



**Hewlett Packard**  
Enterprise

# **Implementing Microsoft® Windows Server® 2016 using HPE ProLiant Servers, Storage, and Options**

# Contents

Abstract .....	4
Windows Server 2016 overview .....	4
Recommended system configurations.....	4
Supported ProLiant servers .....	5
Supported ProLiant server platform options.....	7
Software and drivers .....	7
Network Interface Controllers .....	9
Persistent Memory.....	10
Introduction.....	10
Hardware and Firmware requirements.....	10
Hardware and Firmware configuration .....	11
Windows Server 2016 Storage Class Memory (SCM) enablement.....	12
HPE NVDIMM use scenarios with Windows Server 2016 .....	19
HPE NVDIMM Performance with Windows Server 2016.....	23
SCM Device Management in Nano Server using Windows PowerShell.....	24
Prerequisites .....	24
HPE Persistent Memory Resources, contacts, or additional links.....	28
Security and Assurance.....	29
Introduction.....	29
Protecting UEFI and Windows.....	29
Protecting Cloud and Virtual Environment.....	30
TPM.....	30
HPE ProLiant Gen9 Servers with Hardware Assurance Additional Qualifier (AQ).....	30
Configuring HPE ProLiant Gen9 Servers with UEFI Secure Boot.....	31
Configuring HPE Trusted Platform Module 2.0 for HPE Gen9 servers .....	31
Early Loading Anti-Malware (ELAM).....	32
Credential Guard and Remote Credential Guard.....	32
HPE ProLiant Server Feature for Credential Guard.....	33
Host Guardian Service and Shielded VMs .....	34
Additional Resources .....	36
Installing Windows Server 2016.....	36
Preinstallation tasks.....	36
Installing Windows Server 2016 from the OS media.....	37
Installing Windows Server 2016 using iLO.....	37
Installing components from the HPE Service Pack for ProLiant (SPP) version 2016.10.0 .....	37
Questions, issues, and workarounds.....	38
Nano Server .....	39

Introduction.....	39
Getting Started.....	40
Preparing the environment .....	41
A word about Nano Server images.....	42
Creating a custom Nano Server WIM image.....	43
Deploying a Nano Server WIM from WinPE .....	44
Deploying Nano Server using Windows Deployment Services.....	53
Creating a custom Nano Server VHD(x) image.....	66
Deploying a VHD(x) file to a physical computer in a dual-boot environment .....	67
Performing Windows Update on Nano Server.....	69
Appendix.....	70
Nano Server Packages.....	70
Disk Configuration for WinPE Deployments .....	70
Adding boot-critical drivers to Windows boot images .....	71
Nano Server Unattend.xml file for WIM Images.....	73
Nano Server Unattend XML files for WDS deployments.....	75
Obtaining the MAC address through iLO and System ROM.....	81
Enabling network boot on an HPE ProLiant server.....	81
Nano Server Remote Management.....	82
Installing the HPE ProLiant Agentless Management Service in Nano Server .....	86
Installation of AMS:.....	86
Uninstallation of AMS: .....	86
Implementing Microsoft Container Technology with Windows Server 2016 .....	87
Windows Server Container .....	87
Hyper-V Container .....	90
HPE Storage .....	94
Resources.....	95

## Abstract

This technical white paper provides guidance and describes the support available for the Microsoft® Windows Server® 2016 operating system release on HPE ProLiant servers.

## Windows Server 2016 overview

At the heart of the Microsoft Cloud OS vision, Windows Server delivers global-scale cloud services into your infrastructure. Windows Server 2016 provides a wide range of new and enhanced features and capabilities spanning server virtualization, storage, software-defined networking, server management and automation, Web and application platform, access and information protection, virtual desktop infrastructure, and more. For details on the Windows Server 2016, access the links provided in the “Resources” section of this paper.

## Recommended system configurations

Microsoft has established the recommended system configurations listed in this section for Windows Server 2016 installations. Do not use this document as the sole source of information. For additional information about Windows Server 2016, access the links provided in the “Resources” section.

**Table 1.** Recommended system configuration as established by Microsoft.

<b>Processor</b>	<ul style="list-style-type: none"> <li>• Minimum: 1.4 GHz 64 bit</li> <li>• Recommended: 2 GHz or higher speed</li> <li>• Must support No-eXecute (NX) and Data Execution Prevention (DEP)</li> <li>• Must support virtualization               <ul style="list-style-type: none"> <li>– 1.4 GHz 64-bit processor</li> <li>– Compatible with x64 instruction set</li> <li>– Supports NX and DEP</li> <li>– Supports CMPXCHG16b, LAHF/SAHF, and PrefetchW</li> <li>– Supports Second Level Address Translation (EPT or NPT)</li> </ul> </li> <li>• Must be compatible with the 64-bit instruction set</li> </ul>
<b>RAM per processor (socket)</b>	<ul style="list-style-type: none"> <li>• Minimum: 512 MB (2GB for Server with Desktop experience installation option)</li> <li>• Must support either Error-Correcting Code (ECC) or another single error correcting capability</li> <li>• Maximum OS support: 12 TB</li> <li>• Recommended OS minimum: 4 GB</li> </ul>
<b>Display/Bit Depth</b>	A display is optional for Windows Server. If a display is used, it must have XGA resolution (1024x768) (or higher) with 32 bits of color per pixel
<b>Storage</b>	160 GB (or larger) with at least a 60 GB system partition <sup>1</sup>
<b>Optical storage</b>	DVD-ROM drive
<b>Networking</b>	<ul style="list-style-type: none"> <li>• 1 Gb (or faster) Ethernet controller</li> <li>• Preboot eXecution Environment (PXE) boot capable</li> <li>• Network Kernel Debugging (KDNet) support is recommended</li> <li>• Compliant with the PCI Express architecture specification</li> </ul>
<b>Peripherals</b>	<ul style="list-style-type: none"> <li>• Keyboard</li> <li>• Microsoft mouse or compatible pointing device</li> </ul>

<sup>1</sup> Available Storage Space is the free disk space on the partition that will contain the system files. Additional space is required to copy the Windows Server 2016 DVD contents to the disk during installation. Servers with more than 16 GB of RAM require more disk space for paging and for creating memory dump files in case of system crash.

## Supported ProLiant servers

Table 2 lists ProLiant servers and ROM versions that are certified for Windows Server 2016.

For a link to the ROM updates, see the HPE Support Center at [h20565.www2.hp.com/portal/site/hpsc](http://h20565.www2.hp.com/portal/site/hpsc).

**Table 2.** Supported ProLiant servers.

Server	Rom family	Rom date earliest
<b>HPE Blade Server</b>		
BL920s Superdome X Gen8	8.2.106	8.2.106
BL920s Superdome X Gen9	8.2.106	8.2.106
<b>ProLiant BL Blade Server</b>		
BL465c Gen8	A26	2016.03.07
BL420c Gen8	I30	2014.11.03
BL660c Gen9	I38	2.30 (09/12/2016)
BL460c Gen8	I31	2015.06.01
BL460c Gen9	I36	2.30 (09/12/2016)
BL660c Gen8	I32	2.30 (09/12/2016)
<b>ProLiant DL Rack servers</b>		
DL360p Gen8	P71	2015.07.01
DL380p Gen8	P70	2015.07.01
DL60 Gen9	U15	2.30 (09/12/2016)
DL80 Gen9	U15	2.30 (09/12/2016)
DL120 Gen9	P86	2.30 (09/12/2016)
DL160 Gen8	J03	08/02/2014
DL320e Gen8	J05	2013.11.09
DL320e Gen8 v2	P80	2015.04.02
DL360e Gen8	P73	2014.08.02
DL360p Gen8 SE	P71	2015.07.01
DL380e Gen8	P73	2014.08.02
DL385p Gen8	A28	2016.03.07
DL160 Gen9	U20	2.30 (09/12/2016)
DL180 Gen9	U20	2.30 (09/12/2016)
DL20 Gen9	U22	1.80 (09/12/2016)
DL360 Gen9	P89	2.30 (09/13/2016)
DL380 Gen9	P89	2.30 (09/13/2016)
DL560 Gen8	P77	2014.08.03
DL560 Gen9	P85	2.30 (09/12/2016)

**Table 2.** Supported ProLiant servers.

Server	Rom family	Rom date earliest
DL580 Gen8	P79	1.94_02-19-2016
DL580 Gen9	U17	2.30_07-20-2016
<b>ProLiant Tower servers</b>		
WS460c Gen8	I31	2015.06.01
WS460c Gen9	I36	2.30 (09/12/2016).
Microserver Gen8	J06	2015.11.02
ML30 Gen9	U23	1.80 (09/12/2016).
ML110 Gen9	P99	2.30 (09/12/2016).
ML150 Gen9	P95	2.30 (09/12/2016).
ML310e Gen8	J04	2013.11.09
ML310e Gen8 v2	P78	2014.03.28
ML350e Gen8	J02	2014.08.02
ML350e Gen8 v2	J02	2014.08.02
ML350p Gen8	P72	2015.07.01
ML350 Gen9	P92	2.30_07-28-2016
<b>ProLiant XL/Apollo servers</b>		
XL170r Gen9 (Apollo 2000)	U14	2.30 (09/12/2016).
XL190r Gen9 (Apollo 2000)	U14	2.30 (09/12/2016).
XL220a Gen8 v2	P94	2015.01.26
HPE Apollo 4200 Gen9	U19	2.30 (09/12/2016).
XL230a Gen9 (Apollo 6000)	U13	2.30 (09/12/2016).
XL250a Gen9 (Apollo 6000)	U13	2.30 (09/12/2016).
XL450 Gen9 (Apollo 45xx)	U21	2.30 (09/12/2016).
XL420 Gen9 (Apollo 4200)	U19	2.30 (09/12/2016).
<b>ProLiant SL servers</b>		
SL230s Gen8	P75	2015.07.01
SL250s Gen8	P75	2015.07.01
SL270s Gen 8	P75	2015.07.01
SL270s Gen8 SE	P75	2015.07.01
SL210t Gen8	P83	2016.01.18
<b>HPE Synergy</b>		
HPE Synergy 620	I40	2.20 (09/08/2016)
HPE Synergy 680 Gen9	I40	2.20 (09/08/2016)
HPE Synergy 480 Gen9	I37	2.20 (09/14/2016)

**Table 2.** Supported ProLiant servers.

Server	Rom family	Rom date earliest
HPE Synergy 660 Gen9	I39	2.20 (09/08/2016)
<b>Moonshot</b>		
ProLiant m510 Server Cartridge		05_09_2016
ProLiant m710 Server Cartridge		07/13/2016
ProLiant m710p Server Cartridge		07/13/2016
ProLiant m710x Server Cartridge		05/23/2016

## Supported ProLiant server platform options

Before you install Windows Server 2016 on a ProLiant server, review the following sections for information on the ProLiant server platform options for which HPE provides Windows Server 2016 support.

### Software and drivers

Currently, many drivers for storage options, network interface controllers (NICs), and converged network adapters (CNAs) are available on the Windows Server 2016 media. Some HPE drivers will be made available in the Service Pack for ProLiant (SPP) version 2016.10.0.

Download the Service Pack for ProLiant (SPP) Version 2016.10.0 at: [hpe.com/servers/spp/download](http://hpe.com/servers/spp/download)

For details, see “[Installing components from the HPE Service Pack for ProLiant \(SPP\) version 2016.10.0](#)”

**Table 3.** Recommended HPE ProLiant storage controller options.

Option	Driver
<b>Smart Storage Controllers and drivers</b>	
SAS/SATA Notification Service	CISSESrv.EXE
P222	HPCISS3.SYS
P220i	HPCISS3.SYS
P420	HPCISS3.SYS
P420i	HPCISS3.SYS
P421	HPCISS3.SYS
P721m	HPCISS3.SYS
P822	HPCISS3.SYS

**Table 3a.** HPE Smart Array controller options

Option	Driver
H240	HPCISS3.SYS Supports both legacy BIOS mode and UEFI.
H240ar	HPCISS3.SYS
H241	HPCISS3.SYS
H240nr	HPCISS3.SYS
H44BR	HPCISS3.SYS

**Table 3a.** HPE Smart Array controller options

Option	Driver
P230i	HPCISSS3.SYS
P244br	HPCISSS3.SYS
P246br	HPCISSS3.SYS
P430	HPCISSS3.SYS
P430i	HPCISSS3.SYS
P431	HPCISSS3.SYS
P440	HPCISSS3.SYS
P440ar	HPCISSS3.SYS
P441	HPCISSS3.SYS
P731m	HPCISSS3.SYS
P741m	HPCISSS3.SYS
P830	HPCISSS3.SYS
P840	HPCISSS3.SYS
P841	HPCISSS3.SYS
B140i	HPSA3.SYS
B120i	HPSA2.SYS
B320i	HPSA2.SYS
P240nr	HPCISSS3.SYS
P246br	HPCISSS3.SYS
P542D	HPCISSS3.SYS

**HPE Smart HBA Controller Options**

H220	LSI_SAS2.SYS
H221	LSI_SAS2.SYS
H222	LSI_SAS2.SYS
H210i	LSI_SAS2.SYS
H220i	LSI_SAS2.SYS

**HPE Converged Network Adapters**

536FLB	evbda.sys
530T	bxnd60a.sys
HPE Synergy 3820C 10/20Gb Converged Network Adapter	evbda.sys/bxnd60a.sys
HPE Synergy 2820C 10Gb Ethernet Adapter	evbda.sys/bxnd60a.sys
533FLR-T	
630FLB/M	
530FLR-SFP+/SFP	



**Table 3a.** HPE Smart Array controller options

Option	Driver
534FLB/M/FLR/-SFP+/SFP+	
650FLB/M	ocnd65.sys
CN1100E	
CN1100R-T	evbda.sys/bxnd60a.sys
554FLR-SFP+/FLB/M*	
552M/SFP	
546SFP+/FLR-SFP+	mlx4_bus.sys mlx4eth63.sys

## Network Interface Controllers

We recommend the NICs listed in Table 4 for Windows Server 2016. Currently, all drivers for NICs are available on the Windows Server 2016 media.

**Table 4.** Recommended NICs.

NIC	Inbox driver
330i	b57nd60a.sys
331i/T/FLR	
331i-SPI	
332T	
361i/FLB/T	e1i63x64.sys
366i/FLR/M	
367i	
363i	
364i	ixi63x64.sys
560SFP+/FLR-SFP+/FLB/M	
561FLR-T/T	
562i	
562FLR-SFP+	l40ea65.sys
562SFP+	
640FLR-SFP28	Mlx5.sys
640SFP28	
Mellanox CX4	
620QSFP28	qevbda.sys, qenda.sys
840QSFP28	qevbda.sys, qenda.sys
<b>HPE SAS JBOD Enclosures</b>	
D3600	
D3700	
D6020	

## Persistent Memory

Hewlett Packard Enterprise has developed Non-Volatile DIMM (NVDIMM) technology, a Storage Class Memory medium that effectively combines system memory performance with true power-off data storage. HPE ProLiant DL360 and DL380 Gen9 servers shipped with Intel® XEON E5-V4 processors are equipped to use HPE NVDIMMs when running a supportive operating system. Microsoft® has announced native support of Storage Class Memory with Windows® Server 2016 for using HPE NVDIMMs as Windows storage solutions. This paper discusses the Microsoft Windows Server 2016 support of HPE NVDIMMs and describes recommended scenarios for their implementation.

For the latest information, please refer to the HPE Persistent Memory website, [hpe.com/servers/persistentmemory](http://hpe.com/servers/persistentmemory) and review the following “Persistent Memory Resources” section for additional references.

### Introduction

Hewlett Packard Enterprise introduced HPE NVDIMM technology, a Persistent Memory medium that provides the extreme performance of DRAM components and the persistency of NAND flash-based components for emergency storage. Close (memory bus) association with the system’s processor allows HPE NVDIMMs to be used as extreme-performance workload accelerators in the top tier of a storage hierarchy. JEDEC defines these NVDIMM devices as NVDIMM-N energy-backed devices.

Microsoft has developed Windows Server 2016 with native support of Persistent Memory as Storage Class Memory (SCM). This capability allows:

- Zero-copy access to SCM
- Most existing user-mode applications to be run without modification
- Sector granular failure modes to preserve application compatibility

The initial Persistent Memory enablement offered by Windows Server 2016 through its SCM support allows it to be implemented two different ways:

- With a block-interface overlay, using SCM as block-storage devices applications can readily use no differently than SATA HDDs or SAS SSDs.
- With a byte-addressable memory interface, allowing applications directly accessing physical memory locations on the SCM device using Direct Access Storage (DAX), enabled on a per-volume basis.

### Hardware and Firmware requirements

HPE NVDIMMs must be deployed on HPE platforms designed for NVDIMM functionality. At the general release of Windows Server 2016, the following HPE ProLiant platforms offer NVDIMM functionality:

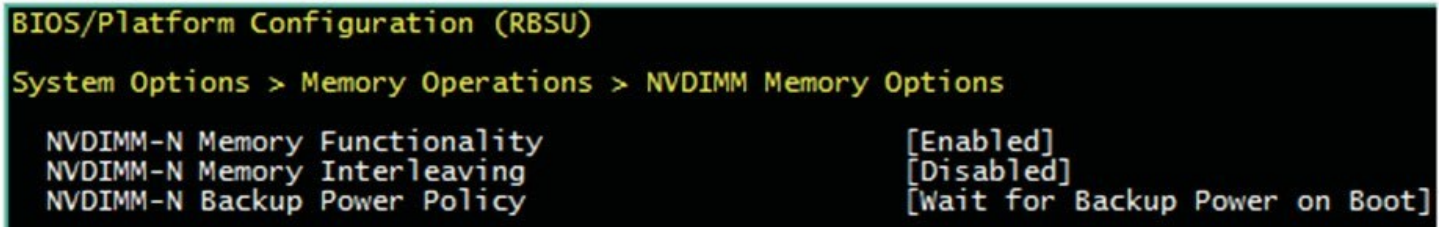
- HPE ProLiant DL380 Gen9 servers shipped with Intel® XEON E5-V4 processors
- HPE ProLiant DL360 Gen9 servers shipped with Intel® XEON E5-V4 processors
- Server with HPE Smart Storage Battery
- Server with a minimum of one registered DIMM (RDIMM) per processor
- HPE ProLiant Gen9 ROM firmware 2.30 or later
- HPE iLO 4 Firmware v2.50 Pass 46 or later

In 2017, HPE Servers will greatly expand the support of Persistent Memory for the HPE ProLiant Gen10 servers and other HPE Servers platforms.

## Hardware and Firmware configuration

To configure a platform for HPE NVDIMM operation, proceed as follows:

1. Install HPE NVDIMMs as described in the “Installation” section of the HPE NVDIMM User Guide for HPE ProLiant Gen9 Servers P/N 860023-001.
2. Verify or Install/update the server BIOS and iLO firmware images as required.
3. Reboot system and press F9 at the POST display to go to RBSU (via System Utilities).
4. In RBSU, enable NVDIMM (as shown in Figure 1) and, if appropriate, sanitize NVDIMM. For more information on the NVDIMM sanitize, refer to the HPE NVDIMM user guide.



**Figure 1.** RBSU control of HPE NVDIMM devices.

---

### Note

In addition to the support of individual HPE NVDIMM devices, HPE Host BIOS allows the interleaving of NVDIMM devices within a processor socket, if desired.

---

5. Press F10 to save configuration and press Esc several times to return to the System Utilities menu and Enter to reset the server and boot Windows Server 2016.

As part of its server management of the HPE ProLiant servers and presented in Figure 2, HPE Integrated Lights Out (iLO) reports the enumeration and health of the HPE NVDIMM devices in the System Information Memory pane and in the Integrated Management Log records.

**iLO 4**  
ProLiant DL380 Gen8

System Information - Memory Information

Summary Fans Temperatures Power Processors **Memory** Network Device Inventory Storage Firmware Software

**Advanced Memory Protection (AMP)**

**AMP Status**

AMP Mode Status: **Advanced ECC**

Configured AMP Mode: **Advanced ECC**

**Supported AMP Modes**

Advanced ECC  
Online Spare (Rank Sparing)  
Intrsocket Mirroring

**Memory Summary**

Location	Number of Sockets	Total Memory	Operating Frequency	Operating Voltage
Processor 1	12	80 GB	1866 MHz	1.2 V
Processor 2	12	N/A	N/A	N/A

**Memory Details** (show empty sockets)

Memory Location	Socket	Status	HPE Memory	Part Number	Type	Size	Maximum Frequency	Minimum Voltage	Ranks	Technology
Processor 1	1	Good, In Use	HPE SmartMemory	752369-081	DIMM DDR4	16384 MB	2133 MHz	1.2 V	2	RDIMM
Processor 1	2	Good, In Use	HPE SmartMemory	N/A	DIMM DDR4	8192 MB	2133 MHz	1.2 V	1	R-NVDIMM
Processor 1	4	Good, In Use	HPE SmartMemory	752369-081	DIMM DDR4	16384 MB	2133 MHz	1.2 V	2	RDIMM
Processor 1	9	Good, In Use	HPE SmartMemory	752369-081	DIMM DDR4	16384 MB	2133 MHz	1.2 V	2	RDIMM
Processor 1	11	Good, In Use	HPE SmartMemory	N/A	DIMM DDR4	8192 MB	2133 MHz	1.2 V	1	R-NVDIMM
Processor 1	12	Good, In Use	HPE SmartMemory	752369-081	DIMM DDR4	16384 MB	2133 MHz	1.2 V	2	RDIMM

POWER: ON UID: OFF

**Figure 2.** The iLO4 Memory Information pane reports the HPE R-NVDIMM devices.

## Windows Server 2016 Storage Class Memory (SCM) enablement

Windows Server 2016 supports in-box the HPE NVDIMM devices. As shown in Figure 3, NVDIMMs are presented in the Windows Device Manager as Persistent Memory Disk devices.

## Notes

Windows Virtualization of SCM devices is not a solution of Windows Server 2016.

In addition to the BIOS interleaving support of HPE NVDIMMs, Windows Server components like Storage Spaces can be used to group NVDIMM devices together and create larger simple or mirrored SCM volumes.

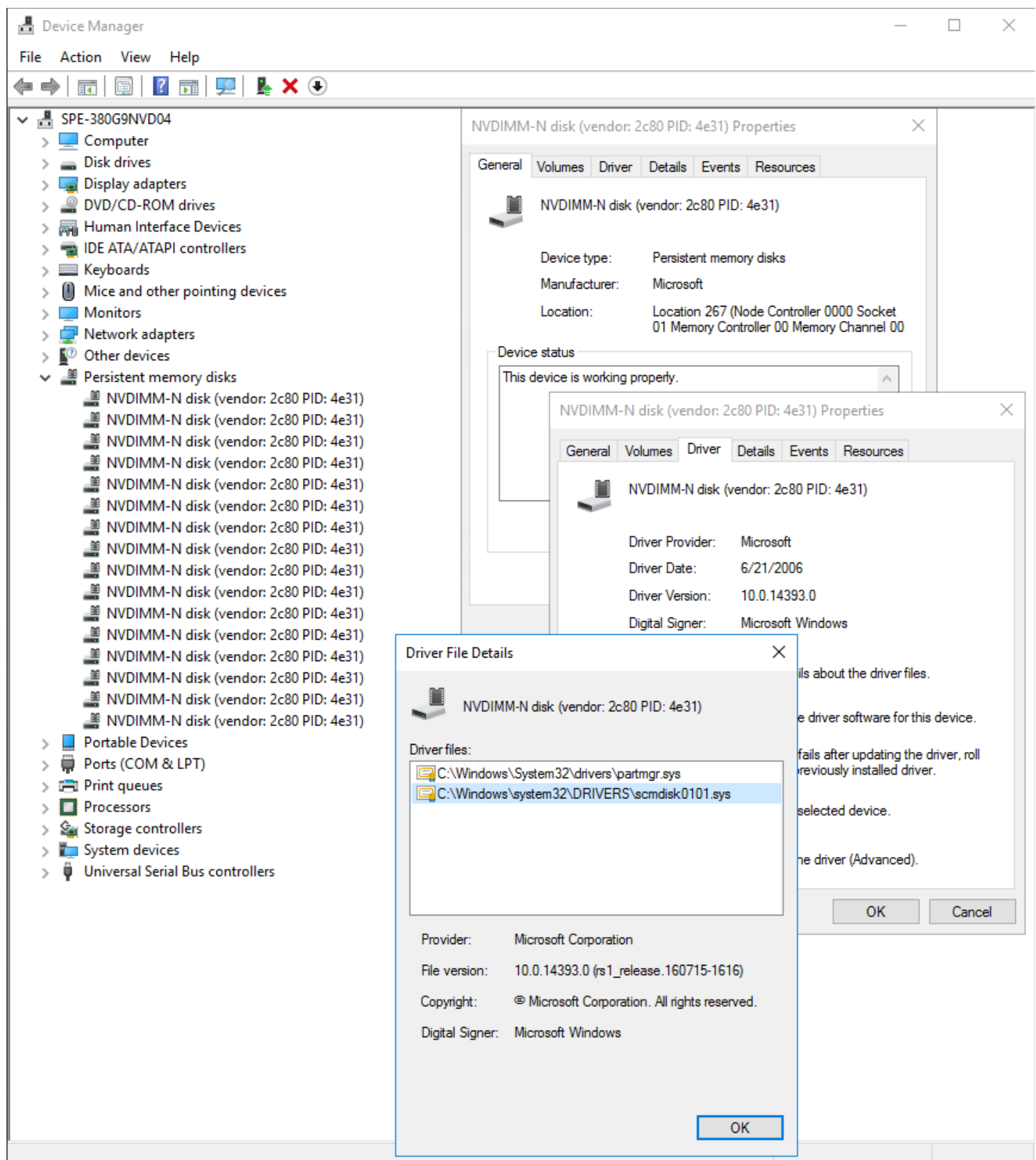


Figure 3. Windows Device Manager screen showing system configuration of 16 HPE NVDIMMs.

A storage volume and then a Windows File System (NTFS or ReFS) can be created on SCM devices. At mount time, the File System detects if a given volume resides on SCM hardware. If this is the case, a decision must be made if an SCM volume should be formatted for Direct Access Storage (DAX) mode or traditional block mode (the default being traditional block mode).

Figure 4 shows how to format a SCM device for DAX through the new **/DAX** parameter of the format command. The Block vs. DAX type can be then determined with the following command:

```
cmd> fsutil fsinfo volumeinfo <driveletter>
```

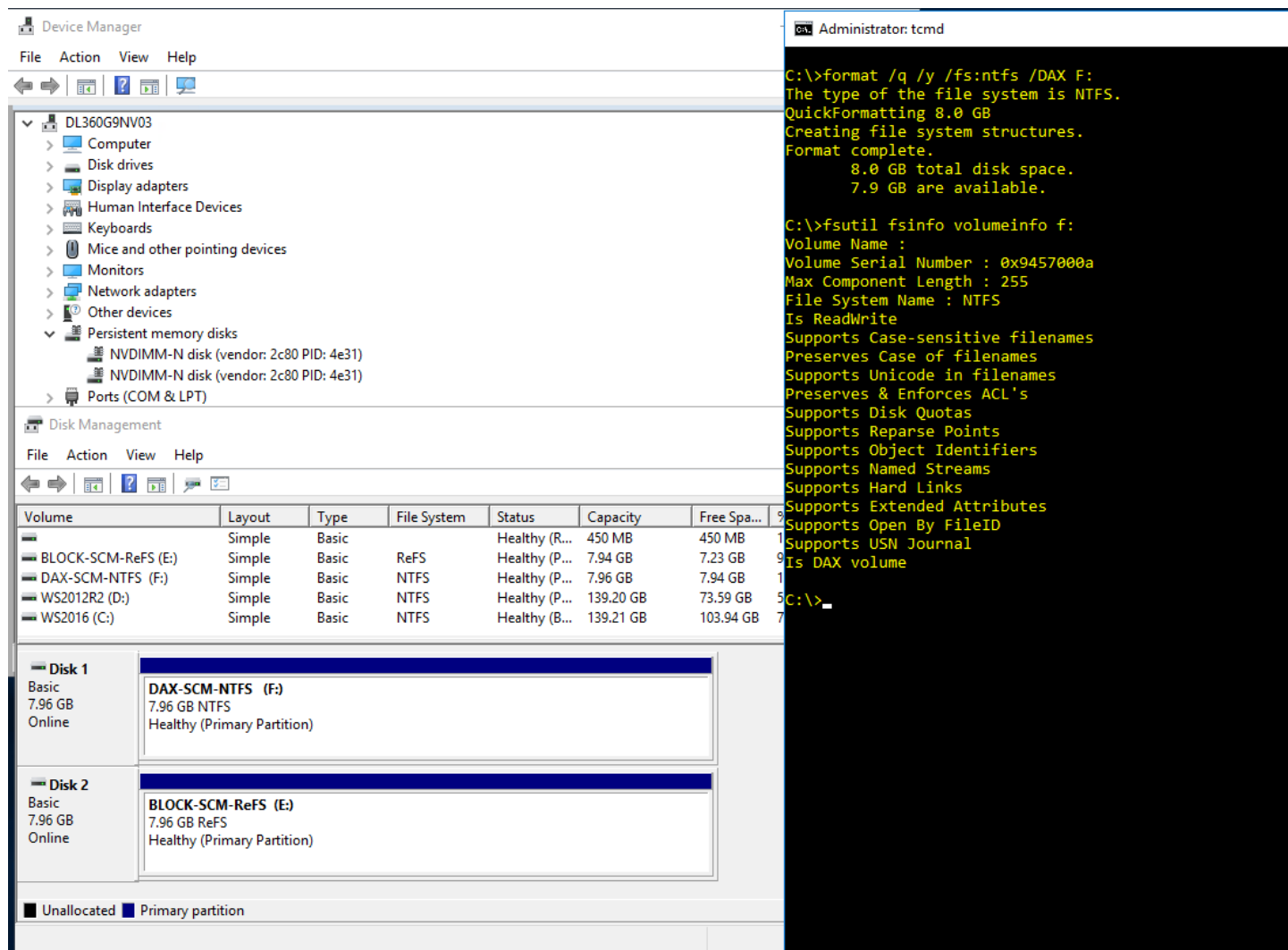
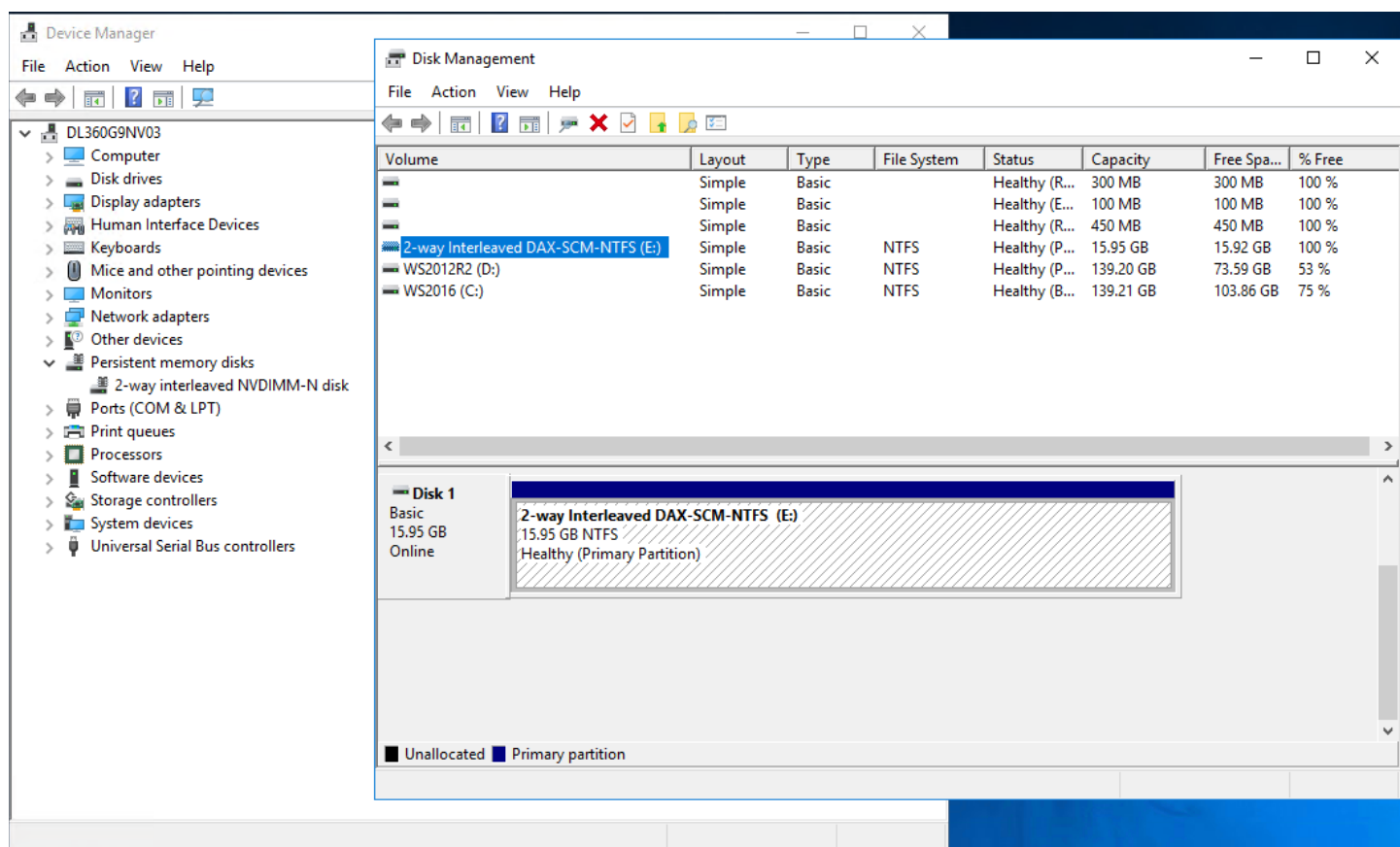


Figure 4. Windows Disk Manager and cmd window screens showing selection of DAX mode for SCM volume format.

Figure 5 Windows Device Manager and Disk Manager screens showing a 2-way interleaved SCM device and the corresponding SCM volume with DAX enabled NTFS file system



**Figure 5.** Windows Device Manager and Disk Manager screens showing a 2-way interleaved SCM device and the corresponding SCM volume with a DAX enabled NTFS file system.

### Windows SCM drivers

Microsoft has implemented a new driver model based on two types of SCM drivers:

- **SCM Bus Driver:** This driver is responsible for the enumeration of physical and logical SCM devices on the host. It should be noted that for optimization the SCM Bus Driver is not part of the IO path.
- **SCM Disk Driver:** An SCM disk driver exists for each SCM disk type and implements the storage abstraction layer to the OS. The SCM disk driver can be in-box or vendor-specific. For HPE NVDIMM devices, the driver is in-box. Windows uses a native 4-KB sector size for the SCM devices.

These SCM drivers enable byte addressable storage and management of the SCM devices.

### SCM Block Volumes

With this default mode at supporting SCM devices, Windows Server 2016 provides full compatibility with existing applications and maintains existing storage semantics. All IO operations traverse the storage stack to the SCM NVDIMM driver. It supports the Windows File Systems and storage filters. As expected, Disk Management and PowerShell cmdlets initialize the SCM disk, create the new block volume and format the volume with a File System.

### Notes

With Windows Server 2016, creating a software RAID array of SCM devices using Disk Management is not supported.

## DAX Volumes

Windows Server 2016 introduces DAX Volumes, a new class of volumes enabling DAX mode: Memory-mapped files provide applications with direct access to byte-addressable SCM to maximize performance. DAX mode is chosen at volume format time.

---

### Note

With Windows Server 2016, DAX mode is not supported with ReFS.

Compatibility issues might exist with specific components like file system filters, bitlocker and volsnap. Please refer to the following [“Persistent Memory Resources”](#) section for additional references and to Microsoft online updates on their support of SCM devices.

---

DAX mode can use three possible IO paths:

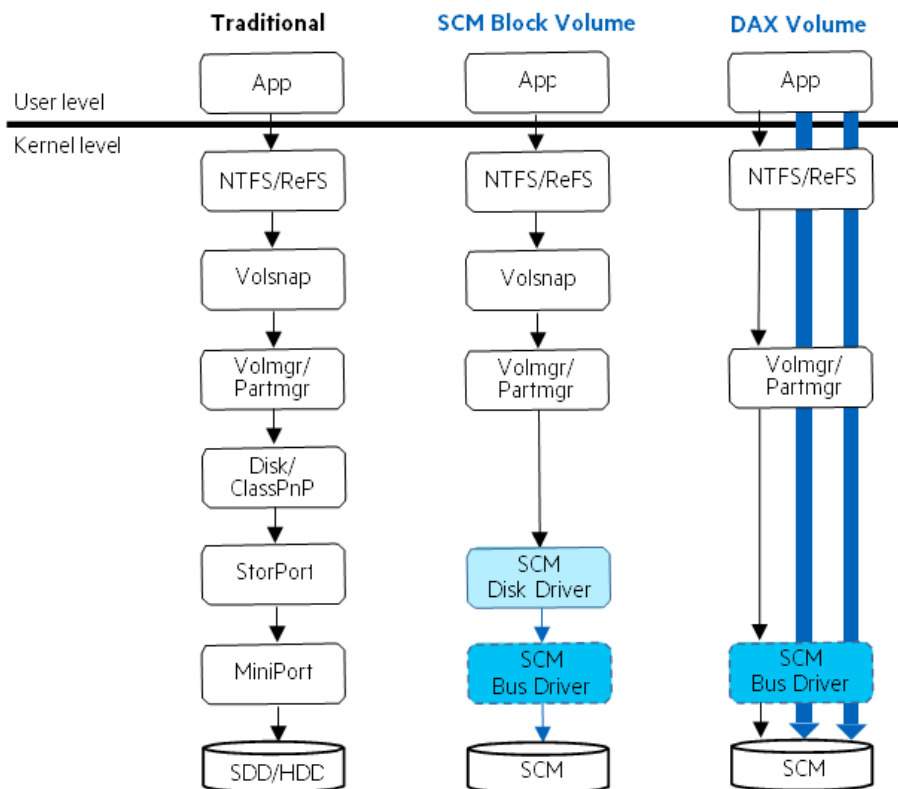
- Memory Mapped I/O:
  - Memory-mapped sections map directly to the SCM devices.
  - Offers true zero-copy access to storage and allows an application direct access to persistent memory.
  - There are no paging reads or writes.
- Cached I/O
  - For an application request for cached I/O on a DAX enabled volume, the Windows cache manager creates a cache map that maps directly to the SCM device.
  - The cache manager copies directly between the user application's buffer and the persistent memory.
  - Cached I/O has one-copy access to persistent memory.
- Non-Cached I/O:
  - I/O operations are sent down the software storage stack to the SCM device disk driver.
  - Existing failure semantics for application compatibility are maintained.

With Windows Server 2016, file system metadata doesn't use DAX mode sections. This results in paging reads and writes needed to maintain existing ordered write guarantees for write-ahead logging support.

Please refer to the Microsoft documentation for the other impacts to File System functionalities by the Windows Server 2016 SCM DAX mode.



Figure 6 compares Block and DAX volume stacks with that of traditional storage volumes.



Source: Microsoft

**Figure 6.** Software I/O stack comparison.**DAX Volume Creation**

The following commands allow the creation of a DAX Volume on a SCM device with E: being used as an example of a driver letter:

- Format E: /dax /q
- PS> Format-Volume -DriveLetter F -IsDAX \$true

**DAX Volume Identification**

The following Win32 APIs and command allow the identification of a DAX Volume:

- Application wanting to identify a DAX Volume
  - calls GetVolumeInformation("F:", ...)
  - checks lpFileSystemFlags for FILE\_DAX\_VOLUME (0x20000000)
- cmd window
  - fsutil fsinfo volumeinfo F:
- Application wanting to know if a specific file is on a DAX Volume
  - Calls GetVolumeInformationByHandleW(hfile, ...)
  - checks lpFileSystemFlags for FILE\_DAX\_VOLUME (0x20000000)

## Windows Server 2016 SCM device Management with Powershell

In addition to Windows Server Device Manager and Disk Manager, Powershell commands like `Get-PhysicalDisk` and `Reset-PhysicalDisk` allow health control of the SCM devices. Figure 7 presents their use with a transient error from a SCM device.

```

Administrator: Windows PowerShell

PS C:\Users\Administrator> get-physicaldisk

FriendlyName      SerialNumber      CanPool OperationalStatus HealthStatus Usage      Size
-----
HP LOGICAL VOLUME PDNLH0BRH7Y78L    False    OK              Healthy      Auto-Select 279.37 GB
Vendor 2c80 PID 4e31 802c-01-1602-117cb620 True      OK              Healthy      Auto-Select 8 GB
Vendor 2c80 PID 4e31 802c-01-1602-117cb64c True      Transient Error Unhealthy     Auto-Select 8 GB

PS C:\Users\Administrator> (get-physicaldisk)[2] | fl

ObjectId           : {1}\DL360G9NV03\root\Microsoft\Windows\Storage\Providers_v2\SPACES_PhysicalDisk.ObjectId="fd5b221e-5d99-11e6-8b3b-806e6f6e6963":PD:{c8368a78-dc88-64d9-3b34-9f95c76d9018}"
PassThroughClass   :
PassThroughIds     :
PassThroughNamespaces :
PassThroughServer  :
UniqueId           : {c8368a78-dc88-64d9-3b34-9f95c76d9018}
Description         :
FriendlyName       : Vendor 2c80 PID 4e31
HealthStatus       : Unhealthy
Manufacturer       : Vendor 2c80
Model              : PID 4e31
OperationalDetails : {Lost Data Persistence, NVDIMM_N Error}
OperationalStatus  : Transient Error
PhysicalLocation   : DIMM Socket 0 : Slot 11
SerialNumber       : 802c-01-1602-117cb64c
AdapterSerialNumber :
AllocatedSize      : 0
BusType            : SCM
CannotPoolReason   :
CanPool            : True
DeviceId           : 1
EnclosureNumber    :
FirmwareVersion    : 288
IsIndicationEnabled :
IsPartial          : False
LogicalSectorSize  : 4096
MediaType          : SCM
OtherCannotPoolReasonDescription :
PartNumber         :
PhysicalSectorSize : 4096
Size               : 8589934592
SlotNumber         :
SoftwareVersion    :
SpindleSpeed       : 0
SupportedUsages    : {Auto-Select, Manual-Select, Hot Spare, Retired...}
UniqueIdFormat     : Vendor Specific
Usage              : Auto-Select
VirtualDiskFootprint : 0
PSComputerName     :
ClassName          : MSFT_PhysicalDisk

PS C:\Users\Administrator> (get-physicaldisk)[2] | reset-physicaldisk
PS C:\Users\Administrator> get-physicaldisk

FriendlyName      SerialNumber      CanPool OperationalStatus HealthStatus Usage      Size
-----
HP LOGICAL VOLUME PDNLH0BRH7Y78L    False    OK              Healthy      Auto-Select 279.37 GB
Vendor 2c80 PID 4e31 802c-01-1602-117cb620 True      OK              Healthy      Auto-Select 8 GB
Vendor 2c80 PID 4e31 802c-01-1602-117cb64c True      OK              Healthy      Auto-Select 8 GB

PS C:\Users\Administrator> _
  
```

Figure 7. Powershell PhysicalDisk commands handling a SCM device with a transient error.

## Notes

A Windows Server software RAID array (RAID-5) of SCM devices using Windows Server 2016 Disk Management is not supported. Storage Spaces allows grouping SCM devices together and create larger simple or mirrored SCM volumes.

## Windows Server 2016 Operational and Diagnostics Channels for SCM devices

Windows Server 2016 reports operational and diagnostic information for SCM devices in the Windows Server Computer Management Services Logs: SCMBus and SCMDisk0101. Figure 6 shows the Windows Server Computer Management presentations of these channels.

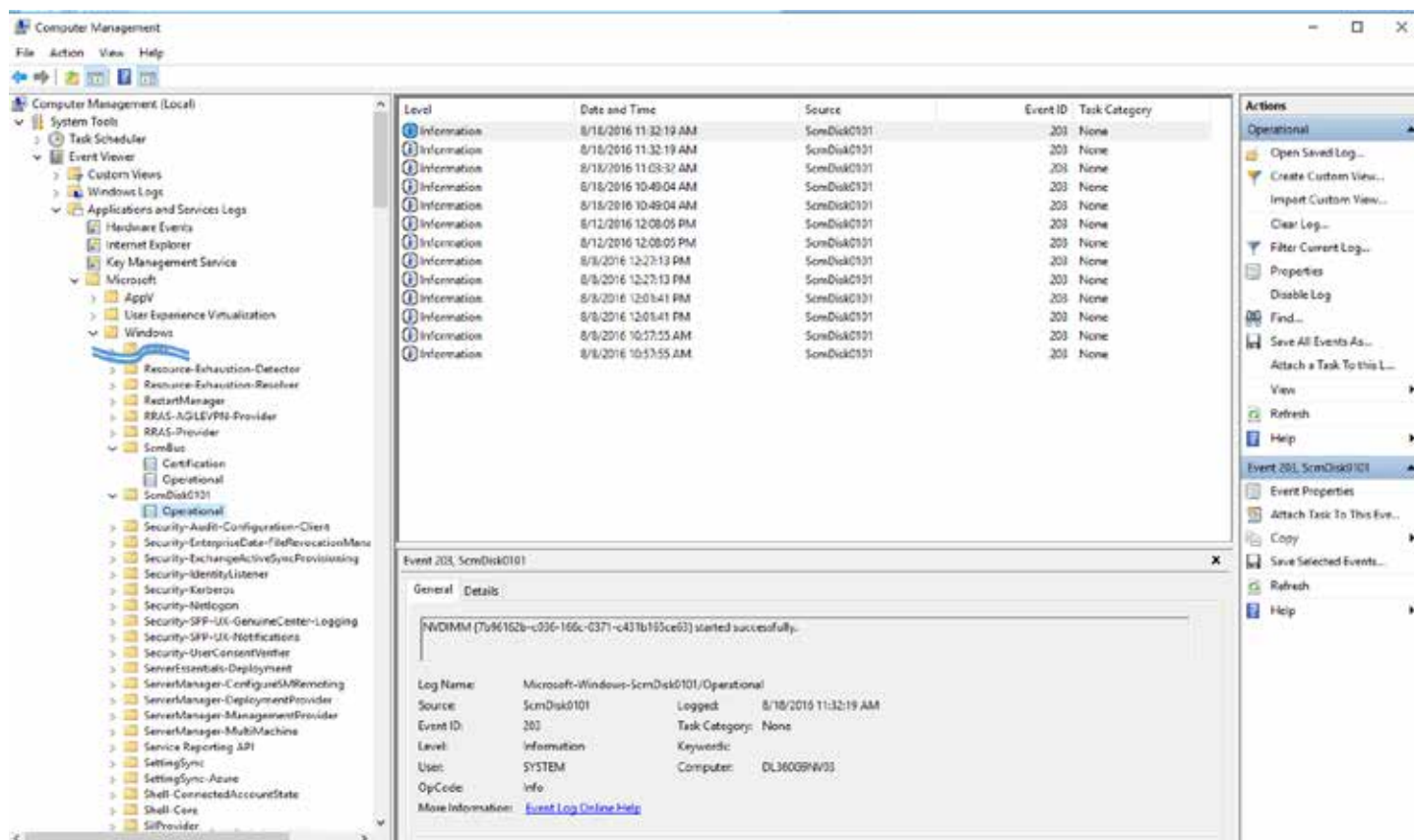


Figure 8. Windows Server 2016 operational and diagnostics channels for SCM devices in Windows Server Computer Management.

## HPE NVDIMM use scenarios with Windows Server 2016

HPE NVDIMM offers various uses for applications in the Windows Server 2016 environment, like:

- SQL server can store transaction logs in the NVDIMM devices
- Exchange server can store logs in the NVDIMM device

This section describes also two scenarios of interest for using HPE NVDIMM with Windows Server 2016:

- Mirrored Storage Space
- SCM DAX

### Mirrored Storage Spaces scenario

This scenario presents a Mirrored Persistent Write-Back cache in front of a Mirrored SSD pool, as illustrated in Figure 9.

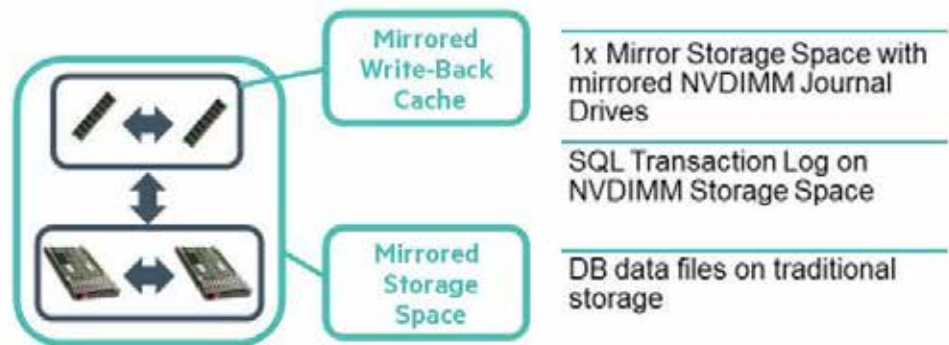
#### Hardware

- Gen9 DL380/380
- 2x HPE 8GiB NVDIMM-Ns
- 2x 400 GiB SSD



#### software

- Windows Server 2016
- Storage Spaces
- SQL2016



**Figure 9.** Mirrored storage space: SQL server SCM with resilience.

To configure NVDIMMs for mirrored storage space, we present as an example these steps using PowerShell cmdlets:

1. Identify NVDIMMs and SSDs:

```
# List physical disks
```

```
PS> Get-PhysicalDisk
```

```
# List disks
```

```
PS> get-disk
```

```
# Ensure there is no partition or any configuration on the disk
```

```
PS> clear-disk -Number <disknumber> -RemoveOEM -RemoveData
```

2. Create a new storage pool with all identified devices:

```
PS> $pd = Get-PhysicalDisk -CanPool $true
```

```
PS> new-storagepool -StorageSubSystemFriendlyName *Storage* -FriendlyName SCM_Pool -PhysicalDisks $pd
```

3. For Windows Server 2016: Change the SSD Media Type to “HDD” by entering:

```
PS> Get-PhysicalDisk -UniqueId 600508B1001CD8368AB7B7FC46D47A07 | Set-PhysicalDisk -MediaType HDD
```

4. Set NVDIMM Usage to “Journal”:

```
# use Get-PhysicalDisk to identify the Friendly Names of the NVDIMM devices
```

```
PS> Get-PhysicalDisk
```

```
PS> Get-PhysicalDisk -FriendlyName 2c* | Set-PhysicalDisk -Usage Journal
```

5. Create the Storage Spaces disk:

```
PS> New-VirtualDisk -StoragePoolFriendlyName SCM_Pool -FriendlyName SCM_Mirror -ResiliencySettingName Mirror -UseMaximumSize -ProvisioningType Fixed
```

## 6. Initialize the SCM Disk:

```
# Get the SCM disk number using Get-Disk
```

```
PS> Get-Disk
```

```
# Initialize SCM disk
```

```
PS> Initialize-Disk -Number <disknumber> -PartitionStyle GPT
```

## 7. Format the filesystem:

```
PS> $partition = New-Partition -DiskNumber <disknumber> -AssignDriveLetter -UseMaximumSize
```

```
PS> Format-Volume -Partition $partition -FileSystem NTFS
```

## 8. List the Windows Disks:

```
PS> Get-Disk
```

Output example:

No.	Friendly Name	Serial Number	Health Status	Op Status	Total Size	Partition
3	HPE LOGICAL	PDNLH0BRH9B8I1	Healthy	Online	279.37 GB	GPT
6	SCM_Mirror	{2faa7a8f-eb4c-472f-acc6-dc2e...	Healthy	Offline	371 GB	RAW

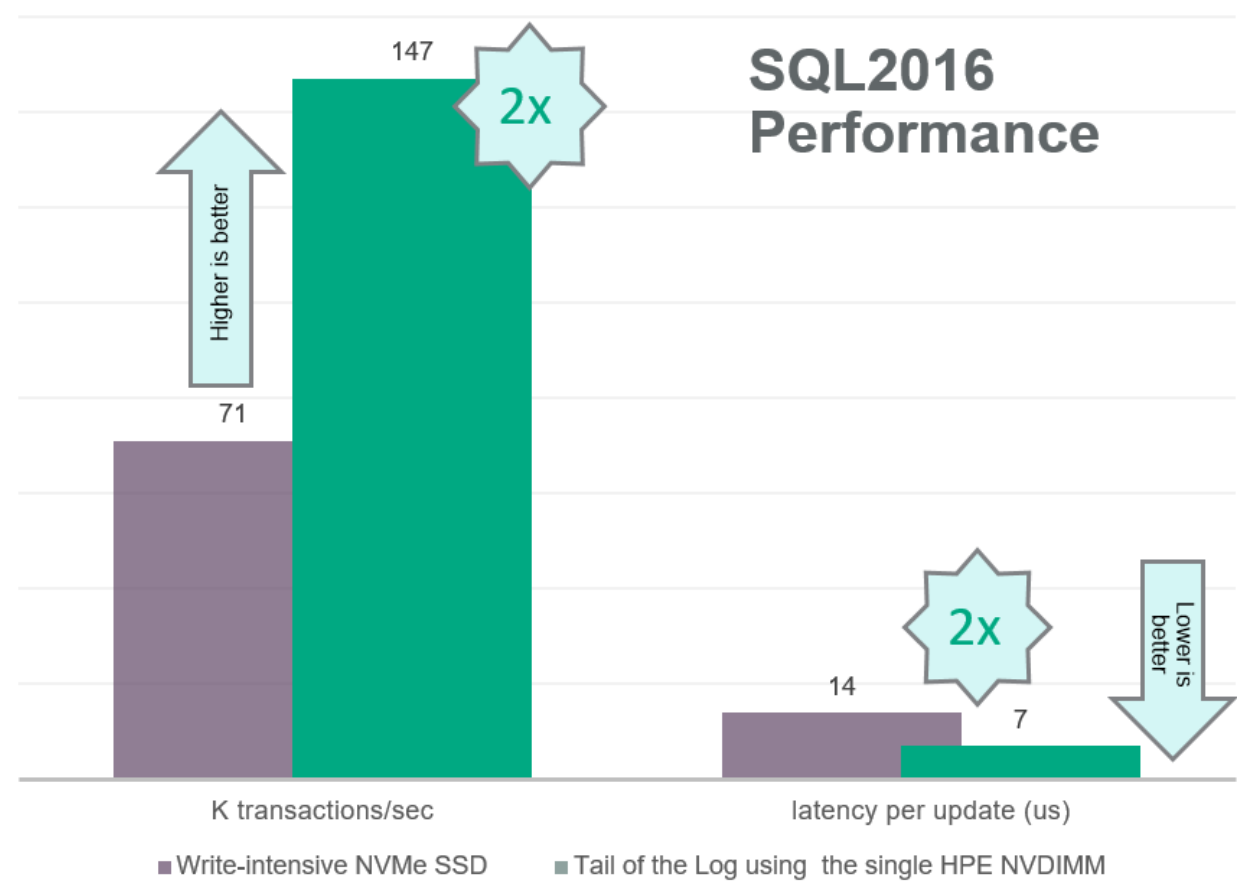
**SCM DAX scenario**

Microsoft SQL Server 2016 introduces its first DAX Volume support with its Tail of Log (or Persistent Main Memory extensions to the SQL Server Log Manager) feature, using a single HPE NVDIMM and de-staging to a SATA, SAS HDDs or SSDs or NVMe SSDs.

With a server configuration as:

- 1x HPE ProLiant DL380 Gen9 (both sockets populated)
- 1x HPE 8GB NVDIMM—for the tail of the log
- 2x SATA SSD (400 GB)—as the store for Database files
- 1x Write-Intensive NVMe SSD (400 GB)—as the store for both logs
- 128 GB Memory
- Software: Windows Server 2016 and SQL Server 2016 RTM
  - SQL Tables are stored on 2x SATA SSDs that are striped (Simple Space)
  - SQL Tail of the Log enabled
  - Table Size configured to match Data and Log storage capacities
  - Threads: 1 per Windows logical processor
  - SQL queries: Create, Insert, Update
  - SQL Performance Collectors: None
  - Batch Size: 1
  - Row Size: 32B

Figure 10 presents the performance results:



**Figure 10.** SQL Server 2016 update transaction performance improvements with its Tail Of Log enablement using HPE Persistent Memory on Windows Server 2016 SCM.

Log solution	Transactions/Sec	Latency per update
NVMe SSD	71K	14 us
Tail of the Log on the single HPE NVDIMM	147K (2x)	7us (2x)

Please refer to the resources presented in the following “[Persistent Memory Resources](#)” section for more information.

## HPE NVDIMM Performance with Windows Server 2016

### Windows Server 2016 SCM Block Volume performance with Microsoft DiskSpd tool

Microsoft DiskSpd is a tool that can simulate many types of workloads in various configurations. The next tables show results of running DiskSpd on the following Server configuration:

Server Model: HPE ProLiant DL360 Gen9

Memory: Six DIMMs and two HPE NVDIMMs (the test accessed only one HPE NVDIMM)

File System1: NTFS SCM Block Volume on one HPE NVDIMM

File System2: NTFS Block Volume on one SATA SSD

Large area of sequential concurrent writes of 64KB blocks. 1 outstanding IOs. Disable both software caching and hardware write caching:

**diskspd.exe -c4G -w100 -b64k -o1 -F4 -T1b -s8b -d10 -h file**

Storage type	Bytes	IOs	MB/s	IOPs
SCM Block Volume	59915567104	914239	5713.89	91422.17
SATA SSD	4767875072	72752	454.69	7275.04

Large area of random concurrent reads of 8-KB blocks. Disable both software caching and hardware write caching:

**diskspd.exe -c4G -r -w0 -b8k -o32 -t8 -d10 -h file**

Storage type	Bytes	IOs	MB/s	IOPs
SCM Block Volume	59640225792	7280301	5687.74	728031.31
SATA SSD	4482473984	547177	427.47	54716.78

Large area of random concurrent writes of 64-KB blocks. Disable both software caching and hardware write caching:

**DiskSpd.exe -c4G -d10 -r -w100 -t8 -o32 -b64K -h file**

Storage type	Bytes	IOs	MB/s	IOPs
SCM Block Volume	81534582784	1244119	7775.57	124409.18
SATA SSD	4595187712	70117	438.22	7011.60

Large area of random concurrent 60% reads and 40% writes of 8-KB blocks. Disable both software caching and hardware write caching:

**DiskSpd.exe -c4G -d10 -r -w40 -t8 -o32 -b8K -h file**

Storage type	Bytes	IOs	MB/s	IOPs
SCM Block Volume	22215933952	2711906	2118.63	271184.18
SATA SSD	1521860608	185774	145.13	18576.96

### Windows Server 2016 SCM DAX Volume performance

The following table presents IO performance results of running a simple application with DAX Volume awareness. Server configuration:

Server Model: HPE ProLiant DL360 Gen9

Memory: Six DIMMs and two HPE NVDIMMs (the test accessed only one HPE NVDIMM)

File System: NTFS DAX Volume on one HPE NVDIMM

	IODEPTH	READS/S	MB/S	WRITES/S	MB/S	
IOPs	1	3.5 M	13441	2.7 M	10494	10 threads on local CPU, 4KB random
	16	3.5 M	13462	2.7 M	10483	
BW	1	57.7 K	14090	37.3 K	9322	10 threads on local CPU, 256KB random
	16	59.0 K	14759	37.5 K	9386	

### Note

It is important to note that in case of errors during NVDIMMs backup, erase or restore operations on that device (DRAM to/from NAND), this type of failure is reported in the HPE ProLiant iLO management log (as well as the Windows Server 2016 diagnosis channels), the device will be unavailable after reboot and data will be lost.

HPE recommends to backup data on any storage device including NVDIMMs, for which in rare circumstances this failure can occur.

## SCM Device Management in Nano Server using Windows PowerShell

SCM devices are supported in Nano Server and are managed using Windows PowerShell commands. Below is a walkthrough on how to configure SCM devices in Nano Server.

### Prerequisites

Be sure to review the Persistent Memory section above, ensure your server meets the hardware and firmware requirements, and you've configured the platform for HPE NVDIMM operation as described in the section, **Hardware and Firmware configuration**.

This section assumes the reader is familiar with creating and deploying Nano Server images as described in this paper. It is also assumed the reader is familiar with managing Nano Server using PowerShell Remoting. For more information on this topic, see the Microsoft article <https://technet.microsoft.com/en-us/windows-server-docs/compute/nano-server/getting-started-with-nano-server>

### Steps:

Create and deploy the Nano Server image in a bare-metal deployment environment

Be sure to include the OEM drivers package as well as drivers from the HPE Windows Server 2016 driver pack as described in this paper

Be sure the Deployment Type is 'Host'

Deploy the image to the bare-metal server



Before we get started, let's make sure we're running Nano Server using the following commands, **Get-Item 'HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Server\ServerLevels'** and **Get-ItemProperty 'HKLM:\Software\Microsoft\Windows NT\CurrentVersion' |select EditionID**

```
[10.177.250.210]: PS C:\> Get-Item 'HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Server\ServerLevels'

Hive: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Server

Name                Property
----                -
ServerLevels        NanoServer : 1

[10.177.250.210]: PS C:\> Get-ItemProperty 'HKLM:\Software\Microsoft\Windows NT\CurrentVersion' |select EditionID
EditionID
-----
ServerDatacenterNano
```

---

#### Note:

We can see from the output of the second command that we are running the Datacenter version of Nano Server

---

Establish a Remote PowerShell session to the Nano Server target. The remaining steps assume the reader has established a remote PowerShell session to the target.

Verify the SCM drivers are loaded and the NVDIMMs are recognized by the OS by issuing the command, **Get-CimInstance -ClassName Win32\_PNPEntity |? Service -like 'SCMBus'**. Output should be similar to the following:

```
[10.177.250.210]: PS C:\> Get-CimInstance -ClassName Win32_PNPEntity |? Service -like 'SCMBus'

Caption                : Storage Class Memory Bus
Description            : Storage Class Memory Bus
InstallDate           :
Name                  : Storage Class Memory Bus
Status                : OK
Availability           :
ConfigManagerErrorCode : 0
ConfigManagerUserConfig : False
CreationClassName      : Win32_PnpEntity
DeviceID              : ACPI\ACPI0012\2&DABA3FF&0
ErrorCleared          :
ErrorDescription       :
LastErrorCode         :
PNPDeviceID           : ACPI\ACPI0012\2&DABA3FF&0
PowerManagementCapabilities :
PowerManagementSupported :
StatusInfo            :
SystemCreationClassName : Win32_ComputerSystem
SystemName            : 360g90hpec0235
ClassGuid              : {4d36e97d-e325-11ce-bfc1-08002be10318}
CompatibleID          :
HardwareID             : {ACPI\VEN_ACPI&DEV_0012, ACPI\ACPI0012, *ACPI0012}
Manufacturer          : Microsoft
PNPClass               : System
Present               : True
Service               : scmbus
PSComputerName         :
```

To obtain a list of the NVDIMMs installed in the server (similar to what you might see in Device Manager), you can use the command, **Get-CimInstance -ClassName Win32\_PnPEntity |? PNPClass -like '\*scm\*'**

```
[10.177.250.210]: PS C:\> Get-CimInstance -ClassName Win32_PnPEntity |? pnpclass -like '*scm*'

Caption                : NVDIMM-N disk (vendor: 2c80 PID: 4e31)
Description            : NVDIMM-N disk
InstallDate           :
Name                  : NVDIMM-N disk (vendor: 2c80 PID: 4e31)
Status                : OK
Availability           :
ConfigManagerErrorCode : 0
ConfigManagerUserConfig : False
CreationClassName     : Win32_PnPEntity
DeviceID              : SCMLD\VEN_3480&DEV_4131&SUBSYS_2C804E31&REV_0001\3&1B1819F6&0&0101802C011528460F0AA2
ErrorCleared          :
ErrorDescription       :
LastErrorCode          :
PNPDeviceID           : SCMLD\VEN_3480&DEV_4131&SUBSYS_2C804E31&REV_0001\3&1B1819F6&0&0101802C011528460F0AA2
PowerManagementCapabilities :
PowerManagementSupported :
StatusInfo            :
SystemCreationClassName : Win32_ComputerSystem
SystemName             : 360g90hpec0235
ClassGuid              : {53966cb1-4d46-4166-bf23-c522403cd495}
CompatibleID           : {SCMLD\VEN_3480&DEV_4131&REV_0001, SCMLD\VEN_3480&DEV_4131, SCMLD\VEN_3480&CC_0101,
SCMLD\VEN_3480&CC_01...}
HardwareID             : {SCMLD\VEN_3480&DEV_4131&SUBSYS_2c804e31&REV_0001,
SCMLD\VEN_3480&DEV_4131&SUBSYS_2c804e31, SCMLD\VEN_3480&DEV_4131&CC_0101,
SCMLD\VEN_3480&DEV_4131&CC_01}
Manufacturer           : Microsoft
PNPClass               : ScmDisk
Present                : True
Service                : scmdisk0101
PSComputerName         :
```

Since NVDIMMs can be used like other disk media such as HDD, SSD, and NVMe, we use disk related commands to configure them. To obtain a list of the NVDIMM devices and properties, issue the following command, **Get-PhysicalDisk |? BusType -eq 'scm'**

You can see from the previous screenshot that there are four 8GB NVDIMMs installed in this system. Also note the size variation. We'll get to that next.

Now that we've verified that Nano Server properly sees all 4 NVDIMM devices as physical disks, it's time to configure one of them for use. First of all, let's get the disk properties of each NVDIMM using the following command, **Get-Disk |? BusType -eq 'scm'**. The command returns the following output:

```
[10.177.250.210]: PS C:\> get-physicaldisk |? BusType -eq 'scm'
```

FriendlyName	SerialNumber	CanPool	OperationalStatus	HealthStatus	Usage	Size
Vendor 2c80 PID 4e31 802c-01-1513-b30091a9	802c-01-1513-b30091a9	True	OK	Healthy	Auto-Select	7.99 GB
Vendor 2c80 PID 4e31 802c-01-1528-460f0aa2	802c-01-1528-460f0aa2	True	OK	Healthy	Auto-Select	8 GB
Vendor 2c80 PID 4e31 802c-01-1513-b3009195	802c-01-1513-b3009195	True	OK	Healthy	Auto-Select	7.99 GB
Vendor 2c80 PID 4e31 802c-01-1513-b300923d	802c-01-1513-b300923d	True	OK	Healthy	Auto-Select	8 GB

As shown in the two screenshots above, the NVDIMMs are in various states. The first one, number 5, has been initialized as a GPT disk and formatted. The second one, number 3, has been initialized but not as any particular partition style, and the two remaining NVDIMMs have not yet been initialized. We'll focus on configuring the third NVDIMM.

The third NVDIMM in the list above is disk number 4. We can see that the partition style of disk 3 is still raw. Let's go ahead and convert it to an GPT disk using the following command, **Get-Disk |? Number -eq 4 |Initialize-Disk -PartitionStyle GPT**. You should see the following output:

```
[10.177.250.210]: PS C:\> Get-Disk |? BusType -eq 'scm'
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
5	Vendor 2c80 PID 4e31	802c-01-1513-b3009195	Healthy	Online	7.99 GB	GPT
3	Vendor 2c80 PID 4e31	802c-01-1513-b30091a9	Healthy	Online	7.99 GB	RAW
4	Vendor 2c80 PID 4e31	802c-01-1513-b300923d	Healthy	Online	8 GB	RAW
2	Vendor 2c80 PID 4e31	802c-01-1528-460f0aa2	Healthy	Online	8 GB	RAW

Now let's create a partition and assign it the next available drive letter using the following command, **\$drvLetter = New-Partition -DiskNumber 4 -UseMaximumSize -AssignDriveLetter**

```
[10.177.250.210]: PS C:\> $drvLetter = New-Partition -DiskNumber 4 -UseMaximumSize -AssignDriveLetter -Verbose
[10.177.250.210]: PS C:\> $drvLetter
```

DiskPath: \\?\scmld#ven\_3480&dev\_4131&subsys\_2c804e31&rev\_0001#3&1b1819f6&0&0101802c011513b300923d#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

PartitionNumber	DriveLetter	Offset	Size	Type
2	E	33579008	7.96 GB	Basic

Finally we'll format the new partition as NTFS using the following command, **Format-Volume -DriveLetter \$drvLetter.DriveLetter**

```
[10.177.250.210]: PS C:\> Format-Volume -DriveLetter $drvLetter.DriveLetter -Verbose
```

DriveLetter	FileSystemLabel	FileSystem	DriveType	HealthStatus	OperationalStatus	SizeRemaining	Size
E		NTFS	Fixed	Healthy	OK	7.93 GB	7.96 GB

Our new disk listing is as follows:

```
[10.177.250.210]: PS C:\> Get-Disk |? BusType -eq 'scm'
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
5	Vendor 2c80 PID 4e31	802c-01-1513-b3009195	Healthy	Online	7.99 GB	GPT
3	Vendor 2c80 PID 4e31	802c-01-1513-b30091a9	Healthy	Online	7.99 GB	RAW
4	Vendor 2c80 PID 4e31	802c-01-1513-b300923d	Healthy	Online	7.99 GB	GPT
2	Vendor 2c80 PID 4e31	802c-01-1528-460f0aa2	Healthy	Online	8 GB	RAW

We now have two NVDIMMs configured and ready for use, one that is partially configured, and one that has yet to be configured

## **HPE Persistent Memory Resources, contacts, or additional links**

### **HPE Persistent Memory**

[hpe.com/servers/persistentmemory](http://hpe.com/servers/persistentmemory)

[persistentmemory.hpe.com](http://persistentmemory.hpe.com)

HPE 8GB NVDIMM Single Rank x4 DDR4-2133 Module (782692-B21)

[hpe.com/us/en/product-catalog/servers/server-memory.html](http://hpe.com/us/en/product-catalog/servers/server-memory.html)

[youtube.com/watch?v=BKA\\_SOPqHfg](https://youtube.com/watch?v=BKA_SOPqHfg)

[twitter.com/hpe/status/717103146204397568](https://twitter.com/hpe/status/717103146204397568)

HPE Persistent Memory HPE Discover London, December 2015

[youtube.com/watch?v=vMrzXOBSeqA](https://youtube.com/watch?v=vMrzXOBSeqA)

HPE Improving Microsoft SQL Server Database Performance with HPE NVDIMM

[h20195.www2.hpe.com/V2/GetDocument.aspx?docname=4AA6-7123ENW](http://h20195.www2.hpe.com/V2/GetDocument.aspx?docname=4AA6-7123ENW)

HPE Persistent Memory HPE Discover Las Vegas, June 2016

SanDisk Blog “Doing More With Less Gets Faster” with HPE NVDIMM

[itblog.sandisk.com/new-sql-server-database-cache-acceleration-solution-preview](http://itblog.sandisk.com/new-sql-server-database-cache-acceleration-solution-preview)

### **Microsoft Windows Server 2016 support of HPE NVDIMM**

“MS Build” conference, San Francisco, March 2016

Microsoft Storage Class Memory (NVDIMM-N) Health Management in Windows

[technet.microsoft.com/en-us/windows-server-docs/storage/storage-spaces/storage-class-memory-health](http://technet.microsoft.com/en-us/windows-server-docs/storage/storage-spaces/storage-class-memory-health)

Microsoft presentation of Windows SCM at SNIA

[snia.org/sites/default/files/SDC15\\_presentations/file\\_sys/NealChristiansen\\_SCM\\_on\\_Windows.pdf](http://snia.org/sites/default/files/SDC15_presentations/file_sys/NealChristiansen_SCM_on_Windows.pdf)

Microsoft Configuring Storage Spaces with a NVDIMM-N write cache

[msdn.microsoft.com/library/mt650885.aspx](http://msdn.microsoft.com/library/mt650885.aspx)

Using Non-volatile Memory as Block Storage in Windows Server 2016

[channel9.msdn.com/events/Build/2016/P466](http://channel9.msdn.com/events/Build/2016/P466)

Using Non-volatile Memory as Byte-Addressable Storage in Windows Server 2016

[channel9.msdn.com/events/Build/2016/P470](http://channel9.msdn.com/events/Build/2016/P470)

Microsoft SQL Server 2016 and Windows Server 2016 SCM – FAST

[channel9.msdn.com/Shows/Data-Exposed/SQL-Server-2016-and-Windows-Server-2016-SCM--FAST](http://channel9.msdn.com/Shows/Data-Exposed/SQL-Server-2016-and-Windows-Server-2016-SCM--FAST)

### **Microsoft DiskSpd Utility**

[gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223](http://gallery.technet.microsoft.com/DiskSpd-a-robust-storage-6cd2f223)

## Security and Assurance

### Introduction

#### What's new

Windows server 2016 designed to protect against known and emerging security threats across the spectrum of attack vectors. The following categories of security work went into Windows server 2016:

**Identity and access control** features have been greatly expanded to both simplify and enhance the security of user authentication. New feature is Credential Guard, access the links provided in the “Credential Guard” section of this paper.

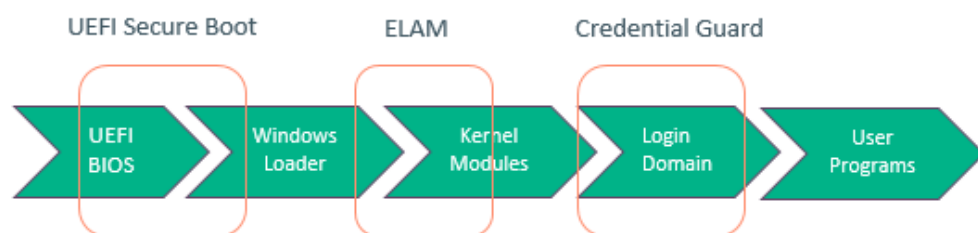
**Malware resistance** includes architectural changes that can isolate critical system and security components from threats. Traditional feature is ELAM, new feature is Device Guard. For ProLiant Gen9 Server, please wait future BIOS and document release for enablement of Device Guard.

**Guarded Fabric** to provide a more secure environment for virtual machines. New feature is Host Guardian Service (HGS), access the links provided in the “Host Guardian Service and Shielded VMs” section of this paper.

Some other system security features utilized by Windows Server 2016 are “Secure Boot”, “TPM”. See related sections in this paper for detail information.

### Protecting UEFI and Windows

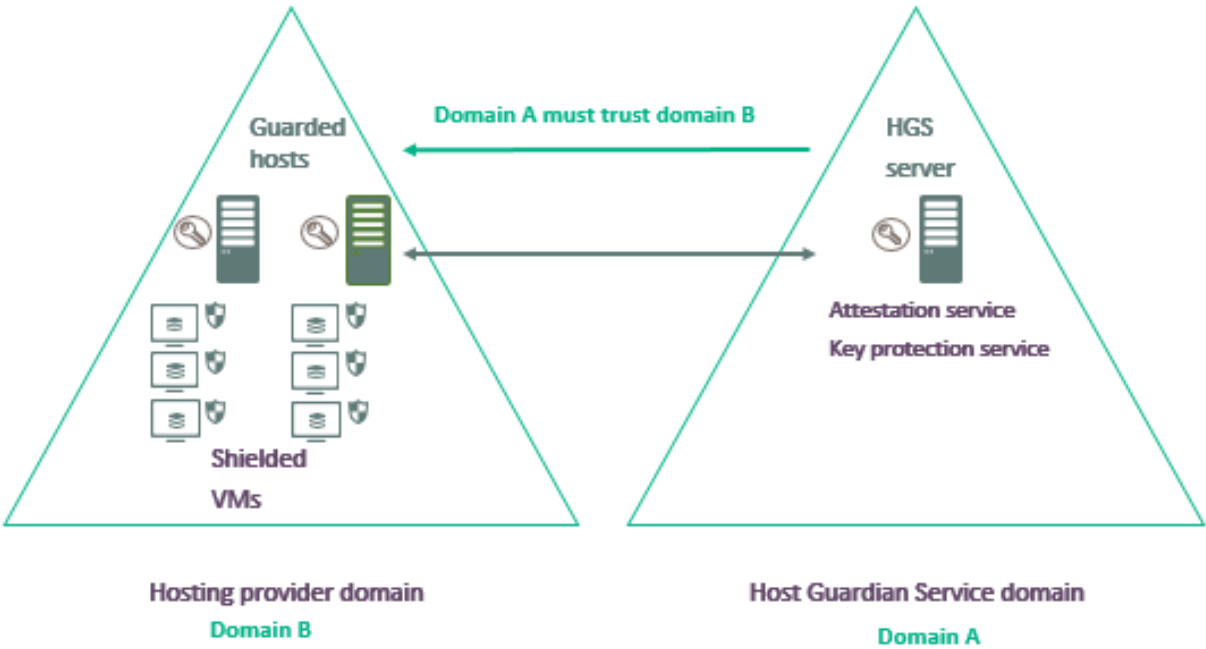
This diagram show HPE ProLiant Server and Windows Server 2016 startup process with all security features.



(Orange color items supported by TPM)

Protecting Cloud and Virtual Environment

Previous section describe how to secure one machine. Today, virtual environment becomes popular, there are many types of administrators who could access VMs. Enterprise need a way to secure VMs. This diagram show how HGS can protect the entire environment.



**TPM**

A TPM is a microchip designed to provide security-related functions, including encryption and decryption keys. The TPM is usually installed on board of a computer. With a TPM, private portions of key pairs are kept separate from the memory controlled by the operating system. Keys can be sealed to the TPM, and certain assurances about the state of a system.

The following existing features that rely on TPM module:

Feature	Version	Support
Secure Boot	1.2 / 2.0	Required
Credential Guard	2.0	Required
Device Guard	2.0	Optional
Host Guardian Service	2.0	Optional

To verify the list of HPE ProLiant Gen9 servers which have earned the Hardware Assurance Additional Qualifier you can check the Microsoft Hardware Catalog:  
[windowsservercatalog.com/results.aspx?&text=HPE+ProLiant+Gen9&bCatID=1282&cpID=0&avc=10&ava=0&avt=0&avq=89&OR=5&PGS=25](https://windowsservercatalog.com/results.aspx?&text=HPE+ProLiant+Gen9&bCatID=1282&cpID=0&avc=10&ava=0&avt=0&avq=89&OR=5&PGS=25)

HPE ProLiant Gen9 Servers with Hardware Assurance Additional Qualifier (AQ)

**Note:** HPE ProLiant Gen8 servers do not support Hardware Assurance

## Configuring HPE ProLiant Gen9 Servers with UEFI Secure Boot

### What is Secure Boot?

UEFI secure Boot is a method used to restrict binaries execution for booting the system. With this option enabled, system BIOS will only allow boot loaders with trusted cryptographic signatures to be executed, thus enable preventing malware from hiding embedded code in the boot chain.

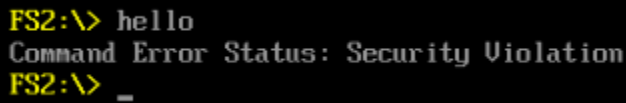
### Configure Secure Boot in System Utilities

Below are the steps for enabling secure boot:

1. Power on the server and press F9 during POST to enter the “System Utilities”
2. Select “System Configuration” and “BIOS/Platform Configuration (RBSU)”
3. From RBSU, select “Server Security” → “Secure Boot Settings”
4. Set the “Secure Boot Enforcement” to [Enabled]
5. Press ‘Y’ to save and exit, and reboot to make the new setting take effect

A simple way for ensuring the effect, below are the steps:

1. Press F9 during POST to enter the “System Utilities”
2. Select “Embedded application” and “Embedded UEFI Shell”
3. The Embedded UEFI Shell screen appears. Run any unsigned UEFI application in command line
4. Application will be blocked and prompt error message. See below example: a unsigned UEFI program “hello.efi”



```
FS2:\> hello
Command Error Status: Security Violation
FS2:\> _
```

---

### Important

Operating systems must support Secure Boot and have a UEFI boot loader signed with one of the authorized keys to boot. Windows Server 2016 is supported. For more information about supported operating systems, see the “UEFI System Utilities and Shell Release Notes for ProLiant Gen9 Servers” on the HPE website ([hpe.com/info/ProLiantUEFI/docs](http://hpe.com/info/ProLiantUEFI/docs)).

---

For additional information about secure boot configuration, see the “HPE UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers” on the HPE website ([hpe.com/info/ProLiantUEFI/docs](http://hpe.com/info/ProLiantUEFI/docs)).

## Configuring HPE Trusted Platform Module 2.0 for HPE Gen9 servers

### Abstract

The HPE TPM 2.0 (Trusted Platform Module 2.0) is a HPE designed security solution, which has a TPM 2.0 chip within it. HPE ProLiant Gen9 servers do not have a TPM module by default. The HPE TPM 2.0 module is an optional kit that can be purchased along with HPE ProLiant Gen9 servers or can be purchased separately. For details on how to get the HPE TPM 2.0, please refer to the following document [hpe.com/h20195/v2/GetPDF.aspx/c04939549.pdf](http://hpe.com/h20195/v2/GetPDF.aspx/c04939549.pdf). Reference HPE Trusted Platform Module 2.0 Kit 745823-B21

### Install TPM board

Please open the user guide of HPE ProLiant Gen9 server. “Installing the Trusted Platform Module board 2.0” section shows user how to install TPM 2.0 module on server.

**Enablement**

Normally, TPM 2.0 will be enabled automatically in the HPE ProLiant Gen9 server after installation. To verify the TPM 2.0 module is functional, follow the steps below:

1. During the server startup sequence, press the F9 key to access System Utilities.
2. From the System Utilities screen, select System Configuration > BIOS/Platform Configuration (RBSU) > Server Security.
3. Select Trusted Platform Module Options and press the Enter key.
4. Select Enabled (if it shows "No Action" or "Disabled") to enable the TPM and BIOS secure startup. The TPM is fully functional in this mode.
5. Press the F10 key to save your selection.
6. When prompted to save the change in System Utilities, press the Y key.
7. Press the ESC key to exit System Utilities. Then, press the Enter key when prompted to reboot the server.

For additional information about TPM 2.0 configuration, see the "HPE UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers" on the HPE website ([hpe.com/info/ProLiantUEFI/docs](http://hpe.com/info/ProLiantUEFI/docs)).

To verify the TPM 2.0 module is functional within the Windows Server 2016 operating system, run the TPM Management Console: Open a Command Prompt console as Administrator and type "tpm.msc"

The status of the TPM should be "The TPM is ready for use", otherwise select the "Clear TPM..." option from the Actions Pane on the right and reboot the OS.

**Early Loading Anti-Malware (ELAM)**

Since the release of Windows Server 2012, Microsoft has provided ELAM support for drivers. It provides a standard interface for anti-malware software to be initialized before other boot drivers are loaded. This ensures all subsequent drivers do not contain malware.

For Windows Server 2016 this interface is still supported. For detail please see below MSDN link:  
[msdn.microsoft.com/en-us/library/windows/desktop/hh848061\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/hh848061(v=vs.85).aspx)

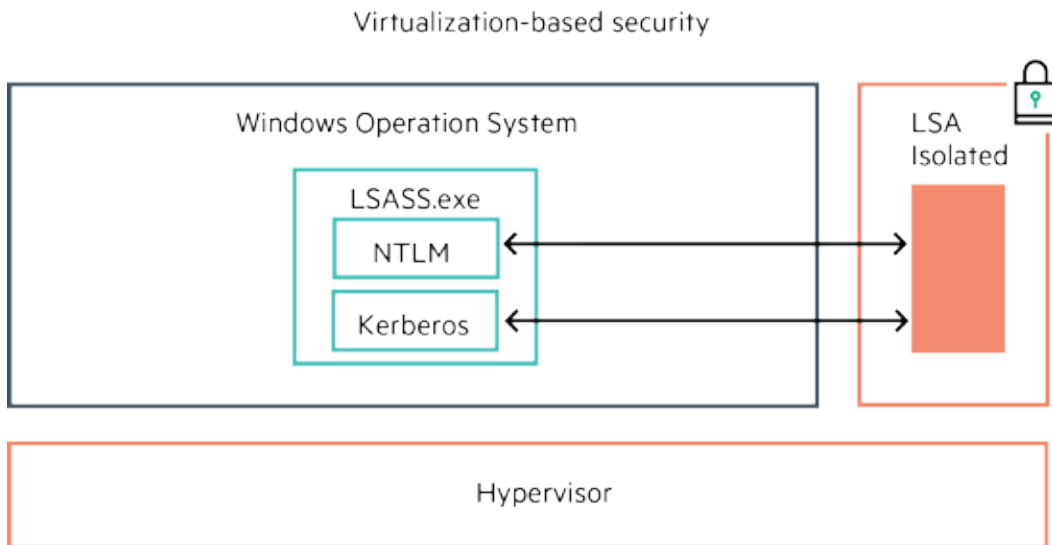
**Credential Guard and Remote Credential Guard****What is Credential Guard?**

Credential Guard is a new feature introduced in Windows Server 2016. It uses hardware security features to protect domain credentials. It is built using virtualization-based security that isolates the domain credential and other secrets in a protected environment which is managed by the Hypervisor.

**Basic Concepts**

Windows Server 2016 uses the Local Security Authority Subsystem Service (LSASS.exe) to enforce the security policy for system. (See the MSDN link for [LSA Authentication](#)). The LSASS.exe is a crucial system service and is often the target of malware or theft attacks to the service in order to access the credential and/or secrets. Credential Guard separates the LSASS.exe process into two parts and isolates the password and secrets in a virtualization-based security environment which is not accessible to the rest of the operating system. This technique can reduce the exposure area from malware attack.





#### Manage Credential Guard Locally or with Group Policy

Credential Guard can be enabled locally by setting local policy (gpedit.msc) and registry key. It can also be managed with Group Policy on a domain controller. Use the system tool **msinfo32.exe** to check if the Credential Guard is running. If you want to review of system Credential Guard status, use the system **Event Viewer** to check Event ID (13~17) which are related to Credential Guard status. Please see below Microsoft link for detail of how to enable them locally or for the entire domain:  
[technet.microsoft.com/en-us/itpro/windows/keep-secure/credential-guard](https://technet.microsoft.com/en-us/itpro/windows/keep-secure/credential-guard)

#### Note

It is recommended that Credential Guard be enabled before the system is utilized and joined to the domain. This can protect against potential risks to users and devices.

### HPE ProLiant Server Feature for Credential Guard

#### HPE ProLiant Servers support below special security features that are required by Credential Guard

- Hardware: 64-bit CPU
- Hardware: Virtualization extensions (Intel VT-x) and extended page tables for virtualization-based security (VBS)
- Intel VT-d with IOMMU (Input/output memory management) for DMA protection
- UEFI Secure Boot. See section "[Configure Secure Boot in System Utilities](#)" section of this paper
- Firmware: Securing boot configuration and management
- Trusted Platform Module (TPM) version 2.0 (Option: 745823-B21)

Some of the security features need to be enabled by system BIOS RBSU menu (for example, the Security Boot). Please see [PC OEM requirements](#) for detail system security features required by Credential Guard.

#### Note

Some of the security features are absent from the above list on ProLiant Gen 9 Servers. This mean the Credential Guard cannot be protected by the specific security feature but it doesn't affect to enable the Credential Guard and have its protection. Please wait future BIOS and document release for enablement of these special security features.

## Host Guardian Service and Shielded VMs

### What is Host Guardian Service?

The Host Guardian Service (HGS) is a new feature in Windows Server 2016 that enables Hyper-V hosts to run Shielded VMs. It acts as an administrative portal that provides attestation services to a group of trusted hosts. There are two different modes of attestation: Admin-trusted and TPM-trusted. Shielded VMs encrypt the information in the virtual machine, thereby protecting the data against threats from unauthorized Hyper-V administrators.

### Configuration

For TPM-trusted attestation mode implementation, each Hyper-V host node must be running UEFI and have secure boot enabled. By default, all HPE Gen9 servers run UEFI, access the links provided in the “Configuring HPE ProLiant Gen9 Servers with UEFI Secure Boot” section of this paper. A TPM 2.0 module is also required for TPM-trusted mode. For additional information about TPM, access the links provided in the “Configuring HPE Trusted Platform Module 2.0 for HPE Gen9 servers” section. Install a TPM 2.0 module on each Hyper-V host node.

### Enhanced Security with HPE TPM 2.0

It is suggested to install a TPM 2.0 module on both the HGS server and the Hyper-V host(s). The following diagram shows the basic environment for TPM-trusted attestation:



Even though you can deploy your guarded fabrics without TPM, your host(s) can be easily hacked if you implement Admin-trusted attestation mode. In addition, the HGS server will boot securely every time. TPM has the following advantages:

1. System disk and Virtual hard disk (.vhdx) BitLocker encryption
2. UEFI secure boot technology support
3. Install HGS using TPM-trusted attestation mode

### HGS and Shielded VM deployment

Install Windows Server 2016 TP5 Datacenter Edition on each of the servers and apply the following drivers from the “Service Pack for ProLiant (SPP) 2016.10.0”.

HPE ProLiant Gen9 Chipset Identifier for Windows	cp028130.exe
HPE ProLiant iLO 3/4 Channel Interface Driver for Windows x64	cp028339.exe
HPE ProLiant iLO 3/4 Management Controller Driver package	cp028266.exe
Matrox G200eH Video Controller Driver for Windows Server 2012 and Server 2012 R2	cp025629.exe

Reboot after the drivers are installed and refer to the following document from Microsoft titled “Shielded VMs and Guarded Fabric Deployment Guide for Windows Server 2016 TP\_9\_30.docx” found in this link: [gallery.technet.microsoft.com/shielded-vm-and-guarded-fabric-deployment-guide-for-windows-server-2016-tp-9-30/98d2b045](https://gallery.technet.microsoft.com/shielded-vm-and-guarded-fabric-deployment-guide-for-windows-server-2016-tp-9-30/98d2b045)

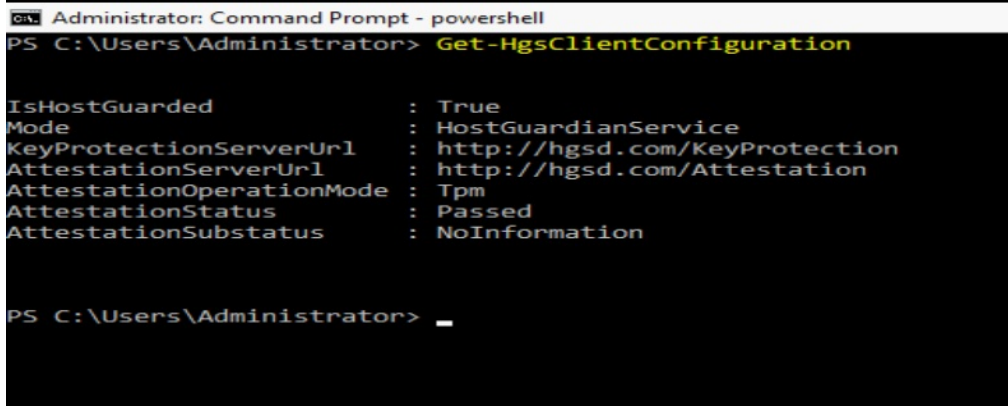
### A Basic User Case: Standalone Host Scenario

Unlike the VMM scenario or the Azure Pack scenario described in Microsoft’s document listed above, setting up a simple standalone Hyper-V host forest domain to communicate with the HGS server is not difficult. This provides a quick and easy ramp up for customers without System Center or Azure Pack. It is helpful for organizations with a smaller scale of server nodes or test environments and still want the protection that Shielded VMs offer.

In this example we use one HGS server and one or more Hyper-V host to run TPM-trusted mode Shielded VMs.

### Start First Shielded VM

1. Configure the HGS server and Hyper-V host environment by following section 8 of “Shielded VMs and Guarded Fabric Deployment Guide for Windows Server 2016 TP\_9\_30.docx” (from the link above)
  - a. Verify that your host is guarded by using the PowerShell cmdlet `Get-HgsClientConfiguration` as shown below:



```

Administrator: Command Prompt - powershell
PS C:\Users\Administrator> Get-HgsClientConfiguration

IsHostGuarded           : True
Mode                    : HostGuardianService
KeyProtectionServerUrl   : http://hgsc.com/KeyProtection
AttestationServerUrl    : http://hgsc.com/Attestation
AttestationOperationMode : Tpm
AttestationStatus       : Passed
AttestationSubstatus    : NoInformation

PS C:\Users\Administrator>
  
```

- b. Assuming the HGS server's AD domain is hgsc.com, get the guardian key using a web browser: [hgsc.com/KeyProtection/service/metadata/2014-07/metadata.xml](http://hgsc.com/KeyProtection/service/metadata/2014-07/metadata.xml)

---

#### Note

replace the “hgsc.com” domain with the domain name of your HGS server.

---

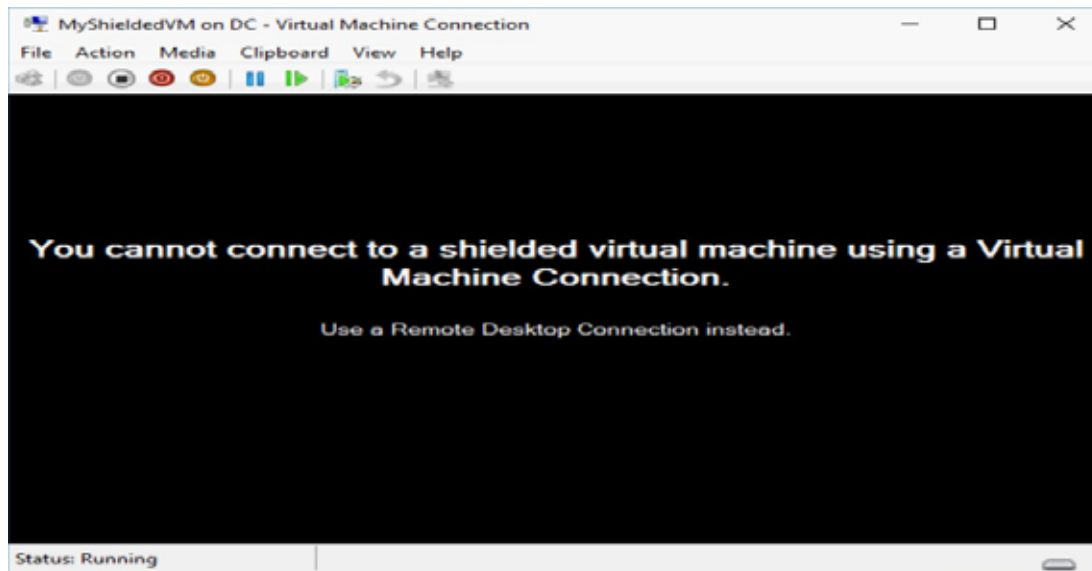
Save this file locally as GuardianKey.xml and import it to the host using PowerShell:

```

PS C:\> Import-HgsGuardian -Path 'C:\GuardianKey.xml' -Name 'Guardian'
-AllowUntrustedRoot
  
```

2. Create a new shielded VM with following cmdlets:
  - a. `New-VM -Generation 2 -Name "MyShieldedVM" -Path C:\VM -NewVHDPPath C:\VM\MyShieldedVM\MyShieldedVM.vhdx -NewVHDSizesBytes 64GB`
  - b. `$Guardian = Get-HgsGuardian -Name 'Guardian'`
  - c. `$Owner = New-HgsGuardian -Name 'Owner' -GenerateCertificates`
  - d. `$KP = New-HgsKeyProtector -Owner $Owner -Guardian $Guardian -AllowUntrustedRoot`
  - e. `Set-VMKeyProtector -VMName "MyShieldedVM" -KeyProtector $KP.RawData`
  - f. `Set-VMSecurityPolicy -VMName "MyShieldedVM" -Shielded $true`
  - g. `Enable-VMTPM -VMName "MyShieldedVM"`

3. MyShieldedVM is now shielded and the Virtual Machine will start on this host only if it is in Guarded state. Any attempt to connect to the running VM from Hyper-V Manager will result in the following screen:



## Additional Resources

### Overview

Security technologies and documentation for Windows Server 2016  
[technet.microsoft.com/windows-server-docs/security/security-and-assurance](https://technet.microsoft.com/windows-server-docs/security/security-and-assurance)

### Credential Guard

[technet.microsoft.com/en-us/itpro/windows/keep-secure/credential-guard](https://technet.microsoft.com/en-us/itpro/windows/keep-secure/credential-guard)

### HGS and Shielded VMs

1. [channel9.msdn.com/events/Ignite/2015/BRK3457](https://channel9.msdn.com/events/Ignite/2015/BRK3457)
2. [technet.microsoft.com/en-us/library/mt599611.aspx](https://technet.microsoft.com/en-us/library/mt599611.aspx)
3. [technet.microsoft.com/en-US/library/mt130644.aspx](https://technet.microsoft.com/en-US/library/mt130644.aspx)
4. [gallery.technet.microsoft.com/Shielded-VMs-and-Guarded-70c5b471](https://gallery.technet.microsoft.com/Shielded-VMs-and-Guarded-70c5b471)

## Installing Windows Server 2016

Windows Server 2016 must be installed manually from the Windows Server 2016 media.

### Preinstallation tasks

Before installing Windows Server 2016, complete the following tasks:

- Review the [Microsoft Windows Server 2016 release notes](#).
- Make sure that your server, storage controller, and NIC are listed in this paper as a supported server or option.
- Reset the server to default settings and configure the boot controller order if necessary:
  - For legacy BIOS servers (ProLiant Gen8 servers, except for the ProLiant DL580 Gen8 server), reboot the server and press F9 from the main boot screen to start the HPE ROM Based Setup Utility (RBSU). For instructions on using the RBSU, see the [HPE ROM-Based Setup Utility User Guide](#).

- For UEFI-capable servers (ProLiant DL580 Gen8 and ProLiant Gen9 servers), reboot the server and press F9 when prompted during the startup sequence to start the System Utilities. Select System Configuration. For instructions on using System Configuration, see the HPE UEFI System Utilities User Guide for HPE ProLiant Gen9 Servers, which is available in the HPE Information Library at [h17007.www1.hpe.com/us/en/enterprise/servers/solutions/infolibrary/index.aspx?cat=hp\\_uefi\\_system\\_utilities](http://h17007.www1.hpe.com/us/en/enterprise/servers/solutions/infolibrary/index.aspx?cat=hp_uefi_system_utilities).
- Make sure that the server firmware is up-to-date. If necessary, complete the following:
  - Update the System ROM to the latest version. You can download the latest BIOS for the ROM from the HPE Support Center at [h20565.www2.hpe.com/portal/site/hpsc](http://h20565.www2.hpe.com/portal/site/hpsc).
  - Update the iLO firmware to the latest version. You can download the latest iLO firmware from the HPE Support Center at [h20565.www2.hpe.com/portal/site/hpsc](http://h20565.www2.hpe.com/portal/site/hpsc).
- Depending on your installation method, make sure that you have either of the following:
  - If installing from the Windows Server 2016 media, the server must have a DVD drive either installed or attached to the server receiving the installation.
  - If using iLO virtual media to install Windows Server 2016 to a ProLiant ML or DL server, you need an Advanced iLO key if not already present.

### Installing Windows Server 2016 from the OS media

To install Windows Server 2016 on a server using the Windows Server 2016 media, complete the following steps:

1. Insert the Windows Server 2016 media into the DVD drive and boot the server to the DVD.
2. Follow the steps on the installation screens to complete the OS installation.

### Installing Windows Server 2016 using iLO

1. Power on the server in which Windows® will be installed. This server is referred to as the Installation Server (IS).
2. From an alternate Windows server or PC, referred to as the Console System (CS), verify that the CS has access to the network of the IS.
3. Open a supported Web browser on the CS and browse to the iLO IP of the IS. Log in to iLO.
4. Expand the Remote Console drop down, select the Remote Console option, and click the Launch button under the Integrated Remote Console section of the Remote Console page.
5. After the iLO Integrated Remote Console window opens, open the Virtual Drives drop down menu and select Image File CD-ROM/DVD.
6. In the Mount Image File pop up window, browse to the ISO location on the CS and Open the ISO. The ISO is virtually mounted as CD-ROM/DVD.
7. Follow the steps on the installation screens to complete the OS installation.

### Installing components from the HPE Service Pack for ProLiant (SPP) version 2016.10.0

Download the HPE Service Pack for ProLiant (SPP) version 2016.10.0 from [hpe.com/servers/spp/download](http://hpe.com/servers/spp/download)

We recommend that you use HPSUM 7.6.0 to install the components that are included in the **HPE Service Pack for ProLiant (SPP) version 2016.10.0**. You can obtain HPSUM 7.6.0 by downloading it from [here](#):

---

#### Note

The download of the SPP requires an active warranty, HPE Care Pack, or support agreement that is linked to your HPE Support Center profile. To determine eligibility, you must sign in to access the download. For information on how warranties, HPE Care Packs, and support agreements enable access to select downloads or site functions, click Help on the HPE Support Materials website.

---

To install components from the **HPE Service Pack for ProLiant (SPP) version 2016.10.0** to Standard version of the Operating System locally, complete the following steps:

1. Mount SPP 2016.10.0 through iLO.
2. Click on "launch\_hpsum.bat" to launch HPSUM.
3. Click on Local-host Guided updates

4. Choose Interactive mode of flashing and Click OK
5. Once inventory of Baseline and node is complete, click Next.
6. On Review page, select the required components to be installed and proceed with the deployment by clicking on "Deploy" button.
7. After Deployment, HPSUM will list all the component that we selected for installation with the installation result.
8. Reboot the server as per requirement and re-launch following steps 1-8, to install the firmware that was not installed on the first launch.

For automatic mode of flashing, follow steps 1-3 as stated above and proceed with below steps:

4. Choose automatic mode of flashing.
5. Baseline inventory, node inventory and deployment completes without any human intervention.
6. Once deployment is complete, HPSUM will list all the component that was installed with the installation result.
7. Reboot the server and re-launch following steps 1-6, to install any firmware components that was not installed on the first launch.

As a security feature, Microsoft disabled access to admin\$ share by default for all admin group users other than default "Administrator". But admin\$ share is accessible for domain admin users.

Adding the below key in the registry will solve accessing the admin\$ share problem. Link for reference [support.microsoft.com/en-us/kb/951016](https://support.microsoft.com/en-us/kb/951016)

Key: HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System

Value: LocalAccountTokenFilterPolicy

Data: 1 (to disable, 0 enables filtering)

Type: REG\_DWORD (32-bit)

To install components from the Service Pack for ProLiant (SPP) version 2016.10.0 for Hyper-V or Essentials version of the Operating System remotely, complete the following steps:

8. Make a folder on the server Desktop and copy the extracted contents of the Service Pack for ProLiant (SPP) version 2016.10.0 and HPSUM 7.6.0 to the folder.
9. Launch HPSUM.
10. Click on baseline library and complete the baseline inventory
11. Click Add node and enter the details (IP, target type, baseline and credentials) of the target machine and click on "**ADD**" to add the target.
12. Click on "**inventory**" to complete the node inventory.
13. After the inventory completes, click "**Review and Deploy updates**" link, click on "**OFF**" button on the screen and choose proceed anyway to view the list of component to be installed and follow the on-screen instructions.

## Questions, issues, and workarounds

We want you to understand our level of support and be aware of potential issues that you may encounter with ProLiant servers and Windows Server 2016. We are working to resolve all issues. Future editions of this paper will include updates to supported servers and options, along with information about any new issues that we find and are working to resolve. Table 5 lists questions, and known issues, and workaround details that we have at this time.

**Table 5.** Known issues and workarounds.

Description of issue	Workaround/Solution
Yellow bang seen for PS/2 compatible mouse and keyboard used with HP ProLiant Gen8 servers	Take Headless server registry component from SPP build and install it outside of the HPSUM environment
Nano Server is not supported on the HPE BL920s Superdome X Gen8/Gen9	Use Windows Server 2016 Server Core or Server Core with full desktop experience
ML310e Gen8 v2 will not boot after enabling the Hyper-V role.	Ensure ROM settings are configured with VT enabled and VT-D disabled before enabling the Hyper-V role.
If you are performing a manual install of Windows Server 2016 using Intelligent Provisioning and using an HPE Smart Storage Adapter set to HBA mode and connected to a single HDD, you may notice that your drive may be set offline during the install process.	If your server is configured to use UEFI, then you can simply click on the drive icon and set it online status. If your server is configured to use legacy BIOS you will need to connect an additional drive to the controller and repeat the installation process.

## Nano Server

### Introduction

Nano Server is a new offering of Windows Server included with Windows Server 2016. It is a re-architected operating system focused on the cloud and born-in-the-cloud applications and follows a zero-footprint model meaning you install only the features and applications you need for a targeted workload. The first notable change with Nano Server is that it is headless, meaning there is little support for local keyboard, video or mouse once the operating system has booted. With the headless nature of Nano Server, the primary method of server management is through PowerShell remoting. Another method of server management is with the HPE ProLiant Integrated Lights-Out infrastructure, namely the iLO Virtual Serial Port and EMS console, which provides many of the basic administrative tasks through a text-based Secure Shell (SSH) environment. Here is a brief listing of the many benefits of using the VSP and EMS console:

- Display the OS IP address
- Obtain basic OS information
- Obtain a list of active Windows processes
- Shutdown or reboot the server
- Ability to log on to the OS to perform basic administrative tasks such as interactively make changes to the Windows firewall, add local user accounts, etc. This is very beneficial if you are unable to establish a remote PowerShell session to the server.

There are several ways to deploy Nano Server. This document covers two primary deployment methods. The first is deploying Nano Server from a WinPE environment. The second is deploying Nano Server using Windows Deployment Services.

This document assumes the reader is familiar with the technologies discussed in this Technical White Paper.

- Microsoft “Getting Started with Nano Server” guide
  - [technet.microsoft.com/en-us/library/mt126167.aspx](https://technet.microsoft.com/en-us/library/mt126167.aspx)
- Windows Deployment Services (WDS)
  - [technet.microsoft.com/en-us/library/jj648426.aspx](https://technet.microsoft.com/en-us/library/jj648426.aspx)
- Windows Assessment and Deployment Kit (ADK)
  - [msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx](https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx)
  - Download link, [go.microsoft.com/fwlink/p/?LinkId=526740](https://go.microsoft.com/fwlink/p/?LinkId=526740)
- Windows Preinstallation Environment (WinPE)
  - [msdn.microsoft.com/library/windows/hardware/dn938389.aspx](https://msdn.microsoft.com/library/windows/hardware/dn938389.aspx)
- Deployment and Imaging Service (Dism.exe)
  - [msdn.microsoft.com/en-us/library/windows/hardware/dn898558\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/dn898558(v=vs.85).aspx)

- Windows System Image Manager (WSIM)
  - [technet.microsoft.com/en-us/library/cc722301\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc722301(v=ws.10).aspx)
- Unattend.xml files
  - [technet.microsoft.com/en-us/library/cc771830\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc771830(v=ws.10).aspx)
- PowerShell
  - [msdn.microsoft.com/en-us/mt173057.aspx](https://msdn.microsoft.com/en-us/mt173057.aspx)
  - [technet.microsoft.com/en-us/library/bb978526.aspx](https://technet.microsoft.com/en-us/library/bb978526.aspx)

## Getting Started

One of the most important decisions the reader will have to make is deciding which method to use to deploy Nano Server. Below is a table that highlights each deployment method along with required components to assist the reader in choosing a method best suited for their environment:

**Table 6.** Deployment scenarios and associated components

Deployment method	Required components	Pros and cons
<b>Windows Deployment Services</b>	<ul style="list-style-type: none"> <li>WDS server and associated storage and network infrastructure</li> <li>Technician computer (WDS server can serve as this role)</li> <li>Windows 10 ADK</li> <li>Windows System Image Manager</li> <li>Windows Server 2016 ISO</li> <li>Requires the creation of a custom Nano Server WIM image</li> <li>iLO IRC</li> </ul>	<p><b>Pros:</b></p> <ul style="list-style-type: none"> <li>Used for bare-metal deployments</li> <li>Fits nicely into an existing WDS environment</li> <li>Scalable, consistent, and automated deployments</li> </ul> <p><b>Cons:</b></p> <ul style="list-style-type: none"> <li>Requires additional storage and network infrastructure than the other methods</li> <li>Requires WDS knowledge</li> <li>Requires considerable unattend.xml knowledge</li> </ul>
<b>WinPE</b>	<ul style="list-style-type: none"> <li>Technician computer (WDS server can serve as this role)</li> <li>Latest Windows 10 ADK</li> <li>Windows System Image Manager</li> <li>Windows Server 2016 ISO</li> <li>iLO IRC w/virtual media support (for ISO boot)</li> <li>USB thumb drive (required for USB boot)</li> <li>WDS Server (required for PXE booting WinPE using WDS)</li> <li>Web Server like IIS (required for utilizing HTTP boot or <b>Boot from URL</b>)</li> <li>Requires the creation of a WinPE boot image</li> <li>Requires the creation of a custom Nano Server image</li> </ul>	<p><b>Pros:</b></p> <ul style="list-style-type: none"> <li>Used for bare-metal deployments</li> <li>Requires less infrastructure than WDS (when not PXE booting WinPE)</li> <li>Relatively easy to setup and configure</li> </ul> <p><b>Cons:</b></p> <ul style="list-style-type: none"> <li>Manual installation (can be automated with additional scripting)</li> <li>Requires some knowledge of WinPE</li> <li>Requires the creation of a WinPE boot image</li> <li>Requires the creation of a WinPE ISO image (for ISO based deployments)</li> </ul>



Not only does Nano Server bring with it special deployment requirements, it also has special operating and manageability requirements that could be viewed as “optional” on the other Windows Server SKU’s. Below is a list of special requirements and considerations for a Nano Server deployment:

**Table 7.** Special considerations for Nano Server

Deployment	Runtime and manageability
<ul style="list-style-type: none"> <li>Ships as a stand-alone image</li> <li>Unable to install from product ISO image</li> <li>Unable to switch between Nano Server and the other SKUs</li> <li>Offline injection required for boot-critical drivers, optional for other drivers</li> <li>Installation can be automated using unattend.xml files</li> <li>Microsoft includes tools to aid in the image creation and customization</li> </ul>	<ul style="list-style-type: none"> <li>KVM functionality is limited during OS runtime</li> <li>Recommend server and OS EMS support and enablement</li> <li>Lack of local online HPE Smart Component support</li> <li>Local, direct access is achievable through SSH, VSP, and EMS</li> <li>Requires PowerShell remoting for manageability</li> <li>Requires additional configuration to manage roles such as Hyper-V</li> <li>Online driver installation and updating is supported using pnputil.exe</li> </ul>

## Preparing the environment

Prior to getting started, the reader should perform the following actions:

- Identify a system to fill the role as the technician computer and install Windows Server 2016 as the operating environment.
- Download and extract the contents of the Windows Server 2016 ISO to a working folder on the technician computer like, c:\ws2016 (This folder will be used throughout this document)
- Download the **Service Pack for ProLiant (SPP) version 2016.10.0** to the technician computer.
  - Mount the SPP ISO image.
  - Open the folder named WIN\_DRV and extract the contents of the win-driverpack-10.60.zip file to a local folder on the technician computer like c:\spp\win-driverpack-10.60.
  - You may now dismount the SPP ISO image.
- Download and install the Windows 10 ADK from the Microsoft website, [msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx](https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx)
  - The minimal set of tools to install are “Deployment Tools” and “Windows Preinstallation Environment (Windows PE)”. The other tools are optional.
  - This document assumes the kit is installed in the default path.
- Open a CMD shell provided by the ADK called, “Windows 10 Deployment and Imaging Tools”. If you are not able to find the shortcut, please follow these instructions:
  - Right-click the desktop and select, Create Shortcut
  - Type the following in the shortcut box, cmd.exe /k “C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Deployment Tools\DandISetEnv.bat” (assuming the ADK was installed in the default directory)
  - Click **Next** to continue
  - In the Shortcut Name box, type anything that helps identify the shortcut. For example, “Deployment and Imaging Tools”.
  - Click **Finish** to complete the process.
  - Depending on your environment, it may necessary to configure the Deployment and Imaging Tools environment to run in an elevated session. This is required when using dism.exe. Follow these steps to configure the shortcut to run elevated:
    - Right-click the shortcut created above and select **properties**
    - Click on the **Advanced button**
    - Place a check mark in the “**Run as Administrator**” box
  - Click **OK**, **Apply**, and **OK** to complete the operation
  - Double-click the **shortcut** to open the ADK shell environment.

---

**Important note**

For the sake of simplicity, the term “ADK CMD session” will be used to refer to this shell environment throughout the remainder of this document.

---

After the environment has been prepared, it is time to determine the method to be used for Nano Server deployment. If the decision is to deploy using WDS or WinPE, refer to the section titled, “[Creating a custom Nano Server WIM image](#)”. If the decision is to deploy a VHD(x) to physical or virtual machines, refer to the Microsoft documentation at [technet.microsoft.com/en-us/library/mt126167.aspx](http://technet.microsoft.com/en-us/library/mt126167.aspx).

Once the image has been created for the chosen deployment method, proceed to the appropriate section for the given deployment method. After successfully deploying the image, refer to the Appendix on how to manage the newly deployed image. With that, let us get started!

Hewlett Packard Enterprise supports a number of boot scenarios in Windows Server. One such device is the HPE MicroSD card. The next section describes how to enable the MicroSD card for general use.

**Preparing the internal Micro SD card for deployment**

Deploying Nano Server to the MicroSD card is a nice option for situations in which you don’t want to consume an internal drive bay for the OS, particularly when Nano Server fits nicely onto the MicroSD card. The steps for configuring the server to use the MicroSD card are listed below. It is assumed that the Micro SD card is already installed in the HPE ProLiant server’s internal Micro SD slot.

1. Log on to the iLO of the target HPE ProLiant Server using a Web browser.
2. Open an Integrated Remote Console (IRC) session:
  - a. On the iLO Overview screen, click the “.NET” or “Java” link, whichever you prefer, to open an IRC session.
  - b. Follow the prompts to complete the connection
3. Reboot the server and press the F9 System Utilities option.
4. At the System Utilities menu, enable the Internal SD Card Slot as follows:
  - a. Select System Configuration
  - b. Select BIOS/Platform Configuration (RBSU)
  - c. Select System Options
  - d. Select USB Options
  - e. Select Internal SD Card Slot and set to “Enabled”.
  - f. Press F10 to save and <Esc> 4 times to return to System Utilities menu.
  - g. Select Reboot the System.

The internal Micro SD card is now ready for deployment.

**A word about Nano Server images**

Regardless of the deployment method used, the default Nano Server image that ships with Windows Server 2016 requires customization. Customizations include adding Nano Server packages based on the role the server will play in the environment, the addition of boot critical and HPE drivers, and an unattend.xml file to complete setup. This paper covers only a few customizations and deployment scenarios. The first part of this section covers the general customizations that apply to all deployments regardless of deployment method. The remaining sections are deployment method specific, covering the additional customizations required for that particular deployment method. Again, it is important to reiterate that there are multiple ways to configure and deploy Nano Server, this paper covers just a few to help the reader get started.

---

**Note**

After deploying Nano Server, it is strongly recommended that the administrator immediately perform a Windows Update. Please refer to the “Nano Server Remote Management” section in the Appendix for instructions.

---

## Creating a custom Nano Server WIM image

Located on the Windows Server 2016 media are the tools necessary to create and edit Nano Server images. These tools will be used in our examples.

A few different deployment scenarios support the use of WIM images including deploying from a WDS server and from a WinPE environment. This section describes the process for both scenarios. Refer to the specific deployment section for any special configuration steps related to that particular deployment method.

Before starting, be sure to perform the steps described in the [“Preparing the environment”](#) section above.

---

### Note

The WIM image contains 2 separate editions:

- Windows Server 2016 SERVERSTANDARDNANO, aka “Standard”
- Windows Server 2016 SERVERDATACENTERNANO aka “Datacenter”

The reader will have to decide which edition of Nano Server to use prior to creating a customized image. The edition is specified during image creation using the argument, **-Edition**, followed by “Standard” or “Datacenter”.

---

1. Open an administrative PowerShell session on the technician computer. Be sure the PowerShell “executionpolicy” is correctly configured. For example, in the PowerShell session, type `Set-Executionpolicy -Executionpolicy RemoteSigned`.
2. Change directory to the location of the Nano Server image. In our example, it is `c:\ws2016\media\NanoServer`.
3. Located in the NanoServer folder is a subfolder named NanoServerImageGenerator. This folder contains the PowerShell tools for image creation and customization.
  - a. **NanoServerImageGenerator.psm1**—PowerShell Module that includes the following three cmdlets:
    - **New-NanoServerImage**—Used to create Nano Server images
    - **Edit-NanoServerImage**—Used to edit an existing Nano Server image
    - **Get-NanoServerPackage**—Obtains a list of available packages
  - b. **Convert-WindowsImage.ps1**—Script used to convert a Nano Server WIM file to a VHD(x) file.
4. Import the NanoServerImageGenerator.psm1 file into the PowerShell session using the following command, `Import-Module c:\ws2016\media\NanoServer\NanoServerImageGenerator\NanoServerImageGenerator.psm1 -Verbose`

---

### Note

To see the list of the three cmdlets provided by the NanoServerImageGenerator module, type the following command: `Get-Command -Module NanoServerImageGenerator`. The command should return the above three cmdlets. To obtain help on using the cmdlets, type `Get-Help <Name>`, where <Name> is the name of the cmdlet. For example, `Get-Help New-NanoServerImage`.

---

5. The next step is to determine which Nano Server packages to add to the image. For a complete list of packages, refer to the Microsoft website for the latest information. For this exercise, Nano Server will be installed on a bare-metal server targeted to run Hyper-V, we will specify a hostname of “hpenanotest”, enable the remote management port, and enable EMS functionality. Finally, we will add the following packages and drivers to the image:
  - Compute
  - HPE driver package
6. The following is the command that will be used to create our custom WIM image:

7. **New-NanoServerImage -MediaPath c:\ws2016\media -Basepath c:\ws2016\media\NanoServer\base -Targetpath c:\ws2016\media\NanoServer\images\NanoServerCompute.wim -Compute -Computername hpenanotest -DeploymentType Host -Edition Standard -DriverPath c:\spp\win-driverpack-10.60 -EnableEMS -EnableRemoteManagementPort -Verbose**

---

**Note**

For uefi secureboot support add

**-package "Microsoft-NanoServer-SecureStartup-Package**

---

8. The script should prompt you for the Administrator password.
9. The script will take a few minutes to run. Verify there were no errors during image creation.
10. Upon successful script execution, the Nano Server image can now be deployed using methods outlined in this paper.

---

**Note**

It is possible to add an unattend file to the image during image creation. This is useful for customizing Nano Server such as adding user accounts and modifying the firewall settings. For more information, refer to the online Microsoft documentation.

---

11. Should you require a driver that is not available in the win-driverpack-10.60, add the Nano OEM drivers package by editing the image as follows: **Edit-NanoServerImage -BasePath c:\ws2016\media\NanoServer\base -TargetPath c:\ws2016\media\NanoServer\images\NanoServerCompute.wim -OEMDrivers -Verbose**

---

**Note**

For uefi secureboot, add

**-package "Microsoft-NanoServer-SecureStartup-Package**

---

## Deploying a Nano Server WIM from WinPE

One of the deployment methods for Nano Server is deploying a WIM image to disk from a WinPE boot environment. There are multiple options available for booting into WinPE on HPE ProLiant servers. This document covers four such scenarios:

- Booting a WinPE ISO image from a web server using the ProLiant feature, **Boot from URL**
- Booting a WinPE ISO image from the iLO4 Integrated Remote Console (IRC) virtual media feature
- Booting WinPE from a USB thumb drive
- Booting WinPE from the network using WDS and PXE.

The reader can choose the method best suited for his/her environment. Here are the topics that will be covered in this section:

1. Setting up the WinPE environment
2. Creating a WinPE ISO image for ISO based deployments
3. Creating a WinPE bootable thumb drive for USB based deployments
4. Example deployment from a WinPE ISO image using **Boot from URL**
5. Example deployment from a WinPE ISO image (iLO4)
6. Example deployment from a WinPE USB thumb drive
7. Example deployment from a WinPE PXE boot

## Prerequisites

Before starting, be sure to perform the steps described in the "Preparing the environment" section. In addition, be sure to complete all desired customizations to the Nano Server WIM as described in the section titled, "[Creating a custom Nano Server WIM image](#)".

**Table 8.** HPE Infrastructure components required for deploying Nano Server WIM images**Required components**

<b>Nano Server WIM</b>	Customized WIM image containing required packages and drivers
<b>Technician computer to drive deployment</b>	A computer for creating and customizing WinPE and Nano Server images and connecting to the iLO IRC on the target HPE ProLiant server.
<b>Windows 10 ADK</b>	The Windows 10 ADK contains the tools necessary for working with Windows images.
<b>Supported HPE ProLiant Servers</b>	Refer to the section, "Supported ProLiant servers"
<b>PXE Boot Infrastructure</b>	For PXE boot deployments, see the prerequisites listed in the "Deploying a Nano Server WIM using Windows Deployment Services" section later in this document.

**Creating a custom WinPE boot image**

The ADK includes a batch file that creates an environment for working with WinPE images. This batch file requires two arguments. The first is the architecture type of the target image. Since Nano Server is 64-bit, we will specify amd64. The second argument is the working folder. The default is c:\winpe\_<arch>. In our case, it is c:\winpe\_amd64.

1. In the Deployment and Tools CMD session, change directory to "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment\"
2. Run the copype.cmd file with the appropriate arguments. For example, "copype.cmd amd64 c:\winpe\_amd64"

**Note**

If this folder structure is already in place from an earlier ADK installation, you may want to choose a different folder name, or rename or delete the existing folder prior to running copype.cmd.

3. Ensure there were no errors during the operation.
4. The WinPE boot image is located at c:\winpe\_amd64\media\sources\boot.wim
5. This boot.wim file is used in all WinPE examples.

**Important Tip**

If it is necessary to include specific boot-critical drivers in your custom WinPE boot image, please refer to the section titled **Adding boot-critical drivers to Windows boot images** in the Appendix section below before proceeding.

Deploying Nano Server from a WinPE environment is a relatively straightforward process. The examples shown in this section are manual in nature, but can be automated through scripting. The next step is to decide which WinPE boot method to use.

**Creating a WinPE ISO image**

This step is only required if you will be booting WinPE from an ISO image, such as booting from the iLO Integrated Remote Console (IRC) or the HPE ProLiant feature, **Boot from URL**. Otherwise, skip to one of the other WinPE deployment sections described in this document.

The ADK ships with a batch file that creates an ISO image from the contents of the c:\winpe\_amd64\media folder. The script name is MakeWinPEMedia.cmd and is located at, "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment".

**Important Tip**

Be sure the boot.wim file you wish to use is located in the c:\winpe\_amd64\media\sources folder, contains customizations desired for your environment, and is not currently mounted.

The script is straightforward, but here is an example:

1. First, make sure your running a ADK CMD session with elevated privileges.
2. Change directory to the folder containing `MakeWinPEMedia.cmd`
3. Make sure the destination folder exists before running the script. For example, `mkdir c:\winpe_amd64\iso`
4. Run the command, `MakeWinPEMedia.cmd /iso c:\winpe_amd64 c:\winpe_amd64\iso\nanoserver_winpe.iso`
5. If the operation was successful, you can skip to one of the ISO deployment sections below.

### Creating a WinPE bootable USB thumb drive

This step is only required if you will be deploying Nano Server from a USB thumb drive. The process for creating a USB bootable WinPE image is nearly identical to creating a bootable ISO image described in the section, [“Creating a WinPE ISO image”](#).

The ADK ships with a batch file that creates a bootable USB WinPE environment using the contents of the `c:\winpe_amd64\media` folder. The script name is `MakeWinPEMedia.cmd` and is located at, “`C:\ProgramFiles (x86)\Windows Kits\10\Assessment and Deployment Kit\Windows Preinstallation Environment`”.

---

### Important Tip

Be sure the `boot.wim` file you wish to use is located in the `c:\winpe_amd64\media\sources` folder, contains any customizations desired for your environment, and is not currently mounted.

---

The script is straightforward, but here is an example:

1. First, make sure your running a ADK CMD session with elevated privileges.
2. Change directory to the folder containing `MakeWinPEMedia.cmd`.
3. Insert the USB thumb drive in the technician computer and be sure Windows assigns it a drive letter and that the drive partition style is MBR.  
If there is no drive letter assigned, format the USB drive as FAT32 and assign it one. Make a note of the drive letter. For this example, we will assume the USB drive letter is E.
4. Run the command, `MakeWinPEMedia.cmd /ufd c:\winpe_amd64 E:`
5. If the operation was successful, the USB drive is ready.
6. Safely eject the USB drive from the technician computer.
7. To deploy Nano Server using the USB drive, refer to the section, [“Example deployment from a WinPE USB thumb drive”](#).

### WinPE Deployment Examples

#### Example deployment from a WinPE ISO image using Boot from URL.

This example involves booting WinPE from an ISO image hosted on a web server such as Windows Internet Information Services (IIS) and assumes the following prerequisites have been met:

1. The reader has already created a custom WinPE ISO image as outlined in the section titled, [“Creating a WinPE ISO image”](#).
2. The reader has a functioning web server that has been configured to host the WinPE ISO image.
3. The reader has created a custom Nano Server WIM image as outlined in the section titled, [“Creating a custom Nano Server WIM image”](#).

---

### Note

The “Boot from URL” option requires an HPE ProLiant Gen9 server with Boot Mode set to UEFI Mode.

---

1. Log on to the iLO of the target HPE ProLiant Server using a Web browser.
2. Open an Integrated Remote Console (IRC) session:
  - a. On the iLO Overview screen, click the ".NET" or "Java" link, whichever you prefer, to open an IRC session
  - b. Follow the prompts to complete the connection
3. Enable **Boot from URL** on an HPE ProLiant Gen9 server as follows:
  - a. Power on or reboot the server
  - b. During boot press the F9 key to access the System Utilities menu
  - c. Select System Configuration and then select BIOS/Platform Configuration (RBSU)
  - d. Select Boot Options and verify Boot Mode is set to UEFI Mode. Press "Esc"
  - e. Select Network Options and then select Pre-Boot Network Settings
  - f. Select **Boot from URL** and enter the URL to the location of the WinPE ISO image created during the "Creating a WinPE ISO image" section earlier in this document. For example, [http://mywebserver/iso/nanoserver\\_winpe.iso](http://mywebserver/iso/nanoserver_winpe.iso)
  - g. Press F10 key to save
4. The BIOS checks to see if the URL is valid and will display an error if it can't locate the file
5. Escape out to the System Utilities menu and select One-Time Boot Menu
6. At the Boot menu select the URL File option near the bottom of the list

After WinPE boots, a cmd.exe shell opens and wpeinit runs to initialize networking. After wpeinit finishes, the CMD shell remains open. This shell will be used to complete deployment
7. The next step is to initialize the target disk used for Windows installation. Refer to the UEFI example in the **Disk Configuration for WinPE Deployments** section of the Appendix.
8. The next step is to apply the Nano Server image to the Windows partition:
  - a. Ensure the Nano Server WIM file is available. This paper assumes the Nano Server WIM file is located on the technician computer with a mapped drive letter of z:\.
  - b. `Dism.exe /apply-image /imagefile:z:\ws2016\media\NanoServer\images\NanoServerCompute.wim /index:1 /applydir:w:\`
9. The next step is to create the BCD store:  
`Bcdboot.exe w:\Windows`
10. Optional: Enable EMS in the BCD store. Refer to the Appendix section, Enabling EMS for WinPE Deployment. This section can be skipped if you will not be using EMS or if you specified "-EnableEMS" argument during image creation like shown in our example above.
11. This completes Nano Server deployment.
12. Reboot the server:  
`Wpeutil.exe reboot`
13. The server may reboot a couple of times and then should boot into Nano Server and display a text based logon screen. After logging on, the text-based screen presents basic information such as computer name, workgroup or domain, OS version, and a few other options. Note: If the Nano Server Recovery Console doesn't respond, reboot Nano Server and retry the operation.
14. Apply the latest updates from Windows Update. Please refer to [Performing Windows Update on Nano Server](#) for more information.

**Example deployment from a WinPE ISO image using iLO Integrated Remote Console (IIRC)**

This example involves booting WinPE from an ISO image using the iLO Integrated Remote Console virtual media feature. It assumes the reader has already created a custom WinPE image as outlined in the section titled, “[Creating a WinPE ISO image](#)”, as well as created a custom Nano Server image as outlined in the section titled, “[Creating a custom Nano Server WIM image](#)”.

1. Log on to the iLO of the target HPE ProLiant Server using a Web browser.
2. Open an Integrated Remote Console (IRC) session:
  - a. On the iLO Overview screen, click the “.NET” or “Java” link, whichever you prefer, to open an IRC session.
  - b. Follow the prompts to complete the connection
3. Mount the ISO image in the IRC:
  - a. On the menu bar, click **Virtual Drives** -> Image File CD-ROM/DVD
  - b. Browse to the ISO image created above, and click the **Open** button.
4. The ISO is now mounted and available in the IRC.
5. If necessary, reboot the server to the Boot Selection Menu.
6. At the boot menu:
  - a. On a UEFI system select the boot option, “iLO Virtual USB 2 : HPE iLO Virtual USB CD/DVD ROM”
  - b. On a BIOS system, select “**One Time Boot to CD-ROM**”

---

**Note**

If you do not see this or similar iLO virtual boot option, ensure the ISO image is properly mounted. On the IRC menu select **Virtual Drives** and verify a check mark is in the box next to **Image File CD-ROM/DVD**. If not, repeat the steps above to mount the ISO image.

After WinPE boots, a cmd.exe shell opens and wpeinit runs to initialize networking. After wpeinit finishes, the CMD shell remains open. This shell will be used to finish deployment.

---

7. The next step is to initialize the target disk used for the Windows. Refer to the example in the “[Disk Configuration for WinPE Deployments](#)” section of the Appendix based on the boot mode of the server:
8. The next step is to apply the Nano Server image to the Windows partition:
  - a. Ensure the Nano Server WIM file is available. This paper assumes the Nano Server WIM file is located on the technician computer with a mapped drive letter of z:\.
  - b. `Dism.exe /apply-image /imagefile:z:\ws2016\media\NanoServer\images\NanoServerCompute.wim /index:1 /applydir:w:\`
9. The next step is to create the BCD store:  
`Bcdboot.exe w:\Windows`
10. Optional: Enable EMS in the BCD store. Refer to the Appendix section, Enabling EMS for WinPE Deployment. This section can be skipped if you will not be using EMS or if you specified “-EnableEMS” argument during image creation like shown in our example above.
11. This completes Nano Server deployment.
12. Unmount the ISO image from the IRC:
  - a. Click **Virtual Drives** on the iLO IRC menu bar
  - b. Clear the check mark next to Image File CD-ROM/DVD
13. Reboot the server:  
`Wpeutil.exe reboot`



14. The server may reboot a couple of times and then should boot into Nano Server and display a text based logon screen. After logging on, the text-based screen presents basic information such as computer name, workgroup or domain, OS version, and a few other options. Note: If the Nano Server Recovery Console doesn't respond, reboot Nano Server and retry the operation. Although the Nano Server Recovery Console provides for some basic configuration, other methods exist for managing Nano Server. Please refer to the "[Nano Server Remote Management](#)" section in the Appendix for more information.
15. Apply the latest updates from Windows Update. Please refer to [Performing Windows Update on Nano Server](#) for more information.

### Example deployment from a WinPE USB thumb drive

This example involves booting WinPE from a USB thumb drive. It assumes the reader has already created a custom WinPE image as outlined in the section titled, "[Creating a WinPE bootable USB thumb drive](#)", as well as created a custom Nano Server image as outlined in the section titled, "[Creating a custom Nano Server WIM image](#)".

This example requires access to the physical server.

1. Insert the WinPE USB thumb drive into an available USB port on the HPE ProLiant Server.
2. Access the server through the iLO IRC or local KVM (keyboard, video, and mouse).
3. If necessary, reboot the server and press the **F11 Boot Menu** option.
4. At the boot menu, scroll down to the boot entry associated with the USB drive. For example, if the USB drive is inserted in the front panel USB connector, the option might be similar to, `Front USB 1 : <USB device name>`.
5. Highlight the entry and press the **Enter key**.
6. The server should start the WinPE boot process.

After WinPE boots, a `cmd.exe` shell opens and `wpeinit` runs to initialize networking. After `wpeinit` finishes, the CMD shell remains open. We will use this shell to finish deployment.

The next step is to initialize the target disk used for the Windows. Refer to the example in the "[Disk Configuration for WinPE Deployments](#)" section of the Appendix based on the boot mode of the server.

7. The next step is to apply the Nano Server image to the Windows partition created in the previous step:
  - a. Ensure the Nano Server WIM file is available. This paper assumes the Nano Server WIM file is located on the technician computer with a mapped drive letter of `z:\`.
  - b. `Dism.exe /apply-image /imagefile:z:\ws2016\media\NanoServer\images\NanoServerCompute.wim /index:1 /applydir:w:\`

8. The next step is to create the BCD store:

```
Bcdboot.exew:\Windows
```

9. Optional: Enable EMS in the BCD store. Refer to the Appendix section, Enabling EMS for WinPE Deployment. This section can be skipped if you will not be using EMS or if you specified "`-EnableEMS`" argument during image creation like shown in our example above.
10. This completes Nano Server deployment.
11. You can now remove the USB drive from the server
12. Reboot the server:

```
Wpeutil.exe reboot
```

13. The server may reboot a couple of times and then should boot into Nano Server and display a text based logon screen. After logging on, the text-based screen presents basic information such as computer name, workgroup or domain, OS version, and a few other options. Note: If the Nano Server Recovery Console doesn't respond, reboot Nano Server and retry the operation. Although the Nano Server Recovery Console provides for some basic configuration, other methods exist for managing Nano Server. Please refer to the "[Nano Server Remote Management](#)" section in the Appendix for more information.
14. Apply the latest updates from Windows Update. Please refer to [Performing Windows Update on Nano Server](#) for more information.

### Example deployment from a WinPE PXE boot

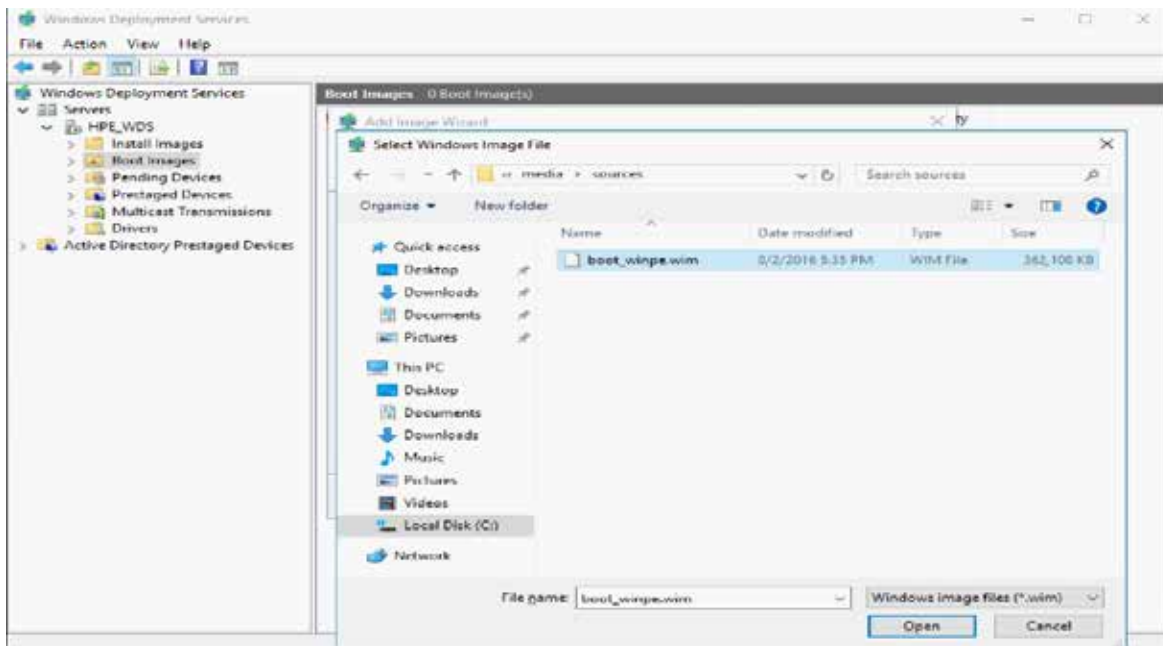
The final example of deploying Nano Server from WinPE involves PXE booting WinPE from a Windows Deployment Services (WDS) Server. The following walkthrough assumes WDS is already installed and operational in the environment, a custom WinPE image exists as described in the section titled, “[Creating a custom WinPE boot image](#)”, and a custom Nano Server image as outlined in the section titled, “[Creating a custom Nano Server WIM image](#)”.

1. Ensure the Nano Server WIM file is available on the network. This paper assumes the Nano Server WIM file is located on the technician computer with a mapped drive letter of z:\.
2. Be sure the boot.wim file created in the section titled, “[Creating a custom WinPE boot image](#)” is available on the network.

### Important Tip

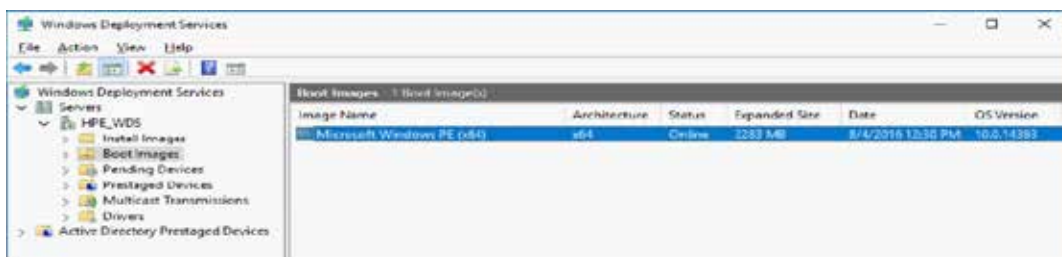
If your chosen hardware configuration requires a specific boot-critical driver in your custom WinPE boot image, please refer to the section titled **Adding boot critical drivers to Windows boot images** in the Appendix section below before proceeding.

3. Copy the boot.wim file from the c:\winpe\_amd64\media\sources folder on the technician computer to a local folder on the WDS server
4. [Optional] Rename the boot.wim file to something more descriptive like, boot\_winpe.wim.
5. Add the custom WinPE image to the WDS server configuration:
  - a. Open the **WDS Management console**
  - b. Right click the **Boot Images** container and select, Add Boot image
  - c. Browse to the location of the WinPE image and select **Open**



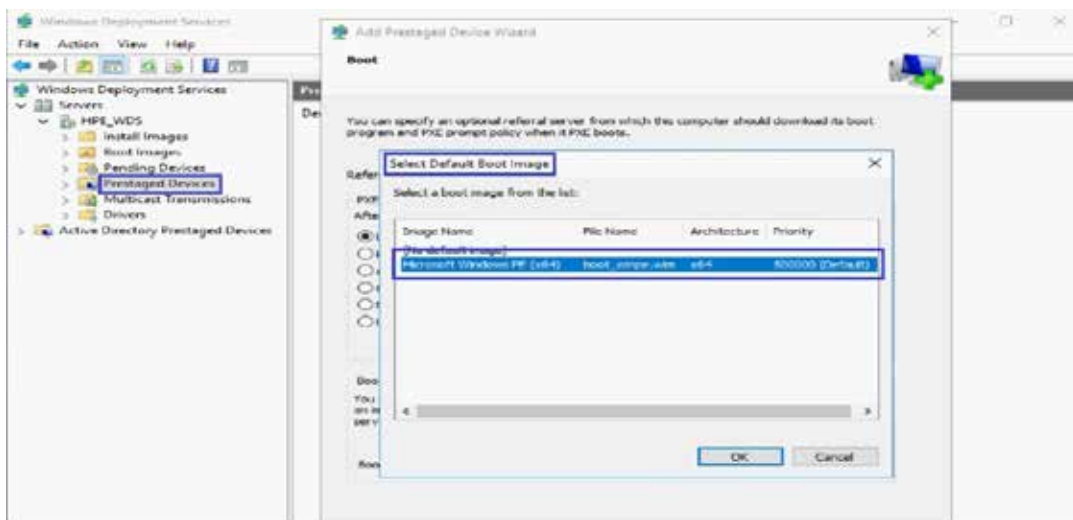
**Figure 5:** Add WinPE boot image to WDS server

- d. Click **Open** and then **Next**
- e. On the **Image Metadata** screen, accept the default **Image Name** and **Image Description** or customize for your environment, and click **Next**.
- f. Click **Next** and **Finish** to complete the operation.
- g. The newly added image should appear in the **Boot images** container as, **Microsoft Windows PE (x64)**.



**Figure 6:** WinPE successfully added to the WDS server

6. The next step is to PXE boot the target server. Depending on your environment, you may need to first prestage the server in WDS. For more information, refer to the section titled [“Configuring prestaged devices in WDS”](#) located in the section titled, [“Deploying Nano Server using Windows Deployment Services”](#).
  - a. If you are prestaging the target server in WDS and are specifying the boot image, be sure to associate the newly added WinPE image to the prestaged device added to the WDS server in step 2 above. For example:



**Figure 7:** Configuring the default boot image for a prestaged device to the WinPE image

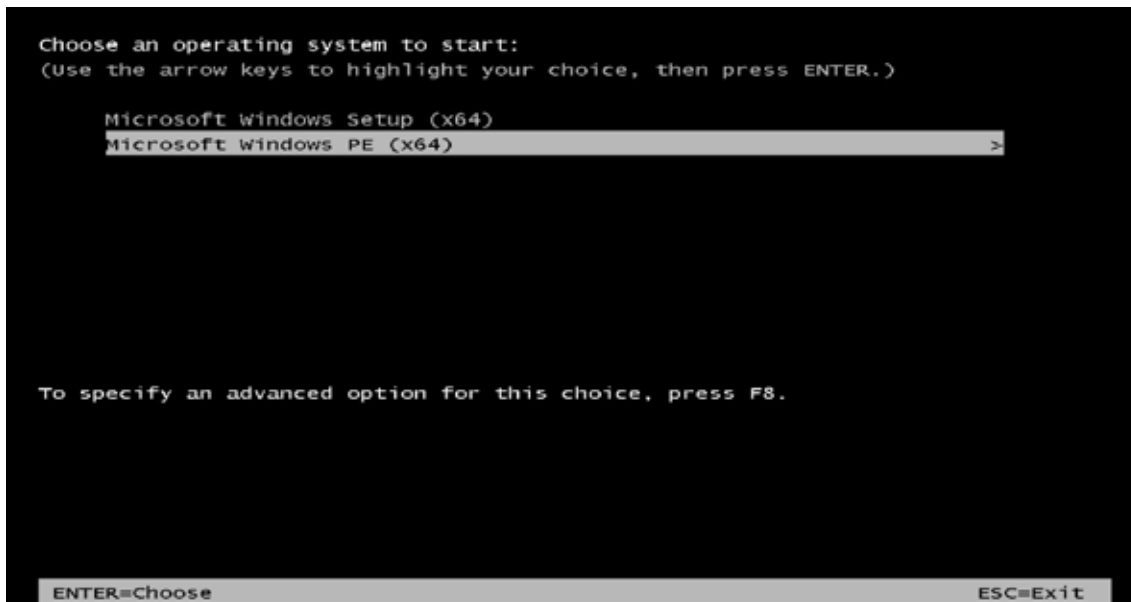
- b. Click on the **Client Unattend** tab and be sure to clear the Unattend File box as shown below:



**Figure 8:** Ensure an Unattend file is not associated with a prestaged device

- c. Click **Apply** and **OK** to complete the operation.

7. The next step is to PXE boot the target server. For instructions, see related section in the chapter, [Deploying Nano Server using Windows Deployment Services](#). If configured correctly, you should see the WinPE boot entry added to the WDS server in step 2 above. If the default image name was used, the entry should be similar to that highlighted below:



**Figure 9:** WinPE is the default boot image of the prestaged device

After WinPE boots, a cmd.exe shell opens and wpeinit runs to initialize networking. After wpeinit finishes, the CMD shell remains open. We will use this shell to finish deployment.

8. The next step is to initialize the target disk used for the Windows. Refer to the example in the [“Disk Configuration for WinPE Deployments”](#) section of the Appendix based on the boot mode of the server.
9. The next step is to apply the Nano Server image using dism.exe to the Windows partition created in the previous step:
- Ensure the Nano Server WIM file is available on the network. This paper assumes the Nano Server WIM file is located on the technician computer with a mapped drive letter of `z:\`.
  - `dism.exe /apply-image /imagefile:z:\ws2016\media\NanoServer\images\NanoServerCompute.wim /index:1 /applydir:w:\`
10. The next step is to create the BCD store:
- ```
Bcdboot.exew:\Windows
```
11. Optional: Enable EMS in the BCD store. Refer to the Appendix section, Enabling EMS for WinPE Deployment. This section can be skipped if you will not be using EMS or if you specified “-EnableEMS” argument during image creation like shown in our example above.
12. This completes Nano Server deployment.
13. Reboot the server:
- ```
Wpeutil.exe reboot
```
14. The server may reboot a couple of times and then should boot into Nano Server and display a text based logon screen. After logging on, the text-based screen presents basic information such as computer name, workgroup or domain, OS version, and a few other options. Note: If the Nano Server Recovery Console doesn't respond, reboot Nano Server and retry the operation. Although the Nano Server Recovery Console provides for some basic configuration, other methods exist for managing Nano Server. Please refer to the [“Nano Server Remote Management”](#) section in the Appendix for more information.
15. Apply the latest updates from Windows Update. Please refer to [Performing Windows Update on Nano Server](#) for more information.

## Deploying Nano Server using Windows Deployment Services

Deploying Nano Server using Windows Deployment Services (WDS) can be accomplished using a Nano Server image in WIM format that contains customizations including Nano Server packages and drivers, and adding the image to the WDS server.

This section describes using Microsoft Windows Deployment Services (WDS) to perform a bare-metal deployment of Nano Server. For instructions on how to create and modify WIM files, please refer to the section [“Creating a custom Nano Server WIM image”](#).

The process described in this section configures the WDS environment for automated deployments using two unattend.xml files. The first is associated with the image for automating post-WinPE stages of setup and is usually referred to as the “image unattend file”. The second discussed below is associated with a prestaged device for customizing the WinPE phase of setup. It is commonly referred to as the “client unattend file”. If you aren’t familiar with unattend files, samples are included in the Appendix and should be created before starting this section.

### Prerequisites

This document assumes the reader is familiar with WDS and a functioning WDS server infrastructure is already in place. For more information, consult the “Windows Deployment Services Getting Started Guide for Windows Server 2012” at [technet.microsoft.com/en-us/library/jj648426.aspx?f=255&MSPPErrors=-2147217396](http://technet.microsoft.com/en-us/library/jj648426.aspx?f=255&MSPPErrors=-2147217396)

Here is a list of components required for WDS:

**Table 9.** Required components for deploying Nano Server using Windows Deployment Services

#### Infrastructure components

<b>Windows Deployment Server (PXE boot server)</b>	<p>The deployment server must:</p> <ul style="list-style-type: none"> <li>• Be accessible from the network</li> <li>• Store operating system images</li> <li>• Store unattend.xml files for automated deployments</li> <li>• Store scripts</li> <li>• Store drivers</li> <li>• Transfer software components to servers over the network</li> <li>• Respond to PXE client requests</li> <li>• Execute the network boot program</li> </ul>
<b>DHCP Server</b>	The DHCP server is responsible for providing TCP/IP based network devices (deployed servers) with valid IP addresses. The DHCP role can be located on the deployment server or can be a standalone server, depending on your network infrastructure and topology.
<b>DNS Server</b>	The DNS server provides name resolution in a TCP/IP based network environment.
<b>Supported HPE ProLiant Servers</b>	Refer to the section, <a href="#">“Supported ProLiant servers”</a> for a complete list of supported servers
<b>Technician computer to drive deployment</b>	The technician computer enables remote access to the target server via iLO IRC and SSH to the iLO Virtual Serial Port (VSP) to manage deployment, and is usually used to access the deployment console.
<b>Network Infrastructure</b>	The network infrastructure configured to support a Windows network based installation (PXE deployment server, DHCP, and DNS servers, firewall, etc.)

In addition to the network components listed above, the following components are also required for a successful deployment and management of Nano Server:

- Microsoft Windows Server 2016 ISO
- Latest windows 10 Assessment and Deployment Kit (ADK)
  - The ADK contains deployment tools such as the Deployment Imaging Servicing and Management (DISM) command line tool, DISM PowerShell cmdlets, Windows System Image Manager (Windows SIM), and so on. The ADK is a free download from the Microsoft download site. DISM is used to perform offline image customization such as adding Windows features and for platform driver injection. Windows SIM is a GUI based utility used to create and modify unattend.xml files.
  - ADK 10 provides support for Windows 10 and Windows Server 2016 image and is available for download at [go.microsoft.com/fwlink/p/?LinkId=526740](http://go.microsoft.com/fwlink/p/?LinkId=526740).

---

**Note**

Previous versions of the ADK are not compatible with Nano Server.

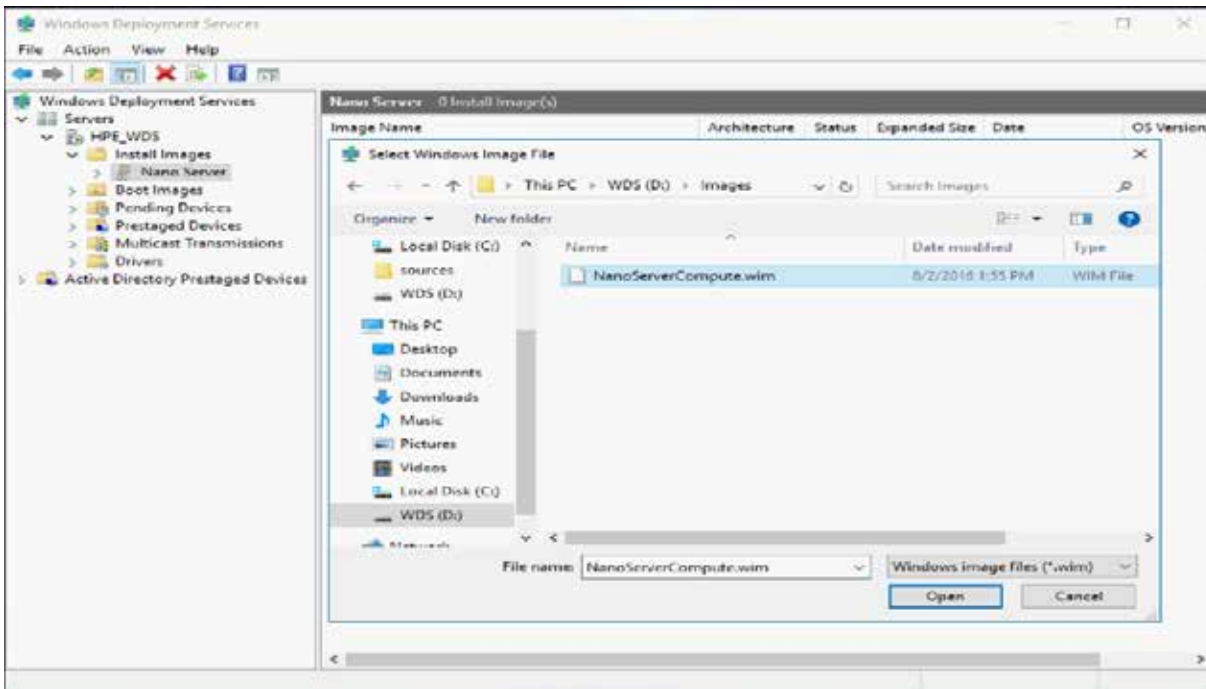
---

- Two unattend files that will work with Nano Server images. Refer to the samples in the Appendix.
- A terminal emulator such as PuTTY to connect to the Virtual Serial port of the HPE ProLiant Server (Optional).

**Adding Nano Server images to the WDS server**

The Nano Server WIM image you wish to deploy must be available on the network or in a local folder such as, `d:\images\NanoServerCompute.wim`. This section describes how to add a Nano Server WIM file to the WDS server and associate a `unattend.xml` file to the image to complete the deployment.

1. The first step is to add the image to an existing image group or create a new group. To create a new install group, open the WDS management console, **right-click** the **Install Images** container and click **Add Image group**. Enter a name for the new image group, for example **Nano Server**.
2. Once the image group has been selected, **right-click** the group and select **Add Install image**.
3. Click **Browse** to locate the custom image at `d:\images\NanoServerCompute.wim` and select it.



**Figure 10:** Adding NanoServerCompute.wim to the Nano Server install image group

4. Click **Open** and **Next** to continue

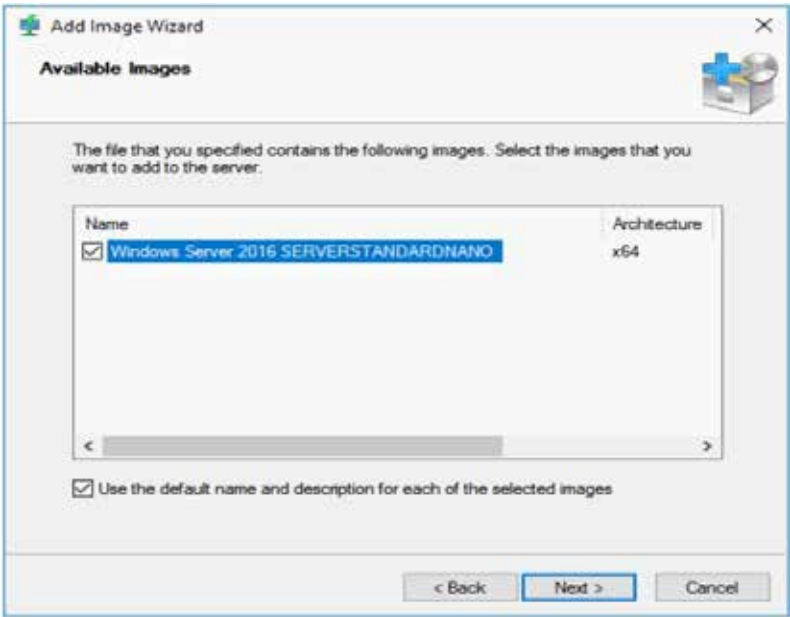


Figure 11: Adding NanoServerCompute.wim images to the Windows Deployment Server.

- 5. Select the image(s) you wish to add to the WDS server.
- 6. Click **Next** to continue
- 7. Click **Next** and then **Finish** to complete the operation
- 8. The image(s) should be added to the Nano Server images folder on the WDS server

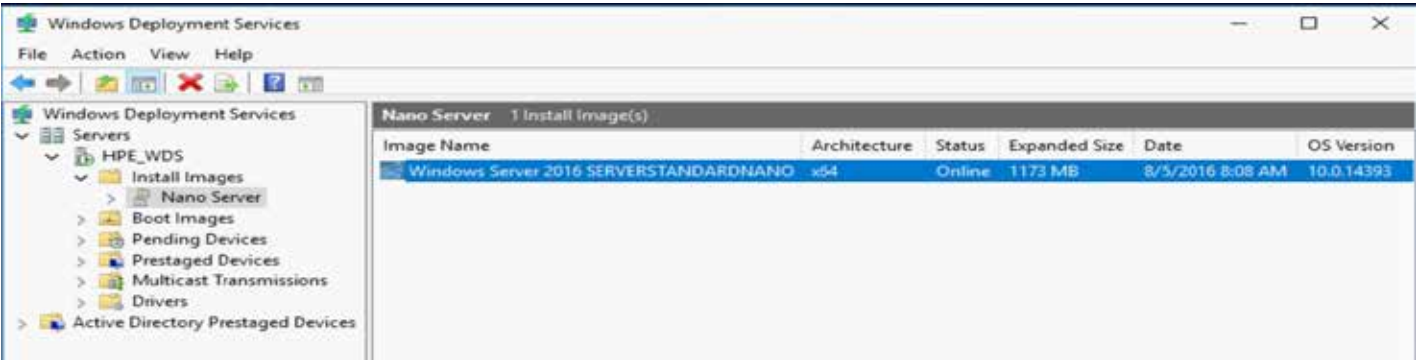


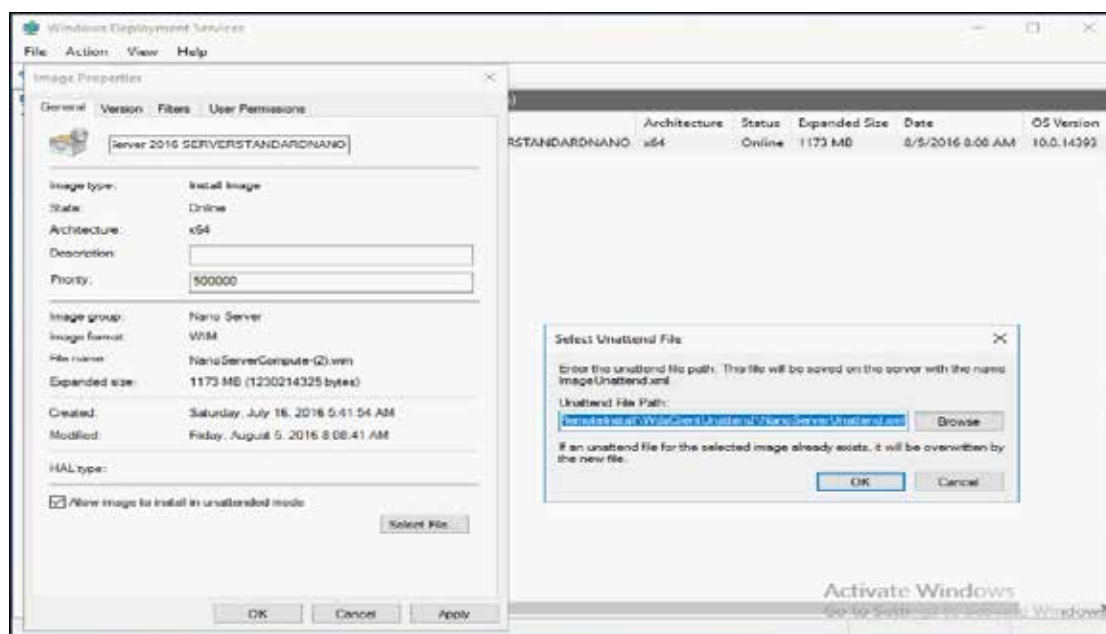
Figure 12: Nano Server successfully added to the WDS server

- 9. After adding the Nano Server image to the WDS server, the next step is to associate an image unattend.xml file to the image in order to automate the installation.

### Associating an image unattend.xml file to a Nano Server image

Automated deployments using WDS require that the unattend xml file is associated to the Nano Server install image as described below.

- First, be sure a Nano Server image exists on the WDS server.
- Next, be sure an image unattend xml file already exists on the server. Refer to the Appendix for a sample image unattend XML file.
- In the WDS Management Console, open the **Install Images** container. In our example, the image group is Nano Server. In the right pane, **right-click** the Nano Server image and select **Properties**.
- Select Allow image to install in unattended mode and click **Select File** to browse to the install image. In this example the image file is located at, `d:\RemoteInstall\WdsClientUnattend\NanoServerUnattend.xml`.



**Figure 13:** Nano Server image properties

- Click **Open** and then click **OK** to continue.
- Click **Apply** and **OK** to complete the operation.
- This completes the process for configuring a Nano Server image for automated deployments.

### TIP

If a change is made to the unattend file in the future, repeat the steps above to reapply the unattend file to the install image. This is necessary because WDS imports the unattend xml file into the folder structure of the associated image on the WDS server. For instance, in our example above, the NanoServerUnattend.xml file is copied to the following location:

`D:\RemoteInstall\Images\Nano Server\NanoServerCompute\Unattend\`



### Customizing the WDS boot environment for Nano Server Deployments

This section covers the steps for adding a boot.wim file to the WDS store, enabling EMS for the new boot entry, and enabling EMS in the WDS boot manager. This capability redirects all Windows setup information to the Windows EMS port allowing for text based SSH connections to the target server for remote management purposes.

#### Adding the Windows Server 2016 boot.wim file to the WDS Server

The boot.wim file found on the Windows Server 2016 media should be used to deploy Windows Server 2016 images using Windows Deployment Services.

The default boot file, boot.wim, included in the sources folder on the media should contain the necessary drivers for most HPE ProLiant servers listed in the supported server list. For more information, refer to the [“Creating a custom Nano Server WIM image”](#) section that describes the controllers requiring out-of-box drivers. If there is a need to modify the boot.wim file to add boot critical drivers, such as the HPE Dynamic Smart Array B140i controller, refer to online help for adding boot critical drivers to Windows images. For customizing existing WDS boot images, refer to the topic, [“Addin boot critical drivers to Windows boot images”](#), located in the Appendix.

The Windows Server 2016 media should be available locally on the WDS server or available on the network. In this example, the contents of the ISO image have been extracted to the folder, c:\ws2016\media.

- Open the WDS Management Console
- Right mouse-click the **Boot Images** container, and select **Add Boot Image**
- Click the **Browse** button to locate the boot.wim file from the WS2016 media. Using our example, it is c:\ws2016\media\sources\boot.wim

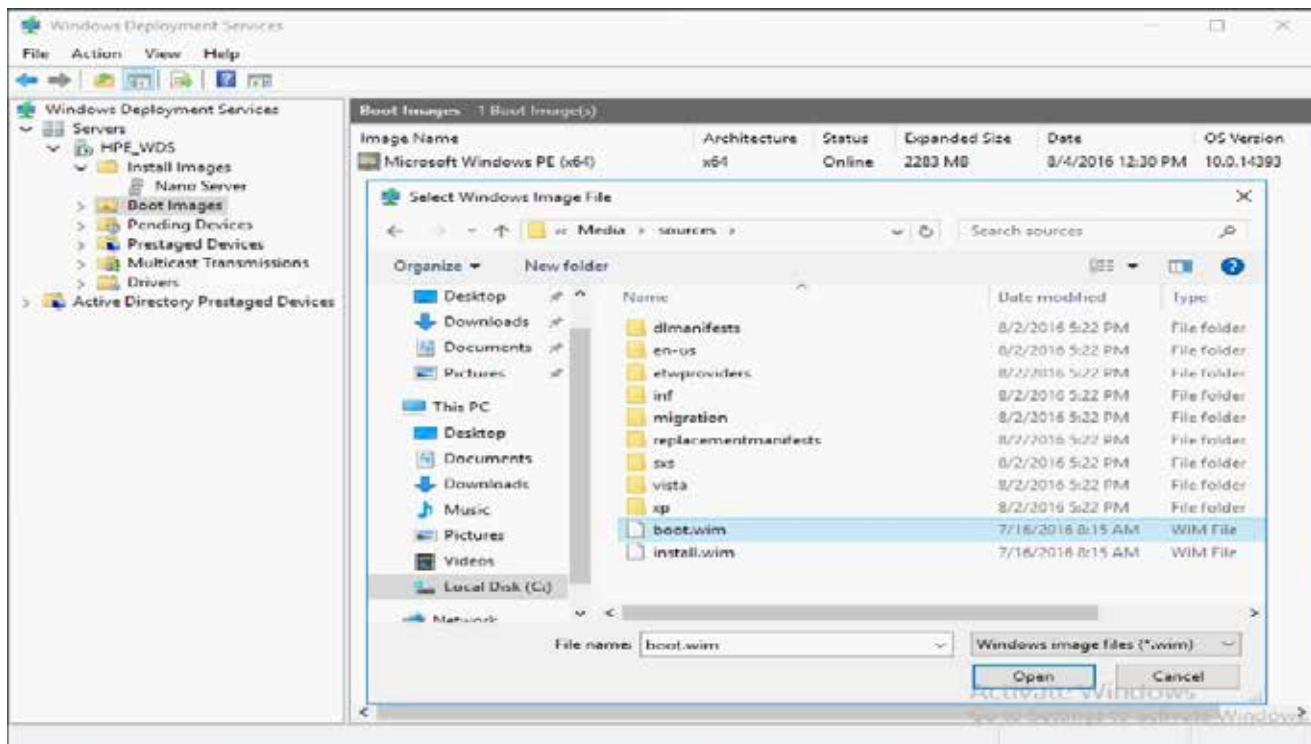
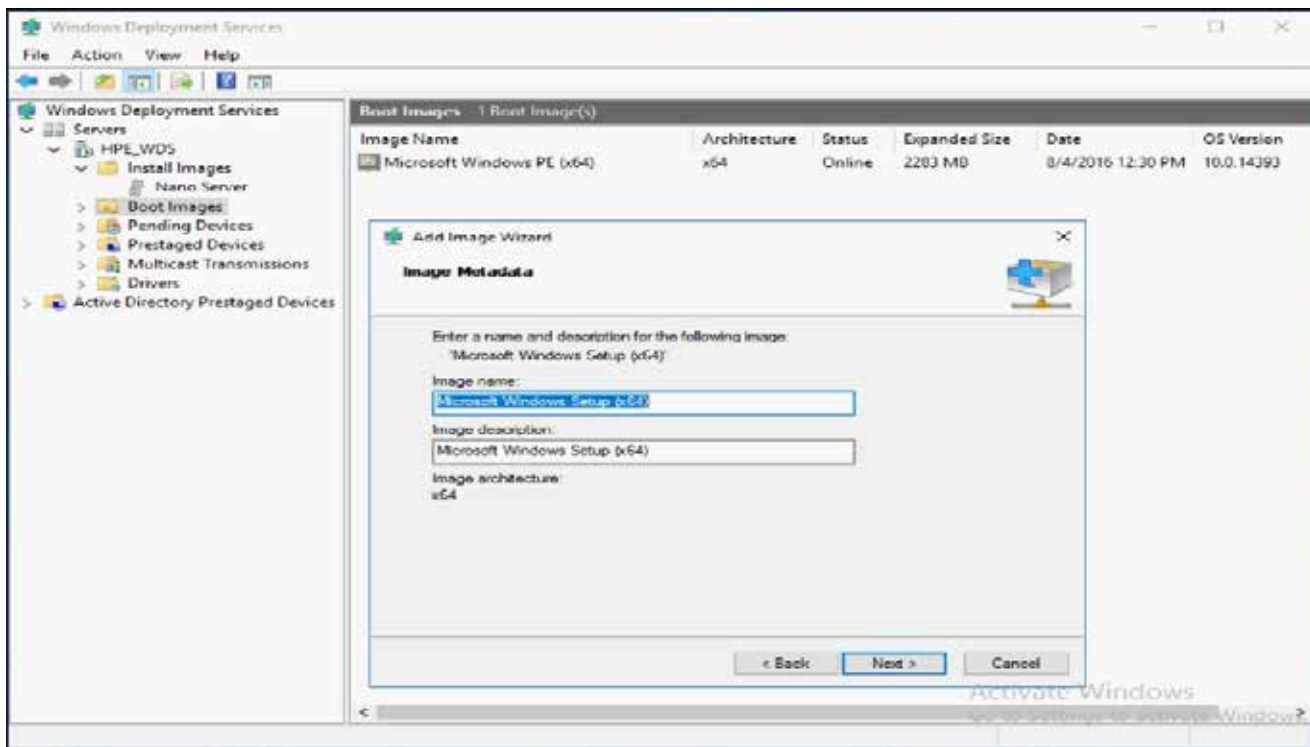


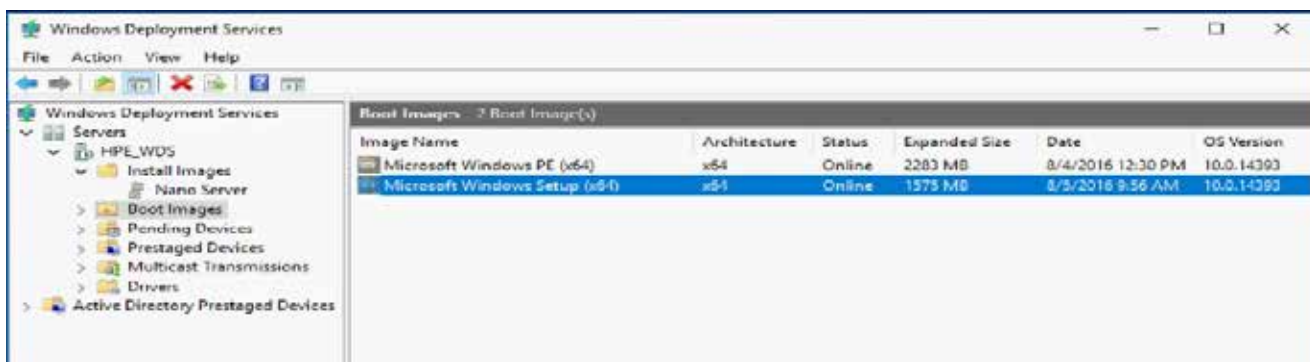
Figure 14: Adding Windows Server 2016 boot image to the WDS server

- Click **Open** and follow the remaining prompts to complete the operation.
  - Optional: Change the default name and description to something more descriptive like, **WS2016 Microsoft Windows Setup (x64)**



**Figure 15:** Configuring Boot image name

- The boot image should now be added to the WDS store



**Figure 16:** Verify the boot image was successfully added to the WDS server

Although the boot.wim file just added required no customizations, the corresponding BCD file should be modified to support EMS for monitoring the deployment process using Windows EMS. At the same time, EMS should also be enabled in the default.bcd file as described in the next section if not already done so.

### Modifying the BCD of the newly added boot file

Every boot.wim file added to WDS includes a corresponding BCD file located in the same folder as the image, \RemoteInstall\boot\x64\images. In our example, we added a second boot image with the default name, boot.wim. Since there is already a boot image with the same name, WDS adds “-(n)” to the tail of the filename to prevent name collisions, where n is a number that is incremented with each new image added. For example, the boot.wim file that just added becomes boot-(2).wim.

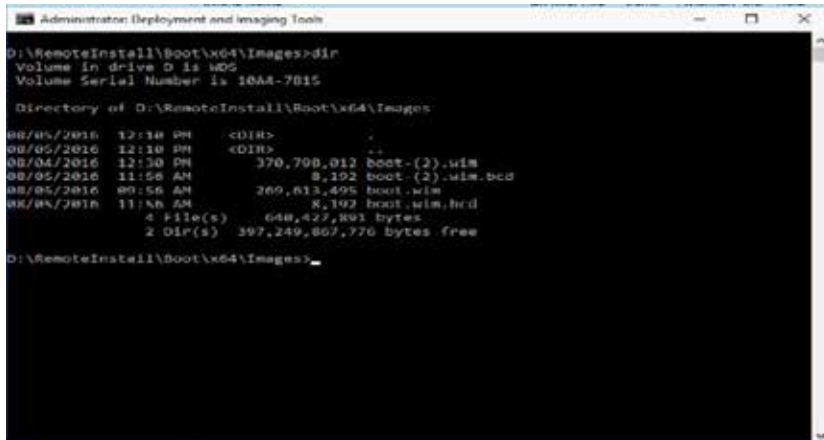


Figure 17: List of boot image and corresponding BCD files added to the WDS server

For each boot image added to WDS, there is a corresponding BCD file. The next step is to modify the BCD file of the newly added image to enable ems and bootems. The steps are as follows:

1. In case of an error, start by making a backup copy of the BCD file corresponding to the image to be modified. Using our example of boot-(2).wim:
2. Open an elevated command prompt to the d:\RemoteInstall\boot\x64\images folder and enter the following command, “Copy boot-(2).wim.bcd” “boot-(2).wim.bcd.orig”
3. Obtain the boot entry identifier string of the newly added boot image. In the \RemoteInstall\boot\x64\images folder, type the following command:  
Bcdedit.exe /store “d:\remoteinstall\boot\x64\images\boot-(2).wim.bcd”

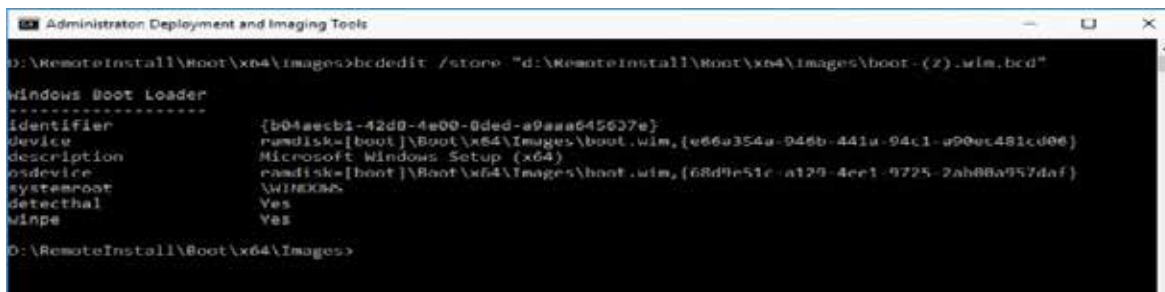


Figure 18: Display BCD settings of the Windows Server 2016 boot image

4. Two things to notice in the screenshot above:
  - a. Under Windows Boot Loader there is no reference to ems or bootems
  - b. The identifier string of the boot loader entry which will be used in the next steps
5. Copy the identifier string to the computer's clipboard as this will be used in the next step
6. Add bootems and ems support to the target BCD file by entering the following commands:
  - a. `bcdedit.exe /store "d:\RemoteInstall\Boot\x64\Images\boot-(2).wim.bcd" /set {GUID} bootems on`
  - b. `bcdedit.exe /store "d:\RemoteInstall\Boot\x64\Images\boot-(2).wim.bcd" /set {GUID} ems on`

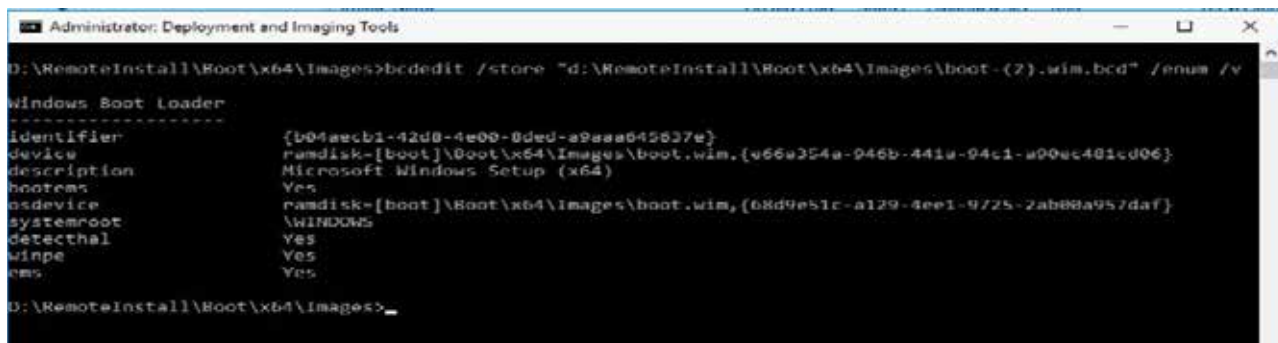
For example, using the identifier in the screen shot above, use the following command:

```
bcdedit.exe /store "d:\RemoteInstall\Boot\x64\Images\boot-(2).wim.bcd" /set {c94281b7-6c73-45a9-b7ac-0465935efe8a} bootems on
```

```
bcdedit.exe /store "d:\RemoteInstall\Boot\x64\Images\boot-(2).wim.bcd" /set {c94281b7-6c73-45a9-b7ac-0465935efe8a} ems on
```

Verify that EMS and BOOTEMS are now configured in the BCD store:

- a. `bcdedit.exe /store "d:\RemoteInstall\Boot\x64\Images\boot-(2).wim.bcd" /enum /v`



**Figure 19:** Verify ems and bootems are configured in the Windows Server 2016 boot image

## Note

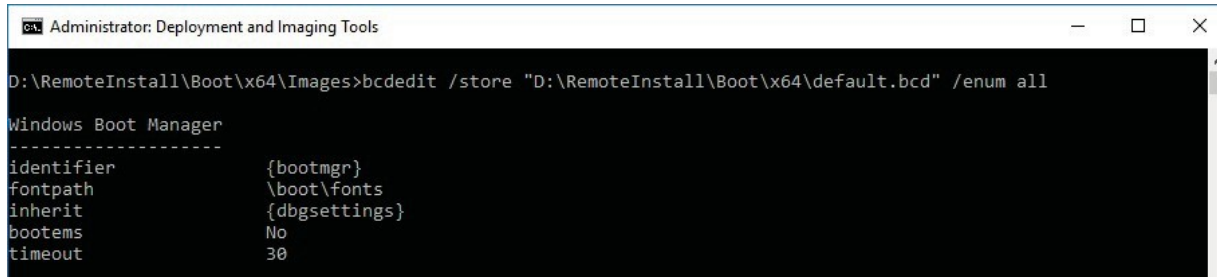
In order for WDS to pick up the changes immediately, the WDS server needs to be restarted. In a CMD shell run the following command:  
`sc control wdsserver 129.`

## Modifying the WDS “default.bcd” BCD file

For each architecture type, there is one default.bcd file that controls the boot settings for that architecture. Since Nano Server is 64-bit, we'll focus on the 64-bit default.bcd file. This process is very similar to the process we just followed for modifying the boot-(2).wim.bcd.

1. Open an elevated command prompt to `\RemoteInstall\Boot\x64`.
2. Make a backup copy of the existing default.bcd in case of error and we need to revert back
  - a. `Copy default.bcd default.bcd.orig`

3. Verify that bootems is not enabled:
4. Enter the following command to display the boot store entries:
  - a. `bcdedit.exe /store "d:\RemoteInstall\Boot\x64\default.bcd" /enum all`

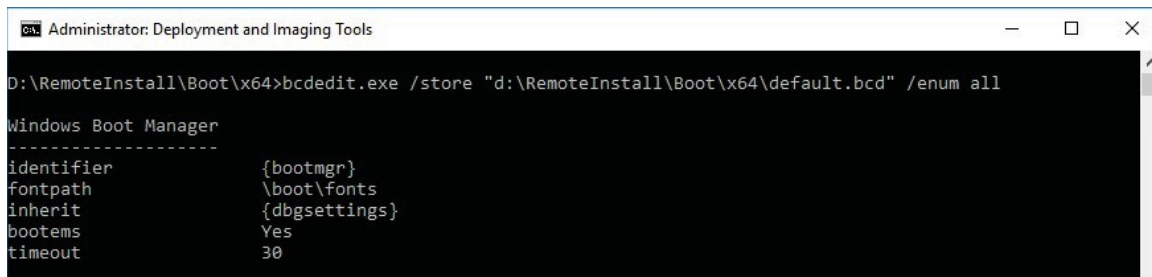


The screenshot shows a command prompt window titled "Administrator: Deployment and Imaging Tools". The command `D:\RemoteInstall\Boot\x64\Images>bcdedit /store "D:\RemoteInstall\Boot\x64\default.bcd" /enum all` has been executed. The output displays the Windows Boot Manager settings for the specified BCD store.

```
Windows Boot Manager
-----
identifier           {bootmgr}
fontpath              \boot\fonts
inherit               {dbgsettings}
bootems               No
timeout               30
```

**Figure 20:** Display BCD settings in the global default BCD store

5. If Bootems is not enabled, enter the following command to enable it:
  - a. `bcdedit.exe /store "d:\RemoteInstall\Boot\x64\default.bcd" /set{bootmgr} bootems on`
6. Enter the following command to verify bootems is enabled in the boot store entry:
  - a. `bcdedit /store "d:\RemoteInstall\Boot\x64\default.bcd" /enum all`



The screenshot shows a command prompt window titled "Administrator: Deployment and Imaging Tools". The command `D:\RemoteInstall\Boot\x64>bcdedit.exe /store "d:\RemoteInstall\Boot\x64\default.bcd" /enum all` has been executed. The output displays the Windows Boot Manager settings, with `bootems` now set to `Yes`.

```
Windows Boot Manager
-----
identifier           {bootmgr}
fontpath              \boot\fonts
inherit               {dbgsettings}
bootems               Yes
timeout               30
```

**Figure 21:** Verify bootems is enabled in the global default BCD store

7. Restart the WDS server to implement the change immediately: `sc control wdsserver 129`
8. After these changes, all output during Windows setup will be redirected to Windows EMS

### Configuring prestaged devices in WDS

In some environments, it may be necessary to prestage devices in WDS. Prestaging devices provides granular control and customization for deployments and is the technique used in this paper. Prestaging a device requires adding the client MAC address or server GUID to the WDS store (or Active Directory if WDS is operating in AD mode) prior to deployment. Devices can be prestaged using the WDS management console or the command line utility, `wdsutil.exe`.

---

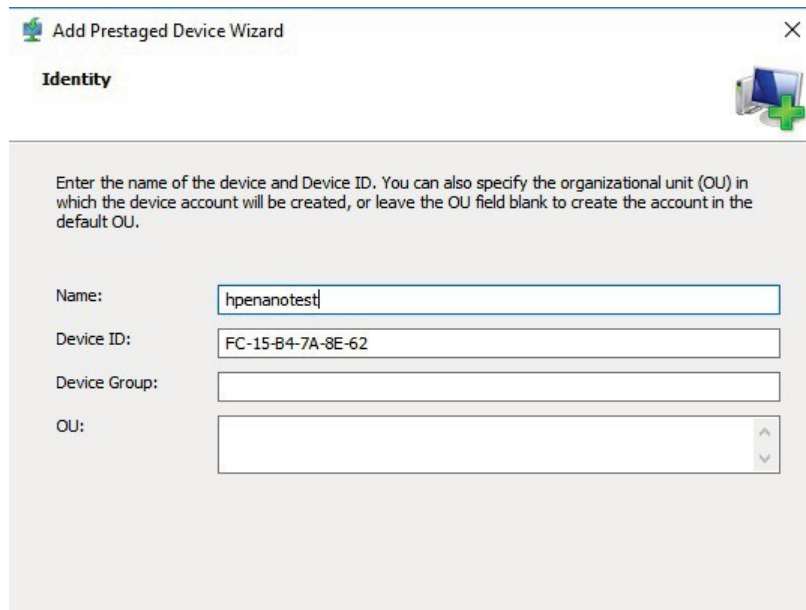
### Note

Adding devices to Active Directory is beyond the scope of this paper. Please refer to the Microsoft Technical documentation for Active Directory information.

---

**Prerequisites:**

- This section requires that user knows the MAC address of the target NIC and network boot is enabled in BIOS.
  - A client unattend xml file already exists on the server. Refer to the Appendix for a sample client unattend XML file.
1. Open the WDS Management Console, select Prestaged Devices, **right-click** and select Add device. The **Add Prestaged** Device Wizard starts.
  2. On the Identity screen, enter a name for the target server in the Name field. Often times the name matches the desired hostname of the server. Click in the Device ID field, and enter the server's MAC address. Remember to use dashes instead of colons as separators for the MAC address.



**Add Prestaged Device Wizard**

**Identity**

Enter the name of the device and Device ID. You can also specify the organizational unit (OU) in which the device account will be created, or leave the OU field blank to create the account in the default OU.

Name:

Device ID:

Device Group:

OU:

**Figure 22:** Add Prestaged Device Wizard

3. Click Next to move to the Boot page.
4. On the **Boot** page, click the **Select** button to select the Windows Server 2016 boot.wim file previously added to the WDS server in the section, [“Adding the Windows Server 2016 boot.wim file to the WDS Server”](#).

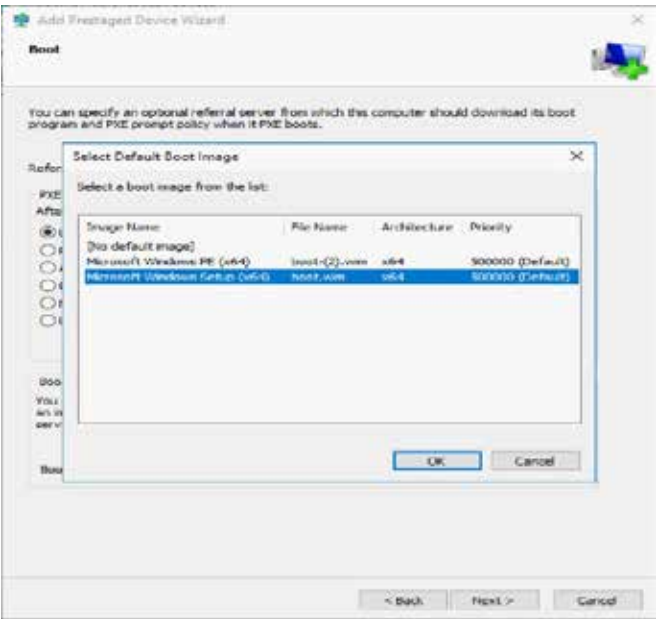


Figure 23: Configuring default boot image for prestaged device

- Click **Next** to continue.
- On the Client Unattend screen, browse to the custom client unattend file previously created. If the target server is a DL580 Gen8 or any Gen 9 server operating in UEFI mode, be sure to select the EFI client unattend file, `NanoServer_Client_UEFI.xml`. Otherwise, select the BIOS unattended file, `NanoServer_Client_BIOS.xml` if the server is operating in BIOS mode. Note: The client unattend file must reside in the `d:\RemoteInstall\WdsClientUnattend` folder.

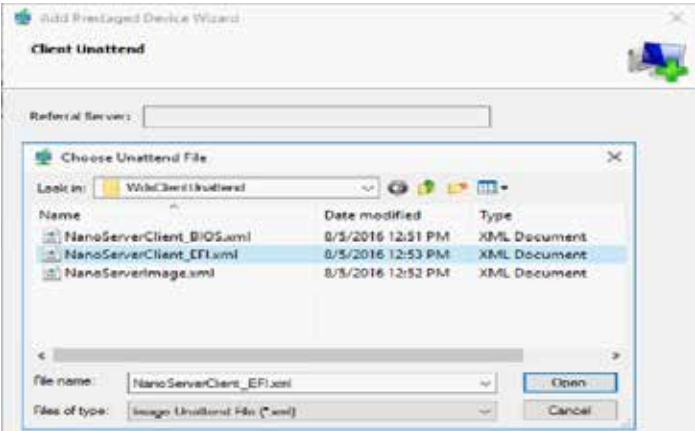


Figure 24: Add client unattend file to prestaged device

- Click **Open**, **Next** and **Finish** to complete the operation.

---

**Note**  
Review or change your settings by right-clicking the device name in the **Prestaged Devices** container and selecting **Properties**.

---

8. A client unattend file is now associated with the prestaged device.
9. The prestaged device is configured for automated deployment.
10. Be sure an image unattend file has been associated with the Nano Server image as described in the section, "[Associating an image unattend.xml file to a Nano Server image](#)".
11. The client and image are configured for fully automated deployments.

### Deploying an image using WDS

The steps in this section describe PXE booting the HPE ProLiant server.

To configure the network boot on the HPE ProLiant server, refer to the section in the Appendix titled, "[Enabling network boot on an HPE ProLiant server](#)".

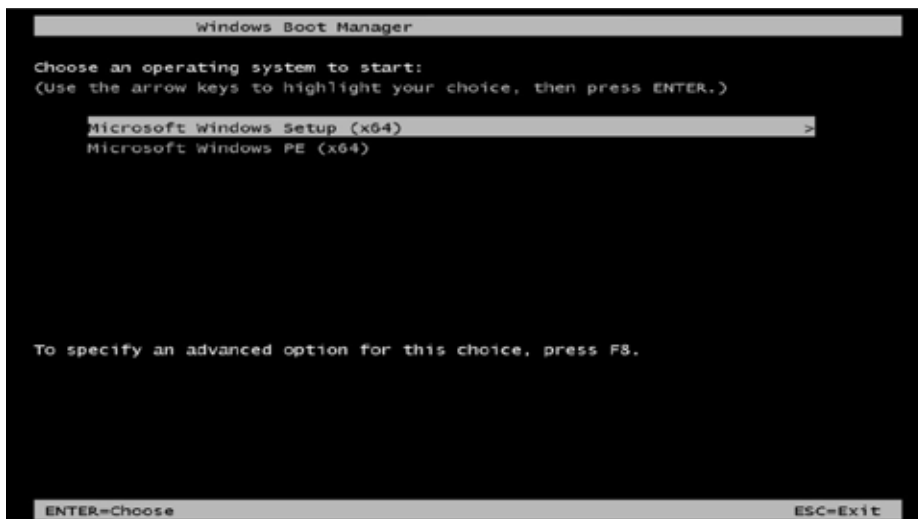
---

### Note

This section assumes the server is prestaged in WDS.

---

1. Be sure the target NIC is enabled for Network Boot in RBSU.
2. Open a session to the iLO IRC as previously described.
3. Power on the server and select **F12** to enter PXE boot.
4. If more than one boot image is displayed on the WDS server, select the image added to the WDS server as described above.



**Figure 25:** PXE Boot image list

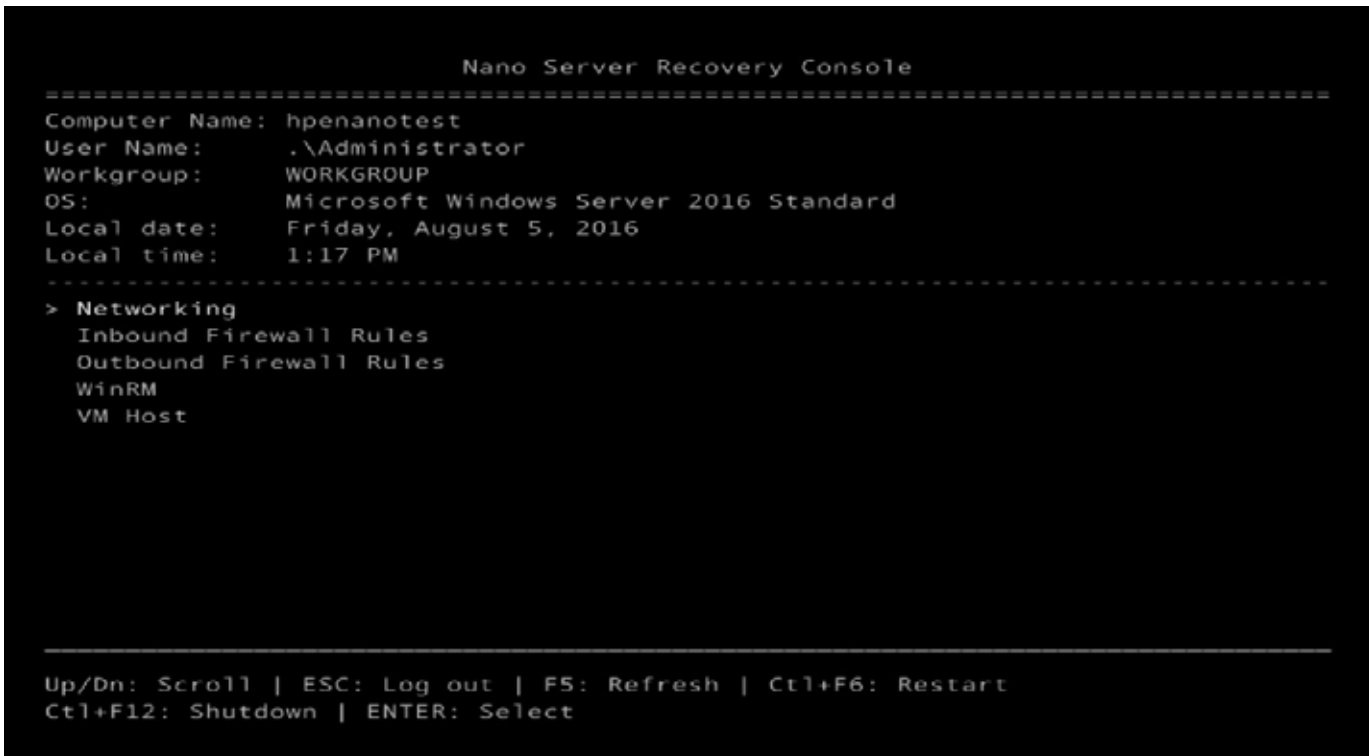


5. The rest of the deployment should be automated.
  - a. At the end of setup, a login screen should appear similar to the screen below:



**Figure 26:** Nano Server Emergency Recovery Console

- b. Enter the credentials configured in the image unattend file to logon to the server and display the Nano Server Recovery Console:



**Figure 27:** Nano Server Emergency Recovery Console listing basic information

As shown in the screen above, there are a limited amount of administrative actions that can be performed from the Nano Server Recovery Console. Although the Nano Server Recovery Console provides for some basic configuration, other methods exist for managing Nano Server. Please refer to the Nano Server Remote Management section in the Appendix for more information.

6. Apply the latest updates from Windows Update. Please refer to [Performing Windows Update on Nano Server](#) for more information.

## Creating a custom Nano Server VHD(x) image

In this section we walk through the process for creating a Nano Server VHD(x) file that can be used for dual-booting Nano Server on a HPE ProLiant Gen9 server booted in UEFI mode.

### Prerequisites

Before starting, be sure to perform the steps described in the “[Preparing the environment](#)” section.

On the technician computer, open an elevated PowerShell session to the Nano Server folder created in the “[Preparing the environment](#)” section of the document. In our example it is `c:\ws2016\media\NanoServer`.

Be sure the PowerShell “executionpolicy” is correctly configured. For example, in the PowerShell session, type `Set-Executionpolicy RemoteSigned`

You should find one PowerShell module and one PowerShell script in the NanoServerImageGenerator folder:

1. **NanoServerImageGenerator.psm1**—PowerShell Module that includes the following three cmdlets:
  - a. **New-NanoServerImage**—Used to create Nano Server images
  - b. **Edit-NanoServerImage**—Used to edit an existing Nano Server image
  - c. **Get-NanoServerPackage**—Obtains a list of available packages
2. **Convert-WindowsImage.ps1**—Script used to convert a Nano Server WIM file to a VHD(x) file.

Import the NanoServerImageGenerator.psm1 file into the PowerShell session using the following command, `Import-Module c:\ws2016\media\NanoServer\NanoServerImageGenerator\NanoServerImageGenerator.psm1 -Verbose`

---

### Note

To see the list of the three cmdlets provided by the NanoServerImageGenerator, type the following command: `Get-Command -Module NanoServerImageGenerator`. The command should return the above three cmdlets. To obtain help on using the cmdlets, type `Get-Help <Name>`, where <Name> is the name of the cmdlet. For example, `Get-Help New-NanoServerImage`

---

Now is a good time to decide which Nano Server packages should be added to the image. For help deciding which package(s) to include, refer to the “[Nano Server Packages](#)” table found in the Appendix.

After you’ve decided which Nano Server package(s) to install, it’s time to create the VHD(x) file. For this example, we’ll install the following packages:

- Compute
- HPE driver Package

**New-NanoServerImage -MediaPath C:\ws2016\media -BasePath c:\ws2016\media\NanoServer\Base -TargetPath C:\ws2016\media\NanoServer\images\nanoserver.vhdx -Compute -Computername hpenanotest -DeploymentType Host -Edition Standard -DriverPath c:\spp\win-driverpack-10.60 -EnableRemoteManagementPort - Verbose**

---

### Note

For uefi secureboot, add  
`-package “Microsoft-NanoServer-SecureStartup-Package”`

---

The script should prompt you for the Administrator password.

The script will take a few minutes to run. Verify there were no errors during image creation.

---

**Note**

Should you require a driver that is not available in the win-driverpack-10.60, add the Nano OEM drivers package by editing the image as follows:

**Edit-NanoServerImage -BasePath c:\ws2016\media\NanoServer\base -TargetPath C:\ws2016\media\NanoServer\images\nanoserver.vhdx -OEMDrivers -Verbose**

---

The VHD(x) image is now ready for deployment. Refer to the section, “Deploying a VHD(x) file to a physical computer in a dual-boot environment”.

**Deploying a VHD(x) file to a physical computer in a dual-boot environment**

This section describes the process for deploying a Nano Server VHD(x) file in a dual-boot environment to a HPE ProLiant Gen9 server booted in UEFI mode.

**Prerequisites**

A Nano Server VHD(x) file has been created as described in the section, “[Creating a custom Nano Server VHD\(x\) image](#)”. Secondly, this section assumes that the target HPE ProLiant server already has Windows Server installed. We'll call this the “safe OS”.

Open an RDP session to the HPE ProLiant Server in which you intend to deploy Nano Server. In this example, the target server is running Windows Server 2016 as the “safe OS”.

Open an administrative PowerShell session. This session will be used for the remaining steps. Create a folder on the local server such as `c:\NanoServer`

Copy the NanoServer VHD(x) file created earlier to the folder created in the previous step. Mount the VHD(x) image using the following command:

```
$mount = Mount-DiskImage c:\NanoServer\nanoserver.vhdx -Passthru
```

---

**Note**

Be sure to include the `-Passthru` argument otherwise obtaining the drive letter using the PowerShell example in the next step will fail. But the drive letter can still be found using Windows explorer.

---

After successfully mounting the image, a new drive letter should have been created. There are a couple of ways to obtain the drive letter: Open a Windows explorer and locate the new local disk and note the drive letter or...

Using PowerShell:

```
$diskImage = $mount | Get-DiskImage  
$disk = $diskImage | Get-Disk  
[$disk | Get-Partition].DriveLetter
```

In our example, we'll assume that the drive letter of the image is 'D'

After you've discovered the drive letter of the mounted image, update the BCD store to point to the Nano Server image using `bcdboot.exe`:  
`Bcdboot.exe <driveLetter>\windows`, where `<driveLetter>` is the letter of the mounted image

---

**Example**

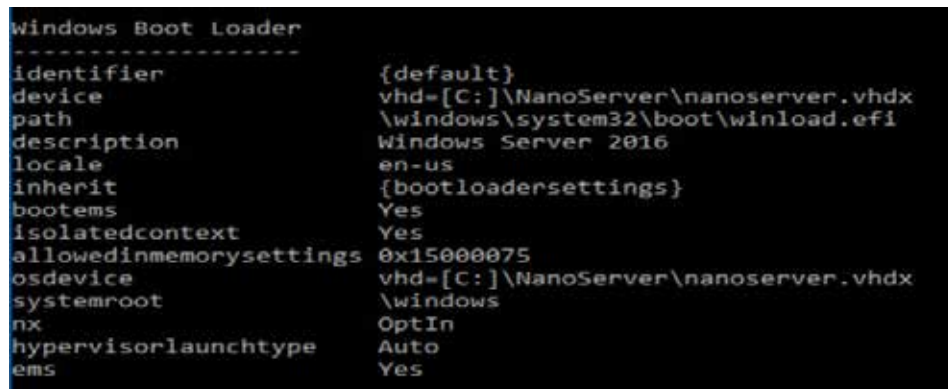
bcdboot.exe d:\Windows

---

If bcdboot.exe worked correctly you should see the following message, “Boot files successfully created.”

Dismount-DiskImage -ImagePath c:\NanoServer\nanoserver.vhdx

Verify the new boot entry in the BCD store by running, bcdedit.exe. You should see a new Windows Boot Loader entry with the “device” setting pointing to the nanoserver VHD(x) file. In our example, vhd=[C:]\nanoserver\nanoserver.vhdx.



```
Windows Boot Loader
-----
identifier                {default}
device                    vhd=[C:]\NanoServer\nanoserver.vhdx
path                      \windows\system32\boot\winload.efi
description               Windows Server 2016
locale                   en-us
inherit                   {bootloadersettings}
bootems                   Yes
isolatedcontext           Yes
allowedinmemorysettings  0x15000075
osdevice                  vhd=[C:]\NanoServer\nanoserver.vhdx
systemroot                \windows
nx                        OptIn
hypervisorlaunchtype      Auto
ems                       Yes
```

**Figure 28:** New Windows Boot Loader entry in the BCD store

Make a note of the identifier string of the boot loader entry. It is most likely to be {default} as shown above

1. **Optional: Change the default description of the new boot entry to be more descriptive**

- We'll assume the identifier string of the newly added boot loader is {default}
- Run bcdedit.exe /set "{default}" description "Windows Nano Server 2016 "

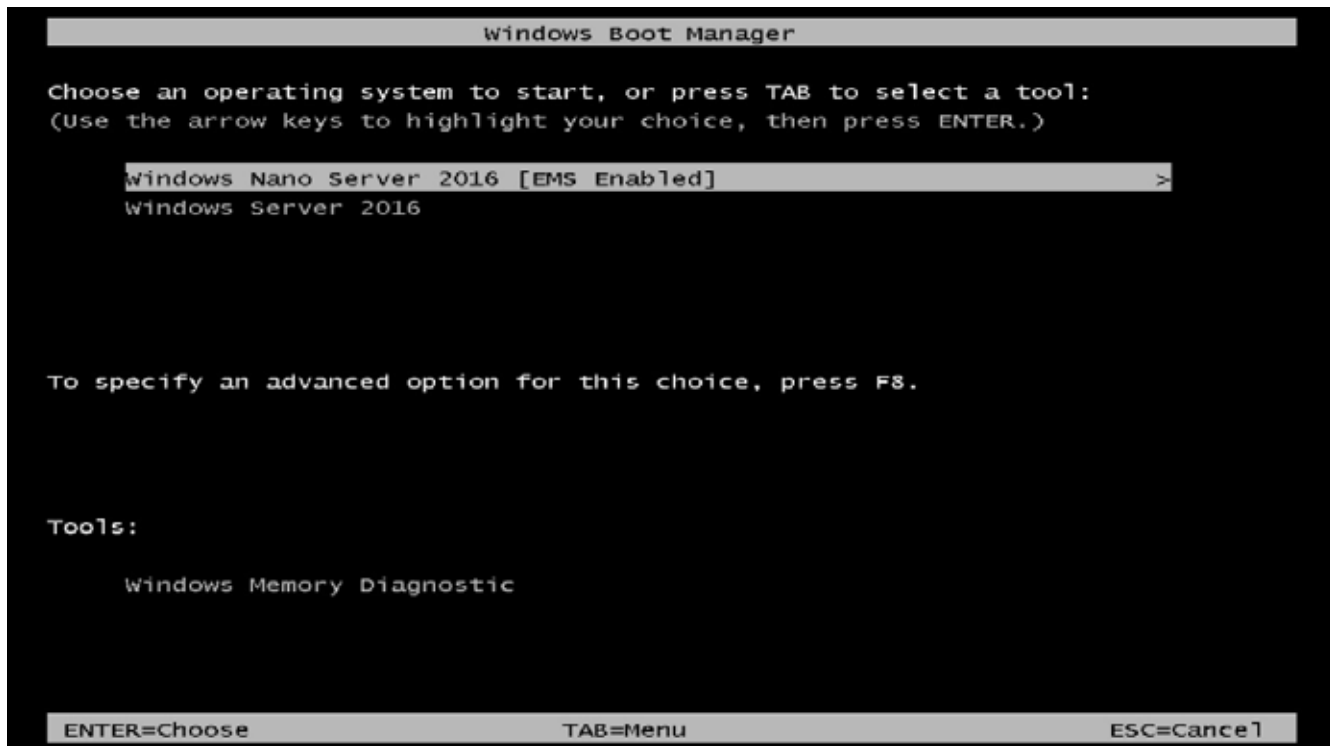
2. **Optional: Enable EMS in the new boot entry**

- We'll assume the identifier string of the newly added boot loader is {default}
- Run bcdedit.exe /set "{default}" bootems on
- Run bcdedit.exe /set "{default}" ems on
- Reboot the server

During POST, press the F11 key to stop at the boot selection screen

Be sure to select the “Windows Boot Manager” and press return

You should be presented with two boot entries. The Nano Server image should appear in the Windows Boot Manager as “Windows Nano Server 2016 [EMS Enabled]”.



**Figure 29:** Dual-Boot entries in the Windows Boot Manager.

3. Boot the Nano Server entry
4. The Nano Server Recovery Console should appear. Although the Nano Server Recovery Console provides for some basic configuration, other methods exist for managing Nano Server. Please refer to the Nano Server Remote Management section in the Appendix for more information.

You should now have a working Nano Server dual boot environment

**This concludes the instructions on how to configure and deploy Nano Server images on HPE ProLiant Servers.**

## Performing Windows Update on Nano Server

After every deployment and at regular intervals based on your IT policy, it is strongly recommended that you download and install all applicable Windows Updates. Here are the steps required to update Nano Server:

1. Open a remote PowerShell session to Nano Server and run the following commands:
  - a. `$sess = New-CimInstance -Namespace root/Microsoft/Windows/WindowsUpdate -ClassName MSFT_WUOperationsSession`
  - b. `$scanResults = Invoke-CimMethod -InputObject $sess -MethodName ApplyApplicableUpdates`
2. Reboot Nano Server:
  - a. Exit the PowerShell session
  - b. `Restart-Computer -Computer <computer name>`

## Appendix

### Nano Server Packages

Nano Server ships with a number of packages for customizing the image. The following is a listing of packages available at the time of this publication. For latest information, refer to the Microsoft Nano Server documentation found at [technet.microsoft.com/en-us/library/mt126167.aspx](https://technet.microsoft.com/en-us/library/mt126167.aspx).

**Table 10.** Nano Server packages

Package name	Description	Supported
Microsoft-NanoServer-Compute-Package.cab	Installs Hyper-V Role	Yes
Microsoft-NanoServer-Containers-Package.cab	Installs Containers feature	Yes
Microsoft-NanoServer-DCB-Package.cab	Install Data Center Bridging feature	Yes
Microsoft-NanoServer-Defender-Package.cab	Installs Windows Defender feature	Yes
Microsoft-NanoServer-DNS-Package.cab	Installs DNS package	Yes
Microsoft-NanoServer-DSC-Package.cab	Installs Desired State Configuration package	Yes
Microsoft-NanoServer-FailoverCluster-Package.cab	Installs Failover Cluster Role	Yes
Microsoft-NanoServer-Guest-Package.cab	Enables Nano Server to operate in a VM	Yes
Microsoft-NanoServer-Host-Package.cab	Enables Nano Server to operate on bare-metal	Yes
Microsoft-NanoServer-IIS-Package.cab	Installs IIS role	Yes
Microsoft-NanoServer-OEM-Drivers-Package.cab	Adds drivers such as NIC and Storage	Yes
Microsoft-NanoServer-SCVMM-Compute-Package.cab	Installs the SCVMM Hyper-V Compute package	Yes
Microsoft-NanoServer-SCVMM-Package.cab	Installs the SCVMM Hyper-V Management agent	Yes
Microsoft-NanoServer-SecureStartup-Package.cab	Adds Secure Startup feature	Yes
Microsoft-NanoServer-ShieldedVM-Package.cab	Adds Shielded VMs feature	Yes
Microsoft-NanoServer-SoftwareInventoryLogging-Package.cab	Adds Software Inventory Logging feature	Yes
Microsoft-NanoServer-Storage-Package.cab	Adds Storage Role	Yes

### Disk Configuration for WinPE Deployments

This section describes how to configure the disk for Nano Server deployments when booting from WinPE. It covers both UEFI and BIOS based deployments. This paper targets Disk 0 and wipes the disk clean. There are several ways of configuring the target disk. For more information on Windows disk partitioning, refer to the TechNet article at [technet.microsoft.com/en-us/library/dd799232\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd799232(v=ws.10).aspx). In the CMD shell, type diskpart.exe to enter the disk configuration utility and type the following commands:

1. UEFI based systems:
  - a. Select disk 0
  - b. Clean
  - c. Convert GPT
  - d. Create partition efi size=100
  - e. Format quick fs=fat32 label="System"
  - f. Assign letter="s"

- g. Create partition msr size=128
  - h. Create partition primary
  - i. Format quick fs=ntfs label="Windows"
  - j. Assign letter="w"
  - k. Detail disk
  - l. Exit
2. BIOS based systems:
- a. Select disk 0
  - b. Clean
  - c. Convert MBR (optional but recommended)
  - d. Create partition primary size=350
  - e. Format quick fs=ntfs label="System"
  - f. Assign letter="s"
  - g. Active
  - h. Create partition primary
  - i. Format quick fs=ntfs label="Windows"
  - j. Assign letter="w"
  - k. Detail disk
  - l. Exit

---

**Note**

Automating this step is possible by copying and pasting the lines from either section above into a text file and passing the file name to diskpart as an argument. For example, on UEFI systems copy the commands from the UEFI section into a file called uefi\_diskpart.txt. For BIOS based systems, copy the commands from the BIOS section into a file called bios\_diskpart.txt. Then invoke "diskpart.exe /s <diskpart.txt>", where <diskpart.txt> is the name of the desired file.

For example, diskpart.exe /s uefi\_diskpart.txt for a UEFI system.

---

**Adding boot-critical drivers to Windows boot images**

In some situations, it may be necessary to add out-of-box drivers for boot-critical devices because they are not in-box on the Windows Server 2016 media. One such example is the HPE Dynamic Smart Array B140i Controller. The driver for this controller is included in the HPE Service Pack for ProLiant v2016.10.0. This section explains the process for adding the B140i driver to the appropriate images supporting the deployment methods discussed in this paper.

**WinPE based deployments**

This section describes the process for injecting out-of-box drivers into the boot.wim file used in WinPE deployments. It is assumed the reader has already created a boot.wim file for WinPE based deployments as described above in the section, **Creating a custom WinPE boot image**. The B140i controller will be used as the example.

1. On the technician computer, open an ADK session to the folder to the location of the boot.wim file. In our example it is, c:\winpe\_amd64\media\sources.
2. Create a mount folder for the image, mkdir c:\winpe\_amd64\media\sources\mount
3. Mount the WinPE boot.wim file using dism.exe  
Dism.exe /mount-wim /wimfile:c:\winpe\_amd64\media\sources\boot.wim /index:1 /mountdir:c:\winpe\_amd64\media\sources\mount

4. Add the B140i driver to the image using the driver found in Nano Driver zip package located at, **c:\spp\win-driverpack-10.60\DynamicSAB140iCtrl**

```
Dism.exe /image:c:\winpe_amd64\media\sources\mount /add-driver /driver:c:\spp\win-driverpack-10.60\DynamicSAB140iCtrl\hpsa3.inf
```

Ensure there were no errors

5. Unmount the image and save the changes
6. Dism.exe /unmount-wim /mountdir:c:\winpe\_amd64\media\sources\mount /commit

This completes the process for adding the B140i driver to the WinPE boot.wim file.

### WDS based deployments

This section describes the process for injecting out-of-box drivers into the boot.wim file used for WDS deployments. The reader can decide to modify an existing image located on the WDS server, or add a new image that has already been customized. This paper will describe the process for customizing an existing WDS boot image. The B140i controller will be used as the example.

---

#### Note

The example below uses dism.exe to customize the image. Depending on the version of boot.wim you'll be modifying in your environment, it may be necessary to use the latest version of dism.exe found in the Windows 10 ADK.

---

1. Logon to the WDS server and open the WDS Management Console
2. Copy the B140i driver folder from the win-driverpack-10.60.zip package to a folder located on the WDS server, d:\drivers\DynamicSAB140iCtrl
3. Locate the desired boot.wim file from the list of boot images
4. Right-click the image and select the **Export Image** option
5. Export the image to a folder on the WDS server and give it a name such as d:\images\CustomBoot.wim
6. Open a ADK or CMD shell to the folder containing the exported image
7. Create a mount folder for the image, mkdir d:\images\mount

8. Mount Index #2 of the image using dism.exe

```
Dism.exe /mount-wim /wimfile:d:\images\CustomBoot.wim /index:2 /mountdir:d:\images\mount
```

9. Add the B140i driver to the image

```
Dism.exe /image:d:\images\mount /add-driver /driver:d:\drivers\DynamicSAB140iCtrl\hpsa3.inf
```

Ensure there were no errors

10. Unmount the image and save the changes

```
Dism.exe /unmount-wim /mountdir:d:\images\mount /commit
```

Add the image back to the WDS server. The reader has a couple of options for adding the new custom WIM back into the WDS server. The reader can replace an existing image, or create a new one. In this example, we'll create a new boot.wim file.

11. In the WDS Management Console, right-click Boot Images folder and select **Add Boot Image**
12. Follow the prompts to import the image from d:\images\CustomBoot.wim
13. A new boot image should exist on the WDS server

---

#### Note

If you would like to enable EMS for the newly added boot image, refer to the WDS section, "[Modifying the BCD of the newly added boot file](#)"

This completes the process for adding the B140i driver to the WDS boot.wim file

---



Nano Server Unattend.xml file for WIM Images

Table 12 contains a sample unattend.xml file to use in your environment. If you are deploying a Nano Server image using WDS, please refer to the section in the Nano Server Appendix titled, “Example unattend files for WDS deployments”.

Table 11. User-defined Unattend.xml file settings.

User-defined settings	Description
<ComputerName>	String containing the OS hostname
<AdministratorPassword>	String containing desired administrator password
<LocalAccounts>	Optional section. Use this section to add additional user accounts to the OS.
<LocalAccounts> <Password>	String containing password for the user account
<LocalAccounts> <DisplayName>	String containing the full user display name
<LocalAccounts> <Name>	String containing logon name
<LocalAccounts> <Group>	Name of local group in which to add the user account, i.e., Administrators
<TimeZone>	Local Timezone
<RegisteredOwner>	String containing the registered user name
<RegisteredOrganization>	String containing the registered organization

To modify this unattend file for your environment, simply change the fields containing “XXXXXXXX” with settings for your particular environment. Simply save a copy of this file to a local folder on the technician computer as Unattend.xml. For example, c:\ws2016\unattend.xml.

**Table 12.** This is an example unattend XML for non-WDS deployments. You will need to customize your unattend XML to fit your content.

---

```

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="offlineServicing">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
      instance">
      <ComputerName>XXXXXXX</ComputerName>
    </component>
  </settings>
  <settings pass="oobeSystem">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
      instance">
      <UserAccounts>
        <AdministratorPassword>
          <Value>XXXXXXX</Value>
          <PlainText>>false</PlainText>
        </AdministratorPassword>
        <LocalAccounts>
          <LocalAccount wcm:action="add">
            <Password>
              <Value>XXXXXXX</Value>
            <PlainText>>false</PlainText>
            </Password>
            <DisplayName>XXXXXXX</DisplayName>
            <Group>Administrators</Group>
            <Name>XXXXXXX</Name>
          </LocalAccount>
          <LocalAccount wcm:action="add">
            <Password>
              <Value>XXXXXXX</Value>
            <PlainText>>false</PlainText>
            </Password>
            <Description>XXXXXXX</Description>
            <DisplayName>XXXXXXX</DisplayName>
            <Group>Administrators</Group>
            <Name>XXXXXXX</Name>
          </LocalAccount>
        </LocalAccounts>
      </UserAccounts>
      <TimeZone>Pacific Standard Time</TimeZone>
    </component>
  </settings>
  <settings pass="specialize">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
      instance">
      <RegisteredOwner>XXXXXXX</RegisteredOwner>
      <RegisteredOrganization>XXXXXXX</RegisteredOrganization>
    </component>
  </settings>

```

---

---

```

    <component name="Networking-MPSSVC-Svc" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <FirewallGroups>
        <FirewallGroup wcm:action="add" wcm:keyValue="FileAndPrinterSharing">
            <Profile>all</Profile>
            <Group>File and Printer Sharing</Group>
            <Active>true</Active>
        </FirewallGroup>
        <FirewallGroup wcm:action="add" wcm:keyValue="WMI">
            <Active>true</Active>
            <Group>Windows Management Instrumentation [WMI]</Group>
            <Profile>all</Profile>
        </FirewallGroup>
    </FirewallGroups>
</component>
</settings>
<cpu:offlineImage
cpu:source="catalog:c:/ws2016/media/nanoserver/nanoserver_orig_coresystemserver_install.clg"
xmlns:cpu="urn:schemas-microsoft-com:cpu" />
</unattend>

```

---

## Nano Server Unattend XML files for WDS deployments

Below are sample unattend files that can be used for deploying Nano Server using Windows Deployment Services. Simply change the fields containing "XXXXXXX" with settings for your particular environment. The highlighted fields should be reviewed to determine if customizations are required.

If deploying on UEFI based systems, be sure to use the UEFI sample client file. If you are deploying to a BIOS system, be sure to use the BIOS sample client unattend file. The sample Image Unattend file is applied to the Nano Server image and therefore supports both BIOS and

UEFI deployments. Be sure to copy the files to the d:\RemoteInstall\WdsClientUnattend folder on your WDS server. For example, save the client unattend file with a descriptive name like, nanoserver\_client\_bios.xml or nanoserver\_client\_uefi.xml. The image file can be named nanoserver\_image.xml.

**Table 13.** Sample UEFI Client unattend.xml file for WDS deployments.

```

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
      language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <DiskConfiguration>
<Disk wcm:action="add">
  <CreatePartitions>
    <CreatePartition wcm:action="add">
      <Order>1</Order>
      <Size>100</Size>
      <Type>EFI</Type>
    </CreatePartition>
    <CreatePartition wcm:action="add">
      <Order>2</Order>
      <Size>128</Size>
      <Type>MSR</Type>
    </CreatePartition>
    <CreatePartition wcm:action="add">
      <Order>3</Order>
      <Type>Primary</Type>
      <Extend>true</Extend>
    </CreatePartition>
  </CreatePartitions>
  <ModifyPartitions>
    <ModifyPartition wcm:action="add">
      <Format>FAT32</Format>
      <Label>EFI System</Label>
      <Order>1</Order>
      <PartitionID>1</PartitionID>
    </ModifyPartition>
    <ModifyPartition wcm:action="add">
      <Order>2</Order>
      <PartitionID>2</PartitionID>
    </ModifyPartition>
    <ModifyPartition wcm:action="add">
      <Format>NTFS</Format>
      <Label>Windows</Label>
      <Order>3</Order>
      <PartitionID>3</PartitionID>
    </ModifyPartition>
  </ModifyPartitions>
  <DiskID>0</DiskID>
  <WillWipeDisk>true</WillWipeDisk>
    </Disk>
  </DiskConfiguration>
  <EnableFirewall>true</EnableFirewall>
<EnableNetwork>true</EnableNetwork>
  <WindowsDeploymentServices>
    <Login>
      <Credentials>
        <Domain>XXXXXXXX</Domain>
        <Password>XXXXXXXX</Password>
        <Username>XXXXXXXX</Username>
      </Credentials>
      <WillShowUI>Never</WillShowUI>
    </Login>
    <ImageSelection>
      <InstallImage>
        <ImageGroup>XXXXXXXX</ImageGroup>
        <ImageName>XXXXXXXX</ImageName>

```

---

```

        <Filename>XXXXXXX</Filename>
      </InstallImage>
    <InstallTo>
      <DiskID>0</DiskID>
      <PartitionID>3</PartitionID>
    </InstallTo>
    <WillShowUI>Never</WillShowUI>

  </ImageSelection>
</WindowsDeploymentServices>
<DynamicUpdate>
  <Enable>true</Enable>
  <WillShowUI>Never</WillShowUI>
</DynamicUpdate>
</component>
<component name="Microsoft-Windows-International-Core-WinPE" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <SetupUILanguage>
    <WillShowUI>Never</WillShowUI>
    <UILanguage>en-US</UILanguage>
  </SetupUILanguage>
  <InputLocale>en-US</InputLocale>
  <SystemLocale>en-US</SystemLocale>
  <UILanguage>en-US</UILanguage>
  <UILanguageFallback>en-US</UILanguageFallback>
  <UserLocale>en-US</UserLocale>
</component>
</settings>
<cpu:offlineImage cpu:source="catalog:c:/ws2016/media/NanoServer/NanoServer_CORESYSTEMSERVER_INSTALL.clg"

```

---

**Table 14.** Sample BIOS Client unattend.xml file for WDS deployments.

---

```

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings pass="windowsPE">
    <component name="Microsoft-Windows-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35"
language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <DiskConfiguration>
        <Disk wcm:action="add">
          <CreatePartitions>
            <CreatePartition wcm:action="add">
              <Size>350</Size>
              <Type>Primary</Type>
              <Order>1</Order>
            </CreatePartition>
            <CreatePartition wcm:action="add">
              <Extend>true</Extend>
              <Type>Primary</Type>
              <Order>2</Order>
            </CreatePartition>
          </CreatePartitions>
          <ModifyPartitions>
            <ModifyPartition wcm:action="add">
              <Active>true</Active>
              <Format>NTFS</Format>
              <Label>Boot</Label>
              <Order>1</Order>
              <PartitionID>1</PartitionID>
            </ModifyPartition>
            <ModifyPartition wcm:action="add">
              <Format>NTFS</Format>

```

---

---

```

<Label>System</Label>
                                <PartitionID>2</PartitionID>
<Order>2</Order>
                                </ModifyPartition>
                                </ModifyPartitions>
                                <DiskID>0</DiskID>
                                <WillWipeDisk>true</WillWipeDisk>
                                </Disk>
</DiskConfiguration>
<EnableFirewall>true</EnableFirewall>
<EnableNetwork>true</EnableNetwork>
<WindowsDeploymentServices>
  <Login>
    <Credentials>
      <Domain>XXXXXXX</Domain>
      <Password>XXXXXXX</Password>
      <Username>XXXXXXX</Username>
    </Credentials>
    <WillShowUI>Never</WillShowUI>
  </Login>
  <ImageSelection>
    <InstallImage>
      <ImageGroup>XXXXXXX</ImageGroup>
      <ImageName>XXXXXXX</ImageName>
      <Filename>XXXXXXX</Filename>
    </InstallImage>
    <InstallTo>
      <DiskID>0</DiskID>
      <PartitionID>2</PartitionID>
    </InstallTo>
    <WillShowUI>Never</WillShowUI>
  </ImageSelection>
</WindowsDeploymentServices>
  <DynamicUpdate>
<Enable>true</Enable>
    <WillShowUI>Never</WillShowUI>
  </DynamicUpdate>
</component>
  <component name="Microsoft-Windows-International-Core-WinPE" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <SetupUILanguage>
      <WillShowUI>Never</WillShowUI>
      <UILanguage>en-US</UILanguage>
    </SetupUILanguage>
    <InputLocale>en-US</InputLocale>
    <SystemLocale>en-US</SystemLocale>
    <UILanguage>en-US</UILanguage>
    <UILanguageFallback>en-US</UILanguageFallback>
    <UserLocale>en-US</UserLocale>
  </component>
</settings>
<cpu:offlineImage
  cpi:source="catalog:c:/ws2016/media/NanoServer/NanoServer_CORESYSTEMSERVER_INSTALL.clg"
  xmlns:cpi="urn:schemas-microsoft-com:cpi" />
</unattend>

```

---

**Table 15.** Sample image unattend.xml file for WDS deployments.

```

<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <settings
    pass="oobeSystem">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance">
      <UserAccounts>
        <AdministratorPassword>
          <Value>XXXXXXXX</Value>
          <PlainText>>false</PlainText>
        </AdministratorPassword>
        <LocalAccounts>
          <LocalAccount wcm:action="add">
            <Password>
              <Value>XXXXXXXX</Value>
              <PlainText>>false</PlainText>
            </Password>
            <DisplayName>XXXXXXXX</DisplayName>
            <Group>Administrators</Group>
            <Name>XXXXXXXX</Name>
          </LocalAccount>
          <LocalAccount wcm:action="add">
            <Password>
              <Value>XXXXXXXX</Value>
              <PlainText>>false</PlainText>
            </Password>
            <Description>XXXXXXXX</Description>
            <DisplayName>XXXXXXXX</DisplayName>
            <Group>Administrators</Group>
            <Name>XXXXXXXX</Name>
          </LocalAccount>
        </LocalAccounts>
      </UserAccounts>
      <TimeZone>Pacific Standard Time</TimeZone>
      <RegisteredOwner>XXXXXXXX</RegisteredOwner>
      <RegisteredOrganization>XXXXXXXX</RegisteredOrganization>
    </component>
  </settings>
  <settings pass="specialize">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance">
      <ComputerName>XXXXXXXX</ComputerName>
    </component>
    <component name="Networking-MPSSVC-Svc" processorArchitecture="amd64"
      publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS"
      xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
        instance">
      <FirewallGroups>
        <FirewallGroup wcm:action="add" wcm:keyValue="FileAndPrinterSharing">
          <Profile>all</Profile>
          <Group>File and Printer Sharing</Group>
          <Active>true</Active>
        </FirewallGroup>
        <FirewallGroup wcm:action="add" wcm:keyValue="WMI">
          <Active>true</Active>
          <Group>Windows Management Instrumentation [WMI]</Group>

```



---

```
        <Profile>all</Profile>
      </FirewallGroup>
    </FirewallGroups>
  </component>
</settings>
<cpu:offlineImage
cpu:source="catalog:c:/ws2016/media/NanoServer/NanoServer_CORESYSTEMSERVER_INSTALL.clg"
xmlns:cpu="urn:schemas-microsoft-com:cpu" />
</unattend>
```

---

## Obtaining the MAC address through iLO and System ROM

---

### Note

This assumes that we are using the onboard NIC

---

Follow the steps below to obtain the MAC address of the onboard NIC through the iLO Web interface:

1. Connect to the **iLO IP address** using a Web browser
2. Expand the Information link and click on **System Information**
3. Click on the **Network** tab.
4. Refer to the table for the appropriate NIC and make a note of the MAC address of the target NIC port.
5. Follow the steps below to obtain the MAC address of the onboard NIC through the System Utilities (RBSU)
6. Power on the server and press **F9** to enter System Utilities
7. For Gen8 BIOS systems:
  - a. Press **TAB** key and make note of the MAC address of the appropriate NIC
8. For Gen9 systems
  - a. With System Configuration highlighted press **enter** key
  - b. Highlight appropriate NIC and press **enter** key
  - c. Make a note of the MAC address of the appropriate NIC
9. Select **ESC** to return back to the main menu or exit the System Utilities

### Enabling network boot on an HPE ProLiant server

1. Power on the server and press **F9** to enter System Utilities.
2. For Gen8 BIOS systems:
  - a. With System Options highlighted press **enter** key
  - b. Highlight Embedded NICs and press **enter** key
  - c. Highlight the appropriate NIC and press **enter** key
  - d. With Network Boot highlighted press **enter** key
3. For Gen9 systems:
  - a. Select System **Configuration -> BIOS/Platform Configuration (RBSU) -> Network Options -> Network Boot** Options
  - b. Select the appropriate NIC, press **enter** and select the **Network Boot** option from the list
  - c. Press **F10** to save.

4. Press the **ESC** key several times until you're back at the System Utilities main menu
5. From the list of options scroll down and select **Reboot the System**

## Nano Server Remote Management

Nano Server is meant to be managed remotely. There are many options available. This document briefly covers two of them, PowerShell Remoting and EMS. The purpose of this section is to provide a brief overview and is not meant to be an instructional document on configuring and troubleshooting Windows remote management. For additional management options, refer to the following Microsoft articles:

- Nano Server: [aka.ms/nanoserver](http://aka.ms/nanoserver)
- Windows Remote Management: [msdn.microsoft.com/en-us/library/aa384426\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa384426(v=vs.85).aspx)

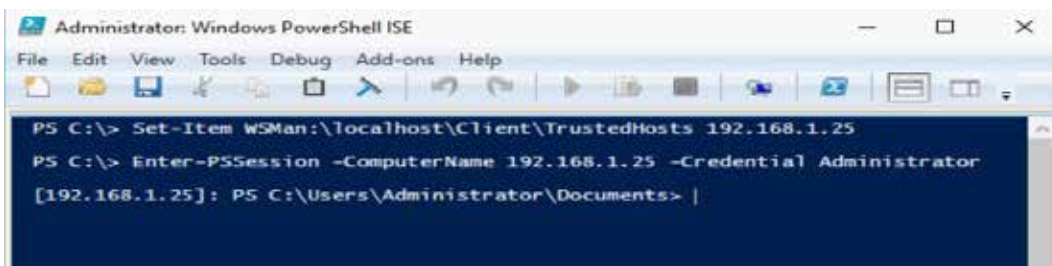
### PowerShell Remoting

The following components are required for PowerShell remoting:

- Remote management station or technician computer
- IP address of the Nano Server computer
  - For this example, we'll assume the IP address is **192.168.1.25**
- Administrative account and password of the Nano Server computer
  - For this example, we'll use the **Administrator** account

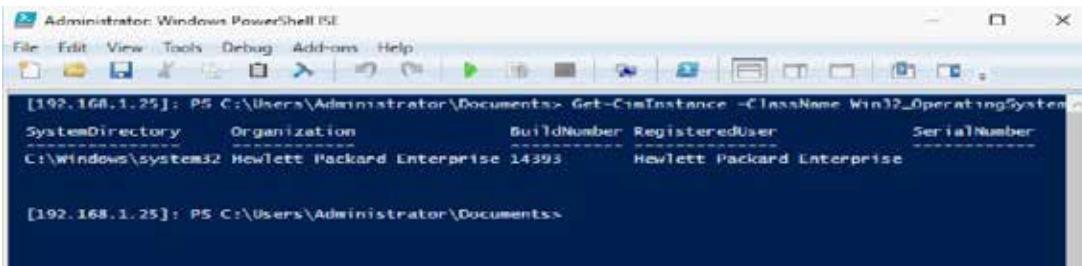
To enable PowerShell remoting, follow these steps on the management station:

- Add the Nano Server's IP address to the management station's trusted hosts list:
  - Open an elevated PowerShell session
  - Type the following command, `Set-Item WSMan:\localhost\Client\TrustedHosts 192.168.1.25`
  - Click **Yes** if warning message appears
- Open a remote PowerShell session to the Nano Server:
  - **Enter-PSSession -ComputerName 192.168.1.25 -Credential Administrator**
  - A credentials dialog box should appear. Enter the password for the Administrator account
  - If successful, you should see something similar to the following:



**Figure 30:** Configuring WSMan and PowerShell session to Nano Server computer

- Notice that the Nano Server's IP address is listed at the beginning of the prompt indicating this is a remote session to that IP address.
- Let's obtain the OS version using the PowerShell remote session and WMI using CIM
  - `Get-CimInstance -ClassName Win32_OperatingSystem`
  - Output should be similar to the following:



**Figure 31:** Obtain operating system information of remote host using WMI

### Performing Windows Update

To detect, download and install all applicable Windows Updates, execute the following commands:

```

$ssess = New-CimInstance -Namespace root/Microsoft/Windows/WindowsUpdate -ClassName
MSFT_WUOperationsSession

$scanResults = Invoke-CimMethod -InputObject $ssess -MethodName ApplyApplicableUpdates

```

Once these commands have finished executing you will need to reboot Nano Server.

This example is just a brief introduction to Nano Server Management using PowerShell remoting.

### Emergency Management Services (Windows EMS)

The second method of remotely managing Nano Server discussed in this paper is Windows Emergency Management Services (EMS). Emergency Management Services (EMS) is a feature that provides remote management and system recovery options when other server administrative options are not possible. It is also required for headless systems in which there is no GUI available, as in the case of Nano Server.

It accomplishes this through the System Administration Channel (SAC) feature of Windows Server. The SAC channel provides a number of administrative features such as enabling a Windows cmd.exe channel for Windows CLI access, list processes currently running on the system, obtain IP address information, retrieve server hardware information, and reboot the server, just to name a few. Another important feature of the SAC channel is the ability to monitor the boot and install progress of the server. For more information on EMS and SAC see the TechNet article at [technet.microsoft.com/en-us/library/cc778042\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc778042(v=ws.10).aspx).

Using Windows EMS to manage the HPE ProLiant server involves the following:

- Enabling the Virtual Serial Port (VSP) in RBSU on the ProLiant server
- Enabling the EMS Port in RBSU on the ProLiant server
- Enabling Windows EMS in the bootloader of the Operating System
- Using PuTTY or similar terminal emulation program to establish an SSH connection to the ProLiant iLO IP address
- Using Windows EMS functionality to perform basic administrative tasks

### Enabling the Virtual Serial Port on HPE ProLiant Servers

HPE Integrated Lights-Out 3/4 (iLO 3 and 4) allows administrators to manage their server remotely using a variety of connection methods including a Web browser, telnet, SSH, and PowerShell. This section focuses on enabling the VSP for use with SSH. For more information on the iLO VSP, refer to the following document, [h20566.www2.hp.com/hpsc/doc/public/display?sp4ts.oid=4154735&docLocale=en\\_US&docId=emr\\_na-c00263709](http://h20566.www2.hp.com/hpsc/doc/public/display?sp4ts.oid=4154735&docLocale=en_US&docId=emr_na-c00263709).

Begin by connecting to the iLO IRC using a Web browser (<http://<IP address of the iLO>>) and booting the server to the System Utilities Screen

**For Gen8 (BIOS) systems:**

- Locate the Virtual Serial Port option using the following path, **System Options -> Serial Port Options -> Virtual Serial Port**
- Press the **Enter** key to select from a list of options. For the purposes of this paper, we'll select **COM 2**.
- A reboot is required for the change to take effect. Press **ESC** three times to get back to Setup Utility and then **F10** to save and reboot the system.

**For Gen9 systems:**

- Locate the Virtual Serial Port option using the following path, **System Configuration -> BIOS/Platform Configuration (RBSU) -> System Options -> Serial Port Options -> Virtual Serial Port**
- Press the **Enter** key to select from a list of options. For the purposes of this paper, we'll select **COM 2**
- Press **F10** to confirm and save the change
- A reboot is required for the change to take effect. Press the **ESC** key until you are back to the System Utilities screen
- Scroll down and select "**Reboot the System**"

The HPE ProLiant VSP is now be enabled. The next step is to enable the EMS console on the server.

**Enabling the EMS console on HPE ProLiant Servers**

The HPE ProLiant EMS Console option configures the ACPI serial port for redirecting output to the Windows EMS console.

Begin by connecting to the iLO IRC using a Web browser (<http://<IP address of the iLO>>) and booting the server to the System Utilities Screen

**For Gen8 (BIOS) systems:**

- Locate the Virtual Serial Port option using the following path, **BIOS Serial Console and EMS -> EMS Console**
- Press the **Enter** key to select from a list of options. Be sure to select the COM port that matches the VSP COM port configured above. For the purposes of this paper, we'll select **COM 2**.
- A reboot is required for the change to take effect. Press the **ESC** key twice and **F10** to save and reboot the system.

**For Gen9 systems:**

- Locate the Virtual Serial Port option using the following path, **System Configuration -> BIOS/Platform Configuration (RBSU) -> BIOS Serial Console and EMS -> EMS Console**
- Press the **Enter** key to select from a list of options. Be sure to select the COM port that matches the VSP COM port configured above. For the purposes of this paper, we'll select **COM 2**
- Press **F10** to confirm and save the change
- A reboot is required for the change to take effect. Press the **ESC** key until you are back to the System Utilities screen. – Scroll down and select "**Reboot the System**"

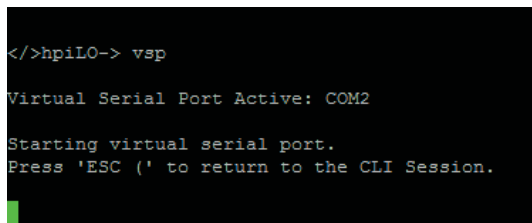
**Using the HPE ProLiant VSP and Windows EMS to manage Nano Server**

This section describes how to use the Windows EMS and SAC channel to perform basic administration of Nano Server. The following prerequisites must be met to continue:

- The OS boot loader entry associated with Nano Server is configured for EMS
- The HPE ProLiant server VSP is enabled
- The HPE ProLiant server EMS console is enabled
- A terminal emulator program such as PuTTY.exe is installed on the management station. PuTTY can be downloaded from [putty.org](http://putty.org).

**Here are the steps for connecting to the HPE ProLiant VSP:**

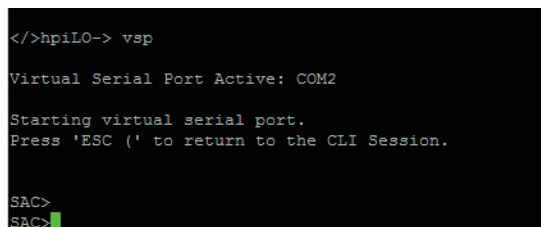
1. Open a **PuTTY** session using `putty.exe`
2. In the Host Name box in the Session category, enter the IP address or FQDN of the iLO
3. Choose **SSH** as the **Connection type** and ensure the Port is set to 22.
4. Optional: Add a friendly name in the **Saved Sessions** box and click the **Save button** to save for future use
5. Click the **Open button** to establish a connection to the iLO
6. If this is the first time a connection has been made to the iLO VSP from this computer, you may receive a **PuTTY Security Alert** pop-up. This is normal. Go ahead and click the **Yes** button to continue.
7. At the login prompt, enter an **iLO user name** and **password**
8. After a successful login you should see the following prompt, `</>hpiLO->`
9. Type **VSP** and press **return** to open a VSP session
10. If successful, you should now have an active VSP session similar to the following:



```
</>hpiLO-> vsp
Virtual Serial Port Active: COM2
Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.
```

**Figure 32:** VSP session of HPE ProLiant Server

11. If not, double-check the VSP configuration following the steps provided elsewhere in this document
12. Once you have established a VSP session, your cursor may appear to be frozen and you are unable to interact with the server. This can be normal depending on the state of the server. With the connection still active, reboot the server.
  - a. Important note: To disconnect the session, following the instructions in the VSP session as shown in the screen shot above. Type the key combination, Shift-ESC-9 (The number 9 on the main keyboard, not on the keypad) to exit the session and revert back to the `</>hpiLO->` prompt. Type `exit` to completely exit the SSH connection.
  - b. After reboot, you should start seeing output to the VSP interface, specifically system POST information.
13. If you are able to see the system POST information displayed to the VSP session, the VSP is configured correctly.
14. If the VSP is working up until Nano Server boots and then ceases, it means that EMS is not enabled in the BCD store of the operating system. If this is the case, then one or more configuration steps failed or were skipped. You will need to retrace the steps to figure out exactly where things went wrong.
15. If VSP and Windows EMS is configured correctly, you should see output in the VSP session similar to:



```
</>hpiLO-> vsp
Virtual Serial Port Active: COM2
Starting virtual serial port.
Press 'ESC (' to return to the CLI Session.
SAC>
SAC>
```

**Figure 33:** VSP session of HPE ProLiant Server connected to Windows EMS

16. Type help in the window to obtain a list of useful commands. Most noteworthy are:
  - a. 'i'—to obtain IP address information
  - b. 'id'—to list OS details
  - c. 't'—to obtain a list of running processes
  - d. 'restart'—to reboot the OS
  - e. 'shutdown'—to initiate a shutdown of the OS
  - f. 'cmd'—to initiate a text based logon to the OS
17. You can logon to the OS through the VSP channel using the credentials specified in the image unattend file that used during Nano Server Deployment. To do so, follow these steps:
  - a. Type "cmd" followed by the **Enter** key at the SAC prompt. You should see a message indicating a new channel has been created and the channel name. Most likely, the channel name is Cmd0001.
  - b. Type "ch" followed by **Enter** to see a list of available channels.
  - c. Find the channel number associated with the CMD channel created above and type "ch -si 1" followed by the **Enter** key twice, where one is the channel number associated with Cmd0001.
  - d. You should be prompted with a login screen. Enter the Username, password, and domain (if the server is domain joined) of an administrative account on Nano Server.
  - e. You should now be a C:\Windows\system32 prompt.
18. Through the local Windows VSP session, several commands are available for interacting with the OS. For example:
  - a. 'ipconfig.exe' – To display IP configuration.
  - b. 'PowerShell -Get-CimInstance -Classname Win32\_OperatingSystem' – To obtain OS information using PowerShell.  
The VSP and EMS is great for monitoring the OS installation, obtaining the IP address, and performing other basic administrative tasks.

## Installing the HPE ProLiant Agentless Management Service in Nano Server

1. Download the HPE Service Pack for ProLiant (SPP) Version 2016.10.0's zip file 'Win-driverpack-10.60.zip' under 'WIN\_DRV' folder.
2. Extract the zip file to a temporary folder and go to 'appx' folder.
3. Copy these 3 files to a local temporary folder: ams.cer, ams.ps1 and ams.appx.

### Installation of AMS:

1. Create a new remote PowerShell session to Nano Server with Administrator account and assign it to an object variable. Example: \$s = new-ssession -computername 192.168.0.11 -credential "~\Administrator".
2. Enter the new remote PowerShell session. Example: enter-ssession \$s
3. Exit the current PowerShell session using "exit-ssession" command.
4. From the local temporary folder, copy the following files to the newly created folder in Nano Server (e.g. c:\Packages) folder using "copy-item" command: ams.cer, ams.appx and ams.ps1. Example: copy-item -path c:\temp\\*\* -destination c:\Packages -ToSession \$s
5. Re-enter the active remote PowerShell session to the Nano Server using "enter-ssession" command. Example: enter-ssession \$s
6. To install AMS, go to the folder where the extracted files were copied (e.g. c:\Packages) and execute: .\ams.ps1 -action install

### Uninstallation of AMS:

1. Follow the same procedure as "Installation of AMS" except change the last step's command from ".\ams.ps1 -action install" to ".\ams.ps1 -action uninstall"

## Implementing Microsoft Container Technology with Windows Server 2016

### Manage containers with Docker

Windows Containers can only be managed with Docker.

Windows Containers are one of the new features of Windows Server 2016. They provide OS level isolation by creating distinct instances of runtime environments that allow multiple isolated applications to be run on a single system. There are two types of containers with a different degree of application isolation; Server Containers and Hyper-V Containers. Windows Server Containers achieve isolation through namespace and process isolation. Hyper-V Containers encapsulates each container in a light weight virtual machine.

### HW and SW requirements

#### Hardware Specifications

Platform	ProLiant DL180 Gen9
CPU	Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz
Memory	8 GB
Storage	1TB
ROM	U20 v2.30 (09/12/2016)
iLO	2.50

#### Software Specifications

OS	Windows Server 2016
----	---------------------

### Windows Server Container

#### Configure System

##### Configure System Environment Variable for Setting Proxy

Configure proxy settings properly so commands such as “docker pull” would work; for example, behind a corporate proxy.

To configure system environment variable, either press “Windows key” and “R” together, and type “sysdm.cpl” or go to Control Panel > System and Security > System and click on “Advanced system settings” link.

Either way would open a window titled “System Properties” and in the “Advanced” tab, click on “Environment Variables”.

In “System Variables”, click “NEW” to add “HTTP\_PROXY” and “HTTPS\_PROXY” and set them to appropriate proxy settings for the deployment environment.

The settings above would need system restart to take effect.

#### Installing Docker and Base Container Image for Windows Server Container

##### Install Container Feature

The container feature needs to be enabled before working with Windows containers. To do so run the following command in an elevated PowerShell session.

```
PS C:\> Enable-WindowsOptionalFeature -Online -FeatureName containers -All
```

When the installation has completed, reboot the computer.

```
PS C:\> Restart-Computer -Force
```

Verify that Containers feature by the following command.

```
PS C:\> Get-WindowsFeature -Name Containers
```

```
PS C:\ProgramData\docker\windowsfilter> Get-WindowsFeature -Name containers
```

Display Name	Name	Install State
[X] Containers	Containers	Installed

### Install Docker

Docker is required in order to work with Windows containers. Docker consists of the Docker Engine, and the Docker client. For this exercise, both will be installed. Run the following commands to do so.

Download the Docker engine and client as a zip archive.

```
PS C:\> Invoke-WebRequest "https://get.docker.com/builds/Windows/x86_64/docker-1.12.0.zip" -OutFile "$env:TEMP\docker-1.12.0.zip" -UseBasicParsing
```

Expand the zip archive into Program Files, the archive contents is already in docker directory.

```
PS C:\> Expand-Archive -Path "$env:TEMP\docker-1.12.0.zip" -DestinationPath $env:ProgramFiles
```

Add the Docker directory to the system path.

```
PS C:\> [Environment]::SetEnvironmentVariable("Path", $env:Path + ";$env:ProgramFiles\docker\", [EnvironmentVariableTarget]::Machine)
```

Restart the PowerShell session so that the modified path is recognized.

Once docker is installed and PowerShell has been restarted, "docker version" can be used to display the version.

```
PS C:\> docker version
```

To install Docker as a Windows service, run the following.

```
PS C:\> & $env:ProgramFiles\docker\docker.exe --register-service
```

Once installed, the service can be started.

```
PS C:\> Start-Service Docker
```

Once docker service is started, running "Get-Process \*docker\*" which gets the processes that are running will show dockerd as below.

```
PS C:\> Start-Service Docker
WARNING: waiting for service 'Docker Engine (Docker)' to start...
PS C:\> Get-Process *docker*
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
201	12	11072	22780	0.20	4528	0	dockerd

### Install Base Container Images

Windows containers are deployed from templates or images. Before a container can be deployed, a container base OS image needs to be downloaded. The following commands will download the Windows Server Core base image.

```
PS C:\> docker pull microsoft/windowsservercore
```

```
PS C:\> docker pull microsoft/iis
```

When the above commands are done, display similar to the following should show.



```
PS C:\ > docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
microsoft/iis	latest	6e30590a2139	6 days ago	7.575 GB
microsoft/windowsservercore	latest	f20579da8e4e	2 weeks ago	7.329 GB

Once the image is pulled, running docker images will return a list of installed images, in this case the Nano Server image.

Deploy Your First Container

```
PS C:\> docker run -d -p 80:80 microsoft/iis ping -t localhost
```

Use docker ps command to see if a new container is running, and something similar as below should be displayed.

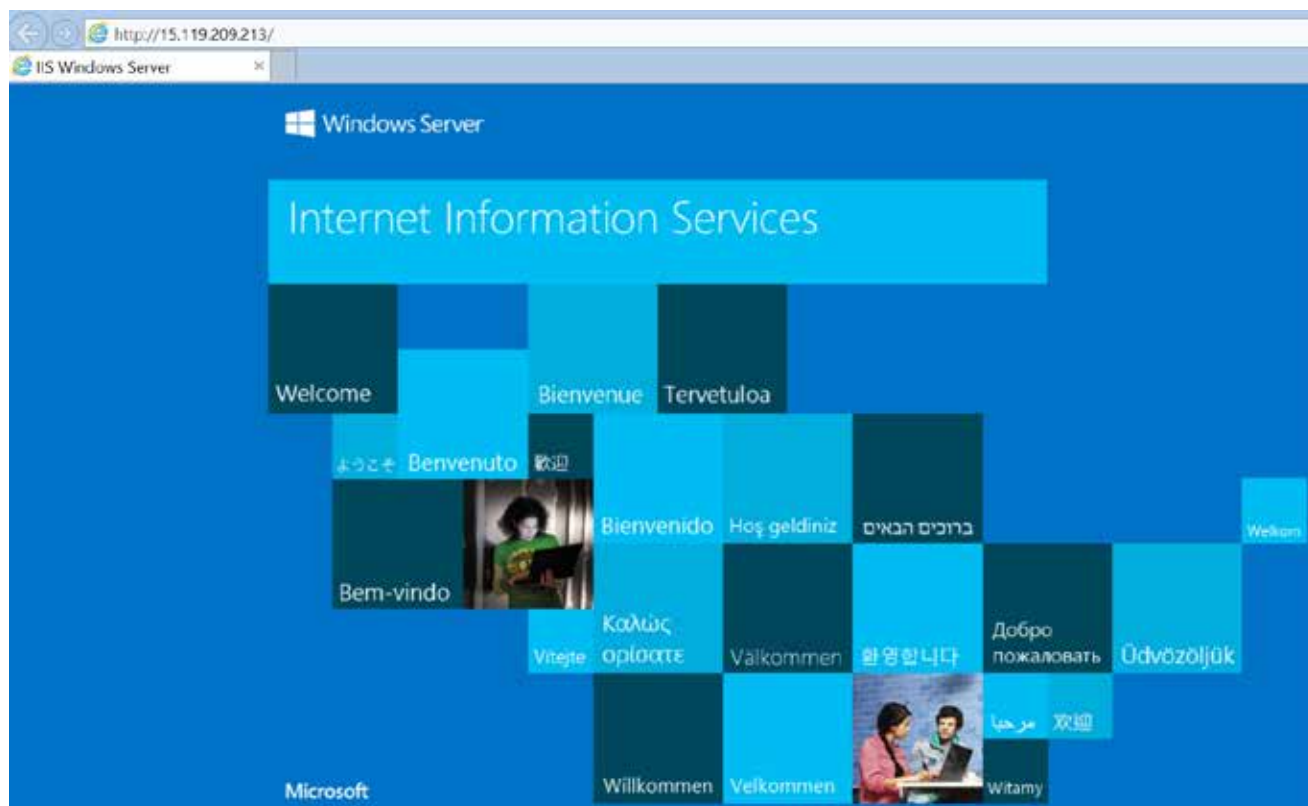
```
PS C:\> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
048abf15c072	microsoft/iis	"C:\\ServiceMonitor.exe"	22 hours ago

STATUS	PORTS	NAMES
Up 22hours	0.0.0.0:80->80/tcp	admiring_panini

From a different computer, open up a web browser and enter the IP address of the container host, a IIS splash screen should be shown which is being served from the IIS instance hosted in the Windows container.



Back on the container host, use the “docker rm” command to remove the container. Note to replace 048abf15c072 to the actual container ID as shown in the previous “docker ps” command.

```
PS C:\> docker rm -f 048abf15c072
```

## Hyper-V Container

In order to run Hyper-V containers, the Hyper-V role is required.

### Hyper-V Container Host

If the Windows container host is itself a Hyper-V virtual machine, nested virtualization will need to be enabled **after** installing the Hyper-V role plus creating a virtual machine that would be served as the container host.

#### Create a Virtual Machine to be used as the Container Host

Install Hyper-V role in Server Manager > Add Roles and Features Wizard and select “Hyper-V” in “Select Server Roles”. This requires system to restart after installation.

Then create a Hyper-V virtual machine in Hyper-V Manager > Action (Menu Var) > New > Virtual Machine. Take note of the name you specified as it would be used in nested virtualization script.

Click next to proceed to Installation Options and install Windows Server 2016 on the virtual machine.

During first boot of the new virtual machine, installation would be started.

After installation completes, follow the steps in Script for nested virtualization on Host OS of Hyper-V Host instead of the OS in virtual machine.

#### Script for nested virtualization

The following script will configure nested virtualization for the container host. This script is run on the parent Hyper-V machine. Ensure that the container host virtual machine is turned off when running this script.

#### Replace with the virtual machine name

```
PS C:\> $vm = "<virtual-machine>"
```

#### Configure virtual processor

```
PS C:\> Set-VMProcessor -VMName $vm -ExposeVirtualizationExtensions $true -Count 2
```

#### Disable dynamic memory

```
PS C:\> Set-VMemory $vm -DynamicMemoryEnabled $false
```

#### Enable mac spoofing

```
PS C:\> Get-VMNetworkAdapter -VMName $vm | Set-VMNetworkAdapter -MacAddressSpoofing On
```

#### Configure system

##### Configure System Environment Variable for Setting Proxy

Configure proxy settings properly so commands such as “docker pull” would work; for example, behind a corporate proxy.

To configure system environment variable, either press “Windows key” and “R” together, and type “sysdm.cpl” or go to Control Panel > System and Security > System and click on “Advanced system settings” link.

Either way would open a window titled “System Properties” and in the “Advanced” tab, click on “Environment Variables”.

In “System Variables”, click “NEW” to add “HTTP\_PROXY” and “HTTPS\_PROXY” and set them to appropriate proxy settings for the deployment environment.

The settings above would need system restart to take effect.

#### Installing Docker and Base Container Image for Hyper-V container

##### Install Container Feature

The container feature needs to be enabled before working with Windows containers. To do so run the following command in an elevated PowerShell session.

```
PS C:\> Enable-WindowsOptionalFeature -Online -FeatureName containers -All
```

Enable the Hyper-V feature using PowerShell. Run the following command in an elevated PowerShell session.

```
PS C:\> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All
```

When the installation has completed, reboot the computer.

```
PS C:\> Restart-Computer -Force
```

Verify that Containers feature by the following command.

```
PS C:\> Get-WindowsFeature -Name Containers
```

```
PS C:\ProgramData\docker\windowsfilter> Get-WindowsFeature -Name containers
```

Display Name	Name	Install State
[X] Containers	Containers	Installed

## Install Docker

Docker is required in order to work with Windows containers. Docker consists of the Docker Engine, and the Docker client. For this exercise, both will be installed. Run the following commands to do so.

Download the Docker engine and client as a zip archive.

```
PS C:\> Invoke-WebRequest "https://get.docker.com/builds/Windows/x86_64/docker-1.12.0.zip" -OutFile "$env:TEMP\docker-1.12.0.zip" -UseBasicParsing
```

Expand the zip archive into Program Files, the archive contents is already in docker directory.

```
PS C:\> Expand-Archive -Path "$env:TEMP\docker-1.12.0.zip" -DestinationPath $env:ProgramFiles
```

Add the Docker directory to the system path.

```
PS C:\> [Environment]::SetEnvironmentVariable("Path", $env:Path + ";$env:ProgramFiles\docker\", [EnvironmentVariableTarget]::Machine)
```

Restart the PowerShell session so that the modified path is recognized.

Once docker is installed and PowerShell has been restarted, "docker version" can be used to display the version.

```
PS C:\> docker version
```

To install Docker as a Windows service, run the following.

```
PS C:\> & $env:ProgramFiles\docker\dockerd.exe --register-service
```

Once installed, the service can be started.

```
PS C:\> Start-Service Docker
```

Once docker service is started, running "Get-Process \*docker\*" which gets the processes that are running will show "dockerd" as below.

```
PS C:\> Start-Service Docker
WARNING: Waiting for service 'Docker Engine (Docker)' to start...
PS C:\> Get-Process *docker*
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
201	12	11072	22780	0.20	4528	0	dockerd

## Install Base Container Images

Windows containers are deployed from templates or images. Before a container can be deployed, a container base OS image needs to be downloaded. The following commands will download the Nano Server base image.

Pull the Nano Server base image. It takes around 5 minutes depending on the environment.

```
"docker pull microsoft/nanoserver"
```

```
PS C:\Users\Administrator> docker pull microsoft/nanoserver
Using default tag: latest
latest: Pulling from microsoft/nanoserver
cf62dbf6d334: Downloading [=====] 147.6 MB/358.4 MB
```

When it is done, display similar to the following should show.

```
PS C:\> docker pull microsoft/nanoserver
Using default tag: latest
latest: Pulling from microsoft/nanoserver
cf62dbf6d334: Pull complete
Digest: sha256:e161c43c9695a20d0b7271e7339bb041026db548667d2d9ecc04e8dc6fba9bed
Status: Downloaded newer image for microsoft/nanoserver:latest
```

Once the image is pulled, running “docker images” will return a list of installed images, in this case the Nano Server image.

```
PS C:\> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
microsoft/nanoserver	latest	3a703c6e97a2	10 weeks ago	969.8 MB

### Deploy Your First Container

First, start a container with an interactive session from the `nanoserver` image. Note that when docker creates Hyper-V containers, the `--isolation=hyperv` parameter is used.

```
PS C:\> docker run -it --isolation=hyperv microsoft/nanoserver cmd
```

Once the container has started, you will be presented with a command shell from within the container.

```
Microsoft Windows [Version 10.0.14300]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\>_
```

Run “docker ps” to know which container is running.

```
PS C:\Users\Administrator> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
a82df48c0e1b	microsoft/nanoserver	"cmd"	5 minutes ago	Up 4 minutes	

And you will see a new process “docker” presented in “get-process”.

```
PS C:\Users\Administrator> get-process docker
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
143	10	8576	15204	0.05	2008	1	docker

If two containers are running, then two “docker” processes would be presented.

```
PS C:\Users\Administrator> get-process docker
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
134	9	7420	14872	0.03	868	1	docker
143	10	8576	15280	0.05	2008	1	docker

Run “dir” inside container running NANO with cmd.

```

C:\>dir
Volume in drive C has no label.
Volume Serial Number is 545B-0BF0

Directory of C:\

08/26/2016  02:43 PM    <DIR>          Program Files
08/26/2016  02:43 PM    <DIR>          Program Files (x86)
08/26/2016  02:43 PM    <DIR>          Users
08/26/2016  05:19 PM    <DIR>          Windows
               0 File(s)                0 bytes
               4 Dir(s)  21,209,468,928 bytes free

```

This is dir C:\ on the container host

```

C:\>dir
Volume in drive C has no label.
Volume Serial Number is C663-80DC

Directory of C:\

07/16/2016  06:23 AM    <DIR>          PerfLogs
08/26/2016  11:46 AM    <DIR>          Program Files
07/16/2016  06:23 AM    <DIR>          Program Files (x86)
08/26/2016  05:20 PM             0 test.txt
08/26/2016  11:22 AM    <DIR>          Users
08/26/2016  11:21 AM    <DIR>          Windows
               1 File(s)                0 bytes
               5 Dir(s)  29,055,746,048 bytes free

```

Inside the container the command “cd” navigate through directories just like normal cmd does.

```
C:\> cd Windows
```

Create a text file called “dir.txt” by redirecting the output of “dir” command.

```
C:\> dir > dir.txt
```

Check if the command succeeded by using “dir” command again.

```

C:\>dir > dir.txt

C:\>dir
Volume in drive C has no label.
Volume Serial Number is 545B-0BF0

Directory of C:\

09/20/2016  02:32 PM             448 dir.txt
08/26/2016  02:43 PM    <DIR>          Program Files
08/26/2016  02:43 PM    <DIR>          Program Files (x86)
08/26/2016  02:43 PM    <DIR>          Users
09/20/2016  02:29 PM    <DIR>          Windows
               1 File(s)                448 bytes
               4 Dir(s)  21,174,751,232 bytes free

```

When completed, exit the container.

```
C:\> Exit
```

You will now create a new container image from the modified container.

To see a list of containers run the following and take note of the container id.

```
docker ps -a
```

```

PS C:\Users\Administrator> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
a82df48c0e1b       microsoft/nanoserver  "cmd"              17 minutes ago     Exited (0) 5 minutes ago
sick diikstra

```

Run the following command to create the new 'dir' image. Replace <containerid> with the id of your container.

"docker commit <containerid> dir"

```

PS C:\Users\Administrator> docker commit e498dccb632f dir20160920
sha256:e4b31ce9ea8ed00abbe341300239a71d0b2d927efffecaf96d6766ad6291e929

```

When completed, you now have a custom image that contains the text file with dir output. This can be seen with the following command.

"docker images"

```

PS C:\Users\Administrator> docker images
REPOSITORY          TAG               IMAGE ID           CREATED            SIZE
dir20160920         latest           e4b31ce9ea8e      About a minute ago 1.007 GB
microsoft/nanoserver latest           3a703c6e97a2      3 months ago      969.8 MB

```

Optionally, to remove the container using the following command.

"Docker rm -f a83df48c0e1b", where "a83df48c0e1b" is the container ID that is shown in "docker ps" output.

## HPE Storage

### HPE Storage 3PAR StoreServ

#### "Supported External Storage" as supported with WS2016

HPE 3PAR StoreServe Series	3PAR OS Version <sup>1,2</sup>
HPE 3PAR StoreServ 8000 Series	3.2.2
HPE 3PAR StoreServ 20000 Series	3.2.2
HPE 3PAR StoreServ 7000 Series	3.2.2
HPE 3PAR StoreServ 10000 Series	3.2.2

<sup>1</sup> Windows Server 2016 Support for 3PAR OS Version 3.2.1 (See SPOCK for details. [hpe.com/storage/spock](http://hpe.com/storage/spock))

<sup>2</sup> Includes support for Boot From SAN and Direct Connect

StoreVirtual 4000: Host supported with 12.6

StoreVirtual VSA: Host supported with 12.6

StoreVirtual 3200: See SPOCK for detailed FW versions and official support details

### MSA StoreEasy: Supported via SMB up to 3.0.2

Supported with HPE MSA 1040 Storage, HPE MSA 2040 Storage, HPE MSA 2042 Storage

Firmware version: GL220P008

#### StoreOnce:

Supported FW version 3.14 or later

#### StoreEver:

Supported with existing FW

#### XP/P9500:

XP7 supported using 80-01-24 and later

P9500 support using 70-06-34 and later

SPOCK should be consulted for the latest interoperability information.

HPE 3PAR StoreServ	3PAR OS Version (Oct' 2016)	3PAR OS Version (Jan' 2017)	3PAR OS Version (Feb' 2017)
SCVMM + ASR certification	Yes	Yes	Yes – Apr' 17
SCOM	Yes – Dec' 16	Yes – Dec' 16	Yes – Apr' 17
Windows Containers	Yes – Jan' 17	Yes – Jan' 17	Yes – Jan' 17
HPE StoreVirtual	LH OS Version (Nov' 2016)	LH OS Version (Mar' 2017)	SV OS Version (Dec' 2016*)
SCVMM Certification	Yes	Yes	Yes
SCOM	Yes – Dec' 16	Yes – Apr' 17	Yes
HPE MSA Storage	MSA FW Version (Nov' 2016)		
SCVMM Certification	T8D – DER route		
SCOM	Yes – Dec' 16		
HPE XP Storage	XP7 FW Version (Oct' 2016)		
SCVMM Certification	Yes – Mar' 17		
SCOM	Yes – Dec' 16		
HPE StoreOnce Backup	StoreOnce FW Version (Oct' 2016)		
SCOM	Yes – Dec' 16		
HPE StoreEver Backup	StoreEver FW Version (Nov' 2016)		
SCOM	Yes – Dec' 16		

HPE StoreEasy Storage (Gen 9 based)	StoreEasy Shavano, Antora Peak Release (Dec' 2016)
SCOM	Yes – Dec' 16

## Resources

[technet.microsoft.com/en-us/library/mt126109.aspx](http://technet.microsoft.com/en-us/library/mt126109.aspx)

[blogs.technet.com/b/clausjor/archive/2015/05/14/storage-spaces-direct.aspx](http://blogs.technet.com/b/clausjor/archive/2015/05/14/storage-spaces-direct.aspx)

[blogs.msdn.com/b/clustering/archive/2015/05/27/10617612.aspx](http://blogs.msdn.com/b/clustering/archive/2015/05/27/10617612.aspx)

[aka.ms/privsec](http://aka.ms/privsec)

[aka.ms/nanoserver](http://aka.ms/nanoserver)

[microsoft.com/en-us/cloud-platform/windows-server](http://microsoft.com/en-us/cloud-platform/windows-server)

What's New in the Windows Server 2016 [technet.microsoft.com/library/dn765472.aspx](http://technet.microsoft.com/library/dn765472.aspx)

What's new in Windows Server 2016 Essentials

[microsoft.com/en-us/evalcenter/evaluate-windows-server-essentials-technicalpreview](http://microsoft.com/en-us/evalcenter/evaluate-windows-server-essentials-technicalpreview)

Microsoft Windows Server 2016 Product Page

[microsoft.com/en-us/server-cloud/products/windows-server-2016/](http://microsoft.com/en-us/server-cloud/products/windows-server-2016/)

Microsoft Windows Server 2016 Release Notes [technet.microsoft.com/library/dn765470.aspx](http://technet.microsoft.com/library/dn765470.aspx)

Shielded VMs and Guarded Fabric Validation Guide for Windows Server 2016 (TPM)

[gallery.technet.microsoft.com/Shielded-VMs-and-Guarded-b05d8078](http://gallery.technet.microsoft.com/Shielded-VMs-and-Guarded-b05d8078)

[channel9.msdn.com/events/Ignite/2015/BRK2482](http://channel9.msdn.com/events/Ignite/2015/BRK2482)

[technet.microsoft.com/en-US/library/mt130644.aspx](http://technet.microsoft.com/en-US/library/mt130644.aspx)

Learn more at

[hpe.com/servers](http://hpe.com/servers)

[microsoft.com/en-us/cloud-platform/windows-server](http://microsoft.com/en-us/cloud-platform/windows-server)

[Microsoft.com/windowsserver](http://Microsoft.com/windowsserver)

[hpe.com/us/en/storage.html](http://hpe.com/us/en/storage.html)



---

**Sign up for updates**

---

---

© Copyright 2014–2016 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Java is a registered trademark of Oracle and/or its affiliates. Intel and Intel Xeon are trademarks of Intel Corporation in the U.S. and other countries. SD is a trademark or registered trademark of SD-3C in the United States, other countries or both. AMD is a trademark of Advanced Micro Devices, Inc.

4AA5-5841ENW, October 2016, Rev. 8



**Hewlett Packard  
Enterprise**