Contents

Get Started with Windows Server 2016

3/17/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server 2016

TIP

Looking for information about older versions of Windows Server? Check out our other Windows Server libraries on docs.microsoft.com. You can also search this site for specific information.

This collection contains detailed information to help you determine if you're ready to move to Windows Server 2016. Once you've checked the system requirements, upgrade options, and other information about moving to Windows Server 2016, you're ready to go back to the main Windows Server 2016 hub and start down the path to installing the best edition and installation option for your needs.

NOTE

To download Windows Server 2016, see Windows Server Evaluations.

System Requirements

Find out the minimum hardware requirements to install and run Windows Server 2016.

Release Notes: Important Issues in Windows Server

Issues that could cause serious problems if you don't avoid or work around them.

Recommendations for moving to Windows Server 2016

Comprehensive table of available approaches for getting to Windows Server 2016 in various scenarios.

Features Removed or Deprecated in Windows Server 2016

Features that have already been removed from Windows Server 2016 or designated for potential future removal.

Upgrade and Conversion Options

Description of all the ways to move to Windows Server 2016 from whatever you're running today.

Server Role Upgrade and Migration Matrix

Information about additional steps needed to bring particular server roles to Windows Server 2016

Server Application Compatibility Table

Does SQL work on Windows Server 2016? What steps are needed to get Exchange running? This topic explains what you'll need to do.

Server Activation Guide

Basic information on activation of Windows Server 2016 itself and other operating systems by using Windows Server 2016.

Introducing Windows Server, version 1803

8/10/2018 • 4 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel)

Windows Server, version 1803 is the current release in the new Semi-Annual Channel

What the Semi-Annual Channel is--and isn't

Windows Server, version 1803 is *not* an "update" or "service pack" for Windows Server 2016. It's the current twice-yearly server release on the release track that is designed for customers who are moving at a "cloud cadence," such as those on rapid development cycles. This track is ideal for modern applications and innovation scenarios such as containers and micro-services. Each release in this track is supported for 18 months from the initial release. For more about Semi-Annual Channel, plus **tips for deciding which channel to join** (or remain on) see Semi-Annual Channel Overview.

Windows Server 2016 is the current Long-Term Servicing Channel (LTSC) product. The LTSC is best if you need long-term stability and predictability in your server operating system to support traditional workloads and applications. If you want to stay in the LTSC, you should install (or continue using) Windows Server 2016, which can be installed in either Server Core mode or Server with Desktop Experience mode. See Get Started with Windows Server 2016 for details.

What's different about Windows Server, version 1803?

Windows Server, version 1803 runs in Server Core mode. Windows Server Core mode offers great advantages such as smaller hardware requirements, much smaller attack surface, and a reduction in the need for updates. Since it has no graphical user interface, Windows Server Core mode is best managed remotely. If you're new to working with Server Core, Manage a Server Core server will help you get used to this environment. Manage Windows Server 2016 shows you the various options for managing servers remotely.

What's New in Windows Server version 1803 introduces you to the new features and functionality added in Windows Server, version 1803.

Why does Windows Server, version 1803 offer only the Server Core installation option?

One of the most important steps we take in planning each release of Windows Server is listening to customer feedback – how are you using Windows Server? What new features will have the greatest impact on your Windows Server deployments, and by extension, your day-to-day business? Your feedback tells us that delivering new innovation as quickly and efficiently as possible is a key priority. At the same time, for those customers innovating most quickly, you've told us that you're primarily using command line scripting with PowerShell to manage your datacenters, and as such don't have a strong need for the desktop GUI available in the installation of Windows Server with Desktop Experience. By focusing on the Server Core installation option, we're able to dedicate more resources toward those new innovations, while also maintaining traditional Windows Server platform functionality and application compatibility. If you have feedback about this or other issues concerning Windows Server and our future releases, you can make suggestions and comments through the Feedback Hub.

What about Nano Server?

Nano Server is available as a container operating system. See Changes to Nano Server in Windows Server Semi-Annual Channel for details.

Additional information about this release

To get a comprehensive view of the key facts about Windows Server, version 1803, you should also review these topics prior to installing it:

- What hardware is needed to run it? See System requirements; the system requirements for this release are the same as for Windows Server 2016.
- What new features and functionality have been added? See What's New in Windows Server version 1803
- What has been removed? See Features Removed or Planned for Replacement starting with Windows Server, version 1803
- What issues unique to this release need to be worked around? See Release notes--Important issues in Windows Server, version 1803

Where to obtain Windows Server, version 1803

This release should be installed as a clean installation.

- Volume Licensing Service Center (VLSC): Volume-licensed customers with Software Assurance can obtain this release by going to the Volume Licensing Service Center and clicking **Sign In**. Then click **Downloads and Keys** and search for this release.
- Windows Server, version 1803 is also available in Microsoft Azure.
- Visual Studio Subscriptions: Visual Studio Subscribers can obtain Windows Server, version 1803 by downloading it from the Visual Studio Subscriber download page. If you are not already a subscriber, go to Visual Studio Subscriptions to sign up, and then visit the Visual Studio Subscriber download page as above. Releases obtained through Visual Studio Subscriptions are for development and testing only.

Activating Windows Server, version 1803

- If you've obtained this release from the Volume Licensing Service Center, you can activate it by using your Windows Server 2016 CSVLK with your Key Management System (KMS) environment.
- If you are using Microsoft Azure, this release should automatically be activated.
- If you obtain this release from Visual Studio Subscriptions, you can activate it by using your Windows Server 2016 CSVLK with your Key Management System (KMS) environment.

What's New in Windows Server

8/28/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel), Windows Server 2016

You can find the new features for each of these Windows Server 2016 and Windows Server (Semi-Annual Channel) releases.

What's New in Windows Server, version 1803
What's New in Windows Server, version 1709
What's New in Windows Server 2016

See Also

- Release Notes: Important Issues in Windows Server 2016

Express updates for Windows Server 2016 re-enabled for November 2018 update

11/13/2018 • 2 minutes to read • Edit Online

By Joel Frauenheim

Applies To: Windows Server 2016

Starting with the November 13, 2018 Update Tuesday, Windows will again publish Express updates for Windows Server 2016. Express updates for Windows Server 2016 stopped in mid-2017 after a significant issue was found that kept the updates from installing correctly. While the issue was fixed in November 2017, the update team took a conservative approach to publishing the Express packages to ensure most customers would have the November 14, 2017 update (KB 4048953) installed on their server environments and not be impacted by the issue.

System administrators for WSUS and System Center Configuration Manager (SCCM) need to be aware that in November 2018 they will once again see two packages for the Windows Server 2016 update: a Full update and an Express update. System administrators who want to use Express for their server environments need to confirm that the device has taken a full update since November 14, 2017 (KB 4048953) to ensure the Express update installs correctly. Any device which has not been updated since the November 14, 2017 update (KB 4048953) will see repeated failures that consume bandwidth and CPU resources in an infinite loop if the Express update is attempted. Remediation for that state would be for the system administrator to stop pushing the Express update and push a recent Full update to stop the failure loop.

With the November 13, 2018 Express update customers will see an immediate reduction of package size between their management system and the Windows Server 2016 end points.

What's New in Windows Server version 1809

10/26/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel)

This topic describes some of the new features in Windows Server, version 1809. To find out What's New in Windows Server 2019, see What's New in Windows Server 2019.

Container networking with Kubernetes

Container networking with Kubernetes in Windows Server 2019 greatly improves usability of Kubernetes on Windows by enhancing platform networking resiliency and support of container networking plugins. Additionally, customers deploying workloads on Kubernetes network security to protect both Linux and Windows services using embedded tooling.

Group managed service accounts for containers

Windows Server, version 1809 has improvements in the scalability and reliability of containers that use group managed service accounts (gMSA) to access network resources.

Host device access for containers

Simple buses can be assigned to process-isolated Windows Server containers. Applications running in containers that need to talk over SPI, I2C, GPIO, and UART/COM are now able to do so.

Additional features

In addition to features that are new in Windows Server version 1809, the following new features and capabilities for Windows Server 2019 also apply to Windows Server version 1809:

- Container improvements
- HTTP/2
- Kubernetes support
- Linux containers on Windows
- Low Extra Delay Background Transport (LEDBAT)
- Network performance improvements for virtual workloads
- Server Core App Compatibility feature on demand (FOD)
- Storage Migration Service (SMS)
- Storage Replica
- System Insights
- Windows Defender Advanced Threat Protection (ATP)
- Windows Defender ATP Exploit Guard
- Windows Time Service

What's New in Windows Server version 1803

6/15/2018 • 4 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel)

The content in this section describes what's new and changed in Windows Server, version 1803. The new features and changes listed here are the ones most likely to have the greatest impact as you work with this release. Also see Windows Server Semi-Annual Channel update.

Windows Admin Center

Project Honolulu is now the **Windows Admin Center**.

Windows Admin Center consolidates all aspects of local and remote server management. Windows Admin Center is a locally deployed, browser-based management experience that does not require an Internet connection, giving you full control of all aspects of your Windows Server deployment.

Windows Server release strategy

Windows Server version 1709 was released in September 2017 as the first release in the Semi-Annual Channel. The Semi-Annual Channel has a faster release cadence and addresses feedback from those who want rapid innovation every few months. This complements the Long-Term Servicing Channel where the release cadence is every 2-3 years.

Based on telemetry and feedback, these channels have demonstrated that they conform well to the following general strategy:

- The Semi-Annual Channel is ideal for modern applications and innovation scenarios, such as containers and micro-services.
- The Long-Term Servicing Channel is the preferred release for core infrastructure scenarios such as software-defined datacenter and hyper-converged infrastructure (HCI).

The specific scenarios for the Semi-Annual Channel and the Long-Term Servicing Channel are as follows:

	LONG TERM SERVICING CHANNEL	SEMI-ANNUAL CHANNEL
Recommended scenarios	General purpose file servers, first and third-party workloads, traditional apps, infrastructure roles, software-defined Datacenter, and Hyper-converged infrastructure	Containerized applications, container hosts, and application scenarios benefiting from faster innovation
New releases	Every 2–3 years	Every 6 months
Support	5 years of Mainstream support + 5 years of Extended support	18 months
Editions	All available Windows Server editions	Standard and Datacenter editions

	LONG TERM SERVICING CHANNEL	SEMI-ANNUAL CHANNEL
Who can use	All customers through all channels	Software Assurance and cloud customers only
Installation options	Server Core and Server with Desktop Experience	Server Core for container host, container image, and Nano Server container image

Application platform and containers

- Optimization
  - The Server Core base container image is reduced by 30% from Windows Server, version 1709.
  - Application compatibility is also improved to help with containerizing of traditional applications.
  - Container boot performance and run-time performance are improved as well thanks to various fixes and optimizations.
- Container networking: Localhost and http proxy support has been added, and container scalability and startup time is improved.
- Tools: Support for Curl.exe, Tar.exe, and SSH has been enhanced to complement PowerShell for building and debugging scenarios.

Server Core container image

A smaller Server Core container with better application compatibility is now available. Detailed information is available here.

- Unused optional features and roles have been removed. For more information see Roles, Role Services, and Features not in Server Core containers.
  - Decreased download size to 1.58 GB, 30% reduction from Windows Server, version 1709.
  - Decreased size on disk to 3.61 GB, 20% reduction from Windows Server, version 1709.
- Nano Server container image is below 100MB

Windows Subsystem for Linux (WSL)

WSL enables server administrators to use existing tools and scripts from Linux on Windows Server. Many improvements showcased in the command line blog are now part of Windows Server, including Background tasks, DriveFS, WSLPath, and much more.

Kubernetes

Kubernetes (commonly referred to as K8s) is an open source system for automating deployment, scaling and management of containerized applications developed under the stewardship of the Cloud Native Computing Foundation.

In Windows Server, version 1709 users were able to take advantage of Kubernetes on Windows networking features, including:

- Shared pod compartments: Infrastructure and worker pods now share a network compartment (analogous to a Linux namespace).
- Endpoint optimization: Thanks to compartment sharing, container services need to track at least half as many endpoints.
- Data-path optimization: Improvements to the Virtual Filtering Platform and the Host Networking Service allow kernel-based load-balancing.

With the release of Windows Server, version 1803 more features will be available in coming Kubernetes releases:

- Storage plugins for Windows containers orchestrated by Kubernetes.

- Cloud scale networking though initiatives like our partnership with Tigera on Project Calico support.
- Windows platform support for Hyper-V isolated Pods with multiple containers per Pod.

Application compatibility and feature parity issues fixed

- Microsoft Message Queuing (MSMQ) now installs in a Server Core container.
- An issue that breaks ASP.net performance counters has been fixed.
- An issue where services running in containers did not receive shutdown notification has been fixed.
  - Specifically, the notification is changed to CTRL_SHUTDOWN_EVENT for both Server Core and Nano Server container based images. In addition, it extends the notification in Server Core container based images to affect all process running in the container, including sending service shutdown notifications to services running in the container.
- An incompatibility of docker pull & docker load with the policy setting that determines whether BitLocker protection is required for fixed data drives to be writable (FDVDenyWriteAccess) has been fixed.

Storage

With this release, it is possible to prevent the File Server Resource Manager service from creating a change journal (also known as a USN journal) on all volumes when the service starts. This can conserve space on each volume, but will disable real-time file classification. For more information, see File Server Resource Manager overview.

Features added to Server Core

The Transport Server role in Windows Deployment Services (WDS) role has been added to Server Core.

Transport Server contains only the core networking parts of WDS. You can use Transport Server to create multicast namespaces that transmit data (including operating system images) from a standalone server. You can also use it if you want to have a PXE server that allows clients to PXE boot and download your own custom setup application. You should use this option if you want to use either of these scenarios.

You can use the following Windows PowerShell command to enable the Transport Server service on Server Core:

```
Install-WindowsFeature -Name WDS
```

See also

Windows Server release information
What's new in Windows 10, version 1803 IT Pro content

What's New in Windows Server version 1709

10/15/2018 • 9 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel)

The content in this section describes what's new and changed in Windows Server, version 1709. The new features and changes listed here are the ones most likely to have the greatest impact as you work with this release. Also see Windows Server, version 1709.

New cadence of releases

Starting with this release, you have two options for receiving Windows Server feature updates:

- **Long-Term Servicing Channel (LTSC)**: This is business as usual with 5 years of mainstream support and 5 years of extended support. You have the option to upgrade to the next LTSC release every 2-3 years in the same way that has been supported for the last 20 years.
- **Semi-Annual Channel (SAC)**: This is a Software Assurance benefit and is fully supported in production. The difference is that it is supported for 18 months and there will be a new version every six months.

Release channels are summarized in the following table.

	SEMI-ANNUAL CHANNEL	LONG TERM SERVICING CHANNEL
Release cadence	Twice a year (spring and fall)	Every 2-3 years
Support schedule	18 months mainstream production support	5 years mainstream support + 5 years extended support
Availability	Software Assurance or Azure (cloud hosted)	All channels
Naming convention	Windows Server, version YYMM	Windows Server YYYY

For more information, see Windows Server Semi-annual Channel Overview.

Application containers and micro-services

- The Server Core container image has been further optimized for lift-and-shift scenarios where you can migrate existing code bases or applications into containers with minimal changes, and it's also 60% smaller.
- The Nano Server container image is nearly 80% smaller.
  - In the Windows Server Semi-Annual Channel, Nano Server as a container base OS image is decreased from 390 MB to 80 MB.
- Linux containers with Hyper-V isolation

For more information, see Changes to Nano Server in the next release of Windows Server and Windows Server, version 1709 for developers.

Modern management

Check out Project Honolulu for a simplified, integrated, secure experience to help IT administrators manage core

troubleshooting, configuration, and maintenance scenarios. Project Honolulu includes next generation tooling with a simplified, integrated, secure, and extensible interface. Project Honolulu includes an intuitive all-new management experience for managing PCs, Windows servers, Failover Clusters, as well as hyper-converged infrastructure based on Storage Spaces Direct, reducing operational costs.

Compute

Nano Container and Server Core Container: First and foremost, this release is about driving application innovation. Nano Server, or Nano as Host is deprecated and replaced by Nano Container, which is Nano running as a container image.

For more information about containers, see Container Networking Overview.

Server Core as a container (and infrastructure) host, provides better flexibility, density and performance for existing applications under a modernization process and brands new apps developed already using the cloud model.

VM Start Ordering is also improved with OS and Application awareness, bringing enhanced triggers for when a VM is considered started before starting the next.

Storage-class memory support for VMs enables NTFS-formatted direct access volumes to be created on non-volatile DIMMs and exposed to Hyper-V VMs. This enables Hyper-V VMs to leverage the low-latency performance benefits of storage-class memory devices.

Virtualized Persistent Memory (vPMEM) is enabled by creating a VHD file (.vhdpmem) on a direct access volume on a host, adding a vPMEM Controller to a VM, and adding the created device (.vhdpmem) to a VM. Using vhdpmem files on direct access volumes on a host to back vPMEM enables allocation flexibility and leverages a familiar management model for adding disks to VMs.

Container storage – persistent data volumes on cluster shared volumes (CSV). In Windows Server, version 1709 as well as Windows Server 2016 with the latest updates, we've added support for containers to access persistent data volumes located on CSVs, including CSVs on Storage Spaces Direct. This gives the application container persistent access to the volume no matter which cluster node the container instance is running on. For more info, see Container Storage Support with Cluster Shared Volumes (CSV), Storage Spaces Direct (S2D), SMB Global Mapping.

Container storage – persistent data volumes with SMB global mapping. In Windows Server, version 1709 we've added support for mapping an SMB file share to a drive letter inside a container – this is called SMB global mapping. This mapped drive is then accessible to all users on the local server so that container I/O on the data volume can go through the mounted drive to the underlying file share. For more info, see Container Storage Support with Cluster Shared Volumes (CSV), Storage Spaces Direct (S2D), SMB Global Mapping.

Virtual machine configuration file format (updated). An additional file (.vmgs) has been added for virtual machines with a configuration version of 8.2 and higher. VMGS stands for VM guest state and is a new internal file which includes device state that was previously part of the VM runtime state file.

Security and Assurance

Network encryption enables you to quickly encrypt network segments on software-defined networking infrastructure to meet security and compliance needs.

Host Guardian Service (HGS) as a shielded VM is enabled. Prior to this release, the recommendation was to deploy a 3-node physical cluster. While this ensures the HGS environment is not compromised by an administrator, it was often cost prohibitive.

Linux as a shielded VM is now supported.

For more information, see Guarded fabric and shielded VMs overview.

SMBLoris vulnerability An issue, known as "SMBLoris", which could result in denial of service, has been addressed.

Storage

Storage Replica: The disaster recovery protection added by Storage Replica in Windows Server 2016 is now expanded to include:

- **Test failover**: the option to mount the destination storage is now possible through the test failover feature. You can mount a snapshot of the replicated storage on destination nodes temporarily for testing or backup purposes. For more information, see Frequently Asked Questions about Storage Replica.
- **Project Honolulu support**: Support for graphical management of server to server replication is now available in Project Honolulu. This removes the requirement to use PowerShell to manage a common disaster protection workload.

SMB:

- **SMB1 and guest authentication removal**: Windows Server, version 1709 no longer installs the SMB1 client and server by default. Additionally, the ability to authenticate as a guest in SMB2 and later is off by default. For more information, review SMBv1 is not installed by default in Windows 10, version 1709 and Windows Server, version 1709.
- **SMB2/SMB3 security and compatibility**: Additional options for security and application compatibility were added, including the ability to disable oplocks in SMB2+ for legacy applications, as well as require signing or encryption on per-connection basis from a client. For more information, review the SMBShare PowerShell module help.

Data Deduplication:

- **Data Deduplication now supports ReFS**: You no longer must choose between the advantages of a modern file system with ReFS and the Data Deduplication: now, you can enable Data Deduplication wherever you can enable ReFS. Increase storage efficiency by upwards of 95% with ReFS.
- **DataPort API for optimized ingress/egress to deduplicated volumes**: Developers can now take advantage of the knowledge Data Deduplication has about how to store data efficiently to move data between volumes, servers, and clusters efficiently.

Remote Desktop Services (RDS)

RDS is integrated with Azure AD, so customers can leverage Conditional Access policies, Multifactor Authentication, Integrated authentication with other SaaS Apps using Azure AD, and many more. For more information, see Integrate Azure AD Domain Services with your RDS deployment.

> **TIP**
> For a sneak peek at other exciting changes coming to RDS, see Remote Desktop Services: Updates & upcoming innovations

Networking

Docker's Routing Mesh is supported. Ingress routing mesh is part of swarm mode, Docker's built-in orchestration solution for containers. For more information, see Docker's routing mesh available with Windows Server version 1709.

New features for Docker are available. For more information, see Exciting new things for Docker with Windows

Server 1709.

Windows Networking at Parity with Linux for Kubernetes: Windows is now on par with Linux in terms of networking. Customers can deploy mixed-OS, Kubernetes clusters in any environment including Azure, on-premises, and on 3rd-party cloud stacks with the same network primitives and topologies supported on Linux without the need for any workarounds or switch extensions.

Core network stack: Several features of the core network stack are improved. For more information about these features, see Core Network Stack Features in the Creators Update for Windows 10.

- **TCP Fast Open (TFO)**: Support for TFO has been added to optimize the TCP 3-way handshake process. TFO establishes a secure TFO cookie in the first connection using a standard 3-way handshake. Subsequent connections to the same server use the TFO cookie instead of a 3-way handshake to connect with zero round trip time.
- **CUBIC**: Experimental Windows native implementation of CUBIC, a TCP congestion control algorithm is available. The following commands enable or disable CUBIC, respectively.

  ```
  netsh int tcp set supplemental template=internet congestionprovider=cubic
  netsh int tcp set supplemental template=internet congestionprovider=compound
  ```

- **Receive Window Autotuning**: TCP autotuning logic computes the "receive window" parameter of a TCP connection. High speed and/or long delay connections need this algorithm to achieve good performance characteristics. In this release, the algorithm is modified to use a step function to converge on the maximum receive window value for a given connection.
- **TCP stats API**: A new API is introduced called SIO_TCP_INFO. SIO_TCP_INFO allows developers to query rich information on individual TCP connections using a socket option.
- **IPv6**: There are multiple improvements in IPv6 in this release.
  - **RFC 6106** support: RFC 6106 which allows for DNS configuration through router advertisements (RAs). You can use the following command to enable or disable RFC 6106 support:

    ```
    netsh int ipv6 set interface <ifindex> rabaseddnsconfig=<enabled | disabled>
    ```

  - **Flow Labels**: Beginning with the Creators Update, outbound TCP and UDP packets over IPv6 have this field set to a hash of the 5-tuple (Src IP, Dst IP, Src Port, Dst Port). This will make IPv6 only datacenters doing load balancing or flow classification more efficient. To enable flowlabels:

    ```
    netsh int ipv6 set flowlabel=[disabled|enabled] (enabled by default)
    netsh int ipv6 set global flowlabel=<enabled | disabled>
    ```

  - **ISATAP and 6to4**: As a step towards future deprecation, the Creators Update will have these technologies disabled by default.
- **Dead Gateway Detection (DGD)**: The DGD algorithm automatically transitions connections over to another gateway when the current gateway is unreachable. In this release, the algorithm is improved to periodically re-probe the network environment.
- Test-NetConnection is a built-in cmdlet in Windows PowerShell that performs a variety of network diagnostics. In this release we have enhanced the cmdlet to provide detailed information about both route selection as well as source address selection.

Software Defined Networking

- **Virtual Network Encryption** is a new feature that provides the ability for the virtual network traffic to be

encrypted between Virtual Machines that communicate with each other within subnets that are marked as "Encryption Enabled". This feature utilizes Datagram Transport Layer Security (DTLS) on the virtual subnet to encrypt the packets. DTLS provides protection against eavesdropping, tampering and forgery by anyone with access to the physical network.

Windows 10 VPN

- **Pre-Logon Infrastructure Tunnels**. By default, Windows 10 VPN does not automatically create Infrastructure Tunnels when users are not logged on to their computer or device. You can configure Windows 10 VPN to automatically create Pre-Logon Infrastructure Tunnels by using the Device Tunnel (prelogon) feature in the VPN profile.
- **Management of Remote Computers and Devices**. You can manage Windows 10 VPN clients by configuring the Device Tunnel (prelogon) feature in the VPN profile. In addition, you must configure the VPN connection to dynamically register the IP addresses that are assigned to the VPN interface with internal DNS services.
- **Specify Pre-Logon Gateways**. You can specify Pre-Logon Gateways with the Device Tunnel (prelogon) feature in the VPN profile, combined with traffic filters to control which management systems on the corporate network are accessible via the device tunnel.

What's New in Windows Server 2016

10/3/2018 • 11 minutes to read • Edit Online

Applies To: Windows Server 2016

The content in this section describes what's new and changed in Windows Server® 2016. The new features and changes listed here are the ones most likely to have the greatest impact as you work with this release.

Compute

The Virtualization area includes virtualization products and features for the IT professional to design, deploy, and maintain Windows Server.

General

Physical and virtual machines benefit from greater time accuracy due to improvements in the Win32 Time and Hyper-V Time Synchronization Services. Windows Server can now host services that are compliant with upcoming regulations which require a 1ms accuracy with regards to UTC.

Hyper-V

- What's new in Hyper-V on Windows Server 2016. This topic explains the new and changed functionality of the Hyper-V role in Windows Server 2016, Client Hyper-V running on Windows 10, and Microsoft Hyper-V Server 2016.
- Windows Containers: Windows Server 2016 container support adds performance improvements, simplified network management, and support for Windows containers on Windows 10. For some additional information on containers, see Containers: Docker, Windows and Trends.

Nano Server

What's New in Nano Server. Nano Server now has an updated module for building Nano Server images, including more separation of physical host and guest virtual machine functionality as well as support for different Windows Server editions.

There are also improvements to the Recovery Console, including separation of inbound and outbound firewall rules as well as the ability to repair the configuration of WinRM.

Shielded Virtual Machines

Windows Server 2016 provides a new Hyper-V-based Shielded Virtual Machine to protect any Generation 2 virtual machine from a compromised fabric. Among the features introduced in Windows Server 2016 are the following:

- New "Encryption Supported" mode that offers more protections than for an ordinary virtual machine, but less than "Shielded" mode, while still supporting vTPM, disk encryption, Live Migration traffic encryption, and other features, including direct fabric administration conveniences such as virtual machine console connections and Powershell Direct.

- Full support for converting existing non-shielded Generation 2 virtual machines to shielded virtual machines, including automated disk encryption.
- Hyper-V Virtual Machine Manager can now view the fabrics upon which a shielded virtual is authorized to run, providing a way for the fabric administrator to open a shielded virtual machine's key protector (KP) and view the fabrics it is permitted to run on.
- You can switch Attestation modes on a running Host Guardian Service. Now you can switch on the fly between the less secure but simpler Active Directory-based attestation and TPM-based attestation.
- End-to-end diagnostics tooling based on Windows PowerShell that is able to detect misconfigurations or errors in both guarded Hyper-V hosts and the Host Guardian Service.
- A recovery environment that offers a means to securely troubleshoot and repair shielded virtual machines within the fabric in which they normally run while offering the same level of protection as the shielded virtual machine itself.
- Host Guardian Service support for existing safe Active Directory – you can direct the Host Guardian Service to use an existing Active Directory forest as its Active Directory instead of creating its own Active Directory instance

For more details and instructions for working with shielded virtual machines, see Shielded VMs and Guarded Fabric Validation Guide for Windows Server 2016 (TPM).

Identity and Access

New features in Identity improve the ability for organizations to secure Active Directory environments and help them migrate to cloud-only deployments and hybrid deployments, where some applications and services are hosted in the cloud and others are hosted on premises.

Active Directory Certificate Services

Active Directory Certificate Services (AD CS) in Windows Server 2016 increases support for TPM key attestation: You can now use Smart Card KSP for key attestation, and devices that are not joined to the domain can now use NDES enrollment to get certificates that can be attested for keys being in a TPM.

Active Directory Domain Services

Active Directory Domain Services includes improvements to help organizations secure Active Directory environments and provide better identity management experiences for both corporate and personal devices. For more information, see What's new in Active Directory Domain Services (AD DS) in Windows Server 2016.

Active Directory Federation Services

What's New in Active Directory Federation Services. Active Directory Federation Services (AD FS) in Windows Server 2016 includes new features that enable you to configure AD FS to authenticate users stored in Lightweight Directory Access Protocol (LDAP) directories. For more information, see What's New in AD FS for Windows Server 2016.

Web Application Proxy

The latest version of Web Application Proxy focuses on new features that enable publishing and preauthentication for more applications and improved user experience. Check out the full list of new features that includes preauthentication for rich client apps such as Exchange ActiveSync and wildcard domains for easier publishing of SharePoint apps. For more information, see Web Application Proxy in Windows Server 2016.

Administration

The Management and Automation area focuses on tool and reference information for IT pros who want to run and manage Windows Server 2016, including Windows PowerShell.

Windows PowerShell 5.1 includes significant new features, including support for developing with classes and new security features that extend its use, improve its usability, and allow you to control and manage Windows-based environments more easily and comprehensively. See New Scenarios and Features in WMF 5.1 for details.

New additions for Windows Server 2016 include: the ability to run PowerShell.exe locally on Nano Server (no longer remote only), new Local Users & Groups cmdlets to replace the GUI, added PowerShell debugging support, and added support in Nano Server for security logging & transcription and JEA.

Here are some other new administration features:

PowerShell Desired State Configuration (DSC) in Windows Management Framework (WMF) 5

Windows Management Framework 5 includes updates to Windows PowerShell Desired State Configuration (DSC), Windows Remote Management (WinRM), and Windows Management Instrumentation (WMI).

For more info about testing the DSC features of Windows Management Framework 5, see the series of blog posts discussed in Validate features of PowerShell DSC. To download, see Windows Management Framework 5.1.

PackageManagement unified package management for software discovery, installation, and inventory

Windows Server 2016 and Windows 10 includes a new PackageManagement feature (formerly called OneGet) that enables IT Professionals or DevOps to automate software discovery, installation, and inventory (SDII), locally or remotely, no matter what the installer technology is and where the software is located.

For more info, see https://github.com/OneGet/oneget/wiki.

PowerShell enhancements to assist digital forensics and help reduce security breaches

To help the team responsible for investigating comprimised systems - sometimes known as the "blue team" - we've added additional PowerShell logging and other digital forensics functionality, and we've added functionality to help reduce vulnerabilities in scripts, such as constrained PowerShell, and secure CodeGeneration APIs.

For more info, see PowerShell ♥ the Blue Team.

Networking

This area addresses networking products and features for the IT professional to design, deploy, and maintain Windows Server 2016.

Software-Defined Networking

You can now both mirror and route traffic to new or existing virtual appliances. Together with a distributed firewall and Network security groups, this enables you to dynamically segment and secure workloads in a manner similar to Azure. Second, you can deploy and manage the entire Software-defined networking (SDN) stack using System Center Virtual Machine Manager. Finally, you can use Docker to manage Windows Server container networking, and associate SDN policies not only with virtual machines but containers as well. For more information, see Plan a Software Defined Network Infrastructure.

TCP performance improvements

The default Initial Congestion Window (ICW) has been increased from 4 to 10 and TCP Fast Open (TFO) has been implemented. TFO reduces the amount of time required to establish a TCP connection and the increased ICW allows larger objects to be transferred in the initial burst. This combination can significantly reduce the time required to transfer an Internet object between the client and the cloud.

In order to improve TCP behavior when recovering from packet loss we have implemented TCP Tail Loss Probe (TLP) and Recent Acknowledgement (RACK). TLP helps convert Retransmit TimeOuts (RTOs) to Fast Recoveries and RACK reduces the time required for Fast Recovery to retransmit a lost packet.

Security and Assurance

Includes security solutions and features for the IT professional to deploy in your datacenter and cloud environment. For information about security in Windows Server 2016 generally, see Security and Assurance.

Just Enough Administration

Just Enough Administration in Windows Server 2016 is security technology that enables delegated administration for anything that can be managed with Windows PowerShell. Capabilities include support for running under a network identity, connecting over PowerShell Direct, securely copying files to or from JEA endpoints, and configuring the PowerShell console to launch in a JEA context by default. For more details, see JEA on GitHub.

Credential Guard

Credential Guard uses virtualization-based security to isolate secrets so that only privileged system software can access them. See Protect derived domain credentials with Credential Guard.

Remote Credential Guard

Credential Guard includes support for RDP sessions so that the user credentials remain on the client side and are not exposed on the server side. This also provides Single Sign On for Remote Desktop. See Protect derived domain credentials with Windows Defender Credential Guard.

Device Guard (Code Integrity)

Device Guard provides kernel mode code integrity (KMCI) and user mode code integrity (UMCI) by creating policies that specify what code can run on the server. See Introduction to Windows Defender Device Guard: virtualization-based security and code integrity policies.

Windows Defender

Windows Defender Overview for Windows Server 2016. Windows Server Antimalware is installed and enabled by default in Windows Server 2016, but the user interface for Windows Server Antimalware is not installed. However, Windows Server Antimalware will update antimalware definitions and protect the computer without the user interface. If you need the user interface for Windows Server Antimalware, you can install it after the operating system installation by using the Add Roles and Features Wizard.

Control Flow Guard

Control Flow Guard (CFG) is a platform security feature that was created to combat memory corruption vulnerabilities. See Control Flow Guard for more information.

Storage

Storage in Windows Server 2016 includes new features and enhancements for software-defined storage, as well as for traditional file servers. Below are a few of the new features, for more enhancements and further details, see What's New in Storage in Windows Server 2016.

Storage Spaces Direct

Storage Spaces Direct enables building highly available and scalable storage using servers with local storage. It simplifies the deployment and management of software-defined storage systems and unlocks use of new classes of disk devices, such as SATA SSD and NVMe disk devices, that were previously not possible with clustered Storage Spaces with shared disks.

For more info, see Storage Spaces Direct.

Storage Replica

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

For more info, see Storage Replica.

Storage Quality of Service (QoS)

You can now use storage quality of service (QoS) to centrally monitor end-to-end storage performance and create management policies using Hyper-V and CSV clusters in Windows Server 2016.

For more info, see Storage Quality of Service.

Failover Clustering

Windows Server 2016 includes a number of new features and enhancements for multiple servers that are grouped together into a single fault-tolerant cluster using the Failover Clustering feature. Some of the additions are listed below; for a more complete listing, see What's New in Failover Clustering in Windows Server 2016.

Cluster Operating System Rolling Upgrade

Cluster Operating System Rolling Upgrade enables an administrator to upgrade the operating system of the cluster nodes from Windows Server 2012 R2 to Windows Server 2016 without stopping the Hyper-V or the Scale-Out File Server workloads. Using this feature, the downtime penalties against Service Level Agreements (SLA) can be avoided.

For more info, see Cluster Operating System Rolling Upgrade.

Cloud Witness

Cloud Witness is a new type of Failover Cluster quorum witness in Windows Server 2016 that leverages Microsoft Azure as the arbitration point. The Cloud Witness, like any other quorum witness, gets a vote and can participate in the quorum calculations. You can configure cloud witness as a quorum witness using the Configure a Cluster Quorum Wizard.

For more info, see Deploy Cloud Witness.

Health Service

The Health Service improves the day-to-day monitoring, operations, and maintenance experience of cluster resources on a Storage Spaces Direct cluster.

For more info, see Health Service.

Application development

Internet Information Services (IIS) 10.0

New features provided by the IIS 10.0 web server in Windows Server 2016 include:

- Support for HTTP/2 protocol in the Networking stack and integrated with IIS 10.0, allowing IIS 10.0 websites to automatically serve HTTP/2 requests for supported configurations. This allows numerous enhancements over HTTP/1.1 such as more efficient reuse of connections and decreased latency, improving load times for web pages.
- Ability to run and manage IIS 10.0 in Nano Server. See IIS on Nano Server.
- Support for Wildcard Host Headers, enabling administrators to set up a web server for a domain and then have the web server serve requests for any subdomain.
- A new PowerShell module (IISAdministration) for managing IIS.

For more details see IIS.

Distributed Transaction Coordinator (MSDTC)

Three new features are added in Microsoft Windows 10 and Windows Server 2016:

- A new interface for Resource Manager Rejoin can be used by a resource manager to determine the outcome of an in-doubt transaction after a database restarts due to an error. See

IResourceManagerRejoinable::Rejoin for details.

- The DSN name limit is enlarged from 256 bytes to 3072 bytes. See IDtcToXaHelperFactory::Create, IDtcToXaHelperSinglePipe::XARMCreate, or IDtcToXaMapper::RequestNewResourceManager for details.
- Improved tracing allowing you to set a registry key to include an image file path in the trace log file name so you can tell which trace log file to check. See How to enable diagnostic tracing for MS DTC on a Windows-based computer for details on configuring tracing for MSDTC.

See Also

- Release Notes: Important Issues in Windows Server 2016

What's New in the Windows Console in Windows Server 2016

6/20/2018 • 6 minutes to read • Edit Online

Applies To: Windows Server 2016

The console host (the underlying code that supports all character-mode applications including the Windows command prompt, the Windows PowerShell prompt, and others) has been updated in several ways to add a variety of new capabilities.

Controlling the new features

The new functionality is enabled by default, but you can switch each of the new features on and off or revert to the previous console host either through the Properties interface (mostly on the **Options** tab) or with these Registry keys (all keys are DWORD values under **HKEY_CURRENT_USER\Console**):

REGISTRY KEY	DESCRIPTION
ForceV2	1 enables all new console features; 0 disables all new features. Note: this value is not stored in shortcuts, but only in this Registry key.
LineSelection	1 enables line selection; 0 to use block mode only
FilterOnPaste	1 enables new paste behavior
LineWrap	1 wraps text when you resize console windows
CtrlKeyShortcutsDisabled	0 enables new key shortcuts; 1 disables them
ExtendedEdit Keys	1 enables the full set of keyboard selection keys; 0 disables them
TrimLeadingZeros	1 trims leading zeroes in selections made by double-clicking; 0 keeps the leading zeros
WindowsAlpha	Sets opacity value between 30% and 100%. Use 0x4C to 0xFF or 76 to 255 to specify value
WordDelimiters	Defines the character that is used to skip to when selecting text a whole word at a time with CTRL+SHIFT+ARROW (the default is the space character). Set this REG_SZ value to contain all characters you want to be treated as delimiters. Note: this value is not stored in shortcuts, but only in this Registry key.

These settings are stored per each window title in the Registry under HKCU\Console. Console windows opened by a shortcut have these settings stored in the shortcut; if the shortcut is copied to another computer, the settings move with it to the new computer. Settings in shortcuts override all other settings, including global settings and defaults. However, if you revert to the original console by using **Use legacy console** in the **Options** tab, this

setting is global and will persist for all windows afterwards, including after restarting the computer.

You can pre-configure or script these settings by configuring the Registry appropriately in an Unattend file or with Windows PowerShell.

16-bit NTVDM apps always revert to the older console host.

> **NOTE**
> If you encounter issues with the new console settings and cannot resolve them with any of the specific options listed here, you can always revert back to the original console by setting ForceV2 to 0 or with the **Use legacy console** control in **Options**.

Console behavior

You can now resize the console window at will by grabbing an edge with the mouse and dragging it. Scroll bars only appear if you set window dimensions manually (by using the **Layout** tab in **Properties**) or if the longest line of text in the buffer is wider than the current window size.

The new console window now supports word wrapping. However, if you used console APIs to change text in a buffer, the console will leave the text as it was originally inserted.

Console windows now can be semi-transparent (to a minimum transparency of 30%). You can adjust the transparency from the Properties menu or with these keyboard commands:

TO DO THIS:	USE THIS KEY COMBINATION:
Increase transparency	CTRL+SHIFT+Plus (+) or CTRL+SHIFT+mouse scroll up
Decrease transparency	CTRL+SHIFT+Minus (-) or CTRL+SHIFT+mouse scroll down
Toggle full-screen mode	ALT+ENTER

Selection

There are many new options for selecting text and lines, as well as for marking text and using the buffer history. The console attempts to avoid clashes with applications that might be using the same keys.

For developers: If a conflict occurs, you can typically control the behavior of the application's use of line input, processed input, and echo input modes with the SetConsoleMode() API. If you run in processed input mode, the shortcuts below apply, but in other modes, your application must handle them. Any key combinations not listed here function as they did in previous versions of the console. You can also try to resolve conflicts with various settings on the **Options** tab. If all else fails, you can always revert to the original console.

You can now use "click-and-drag" selection outside of QuickEdit mode, and this selection can select text across lines as in Notepad, rather than just a rectangular block. Copy operations no longer require you to remove line breaks. In addition to "click-and-drag" selection, these key combinations are available:

Text selection

TO DO THIS:	USE THIS KEY COMBINATION:
Move the cursor to the left one character, extending the selection	SHIFT+LEFT ARROW

TO DO THIS:	USE THIS KEY COMBINATION:
Move the cursor to the right one character, extending the selection	SHIFT+RIGHT ARROW
Selects text line by line up from the insertion point	SHIFT+UP ARROW
Extends text selection down one line from the insertion point	SHIFT+DOWN ARROW
If the cursor is in the line currently being edited, use this command once to extend the selection to the last character in the input line. Use it a second time to extend the selection to the right margin.	SHIFT+END
If the cursor is **not** in the line currently being edited, use this command to select all text from the insertion point to the right margin.	SHIFT+END
If the cursor in the line currently being edited, use this command once to extend the selection to the character immediately after the command prompt. Use it a second time to extend the selection to the right margin.	SHIFT+HOME
If the cursor is **not** in the line currently being edited, use this command to extend the selection to the left margin.	SHIFT+HOME
Extend the selection down one screen	SHIFT+PAGE DOWN
Extend the selection up one screen	SHIFT+PAGE UP
Extend the selection one word to the right. (You can define the delimiters for "word" with the WordDelimiters registry key.)	CTRL+SHIFT+RIGHT ARROW
Extend the selection one word to the left	CTRL+SHIFT+HOME
Extend the selection to the beginning of the screen buffer	CTRL+SHIFT+END
Select all text after the prompt, if the cursor is in the current line and the line is not empty	CTRL+A
Select the entire buffer, if the cursor is **not** in the current line	CTRL+A

Editing text

You can copy and paste text in the console using keyboard commands. CTRL+C now serves two functions. If no text is selected when you use it, it sends the BREAK command as usual. If text is selected, the first use copies the text and clears the selection; the second use sends BREAK. Here are the other editing commands:

TO DO THIS:	USE THIS KEY COMBINATION:
Paste text into the command line	CTRL+V
Copy selected text to the clipboard	CTRL+INS
Copy selected text to the clipboard; send BREAK	CTRL+C

TO DO THIS:	USE THIS KEY COMBINATION:
Paste text into the command line	SHIFT+INS

Mark mode

To enter mark mode at any time, right-click anywhere in the console title bar, point to **Edit**, and select **Mark** from the menu that opens. You can also type CTRL+M. While in mark mode, use the ALT key to identify the start of a line-wrapping selection. (If **Enable line wrapping selection** is disabled, mark mode selects text in a block.) While in mark mode, CTRL+SHIFT+ARROW selects by character and not by word as in normal mode. In addition to the selection keys in the **Editing text** section, these combinations are available in mark mode:

TO DO THIS:	USE THIS KEY COMBINATION:
Enter mark mode to move cursor in the window	CTRL+M
Begin line-wrapping selection in mark mode, in conjunction with other key combinations	ALT
Move cursor in the direction specified	ARROW keys
Move cursor by one page in the direction specified	PAGE keys
Move cursor to beginning of buffer	CTRL+HOME
Move cursor to end of buffer	CTRL+END

Navigating history

TO DO THIS:	USE THIS KEY COMBINATION:
Move up one line in output history	CTRL+UP ARROW
Move down one line in output history	CTRL+DOWN ARROW
Move viewport to top of buffer (if command line is empty) or delete all characters to the left of the cursor (if command line is not empty)	CTRL+HOME
Move viewport to command line (if command line is empty) or delete all characters to the right of the cursor (if command line is not empty)	CTRL+END

Additional keyboard commands

TO DO THIS:	USE THIS KEY COMBINATION:
Open Find dialog	CTRL+F
Close console window	ALT+F4

Windows Server Semi-Annual Channel overview

12/18/2018 • 6 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel)

The Windows Server release model offers a new option in order to align with similar release and servicing models for Windows 10 and Office 365 ProPlus. If you've been working with Windows 10 or Office 365 ProPlus, these improvements might already be familiar to you.

There are two primary release channels available to Windows Server customers, the Long-Term Servicing Channel and the Semi-Annual Channel. You can keep servers on the Long-Term Servicing Channel (LTSC), move them to the Semi-Annual Channel, or have some servers on either track, depending on what works best for your needs.

Long-Term Servicing Channel (LTSC)

This is the release model you're already familiar with (formerly called the "Long-Term Servicing *Branch*") where a new major version of Windows Server is released every 2-3 years. Users are entitled to 5 years of mainstream support and 5 years of extended support. This channel is appropriate for systems that require a longer servicing option and functional stability. Deployments of Windows Server 2016 and earlier versions of Windows Server will not be affected by the new Semi-Annual Channel releases. The Long-Term Servicing Channel will continue to receive security and non-security updates, but it will not receive the new features and functionality.

> **NOTE**
>
> **The current LTSC product is Windows Server 2016**. If you want to stay in this channel, you should install (or continue using) Windows Server 2016, which can be installed in Server Core installation option or Server with Desktop Experience installation option. See Get Started with Windows Server 2016 for details.

Semi-Annual Channel

The Semi-Annual Channel is perfect for customer who are innovating quickly to take advantage of new operating system capabilities at a faster pace, both in applications – particularly those built on containers and microservices, as well as in the software-defined hybrid datacenter. Windows Server products in the Semi-Annual Channel will have new releases available twice a year, in spring and fall. Each release in this channel will be supported for 18 months from the initial release.

Most of the features introduced in the Semi-Annual Channel will be rolled up into the next Long-Term Servicing Channel release of Windows Server. The editions, functionality, and supporting content might vary from release to release depending on customer feedback.

The Semi-Annual Channel will be available to volume-licensed customers with Software Assurance, as well as via the Azure Marketplace or other cloud/hosting service providers and loyalty programs such as Visual Studio Subscriptions.

> **NOTE**
>
> **The current Semi-Annual Channel release is Windows Server, version 1803**. If you want to put servers in this channel, you should install Windows Server, version 1803, which can be installed in Server Core mode or as Nano Server run in a container. See Introducing Windows Server, version 1803 to learn how to obtain and activate Windows Server, version 1803. In-place upgrades from Windows Server 2016 to Windows Server, version 1803 are not supported because they are in **different release channels**. Windows Server, version 1803 is not an update to Windows Server 2016 – it is the next Windows Server release in the Semi-Annual Channel.

In this model, Windows Server releases are identified by the year and month of release: for example, in 2017, a release in the 9th month (September) would be identified as **version 1709**. Fresh releases of Windows Server in the Semi-Annual Channel will occur twice each year. The support lifecycle for each release is 18 months.

Should you keep servers on the LTSC or move them to the Semi-Annual Channel?

These are the key differences to take into account:

- Do you need to innovate rapidly? Do you need early access to the newest Windows Server features? Do you need to support fast-cadence hybrid applications, dev-ops, and Hyper-V fabrics? If so, you should consider **joining the Semi-Annual Channel** by installing Windows Server, version 1803. As described in this topic, you will receive new versions twice a year, with 18 months of mainstream production support per release. You get it through volume licensing, Azure, or Visual Studio Subscription Services. Currently, releases in the Semi-Annual Channel require volume licensing and Software Assurance if you intend to run the product in production.
- Do you need stability and predictability? Do you need to run virtual machines and traditional workloads on physical servers? If so, you should consider **keeping those servers on the Long-Term Servicing Channel**. The current LTSC release is Windows Server 2016. As described in this topic, you'll have access to new versions every 2-3 years, with 5 years of mainstream support followed by 5 years of extended support per release. LTSC releases are available through all release mechanisms. Releases in the LTSC are available to anyone regardless of the licensing model they are using.

The following table summarizes the key differences between the channels, starting with Windows Server, version 1803:

	LONG-TERM SERVICING CHANNEL (WINDOWS SERVER 2016)	SEMI-ANNUAL CHANNEL (WINDOWS SERVER)
Recommended scenarios	General purpose file servers, Microsoft and non-Microsoft workloads, traditional apps, infrastructure roles, software-defined Datacenter, and hyper-converged infrastructure	Containerized applications, container hosts, and application scenarios benefiting from faster innovation
New releases	Every 2–3 years	Every 6 months
Support	5 years of mainstream support, plus 5 years of extended support	18 months
Editions	All available Windows Server editions	Standard and Datacenter editions
Who can use	All customers through all channels	Software Assurance and cloud customers only

	LONG-TERM SERVICING CHANNEL (WINDOWS SERVER 2016)	SEMI-ANNUAL CHANNEL (WINDOWS SERVER)
Installation options	Server Core and Server with Desktop Experience	Server Core for container host and image and Nano Server container image

Device compatibility

Unless otherwise communicated, the minimum hardware requirements to run the Semi-Annual Channel releases will be the same as the most recent Long-Term Servicing Channel release of Windows Server. For example, **the current Long-Term Servicing Channel release is Windows Server 2016**. Most hardware drivers will continue to function in these releases.

Servicing

Both the Long-Term Servicing Channel and the Semi-Annual Channel releases will be supported with security updates and non-security updates. The difference is the length of time that the release is supported, as described above.

Servicing tools

There are many tools with which IT pros can service Windows Server. Each option has its pros and cons, ranging from capabilities and control to simplicity and low administrative requirements. The following are examples of the servicing tools available to manage servicing updates:

- **Windows Update (stand-alone)**: This option is only available for servers that are connected to the Internet and have Windows Update enabled.
- **Windows Server Update Services (WSUS)** provides extensive control over Windows 10 and Windows Server updates and is natively available in the Windows Server operating system. In addition to the ability to defer updates, organizations can add an approval layer for updates and choose to deploy them to specific computers or groups of computers whenever ready.
- **System Center Configuration Manager** provides the greatest control over servicing. IT pros can defer updates, approve them, and have multiple options for targeting deployments and managing bandwidth usage and deployment times.

You've likely already chosen to use at least one of these options based on your resources, staff, and expertise. You can continue using the same process for Semi-Annual Channel Releases: for example, if you already use System Center Configuration Manager to manage updates, you can continue to use it. Similarly, if you are using WSUS, you can continue to use that.

Obtain preview releases through the Windows Insider Program

Testing the early builds of Windows Server helps both Microsoft and its customers because of the opportunity to discover possible issues before release. It also gives customers a unique opportunity to directly influence the functionality in the product.

Microsoft depends on receiving feedback throughout the development process so that adjustments may be made as quickly as possible. Early testing and feedback is essential to the rapid release model.

For more information about how to get involved with the Windows Insider Program, see the Windows Insider Program for Server docs.

Related topics

Windows Server 2019 servicing channels: LTSC and SAC

Changes to Nano Server in Windows Server Semi-Annual Channel

Windows Server support lifecycle

Windows Server 2016 System Requirements

Windows Server Installation and Upgrade

11/6/2018 • 8 minutes to read • Edit Online

Applies To: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, Windows Server 2008

IMPORTANT

Extended support for Windows Server 2008 R2 and Windows Server 2008 ends in January 2020. Learn about your upgrade options.

Is it time to move to a newer version of Windows Server? Depending on what you are running now, you have lots of options to get there.

Installation

If you want to move to a newer version of Windows Server on the same hardware, one way that always works is a **clean installation**, where you just install the newer operating system directly over the old one on the same hardware, thus deleting the previous operating system. That is the simplest way, but you will need to back up your data first and plan to reinstall your applications. There are a few things to be aware of, such as system requirements, so be sure to check the details for Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012.

Moving from any pre-release version (such as Windows Server 2016 Technical Preview) to the released version (Windows Server 2016) always requires a clean installation.

Migration (recommended for Windows Server 2016)

Windows Server [migration] documentation helps you migrate one role or feature at a time from a source computer that is running Windows Server to another destination computer that is running Windows Server, either the same or a newer version. For these purposes, migration is defined as moving one role or feature and its data to a different computer, not upgrading the feature on the same computer. This is the recommended manner in which to move your existing workload and data to a more recent version of Windows Server. To get started, check the server role upgrade and migration matrix for Windows Server 2016.

Cluster OS Rolling Upgrade

Cluster OS Rolling Upgrade is a new feature in Windows Server 2016 that enables an administrator to upgrade the operating system of the cluster nodes from Windows Server 2012 R2 to Windows Server 2016 without stopping the Hyper-V or the Scale-Out File Server workloads. This feature allows you to avoid downtime which could impact Service Level Agreements. This new feature is discussed in more detail at Cluster operating system rolling upgrade.

License Conversion

In some operating system releases, you can convert a particular edition of the release to another edition of the same release in a single step with a simple command and the appropriate license key. This is called **license conversion**. For example, if your server is running Windows Server 2016 Standard, you can convert it to Windows Server 2016 Datacenter. In some releases of Windows Server, you can also freely convert among OEM, volume-licensed, and retail versions with the same command and the appropriate key.

Upgrade

If you want to keep the same hardware and all the server roles you have set up without flattening the server, **upgrading** is an option—and there are lots of ways to do it. In the classic upgrade, you go from an older operating system to a newer one, keeping your settings, server roles, and data intact. For example, if your server is running Windows Server 2012 R2, you can upgrade it to Windows Server 2016. However, not every older operating system has a pathway to every newer one.

> **NOTE**
> Upgrade works best in virtual machines where specific OEM hardware drivers are not needed for a successful upgrade.

You can upgrade from an evaluation version of the operating system to a retail version, from an older retail version to a newer version, or, in some cases, from a volume-licensed edition of the operating system to an ordinary retail edition.

Before you get started with an upgrade, have a look at the tables on this page to see how to get from where you are to where you want to be.

For information about the differences between the installation options available for Windows Server 2016 Technical Preview, including the features that are installed with each option and the management options available after installation, see Windows Server 2016.

> **NOTE**
> Whenever you migrate or upgrade to any version of Windows Server, you should review and understand the support lifecycle policy and timeframe for that version and plan accordingly. You can search for the lifecycle information for the particular Windows Server release that you are interested in.

Upgrading to Windows Server 2016

For details, including important caveats and limitations on upgrade, license conversion between editions of Windows Server 2016, and conversion of evaluation editions to retail, see Supported Upgrade Paths for Windows Server 2016.

> **NOTE**
> Note: Upgrades that switch from a Server Core installation to a Server with a Desktop installation (or vice versa) are not supported. If the older operating system you are upgrading or converting is a Server Core installation, the result will still be a Server Core installation of the newer operating system.

Quick reference table of supported upgrade paths from older Windows Server retail editions to Windows Server 2016 retail editions:

IF YOU ARE RUNNING THESE VERSIONS AND EDITIONS:	YOU CAN UPGRADE TO THESE VERSIONS AND EDITIONS:
Windows Server 2012 Standard	Windows Server 2016 Standard or Datacenter
Windows Server 2012 Datacenter	Windows Server 2016 Datacenter
Windows Server 2012 R2 Standard	Windows Server 2016 Standard or Datacenter

IF YOU ARE RUNNING THESE VERSIONS AND EDITIONS:	YOU CAN UPGRADE TO THESE VERSIONS AND EDITIONS:
Windows Server 2012 R2 Datacenter	Windows Server 2016 Datacenter
Hyper-V Server 2012 R2	Hyper-V Server 2016 (using Cluster OS Rolling Upgrade feature)
Windows Server 2012 R2 Essentials	Windows Server 2016 Essentials
Windows Storage Server 2012 Standard	Windows Storage Server 2016 Standard
Windows Storage Server 2012 Workgroup	Windows Storage Server 2016 Workgroup
Windows Storage Server 2012 R2 Standard	Windows Storage Server 2016 Standard
Windows Storage Server 2012 R2 Workgroup	Windows Storage Server 2016 Workgroup

License conversion

You can convert Windows Server 2016 Standard (retail) to Windows Server 2016 Datacenter (retail).

You can convert Windows Server 2016 Essentials (retail) to Windows Server 2016 Standard (retail).

You can convert the evaluation version of Windows Server 2016 Standard to either Windows Server 2016 Standard (retail) or Datacenter (retail).

You can convert the evaluation version of Windows Server 2016 Datacenter to Windows Server 2016 Datacenter (retail).

Upgrading to Windows Server 2012 R2

For details, including important caveats and limitations on upgrade, license conversion between editions of Windows Server 2012 R2, and conversion of evaluation editions to retail, see Upgrade Options for Windows Server 2012 R2.

Quick reference table of supported upgrade paths from older Windows Server retail editions to Windows Server 2012 R2 retail editions:

IF YOU ARE RUNNING:	YOU CAN UPGRADE TO THESE EDITIONS:
Windows Server 2008 R2 Datacenter with SP1	Windows Server 2012 R2 Datacenter
Windows Server 2008 R2 Enterprise with SP1	Windows Server 2012 R2 Standard or Windows Server 2012 R2 Datacenter
Windows Server 2008 R2 Standard with SP1	Windows Server 2012 R2 Standard or Windows Server 2012 R2 Datacenter
Windows Web Server 2008 R2 with SP1	Windows Server 2012 R2 Standard
Windows Server 2012 Datacenter	Windows Server 2012 R2 Datacenter
Windows Server 2012 Standard	Windows Server 2012 R2 Standard or Windows Server 2012 R2 Datacenter

IF YOU ARE RUNNING:	YOU CAN UPGRADE TO THESE EDITIONS:
Hyper-V Server 2012	Hyper-V Server 2012 R2

License conversion

You can convert Windows Server 2012 Standard (retail) to Windows Server 2012 Datacenter (retail).

You can convert Windows Server 2012 Essentials (retail) to Windows Server 2012 Standard (retail).

You can convert the evaluation version of Windows Server 2012 Standard to either Windows Server 2012 Standard (retail) or Datacenter (retail).

Upgrading to Windows Server 2012

For details, including important caveats and limitations on upgrade, and conversion of evaluation editions to retail, see Evaluation Versions and Upgrade Options for Windows Server 2012.

Quick reference table of supported upgrade paths from older Windows Server retail editions to Windows Server 2012 retail editions:

IF YOU ARE RUNNING:	YOU CAN UPGRADE TO THESE EDITIONS:
Windows Server 2008 Standard with SP2 or Windows Server 2008 Enterprise with SP2	Windows Server 2012 Standard, Windows Server 2012 Datacenter
Windows Server 2008 Datacenter with SP2	Windows Server 2012 Datacenter
Windows Web Server 2008	Windows Server 2012 Standard
Windows Server 2008 R2 Standard with SP1 or Windows Server 2008 R2 Enterprise with SP1	Windows Server 2012 Standard, Windows Server 2012 Datacenter
Windows Server 2008 R2 Datacenter with SP1	Windows Server 2012 Datacenter
Windows Web Server 2008 R2	Windows Server 2012 Standard

License conversion

You can convert Windows Server 2012 Standard (retail) to Windows Server 2012 Datacenter (retail).

You can convert Windows Server 2012 Essentials (retail) to Windows Server 2012 Standard (retail).

You can convert the evaluation version of Windows Server 2012 Standard to either Windows Server 2012 Standard (retail) or Datacenter (retail).

Upgrading from Windows Server 2008 R2 or Windows Server 2008

As described in Upgrade Windows Server 2008 and Windows Server 2008 R2, the extended support for Windows Server 2008 R2/Windows Server 2008 ends in January of 2020. To ensure no gap in support, you need to upgrade to a supported version of Windows Server, or rehost in Azure by moving to specialized Windows Server 2008 R2 VMs. Check out the Migration Guide for Windows Server for information and considerations for planning your migration/upgrade.

For on-premises servers, there is no direct upgrade path from Windows Server 2008 R2 to Windows Server 2016 or later. Instead, upgrade first to Windows Server 2012 R2, and then upgrade to Windows Server 2016.

As you are planning your upgrade, be aware of the following guidelines for the middle step of upgrading to Windows Server 2012 R2.

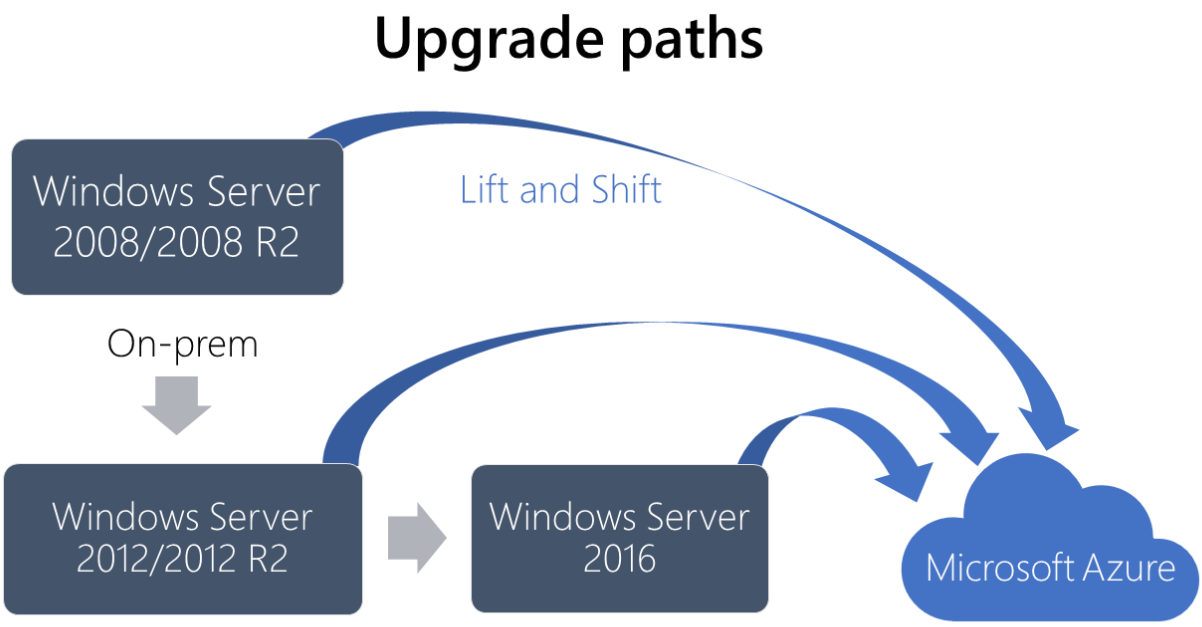
- You can't do an in-place upgrade from a 32-bit to 64-bit architectures or from one build type to another (fre to chk, for example).
- In-place upgrades are only supported in the same language. You can't upgrade from one language to another.
- You can't migrate from a Windows Server 2008 server core installation to Windows Server 2012 R2 with the Server GUI (called "Server with Full Desktop" in Windows Server). You can switch your upgraded server core installation to Server with Full Desktop, but only on Windows Server 2012 R2. Windows Server 2016 and later *do not* support switching from server core to Full Desktop, so make that switch before you upgrade to Windows Server 2016.

For more information, check out Evaluation Versions and Upgrade Options for Windows Server 2012, which includes role-specific upgrade details.

Upgrade Windows Server 2008 and Windows Server 2008 R2

10/3/2018 • 2 minutes to read • Edit Online

Extended support for Windows Server 2008 and Windows Server 2008 R2 is ending on January 14, 2020. There are two modernization paths available: On-premises upgrade, or migration by rehosting in Azure. **If you rehost in Azure, you can migrate your existing Server images free of charge.**



On-premises upgrade

If you need to keep your servers on-premises, and you are running Windows Server 2008 or Windows Server 2008 R2, you will need to upgrade to Windows Server 2012/2012 R2 before you can upgrade to Windows Server 2016. As you upgrade, you still have the option to migrate to Azure by rehosting.

See Upgrading from Windows Server 2008 R2 or Windows Server 2008, for more information about your on-premises upgrade options.

If you are running Windows Server 2003, you will need to upgrade to Windows Server 2008. See upgrade paths for Windows Server 2008 for more information about your on-premises upgrade options.

Migrate to Azure

You can migrate your on-premises Windows Server 2008 and Windows Server 2008 R2 servers to Azure, where you can continue to run them on virtual machines. In Azure you'll stay compliant, become more secure, and add cloud innovation to your work. The benefits of migrating to Azure include:

- Security updates in Azure.
- Get three more years of Windows Server 2008 R2 or 2008 critical and important security updates, included at no additional charge.
- No-charge upgrades in Azure.

- Adopt more cloud services as you are ready.
- By migrating SQL Server to Azure Managed Instances or VMs, you get three more years of Windows Server 2008 R2 or 2008 critical security updates, included at no additional charge.
- Leverage existing SQL Server and Windows Server licenses for cloud savings unique to Azure.

To get started migrating, see Upload a Windows Server 2008/2008 R2 specialized image to Azure.

To help you understand how to analyze existing IT resources, assess what you have, and identify the benefits of moving specific services and applications to the cloud or keeping workloads on-premises and upgrading to the latest version of Windows Server, see Migration Guide for Windows Server.

Upgrade SQL Server 2008/2008 R2 in parallel with your Windows Servers

If you are running SQL Server 2008/2008 R2, you can upgrade to SQL Server 2016 or 2017.

Additional resources

Microsoft Azure

Upload a Windows Server 2008/2008 R2 specialized image to Azure

11/15/2018 • 2 minutes to read • Edit Online

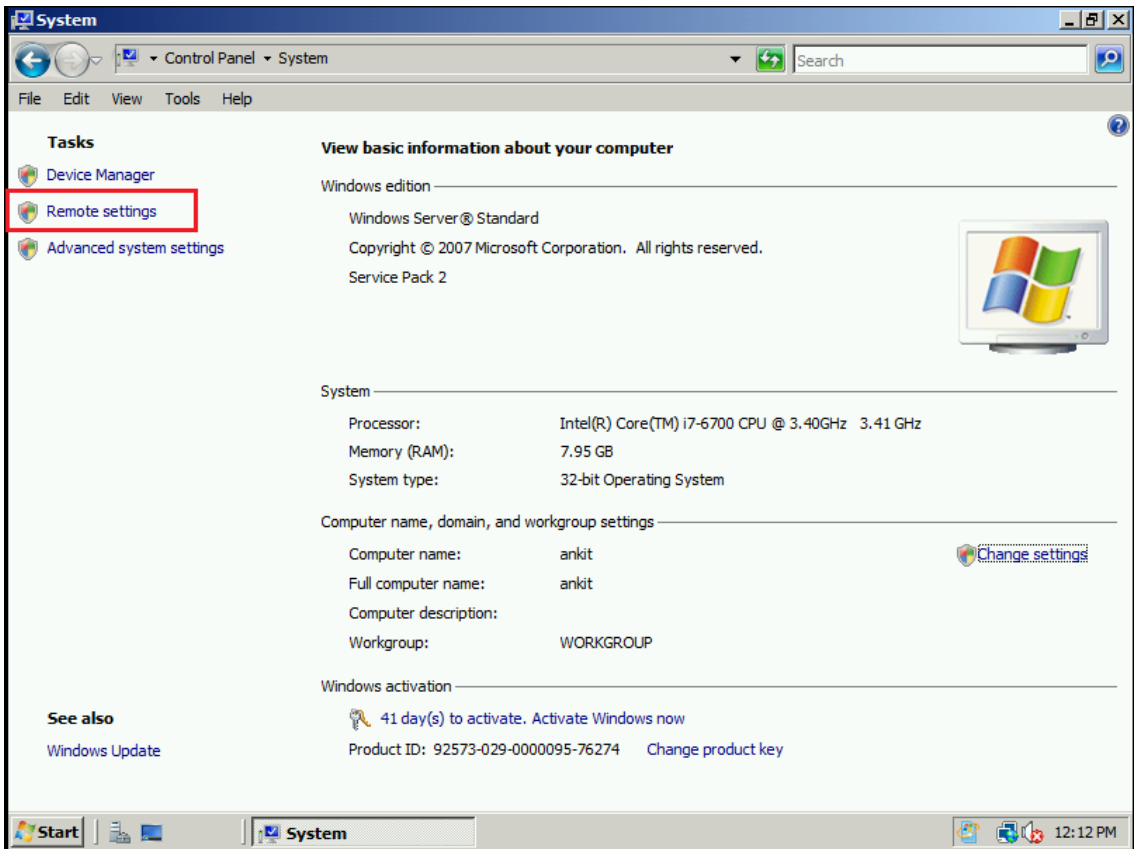
You can now run a Windows Server 2008/2008 R2 VM in the cloud with Azure.

Prep the Windows Server 2008/2008 R2 specialized image

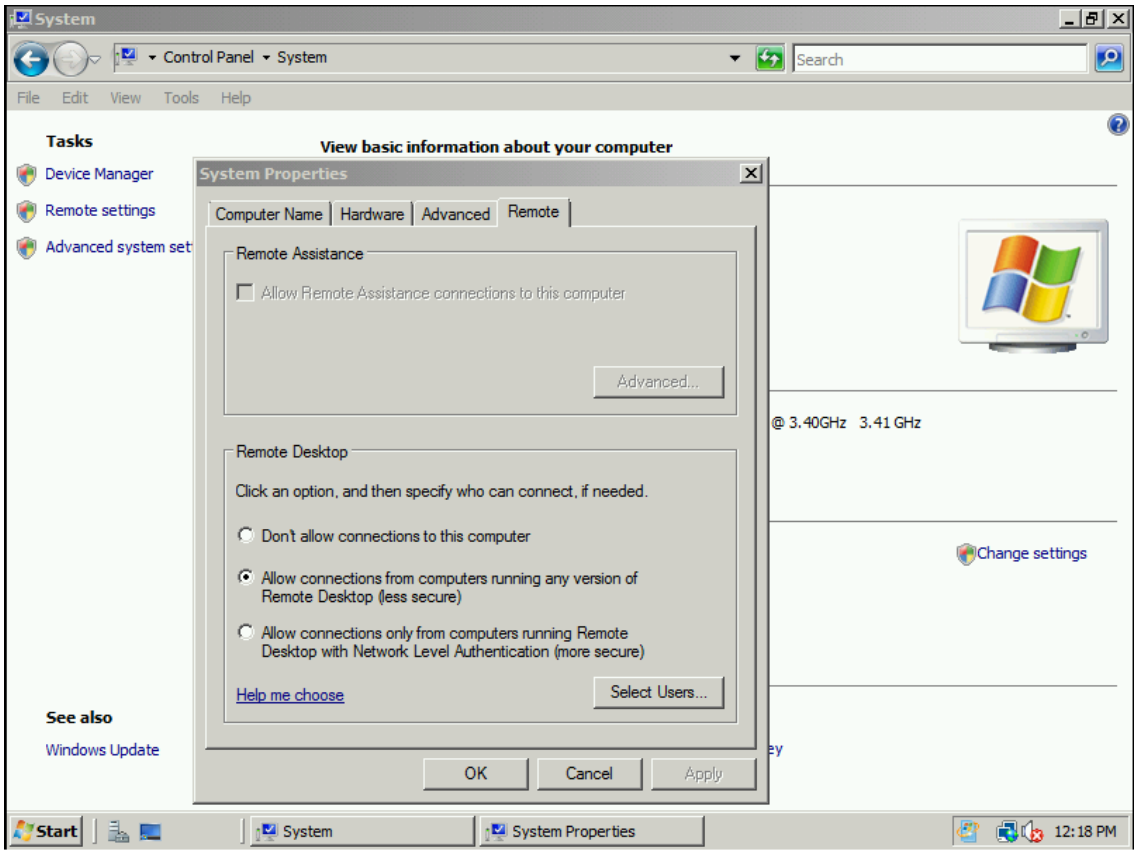
Before you can upload an image, make the following changes:

- Download and install Windows Server 2008 Service Pack 2 (SP2) if you don't already have it installed on your image.
- Configure Remote Desktop (RDP) settings.

  1. Go to **Control Panel** > **System settings**.
  2. Select **Remote settings** in the left-hand menu.



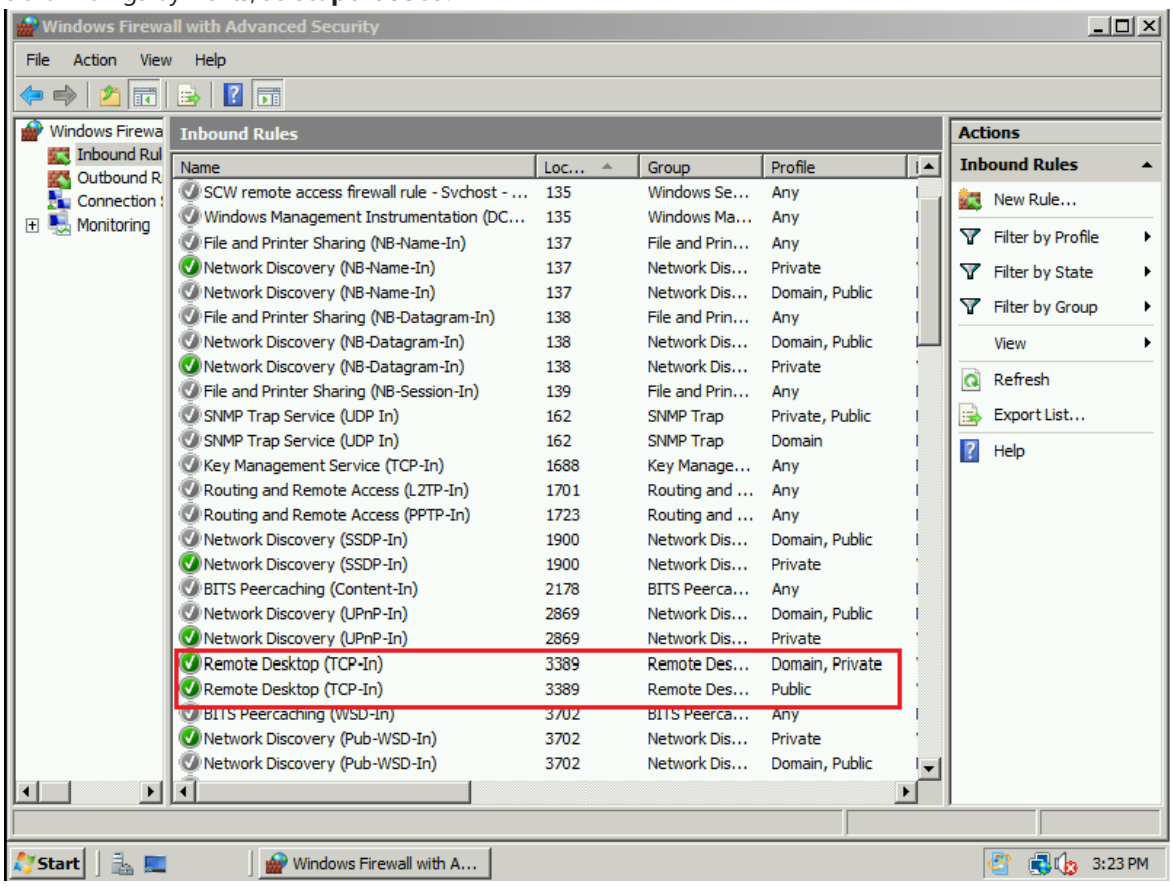
  3. Select the **Remote** tab in System Properties.



4. Select Allow connections from computers running any version of Remote Desktop (less secure).
5. Click **Apply**, and **OK**.

- Configure Windows Firewall settings.

1. At the command prompt in Admin mode, enter "**wf.msc**" for Windows Firewall and advanced security settings.
2. Sort findings by **Ports**, select **port 3389**.

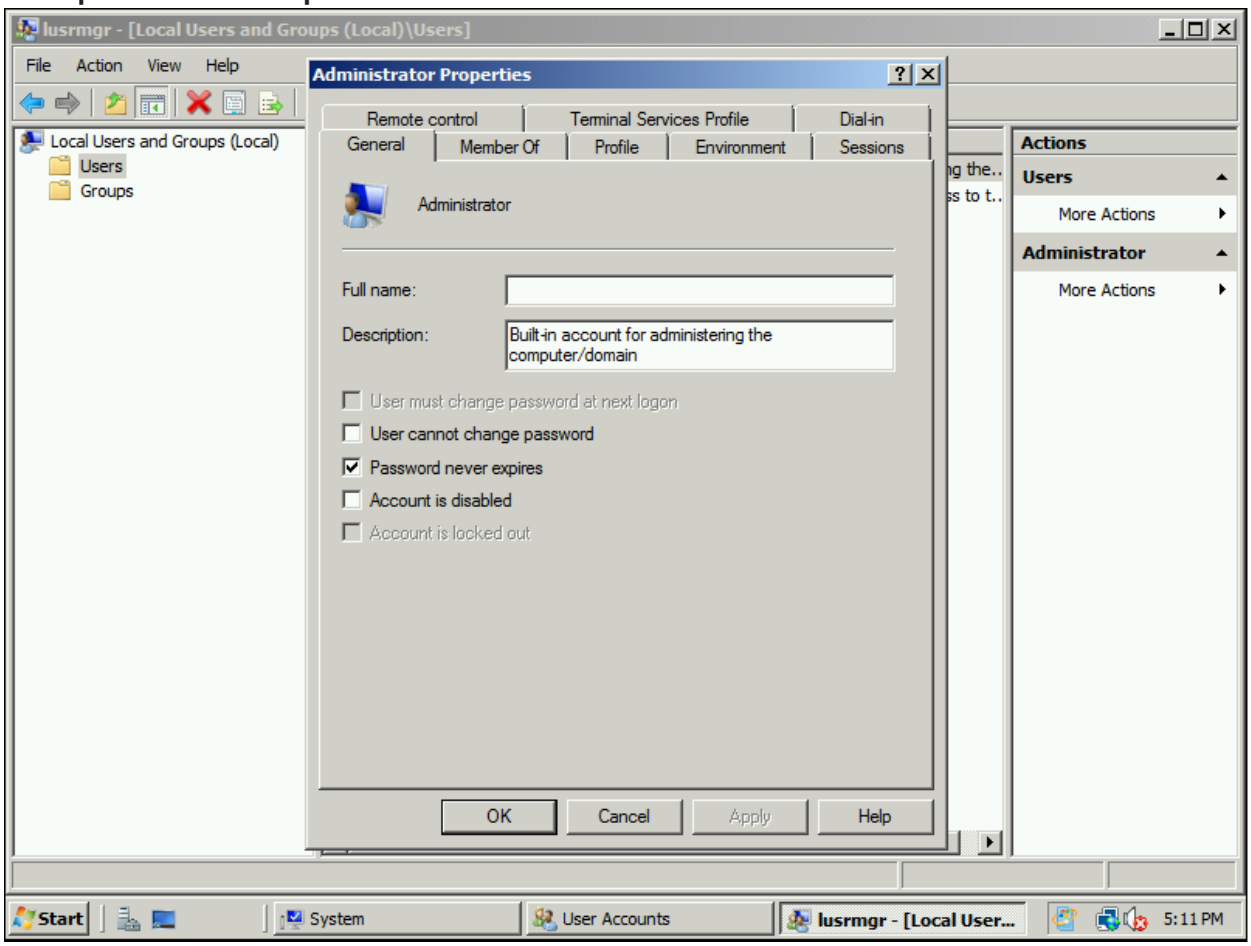


3. Enable Remote Desktop (TCP-IN) for the profiles: **Domain**, **Private**, and **Public** (shown above).

- Save all settings and shut down the image.
- If you're using Hyper-V, make sure the child AVHD is merged into the parent VHD for persisting changes.

A current known bug causes the administrator password on the uploaded image to expire within 24 hours. Follow these steps to avoid this issue:

1. Go to **Start** > **Run**
2. Type **lusrmgr.msc**
3. Select **Users** under Local Users and Groups
4. Right-click **Administrator** and select **Properties**
5. Select **password never expires** and select **OK**



Uploading the image VHD

You can use the script below to upload the VHD. Before you do this, you'll need the publish settings file for your Azure account. Get your Azure file settings.

Here is the script:

```
Get-AzurePublishSettingsFile

Login-AzureRmAccount

      # Import publishsettings
      Import-AzurePublishSettingsFile -PublishSettingsFile <LocationOfPublishingFile>
      $subscriptionId = 'xxxx-xxxx-xxxx-xxxx-xxxxx'

      # Set NodeFlight subscription as default subscription
      Select-AzureRmSubscription -SubscriptionId $subscriptionId
      Set-AzureRmContext -SubscriptionId $subscriptionId
      $rgName = "<resourcegroupname>"

      $urlOfUploadedImageVhd = "<BlobUrl>/<NameForVHD>.vhd"
      Add-AzureRmVhd -ResourceGroupName $rgName -Destination $urlOfUploadedImageVhd -LocalFilePath "<FilePath>"
```

Deploy the image in Azure

In this section, you will be deploying the image VHD in Azure.

> **IMPORTANT**
> Do not use pre-defined user images in Azure.

1. Create a new resource group.
2. Create a new storage blob inside the resource group.
3. Create a container inside the storage blob.
4. Copy the URL of the blob storage from properties.
5. Use the script provided above to upload your image to the new storage blob.
6. Create a disk for your VHD.
   a. Go to Disks, click **Add**.
   b. Enter a name for the disk. Select the subscription you want to use, set the region, and choose the account type.
   c. For Source Type, select storage. Browse to the blob VHD location created using the script.
   d. Select OS type Windows and Size (default: 1023).
   e. Click **Create**.
7. Go to the Disk Created, click **Create VM**.
   a. Name the VM.
   b. Select the existing group you created in step 5 where you uploaded the disk.
   c. Pick a size and SKU plan for your VM.
   d. Select a network interface on the settings page. Make sure the network interface has the following rule specified:

   ```
   PORT:3389 Protocol: TCP Action: Allow Priority: 1000 Name: ‘RDP-Rule’.
   ```

   e. Click **Create**.

Migrating Roles and Features in Windows Server

11/6/2018 • 3 minutes to read • Edit Online

Applies To: Windows Server 2016, Windows Server 2012 R2, Windows Server 2012

This page contains links to information and tools that help guide you through the process of migrating roles and features to Windows Server 2016, Windows Server 2012 R2, and Windows Server 2012. Many roles and features can be migrated by using Windows Server Migration Tools, a set of five Windows PowerShell cmdlets that was introduced in Windows Server 2008 R2 for easily migrating role and feature elements and data.

The migration guides support migrations of specified roles and features from one server to another (not in-place upgrades). Unless otherwise noted in the guides, migrations are supported between physical and virtual computers, and between full installation options of Windows Server and servers that are running the Server Core installation option.

Before you begin

Before you begin migrating roles and features, verify that both source and destination servers are running the most current service packs that are available for their operating systems. An e-book of Windows Server 2012 R2 and Windows Server 2012 migration guides is now available. For more information, and to download the e-book, see the E-Book Gallery for Microsoft Technologies.

> **NOTE**
> Whenever you migrate or upgrade to any version of Windows Server, you should review and understand the support lifecycle policy and timeframe for that version and plan accordingly. You can search for the lifecycle information for the particular Windows Server release that you are interested in.

Windows Server 2016

Migration Guides

Updated migration guides for Windows Server 2016 are under development. Check back at this location for updates as they become available. In many cases, the steps in the Windows Server 2012 R2 migration guides are still relevant for Windows Server 2016.

- Remote Desktop Services
- Web Server (IIS)
- Windows Server Update Services
- MultiPoint Services

Windows Server 2012 R2

Migration Guides

Follow the steps in these guides to migrate roles and features from servers that are running Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, or Windows Server 2012 R2 to Windows Server 2012 R2. Windows Server Migration Tools in Windows Server 2012 R2 supports cross-subnet migrations.

- Install, Use, and Remove Windows Server Migration Tools

- Active Directory Certificate Services Migration Guide for Windows Server 2012 R2
- Migrating Active Directory Federation Services Role Service to Windows Server 2012 R2
- Active Directory Rights Management Services Migration and Upgrade Guide
- Migrate File and Storage Services to Windows Server 2012 R2
- Migrate Hyper-V to Windows Server 2012 R2 from Windows Server 2012
- Migrate Network Policy Server to Windows Server 2012
- Migrate Remote Desktop Services to Windows Server 2012 R2
- Migrate Windows Server Update Services to Windows Server 2012 R2
- Migrate Cluster Roles to Windows Server 2012 R2
- Migrate DHCP Server to Windows Server 2012 R2

Windows Server 2012

Migration Guides

Follow the steps in these guides to migrate roles and features from servers that are running Windows Server 2003, Windows Server 2008, Windows Server 2008 R2, or Windows Server 2012 to Windows Server 2012. Windows Server Migration Tools in Windows Server 2012 supports cross-subnet migrations.

- Install, Use, and Remove Windows Server Migration Tools
- Migrate Active Directory Federation Services Role Services to Windows Server 2012
- Migrate Health Registration Authority to Windows Server 2012
- Migrate Hyper-V to Windows Server 2012 from Windows Server 2008 R2
- Migrate IP Configuration to Windows Server 2012
- Migrate Network Policy Server to Windows Server 2012
- Migrate Print and Document Services to Windows Server 2012
- Migrate Remote Access to Windows Server 2012
- Migrate Windows Server Update Services to Windows Server 2012
- Upgrade Active Directory Domain Controllers to Windows Server 2012
- Migrating Clustered Services and Applications to Windows Server 2012

For additional migration resources, visit Migrate Roles and Features to Windows Server 2012.

Windows Server 2008 R2

Migration Guides

Follow the steps in these guides to migrate roles and features from servers that are running Windows Server 2003, Windows Server 2008, or Windows Server 2008 R2 to Windows Server 2008 R2. Windows Server Migration Tools in Windows Server 2008 R2 does not support cross-subnet migrations.

- Windows Server Migration Tools Installation, Access, and Removal
- Active Directory Certificate Services Migration Guide
- Active Directory Domain Services and Domain Name System (DNS) Server Migration Guide
- BranchCache Migration Guide
- Dynamic Host Configuration Protocol (DHCP) Server Migration Guide
- File Services Migration Guide
- HRA Migration Guide
- Hyper-V Migration Guide
- IP Configuration Migration Guide
- Local User and Group Migration Guide

- NPS Migration Guide
- Print Services Migration Guide
- Remote Desktop Services Migration Guide
- RRAS Migration Guide
- Windows Server Migration Common Tasks and Information
- Windows Server Update Services 3.0 SP2 Migration Guide

For additional migration resources, visit Migrate Roles and Features to Windows Server 2008 R2.

System Requirements

10/29/2018 • 4 minutes to read • Edit Online

Applies To: Windows Server (Semi-Annual Channel), Windows Server 2016

This topic addresses the minimum system requirements to run Windows Server® 2016 or Windows Server, version 1709.

NOTE

In this release, clean installations are recommended.

NOTE

If at the time of installation, you choose to install with the Server Core option, you should be aware that no GUI components are installed at all and you will not be able to install or uninstall them with Server Manager. If you need GUI features, be sure to choose the "Server with Desktop Experience" option when you install Windows Server 2016. For more information, see Install Nano Server

Review system requirements

The following are estimated system requirements Windows Server 2016. If your computer has less than the "minimum" requirements, you will not be able to install this product correctly. Actual requirements will vary based on your system configuration and the applications and features you install.

Unless otherwise specified, these minimum system requirements apply to all installation options (Server Core, Server with Desktop Experience, and Nano Server) and both Standard and Datacenter editions.

IMPORTANT

The highly diverse scope of potential deployments makes it unrealistic to state "recommended" system requirements that would be generally applicable. Consult documentation for each of the server roles you intend to deploy for more details about the resource needs of particular server roles. For the best results, conduct test deployments to determine appropriate system requirements for your particular deployment scenarios.

Processor

Processor performance depends not only on the clock frequency of the processor, but also on the number of processor cores and the size of the processor cache. The following are the processor requirements for this product:

Minimum:

- 1.4 GHz 64-bit processor
- Compatible with x64 instruction set
- Supports NX and DEP
- Supports CMPXCHG16b, LAHF/SAHF, and PrefetchW
- Supports Second Level Address Translation (EPT or NPT)

Coreinfo is a tool you can use to confirm which of these capabilities your CPU has.

RAM

The following are the estimated RAM requirements for this product:

Minimum:

- 512 MB (2 GB for Server with Desktop Experience installation option)
- ECC (Error Correcting Code) type or similar technology

> **IMPORTANT**
>
> If you create a virtual machine with the minimum supported hardware parameters (1 processor core and 512 MB RAM) and then attempt to install this release on the virtual machine, Setup will fail.
>
> To avoid this, do one of the following:
>
> - Allocate more than 800 MB RAM to the virtual machine you intend to install this release on. Once Setup has completed, you can change the allocation to as little as 512 MB RAM, depending on the actual server configuration.
> - Interrupt the boot process of this release on the virtual machine with SHIFT+F10. In the command prompt that opens, use Diskpart.exe to create and format an installation partition. Run **Wpeutil createpagefile /path=C:\pf.sys** (assuming the installation partition you created was C:). Close the command prompt and proceed with Setup.

Storage controller and disk space requirements

Computers that run Windows Server 2016 must include a storage adapter that is compliant with the PCI Express architecture specification. Persistent storage devices on servers classified as hard disk drives must not be PATA. Windows Server 2016 does not allow ATA/PATA/IDE/EIDE for boot, page, or data drives.

The following are the estimated **minimum** disk space requirements for the system partition.

Minimum: 32 GB

> **NOTE**
>
> Be aware that 32 GB should be considered an *absolute minimum* value for successful installation. This minimum should allow you to install Windows Server 2016 in Server Core mode, with the Web Services (IIS) server role. A server in Server Core mode is about 4 GB smaller than the same server in Server with a GUI mode.
>
> The system partition will need extra space for any of the following circumstances:
>
> - If you install the system over a network.
> - Computers with more than 16 GB of RAM will require more disk space for paging, hibernation, and dump files.

Network adapter requirements

Network adapters used with this release should include these features:

Minimum:

- An Ethernet adapter capable of at least gigabit throughput
- Compliant with the PCI Express architecture specification.
- Supports Pre-boot Execution Environment (PXE).

A network adapter that supports network debugging (KDNet) is useful, but not a minimum requirement.

Other requirements

Computers running this release also must have the following:

- DVD drive (if you intend to install the operating system from DVD media)

The following items are not strictly required, but are necessary for certain features:

- UEFI 2.3.1c-based system and firmware that supports secure boot
- Trusted Platform Module
- Graphics device and monitor capable of Super VGA (1024 x 768) or higher-resolution
- Keyboard and Microsoft® mouse (or other compatible pointing device)
- Internet access (fees may apply)

> **NOTE**
> A Trusted Platform Module (TPM) chip is not strictly required to install this release, though it is necessary in order to use certain features such as BitLocker Drive Encryption. If your computer uses TPM, it must meet these requirements:
>
> - Hardware-based TPMs must implement version 2.0 of the TPM specification.
> - TPMs that implement version 2.0 must have an EK certificate that is either pre-provisioned to the TPM by the hardware vendor or be capable of being retrieved by the device during the first boot.
> - TPMs that implement version 2.0 must ship with SHA-256 PCR banks and implement PCRs 0 through 23 for SHA-256. It is acceptable to ship TPMs with a single switchable PCR bank that can be used for both SHA-1 and SHA-256 measurements.
> - A UEFI option to turn off the TPM is not a requirement.

Installation of Nano Server

For detailed steps to install Windows Server 2016 as a Nano Server, see Install Nano Server.

Additional Resources

- Windows Processor Requirements
- Comparison of Standard and Datacenter editions of Windows Server 2016
- Windows 10 System Requirements
- Download the Windows Server 2016 licensing datasheet

Release Notes: Important Issues in Windows Server, version 1803

10/3/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server Semi-Annual Channel

These release notes summarize the most critical issues in the Windows Server operating system, including ways to avoid or work around known issues. To learn about new features in this release, see What's New in Windows Server version 1803. Check out About Windows containers if you're interested in running a Windows Server, version 1803, container.

Unless otherwise specified, each reported issue applies to all editions and installation options of Windows Server, version 1803.

We continuously update this article. If any known issues are identified, we'll document them here.

Software-defined datacenter

Software-defined datacenter features like Storage Spaces Direct, software-defined networking, and shielded virtual machines aren't included in Windows Server, version 1803. As described in Windows Server Semi-Annual Channel update, the Semi-Annual Channel of Windows Server is focused on containers and application scenarios that benefit from faster innovation.

If you need infrastructure roles, use a Long-Term Servicing Channel release: Windows Server 2016 (available now) or Windows Server 2019 (coming later this year).

We're committed to building the best platform for hyper-converged infrastructure, and we continue to develop new features and improve existing ones based on your feedback. Thank you for helping us get to 10,000+ clusters of Storage Spaces Direct! We look forward to sharing more later this year.

Release Notes: Important Issues in Windows Server, version 1709

9/21/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server Semi-Annual Channel

These release notes summarize the most critical issues in the Windows Server® operating system, including ways to avoid or work around the issues, if known. For information about by-design changes, new features, and fixes in this release, see What's New in Windows Server version 1709 and announcements from the specific feature teams. Unless otherwise specified, each reported issue applies to all editions and installation options of Windows Server 2016.

This document is continuously updated. As critical issues requiring a workaround are discovered, they are added, as are new workarounds and fixes as they become available.

Storage Spaces Direct

Storage Spaces Direct is not included in Windows Server, version 1709. If you call *Enable-ClusterStorageSpacesDirect* or its alias *Enable-ClusterS2D*, on a server running Windows Server, version 1709, you will receive an error with the message "The requested operation is not supported".

It is also not supported to introduce servers running Windows Server, version 1709 into a Windows Server 2016 Storage Spaces Direct deployment.

The Windows Server release model is offering a new option in order to align with similar release and servicing models for Windows 10 and Office 365 ProPlus. The Semi-Annual Channel releases deliver new functionality for customers who want to move at a rapid cadence and will have new releases available twice a year, in spring and fall.

The Semi-Annual Channel of Windows Server is focused on containers and application scenarios benefiting from faster innovation, see this blog for additional information. Customers looking for infrastructure roles, such as Storage Spaces Direct, should use Long-Term Servicing Channel releases like Windows Server 2016 (available now) and Windows Server 2019 (coming later this year). We are committed to building the best platform for hyper-converged infrastructure, and we continue to develop new features and improve existing ones based on your feedback.

Storage Spaces Direct was introduced in Windows Server 2016 and is the foundation for our hyper-converged platform. We have been thrilled by the positive adoption of the Microsoft hyper-converged platform and we are committed to our customers.

We've been listening to your feedback and are working to deliver the next set of innovations for our hyper-converged platform. These features are available today in Windows Insiders builds, and we would love for you to try them and share your feedback. For customers looking for a validated hyper-converged solution, we recommend the Windows Server Software Defined program.

Release Notes: Important Issues in Windows Server 2016

11/13/2018 • 4 minutes to read • Edit Online

Applies To: Windows Server 2016

These release notes summarize the most critical issues in the Windows Server® 2016 operating system, including ways to avoid or work around the issues, if known. For information about by-design changes, new features, and fixes in this release, see What's New in Windows Server 2016 and announcements from the specific feature teams. Unless otherwise specified, each reported issue applies to all editions and installation options of Windows Server 2016.

This document is continuously updated. As critical issues requiring a workaround are discovered, they are added, as are new workarounds and fixes as they become available.

Express updates available starting in November 2018 (NEW)

Starting with the November 2018 "Update Tuesday" update, Windows will again publish Express updates for Windows Server 2016. If you are using WSUS and System Center Configuration Manager (SCCM) you will once again see two packages for the Windows Server 2016 update: a Full update and an Express update. If you want to use Express for your server environments, you need to confirm that the server has taken a full update since November 2017 (KB# 4048953) to ensure the Express update installs correctly. If you attempt an Express update on a server that hasn't been updated since the 2017 11B update (KB# 4048953), you'll see repeated failures that consume bandwidth and CPU resources in an infinite loop. If you get into this scenario, stop pushing the Express update, and instead push a recent Full update to stop the failure loop.

Server Core installation option

When you install Windows Server 2016 by using the Server Core installation option, the print spooler is installed and starts by default even when the Print Server role is not installed.

To avoid this, after the first boot, set the print spooler to disabled.

Containers

- Before you use containers, install Servicing stack update for Windows 10 Version 1607: August 23, 2016 or any later updates that are available. Otherwise, a number of problems can occur, including failures in building, starting, or running containers, and errors similar to "CreateProcess failed in Win32: The RPC server is unavailable."
- The NanoServerPackage OneGet provider does not work in Windows Containers. To work around this, use Find-NanoServerPackage and Save-NanoServerPackage on a different computer (not a container) to download the needed package. Then copy the packages into the container and install them.

Device Guard

If you use virtualization-based protection of code integrity or Shielded virtual machines (that use virtualization-based protection of code integrity), you should be aware that these technologies could be incompatible with some devices and applications. You should test such configurations in your lab before enabling the features on

production systems. Failure to do so could result in unexpected data loss or stop errors.

Microsoft Exchange

If you attempt to run Microsoft Exchange 2016 CU3 on Windows Server 2016, you will experience errors in the IIS host process W3WP.exe. There is no workaround at this time. You should postpone deployment of Exchange 2016 CU3 on Windows Server 2016 until a supported fix is available.

Remote Server Administration Tools (RSAT)

If you are running a version of Windows 10 older than the Anniversary Update, and are using Hyper-V and virtual machines with an enabled virtual Trusted Platform Module (including shielded virtual machines), and then install the version of RSAT provided for Windows Server 2016, attempts to start those virtual machines will fail.

To avoid this, upgrade the client computer to Windows 10 Anniversary Update (or later) prior to installing RSAT. If this has already occurred, uninstall RSAT, upgrade the client to Window 10 Anniversary Update, and then reinstall RSAT.

Shielded virtual machines

- Ensure that you have installed all available updates before you deploy Shielded virtual machines in production.
- If you use virtualization-based protection of code integrity or Shielded virtual machines (that use virtualization-based protection of code integrity), you should be aware that these technologies could be incompatible with some devices and applications. You should test such configurations in your lab before enabling the features on production systems. Failure to do so could result in unexpected data loss or stop errors.

Start menu

This issue affects Windows Server 2016 installed with the Server with Desktop Experience option.

If you install any applications which add shortcut items inside a folder on the Start menu, the shortcuts will not work until you log out and log back in again.

Go back to the main Windows Server 2016 hub.

Storport Performance

Some systems may exhibit reduced storage performance when running a new install of Windows Server 2016 versus Windows Server 2012 R2. A number of changes were made during the development of Windows Server 2016 to improve security and reliability of the platform. Some of those changes, like enabling Windows Defender by default, result in longer I/O paths that can reduce I/O performance in certain workloads and patterns. Microsoft does not recommend disabling Windows Defender as it is an important layer of protection for your systems.

Copyright

1.0

Recommendations for moving to Windows Server 2016

3/19/2018 • 3 minutes to read • Edit Online

Applies To: Windows Server 2016

IF YOU ARE RUNNING:	WINDOWS SERVER 2012 R2 OR WINDOWS SERVER 2012	WINDOWS SERVER 2008 R2 OR WINDOWS SERVER 2008
Windows Server role infrastructure	Choose either upgrade or migration depending on specific role guidance.	- To take advantage of new features in Windows Server 2016, deploy new hardware, or install Windows Server 2016 in a virtual machine on an existing host. Some new features work best on a Windows Server 2016 physical host running Hyper-V. - Follow specific role guidance.
Microsoft server management and application workloads	- Application upgrades should include *migration* to Windows Server 2016. See the compatibility list. - Upgrades to Windows Server 2016 only (i.e., without upgrading applications) should use application-specific guidance.	- To take advantage of new features in Windows Server 2016, deploy new hardware, or install Windows Server 2016 in a virtual machine on an existing host. Some new features work best on a Windows Server 2016 physical host running Hyper-V. Follow migration guides as applicable. - Or, remain on your current OS and run in a virtual machine running on a Windows Server 2016 host, or Microsoft Azure. Contact your EA reseller, TAM, or Microsoft for extended support options through Software Assurance.
ISV application workloads	- Upgrades to Windows Server 2016 should use application-specific guidance. - For more information on Windows Server compatibility with non-Microsoft applications, visit the Windows Server Logo Certification portal.	- To take advantage of new features in Windows Server 2016, deploy new hardware, or install Windows Server 2016 in a virtual machine on an existing host. Some new features work best on a Windows Server 2016 physical host running Hyper-V. Follow migration guides as applicable. - Or, remain on your current OS and run in a virtual machine running on a Windows Server 2016 host, or Microsoft Azure. Contact your EA reseller, TAM, or Microsoft for extended support options through Software Assurance.

IF YOU ARE RUNNING:	WINDOWS SERVER 2012 R2 OR WINDOWS SERVER 2012	WINDOWS SERVER 2008 R2 OR WINDOWS SERVER 2008
Custom application workloads	- Consult with application developers on compatibility with Windows Server 2016 and upgrade guidance. - Leverage Microsoft Azure to test application on Windows Server 2016 prior to switch. - See complete options in the next section.	- Consult with your application developers on compatibility with Windows Server 2016 and upgrade guidance. - Leverage Microsoft Azure to test your application on Windows Server 2016 prior to switch. - To take advantage of new features in Windows Server 2016, deploy new hardware, or install Windows Server 2016 in a virtual machine on an existing host. Some new features work best on a Windows Server 2016 physical host running Hyper-V. - See complete options in the next section.

Complete options for moving servers running custom or "in-house" applications on older versions of Windows Server to Windows Server 2016

There are more options than ever before to help you and your customers take advantage of features in Windows Server 2016, with minimal impact to your current services and workloads.

- Try out the latest operating system with your application by downloading the evaluation version of Windows Server for testing on your premises. Once testing is complete and quality confirmed, you can perform a simple license conversion with a retail license key (requires restarting).
- Microsoft Azure can also be used on a trial basis for testing to ensure your custom application will work on the latest server operating system. Once testing is complete and quality confirmed, migrate to the latest Windows Server version on your premises.
- Or alternatively, once testing is complete and quality confirmed, Microsoft Azure can be used as the permanent location for your custom application or service. This allows the old server to remain available until you are ready to switch over to the new server in Azure.
    - If you already have Software Assurance for Windows Server, save money by deploying with the Azure Hybrid Use Benefit.
- In most cases, Microsoft Azure can be used to host the same application on the older version of Windows Server that it is running on today. Migrate the application and workload to a virtual machine with the operating system of your choice by using Azure Marketplace images.
    - If you already have Software Assurance for Windows Server, save money by deploying with the Azure Hybrid Use Benefit.
- The Software Assurance program for Windows Server provides new version rights benefits. Along with a list of other benefits, servers with Software Assurance can be upgraded to the latest version of Window Server when the time is right, without having to purchase a new license.

Additional resources

- Features removed or deprecated in Windows Server 2016
- For general server upgrade and migration options, visit Upgrade and conversion options for Windows Server 2016.

- For more information on product lifecycle and support levels, see the Support Lifecycle Policy FAQ.

Upgrade and conversion options for Windows Server 2016

10/30/2018 • 7 minutes to read • Edit Online

Applies To: Windows Server 2019, Windows Server 2016

This topic includes information about upgrading to Windows Server® 2016 from a variety of previous operating systems using a variety of methods.

The process of moving to Windows Server 2016 might vary greatly depending on which operating system you are starting with and the pathway you take. We use the following terms to distinguish among different actions, any of which could be involved in a new Windows Server 2016 deployment.

- **Installation** is the basic concept of getting the new operating system on your hardware. Specifically, a **clean installation** requires deleting the previous operating system. For information about installing Windows Server 2016, see System Requirements and Installation Information for Windows Server 2016. For information about installing other versions of Windows Server, see Windows Server Installation and Upgrade.
- **Migration** means moving from your existing operating system to Windows Server 2016 by transferring to a different set of hardware or virtual machine. Migration, which might vary considerably depending on the server roles you have installed, is discussed in detail at Windows Server Installation, Upgrade, and Migration.
- **Cluster OS Rolling Upgrade** is a new feature in Windows Server 2016 that enables an administrator to upgrade the operating system of the cluster nodes from Windows Server 2012 R2 to Windows Server 2016 without stopping the Hyper-V or the Scale-Out File Server workloads. This feature allows you to avoid downtime which could impact Service Level Agreements. This new feature is discussed in more detail at Cluster operating system rolling upgrade.
- **License conversion** In some operating system releases, you can convert a particular edition of the release to another edition of the same release in a single step with a simple command and the appropriate license key. We call this "license conversion." For example, if you are running Windows Server 2016 Standard, you can convert it to Windows Server 2016 Datacenter.
- **Upgrade** means moving from your existing operating system release to a more recent release while staying on the same hardware. (This is sometimes called "in-place" upgrade.) For example, if your server is running Windows Server 2012, or Windows Server 2012 R2, you can upgrade it to Windows Server 2016. You can upgrade from an evaluation version of the operating system to a retail version, from an older retail version to a newer version, or, in some cases, from a volume-licensed edition of the operating system to an ordinary retail edition.

IMPORTANT

Upgrade works best in virtual machines where specific OEM hardware drivers are not needed for a successful upgrade.

IMPORTANT

For releases of Windows Server 2016 prior to 14393.0.161119-1705.RS1_REFRESH, **you can only perform conversion from evaluation to retail** with Windows Server 2016 that has been installed by using the Desktop Experience option (not the Server Core option). Starting with version 14393.0.161119-1705.RS1_REFRESH and later releases, you can convert evaluation editions to retail regardless of the installation option used.

IMPORTANT

If your server uses NIC Teaming, disable NIC Teaming prior to upgrade, and then re-enable it after upgrade is complete. See NIC Teaming Overview for details.

Upgrading previous retail versions of Windows Server to Windows Server 2016

The table below briefly summarizes which **already licensed** (that is, not evaluation) Windows operating systems can be upgraded to which editions of Windows Server 2016.

Note the following general guidelines for supported paths:

- Upgrades from 32-bit to 64-bit architectures are not supported. All editions of Windows Server 2016 are 64-bit only.
- Upgrades from one language to another are not supported.
- If the server is a domain controller, see Upgrade Domain Controllers to Windows Server 2012 R2 and Windows Server 2012 for important information.
- Upgrades from pre-release versions (previews) of Windows Server 2016 are not supported. Perform a clean installation to Windows Server 2016.
- Upgrades that switch from a Server Core installation to a Server with a Desktop installation (or vice versa) are not supported.
- Upgrades from a previous Windows Server installation to an evaluation copy of Windows Server are not supported. Evaluation versions should be installed as a clean installation.

If you do not see your current version in the left column, upgrading to this release of Windows Server 2016 is not supported.

If you see more than one edition in the right column, upgrading to **either** edition from the same starting version is supported.

IF YOU ARE RUNNING THIS EDITION:	YOU CAN UPGRADE TO THESE EDITIONS:
Windows Server 2012 Standard	Windows Server 2016 Standard or Datacenter
Windows Server 2012 Datacenter	Windows Server 2016 Datacenter
Windows Server 2012 R2 Standard	Windows Server 2016 Standard or Datacenter
Windows Server 2012 R2 Datacenter	Windows Server 2016 Datacenter
Windows Server 2012 R2 Essentials	Windows Server 2016 Essentials
Windows Storage Server 2012 Standard	Windows Storage Server 2016 Standard

IF YOU ARE RUNNING THIS EDITION:	YOU CAN UPGRADE TO THESE EDITIONS:
Windows Storage Server 2012 Workgroup	Windows Storage Server 2016 Workgroup
Windows Storage Server 2012 R2 Standard	Windows Storage Server 2016 Standard
Windows Storage Server 2012 R2 Workgroup	Windows Storage Server 2016 Workgroup

Per-server-role considerations for upgrading

Even in supported upgrade paths from previous retail versions to Windows Server 2016, certain server roles that are already installed might require additional preparation or actions for the role to continue functioning after the upgrade. Consult the specific TechNet Library topics for each server role you intend to upgrade for details of additional steps that might be required.

Converting a current evaluation version to a current retail version

You can convert the evaluation version of Windows Server 2016 Standard to either Windows Server 2016 Standard (retail) or Datacenter (retail). Similarly, you can convert the evaluation version of Windows Server 2016 Datacenter to the retail version.

> **IMPORTANT**
> For releases of Windows Server 2016 prior to 14393.0.161119-1705.RS1_REFRESH, you can only perform this conversion from evaluation to retail with Windows Server 2016 that has been installed by using the Desktop Experience option (not the Server Core option). Starting with version 14393.0.161119-1705.RS1_REFRESH and later releases, you can convert evaluation editions to retail regardless of the installation option used.

Before you attempt to convert from evaluation to retail, verify that your server is actually running an evaluation version. To do this, do either of the following:

- From an elevated command prompt, run **slmgr.vbs /dlv**; evaluation versions will include "EVAL" in the output.
- From the Start screen, open **Control Panel**. Open **System and Security**, and then **System**. View Windows activation status in the Windows activation area of the **System** page. Click **View details** in Windows activation for more information about your Windows activation status.

If you have already activated Windows, the Desktop shows the time remaining in the evaluation period.

If the server is running a retail version instead of an evaluation version, see the "Upgrading previous retail versions of Windows Server to Windows Server 2016" section of this topic for instructions to upgrade to Windows Server 2016.

For **Windows Server 2016 Essentials**: You can convert to the full retail version by entering a retail, volume license, or OEM key in the command **slmgr.vbs**.

If the server is running an evaluation version of Windows Server 2016 Standard or Windows Server 2016 Datacenter, you can convert it to a retail version as follows:

1. If the server is a **domain controller**, you cannot convert it to a retail version. In this case, install an additional domain controller on a server that runs a retail version and remove AD DS from the domain controller that runs on the evaluation version. For more information, see Upgrade Domain Controllers to Windows Server 2012 R2 and Windows Server 2012.
2. Read the license terms.

3. From an elevated command prompt, determine the current edition name with the command **DISM /online /Get-CurrentEdition**. Make note of the edition ID, an abbreviated form of the edition name. Then run **DISM /online /Set-Edition:<edition ID> /ProductKey:XXXXX-XXXXX-XXXXX-XXXXX-XXXXX /AcceptEula**, providing the edition ID and a retail product key. The server will restart twice.

For the evaluation version of Windows Server 2016 Standard, you can also convert to the retail version of Windows Server 2016 Datacenter in one step using this same command and the appropriate product key.

> Tip For more information about Dism.exe, see DISM Command-line options.

Converting a current retail edition to a different current retail edition

At any time after installing Windows Server 2016, you can run Setup to repair the installation (sometimes called "repair in place") or, in certain cases, to convert to a different edition. You can run Setup to perform a "repair in place" on any edition of Windows Server 2016; the result will be the same edition you started with.

For Windows Server 2016 Standard, you can convert the system to Windows Server 2016 Datacenter as follows: From an elevated command prompt, determine the current edition name with the command **DISM /online /Get-CurrentEdition**. For Windows Server 2016 Standard this will be `ServerStandard`. Run the command **DISM /online /Get-TargetEditions** to get the ID of the edition you can upgrade to. Make note of this edition ID, an abbreviated form of the edition name. Then run **DISM /online /Set-Edition:<edition ID> /ProductKey:XXXXX-XXXXX-XXXXX-XXXXX-XXXXX /AcceptEula**, providing the edition ID of your target and its retail product key. The server will restart twice.

Converting a current retail version to a current volume-licensed version

At any time after installing Windows Server 2016, you can freely convert it between a retail version, a volume-licensed version, or an OEM version. The edition remains the same during this conversion. If you are starting with an evaluation version, convert it to the retail version first, and then you can inter-convert as described here.

To do this, from an elevated command prompt, run: **slmgr /ipk <key>**

Where <key> is the appropriate volume-license, retail, or OEM product key.

Server role upgrade and migration matrix for Windows Server 2016

7/19/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server 2016

The grid on this page explains your server role upgrade and migration options specifically for moving to Windows Server 2016. For individual role migration guides, visit Migrating Roles and Features in Windows Server. For more information about installation and upgrades, see Windows Server Installation, Upgrade, and Migration.

SERVER ROLE	UPGRADEABLE FROM WINDOWS SERVER 2012 R2?	UPGRADEABLE FROM WINDOWS SERVER 2012?	MIGRATION SUPPORTED?	CAN MIGRATION BE COMPLETED WITHOUT DOWNTIME?
Active Directory Certificate Services	Yes	Yes	Yes	No
Active Directory Domain Services	Yes	Yes	Yes	Yes
Active Directory Federation Services	No	No	Yes	No (new nodes need to be added to the farm)
Active Directory Lightweight Directory Services	Yes	Yes	Yes	Yes
Active Directory Rights Management Services	Yes	Yes	Yes	No
DHCP Server	Yes	Yes	Yes	Yes
DNS Server	Yes	Yes	Yes	No
Failover Cluster	Yes with Cluster OS Rolling Upgrade process which includes node Pause-Drain, Evict, upgrade to Windows Server 2016 and rejoin the original cluster. Yes, when the server is removed by the cluster for upgrade and then added to a different cluster.	Not while the server is part of a cluster. Yes, when the server is removed by the cluster for upgrade and then added to a different cluster.	Yes	No for Windows Server 2012 Failover Clusters. Yes for Windows Server 2012 R2 Failover Clusters with Hyper-V VMs or Windows Server 2012 R2 Failover Clusters running the Scale-out File Server role. See Cluster OS Rolling Upgrade.
File and Storage Services	Yes	Yes	Varies by sub-feature	No

SERVER ROLE	UPGRADEABLE FROM WINDOWS SERVER 2012 R2?	UPGRADEABLE FROM WINDOWS SERVER 2012?	MIGRATION SUPPORTED?	CAN MIGRATION BE COMPLETED WITHOUT DOWNTIME?
Hyper-V	Yes. (When the host is part of a cluster with Cluster OS Rolling Upgrade process which includes node Pause-Drain, Evict, upgrade to Windows Server 2016 and rejoin the original cluster.)	No	Yes	No for Windows Server 2012 Failover Clusters. Yes for Windows Server 2012 R2 Failover Clusters with Hyper-V VMs or Windows Server 2012 R2 Failover Clusters running the Scale-out File Server role. See Cluster OS Rolling Upgrade.
Print and Fax Services	No	No	Yes (Printbrm.exe)	No
Remote Desktop Services	Yes, for all sub-roles, but mixed mode farm is not supported	Yes, for all sub-roles, but mixed mode farm is not supported	Yes	No
Web Server (IIS)	Yes	Yes	Yes	No
Windows Server Essentials Experience	Yes	N/A – new feature	Yes	No
Windows Server Update Services	Yes	Yes	Yes	No
Work Folders	Yes	Yes	Yes	Yes from WS 2012 R2 cluster when using Cluster OS Rolling Upgrade.

Windows Server 2016 and Microsoft Server Application Compatibility

11/6/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server 2016

This table lists Microsoft server applications that support installation and functionality on Window Server 2016. This information is for quick reference and is not intended to replace the individual product specifications, requirements, announcements, or general communications of each individual server application. Refer to official documentation for each product to fully understand compatibility and options.

For customers and software vendor partners looking for more information on Windows Server compatibility with non-Microsoft applications, visit the Commercial App Certification portal.

MICROSOFT SERVER APPLICATION	RELEASED?	PRODUCT LINK
Microsoft SQL Server 2012	Yes	Hardware and Software Requirements for Installing SQL Server 2012
Microsoft SQL Server 2014	Yes	Hardware and Software Requirements for Installing SQL Server 2014
Microsoft SQL Server 2016	Yes	SQL Server 2016
Microsoft System Center Virtual Machine Manager 2016	Yes	What's New in System Center
Microsoft System Center Operations Manager 2016	Yes	What's New in System Center
Microsoft System Center Data Protection Manager 2016	Yes	What's New in System Center
Microsoft System Center Configuration Manager (version 1606)	Yes	What's new in version 1606 of System Center Configuration Manager
SharePoint Server 2016	Yes	Hardware and software requirements for SharePoint Server 2016
Project Server 2016	Yes	Software requirements for Project Server 2016
Exchange Server 2016	Yes	Updates for Exchange 2016
Biztalk Server 2016	Yes	Microsoft BizTalk Server
Host Integration Server 2016	Yes	What's New in HIS 2016
Visual Studio Team Foundation Server 2017	Yes	Team Foundation Server 2017

MICROSOFT SERVER APPLICATION	RELEASED?	PRODUCT LINK
Skype for Business Server 2015	Yes	How to install Skype for Business Server 2015 on Windows Server 2016
Office Online Server	Yes	Plan Office Online Server

Features removed or planned for replacement starting with Windows Server, version 1803

8/8/2018 • 4 minutes to read • Edit Online

Applies to: Windows Server, version 1803

Each release of Windows Server adds new features and functionality; we also occasionally remove features and functionality, usually because we've added a better option. Here are the details about the features and functionalities that we removed in Windows Server, version 1803.

TIP

- You can get early access to Windows Server builds by joining the Windows Insider program - this is a great way to test feature changes.
- Have questions about other releases? Check out the information for Windows Server 2016, and Windows Server, version 1709.

The list is subject to change and might not include every affected feature or functionality.

Features we removed in this release

We've removed the following features and functionalities from the installed product image in Windows Server, version 1803. Applications or code that depend on these features won't function in this release unless you use an alternate method.

FEATURE	INSTEAD YOU CAN USE...
File Replication Service	File Replication Services, introduced in Windows Server 2003 R2, has been replaced by DFS Replication. You need to migrate any domain controllers that user FRS to DFS Replication with SYSVOL.
Hyper-V Network Virtualization (HNV)	Network Virtualization is now included in Windows Server as part of the Software Defined Networking (SDN) solution, which also includes the Network Controller, Software Load Balancing, User-Defined Routing, and Access Control Lists.

Features we're no longer developing

We are no longer actively developing these features and may remove them from a future update. Some features have been replaced with other features or functionality, while others are now available from different sources.

NOTE

Please note that some of the features and roles described below are not included in the Server Core installation option, which is provided in Windows Server, version 1803. They *are* present in the Server with Desktop Experience installation option, which we last released with Windows Server 2016 and will release again in Windows Server 2019.

If you have feedback about the proposed replacement of any of these features, you can use the Feedback Hub app.

FEATURE OR ROLE	INSTEAD YOU CAN USE...
Business Scanning, also called Distributed Scan Management (DSM)	The Scan Management functionality was introduced in Windows Server 2008 R2 and enabled secure scanning and the management of scanners in an enterprise. We're no longer investing in this feature, and there are no devices available that support it.
IPv4/6 Transition Technologies (6to4, ISATAP, and Direct Tunnels)	6to4 has been disabled by default since Windows 10, version 1607 (the Anniversary Update), ISATAP has been disabled by default since Windows 10, version 1703 (the Creators Update), and Direct Tunnels has always been disabled by default. Please use native IPv6 support instead.
MultiPoint Services	We're no longer developing the MultiPoint Services role as part of Windows Server. MultiPoint Connector services are available through Feature on Demand for both Windows Server and Windows 10. You can use Remote Desktop Services, in particular the Remote Desktop Services Session Host, to provide RDP connectivity.
Offline symbol packages (Debug symbol MSIs)	We're no longer making the symbol packages available as a downloadable MSI. Instead, the Microsoft Symbol Server is moving to be an Azure-based symbol store. If you need the Windows symbols, connect to the Microsoft Symbol Server to cache your symbols locally or use a manifest file with SymChk.exe on a computer with internet access.
Remote Desktop Connection Broker and Remote Desktop Virtualization Host in a Server Core installation	Most Remote Desktop Services deployments have these roles co-located with the Remote Desktop Session Host (RDSH), which requires Server with Desktop Experience; to be consistent with RDSH we're changing these roles to also require Server with Desktop Experience. We're no longer developing these RDS roles for use in a Server Core installation. If you need to deploy these roles as part of your Remote Desktop infrastructure, you can install them on Windows Server 2016 with Desktop Experience. These roles are also included in the Desktop Experience installation option of Windows Server 2019. You can test them in the Windows Insider build of Windows Server 2019 – just be sure to choose the LTSC image.
RemoteFX vGPU	We're developing new graphics acceleration options for virtualized environments. You can also use Discrete Device Assignment (DDA) as an alternative.
Software Restriction Policies in Group Policy	Instead of using the Software Restriction Policies through Group Policy, you can use AppLocker or Windows Defender Application Control to control which apps users can access and what code can run in the kernel.
Storage Spaces in a Shared configuration using a SAS fabric	Deploy Storage Spaces Direct instead. Storage Spaces Direct supports the use of HLK-certified SAS enclosures, but in a non-shared configuration, as described in the Storage Spaces Direct hardware requirements.

FEATURE OR ROLE	INSTEAD YOU CAN USE...
Windows Server Essentials Experience	We're no longer developing the Essentials Experience role for the Windows Server Standard or Windows Server Datacenter SKUs. If you need an easy-to-use server solution for small-to-medium businesses, check out our new Microsoft 365 for business solution, or use Windows Server 2016 Essentials.

Features removed or planned for replacement starting with Windows Server, version 1709

9/21/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server, version 1709

The following is a list of features and functionalities in Windows Server, version 1709 that have either been removed from the product in that release or are starting to be considered for potential replacement in subsequent releases. It is intended for IT professionals who are updating operating systems in a commercial environment. **This list is subject to change in subsequent releases and might not include every affected feature or functionality.**

Features removed from Windows Server, version 1709

Windows Server, version 1709 contains the same features present in Windows Server 2016. However, this release does offer different installation options than Windows Server 2016 does:

- As a Semi-Annual Channel release, Windows Server, version 1709 offers only the Server Core installation option. For details, see Windows Server Semi-Annual Channel overview.
- Starting with this release, Nano Server is not available as an installable host operating system. Instead, Nano Server is available as a container operating system. See Changes to Nano Server in Windows Server, version 1709.
- Starting with this release, Server Message Block (SMB) version 1 is no longer installed by default. For details, see SMBv1 is not installed by default in Windows 10 Fall Creators Update and Windows Server, version 1709 and later versions.

Features being considered for replacement starting with subsequent releases

The following features and functionalities are being considered for replacement starting with releases after Windows Server, version 1709. Eventually, they may be completely removed from the installed product image and replaced by other features or functionality (or installable from other sources), but they are still available in this release, sometimes with certain functionality removed. You should begin planning now to employ alternate methods or the future replacement for any applications, code, or usage that depend on these features.

If you have feedback to share about the proposed replacement of any of these features, you can use the Feedback Hub app. Even though this app runs on Windows 10, you can use it to send us feedback about the Windows Server product (and documentation) as well.

IIS 6 Management compatibility

Specific features being considered for replacement are:

- IIS 6 Metabase Compatibility (Web-Metabase)
- IIS 6 Management Console (Web-Lgcy-Mgmt-Console)
- IIS 6 Scripting Tools (Web-Lgcy-Scripting)
- IIS 6 WMI Compatibility (Web-WMI)

Instead of IIS 6 Metabase Compatibility (which acts as an emulation layer between IIS 6-based metabase scripts and the file-based configuration used by IIS 7 or newer versions) you should start migrating management scripts

to target IIS file-based configuration directly, by using tools such as the Microsoft.Web.Administration namespace.

You should also start migration from IIS 6.0 or earlier versions, and move to the latest version of IIS, which is always available in the most recent release of Windows Server.

IIS Digest Authentication

This authentication method is planned for replacement. Instead, you should start using other authentication methods such as Client Certificate Mapping (see Configuring One-to-One Client Certificate Mappings) or Windows Authentication (see Application Settings).

Internet Storage Name Service (iSNS)

iSNS is being considered for replacement. The Server Message Block (SMB) feature offers essentially the same functionality with additional features. See Server Message Block Overview for background information on this feature.

RSA/AES Encryption for IIS

This encryption method is being considered for replacement because the superior Cryptography API: Next Generation (CNG) method is already available. To learn more about CNG encryption, see About CNG.

Windows PowerShell 2.0

This early version of Windows PowerShell has been superseded by several more recent versions. For the best features and performance, migrate to Windows PowerShell 5.0 or later. See PowerShell Documentation for plenty of information.

Features Removed or Deprecated in Windows Server 2016

9/21/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server 2016

The following is a list of features and functionalities in Windows Server 2016 that have either been removed from the product in the current release or are planned for potential removal in subsequent releases (deprecated). It is intended for IT professionals who are updating operating systems in a commercial environment. This list is subject to change in subsequent releases and may not include every deprecated feature or functionality. For more details about a particular feature or functionality and its replacement, see the documentation for that feature.

Features removed from Windows Server 2016

The following features and functionalities have been removed from this release of Windows Server 2016. Applications, code, or usage that depend on these features will not function in this release unless you employ an alternate method.

> **NOTE**
> If you are moving to Windows Server 2016 from a server release prior to Windows Server 2012 R2 or Windows Server 2012, you should also review Features Removed or Deprecated in Windows Server 2012 R2 and Features Removed or Deprecated in Windows Server 2012.

File server

The Share and Storage Management snap-in for Microsoft Management Console has been removed. Instead, do any of the following:

- If the computer you want to manage is running an operating system older than Windows Server 2016, connect to it with Remote Desktop and use the local version of the Share and Storage Management snap-in.
- On a computer running Windows 8.1 or earlier, use the Share and Storage Management snap-in from RSAT to view the computer you want to manage.
- Use Hyper-V on a client computer to run a virtual machine running Windows 7, Windows 8, or Windows 8.1 that has the Share and Storage Management snap-in in RSAT.

Journal.dll

The Journal.dll is removed from Windows Server 2016. There is no replacement.

Security Configuration Wizard

The Security Configuration Wizard is removed. Instead, features are secured by default. If you need to control specific security settings, you can use either Group Policy or Microsoft Security Compliance Manager.

SQM

The opt-in components that manage participation in the Customer Experience Improvement Program have been removed.

Windows Update

The **wuauclt.exe /detectnow** command has been removed and is no longer supported. To trigger a scan for

updates, do either of the following:

- Run these PowerShell commands:

```
$AutoUpdates = New-Object -ComObject "Microsoft.Update.AutoUpdate"`
$AutoUpdates.DetectNow()`
```

- Alternately, use this VBScript:

```
Set automaticUpdates = CreateObject("Microsoft.Update.AutoUpdate")
automaticUpdates.DetectNow()
```

Features deprecated starting with Windows Server 2016

The following features and functionalities are deprecated starting with this release. Eventually, they will be completely removed from the product, but they are still available in this release, sometimes with certain functionality removed. You should begin planning now to employ alternate methods for any applications, code, or usage that depend on these features.

Configuration tools

- **Scregedit.exe** is deprecated. If you have scripts that depend on Scregedit.exe, adjust them to use Reg.exe or Windows PowerShell methods.
- **Sconfig.exe** is deprecated. Use Windows PowerShell instead.

NetCfg custom APIs

Installation of PrintProvider, NetClient, and ISDN using NetCfg custom APIs is deprecated.

Remote management

WinRM.vbs is deprecated. Instead, use functionality in the WinRM provider of Windows PowerShell.

SMB

SMB 2+ over NetBT is deprecated. Instead, implement SMB over TCP or RDMA.

Comparison of Standard and Datacenter editions of Windows Server 2016

1/23/2018 • 5 minutes to read • Edit Online

Applies To: Windows Server 2016

Locks and Limits

LOCKS AND LIMITS	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Maximum number of users	Based on CALs	Based on CALs
Maximum SMB connections	16777216	16777216
Maximum RRAS connections	unlimited	unlimited
Maximum IAS connections	2147483647	2147483647
Maximum RDS connections	65535	65535
Maximum number of 64-bit sockets	64	64
Maximum number of cores	unlimited	unlimited
Maximum RAM	24 TB	24 TB
Can be used as virtualization guest	Yes; 2 virtual machines, plus one Hyper-V host per license	Yes; unlimited virtual machines, plus one Hyper-V host per license
Server can join a domain	yes	yes
Edge network protection/firewall	no	no
DirectAccess	yes	yes
DLNA codecs and web media streaming	Yes, if installed as Server with Desktop Experience	Yes, if installed as Server with Desktop Experience

Server roles

WINDOWS SERVER ROLES AVAILABLE	ROLE SERVICES	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Active Directory Certificate Services		Yes	Yes
Active Directory Domain Services		Yes	Yes

WINDOWS SERVER ROLES AVAILABLE	ROLE SERVICES	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Active Directory Federation Services		Yes	Yes
AD Lightweight Directory Services		Yes	Yes
AD Rights Management Services		Yes	Yes
Device Health Attestation		Yes	Yes
DHCP Server		Yes	Yes
DNS Server		Yes	Yes
Fax Server		Yes	Yes
File and Storage Services	File Server	Yes	Yes
File and Storage Services	BranchCache for Network Files	Yes	Yes
File and Storage Services	Data Deduplication	Yes	Yes
File and Storage Services	DFS Namespaces	Yes	Yes
File and Storage Services	DFS Replication	Yes	Yes
File and Storage Services	File Server Resource Manager	Yes	Yes
File and Storage Services	File Server VSS Agent Service	Yes	Yes
File and Storage Services	iSCSI Target Server	Yes	Yes
File and Storage Services	iSCSI Target Storage Provider	Yes	Yes
File and Storage Services	Server for NFS	Yes	Yes
File and Storage Services	Work Folders	Yes	Yes
File and Storage Services	Storage Services	Yes	Yes
Host Guardian Service		Yes	Yes
Hyper-V		Yes	Yes; including Shielded Virtual Machines
MultiPoint Services		Yes	Yes
Network Controller		No	Yes

WINDOWS SERVER ROLES AVAILABLE	ROLE SERVICES	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Network Policy and Access Services		Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Print and Document Services		Yes	Yes
Remote Access		Yes	Yes
Remote Desktop Services		Yes	Yes
Volume Activation Services		Yes	Yes
Web Services (IIS)		Yes	Yes
Windows Deployment Services		Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Windows Server Essentials Experience		Yes	Yes
Windows Server Update Services		Yes	Yes

Features

WINDOWS SERVER FEATURES INSTALLABLE WITH SERVER MANAGER (OR POWERSHELL)	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
.NET Framework 3.5	Yes	Yes
.NET Framework 4.6	Yes	Yes
Background Intelligent Transfer Service (BITS)	Yes	Yes
BitLocker Drive Encryption	Yes	Yes
BitLocker Network Unlock	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
BranchCache	Yes	Yes
Client for NFS	Yes	Yes
Containers	Yes (Windows containers unlimited; Hyper-V containers up to 2)	Yes (all container types unlimited)
Data Center Bridging	Yes	Yes
Direct Play	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience

WINDOWS SERVER FEATURES INSTALLABLE WITH SERVER MANAGER (OR POWERSHELL)	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Enhanced Storage	Yes	Yes
Failover Clustering	Yes	Yes
Group Policy Management	Yes	Yes
Host Guardian Hyper-V Support	No	Yes
I/O Quality of Service	Yes	Yes
IIS Hostable Web Core	Yes	Yes
Internet Printing Client	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
IPAM Server	Yes	Yes
iSNS Server service	Yes	Yes
LPR Port Monitor	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Management OData IIS Extension	Yes	Yes
Media Foundation	Yes	Yes
Message Queueing	Yes	Yes
Multipath I/O	Yes	Yes
MultiPoint Connector	Yes	Yes
Network Load Balancing	Yes	Yes
Peer Name Resolution Protocol	Yes	Yes
Quality Windows Audio Video Experience	Yes	Yes
RAS Connection Manager Administration Kit	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Remote Assistance	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Remote Differential Compression	Yes	Yes
RSAT	Yes	Yes
RPC over HTTP Proxy	Yes	Yes

WINDOWS SERVER FEATURES INSTALLABLE WITH SERVER MANAGER (OR POWERSHELL)	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Setup and Boot Event Collection	Yes	Yes
Simple TCP/IP Services	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
SMB 1.0/CIFS File Sharing Support	Installed	Installed
SMB Bandwidth Limit	Yes	Yes
SMTP Server	Yes	Yes
SNMP Service	Yes	Yes
Software Load Balancer	Yes	Yes
Storage Replica	No	Yes
Telnet Client	Yes	Yes
TFTP Client	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
VM Shielding Tools for Fabric Management	Yes	Yes
WebDAV Redirector	Yes	Yes
Windows Biometric Framework	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Windows Defender features	Installed	Installed
Windows Identity Foundation 3.5	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Windows Internal Database	Yes	Yes
Windows PowerShell	Installed	Installed
Windows Process Activation Service	Yes	Yes
Windows Search Service	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
Windows Server Backup	Yes	Yes
Windows Server Migration Tools	Yes	Yes
Windows Standards-Based Storage Management	Yes	Yes

WINDOWS SERVER FEATURES INSTALLABLE WITH SERVER MANAGER (OR POWERSHELL)	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Windows TIFF IFilter	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience
WinRM IIS Extension	Yes	Yes
WINS Server	Yes	Yes
Wireless LAN Service	Yes	Yes
WoW64 support	Installed	Installed
XPS Viewer	Yes, when installed as Server with Desktop Experience	Yes, when installed as Server with Desktop Experience

FEATURES AVAILABLE GENERALLY	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
Best Practices Analyzer	Yes	Yes
Direct Access	Yes	Yes
Dynamic Memory (in virtualization)	Yes	Yes
Hot Add/Replace RAM	Yes	Yes
Microsoft Management Console	Yes	Yes
Minimal Server Interface	Yes	Yes
Network Load Balancing	Yes	Yes
Windows PowerShell	Yes	Yes
Server Core installation option	Yes	Yes
Nano Server installation option	Yes	Yes
Server Manager	Yes	Yes
SMB Direct and SMB over RDMA	Yes	Yes
Software-defined Networking	No	Yes
Storage Management Service	Yes	Yes
Storage Spaces	Yes	Yes
Storage Spaces Direct	No	Yes
Volume Activation Services	Yes	Yes

FEATURES AVAILABLE GENERALLY	WINDOWS SERVER 2016 STANDARD	WINDOWS SERVER 2016 DATACENTER
VSS (Volume Shadow Copy Service) integration	Yes	Yes
Windows Server Update Services	Yes	Yes
Windows System Resource Manager	Yes	Yes
Server license logging	Yes	Yes
Inherited activation	As guest if hosted on Datacenter	Can be host or guest
Work folders	Yes	Yes

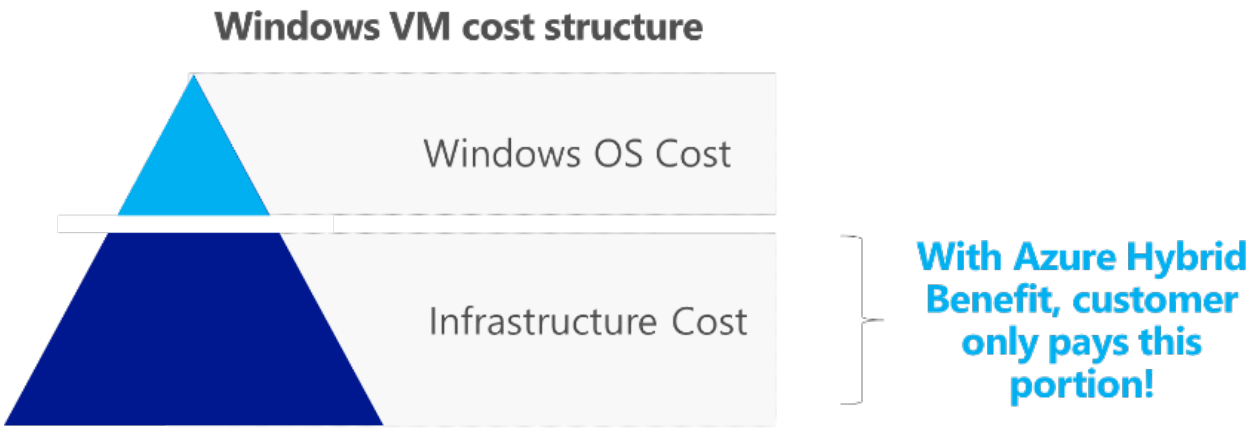
Azure Hybrid Benefit for Windows Server

10/29/2018 • 5 minutes to read • Edit Online

Applies To: Windows Server

Benefit description, rules, and use cases

The Azure Hybrid Benefit for Windows Server allows you to save up to 40% on Windows Server VMs in Azure by utilizing your on-premises Windows Server licenses with Software Assurance. With this benefit, customers need to only pay for the infrastructure costs of the virtual machine because the licensing for Windows Server is covered by the Software Assurance benefit. The benefit is applicable to both Standard and Datacenter editions of Windows Server for the 2008R2, 2012, 2012R2 and 2016 releases. This benefit is available across all regions and sovereign clouds.



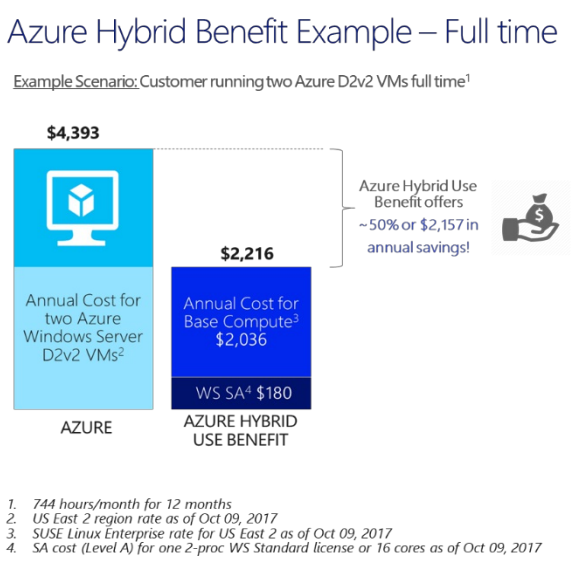
All you need to qualify for the benefit is an active Software Assurance or Subscription License such as EAS, SCE subscription or Open Value Subscription on their Windows Server licenses.

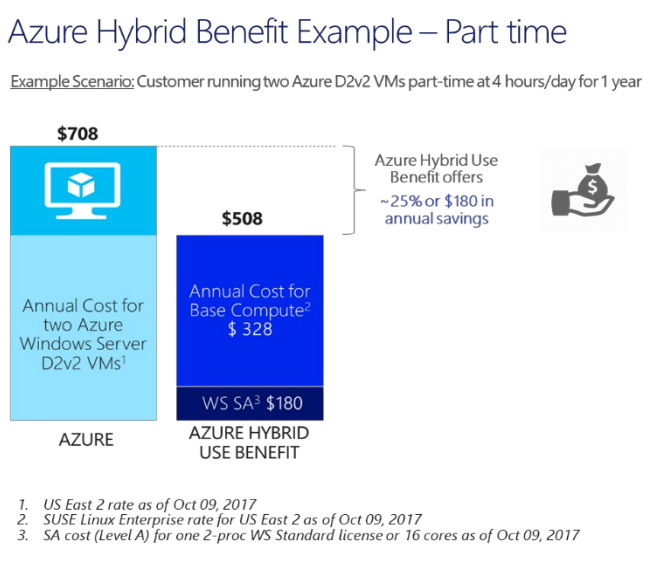
Each Windows Server 2-processor license with active SA/Subscription, and each set of 16 Windows Server core licenses with SA/Subscription, entitles the customer to use Windows Server on Microsoft Azure on up to 16 virtual cores allocated across two or fewer Azure Base Instances (virtual machines). Each additional set of 8 core licenses with SA/Subscription entitles use on up to 8 virtual cores and one Base Instance (VM).

LICENSE WITH SA/SUBSCRIPTION	VMS AND CORES GRANTED	HOW THEY CAN BE USED
WS Datacenter (16 cores or a 2-proc L)	Up to two VMs and up to 16 cores	Run virtual machines both on premises and in Azure
WS Standard (16 cores or a 2-proc L)	Up to two VMs and up to 16 cores	Run virtual machines either on premises or in Azure

VMs utilizing the Azure Hybrid Benefit can run in Azure only during the SA / Subscription term. When approaching SA/Subscription expiry time, the customer has an option to either renew their SA/subscription, turn off the hybrid benefit functionality for that VM or de-provision the VM using the hybrid benefit.

Savings examples





Below you can find a reference table to assist you with understanding the benefit rules with more granularity. The green column shows the quantity of same-type VMs and the blue row shows the core density of each VM. The yellow cells show the number of 2-proc licenses (or sets of 16 cores) one must have to deploy a certain number of VMs of a certain core density.

Windows Server with SA Requirements Reference Table:

VM Core Density

Number of same-type VMs	1	2	4	8	10	12	16	20	32	64	128
1	1	1	1	1	1	1	1	2	2	4	8
2	1	1	1	1	2	2	2	3	4	8	16
3	2	2	2	2	3	3	3	6	6	12	24
4	2	2	2	2	4	4	4	8	8	16	32
5	3	3	3	3	5	5	5	10	10	20	40
6	3	3	3	3	6	6	6	12	12	24	48
7	4	4	4	4	7	7	7	14	14	28	56
8	4	4	4	4	8	8	8	16	16	32	64
9	5	5	5	5	9	9	9	18	18	36	72
10	5	5	5	5	10	10	10	20	20	40	80

*Number of WS licenses covered with SA or WS subscriptions needed for each scenario**

Examples:

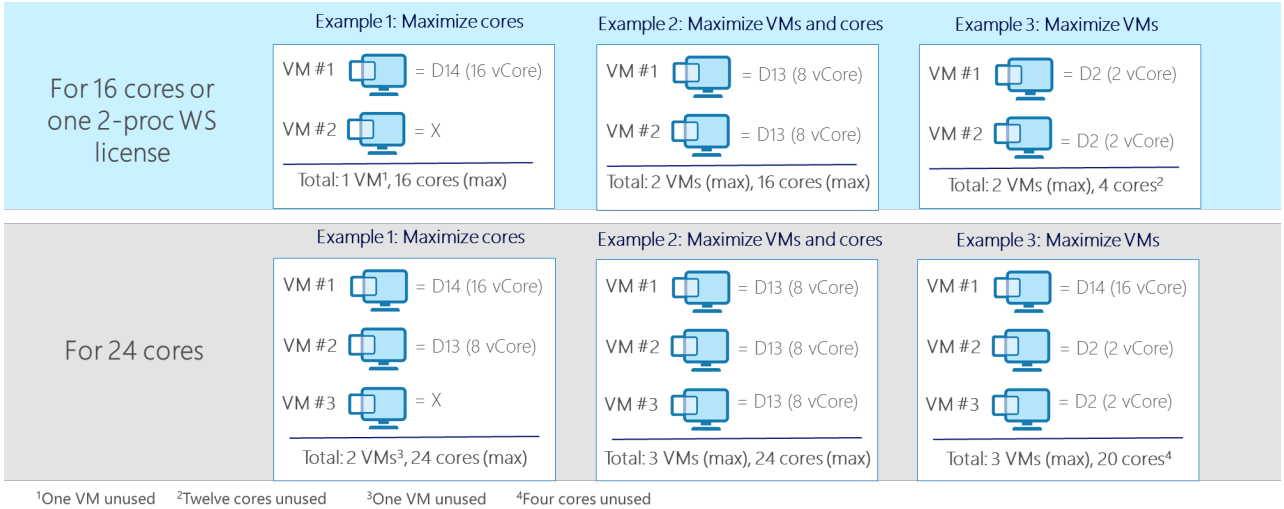
1. To run 2 VMs of 4 cores each (such as A3) you need 1 WS license.

2. To run 5 VMs of 16 cores each (such as D5V2) you need 5 WS licenses

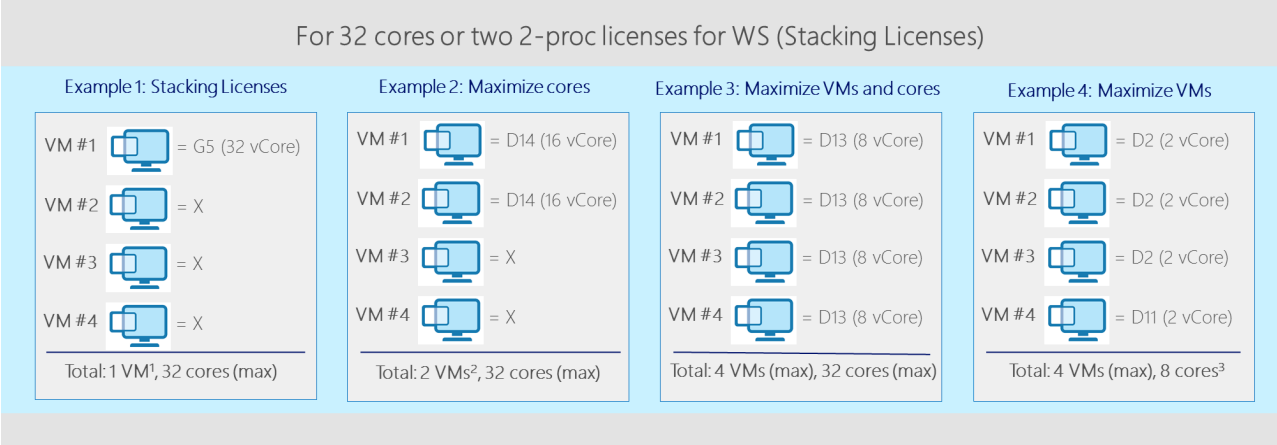
3. To run 10 VMs of 64 cores each (such as M64S) you need 40 WS licenses

The Azure Hybrid Benefit for Windows Server also allows flexibility to run configurations as per your needs as well as combining VMs of different types.

Example configurations for several licensing positions:



[1]One VM unused [2]Twelve cores unused [3]One VM unused [4]Four cores unused



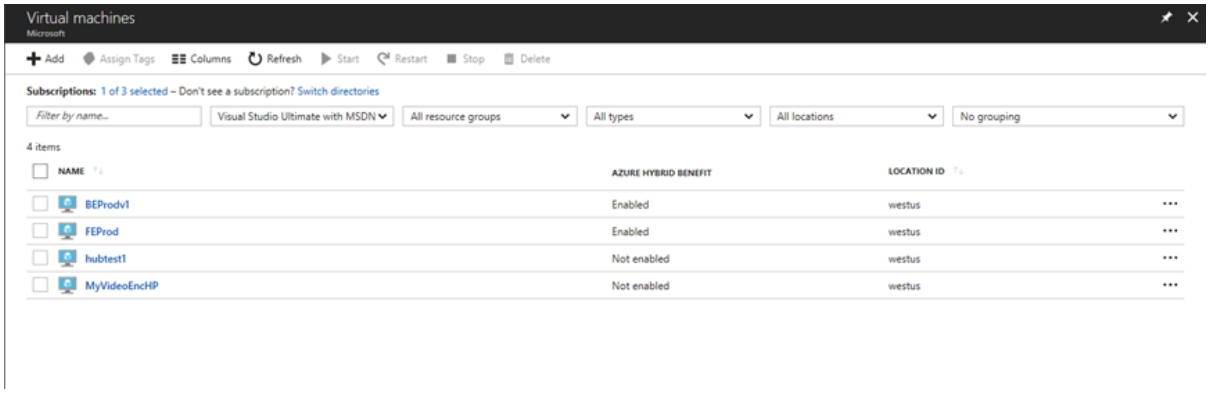
[1]Three VMs unused
[2]Two VMs unused
[3] Twenty-four cores unused

If you want to learn more about the Azure Hybrid Benefit for Windows Server, please go to the Azure Hybrid Benefit website.

How to maintain compliance

Customers seeking to apply the Azure Hybrid Benefit to their Windows Server VMs need to verify the number of eligible licenses and respective coverage period of their SA/Subscription before any activation of this benefit and apply the guidelines above to deploy the correct number of VMs with the benefit. If you already have VMs running with the Azure Hybrid Benefit, you will need to perform an inventory of how many units you are running, and check against the active SA licenses you have. Please contact your Microsoft Enterprise Agreement licensing specialist to validate your SA licensing position. To see and count all virtual machines deployed with Azure Hybrid Benefit for Windows Server in a subscription, you can do one of the below:

1. Configure the Microsoft Azure Portal to show Azure Hybrid Benefit for Windows Server utilization Add the column "Azure Hybrid Benefit" in the list view of the virtual machines section in the Microsoft Azure Portal.



2. Use PowerShell to list Azure Hybrid Benefit for Windows Server utilization

```
$vms = Get-AzureRMVM
foreach ($vm in $vms) {"VM Name: " + $vm.Name, "   Azure Hybrid Benefit for Windows Server: "+
$vm.LicenseType}
```

3. Look at your Microsoft Azure bill to determine how many virtual machines with Azure Hybrid Benefit for Windows Server you are running. The information about the number of instances with the benefit shows under 'Additional Info':

```
"{"ImageType":"WindowsServerBYOL","ServiceType":"Standard_A1","VMName":"","UsageType":"ComputeHR"}"
```

Please note that billing does not apply in real time, i.e. there will be a few hours delay from the time you've activated a VM with the hybrid benefit before it shows on the bill. You can then populate the results in the **Azure Hybrid Benefit for Windows Server SA Count Tool** below to get to the number of WS licenses covered with SA or Subscriptions that are required.

Be sure to perform an inventory in each subscription that you own to generate a comprehensive view of your licensing position.

Azure Hybrid Benefit WS SA Count Tool

If you performed the above and confirmed you are fully licensed for the number of Azure Hybrid Benefit instances you are running, there is no need for any further action. If you discovered you can cover incremental VMs with the benefit, you may want to optimize your costs further by switching to running instances with the benefit vs full cost.

If you do not have enough eligible Windows Server licenses for the number of VMs already deployed, you either need to purchase additional Windows Server on-premises licenses covered with Software Assurance through one of the channels listed below, purchase Windows Server VMs at regular hourly rates or turn off the Hybrid Benefit functionality for some VMs. Please note that you may buy core licenses in the increment of 8 cores, to qualify for each additional Azure Hybrid Benefit VM.

Windows Server Software Assurance and/or Subscriptions are available for purchase through one of a combination of the following Microsoft licensing channels:

CHANNEL	OPEN	OVS	SELECT/ SELECT PLUS	MPSA	EA/EAS
Typical size (# of devices)	5-250	5-250	>250	>250	>500
SA / Subscription	Optional	Included	Optional	Optional	Included

Microsoft reserves the right to audit end customer at any time to verify eligibility for Azure Hybrid Benefit utilization.

Deployment guidance

We have enabled pre-built gallery images availability for all our customers who have eligible licenses, irrespective of where they bought them, as well as enabled partners to be able to perform the deployments on customers' behalf.

Please find the instructions for all available deployment options here, including:

- Detailed video highlighting the new deployment experience utilizing pre-built gallery images
- Detailed instructions on uploading a custom-built VM
- Detailed instructions on migrating existing virtual machines using Azure Site recovery using PowerShell.

Windows Server 2016 Activation

1/7/2019 • 5 minutes to read • Edit Online

The following information outlines initial planning considerations that you need to review for Key Management Services (KMS) activation involving Windows Server 2016. For information about KMS activation involving operating systems older than those listed here, see Step 1: Review and Select Activation Methods.

KMS uses a client-server model to active clients. KMS clients connect to a KMS server, called the KMS host, for activation. The KMS host must reside on your local network.

KMS hosts do not need to be dedicated servers, and KMS can be cohosted with other services. You can run a KMS host on any physical or virtual system that is running Windows 10, Windows Server 2016, Windows Server 2012 R2, Windows 8.1, or Windows Server 2012.

A KMS host running on Windows 10 or Windows 8.1 can only activate computers running client operating systems. The following table summarizes KMS host and client requirements for networks that include Windows Server 2016 and Windows 10 clients.

> **NOTE**
>
> **Note:** Updates might be required on the KMS server to support activation of any of these newer clients. If you receive activation errors, check that you have the appropriate updates listed below this table.

PRODUCT KEY GROUP	KMS CAN BE HOSTED ON	WINDOWS EDITIONS ACTIVATED BY THIS KMS HOST

PRODUCT KEY GROUP	KMS CAN BE HOSTED ON	WINDOWS EDITIONS ACTIVATED BY THIS KMS HOST
Volume License for Windows Server 2016	Windows Server 2012 Windows Server 2012 R2 Windows Server 2016	Windows Server Semi-Annual Channel Windows Server 2016 (all editions) Windows 10 LTSB (2015 and 2016) Windows 10 Professional Windows 10 Enterprise Windows 10 Pro for Workstations Windows 10 Education Windows Server 2012 R2 (all editions) Windows 8.1 Professional Windows 8.1 Enterprise Windows Server 2012 (all editions) Windows 2008 R2 (all editions) Windows 7 Professional Windows 7 Enterprise Windows 2008 (all editions) Windows Vista Business Windows Vista Enterprise
Volume license for Windows 10	Windows 7 Windows 8 Windows 8.1 Windows 10	Windows 10 Professional Windows 10 Professional N Windows 10 Enterprise Windows 10 Enterprise N Windows 10 Education Windows 10 Education N Windows 10 Enterprise LTSB (2015) Windows 10 Enterprise LTSB N (2015) Windows 10 Pro for Workstations Windows 8.1 Professional Windows 8.1 Enterprise Windows 7 Professional Windows 7 Enterprise

PRODUCT KEY GROUP	KMS CAN BE HOSTED ON	WINDOWS EDITIONS ACTIVATED BY THIS KMS HOST
Volume license for "Windows Server 2012 R2 for Windows 10"	Windows Server 2008 R2 Windows Server 2012 Standard Windows Server 2012 Datacenter Windows Server 2012 R2 Standard Windows Server 2012 R2 Datacenter	Windows 10 Professional Windows 10 Enterprise Windows 10 Enterprise LTSB (2015) Windows 10 Pro for Workstations Windows 10 Education Windows Server 2012 R2 (all editions) Windows 8.1 Professional Windows 8.1 Enterprise Windows Server 2012 (all editions) Windows 2008 R2 (all editions) Windows 7 Professional Windows 7 Enterprise

NOTE

Depending on which operating system your KMS server is running and which operating systems you want to activate, you might need to install one or more of these updates:

- Installations of KMS on Windows 7 or Windows Server 2008 R2 must be updated in order to support activation of clients running Windows 10. For more information, see Update that enables Windows 7 and Windows Server 2008 R2 KMS hosts to activate Windows 10.
- Installations of KMS on Windows Server 2012 must be updated in order to support activation of clients running Windows 10 and Windows Server 2016, or newer client or server operating systems. For more information, see July 2016 update rollup for Windows Server 2012.
- Installations of KMS on Windows 8.1 or Windows Server 2012 R2 must be updated in order to support activation of clients running Windows 10 and Windows Server 2016, or newer client or server operating systems. For more information, see July 2016 update rollup for Windows 8.1 and Windows Server 2012 R2.
- Windows Server 2008 R2 cannot be updated to support activation of clients running Windows Server 2016, or newer operating systems.

A single KMS host can support an unlimited number of KMS clients. If you have more than 50 clients, we recommend that you have at least two KMS hosts in case one of your KMS hosts becomes unavailable. Most organizations can operate with as few as two KMS hosts for their entire infrastructure.

Addressing KMS operational requirements

KMS can activate physical and virtual computers, but to qualify for KMS activation, a network must have a minimum number of computers (called the activation threshold). KMS clients activate only after this threshold is met. To ensure that the activation threshold is met, a KMS host counts the number of computers that are requesting activation on the network.

KMS hosts count the most recent connections. When a client or server contacts the KMS host, the host adds the machine ID to its count and then returns the current count value in its response. The client or server will activate if the count is high enough. Clients will activate if the count is 25 or higher. Servers and volume editions of Microsoft

Office products will activate if the count is five or greater. The KMS only counts unique connections from the past 30 days, and only stores the 50 most recent contacts.

KMS activations are valid for 180 days, a period known as the activation validity interval. KMS clients must renew their activation by connecting to the KMS host at least once every 180 days to stay activated. By default, KMS client computers attempt to renew their activation every seven days. After a client's activation is renewed, the activation validity interval begins again.

Addressing KMS functional requirements

KMS activation requires TCP/IP connectivity. KMS hosts and clients are configured by default to use Domain Name System (DNS). By default, KMS hosts use DNS dynamic update to automatically publish the information that KMS clients need to find and connect to them. You can accept these default settings, or if you have special network and security configuration requirements, you can manually configure KMS hosts and clients.

After the first KMS host is activated, the KMS key that is used on the first host can be used to activate up to five more KMS hosts on your network. After a KMS host is activated, administrators can reactivate the same host up to nine times with the same key.

If your organization needs more than six KMS hosts, you should request additional activations for your organization's KMS key—for example, if you have ten physical locations under one volume licensing agreement and you want each location to have a local KMS host.

> **NOTE**
> To request this exception, contact your Activation Call Center. For more information, see Microsoft Volume Licensing.

Computers that are running volume licensing editions of Windows 10, Windows Server 2016, Windows 8.1, Windows Server 2012 R2, Windows Server 2012, Windows 7, Windows Server 2008 R2 are, by default, KMS clients with no additional configuration needed.

If you are converting a computer from a KMS host, MAK, or retail edition of Windows to a KMS client, install the applicable KMS Client Setup Key. For more information, see KMS Client Setup Keys.

KMS Client Setup Keys

10/22/2018 • 4 minutes to read • Edit Online

> Applies To: Windows Server 2019, Windows Server Semi-Annual Channel, Windows Server 2016, Windows 10

Computers that are running volume licensed editions of Windows Server, Windows 10, Windows 8.1, Windows Server 2012 R2, Windows 8, Windows Server 2012, Windows 7, Windows Server 2008 R2, Windows Vista, and Windows Server 2008 are, by default, KMS clients with no additional configuration needed.

> **NOTE**
> In the tables that follow, "LTSC" stands for "Long-Term Servicing Channel," while "LTSB" refers to the "Long-Term Servicing Branch."

To use the keys listed here (which are GVLKs), you must first have a KMS host running in your deployment. If you haven't already configured a KMS host, see Deploy KMS Activation for steps to set one up.

If you are converting a computer from a KMS host, MAK, or retail edition of Windows to a KMS client, install the applicable setup key (GVLK) from the following tables. To install a client setup key, open an administrative command prompt on the client, type **slmgr /ipk <setup key>** and then press **Enter**.

IF YOU WANT TO...	...USE THESE RESOURCES
Activate Windows outside of a volume-activation scenario (that is, you're trying to activate a retail version of Windows), **these keys will not work**.	Use these links for retail versions of Windows:
Fix this error that you get when you try to activate a Windows 8.1, Windows Server 2012 R2 or newer system: "Error: 0xC004F050 The Software Licensing Service reported that the product key is invalid"...	Install this update on the KMS host if it is running Windows 8.1, Windows Server 2012 R2, Windows 8, or Windows Server 2012.

- Get Windows 10
- Get a new Windows product key
- Genuine Windows Help & How-to

> If you are running Windows Server 2008 R2 or Windows 7, be on the lookout for an update to support using those as KMS hosts for Windows 10 clients.

Windows Server Semi-Annual Channel versions

Windows Server, version 1809

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server Datacenter	6NMRW-2C8FM-D24W7-TQWMY-CWH2D
Windows Server Standard	N2KJX-J94YW-TQVFB-DG9YT-724CC

Windows Server, version 1803

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server Datacenter	2HXDN-KRXHB-GPYC7-YCKFJ-7FVDG
Windows Server Standard	PTXN8-JFHJM-4WC78-MPCBR-9W4KR

Windows Server, version 1709

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server Datacenter	6Y6KB-N82V8-D8CQV-23MJW-BWTG6
Windows Server Standard	DPCNP-XQFKJ-BJF7R-FRC8D-GF6G4

Windows Server LTSC/LTSB versions

Windows Server 2019

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2019 Datacenter	WMDGN-G9PQG-XVVXX-R3X43-63DFG
Windows Server 2019 Standard	N69G4-B89J2-4G8F4-WWYCC-J464C
Windows Server 2019 Essentials	WVDHN-86M7X-466P6-VHXV7-YY726

Windows Server 2016

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2016 Datacenter	CB7KF-BWN84-R7R2Y-793K2-8XDDG
Windows Server 2016 Standard	WC2BQ-8NRM3-FDDYY-2BFGV-KHKQY
Windows Server 2016 Essentials	JCKRF-N37P4-C2D82-9YXRT-4M63B

Windows 10, all supported Semi-Annual Channel versions

See the Windows lifecycle fact sheet for information about supported versions and end of service dates.

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 10 Pro	W269N-WFGWX-YVC9B-4J6C9-T83GX
Windows 10 Pro N	MH37W-N47XK-V7XM9-C7227-GCQG9
Windows 10 Pro Workstations	NRG8B-VKK3Q-CXVCJ-9G2XF-6Q84J
Windows 10 Pro Workstations N	9FNHH-K3HBT-3W4TD-6383H-6XYWF
Windows 10 Pro Education	6TP4R-GNPTD-KYYHQ-7B7DP-J447Y

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 10 Pro Education N	YVWGF-BXNMC-HTQYQ-CPQ99-66QFC
Windows 10 Education	NW6C2-QMPVW-D7KKK-3GKT6-VCFB2
Windows 10 Education N	2WH4N-8QGBV-H22JP-CT43Q-MDWWJ
Windows 10 Enterprise	NPPR9-FWDCX-D2C8J-H872K-2YT43
Windows 10 Enterprise N	DPH2V-TTNVB-4X9Q3-TJR4H-KHJW4
Windows 10 Enterprise G	YYVX9-NTFWV-6MDM3-9PT4T-4M68B
Windows 10 Enterprise G N	44RPN-FTY23-9VTTB-MP9BX-T84FV

Windows 10 LTSC/LTSB versions

Windows 10 LTSC 2019

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 10 Enterprise LTSC 2019	M7XTQ-FN8P6-TTKYV-9D4CC-J462D
Windows 10 Enterprise N LTSC 2019	92NFX-8DJQP-P6BBQ-THF9C-7CG2H

Windows 10 LTSB 2016

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 10 Enterprise LTSB 2016	DCPHK-NFMTC-H88MJ-PFHPY-QJ4BJ
Windows 10 Enterprise N LTSB 2016	QFFDN-GRT3P-VKWWX-X7T3R-8B639

Windows 10 LTSB 2015

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 10 Enterprise 2015 LTSB	WNMTR-4C88C-JK8YV-HQ7T2-76DF9
Windows 10 Enterprise 2015 LTSB N	2F77B-TNFGY-69QQF-B8YKP-D69TJ

Earlier versions of Windows Server

Windows Server 2012 R2

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2012 R2 Server Standard	D2N9P-3P6X9-2R39C-7RTCD-MDVJX
Windows Server 2012 R2 Datacenter	W3GGN-FT8W3-Y4M27-J84CP-Q3VJ9

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2012 R2 Essentials	KNC87-3J2TX-XB4WP-VCPJV-M4FWM

Windows Server 2012

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2012	BN3D2-R7TKB-3YPBD-8DRP2-27GG4
Windows Server 2012 N	8N2M2-HWPGY-7PGT9-HGDD8-GVGGY
Windows Server 2012 Single Language	2WN2H-YGCQR-KFX6K-CD6TF-84YXQ
Windows Server 2012 Country Specific	4K36P-JN4VD-GDC6V-KDT89-DYFKP
Windows Server 2012 Server Standard	XC9B7-NBPP2-83J2H-RHMBY-92BT4
Windows Server 2012 MultiPoint Standard	HM7DN-YVMH3-46JC3-XYTG7-CYQJJ
Windows Server 2012 MultiPoint Premium	XNH6W-2V9GX-RGJ4K-Y8X6F-QGJ2G
Windows Server 2012 Datacenter	48HP8-DN98B-MYWDG-T2DCC-8W83P

Windows Server 2008 R2

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2008 R2 Web	6TPJF-RBVHG-WBW2R-86QPH-6RTM4
Windows Server 2008 R2 HPC edition	TT8MH-CG224-D3D7Q-498W2-9QCTX
Windows Server 2008 R2 Standard	YC6KT-GKW9T-YTKYR-T4X34-R7VHC
Windows Server 2008 R2 Enterprise	489J6-VHDMP-X63PK-3K798-CPX3Y
Windows Server 2008 R2 Datacenter	74YFP-3QFB3-KQT8W-PMXWJ-7M648
Windows Server 2008 R2 for Itanium-based Systems	GT63C-RJFQ3-4GMB6-BRFB9-CB83V

Windows Server 2008

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Web Server 2008	WYR28-R7TFJ-3X2YQ-YCY4H-M249D
Windows Server 2008 Standard	TM24T-X9RMF-VWXK6-X8JC9-BFGM2
Windows Server 2008 Standard without Hyper-V	W7VD6-7JFBR-RX26B-YKQ3Y-6FFFJ
Windows Server 2008 Enterprise	YQGMW-MPWTJ-34KDK-48M3W-X4Q6V
Windows Server 2008 Enterprise without Hyper-V	39BXF-X8Q23-P2WWT-38T2F-G3FPG

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows Server 2008 HPC	RCTX3-KWVHP-BR6TB-RB6DM-6X7HP
Windows Server 2008 Datacenter	7M67G-PC374-GR742-YH8V4-TCBY3
Windows Server 2008 Datacenter without Hyper-V	22XQ2-VRXRG-P8D42-K34TD-G3QQC
Windows Server 2008 for Itanium-Based Systems	4DWFP-JF3DJ-B7DTH-78FJB-PDRHK

Earlier versions of Windows

Windows 8.1

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 8.1 Pro	GCRJD-8NW9H-F2CDX-CCM8D-9D6T9
Windows 8.1 Pro N	HMCNV-VVBFX-7HMBH-CTY9B-B4FXY
Windows 8.1 Enterprise	MHF9N-XY6XB-WVXMC-BTDCT-MKKG7
Windows 8.1 Enterprise N	TT4HM-HN7YT-62K67-RGRQJ-JFFXW

Windows 8

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 8 Pro	NG4HW-VH26C-733KW-K6F98-J8CK4
Windows 8 Pro N	XCVCF-2NXM9-723PB-MHCB7-2RYQQ
Windows 8 Enterprise	32JNW-9KQ84-P47T8-D8GGY-CWCK7
Windows 8 Enterprise N	JMNMF-RHW7P-DMY6X-RF3DR-X2BQT

Windows 7

OPERATING SYSTEM EDITION	KMS CLIENT SETUP KEY
Windows 7 Professional	FJ82H-XT6CR-J8D7P-XQJJ2-GPDD4
Windows 7 Professional N	MRPKT-YTG23-K7D7T-X2JMM-QY7MG
Windows 7 Professional E	W82YF-2Q76Y-63HXB-FGJG9-GF7QX
Windows 7 Enterprise	33PXH-7Y6KF-2VJC9-XBBR8-HVTHH
Windows 7 Enterprise N	YDRBP-3D83W-TY26F-D46B2-XCKRJ
Windows 7 Enterprise E	C29WB-22CC8-VJ326-GHFJW-H9DH4

See also

- Plan for Volume Activation

Install Nano Server

9/21/2018 • 4 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

Windows Server 2016 offers a new installation option: Nano Server. Nano Server is a remotely administered server operating system optimized for private clouds and datacenters. It is similar to Windows Server in Server Core mode, but significantly smaller, has no local logon capability, and only supports 64-bit applications, tools, and agents. It takes up far less disk space, sets up significantly faster, and requires far fewer updates and restarts than Windows Server. When it does restart, it restarts much faster. The Nano Server installation option is available for Standard and Datacenter editions of Windows Server 2016.

Nano Server is ideal for a number of scenarios:

- As a "compute" host for Hyper-V virtual machines, either in clusters or not
- As a storage host for Scale-Out File Server.
- As a DNS server
- As a web server running Internet Information Services (IIS)
- As a host for applications that are developed using cloud application patterns and run in a container or virtual machine guest operating system

Important differences in Nano Server

Because Nano Server is optimized as a lightweight operating system for running “cloud-native” applications based on containers and micro-services or as an agile and cost-effective datacenter host with a dramatically smaller footprint, there are important differences in Nano Server versus Server Core or Server with Desktop Experience installations:

- Nano Server is "headless;" there is no local logon capability or graphical user interface.
- Only 64-bit applications, tools, and agents are supported.
- Nano Server cannot serve as an Active Directory domain controller.
- Group Policy is not supported. However, you can use Desired State Configuration to apply settings at scale.
- Nano Server cannot be configured to use a proxy server to access the internet.
- NIC Teaming (specifically, load balancing and failover, or LBFO) is not supported. Switch-embedded teaming (SET) is supported instead.
- System Center Configuration Manager and System Center Data Protection Manager are not supported.
- Best Practices Analyzer (BPA) cmdlets and BPA integration with Server Manager are not supported.
- Nano Server does not support virtual host bus adapters (HBAs).
- Nano Server does not need to be activated with a product key. When functioning as a Hyper-V host, Nano Server does not support Automatic Virtual Machine Activation (AVMA). Virtual machines running on a Nano Server host can be activated using Key Management Service (KMS) with a generic volume license key or using

Active Directory-based activation.

- The version of Windows PowerShell provided with Nano Server has important differences. For details, see PowerShell on Nano Server.
- Nano Server is supported only on the Current Branch for Business (CBB) model--there is no Long-Term Servicing Branch (LTSB) release for Nano Server at this time. See the following subsection for more information.

Current Branch for Business

Nano Server is serviced with a more active model, called Current Branch for Business (CBB), in order to support customers who are moving at a "cloud cadence," using rapid development cycles. In this model, feature update releases of Nano Server are expected two to three times per year. This model requires Software Assurance for Nano Servers deployed and operated in production. To maintain support, administrators must stay no more than two CBB releases behind. However, these releases do not auto-update existing deployments; administrators perform manual installation of a new CBB release at their convenience. For some additional information, see Windows Server 2016 new Current Branch for Business servicing option.

The Server Core and Server with Desktop Experience installation options are still serviced on the Long-Term Servicing Branch (LTSB) model, comprising 5 years of mainstream support and 5 years of extended support.

Installation scenarios

Evaluation

You can obtain a 180-day-licensed evaluation copy of Windows Server from Windows Server Evaluations. To try out Nano Server, choose the **Nano Server | 64-bit EXE option**, and then come back to either Nano Server Quick Start or Deploy Nano Server to get started.

Clean installation

Because you install Nano Server by configuring a VHD, a clean installation is the quickest and simplest deployment method.

- To get started quickly with a basic deployment of Nano Server using DHCP to obtain an IP address, see the Nano Server Quick Start
- If you're already familiar with the basics of Nano Server, the more detailed topics starting with Deploy Nano Server offer a full set of instructions for customizing images, working with domains, installing packages for server roles and other features both online and offline, and much more.

> **IMPORTANT**
> Once Setup has completed and immediately after you have installed all of the server roles and features you need, check for and install updates available for Windows Server 2016. For Nano Server, see the "Managing updates in Nano Server" section of Manage Nano Server.

Upgrade

Since Nano Server is new for Windows Server 2016, there isn't an upgrade path from older operating system versions to Nano Server.

Migration

Since Nano Server is new for Windows Server 2016, there isn't migration path from older operating system versions to Nano Server.

If you need a different installation option, you can head back to the main Windows Server 2016 page

Changes to Nano Server in Windows Server Semi-Annual Channel

8/8/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server, Semi-Annual Channel

As described in Window Server Semi-Annual Channel Overview, Windows Server, version 1803 is the latest release in the Semi-Annual Channel.

If you're already running Nano Server, this servicing model will be familiar, since it was formerly serviced by the Current Branch for Business (CBB) model. Windows Server's new Semi-Annual Channel is just a new name for the same model. In this model, feature update releases of Nano Server are expected two to three times per year.

However, with this release of Windows Server, version 1803, Nano Server is available only as a **container base OS image**. You must run it as a container in a container host, such as a Server Core installation of Windows Server. Running a container based on Nano Server in this release differs from earlier releases in these ways:

- Nano Server has been optimized for .NET Core applications.
- Nano Server is even smaller than the Windows Server 2016 version.
- PowerShell Core, .NET Core, and WMI are no longer included by default, but you can include PowerShell Core and .NET Core container packages when building your container.
- There is no longer a servicing stack included in Nano Server. Microsoft publishes an updated Nano container to Docker Hub that you redeploy.
- You troubleshoot the new Nano Container by using Docker.
- You can now run Nano containers on IoT Core.

Related topics

When the Insider Program starts, find more information in the Windows Container Documentation.

Nano Server Quick Start

6/20/2018 • 5 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

Follow the steps in this section to get started quickly with a basic deployment of Nano Server using DHCP to obtain an IP address. You can run a Nano Server VHD either in a virtual machine or boot to it on a physical computer; the steps are slightly different.

Once you've tried out the basics with these quick-start steps, you can find details of creating your own custom images, package management by several methods, domain operations, and much more in Deploy Nano Server.

Nano Server in a virtual machine

Follow these steps to create a Nano Server VHD that will run in a virtual machine.

To quickly deploy Nano Server in a virtual machine

1. Copy *NanoServerImageGenerator* folder from the \NanoServer folder in the Windows Server 2016 ISO to a folder on your hard drive.
2. Start Windows PowerShell as an administrator, change directory to the folder where you have placed the NanoServerImageGenerator folder and then import the module with `Import-Module .\NanoServerImageGenerator -Verbose`

   NOTE

   You might have to adjust the Windows PowerShell execution policy. `Set-ExecutionPolicy RemoteSigned` should work well.

3. Create a VHD for the Standard edition that sets a computer name and includes the Hyper-V **guest drivers** by running the following command which will prompt you for an administrator password for the new VHD:

   ```
   New-NanoServerImage -Edition Standard -DeploymentType Guest -MediaPath <path to root of media> -BasePath
   .\Base -TargetPath .\NanoServerVM\NanoServerVM.vhd -ComputerName <computer name>
   ```

   where

   - **-MediaPath <path to root of media>** specifies a path to the root of the contents of the Windows Server 2016 ISO. For example if you have copied the contents of the ISO to d:\TP5ISO you would use that path.
   - **-BasePath** (optional) specifies a folder that will be created to copy the Nano Server WIM and packages to.
   - **-TargetPath** specifies a path, including the filename and extension, where the resulting VHD or VHDX will be created.

- **Computer_name** specifies the computer name that the Nano Server virtual machine you are creating will have.

Example:

```
New-NanoServerImage -Edition Standard -DeploymentType Guest -MediaPath f:\ -BasePath .\Base -TargetPath
.\Nano1\Nano.vhd -ComputerName Nano1
```

This example creates a VHD from an ISO mounted as f:\. When creating the VHD it will use a folder called Base in the same directory where you ran New-NanoServerImage; it will place the VHD (called Nano.vhd) in a folder called Nano1 in the folder from where the command is run. The computer name will be Nano1. The resulting VHD will contain the Standard edition of Windows Server 2016 and will be suitable for Hyper-V virtual machine deployment. If you want a Generation 1 virtual machine, create a VHD image by specifying a **.vhd** extension for -TargetPath. For a Generation 2 virtual machine, create a VHDX image by specifying a **.vhdx** extension for -TargetPath. You can also directly generate a WIM file by specifying a **.wim** extension for -TargetPath.

> **NOTE**
> New-NanoServerImage is supported on Windows 8.1, Windows 10, Windows Server 2012 R2, and Windows Server 2016.

4. In Hyper-V Manager, create a new virtual machine and use the VHD created in Step 3.
5. Boot the virtual machine and in Hyper-V Manager connect to the virtual machine.
6. Log on to the Recovery Console (see the "Nano Server Recovery Console" section in this guide), using the administrator and password you supplied while running the script in Step 3.

> **NOTE**
> The Recovery Console only supports basic keyboard functions. Keyboard lights, 10-key sections, and keyboard layout switching such as caps lock and number lock are not supported.

7. Obtain the IP address of the Nano Server virtual machine and use Windows PowerShell remoting or other remote management tool to connect to and remotely manage the virtual machine.

Nano Server on a physical computer

You can also create a VHD that will run Nano Server on a physical computer, using the pre-installed device drivers. If your hardware requires a driver that is not already provided in order to boot or connect to a network, follow the steps in the "Adding Additional Drivers" section of this guide.

To quickly deploy Nano Server on a physical computer

1. Copy *NanoServerImageGenerator* folder from the \NanoServer folder in the Windows Server 2016 ISO to a folder on your hard drive.
2. Start Windows PowerShell as an administrator, change directory to the folder where you have placed the NanoServerImageGenerator folder and then import the module with `Import-Module .\NanoServerImageGenerator -Verbose`

> **NOTE**
> You might have to adjust the Windows PowerShell execution policy. `Set-ExecutionPolicy RemoteSigned` should work well.

1. Create a VHD that sets a computer name and includes the OEM drivers and Hyper-V by running the following command which will prompt you for an administrator password for the new VHD:

```
New-NanoServerImage -Edition Standard -DeploymentType Host -MediaPath <path to root of media> -BasePath
.\Base -TargetPath .\NanoServerPhysical\NanoServer.vhd -ComputerName <computer name> -OEMDrivers -Compute
-Clustering
```

   where

   - **-MediaPath <path to root of media>** specifies a path to the root of the contents of the Windows Server 2016 ISO. For example if you have copied the contents of the ISO to d:\TP5ISO you would use that path.
   - **BasePath** specifies a folder that will be created to copy the Nano Server WIM and packages to. (This parameter is optional.)
   - **TargetPath** specifies a path, including the filename and extension, where the resulting VHD or VHDX will be created.
   - **Computer_name** is the computer name for the Nano Server you are creating.

   Example:

```
New-NanoServerImage -Edition Standard -DeploymentType Host -MediaPath F:\ -BasePath .\Base -TargetPath
.\Nano1\NanoServer.vhd -ComputerName Nano-srv1 -OEMDrivers -Compute -Clustering
```

   This example creates a VHD from an ISO mounted as F:\. When creating the VHD it will use a folder called Base in the same directory where you ran New-NanoServerImage; it will place the VHD in a folder called Nano1 in the folder from where the command is run. The computer name will be Nano-srv1 and will have OEM drivers installed for most common hardware and has the Hyper-V role and clustering feature enabled. The Standard Nano edition is used.

2. Log in as an administrator on the physical server where you want to run the Nano Server VHD.
3. Copy the VHD that this script creates to the physical computer and configure it to boot from this new VHD. To do that, follow these steps:

   a. Mount the generated VHD. In this example, it's mounted under D:\.

   b. Run **bcdboot d:\windows**.

   c. Unmount the VHD.

4. Boot the physical computer into the Nano Server VHD.
5. Log on to the Recovery Console (see the "Nano Server Recovery Console" section in this guide), using the administrator and password you supplied while running the script in Step 3.

   > **NOTE**
   > The Recovery Console only supports basic keyboard functions. Keyboard lights, 10-key sections, and keyboard layout switching such as caps lock and number lock are not supported.

6. Obtain the IP address of the Nano Server computer and use Windows PowerShell remoting or other remote management tool to connect to and remotely manage the virtual machine.

Deploy Nano Server

11/6/2018 • 28 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

This topic covers information you need to deploy Nano Server images that are more customized to your needs compared to the simple examples in the Nano Server Quick Start topic. You'll find information about making a custom Nano Server image with exactly the features you want, installing Nano Server images from VHD or WIM, editing files, working with domains, dealing with packages by several methods, and working with server roles.

Nano Server Image Builder

The Nano Server Image Builder is a tool that helps you create a custom Nano Server image and bootable USB media with the aid of a graphical interface. Based on the inputs you provide, it generates reusable PowerShell scripts that allow you easily automate consistent installations of Nano Server running either Windows Server 2016 Datacenter or Standard editions.

Obtain the tool from the Download Center.

The tool also requires Windows Assessment and Deployment Kit (ADK).

Nano Server Image Builder creates customized Nano Server images in VHD, VHDX or ISO formats and can create bootable USB media to deploy Nano server or detect the hardware configuration of a server. It also can do the following:

- Accept the license terms
- Create VHD, VHDX or ISO formats
- Add server roles
- Add device drivers
- Set machine name, administrator password, logfile path, and timezone
- Join a domain by using an existing Active Directory account or a harvested domain-join blob
- Enable WinRM for communication outside the local subnet
- Enable Virtual LAN IDs and configure static IP addresses
- Inject new servicing packages on the fly
- Add a setupcomplete.cmd or other customer scripts to run after the unattend.xml is processed
- Enable Emergency Management Services (EMS) for serial port console access
- Enable development services to enable test signed drivers and unsigned applications, PowerShell default shell
- Enable debugging over serial, USB, TCP/IP, or IEEE 1394 protocols
- Create USB media using WinPE that will partition the server and install the Nano image
- Create USB media using WinPE that will detect your existing Nano Server hardware configuration and report all the details in a log and on-screen. This includes network adapters, MAC addresses, and firmware Type (BIOS or UEFI). The detection process will also list all of the volumes on the system and the devices that do not have a driver included in the Server Core drivers package.

If any of these are unfamiliar to you, review the remainder of this topic and the other Nano Server topics so that you'll be prepared to provide the tool with the information it will need.

Creating a custom Nano Server image

For Windows Server 2016, Nano Server is distributed on the physical media, where you will find a **NanoServer** folder; this contains a .wim image and a subfolder called **Packages**. It is these package files that you use to add server roles and features to the VHD image, which you then boot to.

You can also find and install these packages with the NanoServerPackage provider of PackageManagement (OneGet) PowerShell module. See the "Installing roles and features online" section of this topic.

This table shows the roles and features that are available in this release of Nano Server, along with the Windows PowerShell options that will install the packages for them. Some packages are installed directly with their own Windows PowerShell switches (such as -Compute); others you install by passing package names to the -Package parameter, which you can combine in a comma-separated list. You can dynamically list available packages using the Get-NanoServerPackage cmdlet.

ROLE OR FEATURE	OPTION
Hyper-V role (including NetQoS)	-Compute
Failover Clustering and other components, detailed after this table	-Clustering
Basic drivers for a variety of network adapters and storage controllers. This is the same set of drivers included in a Server Core installation of Windows Server 2016.	-OEMDrivers
File Server role and other storage components, detailed after this table	-Storage
Windows Defender, including a default signature file	-Defender
Reverse forwarders for application compatibility, for example common application frameworks such as Ruby, Node.js, etc.	Now included by default
DNS Server role	-Package Microsoft-NanoServer-DNS-Package
PowerShell Desired State Configuration (DSC)	-Package Microsoft-NanoServer-DSC-Package **Note:** For full details, see Using DSC on Nano Server.
Internet Information Server (IIS)	-Package Microsoft-NanoServer-IIS-Package **Note:** See IIS on Nano Server for details about working with IIS.
Host support for Windows Containers	-Containers
System Center Virtual Machine Manager agent	-Package Microsoft-NanoServer-SCVMM-Package -Package Microsoft-NanoServer-SCVMM-Compute-Package **Note:** Use the SCVMM Compute package only if you are monitoring Hyper-V. For hyper-converged deployments in VMM, you should also specify the -Storage parameter. For more details, see the VMM documentation.

ROLE OR FEATURE	OPTION
System Center Operations Manager agent	Installed separately. See the System Center Operations Manager documentation for more details at https://technet.microsoft.com/system-center-docs/om/manage/install-agent-on-nano-server.
Data Center Bridging (including DCBQoS)	-Package Microsoft-NanoServer-DCB-Package
Deploying on a virtual machine	-Package Microsoft-NanoServer-Guest-Package
Deploying on a physical machine	- Package Microsoft-NanoServer-Host-Package
BitLocker, trusted platform module (TPM), volume encryption, platform identification, cryptography providers, and other functionality related to secure startup	-Package Microsoft-NanoServer-SecureStartup-Package
Hyper-V support for Shielded VMs	-Package Microsoft-NanoServer-ShieldedVM-Package **Note:** This package is only available for the Datacenter edition of Nano Server.
Simple Network Management Protocol (SNMP) agent	-Package Microsoft-NanoServer-SNMP-Agent-Package.cab **Note:** Not included with Windows Server 2016 installation media. Available online only. See Installing roles and features online for details.
IPHelper service which provides tunnel connectivity using IPv6 transition technologies (6to4, ISATAP, Port Proxy, and Teredo), and IP-HTTPS	-Package Microsoft-NanoServer-IPHelper-Service-Package.cab **Note:** Not included with Windows Server 2016 installation media. Available online only. See Installing roles and features online for details.

NOTE

When you install packages with these options, a corresponding language pack is also installed based on selected server media locale. You can find the available language packs and their locale abbreviations in the installation media in subfolders named for the locale of the image.

NOTE

When you use the -Storage parameter to install File Services, File Services is not actually enabled. Enable this feature from a remote computer with Server Manager.

Failover Clustering items installed by the -Clustering parameter

- Failover Clustering role
- VM Failover Clustering
- Storage Spaces Direct (S2D)
- Storage Quality of Service
- Volume Replication Clustering
- SMB Witness Service

File and storage items installed by the -Storage parameter

- File Server role
- Data Deduplication

- Multipath I/O, including a driver for Microsoft Device-Specific Module (MSDSM)
- ReFS (v1 and v2)
- iSCSI Initiator (but not iSCSI Target)
- Storage Replica
- Storage Management Service with SMI-S support
- SMB Witness Service
- Dynamic Volumes
- Basic Windows storage providers (for Windows Storage Management)

Installing a Nano Server VHD

This example creates a GPT-based VHDX image with a given computer name and including Hyper-V guest drivers, starting with Nano Server installation media on a network share. In an elevated Windows PowerShell prompt, start with this cmdlet:

```
Import-Module <Server media location>\NanoServer\NanoServerImageGenerator; New-NanoServerImage -DeploymentType
Guest -Edition Standard -MediaPath \\Path\To\Media\server_en-us -BasePath .\Base -TargetPath
.\FirstStepsNano.vhdx -ComputerName FirstStepsNano
```

The cmdlet will accomplish all of these tasks:

1. Select Standard as a base edition
2. Prompt you for the Administrator password
3. Copy installation media from \\Path\To\Media\server_en-us into .\Base
4. Convert the WIM image to a VHD. (The file extension of the target path argument determines whether it creates an MBR-based VHD for Generation 1 virtual machines versus a GPT-based VHDX for Generation 2 virtual machines.)
5. Copy the resulting VHD into .\FirstStepsNano.vhdx
6. Set the Administrator password for the image as specified
7. Set the computer name of the image to FirstStepsNano
8. Install the Hyper-V guest drivers

All of this results in an image of .\FirstStepsNano.vhdx.

The cmdlet generates a log as it runs and will let you know where this log is located once it is finished. The WIM-to-VHD conversion accomplished by the companion script generates its own log in %TEMP%\Convert-WindowsImage\<GUID> (where <GUID> is a unique identifier per conversion session).

As long as you use the same base path, you can omit the media path parameter every time you run this cmdlet, since it will use cached files from the base path. If you don't specify a base path, the cmdlet will generate a default one in the TEMP folder. If you want to use different source media, but the same base path, you should specify the media path parameter, however.

> **NOTE**
> You now have the option to specify the Nano Server edition to build either the Standard or Datacenter edition. Use the -Edition parameter to specify *Standard* or *Datacenter* editions.

Once you have an existing image, you can modify it as needed using the *Edit-NanoServerImage* cmdlet.

If you do not specify a computer name, a random name will be generated.

Installing a Nano Server WIM

1. Copy the *NanoServerImageGenerator* folder from the \NanoServer folder in the Windows Server 2016 ISO a local folder on your computer.
2. Start Windows PowerShell as an administrator, change directory to the folder where you placed the NanoServerImageGenerator folder and then import the module with `Import-Module .\NanoServerImageGenerator -Verbose`.

   > **NOTE**
   > You might have to adjust the Windows PowerShell execution policy. `Set-ExecutionPolicy RemoteSigned` should work well.

To create a Nano Server image to serve as a Hyper-V host, run the following:

```
New-NanoServerImage -Edition Standard -DeploymentType Host -MediaPath <path to root of media> -BasePath .\Base -
TargetPath .\NanoServerPhysical\NanoServer.wim -ComputerName <computer name> -OEMDrivers -Compute -Clustering
```

Where

- -MediaPath is the root of the DVD media or ISO image containing Windows Server 2016 .
- -BasePath will contain a copy of the Nano Server binaries, so you can use New-NanoServerImage -BasePath without having to specify -MediaPath in future runs.
- -TargetPath will contain the resulting .wim file containing the roles & features you selected. Make sure to specify the .wim extension.
- -Compute adds the Hyper-V role.
- -OemDrivers adds a number of common drivers.

You will be prompted to enter an administrator password.

For more information, run `Get-Help New-NanoServerImage -Full`.

Boot into WinPE and ensure that the .wim file just created is accessible from WinPE. (You could, for example, copy the .wim file to a bootable WinPE image on a USB flash drive.)

Once WinPE boots, use Diskpart.exe to prepare the target computer's hard drive. Run the following Diskpart commands (modify accordingly, if you're not using UEFI & GPT):

> **WARNING**
> These commands will delete all data on the hard drive.

Diskpart.exe
Select disk 0
Clean
Convert GPT
Create partition efi size=100
Format quick FS=FAT32 label="System"
Assign letter="s"
Create partition msr size=128
Create partition primary
Format quick FS=NTFS label="NanoServer"
Assign letter="n"
List volume
Exit

Apply the Nano Server image (adjust the path of the .wim file):

Dism.exe /apply-image /imagefile:.\NanoServer.wim /index:1 /applydir:n:
Bcdboot.exe n:\Windows /s s:

Remove the DVD media or USB drive and reboot your system with **Wpeutil.exe Reboot**

Editing files on Nano Server locally and remotely

In either case, connect to Nano Server, such as with Windows PowerShell remoting.

Once you've connected to Nano Server, you can edit a file residing on your local computer by passing the file's relative or absolute path to the psEdit command, for example:
`psEdit C:\Windows\Logs\DISM\dism.log` or `psEdit .\myScript.ps1`

Edit a file residing on the remote Nano Server by starting a remote session with `Enter-PSSession -ComputerName "192.168.0.100" -Credential ~\Administrator` and then passing the file's relative or absolute path to the psEdit command like this:
`psEdit C:\Windows\Logs\DISM\dism.log`

Installing roles and features online

> **NOTE**
> If you install an optional Nano Server package from media or online repository, it won't have recent security fixes included. To avoid a version mismatch between the optional packages and base operating system, you should install the latest cumulative update immediately after installing any optional packages and **before** restarting the server.

Installing roles and features from a package repository

You can find and install Nano Server packages from the online package repository by using the NanoServerPackage provider of the PackageManagement PowerShell module. To install this provider, use these cmdlets:

```
Install-PackageProvider NanoServerPackage
Import-PackageProvider NanoServerPackage
```

> **NOTE**
> If you experience errors when running Install-PackageProvider, check that you have installed the latest cumulative update (KB3206632 or later), or use Save-Module as follows:

```
Save-Module -Path "$Env:ProgramFiles\WindowsPowerShell\Modules\" -Name NanoServerPackage -MinimumVersion
1.0.1.0
Import-PackageProvider NanoServerPackage
```

Once this provider is installed and imported, you can search for, download, and install Nano Server packages using cmdlets designed specifically for working with Nano Server packages:

```
Find-NanoServerPackage
Save-NanoServerPackage
Install-NanoServerPackage
```

You can also use the generic PackageManagement cmdlets and specify the NanoServerPackage provider:

```
Find-Package -ProviderName NanoServerPackage
Save-Package -ProviderName NanoServerPackage
Install-Package -ProviderName NanoServerPackage
Get-Package -ProviderName NanoServerPackage
```

To use any of these cmdlets with Nano Server packages on Nano Server, add `-ProviderName NanoServerPackage`. If you don't add the -ProviderName parameter, PackageManagement will iterate all of the providers. For more details on these cmdlets, run `Get-Help <cmdlet>`. Here are some common usage examples:

Searching for Nano Server packages

You can use either `Find-NanoServerPackage` or `Find-Package -ProviderName NanoServerPackage` to search for and return a list of Nano Server packages that are available in the online repository. For example, you can get a list of all the latest packages:

```
Find-NanoServerPackage
```

Running `Find-Package -ProviderName NanoServerPackage -DisplayCulture` displays all available cultures.

If you need a specific locale version, such as US English, you could use `Find-NanoServerPackage -Culture en-us` or `Find-Package -ProviderName NanoServerPackage -Culture en-us` or `Find-Package -Culture en-us -DisplayCulture`.

To find a specific package by package name, use the -Name parameter. This parameter also accepts wildcards. For example, to find all packages with VMM in the name, use `Find-NanoServerPackage -Name *VMM*` or `Find-Package -ProviderName NanoServerPackage -Name *VMM*`.

You can find a particular version with the -RequiredVersion, -MinimumVersion, or -MaximumVersion parameters. To find all available versions, use -AllVersions. Otherwise, only the latest version is returned. For example: `Find-NanoServerPackage -Name *VMM* -RequiredVersion 10.0.14393.0`. Or, for all versions: `Find-Package -ProviderName NanoServerPackage -Name *VMM* -AllVersions`

Installing Nano Server packages

You can install a Nano Server package (including its dependency packages, if any) to Nano Server either locally or an offline image with either `Install-NanoServerPackage` or `Install-Package -ProviderName NanoServerPackage`. Both of these accept input from the pipeline.

To install the latest version of a Nano Server package to an online Nano Server, use either `Install-NanoServerPackage -Name Microsoft-NanoServer-Containers-Package` or `Install-Package -Name Microsoft-NanoServer-Containers-Package`. PackageManagement will use the culture of the Nano Server.

You can install a Nano Server package to an offline image while specifying a particular version and culture, like this:

```
Install-NanoServerPackage -Name Microsoft-NanoServer-DCB-Package -Culture de-de -RequiredVersion 10.0.14393.0 -
ToVhd C:\MyNanoVhd.vhd
```

or:

```
Install-Package -Name Microsoft-NanoServer-DCB-Package -Culture de-de -RequiredVersion 10.0.14393.0 -ToVhd
C:\MyNanoVhd.vhd
```

Here are some examples of pipelining package search results to the installation cmdlet:

`Find-NanoServerPackage *dcb* | Install-NanoServerPackage` finds any packages with "dcb" in the name and then installs them.

`Find-Package *nanoserver-compute-* | Install-Package` finds packages with "nanoserver-compute-" in the name and installs them.

`Find-NanoServerPackage -Name *nanoserver-compute* | Install-NanoServerPackage -ToVhd C:\MyNanoVhd.vhd` finds packages with "compute" in the name and installs them to an offline image.

`Find-Package -ProviderName NanoserverPackage *nanoserver-compute-* | Install-Package -ToVhd C:\MyNanoVhd.vhd` does the same thing with any package that has "nanoserver-compute-" in the name.

Downloading Nano Server packages

`Save-NanoServerPackage` or `Save-Package` allow you to download packages and save them without installing them. Both cmdlets accept input from the pipeline.

For example, to download and save a Nano Server package to a directory that matches the wildcard path, use `Save-NanoServerPackage -Name Microsoft-NanoServer-DNS-Package -Path C:\` In this example, -Culture wasn't specified, so the culture of the local machine will be used. No version was specified, so the latest version will be saved.

```
Save-Package -ProviderName NanoServerPackage -Name Microsoft-NanoServer-IIS-Package -Path C:\ -Culture it-IT -
MinimumVersion 10.0.14393.0
```

saves a particular version and for the Italian language and locale.

You can send search results through the pipeline as in these examples:

```
Find-NanoServerPackage -Name *containers* -MaximumVersion 10.2 -MinimumVersion 1.0 -Culture es-ES | Save-
NanoServerPackage -Path C:\
```

or

```
Find-Package -ProviderName NanoServerPackage -Name *shield* -Culture es-ES | Save-Package -Path
```

Inventory installed packages

You can discover which Nano Server packages are installed with `Get-Package`. For example, see which packages are on Nano Server with `Get-Package -ProviderName NanoserverPackage`.

To check the Nano Server packages that are installed in an offline image, run `Get-Package -ProviderName NanoserverPackage -FromVhd C:\MyNanoVhd.vhd`.

Installing roles and features from local source

Though offline installation of server roles and other packages is recommended, you might need to install them online (with the Nano Server running) in container scenarios. To do this, follow these steps:

1. Copy the Packages folder from the installation media locally to the running Nano Server (for example, to C:\packages).
2. Create a new Unattend.xml file on another computer and then copy it to Nano Server. You can copy and paste this XML content into the XML file you created (this example shows installing the IIS package):

```
<?xml version="1.0" encoding="utf-8"?>
    <unattend xmlns="urn:schemas-microsoft-com:unattend">
    <servicing>
        <package action="install">
            <assemblyIdentity name="Microsoft-NanoServer-IIS-Feature-Package" version="10.0.14393.0"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" />
            <source location="c:\packages\Microsoft-NanoServer-IIS-Package.cab" />
        </package>
        <package action="install">
            <assemblyIdentity name="Microsoft-NanoServer-IIS-Feature-Package" version="10.0.14393.0"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="en-US" />
            <source location="c:\packages\en-us\Microsoft-NanoServer-IIS-Package_en-us.cab" />
        </package>
    </servicing>
    <cpi:offlineImage cpi:source="" xmlns:cpi="urn:schemas-microsoft-com:cpi" />
</unattend>
```

1. In the new XML file you created (or copied), edit C:\packages to the directory you copied the content of Packages to.

2. Switch to the directory with the newly created XML file and run

   dism /online /apply-unattend:.\unattend.xml

3. Confirm that the package and its associated language pack is installed correctly by running:

   dism /online /get-packages

   You should see "Package Identity : Microsoft-NanoServer-IIS-Package~31bf3856ad364e35~amd64~en-US~10.0.10586.0" listed twice, once for Release Type : Language Pack and once for Release Type : Feature Pack.

Customizing an existing Nano Server VHD

You can change the details of an existing VHD by using the Edit-NanoServerImage cmdlet, as in this example:

```
Edit-NanoServerImage -BasePath .\Base -TargetPath .\BYOVHD.vhd
```

This cmdlet does the same things as New-NanoServerImage, but changes the existing image instead of converting a WIM to a VHD. It supports the same parameters as New-NanoServerImage with the exception of -MediaPath and -MaxSize, so the initial VHD must have been created with those parameters before you can make changes with Edit-NanoServerImage.

Additional tasks you can accomplish with New-NanoServerImage and Edit-NanoServerImage

Joining domains

New-NanoServerImage offers two methods of joining a domain; both rely on offline domain provisioning, but one harvests a blob to accomplish the join. In this example, the cmdlet harvests a domain blob for the Contoso domain from the local computer (which of course must be part of the Contoso domain), then it performs offline provisioning of the image using the blob:

```
New-NanoServerImage -Edition Standard -DeploymentType Host -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\JoinDomHarvest.vhdx -ComputerName JoinDomHarvest -DomainName Contoso
```

When this cmdlet completes, you should find a computer named "JoinDomHarvest" in the Active Directory computer list.

You can also use this cmdlet on a computer that is not joined to a domain. To do this, harvest a blob from any computer that is joined to the domain, and then provide the blob to the cmdlet yourself. Note that when you harvest such a blob from another computer, the blob already includes that computer's name--so if you try to add the *-ComputerName* parameter, an error will result.

You can harvest the blob with this command:

djoin
/Provision
/Domain Contoso
/Machine JoiningDomainsNoHarvest
/SaveFile JoiningDomainsNoHarvest.djoin

Run New-NanoServerImage using the harvested blob:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\JoinDomNoHrvest.vhd -DomainBlobPath .\Path\To\Domain\Blob\JoinDomNoHrvestContoso.djoin
```

In the event that you already have a node in the domain with the same computer name as your future Nano Server, you could reuse the computer name by adding the `-ReuseDomainNode` parameter.

Adding additional drivers

Nano Server offers a package that includes a set of basic drivers for a variety of network adapters and storage controllers; it's possible that drivers for your network adapters might not be included. You can use these steps to find drivers in a working system, extract them, and then add them to the Nano Server image.

1. Install Windows Server 2016 on the physical computer where you will run Nano Server.
2. Open Device Manager and identify devices in the following categories:
3. Network adapters
4. Storage controllers
5. Disk drives
6. For each device in these categories, right-click the device name, and click **Properties**. In the dialog that opens, click the **Driver** tab, and then click **Driver Details**.
7. Note the filename and path of the driver file that appears. For example, let's say the driver file is e1i63x64.sys, which is in C:\Windows\System32\Drivers.
8. In a command prompt, search for the driver file and search for all instances with dir e1i*.sys /s /b. In this example, the driver file is also present in the path C:\Windows\System32\DriverStore\FileRepository\net1ic64.inf_amd64_fafa7441408bbecd\e1i63x64.sys.
9. In an elevated command prompt, navigate to the directory where the Nano Server VHD is and run the following commands: **md mountdir**

   dism\dism /Mount-Image /ImageFile:.\NanoServer.vhd /Index:1 /MountDir:.\mountdir

   dism\dism /Add-Driver /image:.\mountdir /driver: C:\Windows\System32\DriverStore\FileRepository\net1ic64.inf_amd64_fafa7441408bbecd

   dism\dism /Unmount-Image /MountDir:.\MountDir /Commit
10. Repeat these steps for each driver file you need.

> **NOTE**
> In the folder where you keep your drivers, both the SYS files and corresponding INF files must be present. Also, Nano Server only supports signed, 64-bit drivers.

Injecting drivers

Nano Server offers a package that includes a set of basic drivers for a variety of network adapters and storage controllers; it's possible that drivers for your network adapters might not be included. You can use this syntax to have New-NanoServerImage search the directory for available drivers and inject them into the Nano Server image:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\InjectingDrivers.vhdx -DriverPath .\Extra\Drivers
```

> **NOTE**
> In the folder where you keep your drivers, both the SYS files and corresponding INF files must be present. Also, Nano Server only supports signed, 64-bit drivers.

Using the -DriverPath parameter, you can also pass a array of paths to driver .inf files:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\InjectingDrivers.vhdx -DriverPath .\Extra\Drivers\netcard64.inf
```

Connecting with WinRM

To be able to connect to a Nano Server computer using Windows Remote Management (WinRM) (from another computer that is not on the same subnet), open port 5985 for inbound TCP traffic on the Nano Server image. Use this cmdlet:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\ConnectingOverWinRM.vhd -EnableRemoteManagementPort
```

Setting static IP addresses

To configure a Nano Server image to use static IP addresses, first find the name or index of the interface you want to modify by using Get-NetAdapter, netsh, or the Nano Server Recovery Console. Use the -Ipv6Address, -Ipv6Dns, -Ipv4Address, -Ipv4SubnetMask, -Ipv4Gateway and -Ipv4Dns parameters to specify the configuration, as in this example:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\StaticIpv4.vhd -InterfaceNameOrIndex Ethernet -Ipv4Address 192.168.1.2 -Ipv4SubnetMask
255.255.255.0 -Ipv4Gateway 192.168.1.1 -Ipv4Dns 192.168.1.1
```

Custom image size

You can configure the Nano Server image to be a dynamically expanding VHD or VHDX with the -MaxSize parameter, as in this example:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\BigBoss.vhd -MaxSize 100GB
```

Embedding custom data

To embed your own script or binaries in the Nano Server image, use the -CopyPath parameter to pass an array of files and directories to be copied. The -CopyPath parameter can also accept a hashtable to specify the destination path for files and directories.

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\BigBoss.vhd -CopyPath .\tools
```

Running custom commands after the first boot

To run custom commands as part of setupcomplete.cmd, use the -SetupCompleteCommand parameter to pass an array of commands:

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\NanoServer.wim -SetupCompleteCommand @("echo foo", "echo bar")
```

Running custom PowerShell scripts as part of image creation

To run custom PowerShell scripts as part of the image creation process, use the -OfflineScriptPath parameter to pass an array of paths to .ps1 scripts. If those scripts take arguments, use the -OfflineScriptArgument to pass a hashtable of additional arguments to the scripts.

```
New-NanoServerImage -DeploymentType Host -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\NanoServer.wim -OfflineScriptPath C:\MyScripts\custom.ps1 -OfflineScriptArgument @{Param1="Value1";
Param2="Value2"}
```

Support for development scenarios

If you want to develop and test on Nano Server, you can use the -Development parameter. This will enable PowerShell as the default local shell, enable installation of unsigned drivers, copy debugger binaries, open a port for debugging, enable test signing, and enable installation of AppX packages without a developer license:

```
New-NanoServerImage -DeploymentType Guest -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\NanoServer.wim -Development
```

Custom unattend file

If you want to use your own unattend file, use the -UnattendPath parameter:

```
New-NanoServerImage -DeploymentType Guest -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\NanoServer.wim -UnattendPath \\path\to\unattend.xml
```

Specifying an administrator password or computer name in this unattend file will override the values set by -AdministratorPassword and -ComputerName.

> **NOTE**
> Nano Server does not support setting TCP/IP settings via unattend files. You can use Setupcomplete.cmd to configure TCP/IP settings.

Collecting log files

If you want to collect the log files during image creation, use the -LogPath parameter to specify a directory where all the log files are copied.

```
New-NanoServerImage -DeploymentType Guest -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\NanoServer.wim -LogPath C:\Logs
```

> **NOTE**
> Some parameters on New-NanoServerImage and Edit-NanoServerImage are for internal use only and can be safely ignored. These include the -SetupUI and -Internal parameters.

Installing apps and drivers

Windows Server App installer

Windows Server App (WSA) installer provides a reliable installation option for Nano Server. Since Windows Installer (MSI) is not supported on Nano Server, WSA is also the only installation technology available for non-Microsoft products. WSA leverages Windows app package technology designed to install and service applications safely and reliably, using a declarative manifest. It extends the Windows app package installer to support Windows Server-specific extensions, with the limitation that WSA does not support installing drivers.

Creating and installing a WSA package on Nano Server involves steps for both the publisher and the consumer of the package.

The package publisher should do the following:

1. Install Windows 10 SDK, which includes the tools needed to create a WSA package: MakeAppx, MakeCert, Pvk2Pfx, SignTool.
2. Declare a manifest: Follow the WSA manifest extension schema to create the manifest file, AppxManifest.xml.
3. Use the **MakeAppx** tool to create a WSA package.
4. Use **MakeCert** and **Pvk2Pfx** tools to create the certificate, and then use **Signtool** to sign the package.

Next, the package consumer should follow these steps:

1. Run the *Import-Certificate* PowerShell cmdlet to import the publisher's certificate from Step 4 above to Nano Server with the certStoreLocation at "Cert:\LocalMachine\TrustedPeople". For example:
   `Import-Certificate -FilePath ".\xyz.cer" -CertStoreLocation "Cert:\LocalMachine\TrustedPeople"`
2. Install the app on Nano Server by running the **Add-AppxPackage** PowerShell cmdlet to install a WSA package on Nano Server. For example: `Add-AppxPackage wsaSample.appx`

Additional resources for creating apps

WSA is server extension of Windows app package technology (though it is not hosted in Microsoft Store). If you want to publish apps with WSA,these topics will help you familiarize youself with the app package pipeline:

- How to create a basic package manifest
- App Packager (MakeAppx.exe)
- How to create an app package signing certificate
- SignTool

Installing drivers on Nano Server

You can install non-Microsoft drivers on Nano Server by using INF driver packages. These include both Plug-and-Play (PnP) driver packages and File System Filter driver packages. Network Filter drivers are not currently supported on Nano Server.

Both PnP and File System Filter driver packages must follow the Universal driver requirements and installation process, as well as general driver package guidelines such as signing. They are documented at these locations:

- Driver Signing
- Using a Universal INF File

Installing driver packages offline

Supported driver packages can be installed on Nano Server offline via DISM.exe or DISM PowerShell cmdlets.

Installing driver packages online

PnP driver packages can be installed to Nano Server online by using PnpUtil. Online driver installation for non-PnP driver packages is not currently supported on Nano Server.

Joining Nano Server to a domain

To add Nano Server to a domain online

1. Harvest a data blob from a computer in the domain that is already running Windows Threshold Server using this command:

   ```
   djoin.exe /provision /domain <domain-name> /machine <machine-name> /savefile .\odjblob
   ```

   This saves the data blob in a file called "odjblob".

2. Copy the "odjblob" file to the Nano Server computer with these commands:

   net use z: \\<ip address of Nano Server>\c$

   > **NOTE**
   > If the net use command fails, you probably need to adjust Windows Firewall rules. To do this, first open an elevated command prompt, start Windows PowerShell and then connect to the Nano Server computer with Windows PowerShell Remoting with these commands:
   >
   > ```
   > Set-Item WSMan:\localhost\Client\TrustedHosts "<IP address of Nano Server>"
   > ```
   >
   > ```
   > $ip = "<ip address of Nano Server>"
   > ```
   >
   > ```
   > Enter-PSSession -ComputerName $ip -Credential $ip\Administrator
   > ```
   >
   > When prompted, provide the Administrator password, then run this command to set the firewall rule:
   >
   > **netsh advfirewall firewall set rule group="File and Printer Sharing" new enable=yes**
   >
   > Exit Windows PowerShell with `Exit-PSSession`, and then retry the net use command. If successful, continue copying the "odjblob" file contents to the Nano Server.

   md z:\Temp

   copy odjblob z:\Temp

3. Check the domain you want to join Nano Server to and ensure that DNS is configured. Also, verify that name resolution of the domain or a domain controller works as expected. To do this, open an elevated command prompt, start Windows PowerShell and then connect to the Nano Server computer with Windows PowerShell remoting with these commands:

   ```
   Set-Item WSMan:\localhost\Client\TrustedHosts "<IP address of Nano Server>"
   ```

   ```
   $ip = "<ip address of Nano Server>"
   ```

   ```
   Enter-PSSession -ComputerName $ip -Credential $ip\Administrator
   ```

   When prompted, provide the Administrator password. Nslookup is not available on Nano Server, so you can verify name resolution with Resolve-DNSName.

4. If name resolution succeeds, then in the same Windows PowerShell session, run this command to join the domain:

   ```
   djoin /requestodj /loadfile c:\Temp\odjblob /windowspath c:\windows /localos
   ```

5. Restart the Nano Server computer, and then exit the Windows PowerShell session:

   ```
   shutdown /r /t 5
   ```

   ```
   Exit-PSSession
   ```

6. After you have joined Nano Server to a domain, add the domain user account to the Administrators group on the Nano Server.

7. For security, remove the Nano Server from the trusted hosts list with this command:
   ```
   Set-Item WSMan:\localhost\client\TrustedHosts ""
   ```

Alternate method to join a domain in one step

First, harvest the data blob from another computer running Windows Threshold Server that is already in your domain using this command:

```
djoin.exe /provision /domain <domain-name> /machine <machine-name> /savefile .\odjblob
```

Open the file "odjblob" (perhaps in Notepad), copy its contents, and then paste the contents into the <AccountData> section of the Unattend.xml file below.

Put this Unattend.xml file into the C:\NanoServer folder, and then use the following commands to mount the VHD and apply the settings in the `offlineServicing` section:

dism\dism /Mount-ImagemediaFile:.\NanoServer.vhd /Index:1 /MountDir:.\mountdir

dism\dismmedia:.\mountdir /Apply-Unattend:.\unattend.xml

Create a "Panther" folder (used by Windows systems for storing files during setup; see Windows 7, Windows Server 2008 R2, and Windows Vista setup log file locations if you're curious), copy the Unattend.xml file to it, and then unmount the VHD with these commands:

md .\mountdir\windows\panther

copy .\unattend.xml .\mountdir\windows\panther

dism\dism /Unmount-Image /MountDir:.\mountdir /Commit

The first time you boot Nano Server from this VHD, the other settings will be applied.

After you have joined Nano Server to a domain, add the domain user account to the Administrators group on the Nano Server.

Working with server roles on Nano Server

Using Hyper-V on Nano Server

Hyper-V works the same on Nano Server as it does on Windows Server in Server Core mode, with two exceptions:

- You must perform all management remotely and the management computer must be running the same build of Windows Server as the Nano Server. Older versions of Hyper-V Manager or Hyper-V Windows PowerShell cmdlets will not work.
- RemoteFX is not available.

In this release, these features of Hyper-V have been verified:

- Enabling Hyper-V
- Creation of Generation 1 and Generation 2 virtual machines
- Creation of virtual switches
- Starting virtual machines and running Windows guest operating systems
- Hyper-V Replica

If you want to perform a live migration of virtual machines, create a virtual machine on an SMB share, or connect resources on an existing SMB share to an existing virtual machine, it is vital that you configure authentication correctly. You have two options for doing this:

Constrained delegation

Constrained delegation works exactly the same as in previous releases. Refer to these articles for more information:

- Enabling Hyper-V Remote Management - Configuring Constrained Delegation For SMB and Highly Available SMB
- Enabling Hyper-V Remote Management - Configuring Constrained Delegation For Non-Clustered Live Migration

CredSSP

First, refer to the "Using Windows PowerShell remoting" section of this topic to enable and test CredSSP. Then, on the management computer, you can use Hyper-V Manager and select the option to "connect as another user." Hyper-V Manager will use CredSSP. You should do this even if you are using your current account.

Windows PowerShell cmdlets for Hyper-V can use CimSession or Credential parameters, either of which work with CredSSP.

Using Failover Clustering on Nano Server

Failover clustering works the same on Nano Server as it does on Windows Server in Server Core mode, but keep these caveats in mind:

- Clusters must be managed remotely with Failover Cluster Manager or Windows PowerShell.
- All Nano Server cluster nodes must be joined to the same domain, similar to cluster nodes in Windows Server.
- The domain account must have Administrator privileges on all Nano Server nodes, as with cluster nodes in Windows Server.
- All commands must be run in an elevated command prompt.

NOTE

Additionally, certain features are not supported in this release:

- You cannot run failover clustering cmdlets on a local Nano Server through Windows PowerShell.
- Clustering roles other than Hyper-V and File Server.

You'll find these Windows PowerShell cmdlets useful in managing Failover clusters:

You can create a new cluster with `New-Cluster -Name <clustername> -Node <comma-separated cluster node list>`

Once you've established a new cluster, you should run `Set-StorageSetting -NewDiskPolicy OfflineShared` on all nodes.

Add an additional node to the cluster with
`Add-ClusterNode -Name <comma-separated cluster node list> -Cluster <clustername>`

Remove a node from the cluster with
`Remove-ClusterNode -Name <comma-separated cluster node list> -Cluster <clustername>`

Create a Scale-Out File Server with `Add-ClusterScaleoutFileServerRole -name <sofsname> -cluster <clustername>`

You can find additional cmdlets for failover clustering at Microsoft.FailoverClusters.PowerShell.

Using DNS Server on Nano Server

To provide Nano Server with the DNS Server role, add the Microsoft-NanoServer-DNS-Package to the image (see the "Creating a custom Nano Server image" section of this topic. Once the Nano Server is running, connect to it and run this command from and elevated Windows PowerShell console to enable the feature:

```
Enable-WindowsOptionalFeature -Online -FeatureName DNS-Server-Full-Role
```

Using IIS on Nano Server

For steps to use the Internet Information Services (IIS) role, see IIS on Nano Server.

Using MPIO on Nano Server

For steps to use MPIO, see MPIO on Nano Server

Using SSH on Nano Server

For instructions on how to install and use SSH on Nano Server with the OpenSSH project, see the Win32-OpenSSH wiki.

Appendix: Sample Unattend.xml file that joins Nano Server to a domain

NOTE

Be sure to delete the trailing space in the contents of "odjblob" once you paste it into the Unattend file.

```
<?xml version='1.0' encoding='utf-8'?>
<unattend xmlns="urn:schemas-microsoft-com:unattend"
xmlns:wcm="https://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">

  <settings pass="offlineServicing">
    <component name="Microsoft-Windows-UnattendedJoin" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS">
        <OfflineIdentification>
           <Provisioning>
             <AccountData>
AAAAAAARUABLEABLEABAoAAAAAAAMABSUABLEABLEABAwAAAAAAAAABbMAAdYABc8ABYkABLAABbMAAEAAAAMAAA0ABY4ABZ8ABbIABa0AAcIAB
Y4ABb8ABZUABAsAAAAAAAQAAZoABNUABOYABZYAANQABMoAAOEAAMIAAOkAANoAAMAAAXwAAJAAAAYAAA0ABY4ABZ8ABbIABa0AAcIABY4ABb8A
BZUABLEAALMABLQABU0AATMABXAAAAAAAKdf/mhfXoAAUAAAQAAAAb8ABLQABbMABcMABb4ABc8ABAIAAAAAAb8ABLQABbMABcMABb4ABc8ABLQ
ABb0ABZIAAGAAAAsAAR4ABTQABUAAAAAAACAAAQwABZMAAZcAAUgABVcAAegAARcABKkABVIAASwAAY4ABbcABW8ABQoAAT0ABN8AAO8ABekAAJ
MAAVkAAZUABckABXEABJUAAQ8AAJ4AAIsABZMABdoAAOsABIsABKkABQEABUEABIwABKoAAaAABXgABNwAAegAAAkAAAAABAMABLIABdIABc8AB
Y4AADAAAA4AAZ4ABbQABcAAAAAAACAAkKBW0ID8nJDWYAHnBAXE77j7BAEWEkl+lKB98XC2G0/9+Wd1DJQW4IYAkKBAADhAnKBWEwhiDAAAM2zz
DCEAM6IAAAgAAAAAAQAAAAAAAAAAAABwzzAAA
</AccountData>
           </Provisioning>
         </OfflineIdentification>
    </component>
  </settings>

  <settings pass="oobeSystem">
    <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS">
      <UserAccounts>
        <AdministratorPassword>
           <Value>Tuva</Value>
           <PlainText>true</PlainText>
        </AdministratorPassword>
      </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">
      <RegisteredOwner>My Team</RegisteredOwner>
      <RegisteredOrganization>My Corporation</RegisteredOrganization>
    </component>
  </settings>
</unattend>
```

IIS on Nano Server

6/20/2018 • 10 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

You can install the Internet Information Services (IIS) server role on Nano Server by using the -Package parameter with Microsoft-NanoServer-IIS-Package. For information about configuring Nano Server, including installing packages, see Install Nano Server.

In this release of Nano Server, the following IIS features are available:

FEATURE	ENABLED BY DEFAULT
Common HTTP Features	
Default document	x
Directory browsing	x
HTTP Errors	x
Static content	x
HTTP redirection	
Health and Diagnostics	
HTTP logging	x
Custom logging	
Request monitor	
Tracing	
Performance	
Static content compression	x
Dynamic content compression	
Security	
Request filtering	x

FEATURE	ENABLED BY DEFAULT
Basic authentication	
Client certificate mapping authentication	
Digest authentication	
IIS client certificate mapping authentication	
IP and domain restrictions	
URL authorization	
Windows authentication	
Application Development	
Application initialization	
CGI	
ISAPI extensions	
ISAPI filters	
Server-side includes	
WebSocket protocol	
Management Tools	
IISAdministration module for Windows PowerShell	x

A series of articles on other configurations of IIS (such as using ASP.NET, PHP, and Java), as well as other related content is published at http://iis.net/learn.

Installing IIS on Nano Server

You can install this server role either offline (with the Nano Server off) or online (with the Nano Server running); offline installation is the recommended option.

For offline installation, add the package with the -Packages parameter of New-NanoServerImage, as in this example:

```
New-NanoServerImage -Edition Standard -DeploymentType Guest -MediaPath f:\ -BasePath .\Base -TargetPath
.\Nano1.vhd -ComputerName Nano1 -Package Microsoft-NanoServer-IIS-Package
```

If you have an existing VHD file, you can install IIS offline with DISM.exe by mounting the VHD, and then using the **Add-Package** option.
The following example steps assume that you are running from the directory specified by BasePath option, which was created after running New-NanoServerImage.

1. mkdir mountdir
2. .\Tools\dism.exe /Mount-Image /ImageFile:.\NanoServer.vhd /Index:1 /MountDir:.\mountdir
3. .\Tools\dism.exe /Add-Package /PackagePath:.\packages\Microsoft-NanoServer-IIS-Package.cab /Image:.\mountdir
4. .\Tools\dism.exe /Add-Package /PackagePath:.\packages\en-us\Microsoft-NanoServer-IIS-Package_en-us.cab /Image:.\mountdir
5. .\Tools\dism.exe /Unmount-Image /MountDir:.\MountDir /Commit

> **NOTE**
> Note that Step 4 adds the language pack--this example installs EN-US.

At this point you can start Nano Server with IIS.

Installing IIS on Nano Server online

Though offline installation of the server role is recommended, you might need to install it online (with the Nano Server running) in container scenarios. To do this, follow these steps:

1. Copy the Packages folder from the installation media locally to the running Nano Server (for example, to C:\packages).

2. Create a new Unattend.xml file on another computer and then copy it to the Nano Server. You can copy and paste this XML content into the XML file you created:

```
    <unattend xmlns="urn:schemas-microsoft-com:unattend">
    <servicing>
        <package action="install">
            <assemblyIdentity name="Microsoft-NanoServer-IIS-Package" version="10.0.14393.0"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" />
            <source location="c:\packages\Microsoft-NanoServer-IIS-Package.cab" />
        </package>
        <package action="install">
            <assemblyIdentity name="Microsoft-NanoServer-IIS-Package" version="10.0.14393.0"
processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="en-US" />
            <source location="c:\packages\en-us\Microsoft-NanoServer-IIS-Package_en-us.cab" />
        </package>
    </servicing>
    <cpi:offlineImage cpi:source="" xmlns:cpi="urn:schemas-microsoft-com:cpi" />
</unattend>
```

1. In the new XML file you created (or copied), edit C:\packages to the directory you copied the content of Packages to.

2. Switch to the directory with the newly created XML file and run

   dism /online /apply-unattend:.\unattend.xml

3. Confirm that the IIS package and its associated language pack is installed correctly by running:

   dism /online /get-packages

   You should see "Package Identity : Microsoft-NanoServer-IIS-Package~31bf3856ad364e35~amd64~~10.0.14393.1000" listed twice, once for Release Type : Language Pack and once for Release Type : Feature Pack.

4. Start the W3SVC service either with **net start w3svc** or by restarting the Nano Server.

Starting IIS

Once IIS is installed and running, it is ready to serve web requests. Verify that IIS is running by browsing the default IIS web page at http://<IP address of Nano Server>. On a physical computer, you can determine the IP address by using the Recovery Console. On a virtual machine, you can get the IP address by using a Windows PowerShell prompt and running:

```
Get-VM -name <VM name> | Select -ExpandProperty networkadapters | select IPAddresses
```

If you are not able to access the default IIS web page, double-check the IIS installation by looking for the **c:\inetpub** directory on the Nano Server.

Enabling and disabling IIS features

A number of IIS features are enabled by default when you install the IIS role (see the table in the "Overview of IIS on Nano Server" section of this topic). You can enable (or disable) additional features using DISM.exe

Each feature of IIS exists as a set of configuration elements. For example, the Windows authentication feature comprises these elements:

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="WindowsAuthenticationModule" image="%windir%\System32\inetsrv\authsspi.dll`
`<modules>`	`<add name="WindowsAuthenticationModule" lockItem="true" \/>`
`<windowsAuthentication>`	`<windowsAuthentication enabled="false" authPersistNonNTLM\="true"><providers><add value="Negotiate" /><add value="NTLM" />  </providers> </windowsAuthentication>`

The full set of IIS sub-features are included in Appendix 1 of this topic and their corresponding configuration elements is included in Appendix 2 of this topic.

Example: installing Windows authentication

1. Open a Windows PowerShell remote session console on the Nano Server.
2. Use `DISM.exe` to install the Windows authentication module:

   ```
   dism /Enable-Feature /online /featurename:IIS-WindowsAuthentication /all
   ```

   The `/all` switch will install any feature that the chosen feature depends on.

Example: uninstalling Windows authentication

1. Open a Windows PowerShell remote session console on the Nano Server.
2. Use `DISM.exe` to uninstall the Windows authentication module:

   ```
   dism /Disable-Feature /online /featurename:IIS-WindowsAuthentication
   ```

Other common IIS configuration tasks

Creating websites

Use this cmdlet:

```
PS D:\> New-IISSite -Name TestSite -BindingInformation "*:80:TestSite" -PhysicalPath c:\test
```

You can then run `Get-IISSite` to verify the state of the site (returns the web site name, ID, state, physical path, and bindings).

Deleting web sites

Run `Remove-IISSite -Name TestSite -Confirm:$false`.

Creating virtual directories

You can create virtual directories by using the IISServerManager object returned by Get-IISServerManager, which exposes the .NET Microsoft.Web.Administration.ServerManager API. In this example, these commands access the "Default Web Site" element of the Sites collection and the root application element ("/") of the Applications section. They then call the Add() method of the VirtualDirectories collection for that application element to create the new directory:

```
PS C:\> $sm = Get-IISServerManager
PS C:\> $sm.Sites["Default Web Site"].Applications["/"].VirtualDirectories.Add("/DemoVirtualDir1",
"c:\test\virtualDirectory1")
PS C:\> $sm.Sites["Default Web Site"].Applications["/"].VirtualDirectories.Add("/DemoVirtualDir2",
"c:\test\virtualDirectory2")
PS C:\> $sm.CommitChanges()
```

Creating application pools

Similarly you can use Get-IISServerManager to create application pools:

```
PS C:\> $sm = Get-IISServerManager
PS C:\> $sm.ApplicationPools.Add("DemoAppPool")
```

Configuring HTTPS and certificates

Use the Certoc.exe utility to import certificates, as in this example, which shows configuring HTTPS for a website on a Nano Server:

1. On another computer that is not running Nano Server, create a certificate (using your own certificate name and password), and then export it to c:\temp\test.pfx.

   ```
   $newCert = New-SelfSignedCertificate -DnsName "www.foo.bar.com" -CertStoreLocation cert:\LocalMachine\my
   ```

   ```
   $mypwd = ConvertTo-SecureString -String "YOUR_PFX_PASSWD" -Force -AsPlainText
   ```

   ```
   Export-PfxCertificate -FilePath c:\temp\test.pfx -Cert $newCert -Password $mypwd
   ```

2. Copy the test.pfx file to the Nano Server computer.

3. On the Nano Server, import the certificate to the "My" store with this command:

   certoc.exe -ImportPFX -p YOUR_PFX_PASSWD My c:\temp\test.pfx

4. Retrieve the thumbprint of this new certificate (in this example, 61E71251294B2A7BB8259C2AC5CF7BA622777E73) with `Get-ChildItem Cert:\LocalMachine\my`.

5. Add the HTTPS binding to the Default Web Site (or whatever website you want to add the binding to) by using these Windows PowerShell commands:

```
$certificate = get-item Cert:\LocalMachine\my\61E71251294B2A7BB8259C2AC5CF7BA622777E73
# Use your actual thumbprint instead of this example
$hash = $certificate.GetCertHash()

Import-Module IISAdministration
$sm = Get-IISServerManager
$sm.Sites["Default Web Site"].Bindings.Add("*:443:", $hash, "My", "0")    # My is the certificate store
name
$sm.CommitChanges()
```

You could also use Server Name Indication (SNI) with a specific host name with this syntax:

```
$sm.Sites["Default Web Site"].Bindings.Add("*:443:www.foo.bar.com", $hash, "My", "Sni".
```

Appendix 1: List of IIS sub-features

- IIS-WebServer
- IIS-CommonHttpFeatures
- IIS-StaticContent
- IIS-DefaultDocument
- IIS-DirectoryBrowsing
- IIS-HttpErrors
- IIS-HttpRedirect
- IIS-ApplicationDevelopment
- IIS-CGI
- IIS-ISAPIExtensions
- IIS-ISAPIFilter
- IIS-ServerSideIncludes
- IIS-WebSockets
- IIS-ApplicationInit
- IIS-Security
- IIS-BasicAuthentication
- IIS-WindowsAuthentication
- IIS-DigestAuthentication
- IIS-ClientCertificateMappingAuthentication
- IIS-IISCertificateMappingAuthentication
- IIS-URLAuthorization
- IIS-RequestFiltering
- IIS-IPSecurity
- IIS-CertProvider
- IIS-Performance
- IIS-HttpCompressionStatic
- IIS-HttpCompressionDynamic
- IIS-HealthAndDiagnostics
- IIS-HttpLogging
- IIS-LoggingLibraries
- IIS-RequestMonitor
- IIS-HttpTracing
- IIS-CustomLogging

Appendix 2: Elements of HTTP features

Each feature of IIS exists as a set of configuration elements. This appendix lists the configuration elements for all of the features in this release of Nano Server

Common HTTP features

Default document

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="DefaultDocumentModule" image="%windir%\System32\inetsrv\defdoc.dll" />`
`<modules>`	`<add name="DefaultDocumentModule" lockItem="true" />`
`<handlers>`	`<add name="StaticFile" path="*" verb="*" modules="DefaultDocumentModule" resourceType="EiSecther" requireAccess="Read" />`
`<defaultDocument>`	`<defaultDocument enabled="true"> <files>  <add value="Default.htm" />  <add value="Default.asp" />  <add value="index.htm" />  <add value="index.html" />  <add value="iisstart.htm" />  </files>  </defaultDocument>`

The `StaticFile <handlers>` entry might already be present; if so, just add "DefaultDocumentModule" to the <modules> attribute, separated by a comma.

Directory browsing

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="DirectoryListingModule" image="%windir%\System32\inetsrv\dirlist.dll" />`
`<modules>`	`<add name="DirectoryListingModule" lockItem="true" />`
`<handlers>`	`<add name="StaticFile" path="*" verb="*" modules="DirectoryListingModule" resourceType="Either" requireAccess="Read" />`

The `StaticFile <handlers>` entry might already be present; if so, just add "DirectoryListingModule" to the <modules> attribute, separated by a comma.

HTTP errors

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="CustomErrorModule" image="%windir%\System32\inetsrv\custerr.dll" />`
`<modules>`	`<add name="CustomErrorModule" lockItem="true" />`

SECTION	CONFIGURATION ELEMENTS
`<httpErrors>`	`<httpErrors lockAttributes="allowAbsolutePathsWhenDelegated,defaultPath"> <error statusCode="401" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="401.htm" > <error statusCode="403" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="403.htm" /> <error statusCode="404" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="404.htm" /> <error statusCode="405" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="405.htm" /> <error statusCode="406" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="406.htm" /> <error statusCode="412" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="412.htm" /> <error statusCode="500" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="500.htm" /> <error statusCode="501" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="501.htm" /> <error statusCode="502" prefixLanguageFilePath="%SystemDrive%\inetpub\custerr" path="502.htm" /> </httpErrors>`

Static content

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="StaticFileModule" image="%windir%\System32\inetsrv\static.dll" />`
`<modules>`	`<add name="StaticFileModule" lockItem="true" />`
`<handlers>`	`<add name="StaticFile" path="*" verb="*" modules="StaticFileModule" resourceType="Either" requireAccess="Read" />`

The `StaticFile \<handlers>` entry might already be present; if so, just add "StaticFileModule" to the <modules> attribute, separated by a comma.

HTTP redirection

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="HttpRedirectionModule" image="%windir%\System32\inetsrv\redirect.dll" />`
`<modules>`	`<add name="HttpRedirectionModule" lockItem="true" />`
`<httpRedirect>`	`<httpRedirect enabled="false" />`

Health and diagnostics

HTTP logging

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="HttpLoggingModule" image="%windir%\System32\inetsrv\loghttp.dll" />`
`<modules>`	`<add name="HttpLoggingModule" lockItem="true" />`
`<httpLogging>`	`<httpLogging dontLog="false" />`

Custom logging

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="CustomLoggingModule" image="%windir%\System32\inetsrv\logcust.dll" />`
`<modules>`	`<add name="CustomLoggingModule" lockItem="true" />`

Request monitor

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="RequestMonitorModule" image="%windir%\System32\inetsrv\iisreqs.dll" />`

Tracing

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="TracingModule" image="%windir%\System32\inetsrv\iisetw.dll" \/> <add name="FailedRequestsTracingModule" image="%windir%\System32\inetsrv\iisfreb.dll" />`
`<modules>`	`<add name="FailedRequestsTracingModule" lockItem="true" />`
`<traceProviderDefinitions>`	`<traceProviderDefinitions> <add name="WWW Server" guid\="{3a2a4e84-4c21-4981-ae10-3fda0d9b0f83}"> <areas> <clear /> <add name="Authentication" value="2" /> <add name="Security" value="4" /> <add name="Filter" value="8" /> <add name="StaticFile" value="16" /> <add name="CGI" value="32" /> <add name="Compression" value="64" /> <add name="Cache" value="128" /> <add name="RequestNotifications" value="256" /> <add name="Module" value="512" /> <add name="FastCGI" value="4096" /> <add name="WebSocket" value="16384" /> </areas> </add> <add name="ISAPI Extension" guid="{a1c2040e-8840-4c31-ba11-9871031a19ea}"> <areas> <clear /> </areas> </add> </traceProviderDefinitions>`

Performance

Static content compression

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="StaticCompressionModule" image="%windir%\System32\inetsrv\compstat.dll" />`
`<modules>`	`<add name="StaticCompressionModule" lockItem="true" />`

SECTION	CONFIGURATION ELEMENTS
`<httpCompression>`	`<httpCompression directory="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files"> <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" /> <staticTypes> <add mimeType="text/*" enabled="true" /> <add mimeType="message/*" enabled="true" /> <add mimeType="application/javascript" enabled="true" \/> <add mimeType="application/atom+xml" enabled="true" /> <add mimeType="application/xaml+xml" enabled="true" /> <add mimeType="**" enabled="false" /> </staticTypes> </httpCompression>`

Dynamic content compression

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="DynamicCompressionModule" image="%windir%\System32\inetsrv\compdyn.dll" />`
`<modules>`	`<add name="DynamicCompressionModule" lockItem="true" />`
`<httpCompression>`	`<httpCompression directory\="%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files"> <scheme name="gzip" dll="%Windir%\system32\inetsrv\gzip.dll" \/> \ <dynamicTypes> <add mimeType="text/*" enabled="true" \/> <add mimeType="message/*" enabled="true" /> <add mimeType="application/x-javascript" enabled="true" /> <add mimeType="application/javascript" enabled="true" /> <add mimeType="*/*" enabled="false" /> <\/dynamicTypes> </httpCompression>`

Security

Request filtering

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="RequestFilteringModule" image="%windir%\System32\inetsrv\modrqflt.dll" />`
`<modules>`	`<add name="RequestFilteringModule" lockItem="true" />`
`	`<requestFiltering>  <fileExtensions allowUnlisted="true" applyToWebDAV="true" />  <verbs allowUnlisted="true" applyToWebDAV="true" />   <hiddenSegments applyToWebDAV="true">  <add segment="web.config" />  </hiddenSegments>  </requestFiltering>`

Basic authentication

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="BasicAuthenticationModule" image="%windir%\System32\inetsrv\authbas.dll" />`
`<modules>`	`<add name="WindowsAuthenticationModule" lockItem="true" />`

SECTION	CONFIGURATION ELEMENTS
`<basicAuthentication>`	`<basicAuthentication enabled="false" />`

Client certificate mapping authentication

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="CertificateMappingAuthentication" image="%windir%\System32\inetsrv\authcert.dll" />`
`<modules>`	`<add name="CertificateMappingAuthenticationModule" lockItem="true" />`
`<clientCertificateMappingAuthentication>`	`<clientCertificateMappingAuthentication enabled="false" />`

Digest authentication

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="DigestAuthenticationModule" image="%windir%\System32\inetsrv\authmd5.dll" />`
`<modules>`	`<add name="DigestAuthenticationModule" lockItem="true" />`
`<other>`	`<digestAuthentication enabled="false" />`

IIS client certificate mapping authentication

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="CertificateMappingAuthenticationModule" image="%windir%\System32\inetsrv\authcert.dll" />`
`<modules>`	`<add name="CertificateMappingAuthenticationModule" lockItem="true"` />`
`<clientCertificateMappingAuthentication>`	`<clientCertificateMappingAuthentication enabled="false" />`

IP and domain restrictions

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="IpRestrictionModule" image="%windir%\System32\inetsrv\iprestr.dll" /> <add name="DynamicIpRestrictionModule" image="%windir%\System32\inetsrv\diprestr.dll" />`
`<modules>`	`<add name="IpRestrictionModule" lockItem="true" \/>  <add name="DynamicIpRestrictionModule" lockItem="true" \/>`
`<ipSecurity>`	`<ipSecurity allowUnlisted="true" />`

URL authorization

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="UrlAuthorizationModule" image="%windir%\System32\inetsrv\urlauthz.dll" />`
`<modules>`	`<add name="UrlAuthorizationModule" lockItem="true" />`
`<authorization>`	`<authorization>  <add accessType="Allow" users="*" /> </authorization>`

Windows authentication

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="WindowsAuthenticationModule" image="%windir%\System32\inetsrv\authsspi.dll" />`
`<modules>`	`<add name="WindowsAuthenticationModule" lockItem="true" />`
`<windowsAuthentication>`	`<windowsAuthentication enabled="false" authPersistNonNTLM\="true"> <providers> <add value="Negotiate" /> <add value="NTLM" /> <\providers> <\windowsAuthentication> <windowsAuthentication enabled="false" authPersistNonNTLM\="true"> <providers> <add value="Negotiate" /> <add value="NTLM" /> <\/providers> <\/windowsAuthentication>`

Application development

Application initialization

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="ApplicationInitializationModule" image="%windir%\System32\inetsrv\warmup.dll" />`
`<modules>`	`<add name="ApplicationInitializationModule" lockItem="true" />`

CGI

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="CgiModule" image="%windir%\System32\inetsrv\cgi.dll" />  <add name="FastCgiModule" image="%windir%\System32\inetsrv\iisfcgi.dll" />`
`<modules>`	`<add name="CgiModule" lockItem="true" /> <add name="FastCgiModule" lockItem="true" />`
`<handlers>`	`<add name="CGI-exe" path="*.exe" verb="\*" modules="CgiModule" resourceType="File" requireAccess="Execute" allowPathInfo="true" />`

ISAPI extensions

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="IsapiModule" image="%windir%\System32\inetsrv\isapi.dll" />`
`<modules>`	`<add name="IsapiModule" lockItem="true" />`
`<handlers>`	`<add name="ISAPI-dll" path="*.dll" verb="*" modules="IsapiModule" resourceType="File" requireAccess="Execute" allowPathInfo="true" />`

ISAPI filters

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="IsapiFilterModule" image="%windir%\System32\inetsrv\filter.dll" />`
`<modules>`	`<add name="IsapiFilterModule" lockItem="true" />`

Server-side includes

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="ServerSideIncludeModule" image="%windir%\System32\inetsrv\iis_ssi.dll" />`
`<modules>`	`<add name="ServerSideIncludeModule" lockItem="true" />`
`<handlers>`	`<add name="SSINC-stm" path="*.stm" verb="GET,HEAD,POST" modules="ServerSideIncludeModule" resourceType="File" \/> <add name="SSINC-shtm" path="*.shtm" verb="GET,HEAD,POST" modules="ServerSideIncludeModule" resourceType="File" /> <add name="SSINC-shtml" path="*.shtml" verb="GET,HEAD,POST" modules="ServerSideIncludeModule" resourceType="File" />`
`<serverSideInclude>`	`<serverSideInclude ssiExecDisable="false" />`

WebSocket protocol

SECTION	CONFIGURATION ELEMENTS
`<globalModules>`	`<add name="WebSocketModule" image="%windir%\System32\inetsrv\iiswsock.dll" />`
`<modules>`	`<add name="WebSocketModule" lockItem="true" />`

MPIO on Nano Server

6/20/2018 • 5 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

This topic introduces the use of MPIO in Nano Server installations of Windows Server 2016. For general information about MPIO in Windows Server, see Multipath I/O Overview.

Using MPIO on Nano Server

You can use MPIO on Nano Server, but with these differences:

- Only MSDSM is supported.
- The Load Balancing Policy is chosen dynamically and cannot be modified. The policy has these characteristics:
  - Default -- RoundRobin (active/active)
  - SAS HDD -- LeastBlocks
  - ALUA -- RoundRobin with Subset
- Path states (active/passive) for ALUA arrays are picked up from the target array.
- Storage devices are claimed by bus type (for example, FC, iSCSI, or SAS). When MPIO is installed on Nano Server, disks are still exposed as duplicates (one available per path) until MPIO is configured to claim and manage particular disks. The sample script in this topic will claim or unclaim disks for MPIO.
- iSCSI boot is not supported.

Enable MPIO with this Windows PowerShell cmdlet:

```
Enable-WindowsOptionalFeature -Online -FeatureName MultiPathIO
```

This sample script will allow the caller to claim or unclaim disks for MPIO by changing certain registry keys. Though you can claim other storage devices by adding them to these keys, manipulating the keys directly is not recommended.

```
#
#  Copyright (c) 2015 Microsoft Corporation.  All rights reserved.
#
#  THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY
#  OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED
#  TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A
#  PARTICULAR PURPOSE
#

<#
.Synopsis
```

```
    This powershell script allows you to enable Multipath-IO support using Microsoft's
    in-box DSM (MSDSM) for storage devices attached by certain bus types.

    After running this script you will have to either:
    1. Disable and then re-enable the relevant Host Bus Adapters (HBAs); or
    2. Reboot the system.

.Description

.Parameter BusType
    Specifies the bus type for which the claim/unclaim should be done.

    If omitted, this parameter defaults to "All".

    "All" - Will claim/unclaim storage devices attached through Fibre Channel, iSCSI, or SAS.

    "FC" - Will claim/unclaim storage devices attached through Fibre Channel.

    "iSCSI" - Will claim/unclaim storage devices attached through iSCSI.

    "SAS" - Will claim/unclaim storage devices attached through SAS.

.Parameter Server
    Allows you to specify a remote system, either via computer name or IP address.

    If omitted, this parameter defaults to the local system.

.Parameter Unclaim
    If specified, the script will unclaim storage devices of the bus type specified by the
    BusType parameter.

    If omitted, the script will default to claiming storage devices instead.

.Example
MultipathIoClaim.ps1

Claims all storage devices attached through Fibre Channel, iSCSI, or SAS.

.Example
MultipathIoClaim.ps1 FC

Claims all storage devices attached through Fibre Channel.

.Example
MultipathIoClaim.ps1 SAS -Unclaim

Unclaims all storage devices attached through SAS.

.Example
MultipathIoClaim.ps1 iSCSI 12.34.56.78

Claims all storage devices attached through iSCSI on the remote system with IP address 12.34.56.78.

#>
[CmdletBinding()]
param
(
    [ValidateSet('all','fc','iscsi','sas')]
    [string]$BusType='all',

    [string]$Server="127.0.0.1",

    [switch]$Unclaim
)

#
# Constants
#
$type = [Microsoft.Win32.RegistryHive]::LocalMachine
```

```
[string]$mpioKeyName = "SYSTEM\CurrentControlSet\Control\MPDEV"
[string]$mpioValueName = "MpioSupportedDeviceList"
[string]$msdsmKeyName = "SYSTEM\CurrentControlSet\Services\msdsm\Parameters"
[string]$msdsmValueName = "DsmSupportedDeviceList"

[string]$fcHwid = "MSFT2015FCBusType_0x6   "
[string]$sasHwid = "MSFT2011SASBusType_0xA  "
[string]$iscsiHwid = "MSFT2005iSCSIBusType_0x9"

#
# Functions
#

function AddHardwareId
{
    param
    (
        [Parameter(Mandatory=$True)]
        [string]$Hwid,

        [string]$Srv="127.0.0.1",

        [string]$KeyName="SYSTEM\CurrentControlSet\Control\MultipathIoClaimTest",

        [string]$ValueName="DeviceList"
    )

    $regKey = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey($type, $Srv)
    $key = $regKey.OpenSubKey($KeyName, 'true')
    $val = $key.GetValue($ValueName)
    $val += $Hwid
    $key.SetValue($ValueName, [string[]]$val, 'MultiString')
}

function RemoveHardwareId
{
    param
    (
        [Parameter(Mandatory=$True)]
        [string]$Hwid,

        [string]$Srv="127.0.0.1",

        [string]$KeyName="SYSTEM\CurrentControlSet\Control\MultipathIoClaimTest",

        [string]$ValueName="DeviceList"
    )

    [string[]]$newValues = @()
    $regKey = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey($type, $Srv)
    $key = $regKey.OpenSubKey($KeyName, 'true')
    $values = $key.GetValue($ValueName)
    foreach($val in $values)
    {
        # Only copy values that don't match the given hardware ID.
        if ($val -ne $Hwid)
        {
            $newValues += $val
            Write-Debug "$($val) will remain in the key."
        }
        else
        {
            Write-Debug "$($val) will be removed from the key."
        }
    }
    $key.SetValue($ValueName, [string[]]$newValues, 'MultiString')
}

function HardwareIdClaimed
```

```
{
    param
    (
        [Parameter(Mandatory=$True)]
        [string]$Hwid,

        [string]$Srv="127.0.0.1",

        [string]$KeyName="SYSTEM\CurrentControlSet\Control\MultipathIoClaimTest",

        [string]$ValueName="DeviceList"
    )

    $regKey = [Microsoft.Win32.RegistryKey]::OpenRemoteBaseKey($type, $Srv)
    $key = $regKey.OpenSubKey($KeyName)
    $values = $key.GetValue($ValueName)
    foreach($val in $values)
    {
        if ($val -eq $Hwid)
        {
            return 'true'
        }
    }

    return 'false'
}

function GetBusTypeName
{
    param
    (
        [Parameter(Mandatory=$True)]
        [string]$Hwid
    )

    if ($Hwid -eq $fcHwid)
    {
        return "Fibre Channel"
    }
    elseif ($Hwid -eq $sasHwid)
    {
        return "SAS"
    }
    elseif ($Hwid -eq $iscsiHwid)
    {
        return "iSCSI"
    }

    return "Unknown"
}

#
# Execution starts here.
#

#
# Create the list of hardware IDs to claim or unclaim.
#
[string[]]$hwids = @()

if ($BusType -eq 'fc')
{
    $hwids += $fcHwid
}
elseif ($BusType -eq 'iscsi')
{
    $hwids += $iscsiHwid
}
elseif ($BusType -eq 'sas')
```

```
{
    $hwids += $sasHwid
}
elseif ($BusType -eq 'all')
{
    $hwids += $fcHwid
    $hwids += $sasHwid
    $hwids += $iscsiHwid
}
else
{
    Write-Host "Please provide a bus type (FC, iSCSI, SAS, or All)."
}

$changed = 'false'

#
# Attempt to claim or unclaim each of the hardware IDs.
#
foreach($hwid in $hwids)
{
    $busTypeName = GetBusTypeName $hwid

    #
    # The device is only considered claimed if it's in both the MPIO and MSDSM lists.
    #
    $mpioClaimed = HardwareIdClaimed $hwid $Server $mpioKeyName $mpioValueName
    $msdsmClaimed = HardwareIdClaimed $hwid $Server $msdsmKeyName $msdsmValueName
    if ($mpioClaimed -eq 'true' -and $msdsmClaimed -eq 'true')
    {
        $claimed = 'true'
    }
    else
    {
        $claimed = 'false'
    }

    if ($mpioClaimed -eq 'true')
    {
        Write-Debug "$($hwid) is in the MPIO list."
    }
    else
    {
        Write-Debug "$($hwid) is NOT in the MPIO list."
    }

    if ($msdsmClaimed -eq 'true')
    {
        Write-Debug "$($hwid) is in the MSDSM list."
    }
    else
    {
        Write-Debug "$($hwid) is NOT in the MSDSM list."
    }

    if ($Unclaim)
    {
        #
        # Unclaim this hardware ID.
        #
        if ($claimed -eq 'true')
        {
            RemoveHardwareId $hwid $Server $mpioKeyName $mpioValueName
            RemoveHardwareId $hwid $Server $msdsmKeyName $msdsmValueName
            $changed = 'true'
            Write-Host "$($busTypeName) devices will not be claimed."
        }
        else
        {
```

```
        {
            Write-Host "$($busTypeName) devices are not currently claimed."
        }

    }
    else
    {
        #
        # Claim this hardware ID.
        #
        if ($claimed -eq 'true')
        {
            Write-Host "$($busTypeName) devices are already claimed."
        }
        else
        {
            AddHardwareId $hwid $Server $mpioKeyName $mpioValueName
            AddHardwareId $hwid $Server $msdsmKeyName $msdsmValueName
            $changed = 'true'
            Write-Host "$($busTypeName) devices will be claimed."
        }
    }
}

#
# Finally, if we changed any of the registry keys remind the user to restart.
#
if ($changed -eq 'true')
{
    Write-Host "The system must be restarted for the changes to take effect."
}
```

Manage Nano Server

6/20/2018 • 12 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

Nano Server is managed remotely. There is no local logon capability at all, nor does it support Terminal Services. However, you have a wide variety of options for managing Nano Server remotely, including Windows PowerShell, Windows Management Instrumentation (WMI), Windows Remote Management, and Emergency Management Services (EMS).

To use any remote management tool, you will probably need to know the IP address of the Nano Server. Some ways to find out the IP address include:

- Use the Nano Recovery Console (see the Using the Nano Server Recovery Console section of this topic for details).
- Connect a serial cable to the computer and use EMS.
- Using the computer name you assigned to the Nano Server while configuring it, you can get the IP address with ping. For example, `ping NanoServer-PC /4`.

Using Windows PowerShell remoting

To manage Nano Server with Windows PowerShell remoting, you need to add the IP address of the Nano Server to your management computer's list of trusted hosts, add the account you are using to the Nano Server's administrators, and enable CredSSP if you plan to use that feature.

NOTE

If the target Nano Server and your management computer are in the same AD DS forest (or in forests with a trust relationship), you should not add the Nano Server to the trusted hosts list--you can connect to the Nano Server by using its fully qualified domain name, for example: PS C:> Enter-PSSession -ComputerName nanoserver.contoso.com -Credential (Get-Credential)

To add the Nano Server to the list of trusted hosts, run this command at an elevated Windows PowerShell prompt:

```
Set-Item WSMan:\localhost\Client\TrustedHosts "<IP address of Nano Server>"
```

To start the remote Windows PowerShell session, start an elevated local Windows PowerShell session, and then run these commands:

```
$ip = "\<IP address of Nano Server>"
$user = "$ip\Administrator"
Enter-PSSession -ComputerName $ip -Credential $user
```

You can now run Windows PowerShell commands on the Nano Server as normal.

> **NOTE**
>
> Not all Windows PowerShell commands are available in this release of Nano Server. To see which are available, run `Get-Command -CommandType Cmdlet`

Stop the remote session with the command `Exit-PSSession`

Using Windows PowerShell CIM sessions over WinRM

You can use CIM sessions and instances in Windows PowerShell to run WMI commands over Windows Remote Management (WinRM).

Start the CIM session by running these commands in a Windows PowerShell prompt:

```
$ip = "<IP address of the Nano Server\>"
$ip\Administrator
$cim = New-CimSession -Credential $user -ComputerName $ip
```

With the session established, you can run various WMI commands, for example:

```
Get-CimInstance -CimSession $cim -ClassName Win32_ComputerSystem | Format-List *
Get-CimInstance -CimSession $Cim -Query "SELECT * from Win32_Process WHERE name LIKE 'p%'"
```

Windows Remote Management

You can run programs remotely on the Nano Server with Windows Remote Management (WinRM). To use WinRM, first configure the service and set the code page with these commands at an elevated command prompt:

winrm quickconfig

winrm set winrm/config/client @{TrustedHosts="<ip address of Nano Server"}

chcp 65001

Now you can run commands remotely on the Nano Server. For example:

winrs -r:<IP address of Nano Server> -u:Administrator -p:<Nano Server administrator password> ipconfig

For more information about Windows Remote Management, see Windows Remote Management (WinRM) Overview.

Running a network trace on Nano Server

Netsh trace, Tracelog.exe, and Logman.exe are not available in Nano Server. To capture network packets, you can use these Windows PowerShell cmdlets:

```
New-NetEventSession [-Name]
Add-NetEventPacketCaptureProvider -SessionName
Start-NetEventSession [-Name]
Stop-NetEventSession [-Name]
```

These cmdlets are documented in detail at Network Event Packet Capture Cmdlets in Windows PowerShell

Installing servicing packages

If you want install a servicing packages, use the -ServicingPackagePath parameter (you can pass an array of paths to .cab files):

```
New-NanoServerImage -DeploymentType Guest -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base -
TargetPath .\NanoServer.wim -ServicingPackagePath \\path\to\kb123456.cab
```

Often, a servicing package or hotfix is downloaded as a KB item which contains a .cab file. Follow these steps to extract the .cab file, which you can then install with the -ServicingPackagePath parameter:

1. Download the servicing package (from the associated Knowledge Base article or from Microsoft Update Catalog. Save it to a local directory or network share, for example: C:\ServicingPackages
2. Create a folder in which you will save the extracted servicing package. Example: c:\KB3157663_expanded
3. Open a Windows PowerShell console and use the `Expand` command specifying the path to the .msu file of the servicing package, including the `-f:*` parameter and the path where you want servicing package to be extracted to. For example:

   ```
   Expand "C:\ServicingPackages\Windows10.0-KB3157663-x64.msu" -f:* "C:\KB3157663_expanded"
   ```

   The expanded files should look similar to this:
   C:>dir C:\KB3157663_expanded
   Volume in drive C is OS
   Volume Serial Number is B05B-CC3D

   Directory of C:\KB3157663_expanded

   04/19/2016 01:17 PM <DIR> .
   04/19/2016 01:17 PM <DIR> ..
   04/17/2016 12:31 AM 517 Windows10.0-KB3157663-x64-pkgProperties.txt
   04/17/2016 12:30 AM 93,886,347 Windows10.0-KB3157663-x64.cab
   04/17/2016 12:31 AM 454 Windows10.0-KB3157663-x64.xml
   04/17/2016 12:36 AM 185,818 WSUSSCAN.cab
   4 File(s) 94,073,136 bytes
   2 Dir(s) 328,559,427,584 bytes free

4. Run `New-NanoServerImage` with the -ServicingPackagePath parameter pointing to the .cab file in this directory, for example:

   ```
   New-NanoServerImage -DeploymentType Guest -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath .\Base
   -TargetPath .\NanoServer.wim -ServicingPackagePath C:\KB3157663_expanded\Windows10.0-KB3157663-x64.cab
   ```

Managing updates in Nano Server

Currently you can use the Windows Update provider for Windows Management Instrumentation (WMI) to find the list of applicable updates, and then install all or a subset of them. If you use Windows Server Update Services (WSUS), you can also configure Nano Server to contact the WSUS server to obtain updates.

In all cases, first establish a remote Windows PowerShell session to the Nano Server computer. These examples use *$sess* for the session; if you are using something else, replace that element as needed.

View all available updates

Obtain the full list of applicable updates with these commands:

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

$scanResults = Invoke-CimMethod -InputObject $sess -MethodName ScanForUpdates -Arguments
@{SearchCriteria="IsInstalled=0";OnlineScan=$true}
```

Note:
If no updates are available, this command will return the following error:

```
Invoke-CimMethod : A general error occurred that is not covered by a more specific error code.

At line:1 char:16

+ ... anResults = Invoke-CimMethod -InputObject $sess -MethodName ScanForUp ...

+                 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    + CategoryInfo          : NotSpecified: (MSFT_WUOperatio...-5b842a3dd45d")

   :CimInstance) [Invoke-CimMethod], CimException

    + FullyQualifiedErrorId : MI RESULT 1,Microsoft.Management.Infrastructure.

   CimCmdlets.InvokeCimMethodCommand
```

Install all available updates

You can detect, download, and install **all** available updates at one time by using these commands:

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

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

Restart-Computer
```

Note:
Windows Defender will prevent updates from installing. To work around this, uninstall Windows Defender, install the updates, and then reinstall Windows Defender. Alternately, you can download the updates on another computer, copy them to the Nano Server, and then apply them with DISM.exe.

Verify installation of updates

Use these commands to get a list of the updates currently installed:

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

$scanResults = Invoke-CimMethod -InputObject $sess -MethodName ScanForUpdates -Arguments
@{SearchCriteria="IsInstalled=1";OnlineScan=$true}
```

Note:
These commands list what is installed, but do not specifically quote "installed" in the output. If you need output including that, such as for a report, you can run

```
Get-WindowsPackage--Online
```

Using WSUS

The commands listed above will query the Windows Update and Microsoft Update serviceon the Internet to find and download updates. If you use WSUS, you can set registry keys on the Nano Server to use your WSUS server instead.

See the "Windows Update Agent Environment Options Registry Keys" table in Configure Automatic Updates in a Non-Active-Directory Environment

You should set at least the **WUServer** and **WUStatusServer** registry keys, but depending on how you have implemented WSUS, other values might be needed. You can always confirm these settings by examining another Windows Server in the same environment.

Once these values are set for your WSUS, the commands in the section above will query that server for updates and use it the download source.

Automatic Updates

Currently, the way to automate update installation is to convert the steps above into a local Windows PowerShell script and then create a scheduled task to run it and restart the system on your schedule.

Performance and event monitoring on Nano Server

Nano Server fully supports the Event Tracing for Windows (ETW) framework, but some familiar tools used to manage tracing and performance counters are not currently available on Nano Server. However, Nano Server has tools and cmdlets to accomplish most common performance analysis scenarios.

The high-level workflow remains the same as on any Window Server installation -- low-overhead tracing is performed on the target (Nano Server) computer, and the resulting trace files and/or logs are post-processed offline on a separate computer using tools such as Windows Performance Analyzer, Message Analyzer, or others.

> **NOTE**
> Refer to How to copy files to and from Nano Server for a refresher on how to transfer files using PowerShell remoting.

The following sections list the most common performance data collection activities along with a supported way to accomplish them on Nano Server.

Query available event providers

Windows Performance Recorder is tool to query available event providers as follows:

```
wpr.exe -providers
```

You can filter the output on the type of events that are of interest. For example:

```
PS C:\> wpr.exe -providers | select-string "Storage"

       595f33ea-d4af-4f4d-b4dd-9dacdd17fc6e                            : Microsoft-Windows-StorageManagement-
WSP-Host
       595f7f52-c90a-4026-a125-8eb5e083f15e                            : Microsoft-Windows-StorageSpaces-
Driver
       69c8ca7e-1adf-472b-ba4c-a0485986b9f6                            : Microsoft-Windows-StorageSpaces-
SpaceManager
       7e58e69a-e361-4f06-b880-ad2f4b64c944                            : Microsoft-Windows-StorageManagement
       88c09888-118d-48fc-8863-e1c6d39ca4df                            : Microsoft-Windows-StorageManagement-
WSP-Spaces
```

Record traces from a single ETW provider

You can use new Event Tracing Management cmdlets for this. Here is an example workflow:

Create and start the trace, specifying a file name for storing the events.

```
PS C:\> New-EtwTraceSession -Name "ExampleTrace" -LocalFilePath c:\etrace.etl
```

Add a provider GUID to the trace. Use `wpr.exe -providers` for Provider Name to GUID translation.

```
PS C:\> wpr.exe -providers | select-string "Kernel-Memory"

       d1d93ef7-e1f2-4f45-9943-03d245fe6c00                              : Microsoft-Windows-Kernel-Memory

PS C:\> Add-EtwTraceProvider -Guid "{d1d93ef7-e1f2-4f45-9943-03d245fe6c00}" -SessionName "ExampleTrace"
```

Remove the trace -- this stops the trace session, flushing events to the associated log file.

```
PS C:\> Remove-EtwTraceSession -Name "ExampleTrace"

PS C:\> dir .\etrace.etl

    Directory: C:\

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        9/14/2016  11:17 AM       16515072 etrace.etl
```

NOTE

This example shows adding a single trace provider to the session, but you can also use the `Add-EtwTraceProvider` cmdlet multiple times on a trace session with different provider GUIDs to enable tracing from multiple sources. Another alternative is to use `wpr.exe` profiles described below.

Record traces from multiple ETW providers

The `-profiles` option of Windows Performance Recorder enables tracing from multiple providers at the same time. There are a number of built-in profiles like CPU, Network, and DiskIO to choose from:

```
PS C:\Users\Administrator\Documents> wpr.exe -profiles

Microsoft Windows Performance Recorder Version 10.0.14393 (CoreSystem)
Copyright (c) 2015 Microsoft Corporation. All rights reserved.

        GeneralProfile              First level triage
        CPU                         CPU usage
        DiskIO                      Disk I/O activity
        FileIO                      File I/O activity
        Registry                    Registry I/O activity
        Network                     Networking I/O activity
        Heap                        Heap usage
        Pool                        Pool usage
        VirtualAllocation           VirtualAlloc usage
        Audio                       Audio glitches
        Video                       Video glitches
        Power                       Power usage
        InternetExplorer            Internet Explorer
        EdgeBrowser                 Edge Browser
        Minifilter                  Minifilter I/O activity
        GPU                         GPU activity
        Handle                      Handle usage
        XAMLActivity                XAML activity
        HTMLActivity                HTML activity
        DesktopComposition          Desktop composition activity
        XAMLAppResponsiveness       XAML App Responsiveness analysis
        HTMLResponsiveness          HTML Responsiveness analysis
        ReferenceSet                Reference Set analysis
        ResidentSet                 Resident Set analysis
        XAMLHTMLAppMemoryAnalysis   XAML/HTML application memory analysis
        UTC                         UTC Scenarios
        DotNET                      .NET Activity
        WdfTraceLoggingProvider     WDF Driver Activity
```

For detailed guidance on creating custom profiles, see the WPR.exe documentation.

Record ETW traces during operating system boot time

Use the `New-AutologgerConfig` cmdlet to collect events during system boot. Usage is very similar to the `New-EtwTraceSession` cmdlet, but providers added to the Autologger's configuration will only be enabled early at next boot. The overall workflow looks like this:

First, create a new Autologger config.

```
PS C:\> New-AutologgerConfig -Name "BootPnpLog" -LocalFilePath c:\bootpnp.etl
```

Add a ETW provider to it. This example uses the Kernel PnP provider. Invoke `Add-EtwTraceProvider` again, specifying the same Autologger name but a different GUID to enable boot trace collection from multiple sources.

```
Add-EtwTraceProvider -Guid "{9c205a39-1250-487d-abd7-e831c6290539}" -AutologgerName BootPnpLog
```

This does not start an ETW session immediately, but rather configures one to start at next boot. After rebooting, a new ETW session with the Autologger configuration name is automatically started with the added trace providers enabled. After Nano Server boots, the following command will stop the trace session after flushing the logged events to the associated trace file:

```
PS C:\> Remove-EtwTraceSession -Name BootPnpLog
```

To prevent another trace session from being auto-created at next boot, remove the Autologger configuration as

follows:

```
PS C:\> Remove-AutologgerConfig -Name BootPnpLog
```

To collect boot and setup traces across a number of systems or on a diskless system, consider using Setup and Boot Event Collection.

Capture performance counter data

Usually, you monitor performance counter data with Perfmon.exe GUI. On Nano Server, use the `Typeperf.exe` command-line equivalent. For example:

Query available counters--you can filter the output to easily find the ones of interest.

```
PS C:\> typeperf.exe -q | Select-String "UDPv6"

\UDPv6\Datagrams/sec
\UDPv6\Datagrams Received/sec
\UDPv6\Datagrams No Port/sec
\UDPv6\Datagrams Received Errors
\UDPv6\Datagrams Sent/sec
```

Options allow specifying the number of times and the interval at which counter values are collected. In the example below, Processor Idle Time is collected 5 times every 3 seconds.

```
PS C:\> typeperf.exe "\Processor Information(0,0)\% Idle Time" -si 3 -sc 5

"(PDH-CSV 4.0)","\\ns-g2\Processor Information(0,0)\% Idle Time"
"09/15/2016 09:20:56.002","99.982990"
"09/15/2016 09:20:59.002","99.469634"
"09/15/2016 09:21:02.003","99.990081"
"09/15/2016 09:21:05.003","99.990454"
"09/15/2016 09:21:08.003","99.998577"
Exiting, please wait...
The command completed successfully.
```

Other command-line options allow you to specify performance counter names of interest in a configuration file, redirecting output to a log file, among other things. See the typeperf.exe documentation for details.

You can also use Perfmon.exe's graphical interface remotely with Nano Server targets. When adding performance counters to the view, specify the Nano Server target in the computer name instead of the default .

Interact with the Windows Event Log

Nano Server supports the `Get-WinEvent` cmdlet, which provides Windows Event Log filtering and querying capabilities, both locally as well as on a remote computer. Detailed options and examples are available at the Get-WinEvent documentation page. This simple example retrieves the *Errors* noted in the *System* log during the past two days.

```
PS C:\> $StartTime = (Get-Date) - (New-TimeSpan -Day 2)
PS C:\> Get-WinEvent -FilterHashTable @{LogName='System'; Level=2; StartTime=$StartTime} | select TimeCreated,
Message

TimeCreated           Message
-----------           -------
9/15/2016 11:31:19 AM Task Scheduler service failed to start Task Compatibility module. Tasks may not be able
to reg...
9/15/2016 11:31:16 AM The Virtualization Based Security enablement policy check at phase 6 failed with status:
{File...
9/15/2016 11:31:16 AM The Virtualization Based Security enablement policy check at phase 0 failed with status:
{File...
```

Nano Server also supports `wevtutil.exe` which allows retrieving information about event logs and publishers. See wevtutil.exe documentation for more details.

Graphical interface tools

Web-based server management tools can be used to remotely manage Nano Server targets and present a Nano Server Event Log by using a web browser. Finally, the MMC snap-in Event Viewer (eventvwr.msc) can also be used to view logs -- just open it on a computer with a desktop and point it to a remote Nano Server.

Using Windows PowerShell Desired State Configuration with Nano Server

You can manage Nano Server as target nodes with Windows PowerShell Desired State Configuration (DSC). Currently, you can manage nodes running Nano Server with DSC in push mode only. Not all DSC features function with Nano Server.

For full details, see Using DSC on Nano Server.

Updating Nano Server

11/6/2018 • 6 minutes to read • Edit Online

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

Nano Server offers a variety of methods for staying up to date. Compared to other installation options of Windows Server, Nano Server follows a more active servicing model similar to that of Windows 10. These periodic releases are known as **Current Branch for Business (CBB)** releases. This approach supports customers who want to innovate more quickly and move at a cloud cadence of rapid development lifecycles. More information about CBB is available on the Windows Server Blog.

Between these CBB releases, Nano Server stays current with a series of *cumulative updates*. For example, the first cumulative update for Nano Server was released on September 26, 2016 with KB4093120. With this and subsequent cumulative updates, we provide various options for installing these updates on Nano Server. In this article, we'll use the KB3192366 update as an example to illustrate how to obtain and apply cumulative updates to Nano Server. For more information on the cumulative update model, see the Microsoft Update blog.

NOTE

If you install an optional Nano Server package from media or online repository, it won't have recent security fixes included. To avoid a version mismatch between the optional packages and base operating system, you should install the latest cumulative update immediately after installing any optional packages and **before** restarting the server.

In the case of the Cumulative Update for Windows Server 2016: September 26, 2016 (KB3192366), you should first install the latest Servicing Stack Update for Windows 10 Version 1607: August 23, 2016 as a prerequisite (KB3176936). For most of the options below, you need the .msu files containing the .cab update packages. Visit the Microsoft Update Catalog to download each of these update packages:

- https://catalog.update.microsoft.com/v7/site/Search.aspx?q=KB3192366
- https://catalog.update.microsoft.com/v7/site/Search.aspx?q=KB3176936

After downloading the .msu files from the Microsoft Update Catalog, save them to a network share or local directory such as C:\ServicingPackages. You can rename the .msu files based on their KB number as we've done below to make them easier to identify. Then use the EXPAND utility to extract the .cab files from the .msu files into separate directories and copy the .cabs into a single folder.

```
mkdir C:\ServicingPackages_expanded
mkdir C:\ServicingPackages_expanded\KB3176936
mkdir C:\ServicingPackages_expanded\KB3192366
Expand C:\ServicingPackages\KB3176936.msu -F:* C:\ServicingPackages_expanded\KB3176936
Expand C:\ServicingPackages\KB3192366.msu -F:* C:\ServicingPackages_expanded\KB3192366
mkdir C:\ServicingPackages_cabs
copy C:\ServicingPackages_expanded\KB3176936\Windows10.0-KB3176936-x64.cab C:\ServicingPackages_cabs
copy C:\ServicingPackages_expanded\KB3192366\Windows10.0-KB3192366-x64.cab C:\ServicingPackages_cabs
```

Now you can use the extracted .cab files to apply the updates to a Nano Server image in a few different ways, depending on your needs. The following options are presented in no particular order of preference - use the option

that makes the most sense for your environment.

> **NOTE**
> When using the DISM tools to service Nano Server, you must use a version of DISM that is the same as or newer than the version of Nano Server you're servicing. You can achieve this by running DISM from a matching version of Windows, installing a matching version of the Windows Asssessment and Deployment Kit (ADK), or running DISM on Nano Server itself.

Option 1: Integrate a cumulative update into a new image

If you are building a new Nano Server image, you can integrate the latest cumulative update directly into the image so that it's fully patched on first boot.

```
New-NanoServerImage -ServicingPackagePath 'C:\ServicingPackages_cabs\Windows10.0-KB3176936-x64.cab',
'C:\ServicingPackages_cabs\Windows10.0-KB3192366-x64.cab' -<other parameters>
```

Option 2: Integrate a cumulative update into an existing image

If you have an existing Nano Server image that you use as a baseline for creating specific instances of Nano Server, you can integrate the latest cumulative update directly into your existing baseline image so that machines created using the image are fully patched on first boot.

```
Edit-NanoServerImage -ServicingPackagePath 'C:\ServicingPackages_cabs\Windows10.0-KB3176936-x64.cab',
'C:\ServicingPackages_cabs\Windows10.0-KB3192366-x64.cab' -TargetPath .\NanoServer.wim
```

Option 3: Apply the cumulative update to an existing offline VHD or VHDX

If you have an existing virtual hard disk (VHD or VHDX), you can use the DISM tools to apply the update to the virtual hard disk. You need to make sure the disk is not in use either by shutting down any VMs using the disk or unmounting the virtual hard disk file.

- Using PowerShell

```
Mount-WindowsImage -ImagePath .\NanoServer.vhdx -Path .\MountDir -Index 1
Add-WindowsPackage -Path .\MountDir -PackagePath  C:\ServicingPackages_cabs
Dismount-WindowsImage -Path .\MountDir -Save
```

- Using dism.exe

```
dism.exe /Mount-Image /ImageFile:C:\NanoServer.vhdx /Index:1 /MountDir:C:\MountDir
dism.exe /Image:C:\MountDir /Add-Package /PackagePath:C:\ServicingPackages_cabs
dism.exe /Unmount-Image /MountDir:C:\MountDir /Commit
```

Option 4: Apply the cumulative update to a running Nano Server

If you have a running Nano Server VM or physical host and you've downloaded the .cab file for the update, you can use the DISM tools to apply the update while the operating system is online. You will need to copy the .cab file locally on the Nano Server or to an accessible network location. If you're applying a servicing stack update, make sure to restart the server after applying the servicing stack update before applying additional updates.

NOTE

If you've created the Nano Server VHD or VHDX image using the New-NanoServerImage cmdlet and didn't specify a MaxSize for the virtual hard disk file, the default size of 4GB is too small to apply the cumulative update. Prior to installing the update, use Hyper-V Manager, Disk Management, PowerShell, or other tool to expand the size of the virtual hard disk and system volume to at least 10GB, or use the ScratchDir parameter on the DISM tools to set the scratch directory to a volume with at least 10GB of free space.

```
$s = New-PSSession -ComputerName (Read-Host "Enter Nano Server IP address") -Credential (Get-Credential)
Copy-Item -ToSession $s -Path C:\ServicingPackages_cabs -Destination C:\ServicingPackages_cabs -Recurse
Enter-PSSession $s
```

- Using PowerShell

```
# Apply the servicing stack update first and then restart
Add-WindowsPackage -Online -PackagePath C:\ServicingPackages_cabs\Windows10.0-KB3176936-x64.cab
Restart-Computer; exit

# After restarting, apply the cumulative update and then restart
Enter-PSSession -ComputerName (Read-Host "Enter Nano Server IP address") -Credential (Get-Credential)
Add-WindowsPackage -Online -PackagePath C:\ServicingPackages_cabs\Windows10.0-KB3192366-x64.cab
Restart-Computer; exit
```

- Using dism.exe

```
# Apply the servicing stack update first and then restart
dism.exe /Online /Add-Package /PackagePath:C:\ServicingPackages_cabs\Windows10.0-KB3176936-x64.cab

# After the operation completes successfully and you are prompted to restart, it's safe to
# press Ctrl+C to cancel the pipeline and return to the prompt
Restart-Computer; exit

# After restarting, apply the cumulative update and then restart
Enter-PSSession -ComputerName (Read-Host "Enter Nano Server IP address") -Credential (Get-Credential)
dism.exe /Online /Add-Package /PackagePath:C:\ServicingPackages_cabs\Windows10.0-KB3192366-x64.cab
Restart-Computer; exit
```

Option 5: Download and install the cumulative update to a running Nano Server

If you have a running Nano Server VM or physical host, you can use the Windows Update WMI provider to download and install the update while the operating system is online. With this method, you don't need to download the .msu file separately from the Microsoft Update Catalog. The WMI provider will detect, download, and install all available updates at once.

```
Enter-PSSession -ComputerName (Read-Host "Enter Nano Server IP address") -Credential (Get-Credential)
```

- Scan for available updates

```
$ci = New-CimInstance -Namespace root/Microsoft/Windows/WindowsUpdate -ClassName
MSFT_WUOperationsSession
$result = $ci | Invoke-CimMethod -MethodName ScanForUpdates -Arguments
@{SearchCriteria="IsInstalled=0";OnlineScan=$true}
$result.Updates
```

- Install all available updates

```
$ci = New-CimInstance -Namespace root/Microsoft/Windows/WindowsUpdate -ClassName
MSFT_WUOperationsSession
Invoke-CimMethod -InputObject $ci -MethodName ApplyApplicableUpdates
Restart-Computer; exit
```

- Get a list of installed updates

```
$ci = New-CimInstance -Namespace root/Microsoft/Windows/WindowsUpdate -ClassName
MSFT_WUOperationsSession
$result = $ci | Invoke-CimMethod -MethodName ScanForUpdates -Arguments
@{SearchCriteria="IsInstalled=1";OnlineScan=$true}
$result.Updates
```

Additional Options

Other methods for updating Nano Server might overlap or complement the options above. Such options include using Windows Server Update Services (WSUS), System Center Virtual Machine Manager (VMM), Task Scheduler, or a non-Microsoft solution.

- Configuring Windows Update for WSUS by setting the following registry keys:
  - WUServer
  - WUStatusServer (generally uses the same value as WUServer)
  - UseWUServer
  - AUOptions
- Managing Fabric Updates in VMM
- Registering a Scheduled Task

Developing for Nano Server

6/20/2018 • 2 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

These topics explain important differences in PowerShell on Nano Server and also provide guidance for developing your own PowerShell cmdlets for use on Nano Server.

- PowerShell on Nano Server
- Developing PowerShell Cmdlets for Nano Server

Using Windows PowerShell remoting

To manage Nano Server with Windows PowerShell remoting, you need to add the IP address of the Nano Server to your management computer's list of trusted hosts, add the account you are using to the Nano Server's administrators, and enable CredSSP if you plan to use that feature.

NOTE

If the target Nano Server and your management computer are in the same AD DS forest (or in forests with a trust relationship), you should not add the Nano Server to the trusted hosts list--you can connect to the Nano Server by using its fully qualified domain name, for example: PS C:> Enter-PSSession -ComputerName nanoserver.contoso.com -Credential (Get-Credential)

To add the Nano Server to the list of trusted hosts, run this command at an elevated Windows PowerShell prompt:

```
Set-Item WSMan:\localhost\Client\TrustedHosts "<IP address of Nano Server>"
```

To start the remote Windows PowerShell session, start an elevated local Windows PowerShell session, and then run these commands:

```
$ip = "\<IP address of Nano Server>"
$user = "$ip\Administrator"
Enter-PSSession -ComputerName $ip -Credential $user
```

You can now run Windows PowerShell commands on the Nano Server as normal.

NOTE

Not all Windows PowerShell commands are available in this release of Nano Server. To see which are available, run `Get-Command -CommandType Cmdlet`

Stop the remote session with the command `Exit-PSSession`

Using Windows PowerShell CIM sessions over WinRM

You can use CIM sessions and instances in Windows PowerShell to run WMI commands over Windows Remote Management (WinRM).

Start the CIM session by running these commands in a Windows PowerShell prompt:

```
$ip = "<IP address of the Nano Server\>"
$ip\Administrator
$cim = New-CimSession -Credential $user -ComputerName $ip
```

With the session established, you can run various WMI commands, for example:

```
Get-CimInstance -CimSession $cim -ClassName Win32_ComputerSystem | Format-List *
Get-CimInstance -CimSession $Cim -Query "SELECT * from Win32_Process WHERE name LIKE 'p%'"
```

PowerShell on Nano Server

6/20/2018 • 3 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

PowerShell Editions

Starting with version 5.1, PowerShell is available in different editions which denote varying feature sets and platform compatibility.

- **Desktop Edition:** Built on .NET Framework and provides compatibility with scripts and modules targeting versions of PowerShell running on full footprint editions of Windows such as Server Core and Windows Desktop.
- **Core Edition:** Built on .NET Core and provides compatibility with scripts and modules targeting versions of PowerShell running on reduced footprint editions of Windows such as Nano Server and Windows IoT.

The running edition of PowerShell is shown in the PSEdition property of $PSVersionTable.

```
$PSVersionTable

Name                           Value
----                           -----
PSVersion                      5.1.14300.1000
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
CLRVersion                     4.0.30319.42000
BuildVersion                   10.0.14300.1000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
```

Module authors can declare their modules to be compatible with one or more PowerShell editions using the CompatiblePSEditions module manifest key. This key is only supported on PowerShell 5.1 or later.

```
New-ModuleManifest -Path .\TestModuleWithEdition.psd1 -CompatiblePSEditions Desktop,Core -PowerShellVersion 5.1
$moduleInfo = Test-ModuleManifest -Path \TestModuleWithEdition.psd1
$moduleInfo.CompatiblePSEditions
Desktop
Core

$moduleInfo | Get-Member CompatiblePSEditions

   TypeName: System.Management.Automation.PSModuleInfo

Name                 MemberType Definition
----                 ---------- ----------
CompatiblePSEditions Property   System.Collections.Generic.IEnumerable[string] CompatiblePSEditions {get;}
```

When getting a list of available modules, you can filter the list by PowerShell edition.

```
Get-Module -ListAvailable | ? CompatiblePSEditions -Contains "Desktop"

    Directory: C:\Program Files\WindowsPowerShell\Modules


ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   1.0        ModuleWithPSEditions

Get-Module -ListAvailable | ? CompatiblePSEditions -Contains "Core" | % CompatiblePSEditions
Desktop
Core
```

Script authors can prevent a script from executing unless it is run on a compatible edition of PowerShell using the PSEdition parameter on a #requires statement.

```
Set-Content C:\script.ps1 -Value "#requires -PSEdition Core
Get-Process -Name PowerShell"
Get-Content C:\script.ps1
#requires -PSEdition Core
Get-Process -Name PowerShell

C:\script.ps1
C:\script.ps1 : The script 'script.ps1' cannot be run because it contained a "#requires" statement for
PowerShell editions 'Core'. The edition of PowerShell that is required by the script does not match the
currently running PowerShell Desktop edition.
At line:1 char:1
+ C:\script.ps1
+ ~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (script.ps1:String) [], RuntimeException
    + FullyQualifiedErrorId : ScriptRequiresUnmatchedPSEdition
```

Differences in PowerShell on Nano Server

Nano Server includes PowerShell Core by default in all Nano Server installations. PowerShell Core is a reduced footprint edition of PowerShell that is built on .NET Core and runs on reduced footprint editions of Windows, such as Nano Server and Windows IoT Core. PowerShell Core functions in the same way as other editions of PowerShell, such as Windows PowerShell running on Windows Server 2016. However, the reduced footprint of Nano Server means that not all PowerShell features from Windows Server 2016 are available in PowerShell Core on Nano Server.

Windows PowerShell features not available in Nano Server

- ADSI, ADO, and WMI type adapters
- Enable-PSRemoting, Disable-PSRemoting (PowerShell remoting is enabled by default; see the "Using Windows PowerShell Remoting" section of Install Nano Server).
- Scheduled jobs and PSScheduledJob module
- Computer cmdlets for joining a domain { Add | Remove } (for different methods to join Nano Server to a domain, see the "Joining Nano Server to a domain" section of Install Nano Server).
- Reset-ComputerMachinePassword, Test-ComputerSecureChannel
- Profiles (you can add a startup script for incoming remote connections with `Set-PSSessionConfiguration`)
- Clipboard cmdlets
- EventLog cmdlets { Clear | Get | Limit | New | Remove | Show | Write } (use the New-WinEvent and Get-WinEvent cmdlets instead).
- Get-PfxCertificate cmdlet

- TraceSource cmdlets { Get | Set }
- Counter cmdlets { Get | Export | Import }
- Some web-related cmdlets { New-WebServiceProxy, Send-MailMessage, ConvertTo-Html }
- Logging and tracing using PSDiagnostics module
- Get-HotFix (to obtain and manage updates on Nano Server, see Manage Nano Server).
- Implicit remoting cmdlets { Export-PSSession | Import-PSSession }
- New-PSTransportOption
- PowerShell transactions and Transaction cmdlets { Complete | Get | Start | Undo | Use }
- PowerShell Workflow infrastructure, modules, and cmdlets
- Out-Printer
- Update-List
- WMI v1 cmdlets: Get-WmiObject, Invoke-WmiMethod, Register-WmiEvent, Remove-WmiObject, Set-WmiInstance (use CimCmdlets module instead.)

Using Windows PowerShell Desired State Configuration with Nano Server

You can manage Nano Server as target nodes with Windows PowerShell Desired State Configuration (DSC). Currently, you can manage nodes running Nano Server with DSC in push mode only. Not all DSC features function with Nano Server.

For full details, see Using DSC on Nano Server.

Developing PowerShell Cmdlets for Nano Server

6/20/2018 • 9 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

Overview

Nano Server includes PowerShell Core by default in all Nano Server installations. PowerShell Core is a reduced-footprint edition of PowerShell that is built on .NET Core and runs on reduced-footprint editions of Windows, such as Nano Server and Windows IoT Core. PowerShell Core functions in the same way as other editions of PowerShell, such as Windows PowerShell running on Windows Server 2016. However, the reduced footprint of Nano Server means that not all PowerShell features from Windows Server 2016 are available in PowerShell Core on Nano Server.

If you have existing PowerShell cmdlets that you'd like to run on Nano Server, or are developing new ones for that purpose, this topic includes tips and suggestions that should help make that easier.

PowerShell editions

Starting with version 5.1, PowerShell is available in different editions which denote varying feature sets and platform compatibility.

- **Desktop Edition:** Built on .NET Framework and provides compatibility with scripts and modules targeting versions of PowerShell running on full footprint editions of Windows such as Server Core and Windows Desktop.
- **Core Edition:** Built on .NET Core and provides compatibility with scripts and modules targeting versions of PowerShell running on reduced footprint editions of Windows such as Nano Server and Windows IoT.

The running edition of PowerShell is shown in the PSEdition property of $PSVersionTable.

```
$PSVersionTable

Name                           Value
----                           -----
PSVersion                      5.1.14300.1000
PSEdition                      Desktop
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0...}
CLRVersion                     4.0.30319.42000
BuildVersion                   10.0.14300.1000
WSManStackVersion              3.0
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
```

Module authors can declare their modules to be compatible with one or more PowerShell editions using the CompatiblePSEditions module manifest key. This key is only supported on PowerShell 5.1 or later.

```
New-ModuleManifest -Path .\TestModuleWithEdition.psd1 -CompatiblePSEditions Desktop,Core -PowerShellVersion 5.1
$moduleInfo = Test-ModuleManifest -Path \TestModuleWithEdition.psd1
$moduleInfo.CompatiblePSEditions
Desktop
Core

$moduleInfo | Get-Member CompatiblePSEditions

   TypeName: System.Management.Automation.PSModuleInfo

Name                 MemberType Definition
----                 ---------- ----------
CompatiblePSEditions Property   System.Collections.Generic.IEnumerable[string] CompatiblePSEditions {get;}
```

When getting a list of available modules, you can filter the list by PowerShell edition.

```
Get-Module -ListAvailable | ? CompatiblePSEditions -Contains "Desktop"

    Directory: C:\Program Files\WindowsPowerShell\Modules


ModuleType Version    Name                                ExportedCommands
---------- -------    ----                                ----------------
Manifest   1.0        ModuleWithPSEditions

Get-Module -ListAvailable | ? CompatiblePSEditions -Contains "Core" | % CompatiblePSEditions
Desktop
Core
```

Script authors can prevent a script from executing unless it is run on a compatible edition of PowerShell using the PSEdition parameter on a #requires statement.

```
Set-Content C:\script.ps1 -Value "#requires -PSEdition Core
Get-Process -Name PowerShell"
Get-Content C:\script.ps1
#requires -PSEdition Core
Get-Process -Name PowerShell

C:\script.ps1
C:\script.ps1 : The script 'script.ps1' cannot be run because it contained a "#requires" statement for
PowerShell editions 'Core'. The edition of PowerShell that is required by the script does not match the
currently running PowerShell Desktop edition.
At line:1 char:1
+ C:\script.ps1
+ ~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (script.ps1:String) [], RuntimeException
    + FullyQualifiedErrorId : ScriptRequiresUnmatchedPSEdition
```

Installing Nano Server

Quick-start and detailed steps for installing Nano Server on virtual or physical machines are provided in Install Nano Server, which is the parent topic for this one.

NOTE

For development work on Nano Server, you might find it useful to install Nano Server by using the -Development parameter of New-NanoServerImage. This will enable installation of unsigned drivers, copy debugger binaries, open a port for debugging, enable test signing and enable installation of AppX packages without a developer license. For example:

```
New-NanoServerImage -DeploymentType Guest -Edition Standard -MediaPath \\Path\To\Media\en_us -BasePath
.\Base -TargetPath .\NanoServer.wim -Development
```

Determining the type of cmdlet implementation

PowerShell supports a number of implementation types for cmdlets, and the one you've used determines the process and tools involved in creating or porting it to work on Nano Server. Supported implementation types are:

- CIM - consists of CDXML files layered over CIM (WMIv2) providers
- .NET - consists of .NET assemblies implementing managed cmdlet interfaces, typically written in C#
- PowerShell Script - consists of script modules (.psm1) or scripts (.ps1) written in the PowerShell language

If you're not sure which implementation you've used for existing cmdlets you want to port, install your product or feature and then look for the PowerShell module folder in one of the following locations:

- %windir%\system32\WindowsPowerShell\v1.0\Modules
- %ProgramFiles%\WindowsPowerShell\Modules
- %UserProfile%\Documents\WindowsPowerShell\Modules
- <your product installation location>

  Check in these locations for these details:

  - CIM cmdlets have .cdxml file extensions.
  - .NET cmdlets have .dll file extensions, or have assemblies installed to the GAC listed in the .psd1 file under the RootModule, ModuleToProcess, or NestedModules fields.
- PowerShell script cmdlets have .psm1 or .ps1 file extensions.

Porting CIM cmdlets

Generally, these cmdlets should work in Nano Server without any conversion necessary. However, you must port the underlying WMI v2 provider to run on Nano Server if that has not already been done.

Building C++ for Nano Server

To get C++ DLLs working on Nano Server, compile them for Nano Server rather than for a specific edition.

For prerequisites and a walkthrough of developing C++ on Nano Server, see Developing Native Apps on Nano Server.

Porting .NET cmdlets

Most C# code is supported on Nano Server. You can use ApiPort to scan for incompatible APIs.

Powershell Core SDK

The module "Microsoft.PowerShell.NanoServer.SDK" is available in the PowerShell Gallery to facilitate developing .NET cmdlets using Visual Studio 2015 Update 2 that target the versions of CoreCLR and PowerShell Core available in Nano Server. You can install the module using PowerShellGet with this command:

```
Find-Module Microsoft.PowerShell.NanoServer.SDK -Repository PSGallery | Install-Module -Scope <scope>
```

The PowerShell Core SDK module exposes cmdlets to set up the correct CoreCLR and PowerShell Core reference

assemblies, create a C# project in Visual Studio 2015 targeting those reference assemblies, and set up the remote debugger on a Nano Server machine so that developers can debug their .NET cmdlets running on Nano Server remotely in Visual Studio 2015.

The PowerShell Core SDK module requires Visual Studio 2015 Update 2. If you do not have Visual Studio 2015 installed, you can install Visual Studio Community 2015.

The SDK module also depends on the following feature to be installed in Visual Studio 2015:

- Windows and Web Development -> Universal Windows App Development Tools -> Tools (1.3.1) and Windows 10 SDK

Review your Visual Studio installation before using the SDK module to ensure these prerequisites are satisfied. Make sure you select to install the above feature during the Visual Studio installation, or modify your existing Visual Studio 2015 installation to install it.

The PowerShell Core SDK module includes the following cmdlets:

- New-NanoCSharpProject: Creates a new Visual Studio C# project targeting CoreCLR and PowerShell Core included in the Windows Server 2016 release of Nano Server.
- Show-SdkSetupReadMe: Opens the SDK root folder in File Explorer and opens the README.txt file for manual setup.
- Install-RemoteDebugger: Installs and configures the Visual Studio remote debugger on a Nano Server machine.
- Start-RemoteDebugger: Starts the remote debugger on a remote machine running Nano Server.
- Stop-RemoteDebugger: Stops the remote debugger on a remote machine running Nano Server.

For detailed information about how to use those cmdlets, run Get-Help on each cmdlet after installing and importing the module as follows:

```
Get-Command -Module Microsoft.PowerShell.NanoServer.SDK | Get-Help -Full
```

Searching for compatible APIs

You can search in the API catalog for .NET Core or disassemble Core CLR reference assemblies. For more information about platform portability of .NET APIs, see Platform Portability

PInvoke

In the Core CLR that Nano Server uses, some fundamental DLLs such as kernel32.dll and advapi32.dll were split into numerous API sets, so you'll need to ensure that your PInvokes reference the correct API. Any incompatibility will manifest as a runtime error.

For a list of native APIs supported on Nano Server, see Nano Server APIs.

Building C# for Nano Server

Once a C# project is created in Visual Studio 2015 by using `New-NanoCSharpProject`, you can simply build it in Visual Studio by clicking the **Build** menu and selecting **Build Project** or **Build Solution**. The generated assemblies will be targeting the correct CoreCLR and PowerShell Core shipped in Nano Server, and you can just copy the assemblies to a computer running Nano Server and use them.

Building managed C++ (CPP/CLI) for Nano Server

Managed C++ is not supported for CoreCLR. When porting to CoreCLR, rewrite managed C++ code in C# and make all native calls through PInvoke.

Porting PowerShell script cmdlets

PowerShell Core has full PowerShell language parity with other editions of PowerShell, including the edition running on Windows Server 2016 and Windows 10. However, when porting PowerShell script cmdlets to Nano

Server, keep these factors in mind:

- Are there dependencies on other cmdlets? If so, are those cmdlets available on Nano Server. See PowerShell on Nano Server for information about what is not available.
- If you have dependencies on assemblies that are loaded at runtime, will they still work?
- How can you debug the script remotely?
- How can you migrate from WMI .Net to MI .Net?

Dependency on built-in cmdlets

Not all cmdlets in Windows Server 2016 are available on Nano Server (see PowerShell on Nano Server). The best approach is to set up a Nano Server virtual machine and discover whether the cmdlets you need are available. To do this, run `Enter-PSSession` to connect to the target Nano Server and then run `Get-Command -CommandType Cmdlet, Function` to get the list of available cmdlets.

Consider using PowerShell classes

Add-Type is supported on Nano Server for compiling inline C# code. If you're writing new code or porting existing code, you might also consider using PowerShell classes to define custom types. You can use PowerShell classes for property bag scenarios as well as for Enums. If you need to do a PInvoke, do this via C# using Add-Type or in a pre-compiled assembly.
Here's a sample showing the use of Add-Type:

```
Add-Type -ReferencedAssemblies ([Microsoft.Management.Infrastructure.Ciminstance].Assembly.Location) -
TypeDefinition @'
public class TestNetConnectionResult
{
   // The compute name
   public string ComputerName = null;
   // The Remote IP address used for connectivity
   public System.Net.IPAddress RemoteAddress = null;
}
'@
# Create object and set properties
$result = New-Object TestNetConnectionResult
$result.ComputerName = "Foo"
$result.RemoteAddress = 1.1.1.1
```

This sample shows using PowerShell classes on Nano Server:

```
class TestNetConnectionResult
{
   # The compute name
  [string] $ComputerName

  #The Remote IP address used for connectivity
  [System.Net.IPAddress] $RemoteAddress
}
# Create object and set properties
$result = [TestNetConnectionResult]::new()
$result.ComputerName = "Foo"
$result.RemoteAddress = 1.1.1.1
```

Remotely debugging scripts

To remotely debug a script, connect to the remote computer using `Enter-PSsession` from the PowerShell ISE. Once inside the session, you can run `psedit <file_path>` and a copy of the file will be open in your local PowerShell ISE. Then, you can debug the script as if it were running locally by setting breakpoints. Also, any changes you make to this file will be saved in the remote version.

Migrating from WMI .NET to MI .NET

WMI .NET is not supported, so all cmdlets using the old API must migrate to the supported WMI API: MI. NET. You can access MI .NET directly through C# or through the cmdlets in the CimCmdlets module.

CimCmdlets module

The WMI v1 cmdlets (e.g., Get-WmiObject) are not supported on Nano Server. However, the CIM cmdlets (e.g., Get-CimInstance) in the CimCmdlets module are supported. The CIM cmdlets map pretty closely to the WMI v1 cmdlets. For example, Get-WmiObject correlates with Get-CimInstance using very similar parameters. Method invocation syntax is slightly different, but is well documented via Invoke-CimMethod. Be careful regarding parameter typing. MI .NET has stricter requirements regarding method parameter types.

C# API

WMI .NET wraps the WMIv1 interface, while MI .NET wraps the WMIv2 (CIM) interface. The classes exposed might be different, but the underlying operations are very similar. You enumerate or get instances of objects and invoke operations on them to accomplish tasks.

Troubleshooting Nano Server

6/20/2018 • 3 minutes to read • Edit Online

Applies To: Windows Server 2016

IMPORTANT

Starting in Windows Server, version 1709, Nano Server will be available only as a container base OS image. Check out Changes to Nano Server to learn what this means.

This topic includes information about tools you can use to connect to, diagnose, and repair Nano Server installations.

Using the Nano Server Recovery Console

Nano Server includes a Recovery Console that ensures you can access your Nano Server even if a network mis-configuration interferes with connecting to the Nano Server. You can use the Recovery Console to fix the network and then use your usual remote management tools.

When you boot Nano Server in either a virtual machine or on a physical computer that has a monitor and keyboard attached, you'll see a full-screen, text-mode logon prompt. Log into this prompt with an administrator account to see the computer name and IP address of the Nano Server. You can use these commands to navigate in this console:

- Use arrow keys to scroll
- Use TAB to move to any text that starts with >; then press ENTER to select.
- To go back one screen or page, press ESC. If you're on the home page, pressing ESC will log you off.
- Some screens have additional capabilities displayed on the last line of the screen. For example, if you explore a network adapter, F4 will disable the network adapter.

The Recovery Console allows you to view and configure network adapters and TCP/IP settings, as well as firewall rules.

NOTE

The Recovery Console only supports basic keyboard functions. Keyboard lights, 10-key sections, and keyboard layout switching such as caps lock and number lock are not supported. Only English keyboards and character set are supported.

Accessing Nano Server over a serial port with Emergency Management Services

Emergency Management Services (EMS) lets you perform basic troubleshooting, get network status, and open console sessions (including CMD/PowerShell) by using a terminal emulator over a serial port. This replaces the need for a keyboard and monitor to troubleshoot a server. For more information about EMS, see Emergency Management Services Technical Reference.

To enable EMS on a Nano Server image so that it's ready should you need it later, run this cmdlet:

```
New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath .\EnablingEMS.vhdx -EnableEMS
-EMSPort 3 -EMSBaudRate 9600
```

This example cmdlet enables EMS on serial port 3 with a baud rate of 9600 bps. If you don't include those parameters, the defaults are port 1 and 115200 bps. To use this cmdlet for VHDX media, be sure to include the Hyper-V feature and the corresponding Windows PowerShell modules.

Kernel debugging

You can configure the Nano Server image to support kernel debugging by a variety of methods. To use kernel debugging with a VHDX image, be sure to include the Hyper-V feature and the corresponding Windows PowerShell modules. For more information about remote kernel debugging generally see Setting Up Kernel-Mode Debugging over a Network Cable Manually and Remote Debugging Using WinDbg.

Debugging using a serial port

Use this example cmdlet to enable the image to be debugged using a serial port:

```
New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath .\KernelDebuggingSerial -
DebugMethod Serial -DebugCOMPort 1 -DebugBaudRate 9600
```

This example enables serial debugging over port 2 with a baud rate of 9600 bps. If you don't specify these parameters, the defaults are prot 2 and 115200 bps. If you intend to use both EMS and kernel debugging, you'll have to configure them to use two separate serial ports.

Debugging over a TCP/IP network

Use this example cmdlet to enable the image to be debugged over a TCP/IP network:

```
New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath .\KernelDebuggingNetwork -
DebugMethod Net -DebugRemoteIP 192.168.1.100 -DebugPort 64000
```

This cmdlet enables kernel debugging such that only the computer with the IP address of 192.168.1.100 is allowed to connect, with all communications over port 64000. The -DebugRemoteIP and -DebugPort parameters are mandatory, with a port number greater than 49152. This cmdlet generates an encryption key in a file alongside the resulting VHD which is required for communication over the port. Alternately, you can specify your own key with the -DebugKey parameter, as in this example:

```
New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath .\KernelDebuggingNetwork -
DebugMethod Net -DebugRemoteIP 192.168.1.100 -DebugPort 64000 -DebugKey 1.2.3.4
```

Debugging using the IEEE1394 protocol (Firewire)

To enable debugging over IEEE1394 use this example cmdlet:

```
New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath .\KernelDebuggingFireWire -
DebugMethod 1394 -DebugChannel 3
```

The -DebugChannel parameter is mandatory.

Debugging using USB

You can enable debugging over USB with this cmdlet:

```
New-NanoServerImage -MediaPath \\Path\To\Media\en_us -BasePath .\Base -TargetPath .\KernelDebuggingUSB -
DebugMethod USB -DebugTargetName KernelDebuggingUSBNano
```

When you connect the remote debugger to the resulting Nano Server, specify the target name as set by the -DebugTargetName parameter.

Install Server Core

1/4/2019 • 2 minutes to read • Edit Online

Applies to: Windows Server 2019, Windows Server 2016, Windows Server (Semi-Annual Channel)

When you install Windows Server for the first time, you have the following installation options:

> **NOTE**
> In the following list, editions without "Desktop Experience" are the Server Core installation options

- Windows Server Standard
- Windows Server Standard with Desktop Experience
- Windows Server Datacenter
- Windows Server Datacenter with Desktop Experience

When you install Windows Server (Semi-Annual Channel), including versions 1709, 1803, and 1809, you have the following installation options:

- Windows Server Standard
- Windows Server Datacenter

The Server Core option reduces the space required on disk and the potential attack surface, so we recommend that you choose the Server Core installation unless you have a particular need for the additional user interface elements and graphical management tools that are included in the Server with Desktop Experience option. If you do feel you need the additional user interface elements, see Install Server with Desktop Experience.

With the Server Core option, the standard user interface (the Desktop Experience) is not installed; you manage the server using the command line, Windows PowerShell, or by remote methods.

> **NOTE**
> Unlike some previous releases of Windows Server, you cannot convert between Server Core and Server with Desktop Experience after installation. If you install Server Core and later decide to use Server with Desktop Experience, you should do a fresh installation.

User interface: command prompt

Install, configure, uninstall server roles locally: at a command prompt with Windows PowerShell.

Install, configure, uninstall server roles remotely from a Windows client computer (or a server with the Desktop Experience installed): with Server Manager, Remote Server Administration Tools (RSAT), Windows PowerShell, or Windows Admin Center.

> **NOTE**
> For RSAT, you must use the Windows 10 version. Microsoft Management Console is not available locally.

Example server roles available:

- Active Directory Certificate Services
- Active Directory Domain Services
- DHCP Server
- DNS Server
- File Services (including File Server Resource Manager)
- Active Directory Lightweight Directory Services (AD LDS)
- Hyper-V
- Print and Document Services
- Streaming Media Services
- Web Server (including a subset of ASP.NET)
- Windows Server Update Server
- Active Directory Rights Management Server
- Routing and Remote Access Server and the following sub-roles:
- Remote Desktop Services Connection Broker
- Licensing
- Virtualization
- Volume Activation Services

For roles not included in Server Core, see Roles, Role Services, and Features not in Windows Server - Server Core.

Installing on Windows Server 2019 or Windows Server 2016

For general installation steps and options for Windows Server (Long Term Servicing Channel), see Windows Server Installation and Upgrade.

Installing on Windows Server (Semi-Annual Channel)

Installation steps for Windows Server (Semi-Annual Channel) are the same as installing previous versions of Windows Server (from an .ISO image), with the following exceptions:

- No supported upgrades from previous versions of Windows Server to Windows Server, version 1709. A fresh installation is always required. This means that when you run setup.exe from the desktop of a Windows computer, the setup experience does not permit the upgrade option (it is grayed out).
- There's no evaluation version for Windows Server (Semi-Annual Channel)
- There's no OEM or retail version. Windows Server (Semi-Annual Channel) can only be licensed through Software Assurance or loyalty programs.

To get Windows Server version 1709, see Introducing Windows Server, version 1709.

To get Windows Server version 1803, see Introducing Windows Server, version 1803.

To see what's new in Windows Server, version 1809, see What's New in Windows Server version 1809

Configure a Server Core installation of Windows Server 2016 or Windows Server, version 1709, with Sconfig.cmd

8/8/2018 • 4 minutes to read • Edit Online

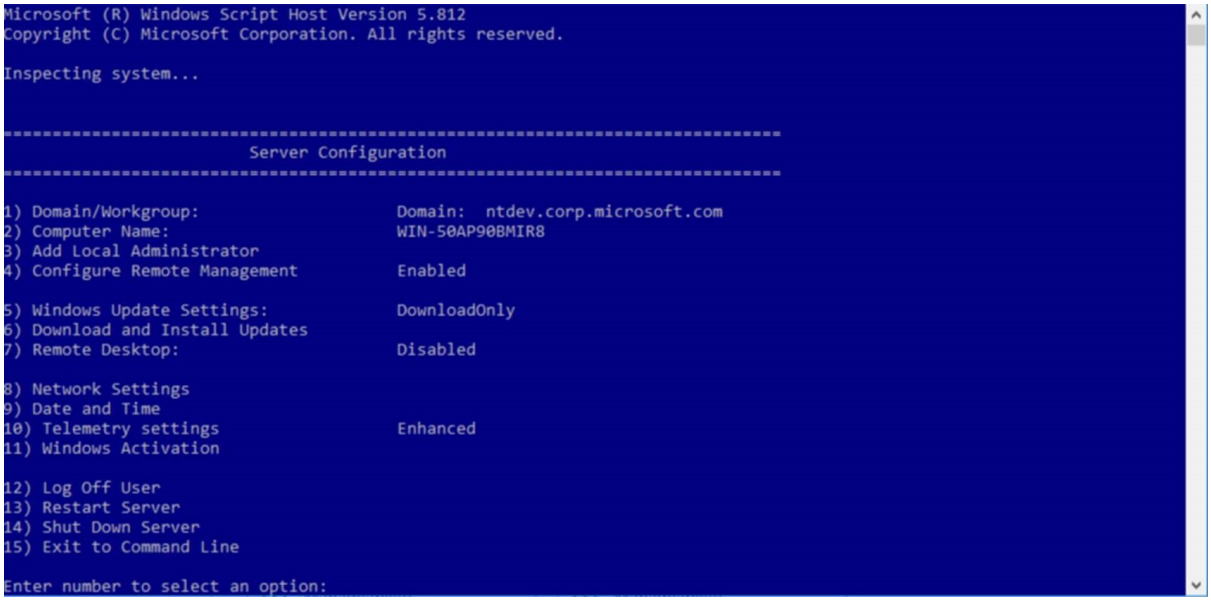
Applies To: Windows Server (Semi-Annual Channel) and Windows Server 2016

In Windows Server 2016 and Windows Server, version 1709, you can use the Server Configuration tool (Sconfig.cmd) to configure and manage several common aspects of Server Core installations. You must be a member of the Administrators group to use the tool.

You can use Sconfig.cmd in Server Core and Server with Desktop Experience (Windows Server 2016 only) installations.

Start the Server Configuration Tool

1. Change to the system drive.
2. Type `Sconfig.cmd`, and then press ENTER. The Server Configuration tool interface opens:



Screenshot of Sconfig.cmd user interface

Domain/Workgroup settings

The current Domain/Workgroup settings are displayed in the default Server Configuration tool screen. You can join a domain or a workgroup by accessing the **Domain/Workgroup** settings page from the main menu and following the instructions on the following pages, supplying any required information.

If a domain user has not been added to the local administrators group, you will not be able to make system changes, such as changing the computer name, by using the domain user. To add a domain user to the local administrators group, allow the computer to restart. Next, log on to the computer as the local administrator and follow the steps in the Local administrator settings section later in this document.

> **NOTE**
> You are required to restart the server to apply any changes to domain or workgroup membership. However, you can make additional changes and restart the server after all of the changes to avoid the need to restart the server multiple times. By default, running virtual machines are automatically saved prior to restarting the Hyper-V Server.

Computer name settings

The current computer name is displayed in the default Server Configuration Tool screen. You can change the computer name by accessing the "Computer Name" settings page from the main menu and following the instructions.

> **NOTE**
> You are required to restart the server to apply any changes to domain or workgroup membership. However, you can make additional changes and restart the server after all of the changes to avoid the need to restart the server multiple times. By default, running virtual machines are automatically saved prior to restarting the Hyper-V Server.

Local administrator settings

To add additional users to the local administrators group, use the **Add Local Administrator** option on the main menu. On a domain joined machine, enter the user in the following format: domain\username. On a non-domain joined machine (workgroup machine), enter only the user name. The changes take effect immediately.

Network settings

You can configure the IP address to be assigned automatically by a DHCP Server or you can assign a static IP address manually. This option allows you to configure DNS Server settings for the server as well.

> **NOTE**
> These options and many more are now available using the Networking Windows PowerShell cmdlets. For more information, see Network Adapter Cmdlets in the Windows Server Library.

Windows Update settings

The current Windows Update settings are displayed in the default Server Configuration Tool screen. You can configure the server to use Automatic or Manual updates on the **Windows Update Settings** configuration option on the main menu.

When **Automatic Updates** are selected, the system will check for and install updates every day at 3:00 AM. The settings take effect immediately. When **Manual** updates are selected, the system will not check for updates automatically.

At any time, you can download and install applicable updates from the **Download and Install Updates** option on the main menu.

The **Download Only** option will scan for updates, download any that are available, and then notify you in the Action Center that they are ready for installation. This is default option.

Remote Desktop settings

The current status of remote desktop settings is displayed in the default Server Configuration Tool screen. You can

configure the following Remote Desktop settings by accessing the **Remote Desktop** main menu option and following the instructions on screen.

- Enable Remote Desktop for Clients running Remote Desktop with Network Level Authentication
- Enable Remote Desktop for clients running any version of Remote Desktop
- Disable Remote

Date and time settings

You can access and change date and time settings by accessing the **Date and Time** main menu option

Telemetry settings

This option lets you configure what data is sent to Microsoft.

Windows Activation settings

This option lets you configure Windows Activation.

To enable remote management

You can enable various remote management scenarios from the **Configure Remote Management** main menu option:

- Microsoft Management Console remote management
- Windows PowerShell
- Server Manager

To log off, restart, or shut down the server

To log off, restart, or shut down the server, access the corresponding menu item from the main menu. These options are also available from the Windows Security menu which can be accessed from any application at any time by pressing CTRL+ALT+DEL.

To exit to the command line

Select the **Exit to the Command Line** option and press ENTER to exit to the command line. To return to the Server Configuration Tool, type **Sconfig.cmd**, and then press ENTER

Install Server with Desktop Experience

8/8/2018 • 6 minutes to read • Edit Online

Applies To: Windows Server 2016

When you install Windows Server 2016 using the Setup wizard, you can choose between **Windows Server 2016** and **Windows Server (Server with Desktop Experience)**. The Server with Desktop Experience option is the Windows Server 2016 equivalent of the Full installation option available in Windows Server 2012 R2 with the Desktop Experience feature installed. If you do not make a choice in the Setup wizard, **Windows Server 2016** is installed; this is the **Server Core** installation option.

The Server with Desktop Experience option installs the standard user interface and all tools, including client experience features that required a separate installation in Windows Server 2012 R2. Server roles and features are installed with Server Manager or by other methods. Compared with the Server Core option, it requires more space on disk,and has higher servicing requirements, so we recommend that you choose the Server Core installation unless you have a particular need for the additional user interface elements and graphical management tools that are included in the Server with Desktop Experience option. If you feel you can work without the additional elements, see Install Server Core. For an even more lightweight option, see Install Nano Server.

> **NOTE**
> Unlike some previous releases of Windows Server, you cannot convert between Server Core and Server with Desktop Experience after installation. If you install Server with Desktop Experience and later decide to use Server Core, you should do a fresh installation.

User interface: standard graphical user interface ("Server Graphical Shell"). The Server Graphical Shell includes the new Windows 10 shell. The specific Windows features installed by default with this option are User-Interfaces-Infra, Server-GUI-Shell, Server-GUI-Mgmt-Infra, InkAndHandwritingServices, ServerMediaFoundation and Desktop Experience. While these features do appear in Server Manager in this release, uninstalling them is not supported and they will not be available in future releases.

Install, configure, uninstall server roles locally: with Server Manager or with Windows PowerShell

Install, configure, uninstall server roles remotely: with Server Manager, Remote Server, RSAT, or Windows PowerShell

Microsoft Management Console: installed

Installation scenarios

Evaluation

You can obtain a 180-day-licensed evaluation copy of Windows Server from Windows Server Evaluations. Choose the **Windows Server 2016 | 64-bit ISO option** to download, or you can visit the **Windows Server 2016 | Virtual Lab**.

> **IMPORTANT**
> For releases of Windows Server 2016 prior to 14393.0.161119-1705.RS1_REFRESH, you can only perform this conversion from evaluation to retail with Windows Server 2016 that has been installed by using the Desktop Experience option (not the Server Core option). Starting with version 14393.0.161119-1705.RS1_REFRESH and later releases, you can convert evaluation editions to retail regardless of the installation option used.

Clean installation

To install the Server with Desktop Experience installation option from the media, insert the media in a drive, restart the computer, and run Setup.exe. In the wizard that opens, select **Windows Server (Server with Desktop Experience)** (Standard or Datacenter), and then complete the wizard.

Upgrade

Upgrade means moving from your existing operating system release to a more recent release while staying on the same hardware.

If you already have a Full installation of the appropriate Windows Server product, you can upgrade it to a Server with Desktop Experience installation of the appropriate edition of Windows Server 2016, as indicated below.

> **IMPORTANT**
> In this release, upgrade works best in virtual machines where specific OEM hardware drivers are not needed for a successful upgrade. Otherwise, migration is the recommended option.

- In-place upgrades from 32-bit to 64-bit architectures are not supported. All editions of Windows Server 2016 are 64-bit only.
- In-place upgrades from one language to another are not supported.
- If the server is a domain controller, see Upgrade Domain Controllers to Windows Server 2012 R2 and Windows Server 2012 for important information.
- Upgrades from pre-release versions (previews) of Windows Server 2016 are not supported. Perform a clean installation to Windows Server 2016.
- Upgrades that switch from a Server Core installation to a Server with a Desktop installation (or vice versa) are not supported.

If you do not see your current version in the left column, upgrading to this release of Windows Server 2016 is not supported.

If you see more than one edition in the right column, upgrading to **either** edition from the same starting version is supported.

IF YOU ARE RUNNING THIS EDITION:	YOU CAN UPGRADE TO THESE EDITIONS:
Windows Server 2012 Standard	Windows Server 2016 Standard or Datacenter
Windows Server 2012 Datacenter	Windows Server 2016 Datacenter
Windows Server 2012 R2 Standard	Windows Server 2016 Standard or Datacenter
Windows Server 2012 R2 Datacenter	Windows Server 2016 Datacenter
Windows Server 2012 R2 Essentials	Windows Server 2016 Essentials

IF YOU ARE RUNNING THIS EDITION:	YOU CAN UPGRADE TO THESE EDITIONS:
Windows Storage Server 2012 Standard	Windows Storage Server 2016 Standard
Windows Storage Server 2012 Workgroup	Windows Storage Server 2016 Workgroup
Windows Storage Server 2012 R2 Standard	Windows Storage Server 2016 Standard
Windows Storage Server 2012 R2 Workgroup	Windows Storage Server 2016 Workgroup

For many additional options for moving to Windows Server 2016, such as license conversion among volume licensed editions, evaluation editions, and others, see details at Upgrade Options.

Migration

Migration means moving from your existing operating system to Windows Server 2016 by performing a clean installation on a different set of hardware or virtual machine and then transferring the older server's workloads to the new server. Migration, which might vary considerably depending on the server roles you have installed, is discussed in detail at Windows Server Installation, Upgrade, and Migration.

The ability to migrate varies among different server roles. The follwogin grid explains your server role upgrade and migration options specifically for moving to Windows Server 2016. For individual role migration guides, visit Migrating Roles and Features in Windows Server. For more information about installation and upgrades, see Windows Server Installation, Upgrade, and Migration.

SERVER ROLE	UPGRADEABLE FROM WINDOWS SERVER 2012 R2?	UPGRADEABLE FROM WINDOWS SERVER 2012?	MIGRATION SUPPORTED?	CAN MIGRATION BE COMPLETED WITHOUT DOWNTIME?
Active Directory Certificate Services	Yes	Yes	Yes	No
Active Directory Domain Services	Yes	Yes	Yes	Yes
Active Directory Federation Services	No	No	Yes	No (new nodes need to be added to the farm)
Active Directory Lightweight Directory Services	Yes	Yes	Yes	Yes
Active Directory Rights Management Services	Yes	Yes	Yes	No

SERVER ROLE	UPGRADEABLE FROM WINDOWS SERVER 2012 R2?	UPGRADEABLE FROM WINDOWS SERVER 2012?	MIGRATION SUPPORTED?	CAN MIGRATION BE COMPLETED WITHOUT DOWNTIME?
Failover Cluster	Yes with Cluster OS Rolling Upgrade process which includes node Pause-Drain, Evict, upgrade to Windows Server 2016 and rejoin the original cluster. Yes, when the server is removed by the cluster for upgrade and then added to a different cluster.	Not while the server is part of a cluster. Yes, when the server is removed by the cluster for upgrade and then added to a different cluster.	Yes	No for Windows Server 2012 Failover Clusters. Yes for Windows Server 2012 R2 Failover Clusters with Hyper-V VMs or Windows Server 2012 R2 Failover Clusters running the Scale-out File Server role. See Cluster OS Rolling Upgrade.
File and Storage Services	Yes	Yes	Varies by sub-feature	No
Print and Fax Services	No	No	Yes (Printbrm.exe)	No
Remote Desktop Services	Yes, for all sub-roles, but mixed mode farm is not supported	Yes, for all sub-roles, but mixed mode farm is not supported	Yes	No
Web Server (IIS)	Yes	Yes	Yes	No
Windows Server Essentials Experience	Yes	N/A – new feature	Yes	No
Windows Server Update Services	Yes	Yes	Yes	No
Work Folders	Yes	Yes	Yes	Yes from WS 2012 R2 cluster when using Cluster OS Rolling Upgrade.

IMPORTANT

Once Setup has completed and immediately after you have installed all of the server roles and features you need, check for and install updates available for Windows Server 2016 by using Windows Update or other update methods.

If you need a different installation option, or if you've completed installation and are ready to deploy specific workloads, you can head back to the main Windows Server 2016 page.

Windows Server release information

11/13/2018 • 2 minutes to read • Edit Online

Microsoft has updated its servicing model. The Semi-Annual Channel is a twice-per-year feature update release with 18-month servicing timelines for each release. This page is designed to help you determine the end of support date for the Semi-Annual Channel releases.

The Semi-Annual Channel provides opportunity for customers who are innovating quickly to take advantage of new operating system capabilities at a faster pace, both in applications - particularly those built on containers and microservices - and in the software-defined hybrid datacenter. For more information see the Windows Server Semi-Annual Channel overview. Customers also have the option to continue using the Long-Term Servicing Channel releases, which continue to be released every 2-3 years. Each Long-Term Servicing Channel release is supported for 5 years of mainstream support and 5 years of extended support.

Windows Server current versions by servicing option

WINDOWS SERVER RELEASE	VERSION	OS BUILD	AVAILABILITY	MAINSTREAM SUPPORT END DATE	EXTENDED SUPPORT END DATE
Windows Server 2019 (Long-Term Servicing Channel) (Datacenter, Essentials, Multipoint Premium Server, Standard)		17763.107.1010129-1455	11/13/2018	01/09/2024	01/09/2029
Windows Server, version 1809 (Semi-Annual Channel) (Datacenter Core, Standard Core)	1809	17763.107.1010129-1455	11/13/2018	5/11/2020	Review note
Windows Server, version 1803 (Semi-Annual Channel) (Datacenter, Standard)	1803	17134.1.180410-1804	04/30/2018	11/12/2019	Review note
Windows Server, version 1709 (Semi-Annual Channel)	1709	16299.15	10/17/2017	04/09/2019	N/A
Windows Server 2016 (Long-Term Servicing Channel)	1607	14393.0	10/15/2016	01/11/2022	01/11/2027

NOTE

Windows Server, version 1803, and Windows Server, version 1809, are governed by the Modern Lifecycle Policy. See the Windows Lifecycle FAQ and Windows Server Semi-Annual Channel overview for details regarding servicing requirements and other important information.