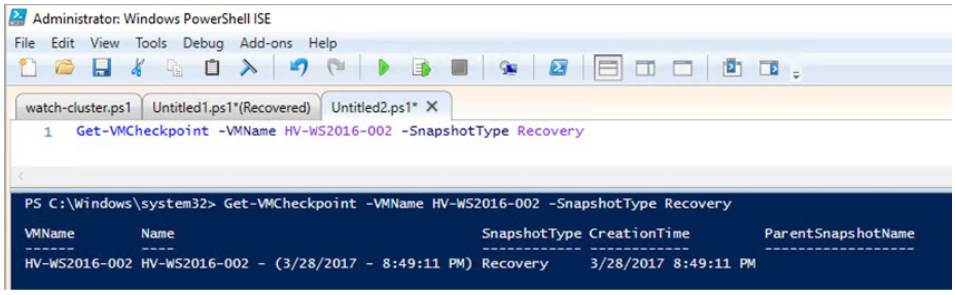

Figure 14 Executing PowerShell to illustrate the -SnapshotType during backup window checkpoint

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 one VMs, dynamic hard drives suffered around 25% performance loss over fixed disks. Generation two VMs have reduced this performance penalty drastically, making it feasible to provision dynamic virtual hard disks when running generation two 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 an 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	Measure-Command { New-VHD -Path C:\Temp\NTFS.vhdx -SizeBytes 30GB -Fixed } | fl TotalSeconds	17.0601 seconds
ReFS	Measure-Command { New-VHD -Path C:\Temp\REFS.vhdx -SizeBytes 30GB -Fixed } | fl TotalSeconds	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 except for Guest Services. The diagram below displays the VM Settings. To navigate to the VM Settings, right click on the VM and choosing Settings, then Integration Services under the Management area.

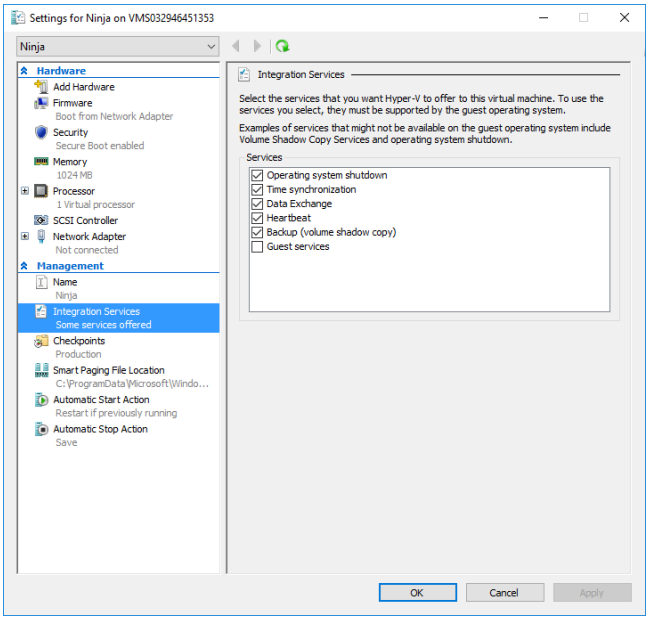

Figure 15: Hyper-V integration settings

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 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 saved 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. On 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.

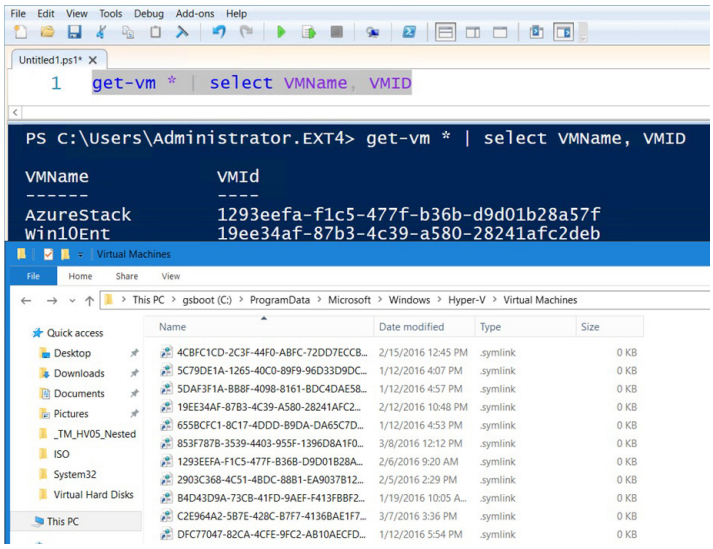

Figure 16: VMId Output from PowerShell and VM configuration files new in Hyper-V 2016

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 and 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 two 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 Hoye 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 have been significant increases in the amount of security breaches that have stemmed from hackers, malware and phishing attempts. In the digital era of today, 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 data center.

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?

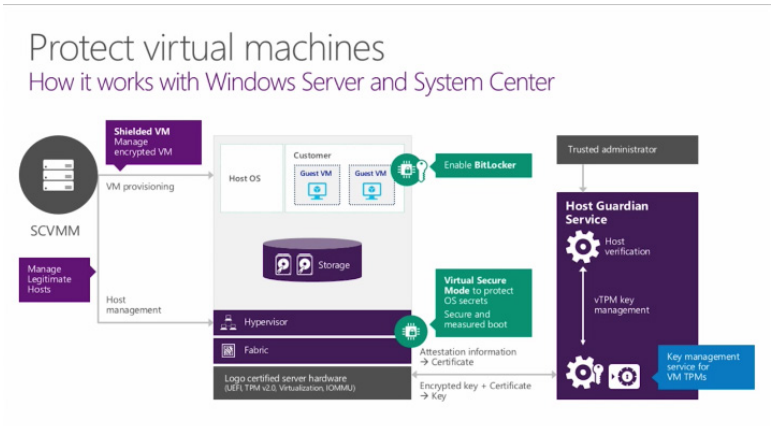

Figure 17: How security works with Windows Server and System Center Microsoft

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 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 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 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 two) 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 vTPM inside the VM along with the presence of either Bitlocker or a third-party full-disk encryption solution to ensure that only the designated owners can run the VM. It's important to understand that a physical TPM is *NOT* required to utilize vTPM inside the VM with Windows Server 2016.

Shielded VMs and vTPMs 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 VM into a shielded VM, making it easy to move from traditional mode to shielded. Meanwhile, 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, can utilize this.

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 are the ability to register each Hyper-V host's TPM and establish baseline configuration item policies for each node.

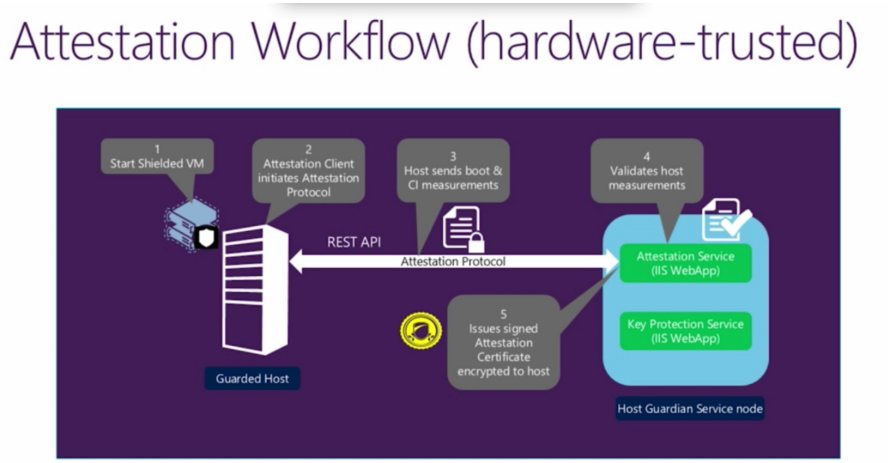

Figure 18: Attestation workflow for hardware trusted Host Guardian Service ©Microsoft

Mode 2 is Active Directory-based (admin-trusted mode) and is easier to set up. 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.

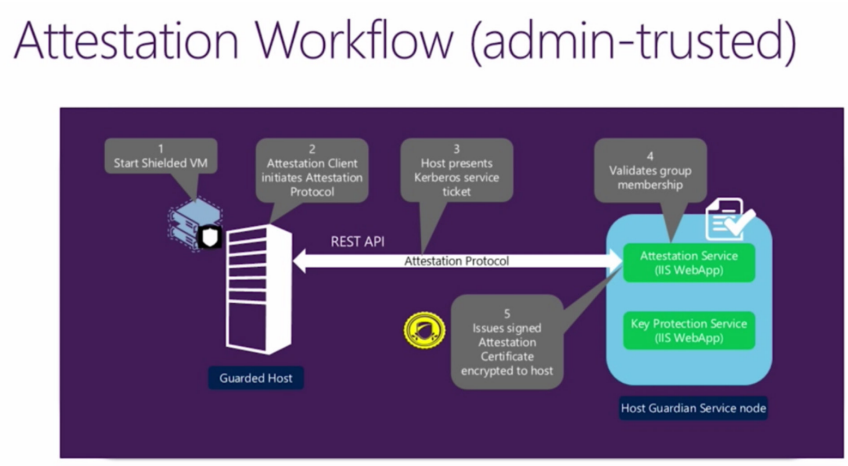

Figure 19: Attestation workflow for admin trusted Host Guardian Service ©Microsoft

A typical deployment scenario would include a separate Active Directory Forest for the Host Guardian Services 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.

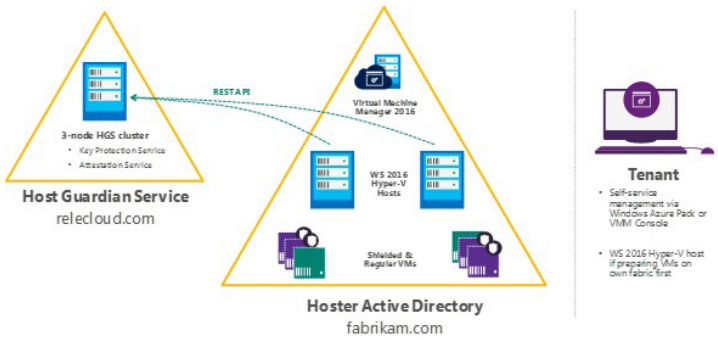

Figure 20: Logical representation of the deployment topology

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 (Active Directory)
Step-by-Step Guide: Deploy Shielded VMs Using TPM-trusted Attestation

Summary

Windows Server 2016 and specifically the virtualization pieces, are making big 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 ensuring that the workloads being run off-premises are secure.

Performance isolation techniques

One of the great benefits of the virtualized data center 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 data center, 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 sets 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 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. 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 four Normalized IOPs.

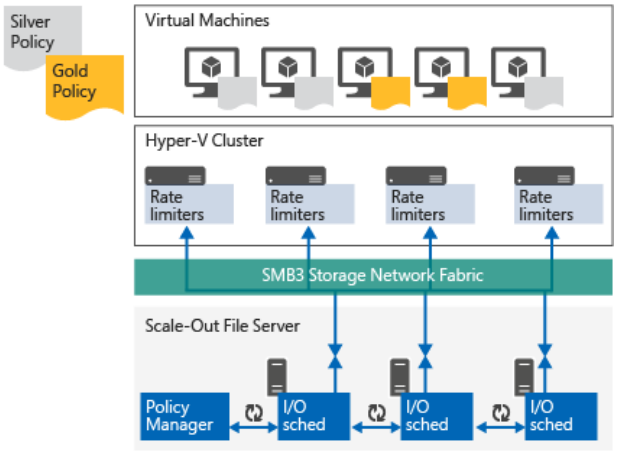

Figure 21: Using Storage QoS on a Scale-Out File Server ®Microsoft

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 when *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 similarly 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.

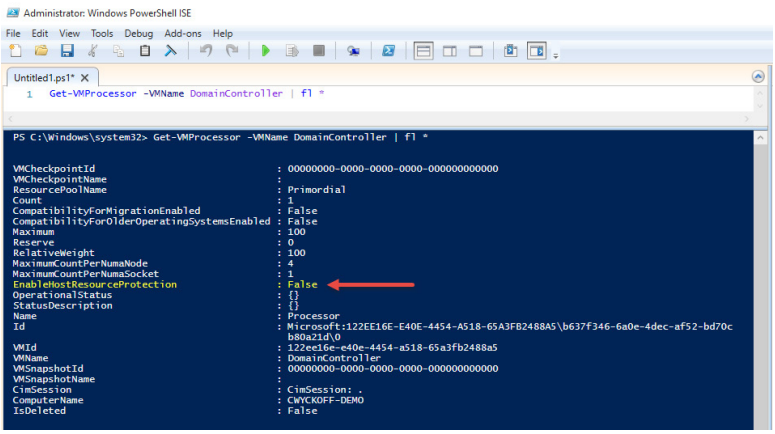

Figure 20: Get-VMProcessor cmdlet used to view status of HostResourceProtection

```
Set-VMProcessor -VMName $VM -EnableHostResourceProtection 1
```

Above is the PowerShell required to enable host resource protection on Windows Server 2016. You can run the Get-VMProcessor command to review and to ensure it has been applied correctly.

Server 2016 networking

A fundamental part of any cloud capable deployment is networking and the Software-Defined Data Center (SDDC) is no different. Windows Server 2016 provides new and improved Software Defined Networking (SDN) technologies to help you move to a fully realized SDDC solution for your organization.

When you manage networks as a software defined resource, you can describe an application's infrastructure requirements one time and then choose where the application runs — on premises or in the cloud. These consistently applied requirements and specifications mean that your applications are now easier to scale and can be seamlessly run anywhere with equal confidence in security, performance, quality of service and Availability.

There is a great guide that has been written by James McIllece at Microsoft called "The Windows Server 2016 NIC and Switch Embedded Teaming User Guide." This was used as a guide for this section and can be downloaded for your reference here: https://gallery.technet.microsoft.com/windows-server-2016-839cb607?redir=0

Windows Server 2016 network architecture

Microsoft's latest Windows Server release provides an alternative NIC Teaming solution for environments where Hyper-V is installed and the Software Defined Networking stack (SDN-stack) is being used. This solution integrates the teaming logic into the Hyper-V switch. This technology will be referred to as Switch-Embedded Teaming (SET) for the rest of this chapter.

Teaming configurations

There are two basic configurations for NIC Teaming.

Switch-independent teaming.

With Switch-independent teaming, teaming is configured without the knowledge or participation of the switch. The switch is not aware that the network adapter is part of a team in the host, allowing the adapters the flexibility to be connected to different switches. Switch independent modes of operation do not require that the team members connect to different switches; they merely make it possible.

Active/stand-by teaming: Rather than take advantage of the bandwidth aggregation capabilities of NIC Teaming, some administrators prefer to build in an extra layer of redundancy. These administrators choose to use one or more team members for traffic (active) and one team member to be held in reserve (stand-by) to come into action if an active team member fails. This mode of teaming can be set by first setting the team to Switch-independent teaming, then selecting a stand-by team member through the management tool you are using. Fault-tolerance is present whether Active/Stand-by is set or not, provided there are at least two network adapters in a team. Furthermore, if your Switch Independent team has at least two members, one adapter can be marked by Windows NIC Teaming as a stand-by adapter. This stand-by adapter will only be used for inbound traffic, unless the active adapter fails. Inbound traffic (e.g., broadcast packets) received on the stand-by adapter will be delivered up the stack. Once the failed team member or team members are restored, the stand-by team member will return to stand-by status.

Once a stand-by member of a team is connected to the network, all network resources required to service traffic on the member are in place and active. While Active/Standby configuration provides administrators with peace of mind, clients will see better network utilization and lower latency by operating their teams with all team members active. In a failover situation, the redistribution of traffic across the remaining healthy team members will occur anytime one or more of the team members reports an error state, whether adapters are set to active or not.

Switch-dependent teaming.

As you might have guessed, this second option requires the participation of the switch. All members of the team must be connected to the same physical switch. There are two modes of operation for switch-dependent teaming:

Generic or static teaming (IEEE 802.3ad draft v1): Both the switch and the host require configuration for this mode to function. It is a statically configured solution and for that reason there is no additional protocol to assist the switch and the host to identify incorrectly plugged cables or other errors that could cause the team to fail to perform. Typically, this mode is supported by and used for server-class switches.

Link Aggregation Control Protocol teaming (IEEE 802.1ax, LACP). This mode is also commonly referred to as IEEE 802.3ad as it was developed in the IEEE 802.3ad committee before being published as IEEE 802.1ax. This mode uses the Link Aggregation Control Protocol (LACP) to dynamically identify links that are connected between the host and a given switch. Teams are thus automatically created and in theory (but rarely in practice), the team can be expanded or reduced simply by the transmission or receipt of LACP packets from the peer entity. IEEE 802.1ax is supported by typical server-class switches, but most require the network operator to administratively enable LACP on the port. Windows NIC Teaming always operates in LACP's Active mode with a short timer. No option is presently available to modify the timer or change the LACP mode.

The above modes allow both inbound and outbound traffic to approach the practical limits of the aggregated bandwidth because the pool of team members is seen as a single pipe.

Inbound load distribution is governed by the switch. For this reason, it is important to research the options available for inbound load distribution management. For example, a good number of switches only support destination IP address to team member mapping, resulting in a less granular distribution than is needed for a good inbound load distribution. Covering all the settings on all switches is not feasible in this guide, so it remains up to the reader to research and understand the capabilities of the adjacent network switches in their environment.

In Windows Server 2016 Stand-alone NIC Teaming supports all these modes;

Switch-embedded NIC Teaming supports Switch Independent mode with no stand-by team members.

Note: *If you have previously configured your environments using Switch-Dependent Teaming (LACP) configurations, these are no longer supported for Storage Spaces Direct deployments and you will need to move your configurations to 100% Switch Independent.*

Algorithms for load distribution

Distribution of outbound traffic among the available links can be configured in many ways. A rule-of-thumb governing any distribution algorithm is to try to keep all packets associated with a single flow (TCP-stream) on a single network adapter. This minimizes performance degradation caused by reassembling out-of-order TCP segments.

Stand-alone NIC teaming supports the following traffic load distribution algorithms:

Hyper-V switch port: Because virtual machines are each assigned independent MAC addresses, the MAC Address or the port it is connected to on the Hyper-V switch can be the basis for dividing traffic. This scheme can be advantageous in virtualization. The adjacent switch sees a particular MAC Address connected to only one port, allowing the switch to automatically distribute the ingress load (the traffic from the switch to the host) on multiple links based on the destination MAC (VM MAC) address. This is particularly useful when Virtual Machine Queues (VMQs) can be placed on the

specific NIC where the traffic is expected to arrive. However, if the host has only a few VMs, this mode may not be granular enough to obtain a well-balanced distribution. This mode will also always limit a single VM (i.e., the traffic from a single switch port) to the bandwidth available on a single interface. Windows Server 2012 R2 uses the Hyper-V Switch Port as the identifier rather than the source MAC address as in some instances a VM may be using more than one MAC address on a switch port.

Address hashing: An algorithm is used to create a hash based on address components of the packet and then assigns packets with that particular hash value to one of the available adapters. This mechanism alone is usually sufficient to create a reasonable balance across the available adapters.

The components that can be specified, using PowerShell, as inputs to the hashing function include the following:

- Source and destination TCP ports and source and destination IP addresses (this is used by the user interface when "Address Hash" is selected)
- Source and destination IP addresses only
- Source and destination MAC addresses only

The TCP ports hash creates the most granular distribution of traffic streams, resulting in smaller streams that can be independently moved between members. However, it cannot be used for traffic that is not TCP or UDP-based, nor can it be used where the TCP and UDP ports are hidden from the stack, such as IPsec-protected traffic. In these cases the hash automatically falls back to the IP address hash or, if the traffic is not IP traffic, to the MAC address hash.

Dynamic: The best aspects of the previous two modes are combined in this algorithm.

A hash is created based on TCP ports and IP addresses and is used to distribute outbound loads. Also, in Dynamic mode loads are rebalanced in real time so that a given outbound flow may move back and forth between team members.

For inbound loads, distribution occurs as if the Hyper-V port mode was in use.

The outbound loads in this mode are dynamically balanced based on the concept of "flowlets." TCP flows have naturally occurring breaks, much like the natural breaks between words and sentences found in human speech. A flowlet is the 'body' of a particular portion of the TCP flow, or the packet-flow between two such breaks. The dynamic mode algorithm detects these flowlet boundaries (the boundary being any break in the packet flow of sufficient length) and redistributes the flow to the least-taxed team member as appropriate. If flows do not contain any discernible flowlets, the algorithm may periodically rebalance flows regardless, if circumstances require it. The dynamic balancing algorithm can change the affinity between TCP flow and a team member at any time as it works to balance the workload of the team members.

Switch-embedded teaming supports only the Hyper-V switch port and Dynamic load distribution algorithms.

Note: *From our experience Dynamic is the preferred choice 99% of the time. It is even recommended by Microsoft Consulting Services (MCS) in most cases.*

Converged Network Interface Card (NIC)

With the Converged Network Interface Card (NIC), a single network adapter can be used for management, Remote Direct Memory Access (RDMA)-enabled storage and tenant traffic. This allows you to use fewer network adapters to manage different types of traffic, potentially reducing overall capital expenditure.

Switch Independent / address hash

With this configuration, loads are distributed through the selected level of address hashing. TCP ports and IP addresses are used by default to seed the hash function.

Because a given IP address can only be associated with a single MAC address for routing purposes, this mode essentially restricts inbound traffic to only one team member (the primary member). This means that the inbound traffic is limited to the bandwidth of one team member no matter how much is getting sent.

This mode is best used for teaming in a VM.

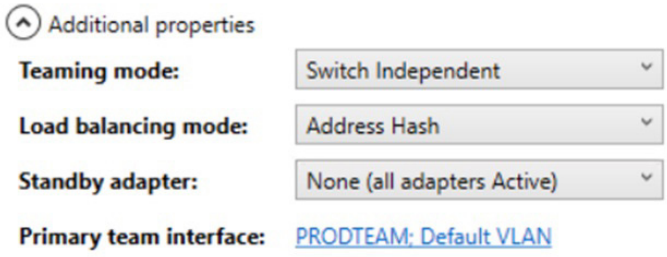

Figure 21: LBFO Team configuration Switch Independent / address hash

Switch Independent / Hyper-V port

In this configuration, packets are sent using all active team members, distributing the load based on the Hyper-V switch port number. Bandwidth will be limited to no more than one team member's bandwidth because the port is affinitized to exactly one team member at any point in time.

With the Hyper-V port associated to only a single team member, inbound traffic for the VM's switch port is received on the same team member the switch port's outbound traffic uses. This also allows maximum use of VMQs for better performance overall.

Hyper-V Port configuration is best used only when teaming NICs that operate at or above 10Gbps. For high bit-rate NICS such as these, Hyper-V port distribution mode may provide better performance than Dynamic distribution. In all other cases where Hyper-V port was recommended in previous Windows Server versions, Switch-Independent/Dynamic, covered next, will provide better performance.

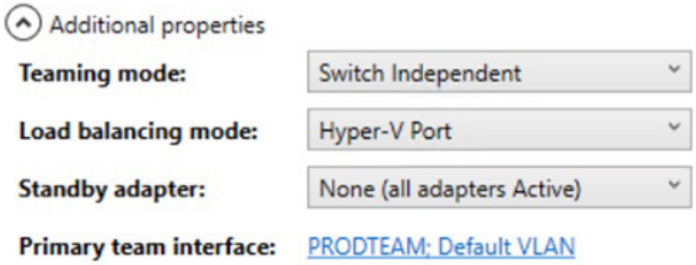

Figure 22: LBFO Team configuration Switch Independent / Hyper-V Port

Switch Independent / Dynamic

With Switch-Independent/Dynamic mode, the load is distributed based on a TCP port's hash modified by the Dynamic load-balancing algorithm. The algorithm redistributes flows to optimize team member bandwidth utilization and as a result, individual flow transmissions may move from one active team member to another. When redistributing traffic, there

is always a small possibility that out-of-order delivery could occur, but the dynamic algorithm takes that into account and takes steps to minimize that possibility.

On the receiving side, distribution will look identical to Hyper-V port mode. The traffic of each Hyper-V switch port, whether bound for a virtual NIC in a VM (vmNIC) or a virtual NIC in the host (vNIC), will see all inbound traffic arriving on a single NIC.

This mode is best used for teaming in both native and Hyper-V environments except when:

- Teaming is being performed in a VM
- Switch dependent teaming (e.g., LACP) is required by policy
- Operation of an Active/Stand-by team is required by policy

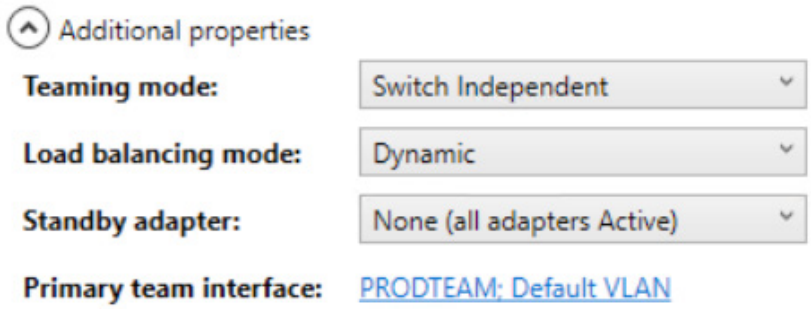

Figure 23: LBFO Team configuration Switch Independent / Address Dynamic

Switch Embedded Teaming (SET)

Switch Embedded Teaming is an NIC Teaming solution that is integrated in the Hyper-V Virtual Switch. Up to eight physical NICS can be added into a single SET team, improving Availability and ensuring failover. Windows Server 2016 lets you create SET teams that are restricted to the use of Server Message Block (SMB) and RDMA. In addition, SET teams can be used to distribute network traffic for Hyper-V network virtualization.

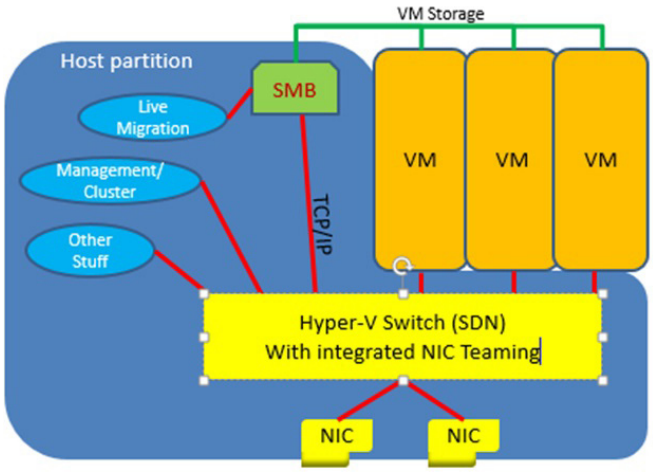

Figure 24: SET Team architecture ®Microsoft

SET cannot be used in a virtual machine, as it is integrated into the Hyper-V Virtual Switch. However, you can use other forms of NIC Teaming within said VMs.

Another benefit of SET architecture is that it does not expose team interfaces. Instead, you must configure Hyper-V Virtual Switch ports.

SET Availability

If your version of Windows Server 2016 includes Hyper-V and the SDN stack, SET is available. Windows PowerShell commands and Remote Desktop connections are available to let you manage SET from remote computers that are running a client operating system upon which the tools are supported.

SET Supported NICs

Any Ethernet NIC that is trusted and certified by the Windows Hardware Qualification and Logo (WHQL) test can be added in a SET team in Windows Server 2016. All network adapters that are members of a SET team must be of an identical make and model (same manufacturer, model, firmware and driver). As stated earlier, SET supports anywhere from one up to eight network adapters in a team.

SET cannot be used in a virtual machine, as it is integrated into the Hyper-V Virtual Switch. However, you can use other forms of NIC Teaming within said VMs.

Another benefit to SET architecture is that it does not expose team interfaces. Instead, you must configure Hyper-V Virtual Switch ports.

SET Modes and settings

When creating a SET Team, unlike NIC Teaming, a team name cannot be configured. Another difference between SET and standard NIC Teaming is that stand-by adapters cannot be used in SET. When you deploy SET, all network adapters must be active. Also, NIC Teaming provides three different teaming modes, while SET only supports Switch Independent.

With Switch Independent mode, the switch or switches to which the SET Team members are connected are unaware of the presence of the SET team and do not determine how to distribute network traffic to SET team members — instead, the SET team distributes inbound network traffic across the SET team members.

When you create a new SET team, you must configure the following team properties:

- Member adapters
- Load balancing mode

Member adapters

When you create a SET team, you must specify up to eight identical network adapters that are bound to the Hyper-V Virtual Switch as SET team member adapters.

Load balancing modes for SET

The options for SET team load balancing distribution mode are Hyper-V Port and dynamic.

Note: *When you use SET in conjunction with Packet Direct, the teaming mode Switch Independent and the load balancing mode Hyper-V Port are required.*

SET and virtual machine queues (VMQs)

VMQ and SET work well together and you should enable VMQ whenever you are using Hyper-V and SET.

Note: *SET always presents the total number of queues that are available across all SET team members. In NIC Teaming, this is called Sum-of-Queues mode.*

Most network adapters have queues that can be used for either Receive Side Scaling (RSS) or VMQ, but not both at the same time.

Some VMQ settings appear to be settings for RSS queues but are really settings on the generic queues that both RSS and VMQ use depending on which feature is presently in use. Each NIC has, in its advanced properties, values for *RssBaseProcNumber and *MaxRssProcessors.

Following are a few VMQ settings that provide better system performance:

- Ideally each NIC should have the *RssBaseProcNumber set to an even number greater than or equal to two (2). This is because the first physical processor, Core 0 (logical processors zero and one), typically does most of the system processing so the network processing should be steered away from this physical processor.
- The team members' processors should be, to the extent that it's practical, non-overlapping. For example, in a four-core host (eight logical processors) with a team of two 10Gbps NICs, you could set the first one to use a base processor of two and to use four cores; the second would be set to use base processor six and use two cores.

SET and Hyper-V Network Virtualization

SET is fully compatible with Hyper-V Network Virtualization in Windows Server 2016. The HNV management system provides information to the SET driver that allows SET to distribute the network traffic load in a manner that is optimized for the HNV traffic.

SET and Live Migration

The use of SET teams and Live Migration is fully supported in Windows Server 2016.

MAC Address use on transmitted packets

When a SET team is configured with dynamic load distribution, the packets from a single source (such as a single VM) are simultaneously distributed across multiple team members.

SET replaces the source MAC address with a different MAC Address on the frames transmitted on team members other than the affinitized team member. This prevents the switches from getting confused and prevents MAC flapping alarms. For this reason, each team member uses a different MAC address, preventing MAC address conflicts unless and until failure occurs.

If a failure occurs on the primary NIC and is detected by SET, the teaming software uses the VM's MAC address on the team member that is chosen to serve as the temporary affinitized team member (i.e., the one that will now appear to the switch as the VM's interface).

This MAC Address change only applies to traffic that was going to be sent on the VM's affinitized team member with the VM's own MAC address as its source MAC address. Other traffic continues to be sent with whatever source MAC address it would have used prior to the failure.

Following are lists that describe SET teaming MAC address replacement behavior, based on how the team is configured:

In Switch Independent mode with Hyper-V Port distribution:

- Every vmSwitch port is affinitized to a team member
- Every packet is sent on the team member to which the port is affinitized
- No source MAC replacement is done

In Switch Independent mode with Dynamic distribution:

- Every vmSwitch port is affinitized to a team member
- All ARP/NS packets are sent on the team member to which the port is affinitized
- Packets sent on the team member that is the affinitized team member have no source MAC address replacement done
- Packets sent on a team member other than the affinitized team member will have source MAC address replacement done

SET vs. LBFO Teaming

The table below is a comparison of the features supported by LBFO teaming and SET teaming.

LBFO/SET Feature comparison

Feature	LBFO	SET	**Feature interactions:** works with	LBFO	SET
Switch independent teaming			Checksum offloads		
Switch dependent teaming: Static			DCB		
Switch dependent teaming: LACP			HNV v1		
Dynamic load distribution			HNV v2		
HyperVPort mode load distribution			IEEE 802.1X		
Address hash load distribution			IPsecTO		
Active/Standby operation			LSO		
Teams of up to ___ members	32	8	RDMA		
VMM managed		RTM	RSC		
Inbox UI managed			RSS		
PowerShell managed			SDN-QoS		
Works in Native stack			SR-IOV		
Works in a VM			TCP Chimney		
Teams different speed NICs			VMMQ		
Teams different NICs			VMQ (filter)		
vNICs/vmNICs affinitized to team members			VMQ (NIC Switch)		
			vmQoS		
			vRSS		

Figure 25: LBFO vs. SET Team comparison ®Microsoft

Managing SET teams

Switch-embedded teams are best managed using the Virtual Machine Manager (VMM).

In the event an administrator prefers to use PowerShell to manage a switch-embedded team, here are the cmdlets to use.

Creating a SET team

A switch-embedded team must be created at the time the Hyper-V switch is created.

When creating the Hyper-V switch using the New-VMSwitch PowerShell cmdlet, the "EnableEmbeddedTeaming" option must be selected.

For example, the PowerShell cmdlet shown here will create a Hyper-V switch named TeamedvSwitch with embedded teaming and two initial team members.

```
New-VMSwitch -Name TeamedvSwitch -NetAdapterName "NIC 1","NIC 2"
-EnableEmbeddedTeaming $true
```

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.

Failover Clustering

Failover Clustering –– a Windows Server feature that enables you to group multiple servers together into a fault-tolerant cluster –– provides new and improved features for software-defined data center customers and many other workloads running clusters on physical hardware or in virtual machines.

Independent servers and computers can fail for any number of reasons and it is rarely, if ever, a desirable outcome. A failover cluster groups together independent machines to work together to ensure that data or applications remain functional even should such a failure occur. Connected by physical cables and software, each server can leverage the resources of their clustered counterparts to ensure continuity. Each clustered server (or sometimes group of servers) is called a node. If one or more of the cluster nodes fail, other nodes begin to provide service (a process known as failover). In addition, the clustered roles are proactively monitored to verify that they are working properly. If they are not working, they are restarted or moved to another node.

Failover clusters also provide Cluster Shared Volume functionality that provides a consistent, distributed namespace that clustered roles can use to access shared storage from all nodes. With the Failover Clustering feature, users experience a minimum of disruptions in service.

Failover Clustering has many practical applications, including:

- Highly available or continuously available file share storage for applications such as Microsoft SQL Server and Hyper-V virtual machines
- Highly available clustered roles that run on physical servers or on virtual machines that are installed on servers running Hyper-V
- Primary interface for configuring Storage Spaces Direct

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 Resilience sets out to resolve these problems. VM Compute Resiliency will provide clustered Hyper-V environments the ability to withstand minor service disruptions without aggressively failing over VMs and their services to other surviving cluster nodes. The configurable default setting on Server 2016 is four 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:** A disconnected cluster node (host) that cannot communicate with any other node within the active cluster. The node will however continue to host the VM role.
- **Quarantine**: Host state that is noted whenever the node is no longer permitted to join the cluster (Default two hours).
  - Node is quarantined when the node ungracefully leaves / joins the cluster three times 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 while introducing many environmental limitations.

Windows Server 2016 is introducing the following abilities:

- 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 benefit 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, five 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."

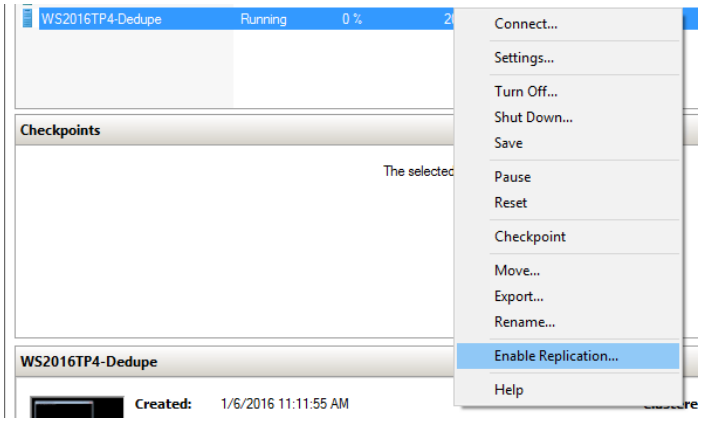

Figure 21: Enable Hyper-V Replication on Windows Server 2012 R2

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 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. When 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.

Storage Replica

Another new feature in Windows Server 2016 is Storage Replica. This new feature enables storage-agnostic, block-level, synchronous replication between servers or clusters for disaster recovery, as well as the ability to stretch a failover cluster between sites. Synchronous replication enables mirroring of data in physical sites with crash-consistent volumes to ensure zero data loss at the file-system level. Asynchronous replication allows site extension beyond metropolitan ranges with the possibility of data loss.

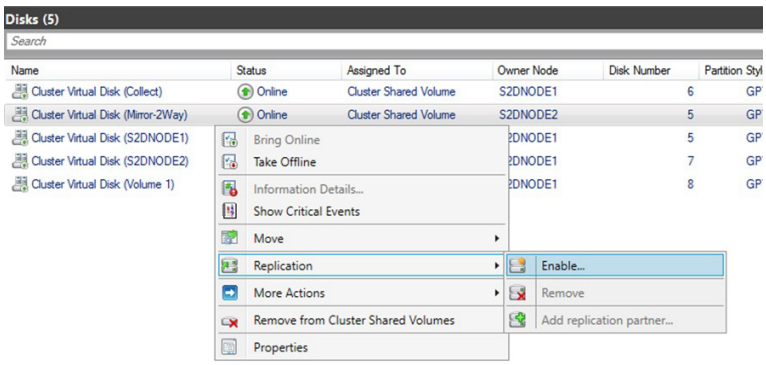

Figure 22: Configuring Storage Replica from failover cluster manager

Memory management enhancements

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
```

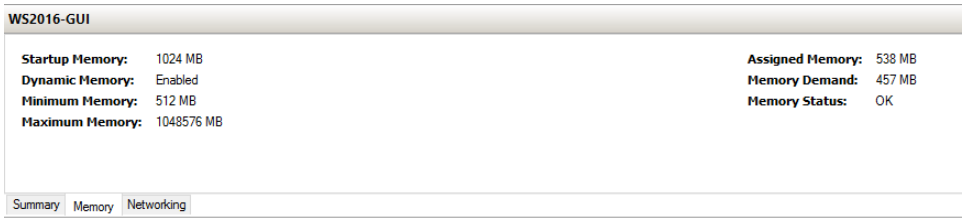

Figure 23: Hyper-V Manager: Memory demand vs. assigned memory

Hot-add applies the ability to dynamically expand 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 of whether 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 two 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
```

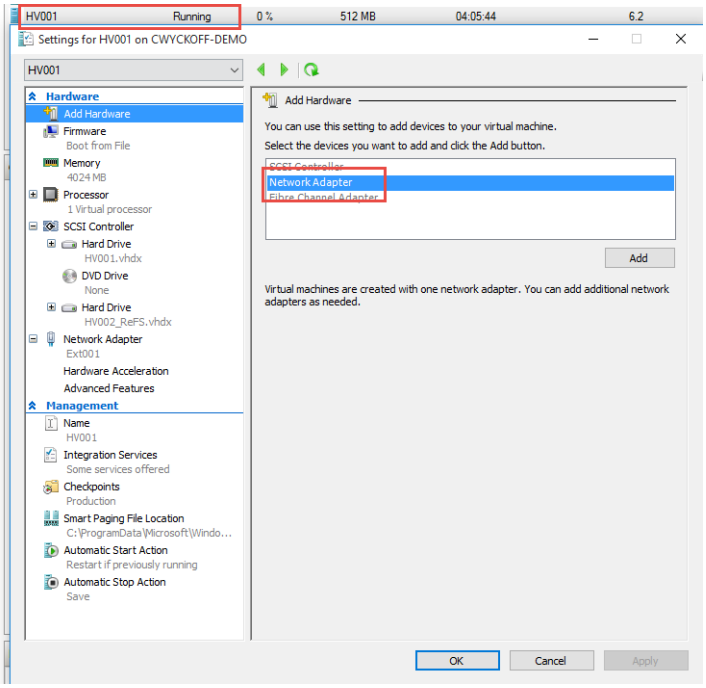

Figure 24: Hot-add vNic to VM through Hyper-V Manager

Cloud Witness for a failover cluster

Windows Server 2016 also enhances standard quorum witness functionality by extending it to the cloud. Cloud Witness is a new type of failover cluster quorum witness that leverages Microsoft Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations. You can configure cloud witness as a quorum witness using the Configure a Cluster Quorum Wizard.

What value does this change add?

Using Cloud Witness as a failover cluster quorum witness provides the following advantages:

- Leverages Microsoft Azure and eliminates the need for a third separate data center
- Uses the standard publicly available Microsoft Azure blob storage which eliminates the extra maintenance overhead of VMs hosted in a public cloud
- The same Microsoft Azure storage account can be used for multiple clusters (one blob file per cluster; the cluster's unique id used as blob file name)
- Provides a very low ongoing cost to the Storage Account (a very small amount of data written per blob file, with the blob file updated only once when cluster node's state changes)

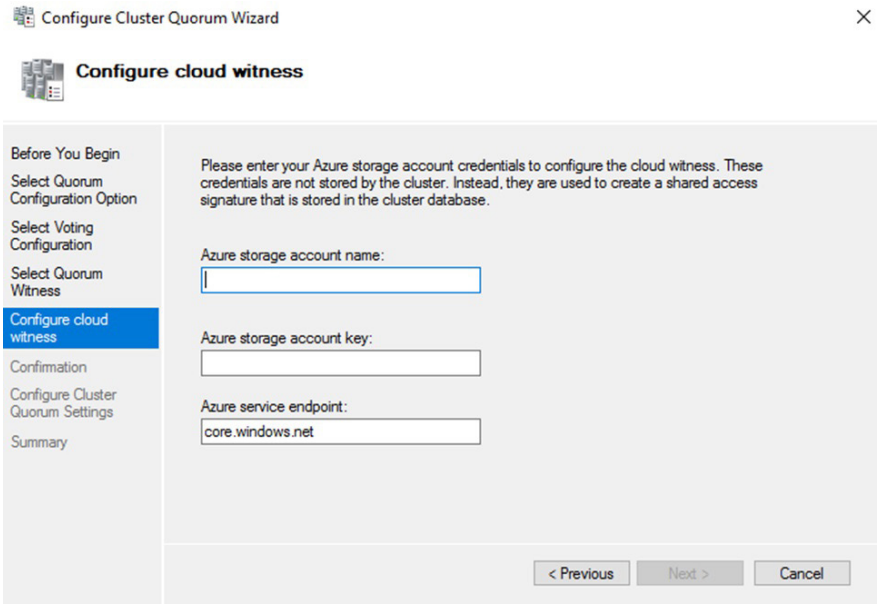

Figure 25: Configuring a Cloud Witness using Microsoft Azure

This is configured via PowerShell. Below is an example:

```
Set-ClusterQuorum -CloudWitness -AccountName <AzureStorageAccountNameHere>
-AccessKey <AccessKeyHere> -Endpoint core.windows.net
```

Workgroup and multi-domain clusters

In Windows Server 2012 R2 and previous versions, a cluster can only be created between member nodes joined to the same domain. Windows Server 2016 breaks down these barriers and introduces the ability to create a failover cluster without Active Directory dependencies. You can now create failover clusters in the following configurations:

- Single-domain clusters. Clusters with all nodes joined to the same domain.
- Multi-domain clusters. Clusters with nodes which are members of different domains.
- Workgroup clusters. Clusters with nodes which are member servers / workgroup (not domain joined).

VM Load balancing

Virtual machine Load Balancing is a new feature in failover clustering that facilitates the seamless load balancing of virtual machines across the nodes in a cluster. Over-committed nodes are identified based on virtual machine memory and CPU utilization on the node. Virtual machines are then moved (live migrated) from an over-committed node to nodes with available bandwidth (if applicable). The aggressiveness of the balancing can be tuned to ensure optimal cluster performance and utilization. Load balancing is enabled by default in Windows Sever 2016. However, load balancing is disabled when SCVMM Dynamic Optimization is enabled.

Virtual machine Start Order

Virtual machine Start Order is a new feature in failover clustering that introduces start order orchestration for virtual machines (and all groups) in a cluster. Virtual machines can now be grouped into tiers and start order dependencies can be created between different tiers. This ensures that the most important virtual machines (such as Domain Controllers or utility virtual machines) are started first. Virtual machines are not started until the virtual machines that they have a dependency on are also started.

Simplified SMB Multi-channel and Multi-NIC Cluster Networks

Failover cluster networks are no longer limited to a single NIC per subnet / network. With Simplified SMB Multi-channel and Multi-NIC Cluster Networks, network configuration is automatic and every NIC on the subnet can be used for cluster and workload traffic. This enhancement allows customers to maximize network throughput for Hyper-V, SQL Server Failover Cluster Instance and other SMB workloads. It also enables easier configuration of multiple network adapters in a cluster.

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 one within Microsoft Failover Cluster Manager or Virtual Machine Manager
2. Evict node one 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 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 five, 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, is 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 PoweShell 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 two Windows VMs on Server 2012.

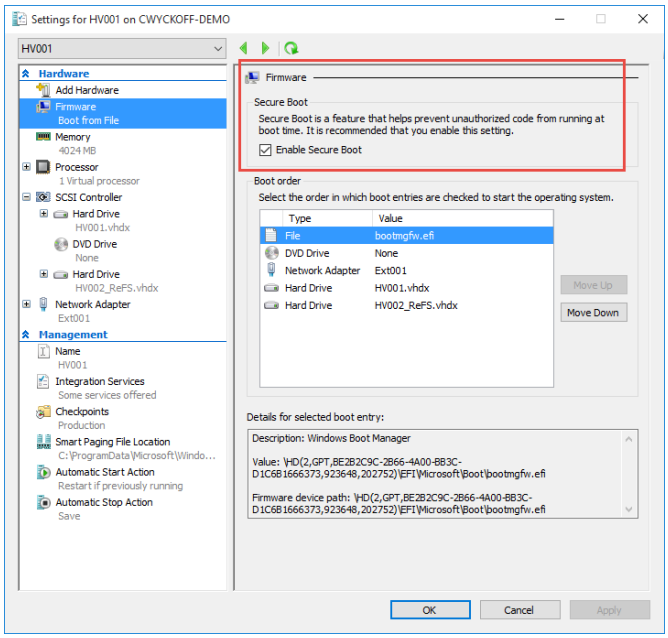

Figure 26: Secure Boot on powered-off Windows VM

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 two 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 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 has been generally available 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 the cores within a physical server. The minimum number of cores to license equals eight per processor or 16 per server while the minimum count of cores to license per physical server equals 16.

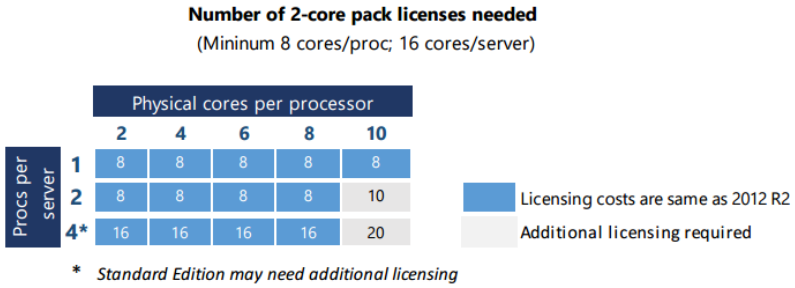

Figure 2: Server CPU core licensing, courtesy of Microsoft®

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.

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

With the ISO file, we are now ready to deploy our first Windows Server 2016 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.

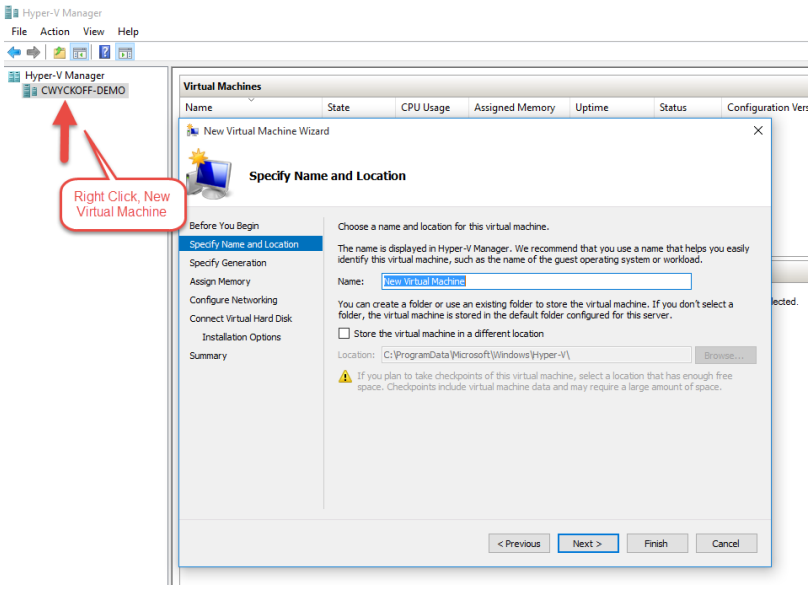

Figure 3: New Virtual Machine Wizard

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 two 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.

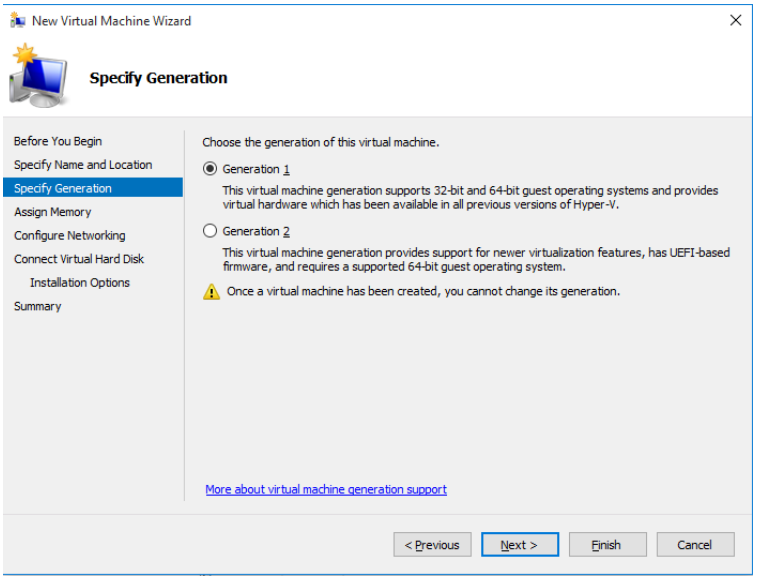

Figure 4: New Virtual Machine Wizard, VM generation

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.

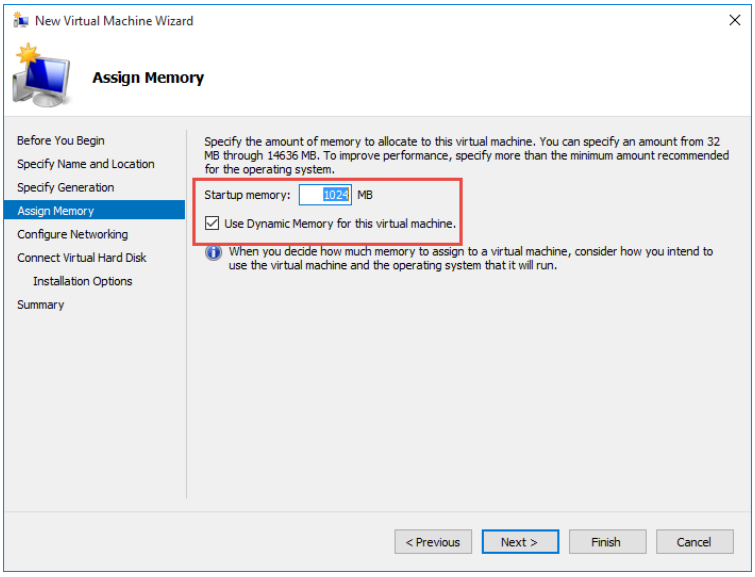

Figure 5: New Virtual Machine Wizard, Assign Memory

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.

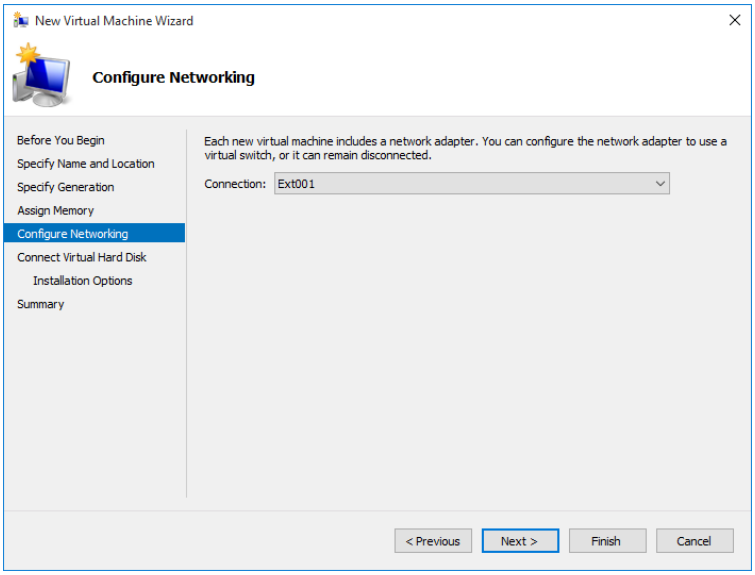

Figure 6: New Virtual Machine Wizard, Configure Networking

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.

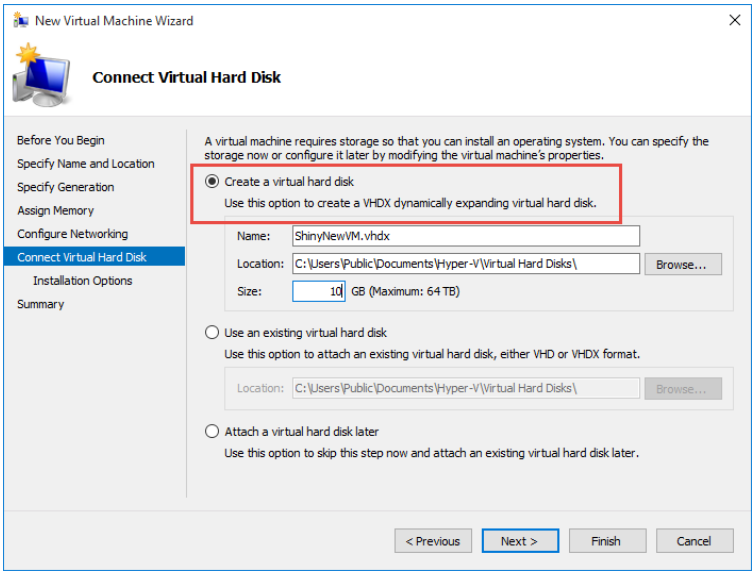

Figure 7: New Virtual Machine Wizard, connect virtual hard disk

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

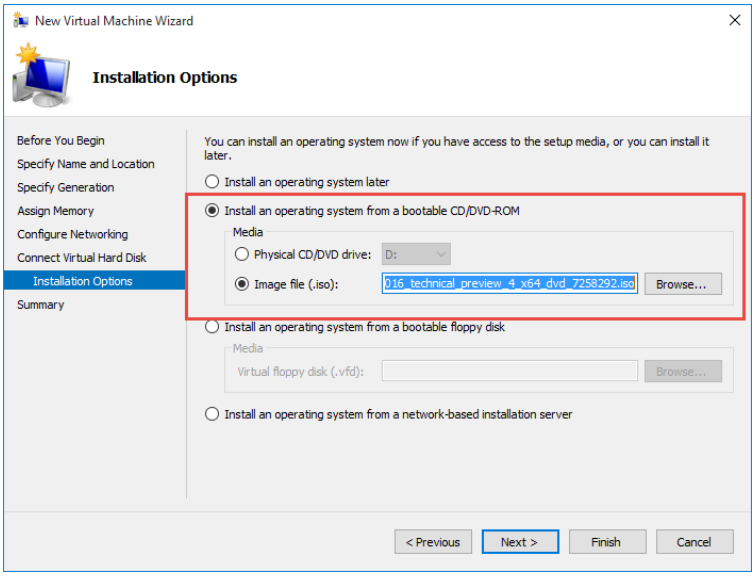

Figure 8: New Virtual Machine Wizard, installation options

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**.

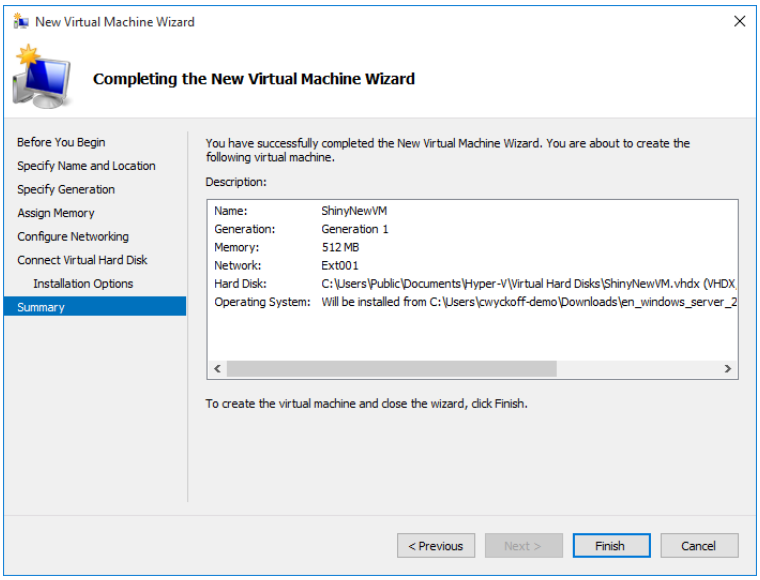

Figure 9: New Virtual Machine Wizard, completing the New Virtual Machine Wizard

Upon completion, we are now ready to power on the VM.

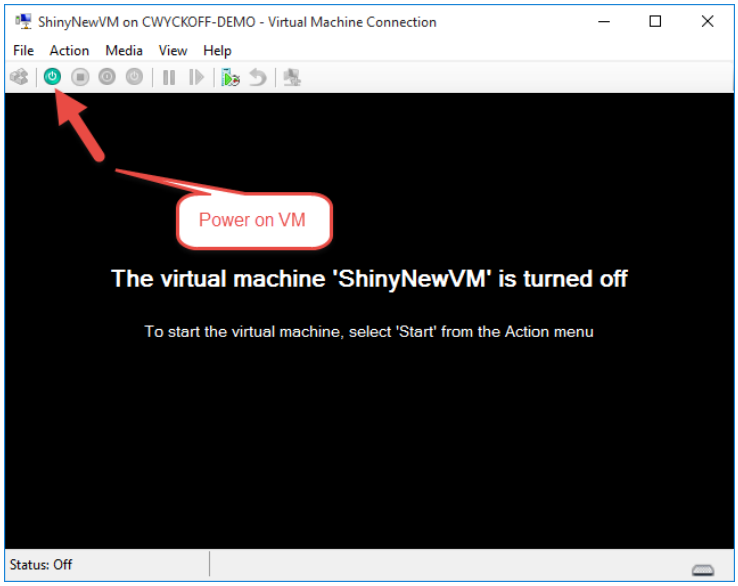

Figure 10: Virtual machine connection, power on VM

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.

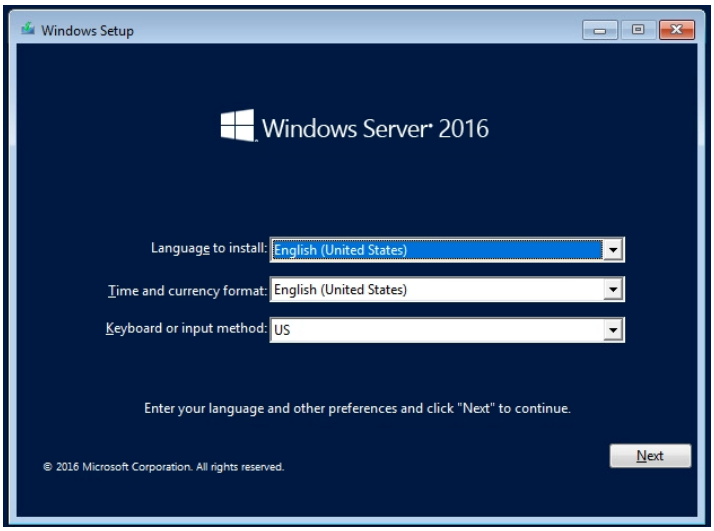

Figure 11: Windows Server 2016 Installation Wizard

With the VM powered on, the Windows Server 2016 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**.

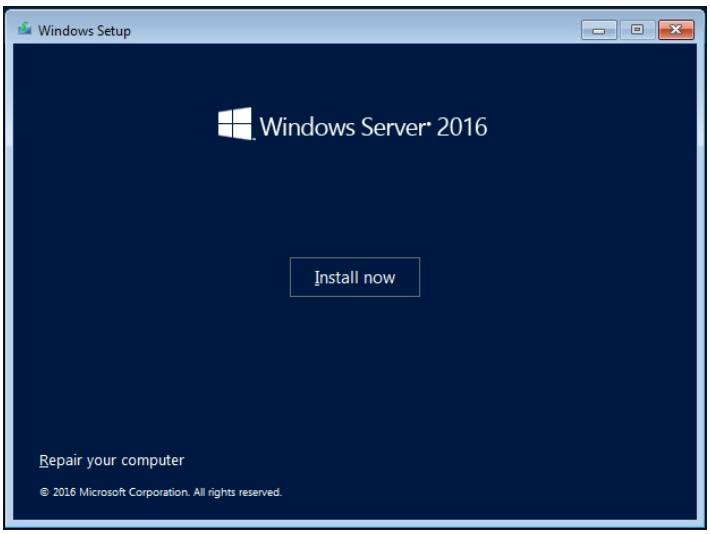

Figure 12: Windows Server 2016 Installation Wizard, Install Now

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.

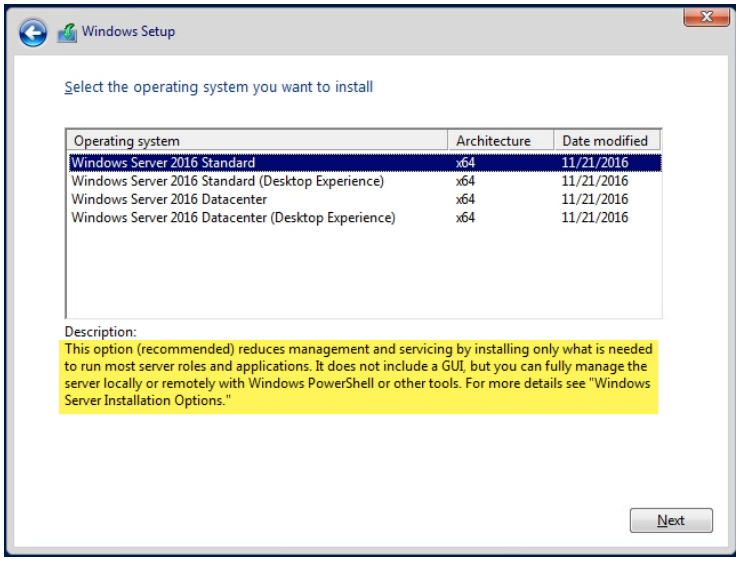

Figure 13: Windows Server 2016 Installation Wizard, choose GUI or Core installation

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

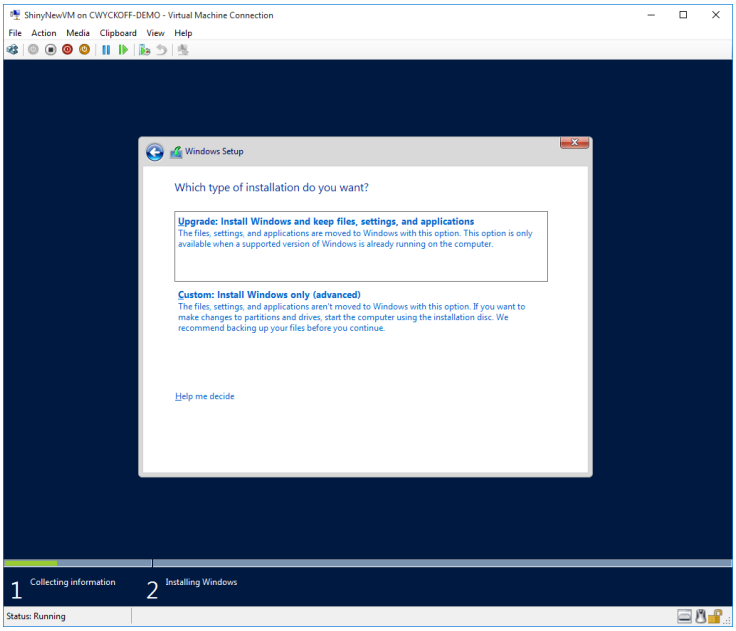

Figure 14: Windows Server 2016 Installation Wizard, installation options

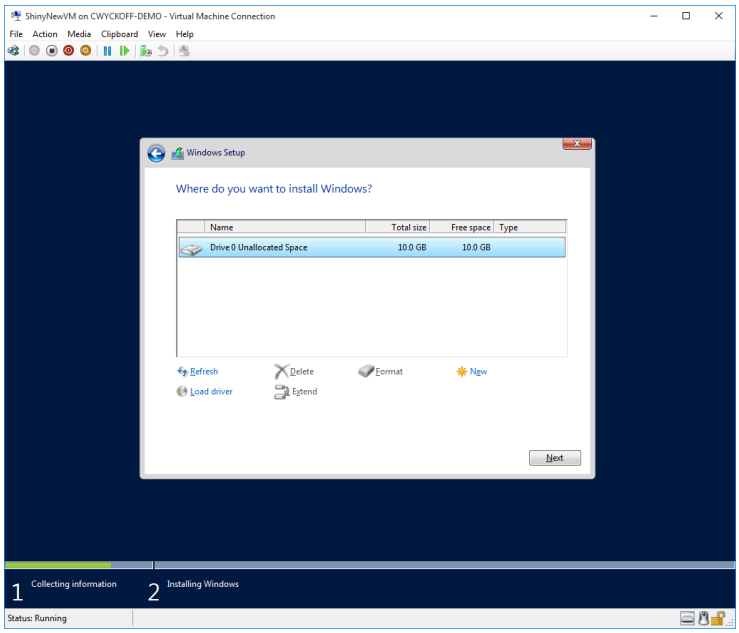

Figure 15: Windows Server 2016 Installation Wizard, disk allocation and partition setup

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

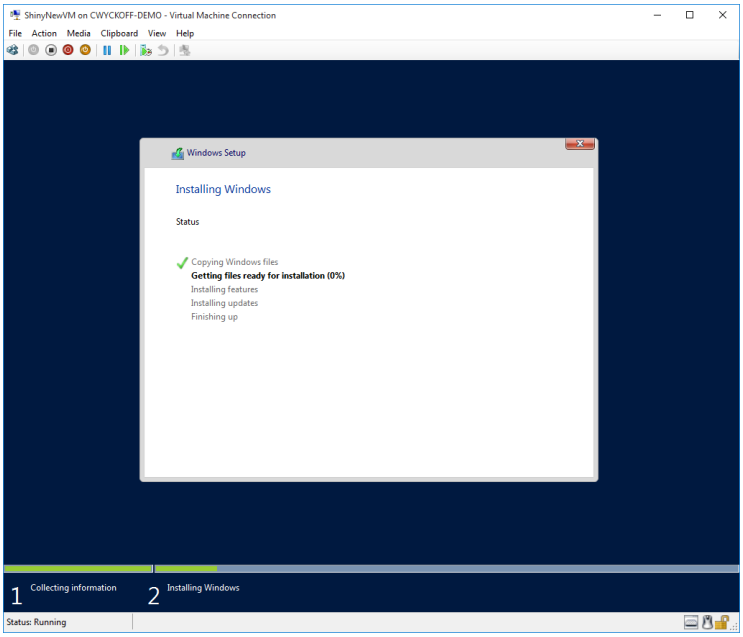

Figure 16: Windows Server 2016 Installation Wizard, Windows installation

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

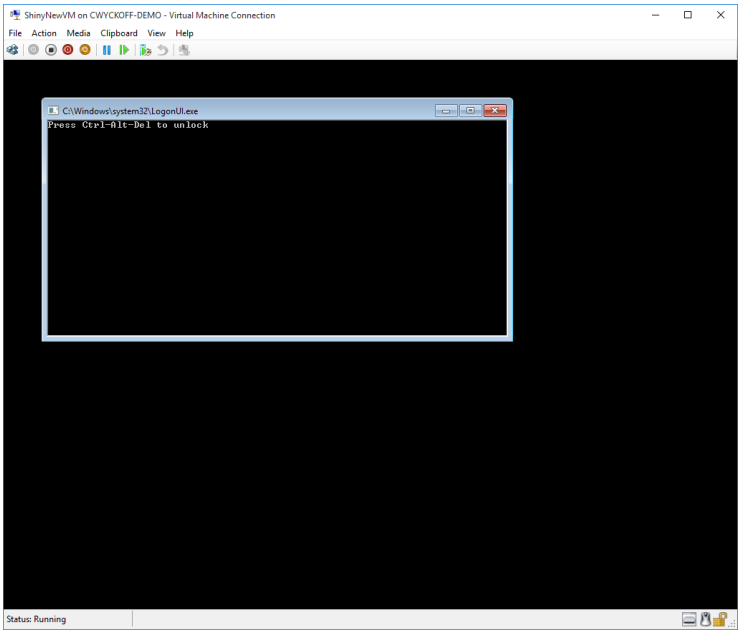

Figure 17: Press Ctrl + Alt + Del to unlock and set up Windows Server 2016 for application deployment

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 Senior Technical Evangelist at Veeam with a focus on all things Microsoft. He is an avid technologist and virtualization fanatic with more than a decade of enterprise data center architecture experience. Clint is an energetic and engaging speaker and places a large emphasis on solving the real-world challenges IT professionals face. Additionally, he is a two time Microsoft Most Valuable Professional (MVP) for Cloud and Datacenter Management and a three-time VMware vExpert for 2015, 2016 and 2017. Clint is a Cisco Champion, a Veeam Certified Engineer (VCME) and Microsoft Certified Professional (MCP). You can follow Clint on Twitter @ClintWyckoff or @Veeam.

Dave is a Microsoft Most Valuable Professional (MVP) with over 20 years of experience in the IT industry. His background includes data communications networks within multi-server environments and he has led architecture teams for virtualization, System Center, Exchange, Active Directory and Internet gateways. Very active within the Microsoft technical and consulting teams, Dave has provided deep-dive technical knowledge and subject matter expertise on various System Center and operating system topics.

Dave is well-known in the community as an evangelist for Microsoft, 1E and Veeam technologies. Locating Dave is easy as he speaks at several conferences and sessions each year, including TechEd, Ignite, MVP Days Community Roadshow and VeeamOn.

As the founder and Managing Principal Consultant at TriCon Elite Consulting, Dave is a leading technology expert for both local customers and large international enterprises, providing optimal guidance and methodologies to achieve and maintain an efficient infrastructure.

Blog: http://www.checkyourlogs.net

Twitter: @DaveKawula

About Veeam Software

Veeam® recognizes the new challenges companies across the globe face in enabling the Always-On Business™, a business that must operate 24.7.365. To address this, Veeam has pioneered a new market of Availability for the Always-On Enterprise™ by helping organizations meet recovery time and point objectives (RTPO™) of < 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™, which includes Veeam Backup & Replication™, 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 43,000 ProPartners and more than 216,500 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™

VEEAM

Veeam makes the Fortune 500 Available. 24.7.365

To enable its **Digital Transformation,** 70% of the Fortune 500 rely on Veeam to ensure Availability of all data and applications. **24.7.365**