

Virtualization documentation

Virtualization in Windows Server is one of the foundational technologies required to create your software defined infrastructure. Along with networking and storage, virtualization features deliver the flexibility you need to power workloads for your customers.

Windows Containers

OVERVIEW

[Windows Containers](#)

[About Window containers](#)

[Containers vs. virtual machines](#)

Hyper-V on Windows 10

OVERVIEW

[Hyper-V on Windows 10 overview](#)

[About Hyper-V on Windows 10](#)

GET STARTED

[Install Hyper-V](#)

[Create a Virtual Machine](#)

Hyper-V on Windows Server

OVERVIEW

[Hyper-V on Windows Server](#)

[Hyper-V Technology Overview](#)

REFERENCE

[Microsoft Hyper-V Server 2016](#)

Hyper-V Virtual Switch

OVERVIEW

[Hyper-V Virtual Switch](#)

HOW-TO GUIDE

[Manage Hyper-V Virtual Switch](#)

Use a guarded fabric to provide a secure environment for VMs

OVERVIEW

[Guarded Fabric and Shielded VMs](#)

Guarded fabric and shielded VMs

Article • 06/09/2022

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

One of the most important goals of providing a hosted environment is to guarantee the security of the virtual machines running in the environment. As a cloud service provider or enterprise private cloud administrator, you can use a guarded fabric to provide a more secure environment for VMs. A guarded fabric consists of one Host Guardian Service (HGS) - typically, a cluster of three nodes - plus one or more guarded hosts, and a set of shielded virtual machines (VMs).

Important

Ensure that you have installed the latest cumulative update before you deploy shielded virtual machines in production.

Videos, blog, and overview topic about guarded fabrics and shielded VMs

- Video: [How to protect your virtualization fabric from insider threats with Windows Server 2019](#)
- Video: [Introduction to Shielded Virtual Machines in Windows Server 2016](#)
- Video: [Dive into Shielded VMs with Windows Server 2016 Hyper-V](#)
- Video: [Deploying Shielded VMs and a Guarded Fabric with Windows Server 2016](#) ↗
- Blog: [Datacenter and Private Cloud Security Blog](#)
- Overview: [Guarded fabric and shielded VMs overview](#)

Planning topics

- [Planning guide for hosters](#)
- [Planning guide for tenants](#)

Deployment topics

- [Deployment Guide](#)
 - [Quick start](#)

- Deploy HGS
- Deploy guarded hosts
 - Configuring the fabric DNS for hosts that will become guarded hosts
 - Deploy a guarded host using AD mode
 - Deploy a guarded host using TPM mode
 - Confirm guarded hosts can attest
 - Shielded VMs - Hosting service provider deploys guarded hosts in VMM
- Deploy shielded VMs
 - Create a shielded VM template
 - Prepare a VM Shielding helper VHD
 - Set up Windows Azure Pack
 - Create a shielding data file
 - Deploy a shielded VM by using Windows Azure Pack
 - Deploy a shielded VM by using Virtual Machine Manager

Operations and management topic

- Managing the Host Guardian Service

Hyper-V on Windows Server

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Windows Server 2019

The Hyper-V role in Windows Server lets you create a virtualized computing environment where you can create and manage virtual machines. You can run multiple operating systems on one physical computer and isolate the operating systems from each other. With this technology, you can improve the efficiency of your computing resources and free up your hardware resources.

See the topics in the following table to learn more about Hyper-V on Windows Server.

Hyper-V resources for IT Pros

Task	Resources
 Evaluate Hyper-V	<ul style="list-style-type: none">- Hyper-V Technology Overview- What's new in Hyper-V on Windows Server- System requirements for Hyper-V on Windows Server- Supported Windows guest operating systems for Hyper-V- Supported Linux and FreeBSD virtual machines- Feature compatibility by generation and guest
 Plan for Hyper-V	<ul style="list-style-type: none">- Should I create a generation 1 or 2 virtual machine in Hyper-V?- Plan for Hyper-V scalability in Windows Server- Plan for Hyper-V networking in Windows Server- Plan for Hyper-V security in Windows Server
 Get started with Hyper-V	<ul style="list-style-type: none">- Download and install Windows Server 2019 ↗Server Core or GUI installation option of Windows Server 2019 as virtual machine host<ul style="list-style-type: none">- Install the Hyper-V role on Windows Server- Create a virtual switch for Hyper-V virtual machines- Create a virtual machine in Hyper-V

Task	Resources
	<p>Upgrade Hyper-V hosts and virtual machines</p> <ul style="list-style-type: none"> - Upgrade Windows Server cluster nodes - Upgrade virtual machine version <p>Configure and manage Hyper-V</p> <ul style="list-style-type: none"> - Set up hosts for live migration without Failover Clustering - Managing Nano Server remotely - Choose between standard or production checkpoints - Enable or disable checkpoints - Manage Windows virtual machines with PowerShell Direct - Set up Hyper-V Replica
	<p>Blogs</p> <p>Check out the latest posts from Program Managers, Product Managers, Developers and Testers in the Microsoft Virtualization and Hyper-V teams.</p> <ul style="list-style-type: none"> - Virtualization Blog ↗ - Windows Server Blog ↗ - Ben Armstrong's Virtualization Blog (archived)
	<p>Forum and newsgroups</p> <p>Got questions? Talk to your peers, MVPs, and the Hyper-V product team.</p> <ul style="list-style-type: none"> - Windows Server Community ↗ - Windows Server Hyper-V TechNet forum

Related technologies

The following table lists technologies that you might want to use in your virtualization computing environment.

Technology	Description
Client Hyper-V	Virtualization technology included with Windows 8, Windows 8.1 and Windows 10 that you can install through Programs and Features in the Control Panel .
Failover Clustering	Windows Server feature that provides high availability for Hyper-V hosts and virtual machines.
Virtual Machine Manager	System Center component that provides a management solution for the virtualized datacenter. You can configure and manage your virtualization hosts, networks, and storage resources so you can create and deploy virtual machines and services to private clouds that you've created.

Technology	Description
Windows Containers	Use Windows Server and Hyper-V containers to provide standardized environments for development, test, and production teams.

Hyper-V Technology Overview

Article • 07/29/2021 • 5 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Hyper-V is Microsoft's hardware virtualization product. It lets you create and run a software version of a computer, called a *virtual machine*. Each virtual machine acts like a complete computer, running an operating system and programs. When you need computing resources, virtual machines give you more flexibility, help save time and money, and are a more efficient way to use hardware than just running one operating system on physical hardware.

Hyper-V runs each virtual machine in its own isolated space, which means you can run more than one virtual machine on the same hardware at the same time. You might want to do this to avoid problems such as a crash affecting the other workloads, or to give different people, groups or services access to different systems.

Some ways Hyper-V can help you

Hyper-V can help you:

- **Establish or expand a private cloud environment.** Provide more flexible, on-demand IT services by moving to or expanding your use of shared resources and adjust utilization as demand changes.
- **Use your hardware more effectively.** Consolidate servers and workloads onto fewer, more powerful physical computers to use less power and physical space.
- **Improve business continuity.** Minimize the impact of both scheduled and unscheduled downtime of your workloads.
- **Establish or expand a virtual desktop infrastructure (VDI).** Use a centralized desktop strategy with VDI can help you increase business agility and data security, as well as simplify regulatory compliance and manage desktop operating systems and applications. Deploy Hyper-V and Remote Desktop Virtualization Host (RD Virtualization Host) on the same server to make personal virtual desktops or virtual desktop pools available to your users.
- **Make development and test more efficient.** Reproduce different computing environments without having to buy or maintain all the hardware you'd need if you

only used physical systems.

Hyper-V and other virtualization products

Hyper-V in Windows and Windows Server replaces older hardware virtualization products, such as Microsoft Virtual PC, Microsoft Virtual Server, and Windows Virtual PC. Hyper-V offers networking, performance, storage and security features not available in these older products.

Hyper-V and most third-party virtualization applications that require the same processor features aren't compatible. That's because the processor features, known as hardware virtualization extensions, are designed to not be shared. For details, see [Virtualization applications do not work together with Hyper-V, Device Guard, and Credential Guard](#).

What features does Hyper-V have?

Hyper-V offers many features. This is an overview, grouped by what the features provide or help you do.

Computing environment - A Hyper-V virtual machine includes the same basic parts as a physical computer, such as memory, processor, storage, and networking. All these parts have features and options that you can configure different ways to meet different needs. Storage and networking can each be considered categories of their own, because of the many ways you can configure them.

Disaster recovery and backup - For disaster recovery, Hyper-V Replica creates copies of virtual machines, intended to be stored in another physical location, so you can restore the virtual machine from the copy. For backup, Hyper-V offers two types. One uses saved states and the other uses Volume Shadow Copy Service (VSS) so you can make application-consistent backups for programs that support VSS.

Optimization - Each supported guest operating system has a customized set of services and drivers, called *integration services*, that make it easier to use the operating system in a Hyper-V virtual machine.

Portability - Features such as live migration, storage migration, and import/export make it easier to move or distribute a virtual machine.

Remote connectivity - Hyper-V includes Virtual Machine Connection, a remote connection tool for use with both Windows and Linux. Unlike Remote Desktop, this tool gives you console access, so you can see what's happening in the guest even when the operating system isn't booted yet.

Security - Secure boot and shielded virtual machines help protect against malware and other unauthorized access to a virtual machine and its data.

For a summary of the features introduced in this version, see [What's new in Hyper-V on Windows Server](#). Some features or parts have a limit to how many can be configured. For details, see [Plan for Hyper-V scalability in Windows Server 2016](#).

How to get Hyper-V

Hyper-V is available in Windows Server and Windows, as a server role available for x64 versions of Windows Server. For server instructions, see [Install the Hyper-V role on Windows Server](#). On Windows, it's available as [feature](#) in some 64-bit versions of Windows. It's also available as a downloadable, standalone server product, [Microsoft Hyper-V Server](#).

Supported operating systems

Many operating systems will run on virtual machines. In general, an operating system that uses an x86 architecture will run on a Hyper-V virtual machine. Not all operating systems that can be run are tested and supported by Microsoft, however. For lists of what's supported, see:

- [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#)
- [Supported Windows guest operating systems for Hyper-V on Windows Server](#)

How Hyper-V works

Hyper-V is a hypervisor-based virtualization technology. Hyper-V uses the Windows hypervisor, which requires a physical processor with specific features. For hardware details, see [System requirements for Hyper-V on Windows Server](#).

In most cases, the hypervisor manages the interactions between the hardware and the virtual machines. This hypervisor-controlled access to the hardware gives virtual machines the isolated environment in which they run. In some configurations, a virtual machine or the operating system running in the virtual machine has direct access to graphics, networking, or storage hardware.

What does Hyper-V consist of?

Hyper-V has required parts that work together so you can create and run virtual machines. Together, these parts are called the virtualization platform. They're installed as a set when you install the Hyper-V role. The required parts include Windows hypervisor, Hyper-V Virtual Machine Management Service, the virtualization WMI provider, the virtual machine bus (VMbus), virtualization service provider (VSP) and virtual infrastructure driver (VID).

Hyper-V also has tools for management and connectivity. You can install these on the same computer that Hyper-V role is installed on, and on computers without the Hyper-V role installed. These tools are:

- Hyper-V Manager
- [Hyper-V module for Windows PowerShell](#)
- [Virtual Machine Connection](#) (sometimes called VMConnect)
- [Windows PowerShell Direct](#)

Related technologies

These are some technologies from Microsoft that are often used with Hyper-V:

- [Failover Clustering](#)
- [Remote Desktop Services](#)
- [System Center Virtual Machine Manager](#)

Various storage technologies: cluster shared volumes, SMB 3.0, storage spaces direct

Windows containers offer another approach to virtualization. See the [Windows Containers](#) library on MSDN.

What's new in Hyper-V on Windows Server

Article • 07/11/2022 • 15 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Microsoft Hyper-V Server 2016, Windows Server 2016

This article explains the new and changed functionality of Hyper-V on Windows Server 2019, Windows Server 2016, and Microsoft Hyper-V Server 2016. To use new features on virtual machines created with Windows Server 2012 R2 and moved or imported to a server that runs Hyper-V on Windows Server 2019 or Windows Server 2016, you'll need to manually upgrade the virtual machine configuration version. For instructions, see [Upgrade virtual machine version](#).

Here's what's included in this article and whether the functionality is new or updated.

Windows Server, version 1903

Add Hyper-V Manager to Server Core installations (updated)

As you might know, we recommend using the Server Core installation option when using Windows Server, Semi-Annual Channel in production. However, Server Core by default omits a number of useful management tools. You can add many of the most commonly used tools by installing the App Compatibility feature, but there have still been some missing tools.

So, based on customer feedback, we added one more tool to the App Compatibility feature in this version: Hyper-V Manager (`virtmgmt.msc`).

For more info, see [Server Core App Compatibility Feature on Demand \(FOD\)](#).

Windows Server 2019

Security: Shielded Virtual Machines improvements (new)

- Branch office improvements

You can now run shielded virtual machines on machines with intermittent connectivity to the Host Guardian Service by leveraging the new [fallback HGS](#) and [offline mode](#) features. Fallback HGS allows you to configure a second set of URLs for Hyper-V to try if it can't reach your primary HGS server.

Offline mode allows you to continue to start up your shielded VMs, even if HGS can't be reached, as long as the VM has started successfully once, and the host's security configuration has not changed.

- **Troubleshooting improvements**

We've also made it easier to [troubleshoot your shielded virtual machines](#) by enabling support for VMConnect Enhanced Session Mode and PowerShell Direct. These tools are particularly useful if you've lost network connectivity to your VM and need to update its configuration to restore access.

These features do not need to be configured, and they become available automatically when a shielded VM is placed on a Hyper-V host running Windows Server version 1803 or later.

- **Linux support**

If you run mixed-OS environments, Windows Server 2019 now supports running Ubuntu, Red Hat Enterprise Linux, and SUSE Linux Enterprise Server inside shielded virtual machines.

Windows Server 2016

Compatible with Connected Standby (new)

When the Hyper-V role is installed on a computer that uses the Always On/Always Connected (AOAC) power model, the **Connected Standby** power state is now available.

Discrete device assignment (new)

This feature lets you give a virtual machine direct and exclusive access to some PCIe hardware devices. Using a device in this way bypasses the Hyper-V virtualization stack, which results in faster access. For details on supported hardware, see "Discrete device assignment" in [System requirements for Hyper-V on Windows Server 2016](#). For details, including how to use this feature and considerations, see the post "[Discrete Device Assignment — Description and background](#)" in the Virtualization blog.

Encryption support for the operating system disk in generation 1 virtual machines (new)

You can now protect the operating system disk using BitLocker drive encryption in generation 1 virtual machines. A new feature, key storage, creates a small, dedicated drive to store the system drive's BitLocker key. This is done instead of using a virtual Trusted Platform Module (TPM), which is available only in generation 2 virtual machines. To decrypt the disk and start the virtual machine, the Hyper-V host must either be part of an authorized guarded fabric or have the private key from one of the virtual machine's guardians. Key storage requires a version 8 virtual machine. For information on virtual machine version, see [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#).

Host resource protection (new)

This feature helps prevent a virtual machine from using more than its share of system resources by looking for excessive levels of activity. This can help prevent a virtual machine's excessive activity from degrading the performance of the host or other virtual machines. When monitoring detects a virtual machine with excessive activity, the virtual machine is given fewer resources. This monitoring and enforcement is off by default. Use Windows PowerShell to turn it on or off. To turn it on, run this command:

```
Set-VMProcessor TestVM -EnableHostResourceProtection $true
```

For details about this cmdlet, see [Set-VMProcessor](#).

Hot add and remove for network adapters and memory (new)

You can now add or remove a network adapter while the virtual machine is running, without incurring downtime. This works for generation 2 virtual machines that run either Windows or Linux operating systems.

You can also adjust the amount of memory assigned to a virtual machine while it's running, even if you haven't enabled Dynamic Memory. This works for both generation 1 and generation 2 virtual machines, running Windows Server 2016 or Windows 10.

Hyper-V Manager improvements (updated)

- **Alternate credentials support** - You can now use a different set of credentials in Hyper-V Manager when you connect to another Windows Server 2016 or Windows 10 remote host. You can also save these credentials to make it easier to log on again.
- **Manage earlier versions** - With Hyper-V Manager in Windows Server 2019, Windows Server 2016, and Windows 10, you can manage computers running Hyper-V on Windows Server 2012, Windows 8, Windows Server 2012 R2 and Windows 8.1.
- **Updated management protocol** - Hyper-V Manager now communicates with remote Hyper-V hosts using the WS-MAN protocol, which permits CredSSP, Kerberos or NTLM authentication. When you use CredSSP to connect to a remote Hyper-V host, you can do a live migration without enabling constrained delegation in Active Directory. The WS-MAN-based infrastructure also makes it easier to enable a host for remote management. WS-MAN connects over port 80, which is open by default.

Integration services delivered through Windows Update (updated)

Updates to integration services for Windows guests are distributed through Windows Update. For service providers and private cloud hosters, this puts the control of applying updates into the hands of the tenants who own the virtual machines. Tenants can now update their Windows virtual machines with all updates, including the integration services, using a single method. For details about integration services for Linux guests, see [Linux and FreeBSD Virtual Machines on Hyper-V](#).

Important

The vmguest.iso image file is no longer needed, so it isn't included with Hyper-V on Windows Server 2016.

Linux Secure Boot (new)

Linux operating systems running on generation 2 virtual machines can now boot with the Secure Boot option enabled. Ubuntu 14.04 and later, SUSE Linux Enterprise Server 12 and later, Red Hat Enterprise Linux 7.0 and later, and CentOS 7.0 and later are enabled for Secure Boot on hosts that run Windows Server 2016. Before you boot the virtual machine for the first time, you must configure the virtual machine to use the

Microsoft UEFI Certificate Authority. You can do this from Hyper-V Manager, Virtual Machine Manager, or an elevated Windows Powershell session. For Windows PowerShell, run this command:

```
Set-VMFirmware TestVM -SecureBootTemplate MicrosoftUEFICertificateAuthority
```

For more information about Linux virtual machines on Hyper-V, see [Linux and FreeBSD Virtual Machines on Hyper-V](#). For more information about the cmdlet, see [Set-VMFirmware](#).

More memory and processors for generation 2 virtual machines and Hyper-V hosts (updated)

Starting with version 8, generation 2 virtual machines can use significantly more memory and virtual processors. Hosts also can be configured with significantly more memory and virtual processors than were previously supported. These changes support new scenarios such as running e-commerce large in-memory databases for online transaction processing (OLTP) and data warehousing (DW). The Windows Server blog recently published the performance results of a virtual machine with 5.5 terabytes of memory and 128 virtual processors running 4 TB in-memory database. Performance was greater than 95% of the performance of a physical server. For details, see [Windows Server 2016 Hyper-V large-scale VM performance for in-memory transaction processing](#). For details about virtual machine versions, see [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#). For the full list of supported maximum configurations, see [Plan for Hyper-V scalability in Windows Server 2016](#).

Nested virtualization (new)

This feature lets you use a virtual machine as a Hyper-V host and create virtual machines within that virtualized host. This can be especially useful for development and test environments. To use nested virtualization, you'll need:

- To run at least Windows Server 2019, Windows Server 2016 or Windows 10 on both the physical Hyper-V host and the virtualized host.
- A processor with Intel VT-x (nested virtualization is available only for Intel processors at this time).

For details and instructions, see [Run Hyper-V in a Virtual Machine with Nested Virtualization](#).

Networking features (new)

New networking features include:

- **Remote direct memory access (RDMA) and switch embedded teaming (SET).** You can set up RDMA on network adapters bound to a Hyper-V virtual switch, regardless of whether SET is also used. SET provides a virtual switch with some of same capabilities as NIC teaming. For details, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#).
- **Virtual machine multi queues (VMMQ).** Improves on VMQ throughput by allocating multiple hardware queues per virtual machine. The default queue becomes a set of queues for a virtual machine, and traffic is spread between the queues.
- **Quality of service (QoS) for software-defined networks.** Manages the default class of traffic through the virtual switch within the default class bandwidth.

For more about new networking features, see [What's new in Networking](#).

Production checkpoints (new)

Production checkpoints are "point-in-time" images of a virtual machine. These give you a way to apply a checkpoint that complies with support policies when a virtual machine runs a production workload. Production checkpoints are based on backup technology inside the guest instead of a saved state. For Windows virtual machines, the Volume Snapshot Service (VSS) is used. For Linux virtual machines, the file system buffers are flushed to create a checkpoint that's consistent with the file system. If you'd rather use checkpoints based on saved states, choose standard checkpoints instead. For details, see [Choose between standard or production checkpoints in Hyper-V](#).

 **Important**

New virtual machines use production checkpoints as the default.

Rolling Hyper-V Cluster upgrade (new)

You can now add a node running Windows Server 2019 or Windows Server 2016 to a Hyper-V Cluster with nodes running Windows Server 2012 R2. This allows you to upgrade the cluster without downtime. The cluster runs at a Windows Server 2012 R2 feature level until you upgrade all nodes in the cluster and update the cluster functional level with the Windows PowerShell cmdlet, [Update-ClusterFunctionalLevel](#).

Important

After you update the cluster functional level, you can't return it to Windows Server 2012 R2.

For a Hyper-V cluster with a functional level of Windows Server 2012 R2 with nodes running Windows Server 2012 R2, Windows Server 2019 and Windows Server 2016, note the following:

- Manage the cluster, Hyper-V, and virtual machines from a node running Windows Server 2016 or Windows 10.
- You can move virtual machines between all of the nodes in the Hyper-V cluster.
- To use new Hyper-V features, all nodes must run Windows Server 2016 or and the cluster functional level must be updated.
- The virtual machine configuration version for existing virtual machines isn't upgraded. You can upgrade the configuration version only after you upgrade the cluster functional level.
- Virtual machines that you create are compatible with Windows Server 2012 R2, virtual machine configuration level 5.

After you update the cluster functional level:

- You can enable new Hyper-V features.
- To make new virtual machine features available, use the `Update-vmVersion` cmdlet to manually update the virtual machine configuration level. For instructions, see [Upgrade virtual machine version](#).
- You can't add a node to the Hyper-V Cluster that runs Windows Server 2012 R2.

Note

Hyper-V on Windows 10 doesn't support failover clustering.

For details and instructions, see the [Cluster Operating System Rolling Upgrade](#).

Shared virtual hard disks (updated)

You can now resize shared virtual hard disks (.vhdx files) used for guest clustering, without downtime. Shared virtual hard disks can be grown or shrunk while the virtual machine is online. Guest clusters can now also protect shared virtual hard disks by using Hyper-V Replica for disaster recovery.

Enable replication on the collection. Enabling replication on a collection is **only exposed through the WMI interface**. See the documentation for

[MsVm_CollectionReplicationService class](#) for more details. **You cannot manage replication of a collection through PowerShell cmdlet or UI.** The VMs should be on hosts that are part of a Hyper-V cluster to access features that are specific to a collection. This includes Shared VHD - shared VHDs on stand-alone hosts are not supported by Hyper-V Replica.

Follow the guidelines for shared VHDs in [Virtual Hard Disk Sharing Overview](#), and be sure that your shared VHDs are part of a guest cluster.

A collection with a shared VHD but no associated guest cluster cannot create reference points for the collection (regardless of whether the shared VHD is included in the reference point creation or not).

Virtual machine backup(new)

If you are backing up a single virtual machine (regardless of whether host is clustered or not), you should not use a VM group. Nor should you use a snapshot collection. VM groups and snapshot collection are meant to be used solely for backing up guest clusters that are using shared vhdx. Instead, you should take a snapshot using the [Hyper-V WMI v2 provider](#). Likewise, do not use the [Failover Cluster WMI provider](#).

Shielded virtual machines (new)

Shielded virtual machines use several features to make it harder for Hyper-V administrators and malware on the host to inspect, tamper with, or steal data from the state of a shielded virtual machine. Data and state is encrypted, Hyper-V administrators can't see the video output and disks, and the virtual machines can be restricted to run only on known, healthy hosts, as determined by a Host Guardian Server. For details, see [Guarded Fabric and Shielded VMs](#).

Note

Shielded virtual machines are compatible with Hyper-V Replica. To replicate a shielded virtual machine, the host you want to replicate to must be authorized to run that shielded virtual machine.

Start order priority for clustered virtual machines (new)

This feature gives you more control over which clustered virtual machines are started or restarted first. This makes it easier to start virtual machines that provide services before virtual machines that use those services. Define sets, place virtual machines in sets, and specify dependencies. Use Windows PowerShell cmdlets to manage the sets, such as [New-ClusterGroupSet](#), [Get-ClusterGroupSet](#), and [Add-ClusterGroupSetDependency](#). .

Storage quality of service (QoS) (updated)

You can now create storage QoS policies on a Scale-Out File Server and assign them to one or more virtual disks on Hyper-V virtual machines. Storage performance is automatically readjusted to meet policies as the storage load fluctuates. For details, see [Storage Quality of Service](#).

Virtual machine configuration file format (updated)

Virtual machine configuration files use a new format that makes reading and writing configuration data more efficient. The format also makes data corruption less likely if a storage failure occurs. Virtual machine configuration data files use a .vmcx file name extension and runtime state data files use a .vmrs file name extension.

Important

The .vmcx file name extension indicates a binary file. Editing .vmcx or .vmrs files isn't supported.

Virtual machine configuration version (updated)

The version represents the compatibility of the virtual machine's configuration, saved state, and snapshot files with the version of Hyper-V. Virtual machines with version 5 are compatible with Windows Server 2012 R2 and can run on both Windows Server 2012 R2

and Windows Server 2016. Virtual machines with versions introduced in Windows Server 2016 and Windows Server 2019 won't run in Hyper-V on Windows Server 2012 R2.

If you move or import a virtual machine to a server that runs Hyper-V on Windows Server 2016 or Windows Server 2019 from Windows Server 2012 R2, the virtual machine's configuration isn't automatically updated. This means you can move the virtual machine back to a server that runs Windows Server 2012 R2. But, this also means you can't use the new virtual machine features until you manually update the version of the virtual machine configuration.

For instructions on checking and upgrading the version, see [Upgrade virtual machine version](#). This article also lists the version in which some features were introduced.

Important

- After you update the version, you can't move the virtual machine to a server that runs Windows Server 2012 R2.
- You can't downgrade the configuration to a previous version.
- The **Update-VMVersion** cmdlet is blocked on a Hyper-V Cluster when the cluster functional level is Windows Server 2012 R2.

Virtualization-based security for generation 2 virtual machines (new)

Virtualization-based security powers features such as Device Guard and Credential Guard, offering increased protection of the operating system against exploits from malware. Virtualization based-security is available in generation 2 guest virtual machines starting with version 8. For information on virtual machine version, see [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#).

Windows Containers (new)

Windows Containers allow many isolated applications to run on one computer system. They're fast to build and are highly scalable and portable. Two types of container runtime are available, each with a different degree of application isolation. Windows Server Containers use namespace and process isolation. Hyper-V Containers use a light-weight virtual machine for each container.

Key features include:

- Support for web sites and applications using HTTPS
- Nano server can host both Windows Server and Hyper-V Containers
- Ability to manage data through container shared folders
- Ability to restrict container resources

For details, including quick start guides, see the [Windows Containers Documentation](#).

Windows PowerShell Direct (new)

This gives you a way to run Windows PowerShell commands in a virtual machine from the host. Windows PowerShell Direct runs between the host and the virtual machine. This means it doesn't require networking or firewall requirements, and it works regardless of your remote management configuration.

Windows PowerShell Direct is an alternative to the existing tools that Hyper-V administrators use to connect to a virtual machine on a Hyper-V host:

- Remote management tools such as PowerShell or Remote Desktop
- Hyper-V Virtual Machine Connection (VMConnect)

Those tools work well, but have trade-offs: VMConnect is reliable, but can be hard to automate. Remote PowerShell is powerful, but can be hard to set up and maintain. These trade-offs may become more important as your Hyper-V deployment grows. Windows PowerShell Direct addresses this by providing a powerful scripting and automation experience that's as simple as using VMConnect.

For requirements and instructions, see [Manage Windows virtual machines with PowerShell Direct](#).

System requirements for Hyper-V on Windows Server

Article • 07/29/2021 • 3 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Hyper-V has specific hardware requirements, and some Hyper-V features have additional requirements. Use the details in this article to decide what requirements your system must meet so you can use Hyper-V the way you plan to. Then, review the [Windows Server catalog](#). Keep in mind that requirements for Hyper-V exceed the general minimum requirements for Windows Server 2016 because a virtualization environment requires more computing resources.

If you're already using Hyper-V, it's likely that you can use your existing hardware. The general hardware requirements haven't changed significantly from Windows Server 2012 R2. But, you will need newer hardware to use shielded virtual machines or discrete device assignment. Those features rely on specific hardware support, as described below. Other than that, the main difference in hardware is that second-level address translation (SLAT) is now required instead of recommended.

For details about maximum supported configurations for Hyper-V, such as the number of running virtual machines, see [Plan for Hyper-V scalability in Windows Server 2016](#). The list of operating systems you can run in your virtual machines is covered in [Supported Windows guest operating systems for Hyper-V on Windows Server](#).

General requirements

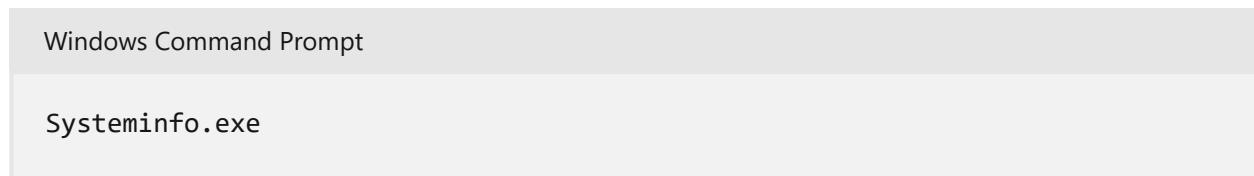
Regardless of the Hyper-V features you want to use, you'll need:

- A 64-bit processor with second-level address translation (SLAT). To install the Hyper-V virtualization components such as Windows hypervisor, the processor must have SLAT. However, it's not required to install Hyper-V management tools like Virtual Machine Connection (VMConnect), Hyper-V Manager, and the Hyper-V cmdlets for Windows PowerShell. See "How to check for Hyper-V requirements," below, to find out if your processor has SLAT.
- VM Monitor Mode extensions

- Enough memory - plan for *at least* 4 GB of RAM. More memory is better. You'll need enough memory for the host and all virtual machines that you want to run at the same time.
- Virtualization support turned on in the BIOS or UEFI:
 - Hardware-assisted virtualization. This is available in processors that include a virtualization option - specifically processors with Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) technology.
 - Hardware-enforced Data Execution Prevention (DEP) must be available and enabled. For Intel systems, this is the XD bit (execute disable bit). For AMD systems, this is the NX bit (no execute bit).

How to check for Hyper-V requirements

Open Windows PowerShell or a command prompt and type:



Scroll to the Hyper-V Requirements section to review the report.

Requirements for specific features

Here are the requirements for discrete device assignment and shielded virtual machines. For descriptions of these features, see [What's new in Hyper-V on Windows Server](#).

Discrete device assignment

Host requirements are similar to the existing requirements for the SR-IOV feature in Hyper-V.

- The processor must have either Intel's Extended Page Table (EPT) or AMD's Nested Page Table (NPT).
- The chipset must have:
 - Interrupt remapping - Intel's VT-d with the Interrupt Remapping capability (VT-d2) or any version of AMD I/O Memory Management Unit (I/O MMU).
 - DMA remapping - Intel's VT-d with Queued Invalidations or any AMD I/O MMU.

- Access control services (ACS) on PCI Express root ports.
- The firmware tables must expose the I/O MMU to the Windows hypervisor. Note that this feature might be turned off in the UEFI or BIOS. For instructions, see the hardware documentation or contact your hardware manufacturer.

Devices need GPU or non-volatile memory express (NVMe). For GPU, only certain devices support discrete device assignment. To verify, see the hardware documentation or contact your hardware manufacturer. For details about this feature, including how to use it and considerations, see the post "[Discrete Device Assignment -- Description and background ↗](#)" in the Virtualization blog.

Shielded virtual machines

These virtual machines rely on virtualization-based security and are available starting with Windows Server 2016.

Host requirements are:

- UEFI 2.3.1c - supports secure, measured boot

The following two are optional for virtualization-based security in general, but required for the host if you want the protection these features provide:

- TPM v2.0 - protects platform security assets
- IOMMU (Intel VT-D) - so the hypervisor can provide direct memory access (DMA) protection

Virtual machine requirements are:

- Generation 2
- Windows Server 2012 or newer as the guest operating system

Supported Windows guest operating systems for Hyper-V on Windows Server

Article • 09/20/2021 • 3 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Windows Server 2019, Azure Stack HCI, version 20H2

Hyper-V supports several versions of Windows Server, Windows, and Linux distributions to run in virtual machines, as guest operating systems. This article covers supported Windows Server and Windows guest operating systems. For Linux and FreeBSD distributions, see [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#).

Some operating systems have the integration services built-in. Others require that you install or upgrade integration services as a separate step after you set up the operating system in the virtual machine. For more information, see the sections below and [Integration Services](#).

Supported Windows Server guest operating systems

Following are the versions of Windows Server that are supported as guest operating systems for Hyper-V on Windows Server.

Guest operating system (server)	Maximum number of virtual processors	Integration Services	Notes
Windows Server 2022	240 for generation 2; 64 for generation 1	Built-in	Hosted on Windows Server 2019 or above, Azure Stack HCI, version 20H2 or above.

Guest operating system (server)	Maximum number of virtual processors	Integration Services	Notes
Windows Server, version 1909	240 for generation 2; 64 for generation 1	Built-in	Greater than 240 virtual processor support requires Windows Server, version 1903 or later guest operating systems.
Windows Server, version 1903	240 for generation 2; 64 for generation 1	Built-in	
Windows Server, version 1809	240 for generation 2; 64 for generation 1	Built-in	
Windows Server 2019	240 for generation 2; 64 for generation 1	Built-in	
Windows Server, version 1803	240 for generation 2; 64 for generation 1	Built-in	
Windows Server 2016	240 for generation 2; 64 for generation 1	Built-in	
Windows Server 2012 R2	64	Built-in	

Guest operating system (server)	Maximum number of virtual processors	Integration Services	Notes
Windows Server 2012	64	Built-in	
Windows Server 2008 R2 with Service Pack 1 (SP 1)	64	Install all critical Windows updates after you set up the guest operating system.	Datacenter, Enterprise, Standard and Web editions.
Windows Server 2008 with Service Pack 2 (SP2)	8	Install all critical Windows updates after you set up the guest operating system.	Datacenter, Enterprise, Standard and Web editions (32-bit and 64-bit).

Supported Windows client guest operating systems

Following are the versions of Windows client that are supported as guest operating systems for Hyper-V on Windows Server.

Guest operating system (client)	Maximum number of virtual processors	Integration Services	Notes
Windows 11	32	Built-in	Generation 2 virtual machine hosted on Windows Server 2019 or above, Azure Stack HCI, version 20H2 or above.
Windows 10	32	Built-in	
Windows 8.1	32	Built-in	
Windows 7 with Service Pack 1 (SP1)	4	Upgrade the integration services after you set up the guest operating system.	Ultimate, Enterprise, and Professional editions (32-bit and 64-bit).

Guest operating system support on other versions of Windows

The following table gives links to information about guest operating systems supported for Hyper-V on other versions of Windows.

Host operating system	Topic
Windows 10	Supported Guest Operating Systems for Client Hyper-V in Windows 10
Windows Server 2012 R2 and Windows 8.1	<ul style="list-style-type: none">- Supported Windows Guest Operating Systems for Hyper-V in Windows Server 2012 R2 and Windows 8.1- Linux and FreeBSD Virtual Machines on Hyper-V
Windows Server 2012 and Windows 8	Supported Windows Guest Operating Systems for Hyper-V in Windows Server 2012 and Windows 8
Windows Server 2008 and Windows Server 2008 R2	About Virtual Machines and Guest Operating Systems

How Microsoft provides support for guest operating systems

Microsoft provides support for guest operating systems in the following manner:

- Issues found in Microsoft operating systems and in integration services are supported by Microsoft support.
- For issues found in other operating systems that have been certified by the operating system vendor to run on Hyper-V, support is provided by the vendor.
- For issues found in other operating systems, Microsoft submits the issue to the multi-vendor support community, [TSANet](#).

Additional References

- [Linux and FreeBSD Virtual Machines on Hyper-V](#)
- [Supported Guest Operating Systems for Client Hyper-V in Windows 10](#)

Supported Linux and FreeBSD virtual machines for Hyper-V on Windows Server and Windows

Article • 08/10/2022 • 2 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

Hyper-V supports both emulated and Hyper-V-specific devices for Linux and FreeBSD virtual machines. When running with emulated devices, no additional software is required to be installed. However emulated devices do not provide high performance and cannot leverage the rich virtual machine management infrastructure that the Hyper-V technology offers. In order to make full use of all benefits that Hyper-V provides, it is best to use Hyper-V-specific devices for Linux and FreeBSD. The collection of drivers that are required to run Hyper-V-specific devices are known as Linux Integration Services (LIS) or FreeBSD Integration Services (BIS).

LIS has been added to the Linux kernel and is updated for new releases. But Linux distributions based on older kernels may not have the latest enhancements or fixes. Microsoft provides a download containing installable LIS drivers for some Linux installations based on these older kernels. Because distribution vendors include versions of Linux Integration Services, it is best to install the latest downloadable version of LIS, if applicable, for your installation.

For other Linux distributions LIS changes are regularly integrated into the operating system kernel and applications so no separate download or installation is required.

For older FreeBSD releases (before 10.0), Microsoft provides ports that contain the installable BIS drivers and corresponding daemons for FreeBSD virtual machines. For newer FreeBSD releases, BIS is built in to the FreeBSD operating system, and no separate download or installation is required except for a KVP ports download that is needed for FreeBSD 10.0.

Tip

- Download [Windows Server](#) from the Evaluation Center.

The goal of this content is to provide information that helps facilitate your Linux or FreeBSD deployment on Hyper-V. Specific details include:

- Linux distributions or FreeBSD releases that require the download and installation of LIS or BIS drivers.
- Linux distributions or FreeBSD releases that contain built-in LIS or BIS drivers.
- Feature distribution maps that indicate the features in major Linux distributions or FreeBSD releases.
- Known issues and workarounds for each distribution or release.
- Feature description for each LIS or BIS feature.

In this section

- [Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V](#)
- [Supported Debian virtual machines on Hyper-V](#)
- [Supported Oracle Linux virtual machines on Hyper-V](#)
- [Supported SUSE virtual machines on Hyper-V](#)
- [Supported Ubuntu virtual machines on Hyper-V](#)
- [Supported FreeBSD virtual machines on Hyper-V](#)
- [Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V](#)
- [Best Practices for running Linux on Hyper-V](#)
- [Best practices for running FreeBSD on Hyper-V](#)

Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V

Article • 02/06/2023 • 19 minutes to read

Applies To: Azure Stack HCI; Windows Server 2022, Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 11, Windows 10, Windows 8.1

The following feature distribution maps indicate the features that are present in built-in and downloadable versions of Linux Integration Services. The known issues and workarounds for each distribution are listed after the tables.

The built-in Red Hat Enterprise Linux Integration Services drivers for Hyper-V (available since Red Hat Enterprise Linux 6.4) are sufficient for Red Hat Enterprise Linux guests to run using the high performance synthetic devices on Hyper-V hosts. These built-in drivers are certified by Red Hat for this use. Certified configurations can be viewed on this Red Hat web page: [Red Hat Certification Catalog](#). It isn't necessary to download and install Linux Integration Services packages from the Microsoft Download Center, and doing so may limit your Red Hat support as described in Red Hat Knowledgebase article 1067: [Red Hat Knowledgebase 1067](#).

Because of potential conflicts between the built-in LIS support and the downloadable LIS support when you upgrade the kernel, disable automatic updates, uninstall the LIS downloadable packages, update the kernel, reboot, and then install the latest LIS release, and reboot again.

ⓘ Note

Official Red Hat Enterprise Linux certification information is available through the [Red Hat Customer Portal](#).

In this section:

- [RHEL/CentOS 9.x Series](#)
- [RHEL/CentOS 8.x Series](#)
- [RHEL/CentOS 7.x Series](#)

- RHEL/CentOS 6.x Series
- RHEL/CentOS 5.x Series
- Notes

Table legend

- **Built in** - LIS are included as part of this Linux distribution. The kernel module version numbers for the built-in LIS (as shown by `lsmod`, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch does not indicate that the built-in LIS is out of date.
- ✓ - Feature available
- (blank) - Feature not available

RHEL/CentOS 9.x Series

Feature	Host OS	9.x
LIS Availability		Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
>256 vCPUs		✓
Networking		
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓

Feature	Host OS	9.x
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
Storage		
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Memory		
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9,10
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
Video		
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Miscellaneous		
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓

Feature	Host OS	9.x
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓
Generation 2 virtual machines		
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14, 17
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓

RHEL/CentOS 8.x Series

Feature	Host OS	8.1-8.6	8.0
LIS Availability		Built in	Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Windows Server 2016 Accurate Time	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
>256 vCPUs		✓	
Networking			
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓

Feature	Host OS	8.1-8.6	8.0
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
Storage			
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3	✓ Note 3
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5	✓ Note 5
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Memory			
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	N/A	N/A
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10	✓ Note 9, 10
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 9, 10	✓ Note 9, 10
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
Video			
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓

Feature	Host OS	8.1-8.6	8.0
Miscellaneous			
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓
Generation 2 virtual machines			
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14, 17	✓ Note 14
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓

RHEL/CentOS 7.x Series

This series only has 64-bit kernels.

Feature	Host OS	7.5- 7.7	7.3- 7.4	7.0- 7.2	7.6- 7.9	7.5	7.4	7.3	7.2	7.1	7.0
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓ Note 4						
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓							
Hyper-V Sockets	Windows Server 2022, 2019, 2016	✓	✓	✓							
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓		✓	✓	✓	✓	✓		

RHEL/CentOS 6.x Series

The 32-bit kernel for this series is PAE enabled. There is no built-in LIS support for RHEL/CentOS 6.0-6.3.

Feature	Host OS	6.7- 6.10	6.4- 6.6	6.0- 6.3	6.10, 6.9,	6.6, 6.7	6.5	6.4 6.8
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5	✓ Note 5	✓ Note 5	✓ Note 4, 5	✓ Note 4, 5	✓ Note 4, 5, 6	✓ Note 4, 5, 6
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓			
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓				
Memory								
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 8, 9, 10	✓ Note 7, 8, 9, 10	✓ Note 7, 8, 9, 10
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 7, 9, 10	✓ Note 7, 9, 10	✓ Note 7, 9, 10, 11				
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI							
Video								
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	

Feature	Host OS	6.7- 6.10	6.4- 6.6	6.0- 6.3	6.10, 6.9,	6.6, 6.7	6.5	6.4 6.8
Miscellaneous								
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓ Note 12	✓ Note 12	✓ Note 12, 13	✓ Note 12, 13
Non-Maskable Interrupt								
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓	✓	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓	✓ Note 12	✓ Note 12		
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓				
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓	✓	✓				
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI	✓						
Generation 2 virtual machines								
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 14	✓ Note 14	✓ Note 14	✓ Note 14			
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI							

RHEL/CentOS 5.x Series

This series has a supported 32-bit PAE kernel available. There is no built-in LIS support for RHEL/CentOS before 5.9.

Feature	Host OS	5.2 - 5.11	5.2-5.11	5.9 - 5.11
LIS Availability		LIS 4.3 ↗	LIS 4.1	Built in
Core	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Windows Server 2016	Windows Server 2022, 2019,			
Accurate Time	2016			
	Azure Stack HCI			
>256 vCPUs				
Networking				
Jumbo frames	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
VLAN tagging and trunking	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 1	✓ Note 1	✓ Note 1
Live Migration	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Static IP Injection	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 2	✓ Note 2	✓ Note 2
vRSS	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
TCP Segmentation and Checksum Offloads	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
SR-IOV	Windows Server 2022, 2019, 2016 Azure Stack HCI			
Storage				

Feature	Host OS	5.2 - 5.11	5.2-5.11	5.9 - 5.11
VHDX resize	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
Virtual Fibre Channel	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 3	✓ Note 3	
Live virtual machine backup	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 5, 15	✓ Note 5	✓ Note 4, 5, 6
TRIM support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
SCSI WWN	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Memory				
PAE Kernel Support	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Configuration of MMIO gap	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
Dynamic Memory - Hot-Add	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Dynamic Memory - Ballooning	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓ Note 7, 9, 10, 11	✓ Note 7, 9, 10, 11	
Runtime Memory Resize	Windows Server 2022, 2019, 2016 Azure Stack HCI			
Video				
Hyper-V-specific video device	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
Miscellaneous				

Feature	Host OS	5.2 -5.11	5.2-5.11	5.9 - 5.11
Key-Value Pair	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
Non-Maskable Interrupt	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	✓
File copy from host to guest	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI	✓	✓	
lsvmbus command	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Hyper-V Sockets	Windows Server 2022, 2019, 2016 Azure Stack HCI			
PCI Passthrough/DDA	Windows Server 2022, 2019, 2016 Azure Stack HCI			
Generation 2 virtual machines				
Boot using UEFI	Windows Server 2022, 2019, 2016, 2012 R2 Azure Stack HCI			
Secure boot	Windows Server 2022, 2019, 2016 Azure Stack HCI			

Notes

1. For this RHEL/CentOS release, VLAN tagging works but VLAN trunking does not.
2. Static IP injection may not work if Network Manager has been configured for a given synthetic network adapter on the virtual machine. For smooth functioning of static IP injection please make sure that either Network Manager is either turned off completely or has been turned off for a specific network adapter through its ifcfg-ethX file.

3. On Windows Server 2012 R2 while using virtual fibre channel devices, make sure that logical unit number 0 (LUN 0) has been populated. If LUN 0 has not been populated, a Linux virtual machine might not be able to mount fibre channel devices natively.
4. For built-in LIS, the "hyperv-daemons" package must be installed for this functionality.
5. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (fsck) on restore. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).
6. While the Linux Integration Services download is preferred, live backup support for RHEL/CentOS 5.9 - 5.11/6.4/6.5 is also available through [Hyper-V Backup Essentials for Linux](#).
7. Dynamic memory support is only available on 64-bit virtual machines.
8. Hot-Add support is not enabled by default in this distribution. To enable Hot-Add support you need to add a udev rule under /etc/udev/rules.d/ as follows:
 - a. Create a file `/etc/udev/rules.d/100-balloon.rules`. You may use any other desired name for the file.
 - b. Add the following content to the file: `SUBSYSTEM=="memory", ACTION=="add", ATTR{state}="online"`
 - c. Reboot the system to enable Hot-Add support.

While the Linux Integration Services download creates this rule on installation, the rule is also removed when LIS is uninstalled, so the rule must be recreated if dynamic memory is needed after uninstallation.

9. Dynamic memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:
 - Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.
 - Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

10. If you are using Dynamic Memory on a Windows Server 2016 or Windows Server 2012 R2 operating system, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to hot-add failures, and you may not see any memory increase in a guest operating system.
11. Certain distributions, including those using LIS 4.0 and 4.1, only provide Ballooning support and do not provide Hot-Add support. In such a scenario, the dynamic memory feature can be used by setting the Startup memory parameter to a value which is equal to the Maximum memory parameter. This results in all the requisite memory being Hot-Added to the virtual machine at boot time and then later depending upon the memory requirements of the host, Hyper-V can freely allocate or deallocate memory from the guest using Ballooning. Please configure **Startup Memory** and **Minimum Memory** at or above the recommended value for the distribution.
12. To enable key/value pair (KVP) infrastructure, install the hypervkvpd or hyperv-daemons rpm package from your RHEL ISO. Alternatively the package can be installed directly from RHEL repositories.
13. The key/value pair (KVP) infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.
14. On Windows Server 2012 R2 Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using PowerShell:

```
Powershell
```

```
Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
```

The Linux Integration Services download can be applied to existing Generation 2 VMs but does not impart Generation 2 capability.

15. In Red Hat Enterprise Linux or CentOS 5.2, 5.3, and 5.4 the filesystem freeze functionality is not available, so Live Virtual Machine Backup is also not available.
16. For RHEL 7.6, support for >256 vcpus is available in kernel 3.10.0-957.38.1 or later and kernel 3.10.0-1062.4.1 or later is required for RHEL 7.7.

17. RHEL 8.5 requires Windows Server 2019 or newer, or Azure Stack HCI 20H2 or newer.

See Also

- [Set-VMFirmware](#)
- [Supported Debian virtual machines on Hyper-V](#)
- [Supported Oracle Linux virtual machines on Hyper-V](#)
- [Supported SUSE virtual machines on Hyper-V](#)
- [Supported Ubuntu virtual machines on Hyper-V](#)
- [Supported FreeBSD virtual machines on Hyper-V](#)
- [Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V](#)
- [Best Practices for running Linux on Hyper-V](#)
- [Red Hat Hardware Certification ↗](#)

Supported Debian virtual machines on Hyper-V

Article • 10/19/2022 • 3 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features that are present in each version. The known issues and workarounds for each distribution are listed after the table.

Table legend

- **Built in** - LIS are included as part of this Linux distribution. The Microsoft-provided LIS download package doesn't work for this distribution so do not install it. The kernel module version numbers for the built in LIS (as shown by `lsmod`, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch does not indicate that the built in LIS is out of date.
- ✓ - Feature available
- (blank) - Feature not available

Feature	Windows Server operating system version	11 (Bullseye)	10.0-10.3 (Buster)
Availability		Built in	Built in
Core	2019, 2016, 2012 R2	✓	✓
Windows Server 2016 Accurate Time	2019, 2016	✓ Note 4	✓ Note 4
Networking			
Jumbo frames	2019, 2016, 2012 R2	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓
Live Migration	2019, 2016, 2012 R2	✓	✓

Feature	Windows Server operating system version	11 (Bullseye)	10.0-10.3 (Buster)
Static IP Injection	2019, 2016, 2012 R2		
vRSS	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
SR-IOV	2019, 2016	✓ Note 4	✓ Note 4
Storage			
VHDX resize	2019, 2016, 2012 R2	✓ Note 1	✓ Note 1
Virtual Fibre Channel	2019, 2016, 2012 R2		
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2
TRIM support	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
SCSI WWN	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
Memory			
PAE Kernel Support	2019, 2016, 2012 R2	✓	✓
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4
Runtime Memory Resize	2019, 2016	✓ Note 4	✓ Note 4
Video			
Hyper-V-specific video device	2019, 2016, 2012 R2	✓	✓
Miscellaneous			
Key-Value Pair	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓
File copy from host to guest	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2
lsvmbus command	2019, 2016, 2012 R2		
Hyper-V Sockets	2019, 2016	✓ Note 4	✓ Note 4
PCI Passthrough/DDA	2019, 2016	✓ Note 4	✓ Note 4

Feature	Windows Server operating system version	11 (Bullseye)	10.0-10.3 (Buster)
Generation 2 virtual machines			
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3
Secure boot	2019, 2016	✓	✓

Notes

1. Creating file systems on VHDs larger than 2TB is not supported.
2. Starting with Debian 8.3 the manually-installed Debian package "hyperv-daemons" contains the key-value pair, fcopy, and VSS daemons. On Debian 7.x and 8.0-8.2 the hyperv-daemons package must come from [Debian backports ↗](#).
3. On Windows Server 2012 R2 Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using Powershell:

Powershell

```
Set-VMFirmware -VMName "Vmname" -EnableSecureBoot Off
```

4. The latest upstream kernel capabilities are only available by using the kernels available in the [Debian Backports repository ↗](#).
5. While Debian 7.x is out of support and uses an older kernel, the kernel included in Debian backports for Debian 7.x has improved Hyper-V capabilities.

See Also

- [Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V](#)
- [Supported Oracle Linux virtual machines on Hyper-V](#)
- [Supported SUSE virtual machines on Hyper-V](#)
- [Supported Ubuntu virtual machines on Hyper-V](#)
- [Supported FreeBSD virtual machines on Hyper-V](#)

- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V

Supported Oracle Linux virtual machines on Hyper-V

Article • 10/19/2022 • 10 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features that are present in each version. The known issues and workarounds for each distribution are listed after the table.

In this section:

- [Oracle Linux 9.x Series](#)
- [Oracle Linux 8.x Series](#)
- [Oracle Linux 7.x Series](#)
- [Oracle Linux 6.x Series](#)

Table legend

- **Built in** - LIS are included as part of this Linux distribution. The kernel module version numbers for the built in LIS (as shown by `lsmod`, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built in LIS is out of date.
- ✓ - Feature available
- (blank) - Feature not available
- RHCK - Red Hat Compatible Kernel
- UEK - Unbreakable Enterprise Kernel (UEK)
 - UEK4 - built on upstream Linux Kernel release 4.1.12
 - UEK5 - built on upstream Linux Kernel release 4.14
 - UEK6 - built on upstream Linux Kernel release 5.4

Oracle Linux 9.x Series

Feature	Windows Server version	9.0 (RHCK)
Availability		
Core	2019, 2016, 2012 R2	✓
Windows Server 2016 Accurate Time	2019, 2016	✓
Networking		
Jumbo frames	2019, 2016, 2012 R2	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓
Live Migration	2019, 2016, 2012 R2	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 2
vRSS	2019, 2016, 2012 R2	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓
SR-IOV	2019, 2016	✓
Storage		
VHDX resize	2019, 2016, 2012 R2	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓
SCSI WWN	2019, 2016, 2012 R2	✓
Memory		
PAE Kernel Support	2019, 2016, 2012 R2	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Runtime Memory Resize	2019, 2016	✓
Video		
Hyper-V-specific video device	2019, 2016, 2012 R2	✓
Miscellaneous		

Feature	Windows Server version	9.0 (RHCK)
Key-Value Pair	2019, 2016, 2012 R2	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓
File copy from host to guest	2019, 2016, 2012 R2	✓
lsvmbus command	2019, 2016, 2012 R2	✓
Hyper-V Sockets	2019, 2016	✓
PCI Passthrough/DDA	2019, 2016	✓
Generation 2 virtual machines		
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12
Secure boot	2019, 2016	✓

Oracle Linux 8.x Series

Feature	Windows Server version	8.0-8.5 (RHCK)
Availability		
Core	2019, 2016, 2012 R2	✓
Windows Server 2016 Accurate Time	2019, 2016	✓
Networking		
Jumbo frames	2019, 2016, 2012 R2	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓
Live Migration	2019, 2016, 2012 R2	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 2
vRSS	2019, 2016, 2012 R2	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓
SR-IOV	2019, 2016	✓
Storage		
VHDX resize	2019, 2016, 2012 R2	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3

Feature	Windows Server version	8.0-8.5 (RHCK)
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓
SCSI WWN	2019, 2016, 2012 R2	✓
Memory		
PAE Kernel Support	2019, 2016, 2012 R2	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 7, 8, 9
Runtime Memory Resize	2019, 2016	✓
Video		
Hyper-V-specific video device	2019, 2016, 2012 R2	✓
Miscellaneous		
Key-Value Pair	2019, 2016, 2012 R2	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓
File copy from host to guest	2019, 2016, 2012 R2	✓
lsvmbus command	2019, 2016, 2012 R2	✓
Hyper-V Sockets	2019, 2016	✓
PCI Passthrough/DDA	2019, 2016	✓
Generation 2 virtual machines		
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12
Secure boot	2019, 2016	✓

Oracle Linux 7.x Series

This series only has 64-bit kernels.

Feature	Windows Server Version	7.5-7.8	7.3-7.4		
		RHCK	UEK5	RHCK	UEK4

Availability		LIS 4.3	Built in	Built in	LIS 4.3	Built in	Built in
Core	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Windows Server Time	2019, 2016 2016 Accurate	✓			✓		
Networking							
Jumbo frames	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Live Migration	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Static IP injection	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2
vRSS	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
SR-IOV	2019, 2016	✓	✓	✓	✓	✓	✓
Storage							
VHDX resize	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5	✓ Note 4,5	✓ Note 5	✓ Note 5	✓ Note 4,5	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
SCSI WWN	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓

Memory							
PAE Kernel Support	2019, 2016, 2012 R2	N/A	N/A	N/A	N/A	N/A	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Dynamic Memory Hot-Add	2019, 2016, 2012 R2	✓ Note 7,8,9	✓ Note 8,9				
Dynamic Memory Ballooning	2019, 2016, 2012 R2	✓ Note 7,8,9	✓ Note 8,9				
Runtime Memory Resize	2019, 2016	✓		✓	✓		
Video							
Hyper-V specific video	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Miscellaneous							
Key-value pair	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
File copy from host to guest	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓
lsvmbus command	2019, 2016, 2012 R2	✓		✓	✓		✓
Hyper-V Sockets	2019, 2016	✓		✓	✓		✓
PCI Passthrough/DDA	2019, 2016	✓	✓	✓	✓	✓	✓
Generation 2 virtual machines							
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12	✓ Note 12	✓ Note 12	✓ Note 12	✓ Note 12	✓ Note 12
Secure boot	2019, 2016, 2012 R2	✓	✓	✓	✓	✓	✓

Oracle Linux 6.x Series

This series only has 64-bit kernels.

Feature	Windows Server version	6.8-6.10 (RHCK)	6.8-6.10 (UEK4)
Availability		LIS 4.3	Built in
Core	2019, 2016, 2012 R2	✓	✓
Windows Server 2016 Accurate Time	2019, 2016		
Networking			
Jumbo frames	2019, 2016, 2012 R2	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓ Note 1	✓ Note 1
Live Migration	2019, 2016, 2012 R2	✓	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 2	✓
vRSS	2019, 2016, 2012 R2	✓	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓
SR-IOV	2019, 2016		
Storage			
VHDX resize	2019, 2016, 2012 R2	✓	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 5	✓ Note 5
TRIM support	2019, 2016, 2012 R2	✓	✓
SCSI WWN	2019, 2016, 2012 R2	✓	✓
Memory			
PAE Kernel Support	2019, 2016, 2012 R2	N/A	N/A
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 6, 8, 9	✓ Note 6, 8, 9
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 6, 8, 9	✓ Note 6, 8, 9

Feature	Windows Server version	6.8-6.10 (RHCK)	6.8-6.10 (UEK4)
Runtime Memory Resize	2019, 2016		
Video			
Hyper-V-specific video device	2019, 2016, 2012 R2	✓	✓
Miscellaneous			
Key-Value Pair	2019, 2016, 2012 R2	✓ Note 10,11	✓ Note 10,11
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓
File copy from host to guest	2019, 2016, 2012 R2	✓	✓
lsvmbus command	2019, 2016, 2012 R2	✓	✓
Hyper-V Sockets	2019, 2016	✓	✓
PCI Passthrough/DDA	2019, 2016	✓	✓
Generation 2 virtual machines			
Boot using UEFI	2019, 2016, 2012 R2	✓ Note 12	✓ Note 12
Secure boot	2019, 2016		

Notes

1. For this Oracle Linux release, VLAN tagging works but VLAN trunking does not.
2. Static IP injection may not work if Network Manager has been configured for a given synthetic network adapter on the virtual machine. For smooth functioning of static IP injection please make sure that either Network Manager is either turned off completely or has been turned off for a specific network adapter through its `ifcfg-ethX` file.
3. On Windows Server 2012 R2 while using virtual fibre channel devices, make sure that logical unit number 0 (LUN 0) has been populated. If LUN 0 has not been populated, a Linux virtual machine might not be able to mount fibre channel devices natively.
4. For built-in LIS, the "hyperv-daemons" package must be installed for this functionality.

5. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (fsck) on restore. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).

6. Dynamic memory support is only available on 64-bit virtual machines.

7. Hot-Add support is not enabled by default in this distribution. To enable Hot-Add support you need to add a udev rule under /etc/udev/rules.d/ as follows:

a. Create a file `/etc/udev/rules.d/100-balloon.rules`. You may use any other desired name for the file.

b. Add the following content to the file: `SUBSYSTEM=="memory", ACTION=="add", ATTR{state}="online"`

c. Reboot the system to enable Hot-Add support.

While the Linux Integration Services download creates this rule on installation, the rule is also removed when LIS is uninstalled, so the rule must be recreated if dynamic memory is needed after uninstallation.

8. Dynamic memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:

- Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.
- Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

9. If you are using Dynamic Memory on a Windows Server 2016 or Windows Server 2012 R2 operating system, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to hot-add failures, and you may not see any memory increase in a guest operating system.

10. To enable key/value pair (KVP) infrastructure, install the hypervkvpd or hyperv-daemons rpm package from your Oracle Linux ISO. Alternatively the package can be installed directly from Oracle Linux Yum repositories.

11. The key/value pair (KVP) infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.

12. On Windows Server 2012 R2 Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using Powershell:

Powershell

```
Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
```

The Linux Integration Services download can be applied to existing Generation 2 VMs but does not impart Generation 2 capability.

See Also

- [Set-VMFirmware](#)
- [Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V](#)
- [Supported Debian virtual machines on Hyper-V](#)
- [Supported SUSE virtual machines on Hyper-V](#)
- [Supported Ubuntu virtual machines on Hyper-V](#)
- [Supported FreeBSD virtual machines on Hyper-V](#)
- [Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V](#)
- [Best Practices for running Linux on Hyper-V](#)

Supported SUSE Linux Enterprise Server (SLES) virtual machines on Hyper-V

Article • 12/14/2022 • 5 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following is a feature distribution map that indicates the features in each version. The known issues and workarounds for each distribution are listed after the table.

The built-in SUSE Linux Enterprise Service drivers for Hyper-V are certified by SUSE. An example configuration can be viewed in this bulletin: [SUSE YES Certification Bulletin](#).

Table legend

- **Built in** - LIS are included as part of this Linux distribution. The Microsoft-provided LIS download package does not work for this distribution, so do not install it. The kernel module version numbers for the built in LIS (as shown by `lsmod`, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built in LIS is out of date.
 - ✓ - Feature available
 - (blank) - Feature not available

SLES12+ is 64-bit only.

Feature	Windows Server operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12 SP3-SP5	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
Jumbo frames	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓	✓	✓
VLAN tagging and trunking	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓	✓	✓
Live migration	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓	✓	✓
Static IP Injection	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1
vRSS	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓		
TCP Segmentation and Checksum Offloads	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓	✓	✓
SR-IOV	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓			
Storage								
VHDX resize	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓	✓	✓
Virtual Fibre Channel	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓	✓	✓	✓	✓	✓	✓
Live virtual machine backup	WS/Hyper-V 2022,2019,2016,2012,AS HCI	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8	✓ Note 2,3,8
TRIM support	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓	✓	✓	

Feature	Windows Server operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12 SP3-SP5	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
SCSI WWN	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓			
Memory								
PAE Kernel Support	WS/Hyper-V 2022,2019,2016,2012, AS HCI	N/A	N/A	N/A	N/A	N/A	✓	✓
Configuration of MMIO gap	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓	✓	✓	✓
Dynamic Memory - Hot-Add	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 4,5,6	✓ Note 4,5,6
Dynamic Memory - Ballooning	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 4,5,6	✓ Note 4,5,6
Runtime Memory Resize	WS/Hyper-V 2022,2019,2016	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6			
Video								
Hyper-V-specific video device	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓	✓	✓	✓
Miscellaneous								
Key/value pair	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓	✓	✓ Note 7	✓ Note 7
Non-Maskable Interrupt	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓	✓	✓	✓
File copy from host to guest	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓	✓	✓	✓

Feature	Windows Server operating system version	SLES 15 SP1-SP4	SLES 15	SLES 12	SLES 12 SP2	SLES 12 SP1	SLES 11 SP4	SLES 11 SP3
lsvmbus command	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓	✓	✓	✓			
Hyper-V Sockets	WS/Hyper-V 2022,2019,2016	✓	✓	✓				
PCI Passthrough/DDA	WS/Hyper-V 2022,2019,2016	✓	✓	✓	✓	✓		
Generation 2 virtual machines								
Boot using UEFI	WS/Hyper-V 2022,2019,2016,2012, AS HCI	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9	✓ Note 9
Secure boot	WS/Hyper-V 2022,2019,2016	✓	✓	✓	✓	✓		

Notes

1. Static IP injection may not work if **Network Manager** has been configured for a given Hyper-V-specific network adapter on the virtual machine. To ensure smooth functioning of static IP injection please ensure that Network Manager is turned off completely or has been turned off for a specific network adapter through its **ifcfg-ethX** file.
2. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (fsck) on restore.
3. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).
4. Dynamic memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:
 - Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.
 - Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.

5. Dynamic memory support is only available on 64-bit virtual machines.
6. If you are using Dynamic Memory on Windows Server 2016 or Windows Server 2012 operating systems, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to Hot-Add failures, and you may not see any memory increase in a guest operating system.
7. In Windows Server 2016 or Windows Server 2012 R2, the key/value pair infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.
8. VSS backup will fail if a single partition is mounted multiple times.
9. On Windows Server 2012 R2, Generation 2 virtual machines have secure boot enabled by default and Generation 2 Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in Hyper-V Manager or you can disable it using PowerShell:

```
Powershell
```

```
Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
```

See Also

- [Set-VMFirmware](#)
- [Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V](#)
- [Supported Debian virtual machines on Hyper-V](#)
- [Supported Oracle Linux virtual machines on Hyper-V](#)
- [Supported Ubuntu virtual machines on Hyper-V](#)
- [Supported FreeBSD virtual machines on Hyper-V](#)
- [Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V](#)
- [Best Practices for running Linux on Hyper-V](#)

Supported Ubuntu virtual machines on Hyper-V

Article • 02/02/2023 • 5 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features in each version. The known issues and workarounds for each distribution are listed after the table.

Table legend

- **Built in** - Linux Integration Services (LIS) is included as part of this Linux distribution. The Microsoft-provided LIS download package doesn't work for this distribution, so don't install it. The kernel module version numbers for the built in LIS (as shown by `lsmod`, for example) are different from the version number on the Microsoft-provided LIS download package. A mismatch doesn't indicate that the built in LIS is out of date.
- ✓ - Feature available
- (blank) - Feature not available

Feature	Windows Server operating system version	22.04 LTS	20.04 LTS	18.04 LTS	16.04 LTS
Availability		Built-in	Built-in	Built-in	Built-in
Core	2019, 2016, 2012 R2	✓	✓	✓	✓
Windows Server 2016 Accurate Time	2019, 2016	✓	✓	✓	✓
Networking					
Jumbo frames	2019, 2016, 2012 R2	✓	✓	✓	✓
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓	✓	✓
Live migration	2019, 2016, 2012 R2	✓	✓	✓	✓

Feature	Windows Server operating system version	22.04 LTS	20.04 LTS	18.04 LTS	16.04 LTS
Static IP Injection	2019, 2016, 2012 R2	✓ Note 1	✓ Note 1	✓ Note 1	✓ Note 1
vRSS	2019, 2016, 2012 R2	✓	✓	✓	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓	✓	✓
SR-IOV	2019, 2016	✓	✓	✓	✓
Storage					
VHDX resize	2019, 2016, 2012 R2	✓	✓	✓	✓
Virtual Fibre Channel	2019, 2016, 2012 R2	✓ Note 2	✓ Note 2	✓ Note 2	✓ Note 2
Live virtual machine backup	2019, 2016, 2012 R2	✓ Note 3, 4, 5			
TRIM support	2019, 2016, 2012 R2	✓	✓	✓	✓
SCSI WWN	2019, 2016, 2012 R2	✓	✓	✓	✓
Memory					
PAE Kernel Support	2019, 2016, 2012 R2	✓	✓	✓	✓
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2	✓ Note 6, 7, 8			
Dynamic Memory - Ballooning	2019, 2016, 2012 R2	✓ Note 6, 7, 8			
Runtime Memory Resize	2019, 2016	✓	✓	✓	✓
Video					
Hyper-V specific video device	2019, 2016, 2012 R2	✓	✓	✓	✓
Miscellaneous					
Key/value pair	2019, 2016, 2012 R2	✓ Note 5, 9			
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓	✓	✓

Feature	Windows Server operating system version	22.04 LTS	20.04 LTS	18.04 LTS	16.04 LTS
File copy from host to guest	2019, 2016, 2012 R2	✓	✓	✓	✓
lsvmbus command	2019, 2016, 2012 R2	✓	✓	✓	✓
Hyper-V Sockets	2019, 2016	✓	✓	✓	✓
PCI Passthrough/DDA	2019, 2016	✓	✓	✓	✓
Generation 2 virtual machines					
Boot using UEFI	2019, 2016, 2012 R2	✓	✓ Note 10, 11	✓ Note 10, 11	✓ Note 10, 11
Secure boot	2019, 2016	✓	✓	✓	✓

Notes

1. Static IP injection may not work if **Network Manager** has been configured for a given Hyper-V-specific network adapter on the virtual machine. To ensure smooth functioning of static IP injection, ensure that Network Manager is turned off completely or has been turned off for a specific network adapter through its **ifcfg-ethX** file.
2. While using virtual fiber channel devices, ensure that logical unit number 0 (LUN 0) has been populated. If LUN 0 has not been populated, a Linux virtual machine might not be able to mount fiber channel devices natively.
3. If there are open file handles during a live virtual machine backup operation, then in some corner cases, the backed-up VHDs might have to undergo a file system consistency check (**fsck**) on restore.
4. Live backup operations can fail silently if the virtual machine has an attached iSCSI device or direct-attached storage (also known as a pass-through disk).
5. On long term support (LTS) releases use latest virtual Hardware Enablement (HWE) kernel for up to date Linux Integration Services.

To install the Azure-tuned kernel on 16.04, 18.04 and 20.04, run the following commands as root (or sudo):

```
Bash
```

```
# apt-get update  
# apt-get install linux-azure
```

6. Dynamic memory support is only available on 64-bit virtual machines.
7. Dynamic Memory operations can fail if the guest operating system is running too low on memory. The following are some best practices:
 - Startup memory and minimal memory should be equal to or greater than the amount of memory that the distribution vendor recommends.
 - Applications that tend to consume the entire available memory on a system are limited to consuming up to 80 percent of available RAM.
8. If you are using Dynamic Memory on Windows Server 2019, Windows Server 2016 or Windows Server 2012/2012 R2 operating systems, specify **Startup memory**, **Minimum memory**, and **Maximum memory** parameters in multiples of 128 megabytes (MB). Failure to do so can lead to Hot-Add failures, and you might not see any memory increase on a guest operating system.
9. In Windows Server 2019, Windows Server 2016 or Windows Server 2012 R2, the key/value pair infrastructure might not function correctly without a Linux software update. Contact your distribution vendor to obtain the software update in case you see problems with this feature.
10. On Windows Server 2012 R2, Generation 2 virtual machines have secure boot enabled by default and some Linux virtual machines will not boot unless the secure boot option is disabled. You can disable secure boot in the **Firmware** section of the settings for the virtual machine in **Hyper-V Manager** or you can disable it using PowerShell:

Powershell

```
Set-VMFirmware -VMName "VMname" -EnableSecureBoot Off
```

11. Before attempting to copy the VHD of an existing Generation 2 VHD virtual machine to create new Generation 2 virtual machines, follow these steps:
 - a. Log in to the existing Generation 2 virtual machine.
 - b. Change directory to the boot EFI directory:

Bash

```
# cd /boot/efi/EFI
```

c. Copy the ubuntu directory in to a new directory named boot:

Bash

```
# sudo cp -r ubuntu/ boot
```

d. Change directory to the newly created boot directory:

Bash

```
# cd boot
```

e. Rename the shimx64.efi file:

Bash

```
# sudo mv shimx64.efi bootx64.efi
```

See Also

- [Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V](#)
- [Supported Debian virtual machines on Hyper-V](#)
- [Supported Oracle Linux virtual machines on Hyper-V](#)
- [Supported SUSE virtual machines on Hyper-V](#)
- [Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V](#)
- [Best Practices for running Linux on Hyper-V](#)
- [Set-VMFirmware](#)
- [Ubuntu 14.04 in a Generation 2 VM - Ben Armstrong's Virtualization Blog](#)

Supported FreeBSD virtual machines on Hyper-V

Article • 10/19/2022 • 3 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Hyper-V Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows 10, Windows 8.1

The following feature distribution map indicates the features in each version. The known issues and workarounds for each distribution are listed after the table.

Table legend

- **Built in** - BIS (FreeBSD Integration Service) are included as part of this FreeBSD release.
- **✓** - Feature available
- **(blank)** - Feature not available

Feature	Windows Server operating system version	13.0-13.1	12.0-12.3	11.2-11.4	10.4
Availability		Built in	Built in	Built in	Built in
Core	2019, 2016, 2012 R2	✓	✓	✓	✓
Windows Server 2016 Accurate Time	2019, 2016	✓	✓	✓	
Networking					
Jumbo frames	2019, 2016, 2012 R2	✓ Note 3	✓ Note 3	✓ Note 3	✓ Note 3
VLAN tagging and trunking	2019, 2016, 2012 R2	✓	✓	✓	✓
Live migration	2019, 2016, 2012 R2	✓	✓	✓	✓
Static IP Injection	2019, 2016, 2012 R2	✓ Note 4	✓ Note 4	✓ Note 4	✓ Note 4

Feature	Windows Server operating system version	13.0-13.1	12.0-12.3	11.2-11.4	10.4
vRSS	2019, 2016, 2012 R2	✓	✓	✓	✓
TCP Segmentation and Checksum Offloads	2019, 2016, 2012 R2	✓	✓	✓	✓
Large Receive Offload (LRO)	2019, 2016, 2012 R2	✓	✓	✓	✓
SR-IOV	2019, 2016	✓	✓	✓	✓
Storage		Note 1	Note1	Note 1	Note 1
VHDX resize	2019, 2016, 2012 R2	✓ Note 6	✓ Note 6	✓ Note 6	✓ Note 6
Virtual Fibre Channel	2019, 2016, 2012 R2				
Live virtual machine backup	2019, 2016, 2012 R2	✓	✓	✓	
TRIM support	2019, 2016, 2012 R2	✓	✓	✓	
SCSI WWN	2019, 2016, 2012 R2				
Memory					
PAE Kernel Support	2019, 2016, 2012 R2				
Configuration of MMIO gap	2019, 2016, 2012 R2	✓	✓	✓	✓
Dynamic Memory - Hot-Add	2019, 2016, 2012 R2				
Dynamic Memory - Ballooning	2019, 2016, 2012 R2				
Runtime Memory Resize	2019, 2016				
Video					
Hyper-V specific video device	2019, 2016, 2012 R2				
Miscellaneous					
Key/value pair	2019, 2016, 2012 R2	✓	✓	✓	✓
Non-Maskable Interrupt	2019, 2016, 2012 R2	✓	✓	✓	✓
File copy from host to guest	2019, 2016, 2012 R2				
lsvmbus command	2019, 2016, 2012 R2				

Feature	Windows Server operating system version	13.0-13.1	12.0-12.3	11.2-11.4	10.4
Hyper-V Sockets	2019, 2016				
PCI Passthrough/DDA	2019, 2016	✓	✓	✓	
Generation 2 virtual machines					
Boot using UEFI	2019, 2016, 2012 R2	✓	✓	✓	
Secure boot	2019, 2016				

Notes

1. Suggest to [Label Disk Devices](#) to avoid ROOT MOUNT ERROR during startup.
2. The virtual DVD drive may not be recognized when BIS drivers are loaded on FreeBSD 8.x and 9.x unless you enable the legacy ATA driver through the following command.


```
sh
# echo 'hw.ata.disk_enable=1' >> /boot/loader.conf
# shutdown -r now
```
3. 9126 is the maximum supported MTU size.
4. In a failover scenario, you cannot set a static IPv6 address in the replica server. Use an IPv4 address instead.
5. KVP is provided by ports on FreeBSD 10.0. See the [FreeBSD 10.0 ports](#) on FreeBSD.org for more information.
6. To make VHDX online resizing work properly in FreeBSD 11.0, a special manual step is required to work around a GEOM bug which is fixed in 11.0+, after the host resizes the VHDX disk - open the disk for write, and run "gpart recover" as the following.

```
sh
# dd if=/dev/da1 of=/dev/da1 count=0
# gpart recover da1
```

Additional Notes: The feature matrix of 10 stable and 11 stable is same with FreeBSD 11.1 release. In addition, FreeBSD 10.2 and previous versions (10.1, 10.0, 9.x, 8.x) are end of life. Please refer [here](#) for an up-to-date list of supported releases and the latest security advisories.

See Also

- Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V
- Best practices for running FreeBSD on Hyper-V

Feature Descriptions for Linux and FreeBSD virtual machines on Hyper-V

Article • 07/29/2021 • 8 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Azure Stack HCI, version 20H2; Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

This article describes features available in components such as core, networking, storage, and memory when using Linux and FreeBSD on a virtual machine.

Core

Feature	Description
Integrated shutdown	With this feature, an administrator can shut down virtual machines from the Hyper-V Manager. For more information, see Operating system shutdown .
Time synchronization	This feature ensures that the maintained time inside a virtual machine is kept synchronized with the maintained time on the host. For more information, see Time synchronization .
Windows Server 2016 Accurate Time	This feature allows the guest to use the Windows Server 2016 Accurate Time capability, which improves time synchronization with the host to 1ms accuracy. For more information, see Windows Server 2016 Accurate Time
Multiprocessing support	With this feature, a virtual machine can use multiple processors on the host by configuring multiple virtual CPUs.
Heartbeat	With this feature, the host can track the state of the virtual machine. For more information, see Heartbeat .
Integrated mouse support	With this feature, you can use a mouse on a virtual machine's desktop and also use the mouse seamlessly between the Windows Server desktop and the Hyper-V console for the virtual machine.
Hyper-V specific Storage device	This feature grants high-performance access to storage devices that are attached to a virtual machine.
Hyper-V specific Network device	This feature grants high-performance access to network adapters that are attached to a virtual machine.

Networking

Feature	Description
Jumbo frames	With this feature, an administrator can increase the size of network frames beyond 1500 bytes, which leads to a significant increase in network performance.
VLAN tagging and trunking	This feature allows you to configure virtual LAN (VLAN) traffic for virtual machines.
Live Migration	With this feature, you can migrate a virtual machine from one host to another host. For more information, see Virtual Machine Live Migration Overview .
Static IP Injection	With this feature, you can replicate the static IP address of a virtual machine after it has been failed over to its replica on a different host. Such IP replication ensures that network workloads continue to work seamlessly after a failover event.
vRSS (Virtual Receive Side Scaling)	Spreads the load from a virtual network adapter across multiple virtual processors in a virtual machine. For more information, see Virtual Receive-side Scaling in Windows Server 2012 R2 .
TCP Segmentation and Checksum Offloads	Transfers segmentation and checksum work from the guest CPU to the host virtual switch or network adapter during network data transfers.
Large Receive Offload (LRO)	Increases inbound throughput of high-bandwidth connections by aggregating multiple packets into a larger buffer, decreasing CPU overhead.
SR-IOV	Single Root I/O devices use DDA to allow guests access to portions of specific NIC cards allowing for reduced latency and increased throughput. SR-IOV requires up to date physical function (PF) drivers on the host and virtual function (VF) drivers on the guest.

Storage

Feature	Description
VHDX resize	With this feature, an administrator can resize a fixed-size .vhdx file that is attached to a virtual machine. For more information, see Online Virtual Hard Disk Resizing Overview .
Virtual Fibre Channel	With this feature, virtual machines can recognize a fiber channel device and mount it natively. For more information, see Hyper-V Virtual Fibre Channel Overview .

Feature	Description
Live virtual machine backup	This feature facilitates zero down time backup of live virtual machines. Note that the backup operation does not succeed if the virtual machine has virtual hard disks (VHDs) that are hosted on remote storage such as a Server Message Block (SMB) share or an iSCSI volume. Additionally, ensure that the backup target does not reside on the same volume as the volume that you back up.
TRIM support	TRIM hints notify the drive that certain sectors that were previously allocated are no longer required by the app and can be purged. This process is typically used when an app makes large space allocations via a file and then self-manages the allocations to the file, for example, to virtual hard disk files.
SCSI WWN	The storvsc driver extracts World Wide Name (WWN) information from the port and node of devices attached to the virtual machine and creates the appropriate sysfs files.

Memory

Feature	Description
PAE Kernel Support	Physical Address Extension (PAE) technology allows a 32-bit kernel to access a physical address space that is larger than 4GB. Older Linux distributions such as RHEL 5.x used to ship a separate kernel that was PAE enabled. Newer distributions such as RHEL 6.x have pre-built PAE support.
Configuration of MMIO gap	With this feature, appliance manufacturers can configure the location of the Memory Mapped I/O (MMIO) gap. The MMIO gap is typically used to divide the available physical memory between an appliance's Just Enough Operating Systems (JeOS) and the actual software infrastructure that powers the appliance.
Dynamic Memory - Hot-Add	<p>The host can dynamically increase or decrease the amount of memory available to a virtual machine while it is in operation. Before provisioning, the Administrator enables Dynamic Memory in the Virtual Machine Settings panel and specify the Startup Memory, Minimum Memory, and Maximum Memory for the virtual machine. When the virtual machine is in operation Dynamic Memory cannot be disabled and only the Minimum and Maximum settings can be changed. (It is a best practice to specify these memory sizes as multiples of 128MB.)</p> <p>When the virtual machine is first started available memory is equal to the Startup Memory. As Memory Demand increases due to application workloads Hyper-V may dynamically allocate more memory to the virtual machine via the Hot-Add mechanism, if supported by that version of the kernel. The maximum amount of memory allocated is capped by the value of the Maximum Memory parameter.</p> <p>The Memory tab of Hyper-V manager will display the amount of memory assigned to the virtual machine, but memory statistics within the virtual machine will show the highest amount of memory allocated.</p> <p>For more information, see Hyper-V Dynamic Memory Overview.</p>

Feature	Description
Dynamic Memory - Ballooning	<p>The host can dynamically increase or decrease the amount of memory available to a virtual machine while it is in operation. Before provisioning, the Administrator enables Dynamic Memory in the Virtual Machine Settings panel and specify the Startup Memory, Minimum Memory, and Maximum Memory for the virtual machine. When the virtual machine is in operation Dynamic Memory cannot be disabled and only the Minimum and Maximum settings can be changed. (It is a best practice to specify these memory sizes as multiples of 128MB.)</p> <p>When the virtual machine is first started available memory is equal to the Startup Memory. As Memory Demand increases due to application workloads Hyper-V may dynamically allocate more memory to the virtual machine via the Hot-Add mechanism (above). As Memory Demand decreases Hyper-V may automatically deprovision memory from the virtual machine via the Balloon mechanism. Hyper-V will not deprovision memory below the Minimum Memory parameter.</p> <p>The Memory tab of Hyper-V manager will display the amount of memory assigned to the virtual machine, but memory statistics within the virtual machine will show the highest amount of memory allocated.</p> <p>For more information, see Hyper-V Dynamic Memory Overview.</p>
Runtime Memory Resize	<p>An administrator can set the amount of memory available to a virtual machine while it is in operation, either increasing memory ("Hot Add") or decreasing it ("Hot Remove"). Memory is returned to Hyper-V via the balloon driver (see "Dynamic Memory - Ballooning"). The balloon driver maintains a minimum amount of free memory after ballooning, called the "floor", so assigned memory cannot be reduced below the current demand plus this floor amount. The Memory tab of Hyper-V manager will display the amount of memory assigned to the virtual machine, but memory statistics within the virtual machine will show the highest amount of memory allocated. (It is a best practice to specify memory values as multiples of 128MB.)</p>

Video

Feature	Description
Hyper-V-specific video device	This feature provides high-performance graphics and superior resolution for virtual machines. This device does not provide Enhanced Session Mode or RemoteFX capabilities.

Miscellaneous

Feature	Description

Feature	Description
KVP (Key-Value pair) exchange	This feature provides a key/value pair (KVP) exchange service for virtual machines. Typically administrators use the KVP mechanism to perform read and write custom data operations on a virtual machine. For more information, see Data Exchange: Using key-value pairs to share information between the host and guest on Hyper-V .
Non-Maskable Interrupt	With this feature, an administrator can issue Non-Maskable Interrupts (NMI) to a virtual machine. NMIs are useful in obtaining crash dumps of operating systems that have become unresponsive due to application bugs. These crash dumps can be diagnosed after you restart.
File copy from host to guest	This feature allows files to be copied from the host physical computer to the guest virtual machines without using the network adaptor. For more information, see Guest services .
lsvmbus command	This command gets information about devices on the Hyper-V virtual machine bus (VMBus) similar to information commands like lspci.
Hyper-V Sockets	This is an additional communication channel between the host and guest operating system. To load and use the Hyper-V Sockets kernel module, see Make your own integration services .
PCI Passthrough/DDA	<p>With Windows Server 2016 administrators can pass through PCI Express devices via the Discrete Device Assignment mechanism. Common devices are network cards, graphics cards, and special storage devices. The virtual machine will require the appropriate driver to use the exposed hardware. The hardware must be assigned to the virtual machine for it to be used.</p> <p>For more information, see Discrete Device Assignment - Description and Background.</p> <p>DDA is a pre-requisite for SR-IOV networking. Virtual ports will need to be assigned to the virtual machine and the virtual machine must use the correct Virtual Function (VF) drivers for device multiplexing.</p>

Generation 2 virtual machines

Feature	Description
Boot using UEFI	This feature allows virtual machines to boot using Unified Extensible Firmware Interface (UEFI). For more information, see Generation 2 Virtual Machine Overview .
Secure boot	This feature allows virtual machines to use UEFI based secure boot mode. When a virtual machine is booted in secure mode, various operating system components are verified using signatures present in the UEFI data store. For more information, see Secure Boot .

See Also

- Supported CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V
- Supported Debian virtual machines on Hyper-V
- Supported Oracle Linux virtual machines on Hyper-V
- Supported SUSE virtual machines on Hyper-V
- Supported Ubuntu virtual machines on Hyper-V
- Supported FreeBSD virtual machines on Hyper-V
- Best Practices for running Linux on Hyper-V
- Best practices for running FreeBSD on Hyper-V

Best Practices for running Linux on Hyper-V

Article • 07/11/2022 • 5 minutes to read

Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

This topic contains a list of recommendations for running Linux virtual machine on Hyper-V.

Tuning Linux File Systems on Dynamic VHDX Files

Some Linux file systems may consume significant amounts of real disk space even when the file system is mostly empty. To reduce the amount of real disk space usage of dynamic VHDX files, consider the following recommendations:

- When creating the VHDX, use 1MB BlockSizeBytes (from the default 32MB) in PowerShell, for example:

Powershell

```
PS > New-VHD -Path C:\MyVHDs\test.vhdx -SizeBytes 127GB -Dynamic -  
BlockSizeBytes 1MB
```

- The ext4 format is preferred to ext3 because ext4 is more space efficient than ext3 when used with dynamic VHDX files.
- When creating the filesystem specify the number of groups to be 4096, for example:

Bash

```
# mkfs.ext4 -G 4096 /dev/sdX1
```

Grub Menu Timeout on Generation 2 Virtual Machines

Because of legacy hardware being removed from emulation in Generation 2 virtual machines, the grub menu countdown timer counts down too quickly for the grub menu to be displayed, immediately loading the default entry. Until grub is fixed to use the EFI-supported timer, modify `/boot/grub/grub.conf`, `/etc/default/grub`, or equivalent to have "timeout=100000" instead of the default "timeout=5".

PxE Boot on Generation 2 Virtual Machines

Because the PIT timer is not present in Generation 2 Virtual Machines, network connections to the PxE TFTP server can be prematurely terminated and prevent the bootloader from reading Grub configuration and loading a kernel from the server.

On RHEL 6.x, the legacy grub v0.97 EFI bootloader can be used instead of grub2 as described here:

[https://access.redhat.com/documentation/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/s1-netboot-pxe-config-efi.html ↗](https://access.redhat.com/documentation/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/s1-netboot-pxe-config-efi.html)

On Linux distributions other than RHEL 6.x, similar steps can be followed to configure grub v0.97 to load Linux kernels from a PxE server.

Additionally, on RHEL/CentOS 6.6 keyboard and mouse input will not work with the pre-install kernel which prevents specifying installation options in the menu. A serial console must be configured to allow choosing installation options.

- In the `efidefault` file on the PxE server, add the following kernel parameter
`"console=ttyS1"`
- On the VM in Hyper-V, set up a COM port using this PowerShell cmdlet:

Powershell

```
Set-VMComPort -VMName <Name> -Number 2 -Path \\.\pipe\dbg1
```

Specifying a kickstart file to the pre-install kernel would also avoid the need for keyboard and mouse input during installation.

Use static MAC addresses with failover clustering

Linux virtual machines that will be deployed using failover clustering should be configured with a static media access control (MAC) address for each virtual network adapter. In some versions of Linux, the networking configuration may be lost after failover because a new MAC address is assigned to the virtual network adapter. To avoid losing the network configuration, ensure that each virtual network adapter has a static MAC address. You can configure the MAC address by editing the settings of the virtual machine in Hyper-V Manager or Failover Cluster Manager.

Use Hyper-V-specific network adapters, not the legacy network adapter

Configure and use the virtual Ethernet adapter, which is a Hyper-V-specific network card with enhanced performance. If both legacy and Hyper-V-specific network adapters are attached to a virtual machine, the network names in the output of `ifconfig -a` might show random values such as `_tmp12000801310`. To avoid this issue, remove all legacy network adapters when using Hyper-V-specific network adapters in a Linux virtual machine.

Use I/O scheduler noop/none for better disk I/O performance

The Linux kernel offers two sets of disk I/O schedulers to reorder requests. One set is for the older ‘blk’ subsystem and one set is for the newer ‘blk-mq’ subsystem. In either case, with today’s solid state disks it is recommended to use a scheduler that passes the scheduling decisions to the underlying Hyper-V hypervisor. For Linux kernels using the ‘blk’ subsystem, this is the “noop” scheduler. For Linux kernels using the ‘blk-mq’ subsystem, this is the “none” scheduler.

For a particular disk, the available schedulers can be seen at this file system location: `/sys/class/block/ <diskname>/queue/scheduler`, with the currently selected scheduler in square brackets. You can change the scheduler by writing to this file system location. The change must be added to an initialization script in order to persist across reboots. Consult your Linux distro documentation for details.

NUMA

Linux kernel versions earlier than 2.6.37 don't support NUMA on Hyper-V with larger VM sizes. This issue primarily impacts older distributions using the upstream Red Hat 2.6.32 kernel, and was fixed in Red Hat Enterprise Linux (RHEL) 6.6 (kernel-2.6.32-504). Systems running custom kernels older than 2.6.37, or RHEL-based kernels older than 2.6.32-504 must set the boot parameter `numa=off` on the kernel command line in grub.conf. For more information, see [Red Hat KB 436883](#).

Reserve more memory for kdump

In case the dump capture kernel ends up with a panic on boot, reserve more memory for the kernel. For example, change the parameter `crashkernel=384M:-128M` to `crashkernel=384M:-256M` in the Ubuntu grub configuration file.

Shrinking VHDX or expanding VHD and VHDX files can result in erroneous GPT partition tables

Hyper-V allows shrinking virtual disk (VHDX) files without regard for any partition, volume, or file system data structures that may exist on the disk. If the VHDX is shrunk to where the end of the VHDX comes before the end of a partition, data can be lost, that partition can become corrupted, or invalid data can be returned when the partition is read.

After resizing a VHD or VHDX, administrators should use a utility like fdisk or parted to update the partition, volume, and file system structures to reflect the change in the size of the disk. Shrinking or expanding the size of a VHD or VHDX that has a GUID Partition Table (GPT) will cause a warning when a partition management tool is used to check the partition layout, and the administrator will be warned to fix the first and secondary GPT headers. This manual step is safe to perform without data loss.

Additional References

- [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#)
- [Best practices for running FreeBSD on Hyper-V](#)
- [Deploy a Hyper-V Cluster](#)
- [Create Linux Images for Azure](#)

- Optimize your Linux VM on Azure

Best practices for running FreeBSD on Hyper-V

Article • 07/29/2021 • 2 minutes to read

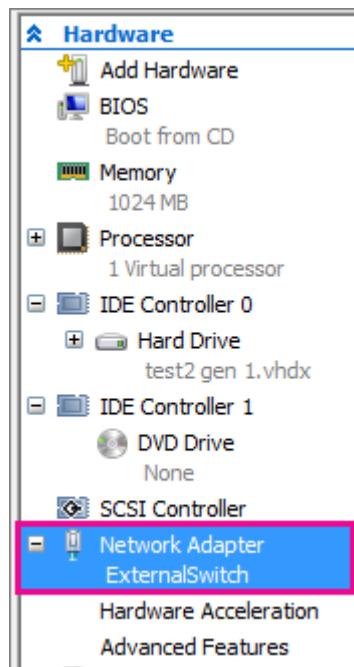
Applies to: Windows Server 2022, Azure Stack HCI, version 20H2; Windows Server 2019, Windows Server 2016, Hyper-V Server 2016, Windows Server 2012 R2, Hyper-V Server 2012 R2, Windows Server 2012, Hyper-V Server 2012, Windows Server 2008 R2, Windows 10, Windows 8.1, Windows 8, Windows 7.1, Windows 7

This topic contains a list of recommendations for running FreeBSD as a guest operating system on a Hyper-V virtual machine.

Enable CARP in FreeBSD 10.2 on Hyper-V

The Common Address Redundancy Protocol (CARP) allows multiple hosts to share the same IP address and Virtual Host ID (VHID) to help provide high availability for one or more services. If one or more hosts fail, the other hosts transparently take over so users won't notice a service failure. To use CARP in FreeBSD 10.2, follow the instructions in the [FreeBSD handbook](#) and do the following in Hyper-V Manager.

- Verify the virtual machine has a Network Adapter and it's assigned a virtual switch. Select the virtual machine and select **Actions > Settings**.



- Enable MAC address spoofing. To do this,

1. Select the virtual machine and select **Actions > Settings**.
2. Expand **Network Adapter** and select **Advanced Features**.
3. Select **Enable MAC Address spoofing**.

Create labels for disk devices

During startup, device nodes are created as new devices are discovered. This can mean that device names can change when new devices are added. If you get a ROOT MOUNT ERROR during startup, you should create labels for each IDE partition to avoid conflicts and changes. To learn how, see [Labeling Disk Devices](#). Below are examples.

 **Important**

Make a backup copy of your fstab before making any changes.

1. Reboot the system into single user mode. This can be accomplished by selecting boot menu option 2 for FreeBSD 10.3+ (option 4 for FreeBSD 8.x), or performing a 'boot -s' from the boot prompt.
2. In Single user mode, create GEOM labels for each of the IDE disk partitions listed in your fstab (both root and swap). Below is an example of FreeBSD 10.3.

Bash

```
# cat /etc/fstab
# Device          Mountpoint      FStype  Options   Dump   Pass#
/dev/da0p2        /               ufs     rw        1       1
/dev/da0p3        none            swap    sw        0       0

# glabel label rootfs  /dev/da0p2
# glabel label swap   /dev/da0p3
# exit
```

Additional information on GEOM labels can be found at: [Labeling Disk Devices](#).

3. The system will continue with multi-user boot. After the boot completes, edit /etc/fstab and replace the conventional device names, with their respective labels. The final /etc/fstab will look like this:

# Device	Mountpoint	FStype	Options	Dump
Pass#				

/dev/label/rootfs	/	ufs	rw	1
1				
/dev/label/swap	none	swap	sw	0

4. The system can now be rebooted. If everything went well, it will come up normally and mount will show:

```
# mount  
/dev/label/rootfs on / (ufs, local, journaled soft-updates)  
devfs on /dev (devfs, local, mutilabel)
```

Use a wireless network adapter as the virtual switch

If the virtual switch on the host is based on wireless network adapter, reduce the ARP expiration time to 60 seconds by the following command. Otherwise the networking of the VM may stop working after a while.

```
# sysctl net.link.ether.inet.max_age=60
```

See also

- [Supported FreeBSD virtual machines on Hyper-V](#)

Hyper-V feature compatibility by generation and guest

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

The tables in this article show you the generations and operating systems that are compatible with some of the Hyper-V features, grouped by categories. In general, you'll get the best availability of features with a generation 2 virtual machine that runs the newest operating system.

Keep in mind that some features rely on hardware or other infrastructure. For hardware details, see [System requirements for Hyper-V on Windows Server 2016](#). In some cases, a feature can be used with any supported guest operating system. For details on which operating systems are supported, see:

- [Supported Linux and FreeBSD virtual machines](#)
- [Supported Windows guest operating systems](#)

Availability and backup

Feature	Generation	Guest operating system
Checkpoints	1 and 2	Any supported guest
Guest clustering	1 and 2	Guests that run cluster-aware applications and have iSCSI target software installed
Replication	1 and 2	Any supported guest
Domain Controller	1 and 2	Any supported Windows Server guest using only production checkpoints. See Supported Windows Server guest operating systems

Compute

Feature	Generation	Guest operating system
Dynamic memory	1 and 2	Specific versions of supported guests. See Hyper-V Dynamic Memory Overview for versions older than Windows Server 2016 and Windows 10.

Feature	Generation	Guest operating system
Hot add/removal of memory	1 and 2	Windows Server 2016, Windows 10
Virtual NUMA	1 and 2	Any supported guest

Development and test

Feature	Generation	Guest operating system
COM/Serial ports	1 and 2 Note: For generation 2, use Windows PowerShell to configure. For details, see Add a COM port for kernel debugging .	Any supported guest

Mobility

Feature	Generation	Guest operating system
Live migration	1 and 2	Any supported guest
Import/export	1 and 2	Any supported guest

Networking

Feature	Generation	Guest operating system
Hot add/removal of virtual network adapter	2	Any supported guest
Legacy virtual network adapter	1	Any supported guest
Single root input/output virtualization (SR-IOV)	1 and 2	64-bit Windows guests, starting with Windows Server 2012 and Windows 8.
Virtual machine multi queue (VMMQ)	1 and 2	Any supported guest

Remote connection experience

Feature	Generation	Guest operating system
Discrete device assignment (DDA)	1 and 2	Windows Server 2016, Windows Server 2012 R2 only with Update 3133690 installed, Windows 10 Note: For details on Update 3133690, see this support article .
Enhanced session mode	1 and 2	Windows Server 2016, Windows Server 2012 R2, Windows 10, and Windows 8.1, with Remote Desktop Services enabled Note: You might need to also configure the host. For details, see Use local resources on Hyper-V virtual machine with VMConnect .
RemoteFx	1 and 2	Generation 1 on 32-bit and 64-bit Windows versions starting with Windows 8. Generation 2 on 64-bit Windows 10 versions

Security

Feature	Generation	Guest operating system
Secure boot	2	Linux: Ubuntu 14.04 and later, SUSE Linux Enterprise Server 12 and later, Red Hat Enterprise Linux 7.0 and later, and CentOS 7.0 and later Windows: All supported versions that can run on a generation 2 virtual machine
Shielded virtual machines	2	Windows: All supported versions that can run on a generation 2 virtual machine

Storage

Feature	Generation	Guest operating system
Shared virtual hard disks (VHDX only)	1 and 2	Windows Server 2016, Windows Server 2012 R2, Windows Server 2012
SMB3	1 and 2	All that support SMB3
Storage spaces direct	2	Windows Server 2016
Virtual Fibre Channel	1 and 2	Windows Server 2016, Windows Server 2012 R2, Windows Server 2012
VHDX format	1 and 2	Any supported guest

Get started with Hyper-V on Windows Server

Article • 08/16/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Use the following resources to set up and try out Hyper-V on the Server Core or GUI installation option of Windows Server 2019 or Windows Server 2016. But before you install anything, check the [System Requirements for Windows Server](#) and the [System Requirements for Hyper-V](#).

- [Download and install Windows Server](#)
- [Install the Hyper-V role on Windows Server](#)
- [Create a virtual switch for Hyper-V virtual machines](#)
- [Create a virtual machine in Hyper-V](#)

Install the Hyper-V role on Windows Server

Article • 03/30/2022 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Windows Server 2019

To create and run virtual machines, install the Hyper-V role on Windows Server by using Server Manager or the **Install-WindowsFeature** cmdlet in Windows PowerShell. For Windows 10, see [Install Hyper-V on Windows 10](#).

To learn more about Hyper-V, see the [Hyper-V Technology Overview](#). To try out Windows Server 2019, you can download and install an evaluation copy. See the [Evaluation Center](#).

Before you install Windows Server or add the Hyper-V role, make sure that:

- Your computer hardware is compatible. For details, see [System Requirements for Windows Server](#) and [System requirements for Hyper-V on Windows Server](#).
- You don't plan to use third-party virtualization apps that rely on the same processor features that Hyper-V requires. Examples include VMWare Workstation and VirtualBox. You can install Hyper-V without uninstalling these other apps. But, if you try to use them to manage virtual machines when the Hyper-V hypervisor is running, the virtual machines might not start or might run unreliably. For details and instructions for turning off the Hyper-V hypervisor if you need to use one of these apps, see [Virtualization applications do not work together with Hyper-V, Device Guard, and Credential Guard](#).

If you want to install only the management tools, such as Hyper-V Manager, see [Remotely manage Hyper-V hosts with Hyper-V Manager](#).

Install Hyper-V by using Server Manager

1. In **Server Manager**, on the **Manage** menu, click **Add Roles and Features**.
2. On the **Before you begin** page, verify that your destination server and network environment are prepared for the role and feature you want to install. Click **Next**.
3. On the **Select installation type** page, select **Role-based or feature-based installation** and then click **Next**.

4. On the **Select destination server** page, select a server from the server pool and then click **Next**.
5. On the **Select server roles** page, select **Hyper-V**.
6. To add the tools that you use to create and manage virtual machines, click **Add Features**. On the **Features** page, click **Next**.
7. On the **Create Virtual Switches** page, **Virtual Machine Migration** page, and **Default Stores** page, select the appropriate options.
8. On the **Confirm installation selections** page, select **Restart the destination server automatically if required**, and then click **Install**.
9. When installation is finished, verify that Hyper-V installed correctly. Open the **All Servers** page in Server Manager and select a server on which you installed Hyper-V. Check the **Roles and Features** tile on the page for the selected server.

Install Hyper-V by using the Install-WindowsFeature cmdlet

1. On the Windows desktop, click the Start button and type any part of the name **Windows PowerShell**.
2. Right-click **Windows PowerShell** and select **Run as Administrator**.
3. To install Hyper-V on a server you're connected to remotely, run the following command and replace `<computer_name>` with the name of server.

```
PowerShell  
  
Install-WindowsFeature -Name Hyper-V -ComputerName <computer_name> -  
IncludeManagementTools -Restart
```

If you're connected locally to the server, run the command without `-ComputerName <computer_name>`.

4. After the server restarts, you can see that the Hyper-V role is installed and see what other roles and features are installed by running the following command:

```
PowerShell  
  
Get-WindowsFeature -ComputerName <computer_name>
```

If you're connected locally to the server, run the command without `-ComputerName <computer_name>`.

Note

If you install this role on a server that runs the Server Core installation option of Windows Server 2016 and use the parameter `-IncludeManagementTools`, only the Hyper-V Module for Windows PowerShell is installed. You can use the GUI management tool, Hyper-V Manager, on another computer to remotely manage a Hyper-V host that runs on a Server Core installation. For instructions on connecting remotely, see [Remotely manage Hyper-V hosts with Hyper-V Manager](#).

Additional References

- [Install-WindowsFeature](#)

Create and configure a virtual switch with Hyper-V

Article • 12/07/2022 • 4 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to create and configure your virtual switch using Hyper-V Manager or PowerShell. A virtual switch allows virtual machines created on Hyper-V hosts to communicate with other computers. When you first install the Hyper-V role on Windows Server, you can optionally create a virtual switch at the same time. To learn more about virtual switches, see [Hyper-V Virtual Switch](#).

For more information about how you can set up your networking infrastructure with Windows Server, review the [Networking](#) documentation.

Prerequisites

Before you can create and configure your virtual switch, your computer must meet the following prerequisites:

- The Hyper-V server role must be installed.
- Determine [what type of virtual switch you need to create](#).
- Identify which network you'll connect your computer to. Review the [Core network planning](#) article for more information.
- Have administrative rights.

Create a virtual switch

Once you've completed the prerequisites, you'll be ready to create your virtual switch. In this section, we'll create a basic virtual switch by following these steps.

Hyper-V Manager

Here's how to create a virtual switch using Hyper-V Manager.

1. Open Hyper-V Manager.
2. From the **Actions** pane, select **Virtual Switch Manager**.

3. Choose the type of virtual switch, then select **Create Virtual Switch**.
4. Enter a name for the virtual switch, then perform one of the following steps.
 - a. If you selected *external*, choose the network adapter (NIC) that you want to use, then select **OK**.

You'll be prompted with a warning that the change may disrupt your network connectivity; select **Yes** if you're happy to continue.
 - b. If you selected *internal* or *private*, select **OK**.

Management Operating System sharing

An external virtual switch allows your virtual machines to connect to an external network. You can also allow the management operating system to share the same selected network adapter. To begin, follow these steps.

Hyper-V Manager

Here's how to allow the management operating system to share the selected network adapter switch using Hyper-V Manager.

1. Open Hyper-V Manager.
2. From the **Actions** pane, select **Virtual Switch Manager**.
3. Select the virtual switch you wish to configure, check the **Allow management operating system to share this network adapter** and select **OK**.

You'll be prompted with a warning that the change may disrupt your network connectivity; select **Yes** if you're happy to continue.

Virtual LAN (VLAN) identification

You can specify the VLAN identification (ID) used by virtual machines network adapters and virtual switches. For virtual switches connected to either an external or internal network you can specify the (VLAN) ID. The VLAN ID number is used by the management operating system and virtual machines communicating through this virtual switch.

You can also configure your virtual switch with other VLAN options, such port mode and the native VLAN ID. For these options, you'll need to use PowerShell and ensure the configuration is compatible with your networks configuration.

To configure the VLAN identification for the switch, follow these steps.

Hyper-V Manager

Here's how to specify the VLAN ID using the Hyper-V Manager.

1. Open Hyper-V Manager.
2. From the **Actions** pane, select **Virtual Switch Manager**.
3. Select the virtual switch you wish to configure, check the **Enable virtual LAN identification for management operating system**.
 - a. You can enter any VLAN ID number or leave the default, then select **OK**.

You'll be prompted to warn you that the change may disrupt your network connectivity, select **Yes** if you're happy to continue.

VLAN identifiers should be consistent with your network to ensure compatibility between your computer, virtual machines, and other network devices.

Next step

Now you've created and configured your virtual switch, here are other articles to help you continue with Hyper-V.

- Learn about [Switch Embedded Teaming \(SET\)](#).
- Learn how to [create a virtual machine in Hyper-V](#).
- Learn about other configuration options in the [Set-VMSwitch](#) and [Set-VMNetworkAdapterVlan](#) PowerShell reference articles.

Create a virtual machine in Hyper-V

Article • 02/22/2023 • 4 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019, Windows 11, Windows 10

Learn how to create a virtual machine by using Hyper-V Manager and Windows PowerShell and what options you have when you create a virtual machine in Hyper-V Manager.

Create a virtual machine

Hyper-V Manager

1. Open Hyper-V Manager.
2. From the **Action** pane, click **New**, and then click **Virtual Machine**.
3. From the **New Virtual Machine Wizard**, click **Next**.
4. Make the appropriate choices for your virtual machine on each of the pages.
For more information, see [New virtual machine options and defaults in Hyper-V Manager](#).
5. After verifying your choices in the **Summary** page, click **Finish**.
6. In Hyper-V Manager, right-click the virtual machine and select **connect**.
7. In the Virtual Machine Connection window, select **Action > Start**.

Options in Hyper-V Manager New Virtual Machine Wizard

The following table lists the options you can pick when you create a virtual machine in Hyper-V Manager and the defaults for each.

Page	Default for Windows Server 2016, Windows 10, and later	Other options
Specify Name and Location	<p>Name: New Virtual Machine.</p> <p>Location: C:\ProgramData\Microsoft\Windows\Hyper-V\.</p>	<p>You can also enter your own name and choose another location for the virtual machine.</p> <p>This is where the virtual machine configuration files will be stored.</p>
Specify Generation	Generation 1	<p>You can also choose to create a Generation 2 virtual machine. For more information, see Should I create a generation 1 or 2 virtual machine in Hyper-V?</p>
Assign Memory	<p>Startup memory: 1024 MB</p> <p>Dynamic memory: not selected</p>	<p>You can set the startup memory from 32 MB to 5902 MB.</p> <p>You can also choose to use Dynamic Memory. For more information, see Hyper-V Dynamic Memory Overview.</p>
Configure Networking	Not connected	<p>You can select a network connection for the virtual machine to use from a list of existing virtual switches. See Create a virtual switch for Hyper-V virtual machines.</p>
Connect Virtual Hard Disk	<p>Create a virtual hard disk</p> <p>Name: <vmname>.vhdx</p> <p>Location: C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\</p> <p>Size: 127 GB</p>	<p>You can also choose to use an existing virtual hard disk or wait and attach a virtual hard disk later.</p>
Installation Options	Install an operating system later	<p>These options change the boot order of the virtual machine so that you can install from an .iso file, bootable floppy disk or a network installation service, like Windows Deployment Services (WDS).</p>

Page	Default for Windows Server 2016, Windows 10, and later	Other options
Summary	<p>Displays the options that you have chosen, so that you can verify they are correct.</p> <ul style="list-style-type: none"> - Name - Generation - Memory - Network - Hard Disk - Operating System 	<p>Tip: You can copy the summary from the page and paste it into e-mail or somewhere else to help you keep track of your virtual machines.</p>

Additional References

- [New-VM](#)
- [Supported virtual machine configuration versions](#)
- [Should I create a generation 1 or 2 virtual machine in Hyper-V?](#)
- [Create a virtual switch for Hyper-V virtual machines](#)

Plan for Hyper-V on Windows Server

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Windows Server 2019

Use these resources to help you plan your Hyper-V deployment.

- [Hyper-V scalability in Windows Server](#)
- [Hyper-V security in Windows Server](#)
- [Networking in Windows Server](#)
- [Should I create a generation 1 or 2 virtual machine?](#)
- [Plan for GPU acceleration in Windows Server](#)
- [Plan for deploying devices using Discrete Device Assignment](#)

Guest operating system and application supportability on Hyper-V

Article • 11/24/2022 • 2 minutes to read

Hyper-V is a hypervisor that is broadly used in many Microsoft server products, including the Windows Server family (Datacenter, Standard, and Essentials editions) and Azure Stack HCI. Hyper-V provides a platform with broad ecosystem support and compatibility. This article clarifies which versions of Windows Server or Azure Stack HCI map to which Hyper-V build numbers. This helps understand supported scenarios where a guest operating system or application has been validated for Hyper-V.

Although the different products that include Hyper-V could contain variations in features, the common codebase provides a consistent platform for guest operating systems and applications running inside of a virtual machine to run on compatible products that share the same Hyper-V build number. This means that any support or compatibility statements for a guest operating system or application that is certified for specific builds of Hyper-V is compatible with all products that share the same build number for Hyper-V.

The table below shows which Hyper-V build numbers are available in which compatible products:

Hyper-V build	Compatible products
20348	Windows Server 2022 Datacenter Windows Server 2022 Standard Windows Server 2022 Essentials Azure Stack HCI version 21H2 Azure Stack HCI version 22H2
17763 and 17784	Windows Server 2019 Datacenter Windows Server 2019 Standard Windows Server 2019 Essentials Hyper-V Server 2019 Azure Stack HCI version 20H2
14393	Windows Server 2016 Datacenter Windows Server 2016 Standard Windows Server 2016 Essentials Hyper-V Server 2016

For more information, see:

- [Windows Server release information](#)

- Supported Windows guest operating systems on Hyper-V
- Supported Linux and FreeBSD guest operating systems on Hyper-V

Should I create a generation 1 or 2 virtual machine in Hyper-V?

Article • 10/18/2022 • 10 minutes to read

Applies to: Windows 10, Windows 11, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019, Windows Server 2022, Azure Stack HCI

ⓘ Note

If you plan to ever upload Windows virtual machines (VMs) from on-premises to Microsoft Azure, both generation 1 and generation 2 VMs are supported as long as they use the VHD file format and have a fixed (not dynamically expanding) sized disk. See [Generation 2 VMs on Azure](#) to learn more about generation 2 capabilities supported on Azure. For more information on uploading a Windows VHD or VHDX, see [Prepare a Windows VHD or VHDX to upload to Azure](#).

Your choice to create a generation 1 or generation 2 virtual machine depends on which guest operating system you want to install and the boot method you want to use to deploy the virtual machine. We recommend that you create a generation 2 virtual machine to take advantage of features like Secure Boot unless one of the following statements is true:

- You're using an existing, pre-built virtual disk (VHD or VHDX) which is not [compatible with UEFI](#).
- Generation 2 doesn't support the operating system you want to run on the virtual machine.
- Generation 2 doesn't support the boot method you want to use.

For more information about what features are available with generation 2 virtual machines, see [Hyper-V feature compatibility by generation and guest](#).

You can't change a virtual machine's generation after you've created it. So, we recommend that you review the considerations here, as well as choose the operating system, boot method, and features you want to use before you choose a generation.

Which guest operating systems are supported?

Generation 1 virtual machines support most guest operating systems. Generation 2 virtual machines support most 64-bit versions of Windows and more current versions of Linux and FreeBSD operating systems. Use the following sections to see which generation of virtual machine supports the guest operating system you want to install.

- [Windows guest operating system support](#)
- [CentOS and Red Hat Enterprise Linux guest operating system support](#)
- [Debian guest operating system support](#)
- [FreeBSD guest operating system support](#)
- [Oracle Linux guest operating system support](#)
- [SUSE guest operating system support](#)
- [Ubuntu guest operating system support](#)

Windows guest operating system support

The following table shows which 64-bit versions of Windows you can use as a guest operating system for generation 1 and generation 2 virtual machines.

64-bit versions of Windows	Generation 1	Generation 2
Windows Server 2022	✓	✓
Windows Server 2019	✓	✓
Windows Server 2016	✓	✓
Windows Server 2012 R2	✓	✓
Windows Server 2012	✓	✓
Windows Server 2008 R2	✓	✗
Windows Server 2008	✓	✗
Windows 11	✗	✓
Windows 10	✓	✓
Windows 8.1	✓	✓
Windows 8	✓	✓
Windows 7	✓	✗

The following table shows which 32-bit versions of Windows you can use as a guest operating system for generation 1 and generation 2 virtual machines.

32-bit versions of Windows	Generation 1	Generation 2
Windows 10	✓	✗
Windows 8.1	✓	✗
Windows 8	✓	✗
Windows 7	✓	✗

CentOS and Red Hat Enterprise Linux guest operating system support

The following table shows which versions of Red Hat Enterprise Linux (RHEL) and CentOS you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Operating system versions	Generation 1	Generation 2
RHEL/CentOS 8.x series	✓	✓
RHEL/CentOS 7.x series	✓	✓
RHEL/CentOS 6.x series	✓	✓ <small>Note: Only supported on Windows Server 2016 and above.</small>
RHEL/CentOS 5.x series	✓	✗

For more information, see [CentOS and Red Hat Enterprise Linux virtual machines on Hyper-V](#).

Debian guest operating system support

The following table shows which versions of Debian you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Operating system versions	Generation 1	Generation 2
Debian 10.x (buster) series	✓	✓
Debian 9.x (stretch) series	✓	✓

Operating system versions	Generation 1	Generation 2
Debian 8.x (jessie) series	✓	✓
Debian 7.x (wheezy) series	✓	✗

For more information, see [Debian virtual machines on Hyper-V](#).

FreeBSD guest operating system support

The following table shows which versions of FreeBSD you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Operating system versions	Generation 1	Generation 2
FreeBSD 12 to 12.1	✓	✓
FreeBSD 11.1 to 11.3	✓	✓
FreeBSD 11	✓	✗
FreeBSD 10 to 10.3	✓	✗
FreeBSD 9.1 and 9.3	✓	✗
FreeBSD 8.4	✓	✗

For more information, see [FreeBSD virtual machines on Hyper-V](#).

Oracle Linux guest operating system support

The following table shows which versions of Red Hat Compatible Kernel Series you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Red Hat Compatible Kernel Series versions	Generation 1	Generation 2
Oracle Linux 8.x series	✓	✓
Oracle Linux 7.x series	✓	✓
Oracle Linux 6.x series	✓	✗

The following table shows which versions of Unbreakable Enterprise Kernel you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Unbreakable Enterprise Kernel (UEK) versions	Generation 1	Generation 2

Unbreakable Enterprise Kernel (UEK) versions	Generation 1	Generation 2
Oracle Linux UEK R3 QU3	✓	✗
Oracle Linux UEK R3 QU2	✓	✗
Oracle Linux UEK R3 QU1	✓	✗

For more information, see [Oracle Linux virtual machines on Hyper-V](#).

SUSE guest operating system support

The following table shows which versions of SUSE you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Operating system versions	Generation 1	Generation 2
SUSE Linux Enterprise Server 15 series	✓	✓
SUSE Linux Enterprise Server 12 series	✓	✓
SUSE Linux Enterprise Server 11 series	✓	✗
Open SUSE 12.3	✓	✗

For more information, see [SUSE virtual machines on Hyper-V](#).

Ubuntu guest operating system support

The following table shows which versions of Ubuntu you can use as a guest operating system for generation 1 and generation 2 virtual machines.

Operating system versions	Generation 1	Generation 2
Ubuntu 20.04	✓	✓
Ubuntu 18.04	✓	✓
Ubuntu 16.04	✓	✓
Ubuntu 14.04	✓	✓
Ubuntu 12.04	✓	✗

For more information, see [Ubuntu virtual machines on Hyper-V](#).

How can I boot the virtual machine?

The following table shows which boot methods are supported by generation 1 and generation 2 virtual machines.

Boot method	Generation 1	Generation 2
PXE boot by using a standard network adapter	✗	✓
PXE boot by using a legacy network adapter	✓	✗
Boot from a SCSI virtual hard disk (.VHDX) or virtual DVD (.ISO)	✗	✓
Boot from IDE Controller virtual hard disk (.VHD), virtual DVD (.ISO) or a physical CD/DVD drive	✓	✗
Boot from virtual floppy (.VFD)	✓	✗

What are the advantages of using generation 2 virtual machines?

Here are some of the advantages you get when you use a generation 2 virtual machine:

- **Secure Boot**

This is a feature that verifies the boot loader is signed by a trusted authority in the UEFI database to help prevent unauthorized firmware, operating systems, or UEFI drivers from running at boot time. Secure Boot is enabled by default for generation 2 virtual machines. If you need to run a guest operating system that's not supported by Secure Boot, you can disable it after the virtual machine's created. For more information, see [Secure Boot](#).

To Secure Boot generation 2 Linux virtual machines, you need to choose the UEFI CA Secure Boot template when you create the virtual machine.

- **Larger boot volume** The maximum boot volume for generation 2 virtual machines is 64 TB. This is the maximum disk size supported by a .VHDX. For generation 1 virtual machines, the maximum boot volume is 2TB for a .VHDX and 2040GB for a .VHD. For more information, see [Hyper-V Virtual Hard Disk Format Overview](#).

You may also see a slight improvement in virtual machine boot and installation times with generation 2 virtual machines.

What's the difference in device support?

The following table compares the devices available between generation 1 and generation 2 virtual machines.

Generation 1 Device	Generation 2 Replacement	Generation 2 Enhancements
IDE controller	Virtual SCSI controller	Boot from .VHDX (64 TB maximum size, and online resize capability)
IDE CD-ROM	Virtual SCSI CD-ROM	Support for up to 64 SCSI DVD devices per SCSI controller.
Legacy BIOS	UEFI firmware	Secure Boot
Legacy network adapter	Synthetic network adapter	Network boot with IPv4 and IPv6
Floppy controller and DMA controller	No floppy controller support	N/A
Universal asynchronous receiver/transmitter (UART) for COM ports	Optional UART for debugging	Faster and more reliable
i8042 keyboard controller	Software-based input	Uses fewer resources because there is no emulation. Also reduces the attack surface from the guest operating system.
PS/2 keyboard	Software-based keyboard	Uses fewer resources because there is no emulation. Also reduces the attack surface from the guest operating system.
PS/2 mouse	Software-based mouse	Uses fewer resources because there is no emulation. Also reduces the attack surface from the guest operating system.
S3 video	Software-based video	Uses fewer resources because there is no emulation. Also reduces the attack surface from the guest operating system.
PCI bus	No longer required	N/A
Programmable interrupt controller (PIC)	No longer required	N/A

Generation 1 Device	Generation 2 Replacement	Generation 2 Enhancements
Programmable interval timer (PIT)	No longer required	N/A
Super I/O device	No longer required	N/A

More about generation 2 virtual machines

Here are some additional tips about using generation 2 virtual machines.

Attach or add a DVD drive

- You can't attach a physical CD or DVD drive to a generation 2 virtual machine. The virtual DVD drive in generation 2 virtual machines only supports ISO image files. To create an ISO image file of a Windows environment, you can use the `OScdimg` command line tool. For more information, see [Oscdimg Command-Line Options](#).
- When you create a new virtual machine with the `New-VM` Windows PowerShell cmdlet, the generation 2 virtual machine doesn't have a DVD drive. You can add a DVD drive while the virtual machine is running.

Use UEFI firmware

- Secure Boot or UEFI firmware isn't required on the physical Hyper-V host. Hyper-V provides virtual firmware to virtual machines that is independent of what's on the Hyper-V host.
- UEFI firmware in a generation 2 virtual machine doesn't support setup mode for Secure Boot.
- We don't support running a UEFI shell or other UEFI applications in a generation 2 virtual machine. Using a non-Microsoft UEFI shell or UEFI applications is technically possible if they are compiled directly from the sources. If these applications are not appropriately digitally signed, you must disable Secure Boot for the virtual machine.

Work with VHDX files

- You can resize a VHDX file that contains the boot volume for a generation 2 virtual machine while the virtual machine is running.

- We don't support or recommend that you create a single virtual disk (VHD or VHDX file) that is bootable to both generation 1 and generation 2 virtual machines.
- The virtual machine generation is a property of the virtual machine, not a property of the virtual hard disk. So you can't tell if a VHDX file was created by a generation 1 or a generation 2 virtual machine.
- A VHDX file created with a generation 2 virtual machine can be attached to the IDE controller or the SCSI controller of a generation 1 virtual machine. However, if this is a bootable VHDX file, the generation 1 virtual machine won't boot.

Use IPv6 instead of IPv4

When booting from network with PXE, generation 2 virtual machines use IPv4 by default. To use IPv6 instead, run the [Set-VMFirmware](#) Windows PowerShell cmdlet. For example, the following command sets the preferred protocol to IPv6 for a virtual machine named TestVM:

```
PowerShell
```

```
Set-VMFirmware -VMName 'TestVM' -IPProtocolPreference IPv6
```

Add a COM port for kernel debugging

COM ports aren't available in generation 2 virtual machines until you add them. You can do this with Windows PowerShell or Windows Management Instrumentation (WMI). These steps show you how to do it with Windows PowerShell.

To add a COM port:

1. Disable Secure Boot. Kernel debugging isn't compatible with Secure Boot. Make sure the virtual machine is in an Off state, then use the [Set-VMFirmware](#) cmdlet. For example, the following command disables Secure Boot on virtual machine TestVM:

```
PowerShell
```

```
Set-VMFirmware -VMName 'TestVM' -EnableSecureBoot Off
```

2. Add a COM port. Use the [Set-VMComPort](#) cmdlet to do this. For example, the following command configures the first COM port on virtual machine, TestVM, to connect to the named pipe, TestPipe, on the local computer:

PowerShell

```
Set-VMComPort -VMName 'TestVM' -Number 1 -Path '\\.\pipe\TestPipe'
```

Note

Configured COM ports aren't listed in the settings of a virtual machine in Hyper-V Manager.

See Also

- [Linux and FreeBSD Virtual Machines on Hyper-V](#)
- [Use local resources on Hyper-V virtual machine with VMConnect](#)
- [Plan for Hyper-V scalability in Windows Server 2016](#)

Plan for Hyper-V networking in Windows Server

Article • 12/07/2022 • 5 minutes to read

Applies to: Windows Server 2022, Microsoft Hyper-V Server 2016, Windows Server 2016, Microsoft Hyper-V Server 2019, Windows Server 2019

A basic understanding of networking in Hyper-V helps you plan networking for virtual machines. This article also covers some networking considerations when using live migration and when using Hyper-V with other server features and roles.

Hyper-V networking basics

Basic networking in Hyper-V is fairly simple. It uses two parts - a virtual switch and a virtual networking adapter. You'll need at least one of each to establish networking for a virtual machine. The virtual switch connects to any Ethernet-based network. The virtual network adapter connects to a port on the virtual switch, which makes it possible for a virtual machine to use a network.

The easiest way to establish basic networking is to create a virtual switch when you install Hyper-V. Then, when you create a virtual machine, you can connect it to the switch. Connecting to the switch automatically adds a virtual network adapter to the virtual machine. For instructions, see [Create a virtual switch for Hyper-V virtual machines](#).

To handle different types of networking, you can add virtual switches and virtual network adapters. All switches are part of the Hyper-V host, but each virtual network adapter belongs to only one virtual machine.

The virtual switch is a software-based layer-2 Ethernet network switch. It provides built-in features for monitoring, controlling, and segmenting traffic, as well as security, and diagnostics. You can add to the set of built-in features by installing plug-ins, also called *extensions*. These are available from independent software vendors. For more information about the switch and extensions, see [Hyper-V Virtual Switch](#).

Switch and network adapter choices

Hyper-V offers three types of virtual switches and two types of virtual network adapters. You'll choose which one of each you want when you create it. You can use Hyper-V

Manager or the Hyper-V module for Windows PowerShell to create and manage virtual switches and virtual network adapters. Some advanced networking capabilities, such as extended port access control lists (ACLs), can only be managed by using cmdlets in the Hyper-V module.

You can make some changes to a virtual switch or virtual network adapter after you create it. For example, it's possible to change an existing switch to a different type, but doing that affects the networking capabilities of all the virtual machines connected to that switch. So, you probably won't do this unless you made a mistake or need to test something. As another example, you can connect a virtual network adapter to a different switch, which you might do if you want to connect to a different network. But, you can't change a virtual network adapter from one type to another. Instead of changing the type, you'd add another virtual network adapter and choose the appropriate type.

Virtual switch types are:

- **External virtual switch** - Connects to a wired, physical network by binding to a physical network adapter.
- **Internal virtual switch** - Connects to a network that can be used only by the virtual machines running on the host that has the virtual switch, and between the host and the virtual machines.
- **Private virtual switch** - Connects to a network that can be used only by the virtual machines running on the host that has the virtual switch, but doesn't provide networking between the host and the virtual machines.

Virtual switch options:

Setting name	Description
Allow management operating system to share this network adapter	Allow the Hyper-V host to share the use of the virtual switch and NIC or NIC team with the virtual machine. With this enabled, the host can use any of the settings that you configure for the virtual switch, such as Quality of Service (QoS) settings, security settings, or other features of the Hyper-V virtual switch.
Enable single-root I/O virtualization (SR-IOV)	Allow virtual machine traffic to bypass the virtual machine switch and go directly to the physical NIC. SR-IOV is only available for virtual machines running Windows Server. For more information, see Single-Root I/O Virtualization in the Poster Companion Reference: Hyper-V Networking.

Virtual network adapter types are:

- **Hyper-V specific network adapter** - Available for both generation 1 and generation 2 virtual machines. It's designed specifically for Hyper-V and requires a driver that's included in Hyper-V integration services. This type of network adapter faster and is the recommended choice unless you need to boot to the network or are running an unsupported guest operating system. The required driver is provided only for supported guest operating systems. Note that in Hyper-V Manager and the networking cmdlets, this type is just referred to as a network adapter.
- **Legacy network adapter** - Available only in generation 1 virtual machines. Emulates an Intel 21140-based PCI Fast Ethernet Adapter and can be used to boot to a network so you can install an operating system from a service such as Windows Deployment Services.

Hyper-V networking and related technologies

Recent Windows Server releases introduced improvements that give you more options for configuring networking for Hyper-V. For example, Windows Server 2012 introduced support for converged networking. This lets you route network traffic through one external virtual switch. Windows Server 2016 builds on this by allowing Remote Direct Memory Access (RDMA) on network adapters bound to a Hyper-V virtual switch. You can use this configuration either with or without Switch Embedded Teaming (SET). For details, see [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#)

Some features rely on specific networking configurations or do better under certain configurations. Consider these when planning or updating your network infrastructure.

Failover clustering - It's a best practice to isolate cluster traffic and use Hyper-V Quality of Service (QoS) on the virtual switch. For details, see [Network Recommendations for a Hyper-V Cluster](#)

Live migration - Use performance options to reduce network and CPU usage and the time it takes to complete a live migration. For instructions, see [Set up hosts for live migration without Failover Clustering](#).

Storage Spaces Direct - This feature relies on the SMB3.0 network protocol and RDMA. For details, see [Storage Spaces Direct in Windows Server 2016](#).

Plan for Hyper-V scalability in Windows Server

Article • 11/15/2022 • 5 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This article gives you details about the maximum configuration for components you can add and remove on a Hyper-V host or its virtual machines, such as virtual processors or checkpoints. As you plan your deployment, consider the maximums that apply to each virtual machine, as well as those that apply to the Hyper-V host. Maximums continue to grow in Windows Server versions, in response to requests to support newer scenarios such as machine learning and data analytics.

ⓘ Note

For information about System Center Virtual Machine Manager (VMM), see [Virtual Machine Manager](#). VMM is a Microsoft product for managing a virtualized data center that is sold separately.

Maximums for virtual machines

These maximums apply to each virtual machine. Not all components are available in both generations of virtual machines. For a comparison of the generations, see [Should I create a generation 1 or 2 virtual machine in Hyper-V?](#)

Component	Maximum	Notes
Checkpoints	50	The actual number may be lower, depending on the available storage. Each checkpoint is stored as an .avhd file that uses physical storage.
Memory	12 TB for generation 2; 1 TB for generation 1	Review the requirements for the specific operating system to determine the minimum and recommended amounts.
Serial (COM) ports	2	None.

Component	Maximum	Notes
Size of physical disks attached directly to a virtual machine	Varies	Maximum size is determined by the guest operating system.
Virtual Fibre Channel adapters	4	As a best practice, we recommended that you connect each virtual Fibre Channel Adapter to a different virtual SAN.
Virtual floppy devices	1 virtual floppy drive	None.
Virtual hard disk capacity	64 TB for VHDX format; 2040 GB for VHD format	Each virtual hard disk is stored on physical media as either a .vhdx or a .vhd file, depending on the format used by the virtual hard disk.
Virtual IDE disks	4	The startup disk (sometimes called the boot disk) must be attached to one of the IDE devices. The startup disk can be either a virtual hard disk or a physical disk attached directly to a virtual machine.
Virtual processors	240 for generation 2; 64 for generation 1; 320 available to the host OS (root partition)	The number of virtual processors supported by a guest operating system might be lower. For details, see the information published for the specific operating system. The root partition limit of 320 applies to Windows Server 2016 and 2019. Beginning in Windows Server 2022, the limit is increased to 1,024.
Virtual SCSI controllers	4	Use of virtual SCSI devices requires integration services, which are available for supported guest operating systems. For details on which operating systems are supported, see Supported Linux and FreeBSD virtual machines and Supported Windows guest operating systems .
Virtual SCSI disks	256	Each SCSI controller supports up to 64 disks, which means that each virtual machine can be configured with as many as 256 virtual SCSI disks. (4 controllers x 64 disks per controller)

Component	Maximum	Notes
Virtual network adapters	Windows Server 2019 and later supports 68 total: <ul style="list-style-type: none"> • 64 Hyper-V specific network adapters • 4 legacy network adapters; Windows Server 2016 supports 12 total: <ul style="list-style-type: none"> • 8 Hyper-V specific network adapters • 4 legacy network adapters 	The Hyper-V specific network adapter provides better performance and requires a driver included in integration services. For more information, see Plan for Hyper-V networking in Windows Server .

Maximums for Hyper-V hosts

These maximums apply to each Hyper-V host.

Component	Maximum	Notes
Logical processors	512	Both of these must be enabled in the firmware: <ul style="list-style-type: none"> • Hardware-assisted virtualization • Hardware-enforced Data Execution Prevention (DEP) On Windows Server 2016 and 2019, the host OS (root partition) will only see maximum 320 logical processors. Beginning in Windows Server 2022 the host OS (root partition) will see maximum 1,024 processors.

Component	Maximum	Notes
Memory	48 TB (Windows Server 2022 and Azure Stack HCI 21H2); 24 TB (Windows Server 2016 and 2019)	None.
Network adapter teams (NIC Teaming)	No limits imposed by Hyper-V.	None.
Physical network adapters	No limits imposed by Hyper-V.	None.
Running virtual machines per server	1024	None.
Storage	Limited by what is supported by the host operating system. No limits imposed by Hyper-V.	Note: Microsoft supports network-attached storage (NAS) when using SMB 3.0. NFS-based storage is not supported.
Virtual network switch ports per server	Varies; no limits imposed by Hyper-V.	The practical limit depends on the available computing resources.
Virtual processors per logical processor	No ratio imposed by Hyper-V.	None.
Virtual processors per server	2048	None.
Virtual storage area networks (SANs)	No limits imposed by Hyper-V.	None.
Virtual switches	Varies; no limits imposed by Hyper-V.	The practical limit depends on the available computing resources.

Failover Clusters and Hyper-V

This table lists the maximums that apply when using Hyper-V and Failover Clustering. It's important to do capacity planning to ensure that there will be enough hardware resources to run all the virtual machines in a clustered environment.

To learn about updates to Failover Clustering, including new features for virtual machines, see [What's New in Failover Clustering in Windows Server 2016](#).

Component	Maximum	Notes
Nodes per cluster	64	Consider the number of nodes you want to reserve for failover, as well as maintenance tasks such as applying updates. We recommend that you plan for enough resources to allow for 1 node to be reserved for failover, which means it remains idle until another node is failed over to it. (This is sometimes referred to as a passive node.) You can increase this number if you want to reserve additional nodes. There is no recommended ratio or multiplier of reserved nodes to active nodes; the only requirement is that the total number of nodes in a cluster can't exceed the maximum of 64.
Running virtual machines per cluster and per node	8,000 per cluster	Several factors can affect the real number of virtual machines you can run at the same time on one node, such as: <ul style="list-style-type: none">- Amount of physical memory being used by each virtual machine.- Networking and storage bandwidth.- Number of disk spindles, which affects disk I/O performance.

Plan for Hyper-V security in Windows Server

Article • 08/17/2021 • 3 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Secure the Hyper-V host operating system, the virtual machines, configuration files, and virtual machine data. Use the following list of recommended practices as a checklist to help you secure your Hyper-V environment.

Secure the Hyper-V host

- **Keep the host OS secure.**
 - Minimize the attack surface by using the minimum Windows Server installation option that you need for the management operating system. For more information, see the [Installation Options section](#) of the Windows Server technical content library. We don't recommend that you run production workloads on Hyper-V on Windows 10.
 - Keep the Hyper-V host operating system, firmware, and device drivers up to date with the latest security updates. Check your vendor's recommendations to update firmware and drivers.
 - Don't use the Hyper-V host as a workstation or install any unnecessary software.
 - Remotely manage the Hyper-V host. If you must manage the Hyper-V host locally, use Credential Guard. For more information, see [Protect derived domain credentials with Credential Guard](#).
 - Enable code integrity policies. Use virtualization-based security protected Code Integrity services. For more information, see [Device Guard Deployment Guide](#).
- **Use a secure network.**
 - Use a separate network with a dedicated network adapter for the physical Hyper-V computer.
 - Use a private or secure network to access VM configurations and virtual hard disk files.
 - Use a private/dedicated network for your live migration traffic. Consider enabling IPSec on this network to use encryption and secure your VM's data going over the network during migration. For more information, see [Set up hosts for live migration without Failover Clustering](#).

- **Secure storage migration traffic.**

Use SMB 3.0 for end-to-end encryption of SMB data and data protection tampering or eavesdropping on untrusted networks. Use a private network to access the SMB share contents to prevent man-in-the-middle attacks. For more information, see [SMB Security Enhancements](#).

- **Configure hosts to be part of a guarded fabric.**

For more information, see [Guarded fabric](#).

- **Secure devices.**

Secure the storage devices where you keep virtual machine resource files.

- **Secure the hard drive.**

Use BitLocker Drive Encryption to protect resources.

- **Harden the Hyper-V host operating system.**

Use the baseline security setting recommendations described in the [Windows Server Security Baseline](#).

- **Grant appropriate permissions.**

- Add users that need to manage the Hyper-V host to the Hyper-V administrators group.
- Don't grant virtual machine administrators permissions on the Hyper-V host operating system.

- **Configure anti-virus exclusions and options for Hyper-V.**

Windows Defender already has [automatic exclusions](#) configured. For more information about exclusions, see [Recommended antivirus exclusions for Hyper-V hosts](#).

- **Don't mount unknown VHDs.** This can expose the host to file system level attacks.
- **Don't enable nesting in your production environment unless it's required.**

If you enable nesting, don't run unsupported hypervisors on a virtual machine.

For more secure environments:

- **Use hardware with a Trusted Platform Module (TPM) 2.0 chip to set up a guarded fabric.**

For more information, see [System requirements for Hyper-V on Windows Server 2016](#).

Secure virtual machines

- **Create generation 2 virtual machines for supported guest operating systems.**

For more information, see [Generation 2 security settings](#).

- **Enable Secure Boot.**

For more information, see [Generation 2 security settings](#).

- **Keep the guest OS secure.**

- Install the latest security updates before you turn on a virtual machine in a production environment.
- Install integration services for the supported guest operating systems that need it and keep it up to date. Integration service updates for guests that run supported Windows versions are available through Windows Update.
- Harden the operating system that runs in each virtual machine based on the role it performs. Use the baseline security setting recommendations that are described in the [Windows Security Baseline](#).

- **Use a secure network.**

Make sure virtual network adapters connect to the correct virtual switch and have the appropriate security setting and limits applied.

- **Store virtual hard disks and snapshot files in a secure location.**
- **Secure devices.**

Configure only required devices for a virtual machine. Don't enable discrete device assignment in your production environment unless you need it for a specific scenario. If you do enable it, make sure to only expose devices from trusted vendors.

- **Configure antivirus, firewall, and intrusion detection software** within virtual machines as appropriate based on the virtual machine role.
- **Enable virtualization based security** for guests that run Windows 10 or Windows Server 2016 or later.

For more information, see the [Device Guard Deployment Guide](#).

- Only enable Discrete Device Assignment if needed for a specific workload.

Due to the nature of passing through a physical device, work with the device manufacturer to understand if it should be used in a secure environment.

For more secure environments:

- Deploy virtual machines with shielding enabled and deploy them to a guarded fabric.

For more information, see [Generation 2 security settings](#) and [Guarded fabric](#).

Plan for GPU acceleration in Windows Server

Article • 07/29/2021 • 4 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article introduces the graphics virtualization capabilities available in Windows Server.

When to use GPU acceleration

Depending on your workload, you may want to consider GPU acceleration. Here's what you should consider before choosing GPU acceleration:

- **App and desktop remoting (VDI/DaaS) workloads:** If you're building an app or desktop remoting service with Windows Server, consider the catalogue of apps you expect your users to run. Some types of apps, such as CAD/CAM apps, simulation apps, games, and rendering/visualization apps, rely heavily on 3D rendering to deliver smooth and responsive interactivity. Most customers consider GPUs a necessity for a reasonable user experience with these kinds of apps.
- **Remote rendering, encoding, and visualization workloads:** These graphics-oriented workloads tend to rely heavily on a GPU's specialized capabilities, such as efficient 3D rendering and frame encoding/decoding, in order to achieve cost-effectiveness and throughput goals. For this kind of workload, a single GPU-enabled VM may be able to match the throughput of many CPU-only VMs.
- **HPC and ML workloads:** For highly data-parallel computational workloads, such as high-performance compute and machine learning model training or inference, GPUs can dramatically shorten time to result, time to inference, and training time. Alternatively, they may offer better cost-effectiveness than a CPU-only architecture at a comparable performance level. Many HPC and machine learning frameworks have an option to enable GPU acceleration; consider whether this might benefit your specific workload.

GPU virtualization in Windows Server

GPU virtualization technologies enable GPU acceleration in a virtualized environment, typically within virtual machines. If your workload is virtualized with Hyper-V, then you'll

need to employ graphics virtualization in order to provide GPU acceleration from the physical GPU to your virtualized apps or services. However, if your workload runs directly on physical Windows Server hosts, then you have no need for graphics virtualization; your apps and services already have access to the GPU capabilities and APIs natively supported in Windows Server.

The following graphics virtualization technologies are available to Hyper-V VMs in Windows Server:

- [Discrete Device Assignment \(DDA\)](#)
- [RemoteFX vGPU](#)

In addition to VM workloads, Windows Server also supports GPU acceleration of containerized workloads within Windows Containers. For more information, see [GPU Acceleration in Windows containers](#).

Discrete Device Assignment (DDA)

Discrete Device Assignment (DDA), also known as GPU pass-through, allows you to dedicate one or more physical GPUs to a virtual machine. In a DDA deployment, virtualized workloads run on the native driver and typically have full access to the GPU's functionality. DDA offers the highest level of app compatibility and potential performance. DDA can also provide GPU acceleration to Linux VMs, subject to support.

A DDA deployment can accelerate only a limited number of virtual machines, since each physical GPU can provide acceleration to at most one VM. If you're developing a service whose architecture supports shared virtual machines, consider hosting multiple accelerated workloads per VM. For example, if you're building a desktop remoting service with RDS, you can improve user scale by leveraging the multi-session capabilities of Windows Server to host multiple user desktops on each VM. These users will share the benefits of GPU acceleration.

For more information, see these topics:

- [Plan for deploying Discrete Device Assignment](#)
- [Deploy graphics devices using Discrete Device Assignment](#)

RemoteFX vGPU

Note

Because of security concerns, RemoteFX vGPU is disabled by default on all versions of Windows starting with the July 14, 2020 Security Update and removed starting with the April 13, 2021 Security Update. To learn more, see [KB 4570006](#).

RemoteFX vGPU is a graphics virtualization technology that allows a single physical GPU to be shared among multiple virtual machines. In a RemoteFX vGPU deployment, virtualized workloads run on Microsoft's RemoteFX 3D adapter, which coordinates GPU processing requests between the host and guests. RemoteFX vGPU is most suitable for knowledge worker and high-burst workloads where dedicated GPU resources are not required. RemoteFX vGPU can only provide GPU acceleration to Windows VMs.

For more information, see these topics:

- [Deploy graphics devices using RemoteFX vGPU](#)
- [RemoteFX 3D Video Adapter \(vGPU\) support](#)

Comparing DDA and RemoteFX vGPU

Consider the following functionality and support differences between graphics virtualization technologies when planning your deployment:

Description	RemoteFX vGPU	Discrete Device Assignment
GPU resource model	Dedicated or shared	Dedicated only
VM density	High (one or more GPUs to many VMs)	Low (one or more GPUs to one VM)
App compatibility	DX 11.1, OpenGL 4.4, OpenCL 1.1	All GPU capabilities provided by vendor (DX 12, OpenGL, CUDA)
AVC444	Enabled by default	Available through Group Policy
GPU VRAM	Up to 1 GB dedicated VRAM	Up to VRAM supported by the GPU
Frame rate	Up to 30fps	Up to 60fps
GPU driver in guest	RemoteFX 3D adapter display driver (Microsoft)	GPU vendor driver (NVIDIA, AMD, Intel)
Host OS support	Windows Server 2016	Windows Server 2016; Windows Server 2019

Description	RemoteFX vGPU	Discrete Device Assignment
Guest OS support	Windows Server 2012 R2; Windows Server 2016; Windows 7 SP1; Windows 8.1; Windows 10	Windows Server 2012 R2; Windows Server 2016; Windows Server 2019; Windows 10; Linux
Hypervisor	Microsoft Hyper-V	Microsoft Hyper-V
GPU hardware	Enterprise GPUs (such as Nvidia Quadro/GRID or AMD FirePro)	Enterprise GPUs (such as Nvidia Quadro/GRID or AMD FirePro)
Server hardware	No special requirements	Modern server, exposes IOMMU to OS (usually SR-IOV compliant hardware)

Plan for Deploying Devices using Discrete Device Assignment

Article • 08/16/2021 • 7 minutes to read

Applies to: Windows Server 2022, Microsoft Hyper-V Server 2016, Windows Server 2016, Microsoft Hyper-V Server 2019, Windows Server 2019

Discrete Device Assignment allows physical PCIe hardware to be directly accessible from within a virtual machine. This guide will discuss the type of devices that can use Discrete Device Assignment, host system requirements, limitations imposed on the virtual machines as well as security implications of Discrete Device Assignment.

For Discrete Device Assignment's initial release, we have focused on two device classes to be formally supported by Microsoft: Graphics Adapters and NVMe Storage devices. Other devices are likely to work and hardware vendors are able to offer statements of support for those devices. For these other devices, please reach out to those hardware vendors for support.

To learn about other methods of GPU virtualization, see [Plan for GPU acceleration in Windows Server](#). If you are ready to try out Discrete Device Assignment, you can jump over to [Deploying Graphics Devices Using Discrete Device Assignment](#) or [Deploying Storage Devices using Discrete Device Assignment](#) to get started.

Supported Virtual Machines and Guest Operating Systems

Discrete Device Assignment is supported for Generation 1 or 2 VMs. Additionally, the guests supported include Windows 10, Windows Server 2019, Windows Server 2016, Windows Server 2012r2 with [KB 3133690](#) applied, and various distributions of the Linux OS.

System Requirements

In addition to the [System Requirements for Windows Server](#) and the [System Requirements for Hyper-V](#), Discrete Device Assignment requires server class hardware that is capable of granting the operating system control over configuring the PCIe fabric (Native PCI Express Control). In addition, the PCIe Root Complex has to support "Access

Control Services" (ACS), which enables Hyper-V to force all PCIe traffic through the I/O MMU.

These capabilities usually aren't exposed directly in the BIOS of the server and are often hidden behind other settings. For example, the same capabilities are required for SR-IOV support and in the BIOS you may need to set "Enable SR-IOV." Please reach out to your system vendor if you are unable to identify the correct setting in your BIOS.

To help ensure hardware the hardware is capable of Discrete Device Assignment, our engineers have put together a [Machine Profile Script](#) that you can run on an Hyper-V enabled host to test if your server is correctly setup and what devices are capable of Discrete Device Assignment.

Device Requirements

Not every PCIe device can be used with Discrete Device Assignment. For example, older devices that leverage legacy (INTx) PCI Interrupts are not supported. Jake Oshin's [blog posts](#) go into more detail - however, for the consumer, running the [Machine Profile Script](#) will display which devices are capable of being used for Discrete Device Assignment.

Device manufactures can reach out to their Microsoft representative for more details.

Device Driver

As Discrete Device Assignment passes the entire PCIe device into the Guest VM, a host driver is not required to be installed prior to the device being mounted within the VM. The only requirement on the host is that the device's [PCIe Location Path](#) can be determined. The device's driver can optionally be installed if this helps in identifying the device. For example, a GPU without its device driver installed on the host may appear as a Microsoft Basic Render Device. If the device driver is installed, its manufacturer and model will likely be displayed.

Once the device is mounted inside the guest, the Manufacturer's device driver can now be installed like normal inside the guest virtual machine.

Virtual Machine Limitations

Due to the nature of how Discrete Device Assignment is implemented, some features of a virtual machine are restricted while a device is attached. The following features are not available:

- VM Save/Restore
- Live migration of a VM
- The use of dynamic memory
- Adding the VM to a high availability (HA) cluster

Security

Discrete Device Assignment passes the entire device into the VM. This means all capabilities of that device are accessible from the guest operating system. Some capabilities, like firmware updating, may adversely impact the stability of the system. As such, numerous warnings are presented to the admin when dismounting the device from the host. We highly recommend that Discrete Device Assignment is only used where the tenants of the VMs are trusted.

If the admin desires to use a device with an untrusted tenant, we have provided device manufacturers with the ability to create a Device Mitigation driver that can be installed on the host. Please contact the device manufacturer for details on whether they provide a Device Mitigation Driver.

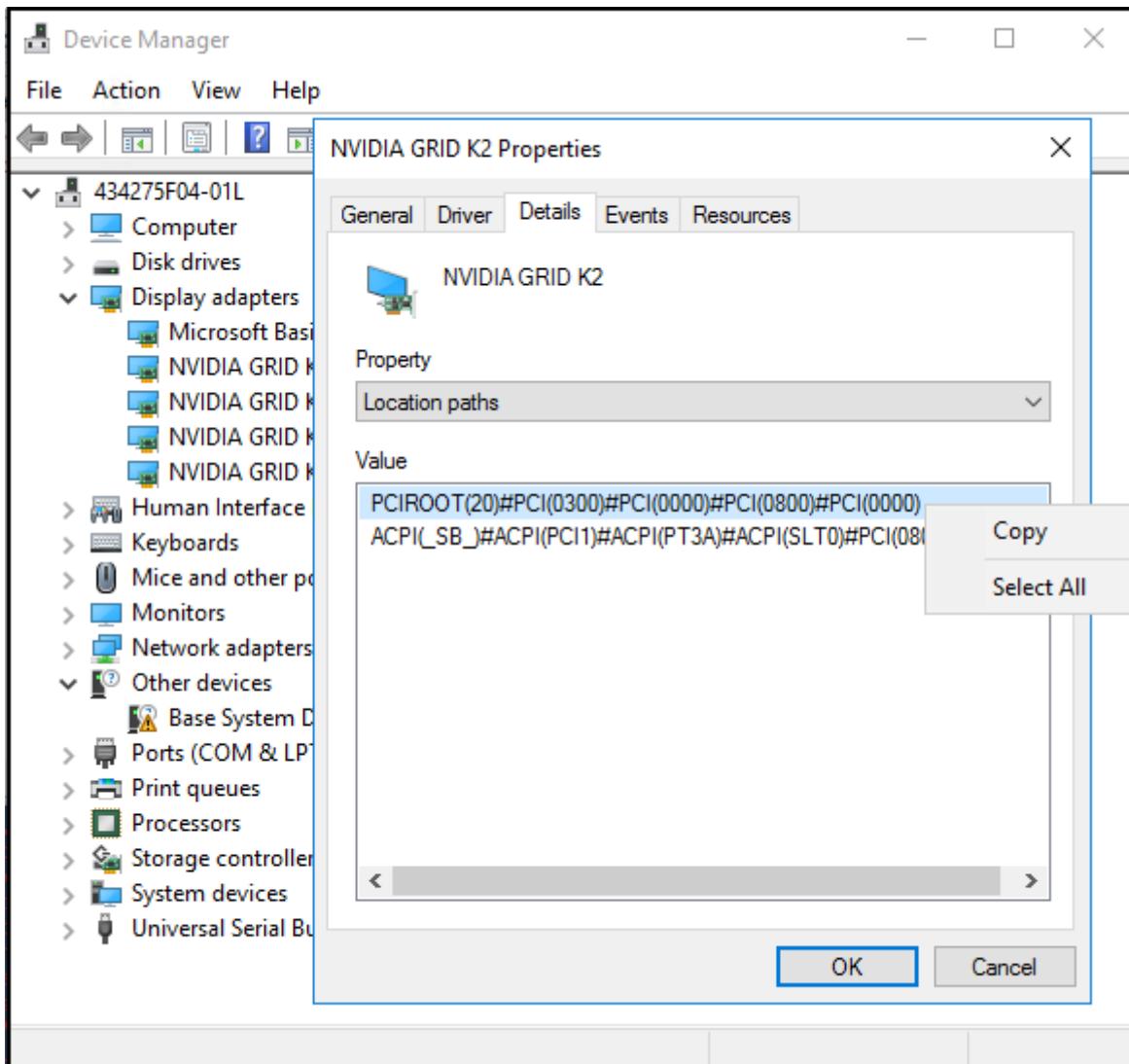
If you would like to bypass the security checks for a device that does not have a Device Mitigation Driver, you will have to pass the `-Force` parameter to the `Dismount-VMHostAssignableDevice` cmdlet. Understand that by doing so, you have changed the security profile of that system and this is only recommended during prototyping or trusted environments.

PCIe Location Path

The PCIe Location path is required to dismount and mount the device from the Host. An example location path looks like the following:

`"PCIROOT(20)#PCI(0300)#PCI(0000)#PCI(0800)#PCI(0000)"`. The [Machine Profile Script](#) will also return the Location Path of the PCIe device.

Getting the Location Path by Using Device Manager



- Open Device Manager and locate the device.
- Right click the device and select “Properties.”
- Navigate to the Details tab and select “Location Paths” in the Property drop down.
- Right click the entry that begins with “PCIROOT” and select “Copy.” You now have the location path for that device.

MMIO Space

Some devices, especially GPUs, require additional MMIO space to be allocated to the VM for the memory of that device to be accessible. By default, each VM starts off with 128MB of low MMIO space and 512MB of high MMIO space allocated to it. However, a device might require more MMIO space, or multiple devices may be passed through such that the combined requirements exceed these values. Changing MMIO Space is straight forward and can be performed in PowerShell using the following commands:

PowerShell

```
Set-VM -LowMemoryMappedIoSpace 3Gb -VMName $vm  
Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName $vm
```

The easiest way to determine how much MMIO space to allocate is to use the [Machine Profile Script](#). To download and run the machine profile script, run the following commands in a PowerShell console:

PowerShell

```
curl -o SurveyDDA.ps1  
https://raw.githubusercontent.com/MicrosoftDocs/Virtualization-  
Documentation/live/hyperv-tools/DiscreteDeviceAssignment/SurveyDDA.ps1  
.\\SurveyDDA.ps1
```

For devices that are able to be assigned, the script will display the MMIO requirements of a given device like the example below:

PowerShell

```
NVIDIA GRID K520  
Express Endpoint -- more secure.  
...  
And it requires at least: 176 MB of MMIO gap space  
...
```

The low MMIO space is used only by 32-bit operating systems and devices that use 32-bit addresses. In most circumstances, setting the high MMIO space of a VM will be enough since 32-bit configurations aren't very common.

Important

When assigning MMIO space to a VM, the user needs to be sure set the MMIO space to the sum of the requested MMIO space for all desired assigned devices plus an additional buffer if there are other virtual devices that require a few MB of MMIO space. Use the default MMIO values described above as the buffer for low and high MMIO (128 MB and 512 MB, respectively).

If a user were to assign a single K520 GPU as in the example above, they must set the MMIO space of the VM to the value outputted by the machine profile script plus a buffer--176 MB + 512 MB. If a user were to assign three K520 GPUs, they must set the MMIO space to three times 176 MB plus a buffer, or 528 MB + 512 MB.

For a more in-depth look at MMIO space, see [Discrete Device Assignment - GPUs](#) on the TechCommunity blog.

Machine Profile Script

In order to simplify identifying if the server is configured correctly and what devices are available to be passed through using Discrete Device Assignment, one of our engineers put together the following PowerShell script: [SurveyDDA.ps1.](#) ↗

Before using the script, please ensure you have the Hyper-V role installed and you run the script from a PowerShell command window that has Administrator privileges.

If the system is incorrectly configured to support Discrete Device Assignment, the tool will display an error message as to what is wrong. If the tool finds the system configured correctly, it will enumerate all the devices it can find on the PCIe Bus.

For each device it finds, the tool will display whether it is able to be used with Discrete Device Assignment. If a device is identified as being compatible with Discrete Device Assignment, the script will provide a reason. When a device is successfully identified as being compatible, the device's Location Path will be displayed. Additionally, if that device requires [MMIO space](#), it will be displayed as well.

```
Intel(R) C600/X79 series chipset USB2 Enhanced Host Controller #1 - 1D26
Old-style PCI device, switch port, etc. Not assignable.

Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 IOAPIC - 0E2C
Embedded Endpoint -- less secure.
    And it has no interrupts at all -- assignment can work.
    And it requires at least: 1 MB of MMIO gap space
PCIROOT(0)#PCI(0504)

NVIDIA GRID K520
Express Endpoint -- more secure.
    And its interrupts are message-based, assignment can work.
    And it requires at least: 176 MB of MMIO gap space
PCIROOT(20)#PCI(0200)#PCI(0000)#PCI(0800)#PCI(0000)

Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 0 - 0E20
Embedded Endpoint -- less secure.
All of the interrupts are line-based, no assignment can work.
```

Deploy Hyper-V on Windows Server

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Use these resources to help you deploy Hyper-V on Windows Server 2016.

- [Configure virtual local area networks for Hyper-V](#)
- [Set up hosts for live migration without Failover Clustering](#)
- [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#)
- [Deploy graphics devices using Discrete Device Assignment](#)
- [Deploy graphics devices using RemoteFX vGPU](#)
- [Deploy storage devices using Discrete Device Assignment](#)

Export and Import virtual machines

Article • 01/07/2022 • 3 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to export and import a virtual machine, which is a quick way to move or copy them. This article also discusses some of the choices to make when doing an export or import.

Export a Virtual Machine

An export gathers all required files into one unit--virtual hard disk files, virtual machine configuration files, and any checkpoint files. You can do this on a virtual machine that is in either a started or stopped state.

Using Hyper-V Manager

To create a virtual machine export:

1. In Hyper-V Manager, right-click the virtual machine and select **Export**.
2. Choose where to store the exported files, and click **Export**.

When the export is done, you can see all exported files under the export location.

Using PowerShell

Open a session as Administrator and run a command like the following, after replacing <vm name> and <path>:

```
PowerShell  
  
Export-VM -Name <vm name> -Path <path>
```

For details, see [Export-VM](#).

Import a Virtual Machine

Importing a virtual machine registers the virtual machine with the Hyper-V host. You can import back into the host, or new host. If you're importing to the same host, you don't need to export the virtual machine first, because Hyper-V tries to recreate the virtual machine from available files. Importing a virtual machine registers it so it can be used on the Hyper-V host.

Important

Hyper-V virtual machine configurations have a specific version number. You can only import a virtual machine if the Hyper-V host supports that configuration version. Typically, this means that you can import a virtual machine to a Hyper-V host running a newer version of Hyper-V, but you cannot import a virtual machine created on a newer version of Hyper-V to an older version of Hyper-V. See [Supported virtual machine configuration versions](#) for more information.

The Import Virtual Machine wizard also helps you fix incompatibilities that can exist when moving from one host to another. This is commonly differences in physical hardware, such as memory, virtual switches, and virtual processors.

Import using Hyper-V Manager

To import a virtual machine:

1. From the **Actions** menu in Hyper-V Manager, click **Import Virtual Machine**.
2. Click **Next**.
3. Select the folder that contains the exported files, and click **Next**.
4. Select the virtual machine to import.
5. Choose the import type, and click **Next**. (For descriptions, see [Import types](#), below.)
6. Click **Finish**.

Import using PowerShell

Use the **Import-VM** cmdlet, following the example for the type of import you want. For descriptions of the types, see [Import types](#), below.

Register in place

This type of import uses the files where they are stored at the time of import and retains the virtual machine's ID. The following command shows an example of an import file. Run a similar command with your own values.

PowerShell

```
Import-VM -Path 'C:\<vm export path>\2B91FEB3-F1E0-4FFF-B8BE-  
29CED892A95A.vmcx'
```

Restore

To import the virtual machine specifying your own path for the virtual machine files, run a command like this, replacing the examples with your values:

PowerShell

```
Import-VM -Path 'C:\<vm export path>\2B91FEB3-F1E0-4FFF-B8BE-  
29CED892A95A.vmcx' -Copy -VhdDestinationPath 'D:\Virtual Machines\WIN10DOC'  
-VirtualMachinePath 'D:\Virtual Machines\WIN10DOC'
```

Import as a copy

To complete a copy import and move the virtual machine files to the default Hyper-V location, run a command like this, replacing the examples with your values:

PowerShell

```
Import-VM -Path 'C:\<vm export path>\2B91FEB3-F1E0-4FFF-B8BE-  
29CED892A95A.vmcx' -Copy -GenerateNewId
```

For details, see [Import-VM](#).

Import types

Hyper-V offers three import types:

- **Register in-place** – This type assumes export files are in the location where you'll store and run the virtual machine. The imported virtual machine has the same ID as it did at the time of export. Because of this, if the virtual machine is already registered with Hyper-V, it needs to be deleted before the import works. When the import has completed, the export files become the running state files and can't be removed.

- **Restore the virtual machine** – Restore the virtual machine to a location you choose, or use the default to Hyper-V. This import type creates a copy of the exported files and moves them to the selected location. When imported, the virtual machine has the same ID as it did at the time of export. Because of this, if the virtual machine is already running in Hyper-V, it needs to be deleted before the import can be completed. When the import has completed, the exported files remain intact and can be removed or imported again.
- **Copy the virtual machine** – This is similar to the Restore type in that you select a location for the files. The difference is that the imported virtual machine has a new unique ID, which means you can import the virtual machine to the same host multiple times.

Set up hosts for live migration without Failover Clustering

Article • 07/29/2021 • 7 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to set up hosts that aren't clustered so you can do live migrations between them. Use these instructions if you didn't set up live migration when you installed Hyper-V, or if you want to change the settings. To set up clustered hosts, use tools for Failover Clustering.

Requirements for setting up live migration

To set up non-clustered hosts for live migration, you'll need:

- A user account with permission to perform the various steps. Membership in the local Hyper-V Administrators group or the Administrators group on both the source and destination computers meets this requirement, unless you're configuring constrained delegation. Membership in the Domain Administrators group is required to configure constrained delegation.
- The Hyper-V role in Windows Server 2016 or Windows Server 2012 R2 installed on the source and destination servers. You can do a live migration between hosts running Windows Server 2016 and Windows Server 2012 R2 if the virtual machine is at least version 5.
For version upgrade instructions, see [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#). For installation instructions, see [Install the Hyper-V role on Windows Server](#).

- Source and destination computers that either belong to the same Active Directory domain, or belong to domains that trust each other.
- The Hyper-V management tools installed on a computer running Windows Server 2016 or Windows 10, unless the tools are installed on the source or destination server and you'll run the tools from the server.

Consider options for authentication and networking

Consider how you want to set up the following:

- **Authentication:** Which protocol will be used to authenticate live migration traffic between the source and destination servers? The choice determines whether you'll need to sign on to the source server before starting a live migration:
 - Kerberos lets you avoid having to sign in to the server, but requires constrained delegation to be set up. See below for instructions.
 - CredSSP lets you avoid configuring constrained delegation, but requires you sign in to the source server. You can do this through a local console session, a Remote Desktop session, or a remote Windows PowerShell session.

CredSPP requires signing in for situations that might not be obvious. For example, if you sign in to TestServer01 to move a virtual machine to TestServer02, and then want to move the virtual machine back to TestServer01, you'll need to sign in to TestServer02 before you try to move the virtual machine back to TestServer01. If you don't do this, the authentication attempt fails, an error occurs, and the following message is displayed:

"Virtual machine migration operation failed at migration Source. Failed to establish a connection with host *computer name*: No credentials are available in the security package 0x8009030E."

- **Performance:** Does it make sense to configure performance options? These options can reduce network and CPU usage, as well as make live migrations go faster. Consider your requirements and your infrastructure, and test different configurations to help you decide. The options are described at the end of step 2.
- **Network preference:** Will you allow live migration traffic through any available network, or isolate the traffic to specific networks? As a security best practice, we recommend that you isolate the traffic onto trusted, private networks because live migration traffic is not encrypted when it is sent over the network. Network isolation can be achieved through a physically isolated network or through another trusted networking technology such as VLANs.

Step 1: Configure constrained delegation (optional)

If you have decided to use Kerberos to authenticate live migration traffic, configure constrained delegation using an account that is a member of the Domain Administrators group.

Use the Users and Computers snap-in to configure constrained delegation

1. Open the Active Directory Users and Computers snap-in. (From Server Manager, select the server if it's not selected, click **Tools >> Active Directory Users and Computers**).
2. From the navigation pane in **Active Directory Users and Computers**, select the domain and double-click the **Computers** folder.
3. From the **Computers** folder, right-click the computer account of the source server and then click **Properties**.
4. From **Properties**, click the **Delegation** tab.
5. On the delegation tab, select **Trust this computer for delegation to the specified services only** and then select **Use any authentication protocol**.
6. Click **Add**.
7. From **Add Services**, click **Users or Computers**.
8. From **Select Users or Computers**, type the name of the destination server. Click **Check Names** to verify it, and then click **OK**.
9. From **Add Services**, in the list of available services, do the following and then click **OK**:
 - To move virtual machine storage, select **cifs**. This is required if you want to move the storage along with the virtual machine, as well as if you want to move only a virtual machine's storage. If the server is configured to use SMB storage for Hyper-V, this should already be selected.
 - To move virtual machines, select **Microsoft Virtual System Migration Service**.
10. On the **Delegation** tab of the Properties dialog box, verify that the services you selected in the previous step are listed as the services to which the destination computer can present delegated credentials. Click **OK**.
11. From the **Computers** folder, select the computer account of the destination server and repeat the process. In the **Select Users or Computers** dialog box, be sure to

specify the name of the source server.

The configuration changes take effect after both of the following happen:

- The changes are replicated to the domain controllers that the servers running Hyper-V are logged into.
- The domain controller issues a new Kerberos ticket.

Step 2: Set up the source and destination computers for live migration

This step includes choosing options for authentication and networking. As a security best practice, we recommend that you select specific networks to use for live migration traffic, as discussed above. This step also shows you how to choose the performance option.

Use Hyper-V Manager to set up the source and destination computers for live migration

1. Open Hyper-V Manager. (From Server Manager, click **Tools >>Hyper-V Manager**.)
2. In the navigation pane, select one of the servers. (If it isn't listed, right-click **Hyper-V Manager**, click **Connect to Server**, type the server name, and click **OK**. Repeat to add more servers.)
3. In the Action pane, click **Hyper-V Settings >>Live Migrations**.
4. In the **Live Migrations** pane, check **Enable incoming and outgoing live migrations**.
5. Under **Simultaneous live migrations**, specify a different number if you don't want to use the default of 2.
6. Under **Incoming live migrations**, if you want to use specific network connections to accept live migration traffic, click **Add** to type the IP address information. Otherwise, click **Use any available network for live migration**. Click **OK**.
7. To choose Kerberos and performance options, expand **Live Migrations** and then select **Advanced Features**.
 - If you have configured constrained delegation, under **Authentication protocol**, select **Kerberos**.

- Under **Performance options**, review the details and choose a different option if it's appropriate for your environment.

8. Click **OK**.

9. Select the other server in Hyper-V Manager and repeat the steps.

Use Windows PowerShell to set up the source and destination computers for live migration

Three cmdlets are available for configuring live migration on non-clustered hosts: [Enable-VMMigration](#), [Set-VMMigrationNetwork](#), and [Set-VMHost](#). This example uses all three and does the following:

- Configures live migration on the local host
- Allows incoming migration traffic only on a specific network
- Chooses Kerberos as the authentication protocol

Each line represents a separate command.

```
PowerShell

PS C:\> Enable-VMMigration

PS C:\> Set-VMMigrationNetwork 192.168.10.1

PS C:\> Set-VMHost -VirtualMachineMigrationAuthenticationType Kerberos
```

`Set-VMHost` also lets you choose a performance option (and many other host settings). For example, to choose SMB but leave the authentication protocol set to the default of CredSSP, type:

```
PowerShell

PS C:\> Set-VMHost -VirtualMachineMigrationPerformanceOption SMB
```

This table describes how the performance options work.

Option	Description
TCP/IP	Copies the memory of the virtual machine to the destination server over a TCP/IP connection.
Compression	Compresses the memory content of the virtual machine before copying it to the destination server over a TCP/IP connection. Note: This is the default setting.

Option	Description
SMB	<p>Copies the memory of the virtual machine to the destination server over a SMB 3.0 connection.</p> <ul style="list-style-type: none">- SMB Direct is used when the network adapters on the source and destination servers have Remote Direct Memory Access (RDMA) capabilities enabled.- SMB Multichannel automatically detects and uses multiple connections when a proper SMB Multichannel configuration is identified. <p>For more information, see Improve Performance of a File Server with SMB Direct.</p>

Next steps

After you set up the hosts, you're ready to do a live migration. For instructions, see [Use live migration without Failover Clustering to move a virtual machine](#).

Upgrade virtual machine version in Hyper-V on Windows or Windows Server

Article • 01/07/2022 • 8 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2019, Windows Server 2016

Make the latest Hyper-V features available on your virtual machines by upgrading the configuration version. Don't do this until:

- You upgrade your Hyper-V hosts to the latest version of Windows or Windows Server.
- You upgrade the cluster functional level.
- You're sure that you won't need to move the virtual machine back to a Hyper-V host that runs a previous version of Windows or Windows Server.

For more information, see [Cluster Operating System Rolling Upgrade](#) and [Perform a rolling upgrade of a Hyper-V host cluster in VMM](#).

Step 1: Check the virtual machine configuration versions

1. On the Windows desktop, click the Start button and type any part of the name **Windows PowerShell**.
2. Right-click Windows PowerShell and select **Run as Administrator**.
3. Use the [Get-VM cmdlet](#). Run the following command to get the versions of your virtual machines.

PowerShell

```
Get-VM * | Format-Table Name, Version
```

You can also see the configuration version in Hyper-V Manager by selecting the virtual machine and looking at the **Summary** tab.

Step 2: Upgrade the virtual machine configuration version

1. Shut down the virtual machine in Hyper-V Manager.
2. Select Action > Upgrade Configuration Version. If this option isn't available for the virtual machine, then it's already at the highest configuration version supported by the Hyper-V host.

To upgrade the virtual machine configuration version by using Windows PowerShell, use the [Update-VMVersion](#) cmdlet. Run the following command where `vmname` is the name of the virtual machine.

```
PowerShell
```

```
Update-VMVersion <vmname>
```

Supported virtual machine configuration versions

Using the PowerShell cmdlet [Get-VMHostSupportedVersion](#) you can see what virtual machine configuration versions your Hyper-V Host supports. When you create a virtual machine, it's created with the default configuration version. To see which virtual machine configuration versions your Hyper-V Host supports and what the default is, run the following command.

```
PowerShell
```

```
Get-VMHostSupportedVersion
```

If you need to create a virtual machine that you can move to a Hyper-V Host that runs an older version of Windows, use the [New-VM](#) cmdlet with the `-Version` parameter. For example, to create a virtual machine named "WindowsCV5" with configuration version 5.0, run the following command:

```
PowerShell
```

```
New-VM -Name "WindowsCV5" -Version 5.0
```

 Note

You can only import a virtual machine if the Hyper-V host supports that configuration version. Typically, this means that you can import a virtual machine to a Hyper-V host running a newer version of Hyper-V, but you cannot import a virtual machine created on a newer version of Hyper-V to an older version of Hyper-V.

If the VM's configuration version is not listed as supported for your Hyper-V host OS in the table below, you either have to upgrade the VM configuration version to a newer version or create a new VM of the same generation using the existing virtual hard disks before you can start the VM.

Supported VM configuration versions for long-term servicing hosts

The following table lists the VM configuration versions for hosts running a long-term servicing version of Windows.

Supported VM configuration versions for Semi-Annual Channel hosts

The following table lists the VM configuration versions for hosts running a Semi-Annual Channel version of Windows. To get more information on Semi-Annual Channel versions of Windows, visit the following pages for [Windows Server](#) and [Windows](#).

Hyper-V host	10.0	9.3	9.2	9.1	9.0	8.3	8.2	8.1	8.0	7.1	7.0	6.2	5.0
Windows version													
Windows 10 May 2019 Update (version 1903)	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server, version 1809	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 October 2018 Update (version 1809)	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓
Windows Server, version 1803	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 April 2018 Update (version 1803)	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Windows 10 Fall Creators Update (version 1709)	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓
Windows 10 Creators Update (version 1703)	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Windows 10 Anniversary Update (version 1607)	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓

Why should I upgrade the virtual machine configuration version?

When you move or import a virtual machine to a computer that runs Hyper-V on Windows Server 2019, Windows Server 2016, or Windows 10, the virtual machine's configuration isn't automatically updated. This means that you can move the virtual machine back to a Hyper-V host that runs a previous version of Windows or Windows Server. But, this also means that you can't use some of the new virtual machine features until you manually update the configuration version.

 **Important**

You can't downgrade a virtual machine configuration version after you've upgraded it.

The virtual machine configuration version represents the compatibility of the virtual machine's configuration, saved state, and snapshot files with the version of Hyper-V. When you update the configuration version, you change the file structure that is used to store the virtual machines configuration and the checkpoint files. You also update the configuration version to the latest version supported by that Hyper-V host. Upgraded virtual machines use a new configuration file format, which is designed to increase the efficiency of reading and writing virtual machine configuration data. The upgrade also reduces the potential for data corruption in the event of a storage failure.

The following table lists descriptions, file name extensions, and default locations for each type of file that's used for new or upgraded virtual machines.

Virtual machine file types	Description
Configuration	Virtual machine configuration information that is stored in binary file format. File name extension: .vmcx Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines
Runtime state	Virtual machine runtime state information that is stored in binary file format. File name extension: .vmrs and .vmgs Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines
Virtual hard disk	Stores virtual hard disks for the virtual machine. File name extension: .vhdx or .vhd Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Hard Disks
Automatic virtual hard disk	Differencing disk files used for virtual machine checkpoints. File name extension: .avhdx Default location: C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Hard Disks
Checkpoint	Checkpoints are stored in multiple checkpoint files. Each checkpoint creates a configuration file and runtime state file. File name extensions: .vmrs and .vmcx Default location: C:\ProgramData\Microsoft\Windows\Snapshots

What happens if I don't upgrade the virtual machine configuration version?

If you have virtual machines that you created with an earlier version of Hyper-V, some features that are available on the newer host OS may not work with those virtual machines until you update the configuration version.

As a general guidance, we recommend updating the configuration version once you have successfully upgraded the virtualization hosts to a newer version of Windows and feel confident that you do not need to roll back. When you are using the [cluster OS rolling upgrade](#) feature, this would typically be after updating the cluster functional level. This way, you will benefit from new features and internal changes and optimizations as well.

 **Note**

Once the VM configuration version is updated, the VM won't be able to start on hosts that do not support the updated configuration version.

The following table shows the minimum virtual machine configuration version required to use some Hyper-V features.

Feature	Minimum VM configuration version
Allow additional processor features for Perfmon	9.0
Automatically expose simultaneous multithreading configuration for VMs running on hosts using the Core Scheduler	9.0
Hibernation support	9.0
Increase the default maximum number for virtual devices to 64 per device (e.g. networking and assigned devices)	8.3
Guest Virtualization-Based Security support (VBS)	8.0
Key storage drive	8.0
Large memory VMs	8.0
Nested Virtualization	8.0
Virtual processor count	8.0
XSAVE support	8.0
Virtual Machine Multi Queues (VMMQ)	7.1
Virtual Trusted Platform Module (vTPM)	7.0

Feature	Minimum VM configuration version
Hot Add/Remove Memory	6.2
PowerShell Direct	6.2
Production Checkpoints	6.2
Secure Boot for Linux VMs	6.2
Virtual Machine Grouping	6.2

For more information about these features, see [What's new in Hyper-V on Windows Server](#).

Deploy graphics devices using Discrete Device Assignment

Article • 07/29/2021 • 6 minutes to read

Applies to: Windows Server 2022, Microsoft Hyper-V Server 2016, Windows Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Starting with Windows Server 2016, you can use Discrete Device Assignment, or DDA, to pass an entire PCIe Device into a VM. This will allow high performance access to devices like [NVMe storage](#) or Graphics Cards from within a VM while being able to leverage the devices native drivers. Please visit the [Plan for Deploying Devices using Discrete Device Assignment](#) for more details on which devices work, what are the possible security implications, etc.

There are three steps to using a device with Discrete Device Assignment:

- Configure the VM for Discrete Device Assignment
- Dismount the Device from the Host Partition
- Assigning the Device to the Guest VM

All command can be executed on the Host on a Windows PowerShell console as an Administrator.

Configure the VM for DDA

Discrete Device Assignment imposes some restrictions to the VMs and the following step needs to be taken.

1. Configure the “Automatic Stop Action” of a VM to TurnOff by executing

```
Set-VM -Name VMName -AutomaticStopAction Turnoff
```

Some Additional VM preparation is required for Graphics Devices

Some hardware performs better if the VM is configured in a certain way. For details on whether or not you need the following configurations for your hardware, please reach

out to the hardware vendor. Additional details can be found on [Plan for Deploying Devices using Discrete Device Assignment](#) and on this [blog post](#).

1. Enable Write-Combining on the CPU

```
Set-VM -GuestControlledCacheTypes $true -VMName VMName
```

2. Configure the 32 bit MMIO space

```
Set-VM -LowMemoryMappedIoSpace 3Gb -VMName VMName
```

3. Configure greater than 32 bit MMIO space

```
Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName VMName
```

💡 Tip

The MMIO space values above are reasonable values to set for experimenting with a single GPU. If after starting the VM, the device is reporting an error relating to not enough resources, you'll likely need to modify these values.

Consult [Plan for Deploying Devices using Discrete Device Assignment](#) to learn how to precisely calculate MMIO requirements.

Dismount the Device from the Host Partition

Optional - Install the Partitioning Driver

Discrete Device Assignment provide hardware vendors the ability to provide a security mitigation driver with their devices. Note that this driver is not the same as the device driver that will be installed in the guest VM. It's up to the hardware vendor's discretion to provide this driver, however, if they do provide it, please install it prior to dismounting the device from the host partition. Please reach out to the hardware vendor for more information on if they have a mitigation driver

If no Partitioning driver is provided, during dismount you must use the `-force` option to bypass the security warning. Please read more about the security

implications of doing this on [Plan for Deploying Devices using Discrete Device Assignment](#).

Locating the Device's Location Path

The PCI Location path is required to dismount and mount the device from the Host. An example location path looks like the following:

"PCIROOT(20)#PCI(0300)#PCI(0000)#PCI(0800)#PCI(0000)". More details on located the Location Path can be found here: [Plan for Deploying Devices using Discrete Device Assignment](#).

Disable the Device

Using Device Manager or PowerShell, ensure the device is "disabled."

Dismount the Device

Depending on if the vendor provided a mitigation driver, you'll either need to use the "-force" option or not.

- If a Mitigation Driver was installed

```
Dismount-VMHostAssignableDevice -LocationPath $locationPath
```

- If a Mitigation Driver was not installed

```
Dismount-VMHostAssignableDevice -force -LocationPath $locationPath
```

Assigning the Device to the Guest VM

The final step is to tell Hyper-V that a VM should have access to the device. In addition to the location path found above, you'll need to know the name of the vm.

```
Add-VMAssignableDevice -LocationPath $locationPath -VMName VMName
```

What's Next

After a device is successfully mounted in a VM, you're now able to start that VM and interact with the device as you normally would if you were running on a bare metal system. This means that you're now able to install the Hardware Vendor's drivers in the VM and applications will be able to see that hardware present. You can verify this by opening device manager in the Guest VM and seeing that the hardware now shows up.

Removing a Device and Returning it to the Host

If you want to return the device back to its original state, you will need to stop the VM and issue the following:

```
#Remove the device from the VM
Remove-VMAssignableDevice -LocationPath $locationPath -VMName VMName
#Mount the device back in the host
Mount-VMHostAssignableDevice -LocationPath $locationPath
```

You can then re-enable the device in device manager and the host operating system will be able to interact with the device again.

Example

Mounting a GPU to a VM

In this example we use PowerShell to configure a VM named "ddatest1" to take the first GPU available by the manufacturer NVIDIA and assign it into the VM.

```
#Configure the VM for a Discrete Device Assignment
$vm = "ddatest1"
#Set automatic stop action to TurnOff
Set-VM -Name $vm -AutomaticStopAction TurnOff
#Enable Write-Combining on the CPU
Set-VM -GuestControlledCacheTypes $true -VMName $vm
#Configure 32 bit MMIO space
Set-VM -LowMemoryMappedIoSpace 3Gb -VMName $vm
#Configure Greater than 32 bit MMIO space
Set-VM -HighMemoryMappedIoSpace 33280Mb -VMName $vm
```

```

#Find the Location Path and disable the Device
#Enumerate all PNP Devices on the system
$pnpdevs = Get-PnpDevice -presentOnly
#Select only those devices that are Display devices manufactured by NVIDIA
$gpudevs = $pnpdevs |where-object {$_.Class -like "Display" -and
$_.Manufacturer -like "NVIDIA"}
#Select the location path of the first device that's available to be
dismounted by the host.
$locationPath = ($gpudevs | Get-PnpDeviceProperty
DEVPKEY_Device_LocationPaths).data[0]
#Disable the PNP Device
Disable-PnpDevice -InstanceId $gpudevs[0].InstanceId

#Dismount the Device from the Host
Dismount-VMHostAssignableDevice -force -LocationPath $locationPath

#Assign the device to the guest VM.
Add-VMAssignableDevice -LocationPath $locationPath -VMName $vm

```

Troubleshooting

If you've passed a GPU into a VM but Remote Desktop or an application isn't recognizing the GPU, check for the following common issues:

- Make sure you've installed the most recent version of the GPU vendor's supported driver and that the driver isn't reporting errors by checking the device state in Device Manager.
- Make sure your device has enough MMIO space allocated within the VM. To learn more, see [MMIO Space](#).
- Make sure you're using a GPU that the vendor supports being used in this configuration. For example, some vendors prevent their consumer cards from working when passed through to a VM.
- Make sure the application being run supports running inside a VM, and that both the GPU and its associated drivers are supported by the application. Some applications have allow-lists of GPUs and environments.
- If you're using the Remote Desktop Session Host role or Windows Multipoint Services on the guest, you will need to make sure that a specific Group Policy entry is set to allow use of the default GPU. Using a Group Policy Object applied to the guest (or the Local Group Policy Editor on the guest), navigate to the following Group Policy item: **Computer Configuration > Administrator Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment > Use the hardware default graphics adapter for all Remote Desktop Services sessions**. Set this value to Enabled, then reboot the VM once the policy has been applied.

Deploy graphics devices using RemoteFX vGPU

Article • 07/29/2021 • 4 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Microsoft Hyper-V Server 2016

ⓘ Note

Because of security concerns, RemoteFX vGPU is disabled by default on all versions of Windows starting with the July 14, 2020 Security Update and removed starting with the April 13, 2021 Security Update. To learn more, see [KB 4570006](#).

The vGPU feature for RemoteFX makes it possible for multiple virtual machines to share a physical GPU. Rendering and compute resources are shared dynamically among virtual machines, making RemoteFX vGPU appropriate for high-burst workloads where dedicated GPU resources are not required. For example, in a VDI service, RemoteFX vGPU can be used to offload app rendering costs to the GPU, with the effect of decreasing CPU load and improving service scalability.

RemoteFX vGPU requirements

Host system requirements:

- Windows Server 2016
- A DirectX 11.0-compatible GPU with a WDDM 1.2-compatible driver
- A CPU with Second Level Address Translation (SLAT) support

Guest VM requirements:

- Supported guest OS. For more information, see [RemoteFX 3D Video Adapter \(vGPU\) support](#).

Additional considerations for guest VMs:

- OpenGL and OpenCL functionality is only available in guests running Windows 10 or Windows Server 2016.
- DirectX 11.0 is only available for guests running Windows 8 or later.

Enable RemoteFX vGPU

To configure RemoteFX vGPU on your Windows Server 2016 host:

1. Install the graphics drivers recommended by your GPU vendor for Windows Server 2016.
2. Create a VM running a guest OS supported by RemoteFX vGPU. To learn more, see [RemoteFX 3D Video Adapter \(vGPU\) support](#).
3. Add the RemoteFX 3D graphics adapter to the VM. To learn more, see [Configure the RemoteFX vGPU 3D adapter](#).

By default, RemoteFX vGPU will use all available and supported GPUs. To limit which GPUs the RemoteFX vGPU uses, follow these steps:

1. Navigate to the Hyper-V settings in Hyper-V Manager.
2. Select **Physical GPUs** in Hyper-V Settings.
3. Select the GPU that you don't want to use, and then clear **Use this GPU with RemoteFX**.

Configure the RemoteFX vGPU 3D adapter

You can use either the Hyper-V Manager UI or PowerShell cmdlets to configure the RemoteFX vGPU 3D graphics adapter.

Configure RemoteFX vGPU with Hyper-V Manager

1. Stop the VM if it's currently running.
2. Open Hyper-V Manager, navigate to **VM Settings**, then select **Add Hardware**.
3. Select **RemoteFX 3D Graphics Adapter**, then select **Add**.
4. Set the maximum number of monitors, maximum monitor resolution, and dedicated video memory, or leave the default values.

Note

- Setting higher values for any of these options will impact your service scale, so you should only set what is necessary.
- When you need to use 1 GB of dedicated VRAM, use a 64-bit guest VM instead of 32-bit (x86) for best results.

5. Select **OK** to finish the configuration.

Configure RemoteFX vGPU with PowerShell cmdlets

Use the following PowerShell cmdlets to add, review, and configure the adapter:

- [Add-VMRemoteFx3dVideoAdapter](#)
- [Get-VMRemoteFx3dVideoAdapter](#)
- [Set-VMRemoteFx3dVideoAdapter](#)
- [Get-VMRemoteFXPhysicalVideoAdapter](#)

Monitor performance

The performance and scale of a RemoteFX vGPU-enabled service are determined by a variety of factors such as number of GPUs on your system, total GPU memory, amount of system memory and memory speed, number of CPU cores and CPU clock frequency, storage speed, and NUMA implementation.

Host system memory

For every VM enabled with a vGPU, RemoteFX uses system memory both in the guest operating system and in the host server. The hypervisor guarantees the availability of system memory for a guest operating system. On the host, each vGPU-enabled virtual desktop needs to advertise its system memory requirement to the hypervisor. When the vGPU-enabled virtual desktop starts, the hypervisor reserves additional system memory in the host.

The memory requirement for the RemoteFX-enabled server is dynamic because the amount of memory consumed on the RemoteFX-enabled server is dependent on the number of monitors that are associated with the vGPU-enabled virtual desktops and the maximum resolution for those monitors.

Host GPU video memory

Every vGPU-enabled virtual desktop uses the GPU hardware video memory on the host server to render the desktop. In addition, a codec uses the video memory to compress the rendered screen. The amount of memory needed for rendering and compression is directly based on the number of monitors provisioned to the virtual machine. The amount of reserved video memory varies based on the system screen resolution and how many monitors there are. Some users require a higher screen resolution for specific

tasks, but there's greater scalability with lower resolution settings if all other settings remain constant.

Host CPU

The hypervisor schedules the host and VMs on the CPU. The overhead is increased on a RemoteFX-enabled host because the system runs an additional process (rdvqm.exe) per vGPU-enabled virtual desktop. This process uses the graphics device driver to run commands on the GPU. The codec also uses the CPU to compress screen data that needs to be sent back to the client.

More virtual processors mean a better user experience. We recommend allocating at least two virtual CPUs per vGPU-enabled virtual desktop. We also recommend using the x64 architecture for vGPU-enabled virtual desktops because the performance on x64 virtual machines is better compared to x86 virtual machines.

GPU processing power

Every vGPU-enabled virtual desktop has a corresponding DirectX process that runs on the host server. This process replays all graphics commands it receives from the RemoteFX virtual desktop onto the physical GPU. This is like running multiple DirectX applications at the same time on the same physical GPU.

Usually, graphics devices and drivers are tuned to run only a few applications on the desktop at a time, but RemoteFX stretches the GPUs to go even further. vGPUs come with performance counters that measure the GPU response to RemoteFX requests and help you make sure the GPUs aren't stretched too far.

When a GPU is low on resources, read and write operations take a long time to complete. Administrators can use performance counters to know when to adjust resources and prevent downtime for users.

Learn more about performance counters for monitoring RemoteFX vGPU behavior at [Diagnose graphics performance issues in Remote Desktop](#).

Deploy NVMe Storage Devices using Discrete Device Assignment

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Microsoft Hyper-V Server 2016, Windows Server 2016

Starting with Windows Server 2016, you can use Discrete Device Assignment, or DDA, to pass an entire PCIe Device into a VM. This will allow high performance access to devices like NVMe storage or Graphics Cards from within a VM while being able to leverage the devices native drivers. Please visit the [Plan for Deploying Devices using Discrete Device Assignment](#) for more details on which devices work, what are the possible security implications, etc. There are three steps to using a device with DDA:

- Configure the VM for DDA
- Dismount the Device from the Host Partition
- Assigning the Device to the Guest VM

All command can be executed on the Host on a Windows PowerShell console as an Administrator.

Configure the VM for DDA

Discrete Device Assignment imposes some restrictions to the VMs and the following step needs to be taken.

1. Configure the “Automatic Stop Action” of a VM to TurnOff by executing

```
Set-VM -Name VMName -AutomaticStopAction TurnOff
```

Dismount the Device from the Host Partition

Locating the Device's Location Path

The PCI Location path is required to dismount and mount the device from the Host. An example location path looks like the following:

"PCIROOT(20)#PCI(0300)#PCI(0000)#PCI(0800)#PCI(0000)". More details on located the Location Path can be found here: [Plan for Deploying Devices using Discrete Device Assignment](#).

Disable the Device

Using Device Manager or PowerShell, ensure the device is "disabled."

Dismount the Device

```
Dismount-VMHostAssignableDevice -LocationPath $locationPath
```

Assigning the Device to the Guest VM

The final step is to tell Hyper-V that a VM should have access to the device. In addition to the location path found above, you'll need to know the name of the vm.

```
Add-VMAssignableDevice -LocationPath $locationPath -VMName VMName
```

What's Next

After a device is successfully mounted in a VM, you're now able to start that VM and interact with the device as you normally would if you were running on a bare metal system. You can verify this by opening device manager in the Guest VM and seeing that the hardware now shows up.

Removing a Device and Returning it to the Host

If you want to return the device back to its original state, you will need to stop the VM and issue the following:

```
#Remove the device from the VM
Remove-VMAssignableDevice -LocationPath $locationPath -VMName VMName
#Mount the device back in the host
Mount-VMHostAssignableDevice -LocationPath $locationPath
```

You can then re-enable the device in device manager and the host operating system will be able to interact with the device again.

Manage Hyper-V on Windows Server

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Windows Server 2019

Use the resources in this section to help you manage Hyper-V on Windows Server:

- [Configuring Persistent Memory Devices for Hyper-V VMs](#)
- [Choose between standard or production checkpoints](#)
- [Create a VHD set](#)
- [Enable or disable checkpoints](#)
- [Manage hosts with Hyper-V Manager](#)
- [Manage host CPU resource controls](#)
- [Using VM CPU Groups](#)
- [Manage hypervisor scheduler types](#)
- [About Hyper-V scheduler type selection](#)
- [Manage Integration Services](#)
- [Manage Windows VMs with PowerShell Direct](#)
- [Set up Hyper-V Replica](#)
- [Enable Intel Performance Monitoring Hardware](#)
- [Move VMs with live migration](#)

Cmdlets for configuring persistent memory devices for Hyper-V VMs

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2019

This article provides system administrators and IT Pros with information about configuring Hyper-V VMs with persistent memory (aka storage class memory or NVDIMM). JDEC-compliant NVDIMM-N persistent memory devices are supported in Windows Server 2016 and Windows 10 and provide byte-level access to very low latency non-volatile devices. VM persistent memory devices are supported in Windows Server 2019.

Create a persistent memory device for a VM

Use the [New-VHD](#) cmdlet to create a persistent memory device for a VM. The device must be created on an existing NTFS DAX volume. The new filename extension (.vhdpmem) is used to specify that the device is a persistent memory device. Only the fixed VHD file format is supported.

Example: `New-VHD d:\VMPMEMDevice1.vhdpmem -Fixed -SizeBytes 4GB`

Create a VM with a persistent memory controller

Use the [New-VM cmdlet](#) to create a Generation 2 VM with specified memory size and path to a VHDX image. Then, use [Add-VMPmemController](#) to add a persistent memory controller to a VM.

Example:

PowerShell

```
New-VM -Name "ProductionVM1" -MemoryStartupBytes 1GB -VHDPath  
c:\vhd\BaseImage.vhdx
```

```
Add-VMPmemController ProductionVM1x
```

Attach a persistent memory device to a VM

Use [Add-VMHardDiskDrive](#) to attach a persistent memory device to a VM

Example: `Add-VMHardDiskDrive ProductionVM1 PMEM -ControllerLocation 1 -Path D:\VPMEMDevice1.vhdpmem`

Persistent memory devices within a Hyper-V VM appear as a persistent memory device to be consumed and managed by the guest operating system. Guest operating systems can use the device as a block or DAX volume. When persistent memory devices within a VM are used as a DAX volume, they benefit from low latency byte-level addressability of the host device (no I/O virtualization on the code path).

Note

Persistent memory is only supported for Hyper-V Gen2 VMs. Live Migration and Storage Migration are not supported for VMs with persistent memory. Production checkpoints of VMs do not include persistent memory state.

Choose between standard or production checkpoints in Hyper-V

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

Starting with Windows Server 2016 and Windows 10, you can choose between standard and production checkpoints for each virtual machine. Production checkpoints are the default for new virtual machines.

- Production checkpoints are "point in time" images of a virtual machine, which can be restored later on in a way that is completely supported for all production workloads. This is achieved by using backup technology inside the guest to create the checkpoint, instead of using saved state technology.
- Standard checkpoints capture the state, data, and hardware configuration of a running virtual machine and are intended for use in development and test scenarios. Standard checkpoints can be useful if you need to recreate a specific state or condition of a running virtual machine so that you can troubleshoot a problem.

Change checkpoints to production or standard checkpoints

1. In **Hyper-V Manager**, right-click the virtual machine and click **Settings**.

2. Under the **Management** section, select **Checkpoints**.

3. Select either production checkpoints or standard checkpoints.

If you choose production checkpoints, you can also specify whether the host should take a standard checkpoint if a production checkpoint can't be taken. If you clear this checkbox and a production checkpoint can't be taken, then no checkpoint is taken.

4. If you want to store the checkpoint configuration files in a different place, change it in the **Checkpoint File Location** section.

5. Click **Apply** to save your changes. If you're done, click **OK** to close the dialog box.

 Note

Only **Production Checkpoints** are supported on guests that run Active Directory Domain Services role (Domain Controller) or Active Directory Lightweight Directory Services role.

Additional References

- [Production checkpoints](#)
- [Enable or disable checkpoints](#)

Create Hyper-V VHD Set files

Article • 12/08/2020 • 2 minutes to read

VHD Set files are a new shared Virtual Disk model for guest clusters in Windows Server 2016. VHD Set files support online resizing of shared virtual disks, support Hyper-V Replica, and can be included in application-consistent checkpoints.

VHD Set files use a new VHD file type, .VHDS. VHD Set files store checkpoint information about the group virtual disk used in guest clusters, in the form of metadata.

Hyper-V handles all aspects of managing the checkpoint chains and merging the shared VHD set. Management software can run disk operations like online resizing on VHD Set files in the same way it does for .VHDX files. This means that management software doesn't need to know about the VHD Set file format.

ⓘ Note

It's important to evaluate the impact of VHD Set files before deployment into production. Make sure that there is no performance or functional degradation in your environment, such as disk latency.

Create a VHD Set file from Hyper-V Manager

1. Open Hyper-V Manager. Click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the Action pane, click **New**, and then click **Hard Disk**.
3. On the **Choose Disk Format** page, select **VHD Set** as the format of the virtual hard disk.
4. Continue through the pages of the wizard to customize the virtual hard disk. You can click **Next** to move through each page of the wizard, or you can click the name of a page in the left pane to move directly to that page.
5. After you have finished configuring the virtual hard disk, click **Finish**.

Create a VHD Set file from Windows PowerShell

Use the [New-VHD](#) cmdlet, with the file type .VHDS in the file path. This example creates a VHD Set file named base.vhds that is 10 Gigabytes in size.

```
PowerShell
```

```
PS c:\>New-VHD -Path c:\base.vhds -SizeBytes 10GB
```

Migrate a shared VHDX file to a VHD Set file

Migrating an existing shared VHDX to a VHDS requires taking the VM offline. This is the recommended process using Windows PowerShell:

1. Remove the VHDX from the VM. For example, run:

```
PowerShell
```

```
PS c:\>Remove-VMHardDiskDrive existing.vhdx
```

2. Convert the VHDX to a VHDS. For example, run:

```
PowerShell
```

```
PS c:\>Convert-VHD existing.vhdx new.vhds
```

3. Add the VHDS to the VM. For example, run:

```
PowerShell
```

```
PS c:\>Add-VMHardDiskDrive new.vhds
```

Enable or disable checkpoints in Hyper-V

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

You can choose to enable or disable checkpoints for each virtual machine.

1. In **Hyper-V Manager**, right-click the virtual machine and click **Settings**.
2. Under the **Management** section, select **Checkpoints**.
3. To allow checkpoints to be taken of this virtual machine, make sure **Enable checkpoints** is selected. To disable checkpoints, clear the **Enable checkpoints** check box.
4. Click **Apply** to apply your changes. If you are done, click **OK** to close the dialog box.

Additional References

[Choose between standard or production checkpoints in Hyper-V](#)

Remotely manage Hyper-V hosts with Hyper-V Manager

Article • 07/29/2021 • 6 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows 10, Windows 8.1

This article lists the supported combinations of Hyper-V hosts and Hyper-V Manager versions and describes how to connect to remote and local Hyper-V hosts so you can manage them.

Hyper-V Manager lets you manage a small number of Hyper-V hosts, both remote and local. It's installed when you install the Hyper-V Management Tools, which you can do either through a full Hyper-V installation or a tools-only installation. Doing a tools-only installation means you can use the tools on computers that don't meet the hardware requirements to host Hyper-V. For details about hardware for Hyper-V hosts, see [System requirements](#).

If Hyper-V Manager isn't installed, see the [instructions](#) below.

Supported combinations of Hyper-V Manager and Hyper-V host versions

In some cases you can use a different version of Hyper-V Manager than the Hyper-V version on the host, as shown in the table. When you do this, Hyper-V Manager provides the features available for the version of Hyper-V on the host you're managing. For example, if you use the version of Hyper-V Manager in Windows Server 2012 R2 to remotely manage a host running Hyper-V in Windows Server 2012, you won't be able to use features available in Windows Server 2012 R2 on that Hyper-V host.

The following table shows which versions of a Hyper-V host you can manage from a particular version of Hyper-V Manager. Only supported operating system versions are listed. For details about the support status of a particular operating system version, use the [Search product lifecycle](#) button on the [Microsoft Lifecycle Policy](#) page. In general, older versions of Hyper-V Manager can only manage a Hyper-V host running the same version or the comparable Windows Server version.

Hyper-V Manager version	Hyper-V host version
-------------------------	----------------------

Hyper-V Manager version	Hyper-V host version
Windows Server 2016, Windows 10	<ul style="list-style-type: none"> - Windows Server 2016—all editions and installation options, including Nano Server, and corresponding version of Hyper-V Server - Windows Server 2012 R2—all editions and installation options, and corresponding version of Hyper-V Server - Windows Server 2012—all editions and installation options, and corresponding version of Hyper-V Server - Windows 10 - Windows 8.1
Windows Server 2012 R2, Windows 8.1	<ul style="list-style-type: none"> - Windows Server 2012 R2—all editions and installation options, and corresponding version of Hyper-V Server - Windows Server 2012—all editions and installation options, and corresponding version of Hyper-V Server - Windows 8.1
Windows Server 2012	<ul style="list-style-type: none"> - Windows Server 2012—all editions and installation options, and corresponding version of Hyper-V Server
Windows Server 2008 R2 Service Pack 1, Windows 7 Service Pack 1	<ul style="list-style-type: none"> - Windows Server 2008 R2—all editions and installation options, and corresponding version of Hyper-V Server
Windows Server 2008, Windows Vista Service Pack 2	<ul style="list-style-type: none"> - Windows Server 2008—all editions and installation options, and corresponding version of Hyper-V Server

 **Note**

Service pack support ended for Windows 8 on January 12, 2016. For more information, see the [Windows 8.1 FAQ](#).

Connect to a Hyper-V host

To connect to a Hyper-V host from Hyper-V Manager, right-click **Hyper-V Manager** in the left pane, and then click **Connect to Server**.

Manage Hyper-V on a local computer

Hyper-V Manager doesn't list any computers that host Hyper-V until you add the computer, including a local computer. To do this:

1. In the left pane, right-click **Hyper-V Manager**.
2. Click **Connect to Server**.

3. From **Select Computer**, click **Local computer** and then click **OK**.

If you can't connect:

- It's possible that only the Hyper-V tools are installed. To check that Hyper-V platform is installed, look for the Virtual Machine Management service. /(Open the Services desktop app: click **Start**, click the **Start Search** box, type **services.msc**, and then press **Enter**. If the Virtual Machine Management service isn't listed, install the Hyper-V platform by following the instructions in [Install Hyper-V](#).
- Check that your hardware meets the requirements. See [System requirements](#).
- Check that your user account belongs to the Administrators group or the Hyper-V Administrators group.

Manage Hyper-V hosts remotely

To manage remote Hyper-V hosts, enable remote management on both the local computer and remote host.

On Windows Server, open **Server Manager > Local Server > Remote management** and then click **Allow remote connections to this computer**.

Or, from either operating system, open Windows PowerShell as Administrator and run:

```
PowerShell
```

```
Enable-PSRemoting
```

Connect to hosts in the same domain

For Windows 8.1 and earlier, remote management works only when the host is in the same domain and your local user account is also on the remote host.

To add a remote Hyper-V host to Hyper-V Manager, select **Another computer** in the **Select Computer** dialogue box and type the remote host's hostname, NetBIOS name, or fully qualified domain name (FQDN).

Hyper-V Manager in Windows Server 2016 and Windows 10 offers more types of remote connection than previous versions, described in the following sections.

Connect to a Windows Server 2016 or Windows 10 remote host as a different user

This lets you connect to the Hyper-V host when you're not running on the local computer as a user that's a member of either the Hyper-V Administrators group or the Administrators group on the Hyper-V host. To do this:

1. In the left pane, right-click **Hyper-V Manager**.
2. Click **Connect to Server**.
3. Select **Connect as another user** in the **Select Computer** dialogue box.
4. Select **Set User**.

 **Note**

This will only work for Windows Server 2016 or Windows 10 **remote** hosts.

Connect to a Windows Server 2016 or Windows 10 remote host using IP address

To do this:

1. In the left pane, right-click **Hyper-V Manager**.
2. Click **Connect to Server**.
3. Type the IP address into the **Another Computer** text field.

 **Note**

This will only work for Windows Server 2016 or Windows 10 **remote** hosts.

Connect to a Windows Server 2016 or Windows 10 remote host outside your domain, or with no domain

To do this:

1. On the Hyper-V host to be managed, open a Windows PowerShell session as Administrator.
2. Create the necessary firewall rules for private network zones:

PowerShell

[Enable-PSRemoting](#)

3. To allow remote access on public zones, enable firewall rules for CredSSP and WinRM:

```
PowerShell
```

```
Enable-WSManCredSSP -Role server
```

For details, see [Enable-PSRemoting](#) and [Enable-WSManCredSSP](#).

Next, configure the computer you'll use to manage the Hyper-V host.

1. Open a Windows PowerShell session as Administrator.
2. Run these commands:

```
PowerShell
```

```
Set-Item WSMan:\localhost\Client\TrustedHosts -Value "fqdn-of-hyper-v-host"
```

```
PowerShell
```

```
Enable-WSManCredSSP -Role client -DelegateComputer "fqdn-of-hyper-v-host"
```

3. You might also need to configure the following group policy:

- Computer Configuration > Administrative Templates > System > Credentials Delegation > Allow delegating fresh credentials with NTLM-only server authentication
- Click **Enable** and add *wsman/fqdn-of-hyper-v-host*.

4. Open **Hyper-V Manager**.

5. In the left pane, right-click **Hyper-V Manager**.

6. Click **Connect to Server**.

 **Note**

This will only work for Windows Server 2016 or Windows 10 **remote** hosts.

For cmdlet details, see [Set-Item](#) and [Enable-WSManCredSSP](#).

Install Hyper-V Manager

To use a UI tool, choose the one appropriate for the operating system on the computer where you'll run Hyper-V Manager:

On Windows Server, open Server Manager > **Manage** > **Add roles and features**. Move to the **Features** page and expand **Remote server administration tools** > **Role administration tools** > **Hyper-V management tools**.

On Windows, Hyper-V Manager is available on [any Windows operating system that includes Hyper-V](#).

1. On the Windows desktop, click the Start button and begin typing **Programs and features**.
2. In search results, click **Programs and Features**.
3. In the left pane, click **Turn Windows features on or off**.
4. Expand the Hyper-V folder, and click **Hyper-V Management Tools**.
5. To install Hyper-V Manager, click **Hyper-V Management Tools**. If you want to also install the Hyper-V module, click that option.

To use Windows PowerShell, run the following command as Administrator:

```
PowerShell
```

```
add-windowsfeature rsat-hyper-v-tools
```

Additional References

[Install Hyper-V](#)

Hyper-V Host CPU Resource Management

Article • 09/17/2020 • 4 minutes to read

Hyper-V host CPU resource controls introduced in Windows Server 2016 or later allow Hyper-V administrators to better manage and allocate host server CPU resources between the "root", or management partition, and guest VMs. Using these controls, administrators can dedicate a subset of the processors of a host system to the root partition. This can segregate the work done in a Hyper-V host from the workloads running in guest virtual machines by running them on separate subsets of the system processors.

For details about hardware for Hyper-V hosts, see [Windows 10 Hyper-V System Requirements](#).

Background

Before setting controls for Hyper-V host CPU resources, it's helpful to review the basics of the Hyper-V architecture. You can find a general summary in the [Hyper-V Architecture](#) section. These are important concepts for this article:

- Hyper-V creates and manages virtual machine partitions, across which compute resources are allocated and shared, under control of the hypervisor. Partitions provide strong isolation boundaries between all guest virtual machines, and between guest VMs and the root partition.
- The root partition is itself a virtual machine partition, although it has unique properties and much greater privileges than guest virtual machines. The root partition provides the management services that control all guest virtual machines, provides virtual device support for guests, and manages all device I/O for guest virtual machines. Microsoft strongly recommends not running any application workloads in a host partition.
- Each virtual processor (VP) of the root partition is mapped 1:1 to an underlying logical processor (LP). A host VP will always run on the same underlying LP – there is no migration of the root partition's VPs.
- By default, the LPs on which host VPs run can also run guest VPs.
- A guest VP may be scheduled by the hypervisor to run on any available logical processor. While the hypervisor scheduler takes care to consider temporal cache

locality, NUMA topology, and many other factors when scheduling a guest VP, ultimately the VP could be scheduled on any host LP.

The Minimum Root, or "Minroot" Configuration

Early versions of Hyper-V had an architectural maximum limit of 64 VPs per partition. This applied to both the root and guest partitions. As systems with more than 64 logical processors appeared on high-end servers, Hyper-V also evolved its host scale limits to support these larger systems, at one point supporting a host with up to 320 LPs. However, breaking the 64 VP per partition limit at that time presented several challenges and introduced complexities that made supporting more than 64 VPs per partition prohibitive. To address this, Hyper-V limited the number of VPs given to the root partition to 64, even if the underlying machine had many more logical processors available. The hypervisor would continue to utilize all available LPs for running guest VPs, but artificially capped the root partition at 64. This configuration became known as the "minimum root", or "minroot" configuration. Performance testing confirmed that, even on large scale systems with more than 64 LPs, the root did not need more than 64 root VPs to provide sufficient support to a large number of guest VMs and guest VPs – in fact, much less than 64 root VPs was often adequate, depending of course on the number and size of the guest VMs, the specific workloads being run, etc.

This "minroot" concept continues to be utilized today. In fact, even as Windows Server 2016 Hyper-V increased its maximum architectural support limit for host LPs to 512 LPs, the root partition will still be limited to a maximum of 320 LPs.

Using Minroot to Constrain and Isolate Host Compute Resources

With the high default threshold of 320 LPs in Windows Server 2016 Hyper-V, the minroot configuration will only be utilized on the very largest server systems. However, this capability can be configured to a much lower threshold by the Hyper-V host administrator, and thus leveraged to greatly restrict the amount of host CPU resources available to the root partition. The specific number of root LPs to utilize must of course be chosen carefully to support the maximum demands of the VMs and workloads allocated to the host. However, reasonable values for the number of host LPs can be determined through careful assessment and monitoring of production workloads, and validated in non-production environments before broad deployment.

Enabling and Configuring Minroot

The minroot configuration is controlled via hypervisor BCD entries. To enable minroot, from a cmd prompt with administrator privileges:

```
bcDEDIT /SET hypervisorRootProc n
```

Where n is the number of root VPs.

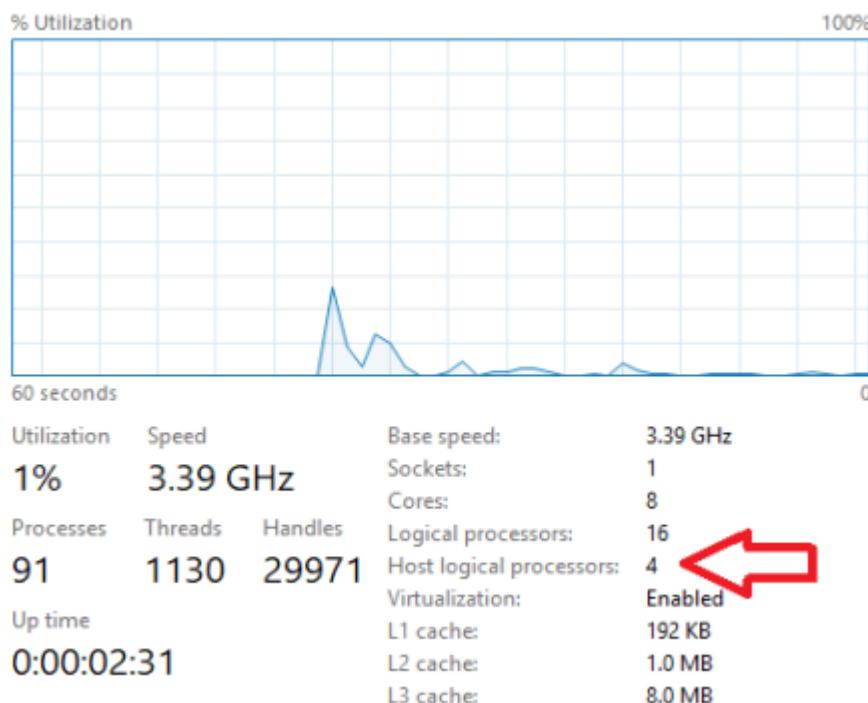
The system must be rebooted, and the new number of root processors will persist for the lifetime of the OS boot. The minroot configuration cannot be changed dynamically at runtime.

If there are multiple NUMA nodes, each node will get $n/\text{NumaNodeCount}$ processors.

Note that with multiple NUMA nodes, you must ensure the VM's topology is such that there are enough free LPs (i.e., LPs without root VPs) on each NUMA node to run the corresponding VM's NUMA node VPs.

Verifying the Minroot Configuration

You can verify the host's minroot configuration using Task Manager, as shown below.



When Minroot is active, Task Manager will display the number of logical processors currently allotted to the host, in addition to the total number of logical processors in the

system.

Virtual Machine Resource Controls

Article • 07/29/2021 • 11 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article describes Hyper-V resource and isolation controls for virtual machines. These capabilities, which we'll refer to as Virtual Machine CPU Groups, or just "CPU groups", were introduced in Windows Server 2016. CPU groups allow Hyper-V administrators to better manage and allocate the host's CPU resources across guest virtual machines. Using CPU groups, Hyper-V administrators can:

- Create groups of virtual machines, with each group having different allocations of the virtualization host's total CPU resources, shared across the entire group. This allows the host administrator to implement classes of service for different types of VMs.
- Set CPU resource limits to specific groups. This "group cap" sets the upper bound for host CPU resources that the entire group may consume, effectively enforcing the desired class of service for that group.
- Constrain a CPU group to run only on a specific set of the host system's processors. This can be used to isolate VMs belonging to different CPU groups from each other.

Managing CPU Groups

CPU groups are managed through the Hyper-V Host Compute Service, or HCS. A great description of the HCS, its genesis, links to the HCS APIs, and more is available on the Microsoft Virtualization team's blog in the posting [Introducing the Host Compute Service \(HCS\)](#).

Note

Only the HCS may be used to create and manage CPU groups; the Hyper-V Manager applet, WMI and PowerShell management interfaces don't support CPU groups.

Microsoft provides a command line utility, cpugroups.exe, on the [Microsoft Download Center](#) which uses the HCS interface to manage CPU groups. This utility can also

display the CPU topology of a host.

How CPU Groups Work

Allocation of host compute resources across CPU groups is enforced by the Hyper-V hypervisor, using a computed CPU group cap. The CPU group cap is a fraction of the total CPU capacity for a CPU group. The value of the group cap depends on the group class, or priority level assigned. The computed group cap can be thought of as "a number of LP's worth of CPU time". This group budget is shared, so if only a single VM were active, it could use the entire group's CPU allocation for itself.

The CPU group cap is calculated as $G = n \times C$, where:

- G is the amount of host LP we'd like to assign to the group
- n is the total number of logical processors (LPs) in the group
- C is the maximum CPU allocation — that is, the class of service desired for the group, expressed as a percentage of the system's total compute capacity

For example, consider a CPU group configured with 4 logical processors (LPs), and a cap of 50%.

- $G = n * C$
- $G = 4 * 50\%$
- $G = 2$ LP's worth of CPU time for the entire group

In this example, the CPU group G is allocated 2 LP's worth of CPU time.

Note that the group cap applies regardless of the number of virtual machines or virtual processors bound to the group, and regardless of the state (e.g., shutdown or started) of the virtual machines assigned to the CPU group. Therefore, each VM bound to the same CPU group will receive a fraction of the group's total CPU allocation, and this will change with the number of VMs bound to the CPU group. Therefore, as VMs are bound or unbound from a CPU group, the overall CPU group cap must be readjusted and set to maintain the resulting per-VM cap desired. The VM host administrator or virtualization management software layer is responsible for managing group caps as necessary to achieve the desired per-VM CPU resource allocation.

Example Classes of Service

Let's look at some simple examples. To start with, assume the Hyper-V host administrator would like to support two tiers of service for guest VMs:

1. A low-end "C" tier. We'll give this tier 10% of the entire host's compute resources.
2. A mid-range "B" tier. This tier is allocated 50% of the entire host's compute resources.

At this point in our example we'll assert that no other CPU resource controls are in use, such as individual VM caps, weights, and reserves. However, individual VM caps are important, as we'll see a bit later.

For simplicity's sake, let's assume each VM has 1 VP, and that our host has 8 LPs. We'll start with an empty host.

To create the "B" tier, the host administrator sets the group cap to 50%:

- $G = n * C$
- $G = 8 * 50\%$
- $G = 4$ LP's worth of CPU time for the entire group

The host administrator adds a single "B" tier VM. At this point, our "B" tier VM can use at most 50% worth of the host's CPU, or the equivalent of 4 LPs in our example system.

Now, the admin adds a second "Tier B" VM. The CPU group's allocation—is divided evenly among all the VMs. We've got a total of 2 VMs in Group B, so each VM now gets half of Group B's total of 50%, 25% each, or the equivalent of 2 LPs worth of compute time.

Setting CPU Caps on Individual VMs

In addition to the group cap, each VM can also have an individual "VM cap". Per-VM CPU resource controls, including a CPU cap, weight, and reserve, have been a part of Hyper-V since its introduction. When combined with a group cap, a VM cap specifies the maximum amount of CPU that each VP can get, even if the group has CPU resources available.

For example, the host administrator might want to place a 10% VM cap on "C" VMs. That way, even if most "C" VPs are idle, each VP could never get more than 10%. Without a VM cap, "C" VMs could opportunistically achieve performance beyond levels allowed by their tier.

Isolating VM Groups to Specific Host Processors

Hyper-V host administrators may also want the ability to dedicate compute resources to a VM. For example, imagine the administrator wanted to offer a premium "A" VM that has a class cap of 100%. These premium VMs also require the lowest scheduling latency and jitter possible; that is, they may not be de-scheduled by any other VM. To achieve this separation, a CPU group can also be configured with a specific LP affinity mapping.

For example, to fit an "A" VM on the host in our example, the administrator would create a new CPU group, and set the group's processor affinity to a subset of the host's LPs. Groups B and C would be affinitized to the remaining LPs. The administrator could create a single VM in Group A, which would then have exclusive access to all LPs in Group A, while the presumably lower tier groups B and C would share the remaining LPs.

Segregating Root VPs from Guest VPs

By default, Hyper-V will create a root VP on each underlying physical LP. These root VPs are strictly mapped 1:1 with the system LPs, and do not migrate — that is, each root VP will always execute on the same physical LP. Guest VPs may be run on any available LP, and will share execution with root VPs.

However, it may be desirable to completely separate root VP activity from guest VPs. Consider our example above where we implement a premium "A" tier VM. To ensure our "A" VM's VPs have the lowest possible latency and "jitter", or scheduling variation, we'd like to run them on a dedicated set of LPs and ensure the root does not run on these LPs.

This can be accomplished using a combination of the "minroot" configuration, which limits the host OS partition to running on a subset of the total system logical processors, along with one or more affinitized CPU groups.

The virtualization host can be configured to restrict the host partition to specific LPs, with one or more CPU groups affinitized to the remaining LPs. In this manner, the root and guest partitions can run on dedicated CPU resources, and completely isolated, with no CPU sharing.

For more information about the "minroot" configuration, see [Hyper-V Host CPU Resource Management](#).

Using the CpuGroups Tool

Let's look at some examples of how to use the CpuGroups tool.

ⓘ Note

Command line parameters for the CpuGroups tool are passed using only spaces as delimiters. No '/' or '-' characters should precede the desired command line switch.

Discovering the CPU Topology

Executing CpuGroups with the GetCpuTopology returns information about the current system, as shown below, including the LP Index, the NUMA node to which the LP belongs, the Package and Core IDs, and the ROOT VP index.

The following example shows a system with 2 CPU sockets and NUMA nodes, a total of 32 LPs, and multi threading enabled, and configured to enable Minroot with 8 root VPs, 4 from each NUMA node. The LPs that have root VPs have a RootVpIndex ≥ 0 ; LPs with a RootVpIndex of -1 are not available to the root partition, but are still managed by the hypervisor and will run guest VPs as allowed by other configuration settings.

Console				
C:\vm\tools>CpuGroups.exe GetCpuTopology				
LpIndex	NodeNumber	PackageId	CoreId	RootVpIndex

0	0	0	0	0
1	0	0	0	1
2	0	0	1	2
3	0	0	1	3
4	0	0	2	-1
5	0	0	2	-1
6	0	0	3	-1
7	0	0	3	-1
8	0	0	4	-1
9	0	0	4	-1
10	0	0	5	-1
11	0	0	5	-1
12	0	0	6	-1
13	0	0	6	-1
14	0	0	7	-1
15	0	0	7	-1
16	1	1	16	4
17	1	1	16	5
18	1	1	17	6
19	1	1	17	7
20	1	1	18	-1
21	1	1	18	-1
22	1	1	19	-1
23	1	1	19	-1
24	1	1	20	-1
25	1	1	20	-1

26	1	1	21	-1
27	1	1	21	-1
28	1	1	22	-1
29	1	1	22	-1
30	1	1	23	-1
31	1	1	23	-1

Example 2 – Print all CPU groups on the host

Here, we'll list all CPU groups on the current host, their GroupId, the group's CPU cap, and the indices of LPs assigned to that group.

Note that valid CPU cap values are in the range [0, 65536], and these values express the group cap in percent (e.g., 32768 = 50%).

Console

```
C:\vm\tools>CpuGroups.exe GetGroups
```

CpuGroupId	CpuCap	LpIndexes
36AB08CB-3A76-4B38-992E-000000000002	32768	4,5,6,7,8,9,10,11,20,21,22,23
36AB08CB-3A76-4B38-992E-000000000003	65536	12,13,14,15
36AB08CB-3A76-4B38-992E-000000000004	65536	24,25,26,27,28,29,30,31

Example 3 – Print a single CPU group

In this example, we'll query a single CPU Group using the GroupId as a filter.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups /GroupId:36AB08CB-3A76-4B38-992E-000000000003
```

CpuGroupId	CpuCap	LpIndexes
36AB08CB-3A76-4B38-992E-000000000003	65536	12,13,14,15

Example 4 – Create a new CPU group

Here, we'll create a new CPU group, specifying the Group ID and the set of LPs to assign to the group.

Console

```
C:\vm\tools>CpuGroups.exe CreateGroup /GroupId:36AB08CB-3A76-4B38-992E-000000000001 /GroupAffinity:0,1,16,17
```

Now display our newly added group.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups
CpuGroupId          CpuCap LpIndexes
-----
36AB08CB-3A76-4B38-992E-000000000001 65536 0,1,16,17
36AB08CB-3A76-4B38-992E-000000000002 32768 4,5,6,7,8,9,10,11,20,21,22,23
36AB08CB-3A76-4B38-992E-000000000003 65536 12,13,14,15
36AB08CB-3A76-4B38-992E-000000000004 65536 24,25,26,27,28,29,30,31
```

Example 5 – Set the CPU group cap to 50%

Here, we'll set the CPU group cap to 50%.

Console

```
C:\vm\tools>CpuGroups.exe SetGroupProperty /GroupId:36AB08CB-3A76-4B38-992E-000000000001 /CpuCap:32768
```

Now let's confirm our setting by displaying the group we just updated.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups /GroupId:36AB08CB-3A76-4B38-992E-000000000001
CpuGroupId          CpuCap LpIndexes
-----
36AB08CB-3A76-4B38-992E-000000000001 32768 0,1,16,17
```

Example 6 – Print CPU group ids for all VMs on the host

Console

```
C:\vm\tools>CpuGroups.exe GetVmGroup
```

VmName	VmId
CpuGroupId	
-----	-----
-----	-----

```
G2 4ABCFC2F-6C22-498C-BB38-7151CE678758 36ab08cb-3a76-4b38-992e-  
000000000002  
P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC 36ab08cb-3a76-4b38-992e-  
000000000003  
P2 A593D93A-3A5F-48AB-8862-A4350E3459E8 36ab08cb-3a76-4b38-992e-  
000000000004  
G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200 36ab08cb-3a76-4b38-992e-  
000000000002  
G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 36ab08cb-3a76-4b38-992e-  
000000000002
```

Example 7 – Unbind a VM from the CPU group

To remove a VM from a CPU group, set to VM's CpuGroupId to the NULL GUID. This unbinds the VM from the CPU group.

```
Console  
  
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:00000000-0000-0000-  
0000-000000000000  
  
C:\vm\tools>CpuGroups.exe GetVmGroup  
VmName VmId  
CpuGroupId  
-----  
----  
G2 4ABCFC2F-6C22-498C-BB38-7151CE678758 36ab08cb-3a76-4b38-992e-  
000000000002  
P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC 36ab08cb-3a76-4b38-992e-  
000000000003  
P2 A593D93A-3A5F-48AB-8862-A4350E3459E8 36ab08cb-3a76-4b38-992e-  
000000000004  
G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200 36ab08cb-3a76-4b38-992e-  
000000000002  
G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 00000000-0000-0000-0000-  
000000000000
```

Example 8 – Bind a VM to an existing CPU group

Here, we'll add a VM to an existing CPU group. Note that the VM must not be bound to any existing CPU group, or setting CPU group id will fail.

```
Console  
  
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:36AB08CB-3A76-4B38-  
992E-000000000001
```

Now, confirm the VM G1 is in the desired CPU group.

Console

```
C:\vm\tools>CpuGroups.exe GetVmGroup
VmName                      VmId
CpuGroupId
-----
-----
G2 4ABCFC2F-6C22-498C-BB38-7151CE678758 36ab08cb-3a76-4b38-992e-
000000000002
P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC 36ab08cb-3a76-4b38-992e-
000000000003
P2 A593D93A-3A5F-48AB-8862-A4350E3459E8 36ab08cb-3a76-4b38-992e-
000000000004
G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200 36ab08cb-3a76-4b38-992e-
000000000002
G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 36AB08CB-3A76-4B38-992E-
000000000001
```

Example 9 – Print all VMs grouped by CPU group id

Console

```
C:\vm\tools>CpuGroups.exe GetGroupVms
CpuGroupId                  VmName
VmId
-----
-----
36AB08CB-3A76-4B38-992E-000000000001      G1 F699B50F-86F2-4E48-8BA5-
EB06883C1FDC
36ab08cb-3a76-4b38-992e-000000000002      G2 4ABCFC2F-6C22-498C-BB38-
7151CE678758
36ab08cb-3a76-4b38-992e-000000000002      G3 B0F3FCD5-FECF-4A21-A4A2-
DE4102787200
36ab08cb-3a76-4b38-992e-000000000003      P1 973B9426-0711-4742-AD3B-
D8C39D6A0DEC
36ab08cb-3a76-4b38-992e-000000000004      P2 A593D93A-3A5F-48AB-8862-
A4350E3459E8
```

Example 10 – Print all VMs for a single CPU group

Console

```
C:\vm\tools>CpuGroups.exe GetGroupVms /GroupId:36ab08cb-3a76-4b38-992e-
000000000002

CpuGroupId                  VmName
VmId
```

```
-----  
36ab08cb-3a76-4b38-992e-000000000002      G2 4ABCFC2F-6C22-498C-BB38-  
7151CE678758  
36ab08cb-3a76-4b38-992e-000000000002      G3 B0F3FCD5-FECF-4A21-A4A2-  
DE4102787200
```

Example 11 – Attempting to delete a non-empty CPU Group

Only empty CPU groups—that is, CPU groups with no bound VMs—can be deleted. Attempting to delete a non-empty CPU group will fail.

Console

```
C:\vm\tools>CpuGroups.exe DeleteGroup /GroupId:36ab08cb-3a76-4b38-992e-  
000000000001  
(null)  
Failed with error 0xc0350070
```

Example 12 – Unbind the only VM from a CPU group and delete the group

In this example, we'll use several commands to examine a CPU group, remove the single VM belonging to that group, then delete the group.

First, let's enumerate the VMs in our group.

Console

```
C:\vm\tools>CpuGroups.exe GetGroupVms /GroupId:36AB08CB-3A76-4B38-992E-  
000000000001  
CpuGroupId          VmName  
VmId  
-----  
-----  
36AB08CB-3A76-4B38-992E-000000000001      G1 F699B50F-86F2-4E48-8BA5-  
EB06883C1FDC
```

We see that only a single VM, named G1, belongs to this group. Let's remove the G1 VM from our group by setting the VM's group ID to NULL.

Console

```
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:00000000-0000-0000-0000-000000000000
```

And verify our change...

Console

```
C:\vm\tools>CpuGroups.exe GetVmGroup /VmName:g1
VmName                      VmId
CpuGroupId
-----
-----
G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC 00000000-0000-0000-0000-000000000000
```

Now that the group is empty, we can safely delete it.

Console

```
C:\vm\tools>CpuGroups.exe DeleteGroup /GroupId:36ab08cb-3a76-4b38-992e-000000000001
```

And confirm our group is gone.

Console

```
C:\vm\tools>CpuGroups.exe GetGroups
CpuGroupId                  CpuCap                  LpIndexes
-----
36AB08CB-3A76-4B38-992E-000000000002 32768 4,5,6,7,8,9,10,11,20,21,22,23
36AB08CB-3A76-4B38-992E-000000000003 65536 12,13,14,15
36AB08CB-3A76-4B38-992E-000000000004 65536 24,25,26,27,28,29,30,31
```

Example 13 – Bind a VM back to its original CPU group

Console

```
C:\vm\tools>CpuGroups.exe SetVmGroup /VmName:g1 /GroupId:36AB08CB-3A76-4B38-992E-000000000002
```

```
C:\vm\tools>CpuGroups.exe GetGroupVms
```

```
CpuGroupId VmName VmId
-----
-----
```

```
36ab08cb-3a76-4b38-992e-000000000002 G2 4ABCFC2F-6C22-498C-BB38-7151CE678758
36ab08cb-3a76-4b38-992e-000000000002 G3 B0F3FCD5-FECF-4A21-A4A2-DE4102787200
```

36AB08CB-3A76-4B38-992E-000000000002 G1 F699B50F-86F2-4E48-8BA5-EB06883C1FDC
36ab08cb-3a76-4b38-992e-000000000003 P1 973B9426-0711-4742-AD3B-D8C39D6A0DEC
36ab08cb-3a76-4b38-992e-000000000004 P2 A593D93A-3A5F-48AB-8862-A4350E3459E8

Managing Hyper-V hypervisor scheduler types

Article • 12/23/2021 • 10 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Windows Server, version 1709, Windows Server, version 1803, Windows Server 2019

This article describes new modes of virtual processor scheduling logic first introduced in Windows Server 2016. These modes, or scheduler types, determine how the Hyper-V hypervisor allocates and manages work across guest virtual processors. A Hyper-V host administrator can select hypervisor scheduler types that are best suited for the guest virtual machines (VMs) and configure the VMs to take advantage of the scheduling logic.

ⓘ Note

Updates are required to use the hypervisor scheduler features described in this document. For details, see [Required updates](#).

Background

Before discussing the logic and controls behind Hyper-V virtual processor scheduling, it's helpful to review the basic concepts covered in this article.

Understanding SMT

Simultaneous multithreading, or SMT, is a technique utilized in modern processor designs that allows the processor's resources to be shared by separate, independent execution threads. SMT generally offers a modest performance boost to most workloads by parallelizing computations when possible, increasing instruction throughput, though no performance gain or even a slight loss in performance may occur when contention between threads for shared processor resources occurs. Processors supporting SMT are available from both Intel and AMD. Intel refers to their SMT offerings as Intel Hyper Threading Technology, or Intel HT.

For the purposes of this article, the descriptions of SMT and how it is utilized by Hyper-V apply equally to both Intel and AMD systems.

- For more information on Intel HT Technology, refer to [Intel Hyper-Threading Technology](#)
- For more information on AMD SMT, refer to [The "Zen" Core Architecture](#)

Understanding how Hyper-V virtualizes processors

Before considering hypervisor scheduler types, it's also helpful to understand the Hyper-V architecture. You can find a general summary in [Hyper-V Technology Overview](#). These are important concepts for this article:

- Hyper-V creates and manages virtual machine partitions, across which compute resources are allocated and shared, under control of the hypervisor. Partitions provide strong isolation boundaries between all guest virtual machines, and between guest VMs and the root partition.
- The root partition is itself a virtual machine partition, although it has unique properties and much greater privileges than guest virtual machines. The root partition provides the management services that control all guest virtual machines, provides virtual device support for guests, and manages all device I/O for guest virtual machines. Microsoft strongly recommends not running any application workloads in the root partition.
- Each virtual processor (VP) of the root partition is mapped 1:1 to an underlying logical processor (LP). A host VP always runs on the same underlying LP – there is no migration of the root partition's VPs.
- By default, the LPs on which host VPs run can also run guest VPs.
- A guest VP may be scheduled by the hypervisor to run on any available logical processor. While the hypervisor scheduler takes care to consider temporal cache locality, NUMA topology, and many other factors when scheduling a guest VP, ultimately the VP could be scheduled on any host LP.

Hypervisor scheduler types

Starting with Windows Server 2016, the Hyper-V hypervisor supports several modes of scheduler logic, which determine how the hypervisor schedules virtual processors on the underlying logical processors. These scheduler types are:

The classic scheduler

The classic scheduler has been the default for all versions of the Windows Hyper-V hypervisor since its inception, including Windows Server 2016 Hyper-V. The classic scheduler provides a fair share, preemptive round-robin scheduling model for guest virtual processors.

The classic scheduler type is the most appropriate for the vast majority of traditional Hyper-V uses – for private clouds, hosting providers, and so on. The performance characteristics are well understood and are best optimized to support a wide range of virtualization scenarios, such as over-subscription of VPs to LPs, running many heterogeneous VMs and workloads simultaneously, running larger scale high performance VMs, supporting the full feature set of Hyper-V without restrictions, and more.

The core scheduler

The hypervisor core scheduler is a new alternative to the classic scheduler logic, introduced in Windows Server 2016 and Windows 10 version 1607. The core scheduler offers a strong security boundary for guest workload isolation, and reduced performance variability for workloads inside of VMs that are running on an SMT-enabled virtualization host. The core scheduler allows running both SMT and non-SMT virtual machines simultaneously on the same SMT-enabled virtualization host.

The core scheduler utilizes the virtualization host's SMT topology, and optionally exposes SMT pairs to guest virtual machines, and schedules groups of guest virtual processors from the same virtual machine onto groups of SMT logical processors. This is done symmetrically so that if LPs are in groups of two, VPs are scheduled in groups of two, and a core is never shared between VMs. When the VP is scheduled for a virtual machine without SMT enabled, that VP consumes the entire core when it runs.

The overall result of the core scheduler is that:

- Guest VPs are constrained to run on underlying physical core pairs, isolating a VM to processor core boundaries, thus reducing vulnerability to side-channel snooping attacks from malicious VMs.
- Variability in throughput is significantly reduced.
- Performance is potentially reduced, because if only one of a group of VPs can run, only one of the instruction streams in the core executes while the other is left idle.

- The OS and applications running in the guest virtual machine can utilize SMT behavior and programming interfaces (APIs) to control and distribute work across SMT threads, just as they would when run non-virtualized.
- A strong security boundary for guest workload isolation - Guest VPs are constrained to run on underlying physical core pairs, reducing vulnerability to side-channel snooping attacks.

The core scheduler is used by default starting in Windows Server 2019. On Windows Server 2016, the core scheduler is optional and must be explicitly enabled by the Hyper-V host administrator, and the classic scheduler is the default.

Core scheduler behavior with host SMT disabled

If the hypervisor is configured to use the core scheduler type but the SMT capability is disabled or not present on the virtualization host, then the hypervisor uses the classic scheduler behavior, regardless of the hypervisor scheduler type setting.

The root scheduler

The root scheduler was introduced with Windows 10 version 1803. When the root scheduler type is enabled, the hypervisor cedes control of work scheduling to the root partition. The NT scheduler in the root partition's OS instance manages all aspects of scheduling work to system LPs.

The root scheduler addresses the unique requirements inherent with supporting a utility partition to provide strong workload isolation, as used with Windows Defender Application Guard (WDAG). In this scenario, leaving scheduling responsibilities to the root OS offers several advantages. For example, CPU resource controls applicable to container scenarios may be used with the utility partition, simplifying management and deployment. In addition, the root OS scheduler can readily gather metrics about workload CPU utilization inside the container and use this data as input to the same scheduling policy applicable to all other workloads in the system. These same metrics also help clearly attribute work done in an application container to the host system. Tracking these metrics is more difficult with traditional virtual machine workloads, where some work on all running VM's behalf takes place in the root partition.

Root scheduler use on client systems

Starting with Windows 10 version 1803, the root scheduler is used by default on client systems only, where the hypervisor may be enabled in support of virtualization-based

security and WDAG workload isolation, and for proper operation of future systems with heterogeneous core architectures. This is the only supported hypervisor scheduler configuration for client systems. Administrators should not attempt to override the default hypervisor scheduler type on Windows 10 client systems.

Virtual Machine CPU resource controls and the root scheduler

The virtual machine processor resource controls provided by Hyper-V are not supported when the hypervisor root scheduler is enabled as the root operating system's scheduler logic is managing host resources on a global basis and does not have knowledge of a VM's specific configuration settings. The Hyper-V per-VM processor resource controls, such as caps, weights, and reserves, are only applicable where the hypervisor directly controls VP scheduling, such as with the classic and core scheduler types.

Root scheduler use on server systems

The root scheduler is not recommended for use with Hyper-V on servers at this time, as its performance characteristics have not yet been fully characterized and tuned to accommodate the wide range of workloads typical of many server virtualization deployments.

Enabling SMT in guest virtual machines

Once the virtualization host's hypervisor is configured to use the core scheduler type, guest virtual machines may be configured to utilize SMT if desired. Exposing the fact that VPs are hyperthreaded to a guest virtual machine allows the scheduler in the guest operating system and workloads running in the VM to detect and utilize the SMT topology in their own work scheduling. On Windows Server 2016, guest SMT is not configured by default and must be explicitly enabled by the Hyper-V host administrator. Starting with Windows Server 2019, new VMs created on the host inherits the host's SMT topology by default. That is, a version 9.0 VM created on a host with 2 SMT threads per core would also see 2 SMT threads per core.

PowerShell must be used to enable SMT in a guest virtual machine; there is no user interface provided in Hyper-V Manager. To enable SMT in a guest virtual machine, open a PowerShell window with sufficient permissions, and type:

```
PowerShell
```

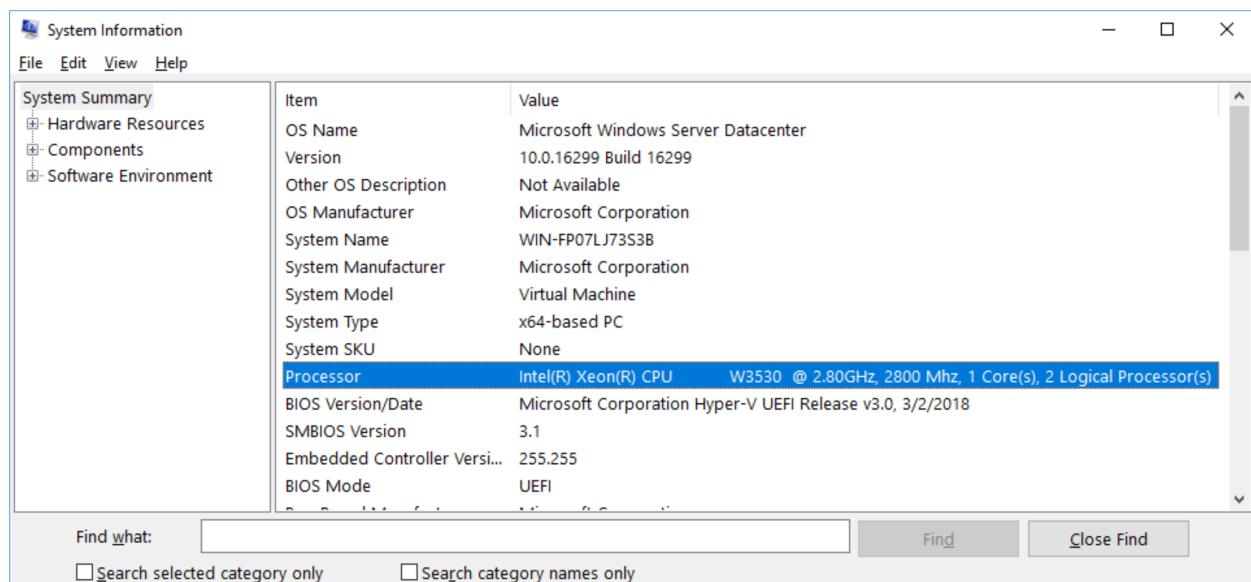
```
Set-VMProcessor -VMName <VMName> -HwThreadCountPerCore <n>
```

Where <n> is the number of SMT threads per core the guest VM sees. Note that <n> = 0 sets the HwThreadCountPerCore value to match the host's SMT thread count per core value.

ⓘ Note

Setting HwThreadCountPerCore = 0 is supported beginning with Windows Server 2019.

Below is an example of System Information taken from the guest operating system running in a virtual machine with 2 virtual processors and SMT enabled. The guest operating system is detecting 2 logical processors belonging to the same core.



Configuring the hypervisor scheduler type on Windows Server 2016 Hyper-V

Windows Server 2016 Hyper-V uses the classic hypervisor scheduler model by default. The hypervisor can be optionally configured to use the core scheduler, to increase security by restricting guest VPs to run on corresponding physical SMT pairs, and to support the use of virtual machines with SMT scheduling for their guest VPs.

ⓘ Note

Microsoft recommends that all customers running Windows Server 2016 Hyper-V select the core scheduler to ensure their virtualization hosts are optimally protected against potentially malicious guest VMs.

Windows Server 2019 Hyper-V defaults to using the core scheduler

To help ensure Hyper-V hosts are deployed in the optimal security configuration, Windows Server 2019 Hyper-V now uses the core hypervisor scheduler model by default. The host administrator may optionally configure the host to use the legacy classic scheduler. Administrators should carefully read, understand and consider the impacts each scheduler type has on the security and performance of virtualization hosts prior to overriding the scheduler type default settings. See [About Hyper-V hypervisor scheduler type selection](#) for more information.

Required updates

Note

The following updates are required to use the hypervisor scheduler features described in this document. These updates include changes to support the new `hypervisorschedulertype` BCD option, which is necessary for host configuration.

Version	Release	Update Required	KB Article
Windows Server 2016	1607	2018.07 C	KB4338822
Windows Server 2016	1703	2018.07 C	KB4338827
Windows Server 2016	1709	2018.07 C	KB4338817
Windows Server 2019	1804	None	None

Selecting the hypervisor scheduler type on Windows Server

The hypervisor scheduler configuration is controlled via the `hypervisorschedulertype` BCD entry.

To select a scheduler type, open a command prompt with administrator privileges:

```
bcdeedit /set hypervisorschedulertype type
```

Where `type` is one of:

- Classic
- Core
- Root

The system must be rebooted for any changes to the hypervisor scheduler type to take effect.

 **Note**

The hypervisor root scheduler is not supported on Windows Server Hyper-V at this time. Hyper-V administrators should not attempt to configure the root scheduler for use with server virtualization scenarios.

Determining the current scheduler type

You can determine the current hypervisor scheduler type in use by examining the System log in Event Viewer for the most recent hypervisor launch event ID 2, which reports the hypervisor scheduler type configured at hypervisor launch. Hypervisor launch events can be obtained from the Windows Event Viewer, or via PowerShell.

Hypervisor launch event ID 2 denotes the hypervisor scheduler type, where:

- 1 = Classic scheduler, SMT disabled
- 2 = Classic scheduler
- 3 = Core scheduler
- 4 = Root scheduler

 Event Properties - Event 2, Hyper-V-Hypervisor X

General Details

Hypervisor scheduler type is 0x2.

Log Name: System
Source: Hyper-V-Hypervisor **Logged:** 4/10/2018 12:10:33 PM
Event ID: 2 **Task Category:** None
Level: Information **Keywords:** (70368744177664)
User: SYSTEM **Computer:** [REDACTED]
OpCode: Info
More Information: [Event Log Online Help](#)

Copy Close

 Event Viewer - □ ×

File Action View Help

Back Forward New Search Help

System Number of events: 3,179 (!) New events available

Level	Date and Time	Source	Event ID	Task Category
Information	4/10/2018 12:10:36 PM	Kernel-Power	172	(203)
Information	4/10/2018 12:10:35 PM	FilterManager	6	None
Information	4/10/2018 12:10:35 PM	FilterManager	6	None
Information	4/10/2018 12:10:34 PM	Ntfs (Microsoft-Wind...	98	None
Information	4/10/2018 12:10:34 PM	FilterManager	6	None
Information	4/10/2018 12:10:33 PM	IsolatedUserMode	3	None
Information	4/10/2018 12:10:33 PM	Hyper-V-Hypervisor	129	None
Information	4/10/2018 12:10:33 PM	Hyper-V-Hypervisor	2	None
Information	4/10/2018 12:10:33 PM	Hyper-V-Hypervisor	1	None
Information	4/10/2018 12:10:33 PM	Kernel-Boot	25	(32)
...

Event 2, Hyper-V-Hypervisor X

General Details

Hypervisor scheduler type is 0x2.

Log Name: System
Source: Hyper-V-Hypervisor **Logged:** 4/10/2018 12:10:33 PM
Event ID: 2 **Task Category:** None
Level: Information **Keywords:** (70368744177664)
User: SYSTEM **Computer:** [REDACTED]
OpCode: Info
More Information: [Event Log Online Help](#)

Querying the Hyper-V hypervisor scheduler type launch event using PowerShell

To query for hypervisor event ID 2 using PowerShell, enter the following commands from a PowerShell prompt.

```
PowerShell

Get-WinEvent -FilterHashTable @{ProviderName="Microsoft-Windows-Hyper-V-Hypervisor"; ID=2} -MaxEvents 1

Administrator: Windows PowerShell
PS C:\> Get-WinEvent -FilterHashTable @{ProviderName="Microsoft-Windows-Hyper-V-Hypervisor"; ID=2} | select -Last 1
ProviderName: Microsoft-Windows-Hyper-V-Hypervisor
TimeCreated          Id LevelDisplayName Message
-----          -- -----
4/10/2018 12:10:33 PM      2 Information    Hypervisor scheduler type is 0x2.

PS C:\>
```

About Hyper-V hypervisor scheduler type selection

Article • 12/23/2021 • 13 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server, version 1709, Windows Server, version 1803

This document describes important changes to Hyper-V's default and recommended use of hypervisor scheduler types. These changes impact both system security and virtualization performance. Virtualization host administrators should review and understand the changes and implications described in this document, and carefully evaluate the impacts, suggested deployment guidance and risk factors involved to best understand how to deploy and manage Hyper-V hosts in the face of the rapidly changing security landscape.

Important

Currently known side-channel security vulnerabilities sighted in multiple processor architectures could be exploited by a malicious guest VM through the scheduling behavior of the legacy hypervisor classic scheduler type when run on hosts with Simultaneous Multithreading (SMT) enabled. If successfully exploited, a malicious workload could observe data outside its partition boundary. This class of attacks can be mitigated by configuring the Hyper-V hypervisor to utilize the hypervisor core scheduler type and reconfiguring guest VMs. With the core scheduler, the hypervisor restricts a guest VM's VPs to run on the same physical processor core, therefore strongly isolating the VM's ability to access data to the boundaries of the physical core on which it runs. This is a highly effective mitigation against these side-channel attacks, which prevents the VM from observing any artifacts from other partitions, whether the root or another guest partition. Therefore, Microsoft is changing the default and recommended configuration settings for virtualization hosts and guest VMs.

Background

Starting with Windows Server 2016, Hyper-V supports several methods of scheduling and managing virtual processors, referred to as hypervisor scheduler types. A detailed

description of all hypervisor scheduler types can be found in [Understanding and using Hyper-V hypervisor scheduler types](#).

Note

New hypervisor scheduler types were first introduced with Windows Server 2016, and are not available in prior releases. All versions of Hyper-V prior to Windows Server 2016 support only the classic scheduler. Support for the core scheduler was only recently published.

About hypervisor scheduler types

This article focuses specifically on the use of the new hypervisor core scheduler type versus the legacy "classic" scheduler, and how these scheduler types intersect with the use of Symmetric Multi-Threading, or SMT. It is important to understand the differences between the core and classic schedulers and how each places work from guest VMs on the underlying system processors.

The classic scheduler

The classic scheduler refers to the fair-share, round robin method of scheduling work on virtual processors (VPs) across the system - including root VPs as well as VPs belonging to guest VMs. The classic scheduler has been the default scheduler type used on all versions of Hyper-V (until Windows Server 2019, as described herein). The performance characteristics of the classic scheduler are well understood, and the classic scheduler is demonstrated to ably support over-subscription of workloads - that is, over-subscription of the host's VP:LP ratio by a reasonable margin (depending on the types of workloads being virtualized, overall resource utilization, etc.).

When run on a virtualization host with SMT enabled, the classic scheduler will schedule guest VPs from any VM on each SMT thread belonging to a core independently. Therefore, different VMs can be running on the same core at the same time (one VM running on one thread of a core while another VM is running on the other thread).

The core scheduler

The core scheduler leverages the properties of SMT to provide isolation of guest workloads, which impacts both security and system performance. The core scheduler ensures that VPs from a VM are scheduled on sibling SMT threads. This is done

symmetrically so that if LPs are in groups of two, VPs are scheduled in groups of two, and a system CPU core is never shared between VMs.

By scheduling guest VPs on underlying SMT pairs, the core scheduler offers a strong security boundary for workload isolation, and can also be used to reduce performance variability for latency sensitive workloads.

Note that when the VP is scheduled for a virtual machine without SMT enabled, that VP will consume the entire core when it runs, and the core's sibling SMT thread will be left idle. This is necessary to provide the correct workload isolation, but impacts overall system performance, especially as the system LPs become over-subscribed - that is, when total VP:LP ratio exceeds 1:1. Therefore, running guest VMs configured without multiple threads per core is a sub-optimal configuration.

Benefits of the using the core scheduler

The core scheduler offers the following benefits:

- A strong security boundary for guest workload isolation - Guest VPs are constrained to run on underlying physical core pairs, reducing vulnerability to side-channel snooping attacks.
- Reduced workload variability - Guest workload throughput variability is significantly reduced, offering greater workload consistency.
- Use of SMT in guest VMs - The OS and applications running in the guest virtual machine can utilize SMT behavior and programming interfaces (APIs) to control and distribute work across SMT threads, just as they would when run non-virtualized.

The core scheduler is currently used on Azure virtualization hosts, specifically to take advantage of the strong security boundary and low workload variability. Microsoft believes that the core scheduler type should be and will continue to be the default hypervisor scheduling type for the majority of virtualization scenarios. Therefore, to ensure our customers are secure by default, Microsoft is making this change for Windows Server 2019 now.

Core scheduler performance impacts on guest workloads

While required to effectively mitigate certain classes of vulnerabilities, the core scheduler may also potentially reduce performance. Customers may see a difference in the performance characteristics with their VMs and impacts to the overall workload capacity of their virtualization hosts. In cases where the core scheduler must run a non-

SMT VP, only one of the instruction streams in the underlying logical core executes while the other must be left idle. This will limit the total host capacity for guest workloads.

These performance impacts can be minimized by following the deployment guidance in this document. Host administrators must carefully consider their specific virtualization deployment scenarios and balance their tolerance for security risk against the need for maximum workload density, over-consolidation of virtualization hosts, etc.

Changes to the default and recommended configurations for Windows Server 2016 and Windows Server 2019

Deploying Hyper-V hosts with the maximum security posture requires use of the hypervisor core scheduler type. To ensure our customers are secure by default, Microsoft is changing the following default and recommended settings.

Note

While the hypervisor's internal support for the scheduler types was included in the initial release of Windows Server 2016, Windows Server 1709, and Windows Server 1803, updates are required in order to access the configuration control which allows selecting the hypervisor scheduler type. Please refer to [Understanding and using Hyper-V hypervisor scheduler types](#) for details on these updates.

Virtualization host changes

- The hypervisor will use the core scheduler by default beginning with Windows Server 2019.
- Microsoft recommends configuring the core scheduler on Windows Server 2016. The hypervisor core scheduler type is supported in Windows Server 2016, however the default is the classic scheduler. The core scheduler is optional and must be explicitly enabled by the Hyper-V host administrator.

Virtual machine configuration changes

- On Windows Server 2019, new virtual machines created using the default VM version 9.0 will automatically inherit the SMT properties (enabled or disabled) of

the virtualization host. That is, if SMT is enabled on the physical host, newly created VMs will also have SMT enabled, and will inherit the SMT topology of the host by default, with the VM having the same number of hardware threads per core as the underlying system. This will be reflected in the VM's configuration with HwThreadCountPerCore = 0, where 0 indicates the VM should inherit the host's SMT settings.

- Existing virtual machines with a VM version of 8.2 or earlier will retain their original VM processor setting for HwThreadCountPerCore, and the default for 8.2 VM version guests is HwThreadCountPerCore = 1. When these guests run on a Windows Server 2019 host, they will be treated as follows:
 1. If the VM has a VP count that is less than or equal to the count of LP cores, the VM will be treated as a non-SMT VM by the core scheduler. When the guest VP runs on a single SMT thread, the core's sibling SMT thread will be idled. This is non-optimal, and will result in overall loss of performance.
 2. If the VM has more VPs than LP cores, the VM will be treated as an SMT VM by the core scheduler. However, the VM will not observe other indications that it is an SMT VM. For example, use of the CPUID instruction or Windows APIs to query CPU topology by the OS or applications will not indicate that SMT is enabled.
- When an existing VM is explicitly updated from earlier VM versions to version 9.0 through the Update-VM operation, the VM will retain its current value for HwThreadCountPerCore. The VM will not have SMT force-enabled.
- On Windows Server 2016, Microsoft recommends enabling SMT for guest VMs. By default, VMs created on Windows Server 2016 would have SMT disabled, that is HwThreadCountPerCore is set to 1, unless explicitly changed.

 **Note**

Windows Server 2016 does not support setting HwThreadCountPerCore to 0.

Managing virtual machine SMT configuration

The guest virtual machine SMT configuration is set on a per-VM basis. The host administrator can inspect and configure a VM's SMT configuration to select from the following options:

1. Configure VMs to run as SMT-enabled, optionally inheriting the host SMT topology automatically
2. Configure VMs to run as non-SMT

The SMT configuration for a VM is displayed in the Summary panes in the Hyper-V Manager console. Configuring a VM's SMT settings may be done by using the VM Settings or PowerShell.

Configuring VM SMT settings using PowerShell

To configure the SMT settings for a guest virtual machine, open a PowerShell window with sufficient permissions, and type:

PowerShell

```
Set-VMProcessor -VMName <VMName> -HwThreadCountPerCore <0, 1, 2>
```

Where:

- 0 = Inherit SMT topology from the host (this setting of HwThreadCountPerCore=0 is not supported on Windows Server 2016)
- 1 = Non-SMT
- Values > 1 = the desired number of SMT threads per core. May not exceed the number of physical SMT threads per core.

To read the SMT settings for a guest virtual machine, open a PowerShell window with sufficient permissions, and type:

PowerShell

```
(Get-VMProcessor -VMName <VMName>).HwThreadCountPerCore
```

Note that guest VMs configured with HwThreadCountPerCore = 0 indicates that SMT will be enabled for the guest, and will expose the same number of SMT threads to the guest as they are on the underlying virtualization host, typically 2.

Guest VMs may observe changes to CPU topology across VM mobility scenarios

The OS and applications in a VM may see changes to both host and VM settings before and after VM lifecycle events such as live migration or save and restore operations. During an operation in which VM state is saved and restored, both the VM's HwThreadCountPerCore setting and the realized value (that is, the computed combination of the VM's setting and source host's configuration) are migrated. The VM will continue running with these settings on the destination host. At the point the VM is shutdown and re-started, it's possible that the realized value observed by the VM will change. This should be benign, as OS and application layer software should look for CPU topology information as part of their normal startup and initialization code flows. However, because these boot time initialization sequences are skipped during live migration or save/restore operations, VMs that undergo these state transitions could observe the originally computed realized value until they are shut down and re-started.

Alerts regarding non-optimal VM configurations

Virtual machines configured with more VPs than there are physical cores on the host result in a non-optimal configuration. The hypervisor scheduler will treat these VMs as if they are SMT-aware. However, OS and application software in the VM will be presented a CPU topology showing SMT is disabled. When this condition is detected, the Hyper-V Worker Process will log an event on the virtualization host warning the host administrator that the VM's configuration is non-optimal, and recommending SMT be enabled for the VM.

How to identify non-optimally configured VMs

You can identify non-SMT VMs by examining the System Log in Event Viewer for Hyper-V Worker Process event ID 3498, which will be triggered for a VM whenever the number of VPs in the VM is greater than the physical core count. Worker process events can be obtained from Event Viewer, or via PowerShell.

Querying the Hyper-V worker process VM event using PowerShell

To query for Hyper-V worker process event ID 3498 using PowerShell, enter the following commands from a PowerShell prompt.

PowerShell

```
Get-WinEvent -FilterHashTable @{ProviderName="Microsoft-Windows-Hyper-V-Worker"; ID=3498}
```

Impacts of guest SMT configuration on the use of hypervisor enlightenments for guest operating systems

The Microsoft hypervisor offers multiple enlightenments, or hints, which the OS running in a guest VM may query and use to trigger optimizations, such as those that might benefit performance or otherwise improve handling of various conditions when running virtualized. One recently introduced enlightenment concerns the handling of virtual processor scheduling and the use of OS mitigations for side-channel attacks that exploit SMT.

ⓘ Note

Microsoft recommends that host administrators enable SMT for guest VMs to optimize workload performance.

The details of this guest enlightenment are provided below, however the key takeaway for virtualization host administrators is that virtual machines should have HwThreadCountPerCore configured to match the host's physical SMT configuration. This allows the hypervisor to report that there is no non-architectural core sharing. Therefore, any guest OS supporting optimizations that require the enlightenment may be enabled. On Windows Server 2019, create new VMs and leave the default value of HwThreadCountPerCore (0). Older VMs migrated from Windows Server 2016 hosts can be updated to the Windows Server 2019 configuration version. After doing so, Microsoft recommends setting HwThreadCountPerCore = 0. On Windows Server 2016, Microsoft recommends setting HwThreadCountPerCore to match the host configuration (typically 2).

NoNonArchitecturalCoreSharing enlightenment details

Starting in Windows Server 2016, the hypervisor defines a new enlightenment to describe its handling of VP scheduling and placement to the guest OS. This enlightenment is defined in the [Hypervisor Top Level Functional Specification v5.0c](#).

Hypervisor synthetic CPUID leaf

CPUID.0x40000004.EAX:18[NoNonArchitecturalCoreSharing = 1] indicates that a virtual processor will never share a physical core with another virtual processor, except for virtual processors that are reported as sibling SMT threads. For example, a guest VP will never run on an SMT thread alongside a root VP running simultaneously on a sibling SMT thread on the same processor core. This condition is only possible when running virtualized, and so represents a non-architectural SMT behavior that also has serious

security implications. The guest OS can use NoNonArchitecturalCoreSharing = 1 as an indication that it is safe to enable optimizations, which may help it avoid the performance overhead of setting STIBP.

In certain configurations, the hypervisor will not indicate that NoNonArchitecturalCoreSharing = 1. As an example, if a host has SMT enabled and is configured to use the hypervisor classic scheduler, NoNonArchitecturalCoreSharing will be 0. This may prevent enlightened guests from enabling certain optimizations. Therefore, Microsoft recommends that host administrators using SMT rely on the hypervisor core scheduler and ensure that virtual machines are configured to inherit their SMT configuration from the host to ensure optimal workload performance.

Summary

The security threat landscape continues to evolve. To ensure our customers are secure by default, Microsoft is changing the default configuration for the hypervisor and virtual machines starting in Windows Server 2019 Hyper-V, and providing updated guidance and recommendations for customers running Windows Server 2016 Hyper-V.

Virtualization host administrators should:

- Read and understand the guidance provided in this document
- Carefully evaluate and adjust their virtualization deployments to ensure they meet the security, performance, virtualization density, and workload responsiveness goals for their unique requirements
- Consider re-configuring existing Windows Server 2016 Hyper-V hosts to leverage the strong security benefits offered by the hypervisor core scheduler
- Update existing non-SMT VMs to reduce the performance impacts from scheduling constraints imposed by VP isolation that addresses hardware security vulnerabilities

Manage Hyper-V Integration Services

Article • 01/18/2022 • 7 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows 11, Windows 10

Hyper-V Integration Services enhance virtual machine performance and provide convenience features by leveraging two-way communication with the Hyper-V host. Many of these services are conveniences, such as guest file copy, while others are important to the virtual machine's functionality, such as synthetic device drivers. This set of services and drivers are sometimes referred to as *integration components*. You can control whether or not individual convenience services operate for any given virtual machine. The driver components are not intended to be serviced manually.

For details about each integration service, see [Hyper-V Integration Services](#).

ⓘ Important

Each service you want to use must be enabled in both the host and guest in order to function. All integration services except *Hyper-V Guest Service Interface* are on by default on Windows guest operating systems. The services can be turned on and off individually. The next sections show you how.

Turn an integration service on or off using Hyper-V Manager

1. From the center pane, right-click the virtual machine and select **Settings**.
2. From the left pane of the **Settings** window, under **Management**, select **Integration Services**.

The Integration Services pane lists all integration services available on the Hyper-V host, and whether the host has enabled the virtual machine to use them.

Turn an integration service on or off using PowerShell

To do this in PowerShell, use [Enable-VMIntegrationService](#) and [Disable-VMIntegrationService](#).

The following examples demonstrate turning the guest file copy integration service on and off for a virtual machine named *DemoVM*.

1. Get a list of running integration services:

```
PowerShell
```

```
Get-VMIntegrationService -VMName "DemoVM"
```

2. The output should look like this:

```
PowerShell
```

VMName	Name	Enabled	PrimaryStatus	Description
SecondaryStatus	Description	-----	-----	-----
DemoVM	Guest Service Interface	False	OK	
DemoVM	Heartbeat	True	OK	OK
DemoVM	Key-Value Pair Exchange	True	OK	
DemoVM	Shutdown	True	OK	
DemoVM	Time Synchronization	True	OK	
DemoVM	VSS	True	OK	

3. Turn on Guest Service Interface:

```
PowerShell
```

```
Enable-VMIntegrationService -VMName "DemoVM" -Name "Guest Service Interface"
```

4. Verify that Guest Service Interface is enabled:

```
PowerShell
```

```
Get-VMIntegrationService -VMName "DemoVM"
```

5. Turn off Guest Service Interface:

```
PowerShell
```

```
Disable-VMIntegrationService -VMName "DemoVM" -Name "Guest Service Interface"
```

Checking the guest's integration services version

Some features may not work correctly or at all if the guest's integration services are not current. To get the version information for Windows, sign in to the guest operating system, open a command prompt, and run this command:

```
REG QUERY "HKLM\Software\Microsoft\Virtual Machine\Auto" /v  
IntegrationServicesVersion
```

Earlier guest operating systems will not have all available services. For example, Windows Server 2008 R2 guests cannot have the Hyper-V Guest Service Interface.

Start and stop an integration service from a Windows guest

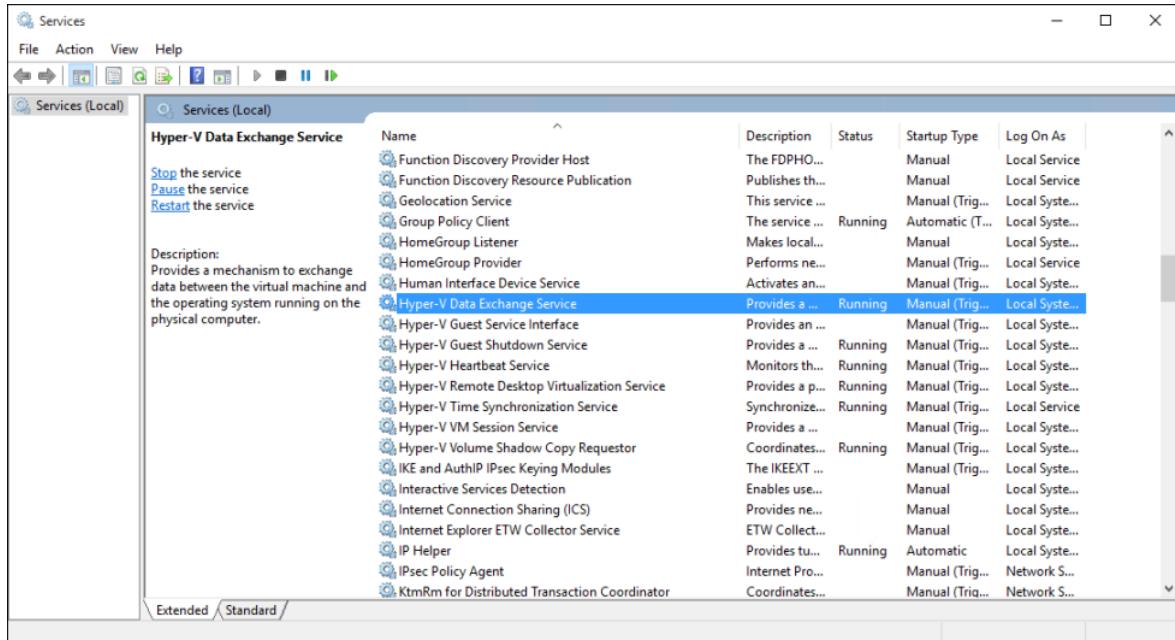
In order for an integration service to be fully functional, its corresponding service must be running within the guest in addition to being enabled on the host. In Windows guests, each integration service is listed as a standard Windows service. You can use the Services applet in Control Panel or PowerShell to stop and start these services.

Important

Stopping an integration service may severely affect the host's ability to manage your virtual machine. To work correctly, each integration service you want to use must be enabled on both the host and guest. As a best practice, you should only control integration services from Hyper-V using the instructions above. The matching service in the guest operating system will stop or start automatically when you change its status in Hyper-V. If you start a service in the guest operating system but it is disabled in Hyper-V, the service will stop. If you stop a service in the guest operating system that is enabled in Hyper-V, Hyper-V will eventually start it again. If you disable the service in the guest, Hyper-V will be unable to start it.

Use Windows Services to start or stop an integration service within a Windows guest

1. Open Services manager by running `services.msc` as an Administrator or by double-clicking the Services icon in Control Panel.



2. Find the services that start with **Hyper-V**.

3. Right-click the service you want start or stop. Select the desired action.

Use PowerShell to start or stop an integration service within a Windows guest

1. To get a list of integration services, run:

```
PowerShell
Get-Service -Name vmic* | FT -AutoSize
```

2. The output should look similar to this:

```
PowerShell
Status   Name           DisplayName
-----   --
Running  vmicguestinterface  Hyper-V Guest Service Interface
Running  vmicheartbeat    Hyper-V Heartbeat Service
Running  vmickvpexchange Hyper-V Data Exchange Service
Running  vmicrdv          Hyper-V Remote Desktop Virtualization
                           Service
Running  vmicshutdown     Hyper-V Guest Shutdown Service
Running  vmiictimesync    Hyper-V Time Synchronization Service
Stopped  vmicvmsession   Hyper-V PowerShell Direct Service
Running  vmicvss          Hyper-V Volume Shadow Copy Requestor
```

-
3. Run either [Start-Service](#) or [Stop-Service](#). For example, to turn off Windows PowerShell Direct, run:

```
PowerShell  
  
Stop-Service -Name vmicvmsession
```

Start and stop an integration service from a Linux guest

Linux integration services are generally provided through the Linux kernel. The Linux integration services driver is named `hv_utils`.

1. To find out if `hv_utils` is loaded, use this command:

```
Bash  
  
lsmod | grep hv_utils
```

2. The output should look similar to this:

```
Bash  
  
Module           Size  Used by  
hv_utils          20480  0  
hv_vmbus         61440   8  
hv_balloon,hyperv_keyboard,hv_netvsc,hid_hyperv,hv_utils,hyperv_fb,hv_storvsc
```

3. To find out if the required daemons are running, use this command.

```
Bash  
  
ps -ef | grep hv
```

4. The output should look similar to this:

```
Bash  
  
root      236      2  0 Jul11 ?          00:00:00 [hv_vmbus_con]  
root      237      2  0 Jul11 ?          00:00:00 [hv_vmbus_ctl]  
...  
root      252      2  0 Jul11 ?          00:00:00 [hv_vmbus_ctl]
```

```
root      1286      1  0 Jul11 ?          00:01:11 /usr/lib/linux-
tools/3.13.0-32-generic/hv_kvp_daemon
root      9333      1  0 Oct12 ?          00:00:00 /usr/lib/linux-
tools/3.13.0-32-generic/hv_kvp_daemon
root      9365      1  0 Oct12 ?          00:00:00 /usr/lib/linux-
tools/3.13.0-32-generic/hv_vss_daemon
user     43774  43755  0 21:20 pts/0    00:00:00 grep --color=auto hv
```

5. To see what daemons are available, run:

```
Bash  
compgen -c hv_
```

6. The output should look similar to this:

```
Bash  
hv_vss_daemon
hv_get_dhcp_info
hv_get_dns_info
hv_set_ifconfig
hv_kvp_daemon
hv_fcopy_daemon
```

Integration service daemons that might be listed include the following. If any are missing, they might not be supported on your system or they might not be installed. Find details, see [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#).

- **hv_vss_daemon**: This daemon is required to create live Linux virtual machine backups.
- **hv_kvp_daemon**: This daemon allows setting and querying intrinsic and extrinsic key value pairs.
- **hv_fcopy_daemon**: This daemon implements a file copying service between the host and guest.

Examples

These examples demonstrate stopping and starting the KVP daemon, named `hv_kvp_daemon`.

1. Use the process ID (PID) to stop the daemon's process. To find the PID, look at the second column of the output, or use `pidof`. Hyper-V daemons run as root, so you'll need root permissions.

Bash

```
sudo kill -15 `pidof hv_kvp_daemon`
```

2. To verify that all `hv_kvp_daemon` processes are gone, run:

Bash

```
ps -ef | hv
```

3. To start the daemon again, run the daemon as root:

Bash

```
sudo hv_kvp_daemon
```

4. To verify that the `hv_kvp_daemon` process is listed with a new process ID, run:

Bash

```
ps -ef | hv
```

Keep integration services up to date

We recommend that you keep integration services up to date to get the best performance and most recent features for your virtual machines. This happens for Windows guests by default if they are set up to get important updates from Windows Update. Linux guests using current kernels contain integration services built in, but there may be optional updates available. You will receive the latest integration components when you update the kernel. For more information about Linux guests, see [Supported Linux and FreeBSD virtual machines for Hyper-V on Windows](#).

Note

The image file *Integration Services disk* (vmguest.iso) isn't included with Hyper-V starting with Windows Server 2016 and Windows 10 because it's no longer needed. Windows Server 2012 and older require the Data Exchange integration service. If the Data Exchange integration service can't be enabled, integration services for these guests are available from the [Download Center](#) as a cabinet (cab) file. Instructions for applying a cab are available in this [Microsoft TechCommunity blog](#)

[post](#). If your Hyper-V host is running Windows Server 2012 R2 and older, see the next section for how to install or update integration services.

Install or update integration services for Hyper-V hosts earlier than Windows Server 2016 and Windows 10

Note

This isn't required for Windows Server 2016 and Windows 10 or newer.

For Hyper-V hosts earlier than Windows Server 2016 and Windows 10, you'll need to **manually install or update** the integration services in the guest operating systems.

To manually install or update the integration services:

1. Open Hyper-V Manager.
2. Connect to the virtual machine. Right-click the virtual machine and select **Connect**.
3. From the Action menu of Virtual Machine Connection, select **Insert Integration Services Setup Disk**. This action loads the setup disk in the virtual DVD drive. Depending on the guest operating system, you might need to start the installation manually from File Explorer.
4. After the installation finishes, integration services are available for use.

Manage Windows virtual machines with PowerShell Direct

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows 10, Windows Server 2016, Windows Server 2019

You can use PowerShell Direct to remotely manage a Windows 10, Windows Server 2016, or Windows Server 2019 virtual machine from a Windows 10, Windows Server 2016, or Windows Server 2019 Hyper-V host. PowerShell Direct allows Windows PowerShell management inside a virtual machine regardless of the network configuration or remote management settings on either the Hyper-V host or the virtual machine. This makes it easier for Hyper-V Administrators to automate and script virtual machine management and configuration.

There are two ways to run PowerShell Direct:

- Create and exit a PowerShell Direct session using `PSSession` cmdlets
- Run script or command with the `Invoke-Command` cmdlet

If you're managing older virtual machines, use Virtual Machine Connection (VMConnect) or [configure a virtual network for the virtual machine](#).

Create and exit a PowerShell Direct session using `PSSession` cmdlets

1. On the Hyper-V host, open Windows PowerShell as Administrator.
2. Use the [Enter-PSSession](#) cmdlet to connect to the virtual machine. Run one of the following commands to create a session by using the virtual machine name or GUID:

```
Enter-PSSession -VMName <VMName>
```

```
Enter-PSSession -VMGUID <VMGUID>
```

3. Type your credentials for the virtual machine.
4. Run whatever commands you need to. These commands run on the virtual machine that you created the session with.
5. When you're done, use the [Exit-PSSession](#) to close the session.

```
Exit-PSSession
```

Run script or command with Invoke-Command cmdlet

You can use the [Invoke-Command](#) cmdlet to run a pre-determined set of commands on the virtual machine. Here is an example of how you can use the Invoke-Command cmdlet where PSTest is the virtual machine name and the script to run (foo.ps1) is in the script folder on the C:/ drive:

```
Invoke-Command -VMName PSTest -FilePath C:\script\foo.ps1
```

To run a single command, use the [-ScriptBlock](#) parameter:

```
Invoke-Command -VMName PSTest -ScriptBlock { cmdlet }
```

What's required to use PowerShell Direct?

To create a PowerShell Direct session on a virtual machine,

- The virtual machine must be running locally on the host and booted.
- You must be logged into the host computer as a Hyper-V administrator.
- You must supply valid user credentials for the virtual machine.
- The host operating system must run at least Windows 10 or Windows Server 2016.
- The virtual machine must run at least Windows 10 or Windows Server 2016.

You can use the [Get-VM](#) cmdlet to check that the credentials you're using have the Hyper-V administrator role and to get a list of the virtual machines running locally on the host and booted.

See Also

[Enter-PSSession](#) [Exit-PSSession](#) [Invoke-Command](#)

Set up Hyper-V Replica

Article • 10/18/2022 • 11 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016; Azure Stack HCI, versions 22H2 and 21H2

Hyper-V Replica is an integral part of the Hyper-V role. It contributes to your disaster recovery strategy by replicating virtual machines from one Hyper-V host server to another to keep your workloads available. Hyper-V Replica creates a copy of a live virtual machine to a replica offline virtual machine. Note the following:

- **Hyper-V hosts:** Primary and secondary host servers can be physically co-located or in separate geographical locations with replication over a WAN link. Hyper-V hosts can be standalone, clustered, or a mixture of both. There's no Active Directory dependency between the servers and they don't need to be domain members.
- **Replication and change tracking:** When you enable Hyper-V Replica for a specific virtual machine, initial replication creates an identical replica virtual machine on a secondary host server. After that happens, Hyper-V Replica change tracking creates and maintains a log file that captures changes on a virtual machine VHD. The log file is played in reverse order to the replica VHD based on replication frequency settings. This means that the latest changes are stored and replicated asynchronously. Replication can be over HTTP or HTTPS.
- **Extended (chained) replication:** This lets you replicate a virtual machine from a primary host to a secondary host, and then replicate the secondary host to a third host. Note that you can't replicate from the primary host directly to the second and the third.

This feature makes Hyper-V Replica more robust for disaster recovery because if an outage occurs you can recover from both the primary and extended replica. You can fail over to the extended replica if your primary and secondary locations go down. Note that the extended replica doesn't support application-consistent replication and must use the same VHDs that the secondary replica is using.

- **Failover:** If an outage occurs in your primary (or secondary in case of extended) location, you can manually initiate a test, planned, or unplanned failover.

Question	Test	Planned	Unplanned
----------	------	---------	-----------

Question	Test	Planned	Unplanned
When should I run this?	Verify that a virtual machine can fail over and start in the secondary site Useful for testing and training	During planned downtime and outages	During unexpected events
Is a duplicate virtual machine created?	Yes	No	No
Where is it initiated?	On the replica virtual machine	Initiated on primary and completed on secondary	On the replica virtual machine
How often should I run?	We recommend once a month for testing	Once every six months or in accordance with compliance requirements	Only in case of disaster when the primary virtual machine is unavailable
Does the primary virtual machine continue to replicate?	Yes	Yes. When the outage is resolved reverse replication replicates the changes back to the primary site so that primary and secondary are synchronized.	No
Is there any data loss?	None	None. After failover Hyper-V Replica replicates the last set of tracked changes back to the primary to ensure zero data loss.	Depends on the event and recovery points
Is there any downtime?	None. It doesn't impact your production environment. It creates a duplicate test virtual machine during failover. After failover finishes you select Failover on the replica virtual machine and it's automatically cleaned up and deleted.	The duration of the planned outage	The duration of the unplanned outage

- **Recovery points:** When you configure replication settings for a virtual machine, you specify the recovery points you want to store from it. Recovery points represent a snapshot in time from which you can recover a virtual machine. Obviously less data is lost if you recover from a very recent recovery point. You can access recovery points up to 24 hours ago.

Deployment prerequisites

Here's what you should verify before you begin:

- **Figure out which VHDs need to be replicated.** In particular, VHDs that contain data that is rapidly changing and not used by the Replica server after failover, such as page file disks, should be excluded from replication to conserve network bandwidth. Make a note of which VHDs can be excluded.
- **Decide how often you need to synchronize data:** The data on the Replica server is synchronized updated according to the replication frequency you configure (30 seconds, 5 minutes, or 15 minutes). The frequency you choose should consider the following: Are the virtual machines running critical data with a low RPO? What are your bandwidth considerations? Your highly-critical virtual machines will obviously need more frequent replication.
- **Decide how to recover data:** By default Hyper-V Replica only stores a single recovery point that will be the latest replication sent from the primary to the secondary. However if you want the option to recover data to an earlier point in time you can specify that additional recovery points should be stored (to a maximum of 24 hourly points). If you do need additional recovery points you should note that this requires more overhead on processing and storage resources.
- **Figure out which workloads you'll replicate:** Standard Hyper-V Replica replication maintains consistency in the state of the virtual machine operating system after a failover, but not the state of applications that running on the virtual machine. If you want to be able to recovery your workload state you can create app-consistent recovery points. Note that app-consistent recovery isn't available on the extended replica site if you're using extended (chained) replication.
- **Decide how to do the initial replication of virtual machine data:** Replication starts by transferring the needs to transfer the current state of the virtual machines. This initial state can be transmitted directly over the existing network, either immediately or at a later time that you configure. You can also use a pre-existing restored virtual machine (for example, if you have restored an earlier backup of the

virtual machine on the Replica server) as the initial copy. Or, you can save network bandwidth by copying the initial copy to external media and then physically delivering the media to the Replica site. If you want to use a preexisting virtual machine delete all previous snapshots associated with it.

Deployment steps

Step 1: Set up the Hyper-V hosts

You'll need at least two Hyper-V hosts with one or more virtual machines on each server. [Get started with Hyper-V](#). The host server that you'll replicate virtual machines to will need to be set up as the replica server.

1. In the Hyper-V settings for the server you'll replicate virtual machines to, in **Replication Configuration**, select **Enable this computer as a Replica server**.
2. You can replicate over HTTP or encrypted HTTPS. Select **Use Kerberos (HTTP)** or **Use certificate-based Authentication (HTTPS)**. By default HTTP 80 and HTTPS 443 are enabled as firewall exceptions on the replica Hyper-V server. If you change the default port settings you'll need to also change the firewall exception. If you're replicating over HTTPS, you'll need to select a certificate and you should have certificate authentication set up.
3. For authorization, select **Allow replication from any authenticated server** to allow the replica server to accept virtual machine replication traffic from any primary server that authenticates successfully. Select **Allow replication from the specified servers** to accept traffic only from the primary servers you specifically select.
For both options you can specify where the replicated VHDs should be stored on the replica Hyper-V server.
4. Click **OK** or **Apply**.

Step 2: Set up the firewall

To allow replication between the primary and secondary servers, traffic must get through the Windows firewall (or any other third-party firewalls). When you installed the Hyper-V role on the servers by default exceptions for HTTP (80) and HTTPS (443) are created. If you're using these standard ports, you'll just need to enable the rules:

- To enable the rules on a standalone host server:

1. Open Windows Firewall with Advance Security and click **Inbound Rules**.
 2. To enable HTTP (Kerberos) authentication, right-click **Hyper-V Replica HTTP Listener (TCP-In)** > **Enable Rule**. To enable HTTPS certificate-based authentication, right-click **Hyper-V Replica HTTPS Listener (TCP-In)** > **Enable Rule**.
- To enable rules on a Hyper-V cluster, open a Windows PowerShell session using **Run as Administrator**, then run one of these commands:
 - For HTTP:


```
get-clusternode | ForEach-Object {Invoke-command -computername $_.name -scriptblock {Enable-Netfirewallrule -displayname "Hyper-V Replica HTTP Listener (TCP-In)"}}
```
 - For HTTPS:


```
get-clusternode | ForEach-Object {Invoke-command -computername $_.name -scriptblock {Enable-Netfirewallrule -displayname "Hyper-V Replica HTTPS Listener (TCP-In)"}}
```

Enable virtual machine replication

Do the following on each virtual machine you want to replicate:

1. In the **Details** pane of Hyper-V Manager, select a virtual machine by clicking it. Right-click the selected virtual machine and click **Enable Replication** to open the Enable Replication wizard.
2. On the **Before you Begin** page, click **Next**.
3. On the **Specify Replica Server** page, in the Replica Server box, enter either the NetBIOS or FQDN of the Replica server. If the Replica server is part of a failover cluster, enter the name of the Hyper-V Replica Broker. Click **Next**.
4. On the **Specify Connection Parameters** page, Hyper-V Replica automatically retrieves the authentication and port settings you configured for the replica server. If values aren't being retrieved check that the server is configured as a replica server, and that it's registered in DNS. If required type in the setting manually.
5. On the **Choose Replication VHDs** page, make sure the VHDs you want to replicate are selected, and clear the checkboxes for any VHDs that you want to exclude from replication. Then click **Next**.

6. On the **Configure Replication Frequency** page, specify how often changes should be synchronized from primary to secondary. Then click **Next**.
7. On the **Configure Additional Recovery Points** page, select whether you want to maintain only the latest recovery point or to create additional points. If you want to consistently recover applications and workloads that have their own VSS writers we recommend you select **Volume Shadow Copy Service (VSS) frequency** and specify how often to create app-consistent snapshots. Note that the Hyper-V VMM Requestor Service must be running on both the primary and secondary Hyper-V servers. Then click **Next**.
8. On the **Choose Initial Replication** page, select the initial replication method to use. The default setting to send initial copy over the network will copy the primary virtual machine configuration file (VMCX) and the virtual hard disk files (VHDX and VHD) you selected over your network connection. Verify network bandwidth availability if you're going to use this option. If the primary virtual machine is already configured on the secondary site as a replicate virtual machine it can be useful to select **Use an existing virtual machine on the replication server as the initial copy**. You can use Hyper-V export to export the primary virtual machine and import it as a replica virtual machine on the secondary server. For larger virtual machines or limited bandwidth you can choose to have initial replication over the network occur at a later time, and then configure off-peak hours, or to send the initial replication information as offline media.

If you do offline replication you'll transport the initial copy to the secondary server using an external storage medium such as a hard disk or USB drive. To do this you'll need to connect the external storage to the primary server (or owner node in a cluster) and then when you select Send initial copy using external media you can specify a location locally or on your external media where the initial copy can be stored. A placeholder virtual machine is created on the replica site. After initial replication completes the external storage can be shipped to the replica site. There you'll connect the external media to the secondary server or to the owner node of the secondary cluster. Then you'll import the initial replica to a specified location and merge it into the placeholder virtual machine.
9. On the **Completing the Enable Replication** page, review the information in the Summary and then click **Finish**. The virtual machine data will be transferred in accordance with your chosen settings. and a dialog box appears indicating that replication was successfully enabled.
10. If you want to configure extended (chained) replication, open the replica server, and right-click the virtual machine you want to replicate. Click **Replication >**

Extend Replication and specify the replication settings.

Run a failover

After completing these deployment steps your replicated environment is up and running. Now you can run failovers as needed.

Test failover: If you want to run a test failover right-click the replica virtual machine and select **Replication > Test Failover**. Pick the latest or other recovery point if configured. A new test virtual machine will be created and started on the secondary site. After you've finished testing, select **Stop Test Failover** on the replica virtual machine to clean it up. Note that for a virtual machine you can only run one test failover at a time. For more information, see [Test failover in Hyper-V Replica](#).

Planned failover: To run a planned failover, right-click the primary virtual machine and select **Replication > Planned Failover**. Planned failover performs prerequisites checks to ensure zero data loss. It checks that the primary virtual machine is shut down before beginning the failover. After the virtual machine is failed over, it starts replicating the changes back to the primary site when it's available. Note that for this to work, the primary server should be configured to receive replication from the secondary server, or from the Hyper-V Replica Broker in the case of a primary cluster. Planned failover sends the last set of tracked changes. For more information, see [Planned failover in Hyper-V Replica](#).

Unplanned failover: To run an unplanned failover, right-click on the replica virtual machine and select **Replication > Unplanned Failover** from Hyper-V Manager or Failover Clustering Manager. You can recover from the latest recovery point or from previous recovery points if this option is enabled. After failover, check that everything is working as expected on the failed over virtual machine, then click **Complete** on the replica virtual machine. [Read more](#).

Enable Intel Performance Monitoring Hardware in a Hyper-V virtual machine

Article • 05/17/2021 • 2 minutes to read

Intel processors contain features collectively called performance monitoring hardware (e.g. PMU, PEBS, LBR). These features are used by performance tuning software like Intel VTune Amplifier to analyze software performance. Prior to Windows Server 2019 and Windows 10 Version 1809, neither the host operating system nor Hyper-V guest virtual machines could use performance monitoring hardware when Hyper-V was enabled. Starting with Windows Server 2019 and Windows 10 Version 1809, the host operating system has access to performance monitoring hardware by default. Hyper-V guest virtual machines do not have access by default, but Hyper-V administrators may choose to grant access to one or more guest virtual machines. This document describes the steps required to expose performance monitoring hardware to guest virtual machines.

Requirements

To enable performance monitoring hardware in a virtual machine, you'll need:

- An Intel processor with performance monitoring hardware (i.e. PMU, PEBS, LBR). Refer to [this document](#) from Intel to determine which performance monitoring hardware your system supports.
- Windows Server 2019 or Windows 10 Version 1809 (October 2018 Update) or later
- A Hyper-V virtual machine *without nested virtualization* that is also in the stopped state

To enable upcoming Intel Processor Trace (IPT) performance monitoring hardware in a virtual machine, you'll need:

- An Intel processor that supports IPT and the PT2GPA feature. [^1] Refer to [this document](#) from Intel to determine which performance monitoring hardware your system supports.
- Windows Server version 1903 (SAC) or Windows 10 Version 1903 (May 2019 Update) or later
- A Hyper-V virtual machine *without nested virtualization* that is also in the stopped state
- PMU needs to be enabled via the command line using the command seen below.

[^1]: PT2GPA refers to the “Intel PT uses guest physical addresses” bit. This is described in 25.5.4.1 of the Intel SDM.

Enabling performance monitoring components in a virtual machine

To enable different performance monitoring components for a specific guest virtual machine, use the `Set-VMProcessor` PowerShell cmdlet while running as Administrator:

ⓘ Note

The virtual machine generation must be 9.1 or higher. If nested virtualization is offered to the guest as well, then this needs 9.3 and up.

Powershell

```
# Enable IPT  
Set-VMProcessor MyVMName -Perfmon @("ipt", "pmu")
```

Powershell

```
# Enable all components  
Set-VMProcessor MyVMName -Perfmon @("ipt", "pmu", "lbr", "pebs")
```

Powershell

```
# Disable all components  
Set-VMProcessor MyVMName -Perfmon @()
```

ⓘ Note

When enabling the performance monitoring components, if `"pebs"` is specified, then `"pmu"` must also be specified.

PEBS is only supported on hardware that has a PMU Version ≥ 4 .

Also, any command that attempts to enable `"ipt"` must also specify `"pmu"`.

Enabling a component that is not supported by the host's physical processors will result in a virtual machine start failure.

Effects of enabling performance monitoring hardware on save/restore, export, and live

migration

Microsoft does not recommend live migrating or saving/restoring virtual machines with performance monitoring hardware between systems with different Intel hardware. The specific behavior of performance monitoring hardware is often non-architectural and changes between Intel hardware systems. Moving a running virtual machine between different systems can result in unpredictable behavior of the non-architectural counters.

Processor compatibility mode in Hyper-V

Article • 04/28/2023

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, Windows Server 2012, Windows Server 2008 R2, Hyper-V Server 2019, Hyper-V Server 2016, Hyper-V Server 2012 R2

Hyper-V offers processor compatibility mode, a feature that was first introduced in Windows Server 2008 R2. Processor compatibility mode allows you to move a running virtual machine (VM) or save state between virtualization hosts that use different generations of processors. This feature works by disabling many modern processor features, which can affect VM performance. This document provides details about processor compatibility mode for Hyper-V.

When to use processor compatibility mode

Processor compatibility mode is applicable to any VM mobility scenario that does not involve rebooting the VM. These include VM live migration, save and restore, and production checkpoints.

VMs cannot be live migrated or saved and restored across virtualization hosts that use processors from different CPU manufacturers. For example, you cannot move running VMs or VM saved state from a host with Intel processors to a host with AMD processors. If you must move a VM in this case, the VM must first be shut down, then restarted on the new host.

If you plan to move VMs without rebooting them, between virtualization hosts that may use different generations of processors, you should enable processor compatibility mode. For example, you would enable processor compatibility mode to ensure that you can live migrate your VMs between cluster nodes that use different processor feature sets. You could also use processor compatibility mode to save a VM and restore it on a host computer that has a different processor feature set than the source host.

Why processor compatibility mode is needed

Instruction set architecture (ISA) extensions are optimizations and features that are introduced by processor manufacturers. These features often improve performance or

security by using specialized hardware for a particular task. For example, many media applications make use of processor features to speed up vector calculations. These features are rarely required for applications to run; they simply boost performance.

The feature set that is available on a processor varies depending on its make, model, and age. Operating systems and application software typically enumerate the system's processor feature set and capabilities when they are first launched. Software doesn't expect the available processor features to change during their lifetime—and this could never happen when running on a physical computer because processor features are static.

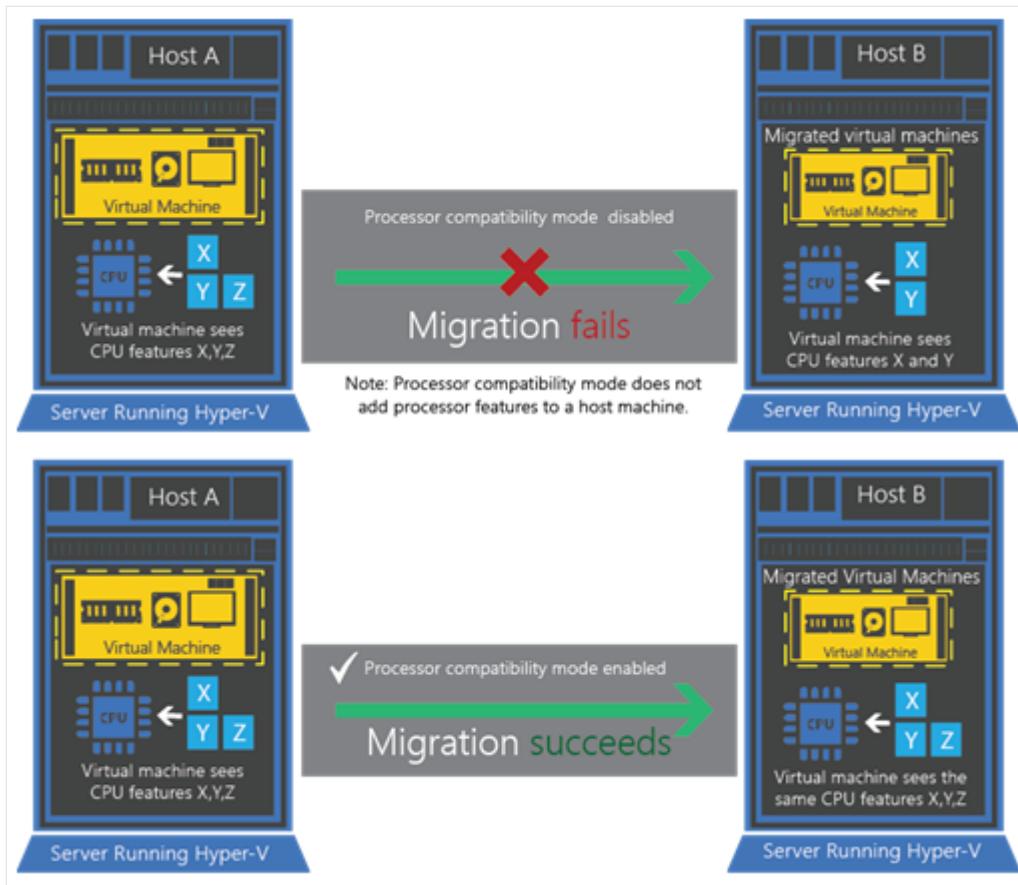
However, VM mobility features allow a running VM to be migrated to a new virtualization host. If software in the VM has detected and started using a particular processor feature, and it gets moved to a new virtualization host that lacks that capability, the software is likely to fail. This could result in the VM crashing.

To avoid these failures, Hyper-V performs "pre-flight" checks whenever a VM live migration or save/restore operation is initiated. These checks compare the set of processor features that are available to the VM on the source host against the set of features that are available on the target host. If these feature sets don't match, the migration or restore operation is canceled.

How processor compatibility mode works

Processor compatibility mode ensures that the set of processor features available to VMs across a disparate set of virtualization hosts will match by presenting only a limited set of processor features to the VM. Processor compatibility mode hides newer processor instructions sets, typically those introduced within the last 10 years. However, hiding these features means that the guest operating system and application software cannot take advantage of these processor instruction set enhancements.

For a complete list of features that are hidden for processor compatibility mode, refer to section 5.2.11 in the [Hypervisor Top-Level Functional Specification](#).



Ramifications of using processor compatibility mode

It is difficult to quantify the overall performance effects of processor compatibility mode. The performance loss is primarily dependent on the workload running in the VM. Some workloads will be unaffected, while others will show a noticeable difference. Software that heavily relies on hardware optimizations (such as encryption, compression, or intensive floating-point calculations) will be impacted the most.

The following example describes how AES encryption is affected by using processor compatibility mode, and there are many more. If you are concerned about the performance impact of processor compatibility mode, it is best to compare VM workload performance with processor compatibility mode enabled and with it disabled.

Example: AES encryption

One example of an operation that is impacted by processor compatibility mode is AES encryption (a common form of encryption). Many new Intel and AMD processors include an ISA extension that accelerates AES by using hardware. Intel claims this optimization provides a 2-3 fold performance gain, with some implementations providing a 10-fold gain. (For more information, see [Intel Advanced Encryption Standard Instructions](#).)

Applications that encrypt or decrypt a large amount of data benefit from this processor feature, so turning it off by enabling processor compatibility mode will impact the performance of these specific operations.

Using processor compatibility mode

There are important concepts to understand when using processor compatibility mode in Hyper-V:

- Running VMs can only be migrated between virtualization hosts that use processors from the same manufacturer.
- You must shut down the VM before you can enable or disable processor compatibility mode.
- Processor compatibility mode isn't needed for VM moves that involve a stop and restart of the VM.
- Anytime a VM is restarted, the guest operating system will enumerate the processor features that are available on the new host computer.

Note

In Windows Server, Microsoft recommends turning on processor compatibility mode only before VM migration scenarios, and then turning it off when the migration is complete.

Enable processor compatibility mode using Hyper-V Manager

To enable processor compatibility mode for a VM using Hyper-V Manager:

1. Shut down the VM.
2. Click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
3. Select the server running Hyper-V and the desired VM.
4. If the VM is running, you must shut down the VM to enable the processor compatibility mode setting.
5. In the Action pane, click **Settings**, and then click **Processor**.

6. Expand **Processor**, and click **Compatibility**.
7. Select **Migrate to a physical computer with a different processor**, and then click **OK**.
8. Restart the VM.

Disable processor compatibility mode using Hyper-V Manager

To disable processor compatibility mode for a VM using Hyper-V Manager:

1. Shut down the VM.
2. Click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
3. Select the server running Hyper-V and the desired VM.
4. If the VM is running, you must shut down the VM to disable the processor compatibility mode setting.
5. In the Action pane, click **Settings**, and then click **Processor**.
6. Expand **Processor**, and click **Compatibility**.
7. De-select the **Migrate to a physical computer with a different processor** checkbox, and then click **OK**.
8. Restart the VM.

Enable processor compatibility mode using PowerShell

To enable processor compatibility mode for a VM using PowerShell, shut down the VM and run the `Set-VMProcessor` cmdlet, setting `CompatibilityForMigrationEnabled` to `$true`:

```
PowerShell
```

```
get-vm -name <name of VM> -ComputerName <target cluster or host> | Set-  
VMProcessor -CompatibilityForMigrationEnabled $true
```

Then restart the VM.

Warning

You may see additional parameters for `Set-VMProcessor` that are intended for use with Azure Stack HCI. Do not attempt to use these with Windows Server, or you will get an error message. The default and only `CompatibilityForMigrationMode` available for Windows Server is `MinimumFeatureSet`. [Learn about dynamic CPU compatibility mode in Azure Stack HCI](#).

Disable processor compatibility mode using PowerShell

To disable processor compatibility mode for a VM using PowerShell, shut down the VM and run the `Set-VMProcessor` cmdlet, setting `CompatibilityForMigrationEnabled` to `$false`:

PowerShell

```
get-vm -name <name of VM> -ComputerName <target cluster or host> | Set-VMProcessor -CompatibilityForMigrationEnabled $false
```

Then restart the VM.

Next steps

See also:

- [Dynamic processor compatibility mode in Azure Stack HCI](#)

Live Migration Overview

Article • 09/17/2020 • 2 minutes to read

Live migration is a Hyper-V feature in Windows Server. It allows you to transparently move running Virtual Machines from one Hyper-V host to another without perceived downtime. The primary benefit of live migration is flexibility; running Virtual Machines are not tied to a single host machine. This allows actions like draining a specific host of Virtual Machines before decommissioning or upgrading it. When paired with Windows Failover Clustering, live migration allows the creation of highly available and fault tolerant systems.

Related Technologies and Documentation

Live migration is often used in conjunction with a few related technologies like Failover Clustering and System Center Virtual Machine Manager. If you're using Live Migration via these technologies, here are pointers to their latest documentation:

- [Failover Clustering](#) (Windows Server 2016)
- [System Center Virtual Machine Manager](#) (System Center 2016)

If you're using older versions of Windows Server, or need details on features introduced in older versions of Windows Server, here are pointers to historical documentation:

- [Live Migration](#) (Windows Server 2008 R2)
- [Live Migration](#) (Windows Server 2012 R2)
- [Failover Clustering](#) (Windows Server 2012 R2)
- [Failover Clustering](#) (Windows Server 2008 R2)
- [System Center Virtual Machine Manager](#) (System Center 2012 R2)
- [System Center Virtual Machine Manager](#) ↗ (System Center 2008 R2)

Live Migration in Windows Server 2016

In Windows Server 2016, there are fewer restrictions on live migration deployment. It now works without Failover Clustering. Other functionality remains unchanged from previous releases of Live Migration. For more details on configuring and using live migration without Failover Clustering:

- [Set up hosts for live migration without Failover Clustering](#)
- [Use live migration without Failover Clustering to move a virtual machine](#)

Live Migration Overview

Article • 09/17/2020 • 2 minutes to read

Live migration is a Hyper-V feature in Windows Server. It allows you to transparently move running Virtual Machines from one Hyper-V host to another without perceived downtime. The primary benefit of live migration is flexibility; running Virtual Machines are not tied to a single host machine. This allows actions like draining a specific host of Virtual Machines before decommissioning or upgrading it. When paired with Windows Failover Clustering, live migration allows the creation of highly available and fault tolerant systems.

Related Technologies and Documentation

Live migration is often used in conjunction with a few related technologies like Failover Clustering and System Center Virtual Machine Manager. If you're using Live Migration via these technologies, here are pointers to their latest documentation:

- [Failover Clustering](#) (Windows Server 2016)
- [System Center Virtual Machine Manager](#) (System Center 2016)

If you're using older versions of Windows Server, or need details on features introduced in older versions of Windows Server, here are pointers to historical documentation:

- [Live Migration](#) (Windows Server 2008 R2)
- [Live Migration](#) (Windows Server 2012 R2)
- [Failover Clustering](#) (Windows Server 2012 R2)
- [Failover Clustering](#) (Windows Server 2008 R2)
- [System Center Virtual Machine Manager](#) (System Center 2012 R2)
- [System Center Virtual Machine Manager](#) ↗ (System Center 2008 R2)

Live Migration in Windows Server 2016

In Windows Server 2016, there are fewer restrictions on live migration deployment. It now works without Failover Clustering. Other functionality remains unchanged from previous releases of Live Migration. For more details on configuring and using live migration without Failover Clustering:

- [Set up hosts for live migration without Failover Clustering](#)
- [Use live migration without Failover Clustering to move a virtual machine](#)

Set up hosts for live migration without Failover Clustering

Article • 07/29/2021 • 7 minutes to read

Applies to: Windows Server 2022, Windows Server 2016, Microsoft Hyper-V Server 2016, Windows Server 2019, Microsoft Hyper-V Server 2019

This article shows you how to set up hosts that aren't clustered so you can do live migrations between them. Use these instructions if you didn't set up live migration when you installed Hyper-V, or if you want to change the settings. To set up clustered hosts, use tools for Failover Clustering.

Requirements for setting up live migration

To set up non-clustered hosts for live migration, you'll need:

- A user account with permission to perform the various steps. Membership in the local Hyper-V Administrators group or the Administrators group on both the source and destination computers meets this requirement, unless you're configuring constrained delegation. Membership in the Domain Administrators group is required to configure constrained delegation.
- The Hyper-V role in Windows Server 2016 or Windows Server 2012 R2 installed on the source and destination servers. You can do a live migration between hosts running Windows Server 2016 and Windows Server 2012 R2 if the virtual machine is at least version 5.
For version upgrade instructions, see [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#). For installation instructions, see [Install the Hyper-V role on Windows Server](#).

- Source and destination computers that either belong to the same Active Directory domain, or belong to domains that trust each other.
- The Hyper-V management tools installed on a computer running Windows Server 2016 or Windows 10, unless the tools are installed on the source or destination server and you'll run the tools from the server.

Consider options for authentication and networking

Consider how you want to set up the following:

- **Authentication:** Which protocol will be used to authenticate live migration traffic between the source and destination servers? The choice determines whether you'll need to sign on to the source server before starting a live migration:
 - Kerberos lets you avoid having to sign in to the server, but requires constrained delegation to be set up. See below for instructions.
 - CredSSP lets you avoid configuring constrained delegation, but requires you sign in to the source server. You can do this through a local console session, a Remote Desktop session, or a remote Windows PowerShell session.

CredSPP requires signing in for situations that might not be obvious. For example, if you sign in to TestServer01 to move a virtual machine to TestServer02, and then want to move the virtual machine back to TestServer01, you'll need to sign in to TestServer02 before you try to move the virtual machine back to TestServer01. If you don't do this, the authentication attempt fails, an error occurs, and the following message is displayed:

"Virtual machine migration operation failed at migration Source. Failed to establish a connection with host *computer name*: No credentials are available in the security package 0x8009030E."

- **Performance:** Does it make sense to configure performance options? These options can reduce network and CPU usage, as well as make live migrations go faster. Consider your requirements and your infrastructure, and test different configurations to help you decide. The options are described at the end of step 2.
- **Network preference:** Will you allow live migration traffic through any available network, or isolate the traffic to specific networks? As a security best practice, we recommend that you isolate the traffic onto trusted, private networks because live migration traffic is not encrypted when it is sent over the network. Network isolation can be achieved through a physically isolated network or through another trusted networking technology such as VLANs.

Step 1: Configure constrained delegation (optional)

If you have decided to use Kerberos to authenticate live migration traffic, configure constrained delegation using an account that is a member of the Domain Administrators group.

Use the Users and Computers snap-in to configure constrained delegation

1. Open the Active Directory Users and Computers snap-in. (From Server Manager, select the server if it's not selected, click **Tools >> Active Directory Users and Computers**).
2. From the navigation pane in **Active Directory Users and Computers**, select the domain and double-click the **Computers** folder.
3. From the **Computers** folder, right-click the computer account of the source server and then click **Properties**.
4. From **Properties**, click the **Delegation** tab.
5. On the delegation tab, select **Trust this computer for delegation to the specified services only** and then select **Use any authentication protocol**.
6. Click **Add**.
7. From **Add Services**, click **Users or Computers**.
8. From **Select Users or Computers**, type the name of the destination server. Click **Check Names** to verify it, and then click **OK**.
9. From **Add Services**, in the list of available services, do the following and then click **OK**:
 - To move virtual machine storage, select **cifs**. This is required if you want to move the storage along with the virtual machine, as well as if you want to move only a virtual machine's storage. If the server is configured to use SMB storage for Hyper-V, this should already be selected.
 - To move virtual machines, select **Microsoft Virtual System Migration Service**.
10. On the **Delegation** tab of the Properties dialog box, verify that the services you selected in the previous step are listed as the services to which the destination computer can present delegated credentials. Click **OK**.
11. From the **Computers** folder, select the computer account of the destination server and repeat the process. In the **Select Users or Computers** dialog box, be sure to

specify the name of the source server.

The configuration changes take effect after both of the following happen:

- The changes are replicated to the domain controllers that the servers running Hyper-V are logged into.
- The domain controller issues a new Kerberos ticket.

Step 2: Set up the source and destination computers for live migration

This step includes choosing options for authentication and networking. As a security best practice, we recommend that you select specific networks to use for live migration traffic, as discussed above. This step also shows you how to choose the performance option.

Use Hyper-V Manager to set up the source and destination computers for live migration

1. Open Hyper-V Manager. (From Server Manager, click **Tools >>Hyper-V Manager**.)
2. In the navigation pane, select one of the servers. (If it isn't listed, right-click **Hyper-V Manager**, click **Connect to Server**, type the server name, and click **OK**. Repeat to add more servers.)
3. In the Action pane, click **Hyper-V Settings >>Live Migrations**.
4. In the **Live Migrations** pane, check **Enable incoming and outgoing live migrations**.
5. Under **Simultaneous live migrations**, specify a different number if you don't want to use the default of 2.
6. Under **Incoming live migrations**, if you want to use specific network connections to accept live migration traffic, click **Add** to type the IP address information. Otherwise, click **Use any available network for live migration**. Click **OK**.
7. To choose Kerberos and performance options, expand **Live Migrations** and then select **Advanced Features**.
 - If you have configured constrained delegation, under **Authentication protocol**, select **Kerberos**.

- Under **Performance options**, review the details and choose a different option if it's appropriate for your environment.

8. Click **OK**.

9. Select the other server in Hyper-V Manager and repeat the steps.

Use Windows PowerShell to set up the source and destination computers for live migration

Three cmdlets are available for configuring live migration on non-clustered hosts: [Enable-VMMigration](#), [Set-VMMigrationNetwork](#), and [Set-VMHost](#). This example uses all three and does the following:

- Configures live migration on the local host
- Allows incoming migration traffic only on a specific network
- Chooses Kerberos as the authentication protocol

Each line represents a separate command.

```
PowerShell

PS C:\> Enable-VMMigration

PS C:\> Set-VMMigrationNetwork 192.168.10.1

PS C:\> Set-VMHost -VirtualMachineMigrationAuthenticationType Kerberos
```

`Set-VMHost` also lets you choose a performance option (and many other host settings). For example, to choose SMB but leave the authentication protocol set to the default of CredSSP, type:

```
PowerShell

PS C:\> Set-VMHost -VirtualMachineMigrationPerformanceOption SMB
```

This table describes how the performance options work.

Option	Description
TCP/IP	Copies the memory of the virtual machine to the destination server over a TCP/IP connection.
Compression	Compresses the memory content of the virtual machine before copying it to the destination server over a TCP/IP connection. Note: This is the default setting.

Option	Description
SMB	<p>Copies the memory of the virtual machine to the destination server over a SMB 3.0 connection.</p> <ul style="list-style-type: none">- SMB Direct is used when the network adapters on the source and destination servers have Remote Direct Memory Access (RDMA) capabilities enabled.- SMB Multichannel automatically detects and uses multiple connections when a proper SMB Multichannel configuration is identified. <p>For more information, see Improve Performance of a File Server with SMB Direct.</p>

Next steps

After you set up the hosts, you're ready to do a live migration. For instructions, see [Use live migration without Failover Clustering to move a virtual machine](#).

Use live migration without Failover Clustering to move a virtual machine

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This article shows you how to move a virtual machine by doing a live migration without using Failover Clustering. A live migration moves running virtual machines between Hyper-V hosts without any noticeable downtime.

To be able to do this, you'll need:

- A user account that's a member of the local Hyper-V Administrators group or the Administrators group on both the source and destination computers.
- The Hyper-V role in Windows Server 2016 or Windows Server 2012 R2 installed on the source and destination servers and set up for live migrations. You can do a live migration between hosts running Windows Server 2016 and Windows Server 2012 R2 if the virtual machine is at least version 5.

For version upgrade instructions, see [Upgrade virtual machine version in Hyper-V on Windows 10 or Windows Server 2016](#). For installation instructions, see [Set up hosts for live migration](#).

- The Hyper-V management tools installed on a computer running Windows Server 2016 or Windows 10, unless the tools are installed on the source or destination server and you'll run them from there.

Use Hyper-V Manager to move a running virtual machine

1. Open Hyper-V Manager. (From Server Manager, click **Tools >>Hyper-V Manager**.)
2. In the navigation pane, select one of the servers. (If it isn't listed, right-click **Hyper-V Manager**, click **Connect to Server**, type the server name, and click **OK**. Repeat to add more servers.)
3. From the **Virtual Machines** pane, right-click the virtual machine and then click **Move**. This opens the Move Wizard.
4. Use the wizard pages to choose the type of move, destination server, and options.

5. On the **Summary** page, review your choices and then click **Finish**.

Use Windows PowerShell to move a running virtual machine

The following example uses the Move-VM cmdlet to move a virtual machine named *LMTTest* to a destination server named *TestServer02* and moves the virtual hard disks and other file, such checkpoints and Smart Paging files, to the *D:\LMTTest* directory on the destination server.

```
PS C:\> Move-VM LMTTest TestServer02 -IncludeStorage -DestinationStoragePath  
D:\LMTTest
```

Troubleshooting

Failed to establish a connection

If you haven't set up constrained delegation, you must sign in to source server before you can move a virtual machine. If you don't do this, the authentication attempt fails, an error occurs, and this message is displayed:

"Virtual machine migration operation failed at migration Source. Failed to establish a connection with host *computer name*: No credentials are available in the security package 0x8009030E."

To fix this problem, sign in to the source server and try the move again. To avoid having to sign in to a source server before doing a live migration, set up constrained delegation. You'll need domain administrator credentials to set up constrained delegation. For instructions, see [Set up hosts for live migration](#).

Failed because the host hardware isn't compatible

If a virtual machine doesn't have processor compatibility turned on and has one or more snapshots, the move fails if the hosts have different processor versions. An error occurs and this message is displayed:

The virtual machine cannot be moved to the destination computer. The hardware on the destination computer is not compatible with the hardware requirements of this

virtual machine.

To fix this problem, shut down the virtual machine and turn on the processor compatibility setting.

1. From Hyper-V Manager, in the **Virtual Machines** pane, right-click the virtual machine and click **Settings**.
2. In the navigation pane, expand **Processors** and click **Compatibility**.
3. Check **Migrate to a computer with a different processor version**.
4. Click **OK**.

To use Windows PowerShell, use the [Set-VMProcessor](#) cmdlet:

```
PS C:\> Set-VMProcessor TestVM -CompatibilityForMigrationEnabled $true
```

Performance Tuning Hyper-V Servers

Article • 05/19/2022 • 2 minutes to read

Hyper-V is the virtualization server role in Windows Server 2016. Virtualization servers can host multiple virtual machines that are isolated from each other but share the underlying hardware resources by virtualizing the processors, memory, and I/O devices. By consolidating servers onto a single machine, virtualization can improve resource usage and energy efficiency and reduce the operational and maintenance costs of servers. In addition, virtual machines and the management APIs offer more flexibility for managing resources, balancing load, and provisioning systems.

Additional References

- [Hyper-V terminology](#)
- [Hyper-V architecture](#)
- [Hyper-V server configuration](#)
- [Hyper-V processor performance](#)
- [Hyper-V memory performance](#)
- [Hyper-V storage I/O performance](#)
- [Hyper-V network I/O performance](#)
- [Detecting bottlenecks in a virtualized environment](#)
- [Linux Virtual Machines](#)

Hyper-V Virtual Switch

Article • 07/29/2021

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This topic provides an overview of Hyper-V Virtual Switch, which provides you with the ability to connect virtual machines (VMs) to networks that are external to the Hyper-V host, including your organization's intranet and the Internet.

You can also connect to virtual networks on the server that is running Hyper-V when you deploy Software Defined Networking (SDN).

Note

In addition to this topic, the following Hyper-V Virtual Switch documentation is available.

- [Manage Hyper-V Virtual Switch](#)
- [Remote Direct Memory Access \(RDMA\) and Switch Embedded Teaming \(SET\)](#)
- [Network Switch Team Cmdlets in Windows PowerShell](#)
- [What's New in VMM 2016](#)
- [Set up the VMM networking fabric](#)
- [Hyper-V forum](#)
- [Hyper-V: The WFP virtual switch extension should be enabled if it is required by third party extensions](#)

For more information about other networking technologies, see [Networking in Windows Server 2016](#).

Hyper-V Virtual Switch is a software-based layer-2 Ethernet network switch that is available in Hyper-V Manager when you install the Hyper-V server role.

Hyper-V Virtual Switch includes programmatically managed and extensible capabilities to connect VMs to both virtual networks and the physical network. In addition, Hyper-V Virtual Switch provides policy enforcement for security, isolation, and service levels.

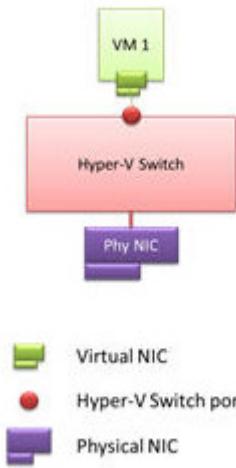
Note

Hyper-V Virtual Switch only supports Ethernet, and does not support any other wired local area network (LAN) technologies, such as Infiniband and Fibre Channel.

Hyper-V Virtual Switch includes tenant isolation capabilities, traffic shaping, protection against malicious virtual machines, and simplified troubleshooting.

With built-in support for Network Device Interface Specification (NDIS) filter drivers and Windows Filtering Platform (WFP) callout drivers, the Hyper-V Virtual Switch enables independent software vendors (ISVs) to create extensible plug-ins, called Virtual Switch Extensions, that can provide enhanced networking and security capabilities. Virtual Switch Extensions that you add to the Hyper-V Virtual Switch are listed in the Virtual Switch Manager feature of Hyper-V Manager.

In the following illustration, a VM has a virtual NIC that is connected to the Hyper-V Virtual Switch through a switch port.



Hyper-V Virtual Switch capabilities provide you with more options for enforcing tenant isolation, shaping and controlling network traffic, and employing protective measures against malicious VMs.

Note

In Windows Server 2016, a VM with a virtual NIC accurately displays the maximum throughput for the virtual NIC. To view the virtual NIC speed in **Network Connections**, right-click the desired virtual NIC icon and then click **Status**. The virtual NIC **Status** dialog box opens. In **Connection**, the value of **Speed** matches the speed of the physical NIC installed in the server.

Uses for Hyper-V Virtual Switch

Following are some use case scenarios for Hyper-V Virtual Switch.

Displaying statistics: A developer at a hosted cloud vendor implements a management package that displays the current state of the Hyper-V virtual switch. The management package can query switch-wide current capabilities, configuration settings, and individual port network statistics using WMI. The status of the switch is then displayed to give administrators a quick view of the state of the switch.

Resource tracking: A hosting company is selling hosting services priced according to the level of membership. Various membership levels include different network performance levels. The administrator allocates resources to meet the SLAs in a manner that balances network availability. The administrator programmatically tracks information such as the current usage of bandwidth assigned, and the number of virtual machine (VM) assigned virtual machine queue (VMQ) or IOV channels. The same program also periodically logs the resources in use in addition to the per-VM resources assigned for double entry tracking or resources.

Managing the order of switch extensions: An enterprise has installed extensions on their Hyper-V host to both monitor traffic and report intrusion detection. During maintenance, some extensions may be updated causing the order of extensions to change. A simple script program is run to reorder the extensions after updates.

Forwarding extension manages VLAN ID: A major switch company is building a forwarding extension that applies all policies for networking. One element that is managed is virtual local area network (VLAN) IDs. The virtual switch cedes control of the VLAN to a forwarding extension. The switch company's installation programmatically call a Windows Management Instrumentation (WMI) application programming interface (API) that turns on the transparency, telling the Hyper-V Virtual Switch to pass and take no action on VLAN tags.

Hyper-V Virtual Switch Functionality

Some of the principal features that are included in the Hyper-V Virtual Switch are:

- **ARP/ND Poisoning (spoofing) protection:** Provides protection against a malicious VM using Address Resolution Protocol (ARP) spoofing to steal IP addresses from other VMs. Provides protection against attacks that can be launched for IPv6 using Neighbor Discovery (ND) spoofing.
- **DHCP Guard protection:** Protects against a malicious VM representing itself as a Dynamic Host Configuration Protocol (DHCP) server for man-in-the-middle attacks.

- **Port ACLs:** Provides traffic filtering based on Media Access Control (MAC) or Internet Protocol (IP) addresses/ranges, which enables you to set up virtual network isolation.
- **Trunk mode to a VM:** Enables administrators to set up a specific VM as a virtual appliance, and then direct traffic from various VLANs to that VM.
- **Network traffic monitoring:** Enables administrators to review traffic that is traversing the network switch.
- **Isolated (private) VLAN:** Enables administrators to segregate traffic on multiple VLANs, to more easily establish isolated tenant communities.

Following is a list of capabilities that enhance Hyper-V Virtual Switch usability:

- **Bandwidth limit and burst support:** Bandwidth minimum guarantees amount of bandwidth reserved. Bandwidth maximum caps the amount of bandwidth a VM can consume.
- **Explicit Congestion Notification (ECN) marking support:** ECN marking, also known as Data Center TCP (DCTCP), enables the physical switch and operating system to regulate traffic flow such that the buffer resources of the switch are not flooded, which results in increased traffic throughput.
- **Diagnostics:** Diagnostics allow easy tracing and monitoring of events and packets through the virtual switch.

Host network requirements for Azure Stack HCI

Article • 04/17/2023

Applies to: Azure Stack HCI, versions 22H2 and 21H2

This topic discusses host networking considerations and requirements for Azure Stack HCI. For information on datacenter architectures and the physical connections between servers, see [Physical network requirements](#).

For information on how to simplify host networking using Network ATC, see [Simplify host networking with Network ATC](#).

Network traffic types

Azure Stack HCI network traffic can be classified by its intended purpose:

- **Management traffic:** Traffic to or from outside the local cluster. For example, storage replica traffic or traffic used by the administrator for management of the cluster like Remote Desktop, Windows Admin Center, Active Directory, etc.
- **Compute traffic:** Traffic originating from or destined to a virtual machine (VM).
- **Storage traffic:** Traffic using Server Message Block (SMB), for example Storage Spaces Direct or SMB-based live migration. This traffic is layer-2 traffic and is not routable.

 **Important**

Storage replica uses non-RDMA based SMB traffic. This and the directional nature of the traffic (North-South) makes it closely aligned to that of "management" traffic listed above, similar to that of a traditional file share.

Select a network adapter

Network adapters are qualified by the **network traffic types** (see above) they are supported for use with. As you review the [Windows Server Catalog](#), the Windows Server 2022 certification now indicates one or more of the following roles. Before purchasing a server for Azure Stack HCI, you must minimally have *at least* one adapter that is qualified for management, compute, and storage as all three traffic types are required on Azure Stack HCI. You can then use [Network ATC](#) to configure your adapters for the appropriate traffic types.

For more information about this role-based NIC qualification, please see this [link](#).

 **Important**

Using an adapter outside of its qualified traffic type is not supported.

Level	Management Role	Compute Role	Storage Role
Role-based distinction	Management	Compute Standard	Compute Storage
Maximum Award	Not Applicable	Compute Premium	Storage Premium

① Note

The highest qualification for any adapter in our ecosystem will contain the **Management**, **Compute Premium**, and **Storage Premium** qualifications.

Compatible with the following versions of Microsoft Windows



Windows Server 2022 x64

Certified for Windows

- Management
- Compute Premium
- Storage Premium



Driver Requirements

Inbox drivers are not supported for use with Azure Stack HCI. To identify if your adapter is using an inbox driver, run the following cmdlet. An adapter is using an inbox driver if the **DriverProvider** property is **Microsoft**.

Powershell

```
Get-NetAdapter -Name <AdapterName> | Select *Driver*
```

Overview of key network adapter capabilities

Important network adapter capabilities used by Azure Stack HCI include:

- Dynamic Virtual Machine Multi-Queue (Dynamic VMMQ or d.VMMQ)
- Remote Direct Memory Access (RDMA)
- Guest RDMA
- Switch Embedded Teaming (SET)

Dynamic VMMQ

All network adapters with the Compute (Premium) qualification support Dynamic VMMQ. Dynamic VMMQ requires the use of Switch Embedded Teaming.

Applicable traffic types: compute

Certifications required: Compute (Premium)

Dynamic VMMQ is an intelligent, receive-side technology. It builds upon its predecessors of Virtual Machine Queue (VMQ), Virtual Receive Side Scaling (vRSS), and VMMQ, to provide three primary improvements:

- Optimizes host efficiency by using fewer CPU cores.
- Automatic tuning of network traffic processing to CPU cores, thus enabling VMs to meet and maintain expected throughput.
- Enables "bursty" workloads to receive the expected amount of traffic.

For more information on Dynamic VMMQ, see the blog post [Synthetic accelerations](#).

RDMA

RDMA is a network stack offload to the network adapter. It allows SMB storage traffic to bypass the operating system for processing.

RDMA enables high-throughput, low-latency networking, using minimal host CPU resources. These host CPU resources can then be used to run additional VMs or containers.

Applicable traffic types: host storage

Certifications required: Storage (Standard)

All adapters with Storage (Standard) or Storage (Premium) qualification support host-side RDMA. For more information on using RDMA with guest workloads, see the "Guest RDMA" section later in this article.

Azure Stack HCI supports RDMA with either the Internet Wide Area RDMA Protocol (iWARP) or RDMA over Converged Ethernet (RoCE) protocol implementations.

ⓘ Important

RDMA adapters only work with other RDMA adapters that implement the same RDMA protocol (iWARP or RoCE).

Not all network adapters from vendors support RDMA. The following table lists those vendors (in alphabetical order) that offer certified RDMA adapters. However, there are hardware vendors not included in this list that also support RDMA. See the [Windows Server Catalog](#) to find adapters with the Storage (Standard) or Storage (Premium) qualification which require RDMA support.

ⓘ Note

InfiniBand (IB) is not supported with Azure Stack HCI.

NIC vendor	iWARP	RoCE
------------	-------	------

NIC vendor	iWARP	RoCE
Broadcom	No	Yes
Intel	Yes	Yes (some models)
Marvell (Qlogic)	Yes	Yes
Nvidia	No	Yes

For more information on deploying RDMA for the host, we highly recommend you use Network ATC. For information on manual deployment see the [SDN GitHub repo ↗](#).

iWARP

iWARP uses Transmission Control Protocol (TCP), and can be optionally enhanced with Priority-based Flow Control (PFC) and Enhanced Transmission Service (ETS).

Use iWARP if:

- You don't have experience managing RDMA networks.
- You don't manage or are uncomfortable managing your top-of-rack (ToR) switches.
- You won't be managing the solution after deployment.
- You already have deployments that use iWARP.
- You're unsure which option to choose.

RoCE

RoCE uses User Datagram Protocol (UDP), and requires PFC and ETS to provide reliability.

Use RoCE if:

- You already have deployments with RoCE in your datacenter.
- You're comfortable managing the DCB network requirements.

Guest RDMA

Guest RDMA enables SMB workloads for VMs to gain the same benefits of using RDMA on hosts.

Applicable traffic types: Guest-based storage

Certifications required: Compute (Premium)

The primary benefits of using Guest RDMA are:

- CPU offload to the NIC for network traffic processing.
- Extremely low latency.
- High throughput.

For more information, download the document from the [SDN GitHub repo ↗](#).

SET

SET is a software-based teaming technology that has been included in the Windows Server operating system since Windows Server 2016. SET requires a Compute (Standard) or Compute (Premium) adapter.

Applicable traffic types: compute, storage, and management

Certifications required: Compute (Standard) or Compute (Premium)

SET is the only teaming technology supported by Azure Stack HCI. SET works well with compute, storage, and management traffic.

Important

Azure Stack HCI doesn't support NIC teaming with the older Load Balancing/Failover (LBFO).

See the blog post [Teaming in Azure Stack HCI](#) for more information on LBFO in Azure Stack HCI.

SET is important for Azure Stack HCI because it's the only teaming technology that enables:

- Teaming of RDMA adapters (if needed).
- Guest RDMA.
- Dynamic VMMQ.
- Other key Azure Stack HCI features (see [Teaming in Azure Stack HCI](#)).

SET requires the use of symmetric (identical) adapters. Symmetric network adapters are those that have the same:

- make (vendor)
- model (version)
- speed (throughput)
- configuration

In 22H2, Network ATC will automatically detect and inform you if the adapters you've chosen are asymmetric. The easiest way to manually identify if adapters are symmetric is if the speeds and interface descriptions are **exact** matches. They can deviate only in the numeral listed in the description. Use the [Get-NetAdapterAdvancedProperty](#) cmdlet to ensure the configuration reported lists the same property values.

See the following table for an example of the interface descriptions deviating only by numeral (#):

Name	Interface description	Link speed
NIC1	Network Adapter #1	25 Gbps
NIC2	Network Adapter #2	25 Gbps
NIC3	Network Adapter #3	25 Gbps

Name	Interface description	Link speed
NIC4	Network Adapter #4	25 Gbps

① Note

SET supports only switch-independent configuration by using either Dynamic or Hyper-V Port load-balancing algorithms. For best performance, Hyper-V Port is recommended for use on all NICs that operate at or above 10 Gbps. Network ATC makes all the required configurations for SET.

RDMA traffic considerations

If you implement DCB, you must ensure that the PFC and ETS configuration is implemented properly across every network port, including network switches. DCB is required for RoCE and optional for iWARP.

For detailed information on how to deploy RDMA, download the document from the [SDN GitHub repo ↗](#).

RoCE-based Azure Stack HCI implementations require the configuration of three PFC traffic classes, including the default traffic class, across the fabric and all hosts.

Cluster traffic class

This traffic class ensures that there's enough bandwidth reserved for cluster heartbeats:

- Required: Yes
- PFC-enabled: No
- Recommended traffic priority: Priority 7
- Recommended bandwidth reservation:
 - 10 GbE or lower RDMA networks = 2 percent
 - 25 GbE or higher RDMA networks = 1 percent

RDMA traffic class

This traffic class ensures that there's enough bandwidth reserved for lossless RDMA communications by using SMB Direct:

- Required: Yes
- PFC-enabled: Yes
- Recommended traffic priority: Priority 3 or 4
- Recommended bandwidth reservation: 50 percent

Default traffic class

This traffic class carries all other traffic not defined in the cluster or RDMA traffic classes, including VM traffic and management traffic:

- Required: By default (no configuration necessary on the host)
- Flow control (PFC)-enabled: No
- Recommended traffic class: By default (Priority 0)
- Recommended bandwidth reservation: By default (no host configuration required)

Storage traffic models

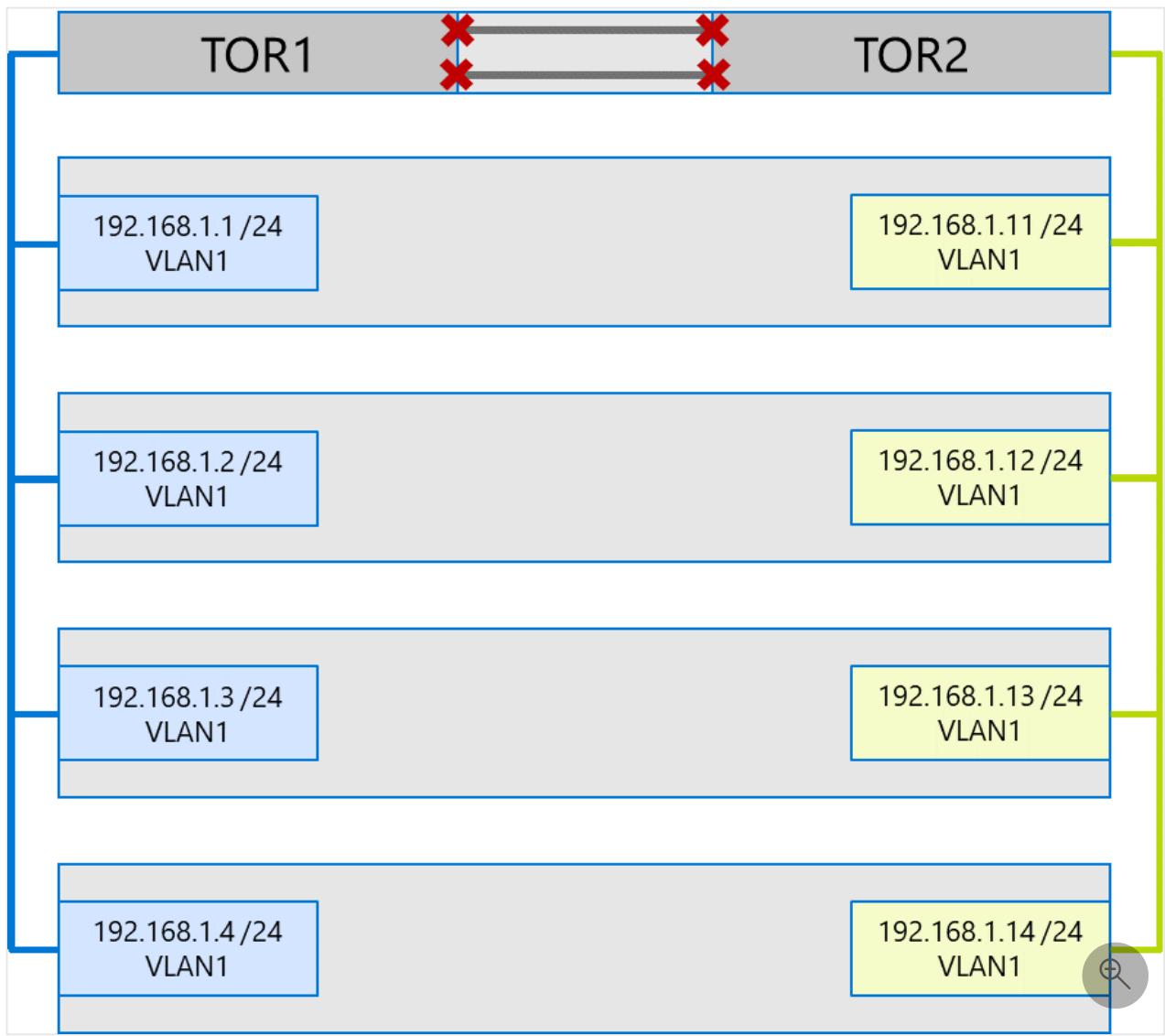
SMB provides many benefits as the storage protocol for Azure Stack HCI, including SMB Multichannel. SMB Multichannel isn't covered in this article, but it's important to understand that traffic is multiplexed across every possible link that SMB Multichannel can use.

① Note

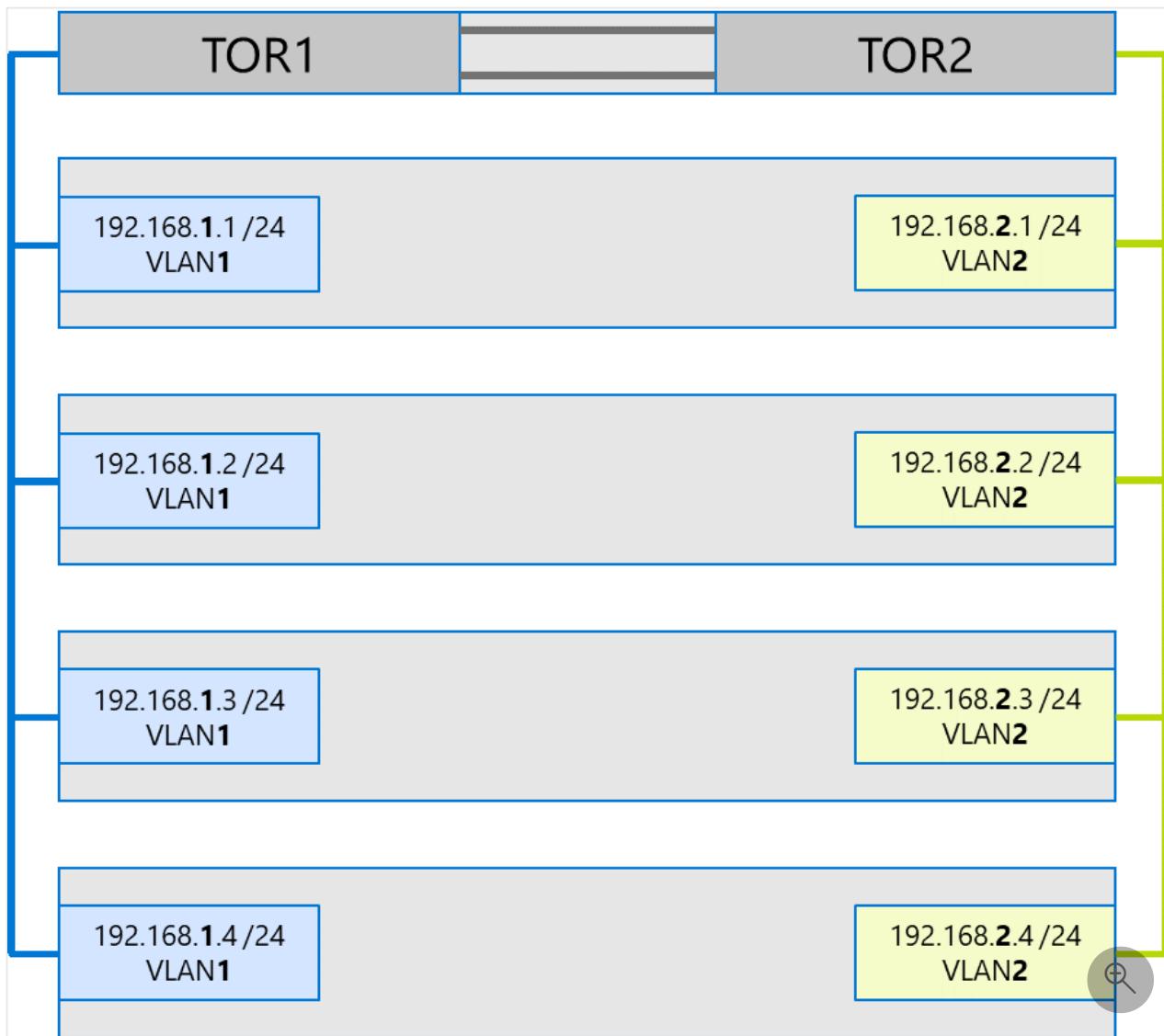
We recommend using multiple subnets and VLANs to separate storage traffic in Azure Stack HCI.

Consider the following example of a four node cluster. Each server has two storage ports (left and right side). Because each adapter is on the same subnet and VLAN, SMB Multichannel will spread connections across all available links. Therefore, the left-side port on the first server (192.168.1.1) will make a connection to the left-side port on the second server (192.168.1.2). The right-side port on the first server (192.168.1.12) will connect to the right-side port on the second server. Similar connections are established for the third and fourth servers.

However, this creates unnecessary connections and causes congestion at the interlink (multi-chassis link aggregation group or MC-LAG) that connects the ToR switches (marked with Xs). See the following diagram:



The recommended approach is to use separate subnets and VLANs for each set of adapters. In the following diagram, the right-hand ports now use subnet 192.168.2.x /24 and VLAN2. This allows traffic on the left-side ports to remain on TOR1 and the traffic on the right-side ports to remain on TOR2.



Traffic bandwidth allocation

The following table shows example bandwidth allocations of various traffic types, using common adapter speeds, in Azure Stack HCI. Note that this is an example of a *converged solution*, where all traffic types (compute, storage, and management) run over the same physical adapters, and are teamed by using SET.

Because this use case poses the most constraints, it represents a good baseline. However, considering the permutations for the number of adapters and speeds, this should be considered an example and not a support requirement.

The following assumptions are made for this example:

- There are two adapters per team.
- Storage Bus Layer (SBL), Cluster Shared Volume (CSV), and Hyper-V (Live Migration) traffic:
 - Use the same physical adapters.
 - Use SMB.
- SMB is given a 50 percent bandwidth allocation by using DCB.

- SBL/CSV is the highest priority traffic, and receives 70 percent of the SMB bandwidth reservation.
- Live Migration (LM) is limited by using the `Set-SMBBandwidthLimit` cmdlet, and receives 29 percent of the remaining bandwidth.
- If the available bandwidth for Live Migration is \geq 5 Gbps, and the network adapters are capable, use RDMA. Use the following cmdlet to do so:

Powershell

```
Set-VMHost -VirtualMachineMigrationPerformanceOption SMB
```

- If the available bandwidth for Live Migration is $<$ 5 Gbps, use compression to reduce blackout times. Use the following cmdlet to do so:

Powershell

```
Set-VMHost -VirtualMachineMigrationPerformanceOption Compression
```

- If you're using RDMA for Live Migration traffic, ensure that Live Migration traffic can't consume the entire bandwidth allocated to the RDMA traffic class by using an SMB bandwidth limit. Be careful, because this cmdlet takes entry in bytes per second (Bps), whereas network adapters are listed in bits per second (bps). Use the following cmdlet to set a bandwidth limit of 6 Gbps, for example:

Powershell

```
Set-SMBBandwidthLimit -Category LiveMigration -BytesPerSecond 750MB
```

ⓘ Note

750 MBps in this example equates to 6 Gbps.

Here is the example bandwidth allocation table:

NIC speed	Teamed bandwidth	SMB bandwidth reservation**	SBL/CSV %	SBL/CSV bandwidth	Live Migration %	Max Live Migration bandwidth	Heartbeat %	Heartbeat bandwidth
10 Gbps	20 Gbps	10 Gbps	70%	7 Gbps	**	200 Mbps		
25 Gbps	50 Gbps	25 Gbps	70%	17.5 Gbps	29%	7.25 Gbps	1%	250 Mbps
40 Gbps	80 Gbps	40 Gbps	70%	28 Gbps	29%	11.6 Gbps	1%	400 Mbps

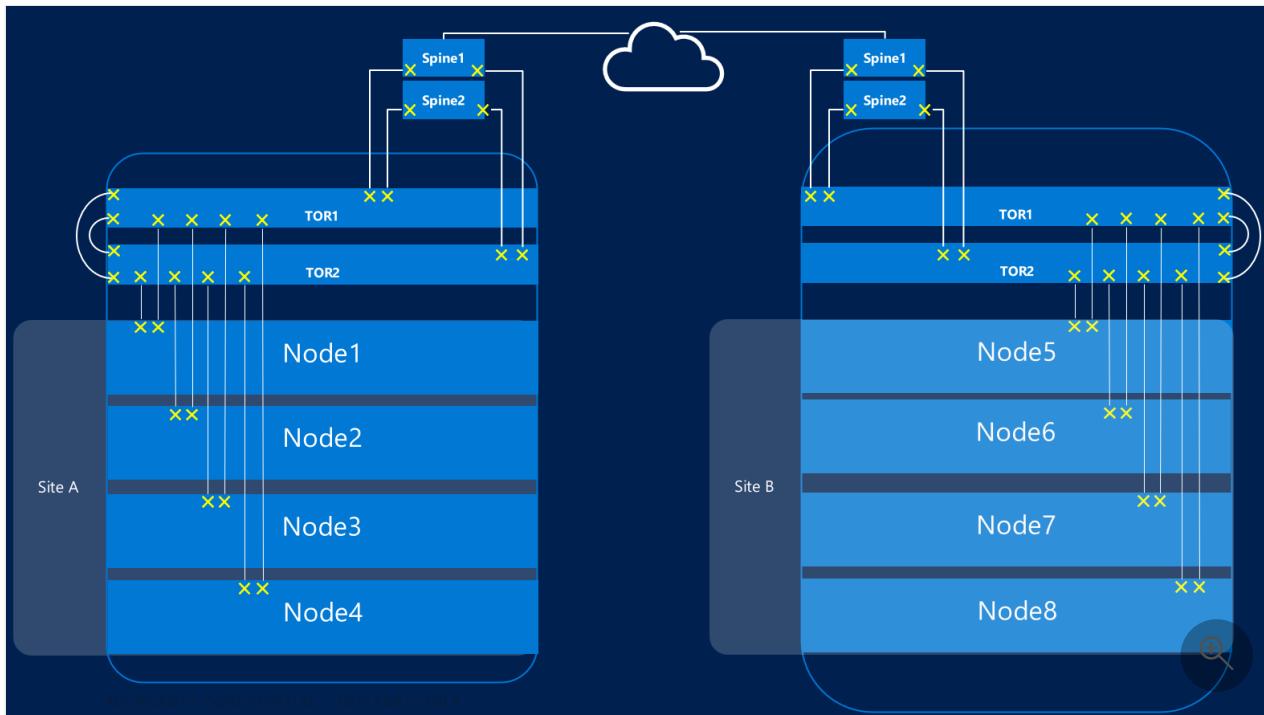
NIC speed	Teamed bandwidth	SMB bandwidth reservation**	SBL/CSV %	SBL/CSV bandwidth	Live Migration %	Max Live Migration bandwidth	Heartbeat %	Heartbeat bandwidth
50 Gbps	100 Gbps	50 Gbps	70%	35 Gbps	29%	14.5 Gbps	1%	500 Mbps
100 Gbps	200 Gbps	100 Gbps	70%	70 Gbps	29%	29 Gbps	1%	1 Gbps
200 Gbps	400 Gbps	200 Gbps	70%	140 Gbps	29%	58 Gbps	1%	2 Gbps

* Use compression rather than RDMA, because the bandwidth allocation for Live Migration traffic is <5 Gbps.

** 50 percent is an example bandwidth reservation.

Stretched clusters

Stretched clusters provide disaster recovery that spans multiple datacenters. In its simplest form, a stretched Azure Stack HCI cluster network looks like this:



Stretched cluster requirements

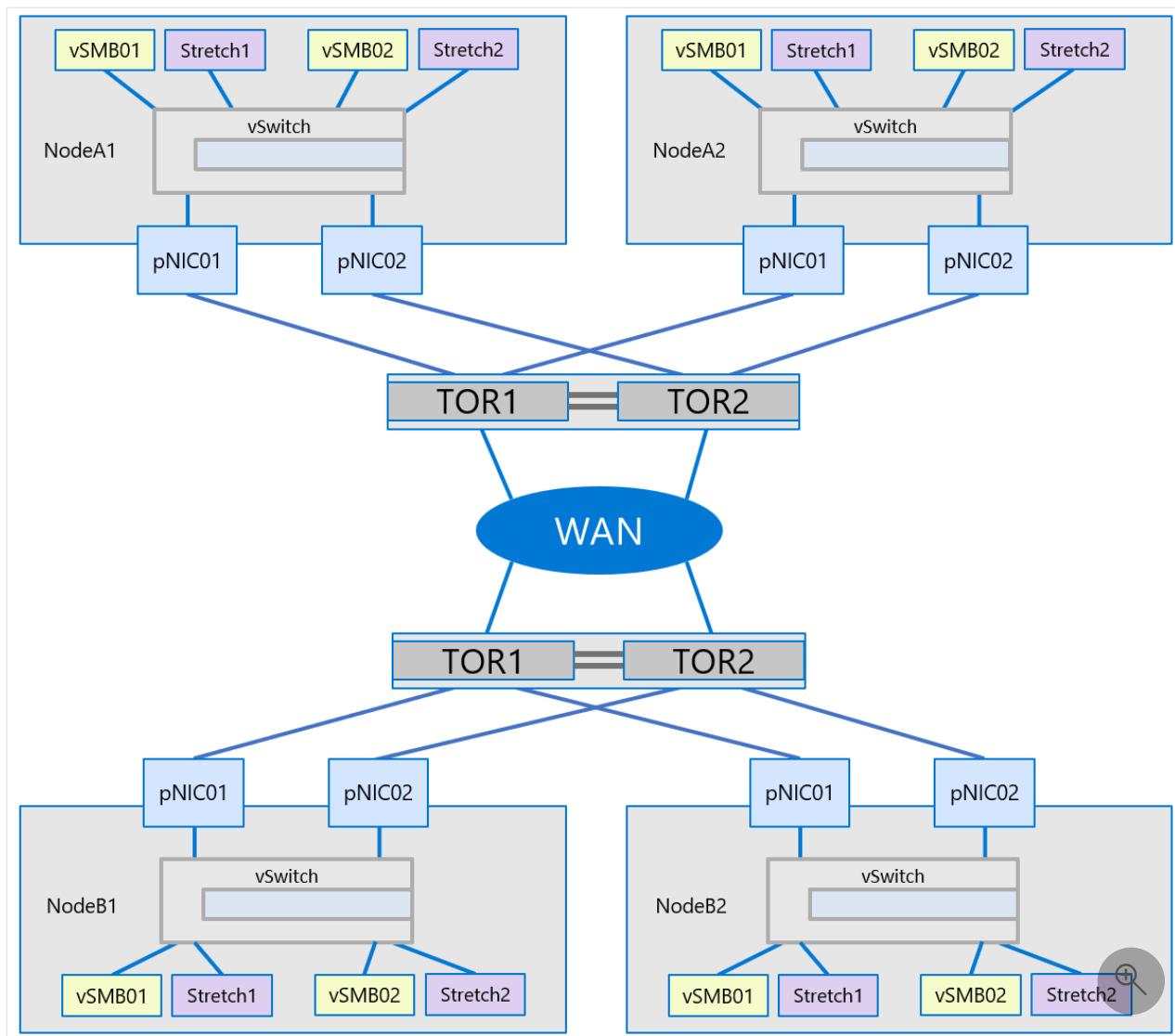
Stretched clusters have the following requirements and characteristics:

- RDMA is limited to a single site, and isn't supported across different sites or subnets.
- Servers in the same site must reside in the same rack and Layer-2 boundary.

- Host communication between sites must cross a Layer-3 boundary; stretched Layer-2 topologies aren't supported.
- Have enough bandwidth to run the workloads at the other site. In the event of a failover, the alternate site will need to run all traffic. We recommend that you provision sites at 50 percent of their available network capacity. This isn't a requirement, however, if you are able to tolerate lower performance during a failover.
- Replication between sites (north/south traffic) can use the same physical NICs as the local storage (east/west traffic). If you're using the same physical adapters, these adapters must be teamed with SET. The adapters must also have additional virtual NICs provisioned for routable traffic between sites.
- Adapters used for communication between sites:
 - Can be physical or virtual (host vNIC). If adapters are virtual, you must provision one vNIC in its own subnet and VLAN per physical NIC.
 - Must be on their own subnet and VLAN that can route between sites.
 - RDMA must be disabled by using the `Disable-NetAdapterRDMA` cmdlet. We recommend that you explicitly require Storage Replica to use specific interfaces by using the `Set-SRNetworkConstraint` cmdlet.
 - Must meet any additional requirements for Storage Replica.

Stretched cluster example

The following example illustrates a stretched cluster configuration. To ensure that a specific virtual NIC is mapped to a specific physical adapter, use the [Set-VMNetworkAdapterTeammapping](#) cmdlet.



The following shows the details for the example stretched cluster configuration.

! Note

Your exact configuration, including NIC names, IP addresses, and VLANs, might be different than what is shown. This is used only as a reference configuration that can be adapted to your environment.

SiteA – Local replication, RDMA enabled, non-routable between sites

Node name	vNIC name	Physical NIC (mapped)	VLAN	IP and subnet	Traffic scope
NodeA1	vSMB01	pNIC01	711	192.168.1.1/24	Local Site Only
NodeA2	vSMB01	pNIC01	711	192.168.1.2/24	Local Site Only
NodeA1	vSMB02	pNIC02	712	192.168.2.1/24	Local Site Only
NodeA2	vSMB02	pNIC02	712	192.168.2.2/24	Local Site Only

SiteB – Local replication, RDMA enabled, non-routable between sites

Node name	vNIC name	Physical NIC (mapped)	VLAN	IP and subnet	Traffic scope
NodeB1	vSMB01	pNIC01	711	192.168.1.1/24	Local Site Only
NodeB2	vSMB01	pNIC01	711	192.168.1.2/24	Local Site Only
NodeB1	vSMB02	pNIC02	712	192.168.2.1/24	Local Site Only
NodeB2	vSMB02	pNIC02	712	192.168.2.2/24	Local Site Only

SiteA – Stretched replication, RDMA disabled, routable between sites

Node name	vNIC name	Physical NIC (mapped)	IP and subnet	Traffic scope
NodeA1	Stretch1	pNIC01	173.0.0.1/8	Cross-Site Routable
NodeA2	Stretch1	pNIC01	173.0.0.2/8	Cross-Site Routable
NodeA1	Stretch2	pNIC02	174.0.0.1/8	Cross-Site Routable
NodeA2	Stretch2	pNIC02	174.0.0.2/8	Cross-Site Routable

SiteB – Stretched replication, RDMA disabled, routable between sites

Node name	vNIC name	Physical NIC (mapped)	IP and subnet	Traffic scope
NodeB1	Stretch1	pNIC01	175.0.0.1/8	Cross-Site Routable
NodeB2	Stretch1	pNIC01	175.0.0.2/8	Cross-Site Routable
NodeB1	Stretch2	pNIC02	176.0.0.1/8	Cross-Site Routable
NodeB2	Stretch2	pNIC02	176.0.0.2/8	Cross-Site Routable

Next steps

- Learn about network switch and physical network requirements. See [Physical network requirements](#).
- Learn how to simplify host networking using Network ATC. See [Simplify host networking with Network ATC](#).
- Brush up on [failover clustering networking basics](#).
- For deployment, see [Create a cluster using Windows Admin Center](#).
- For deployment, see [Create a cluster using Windows PowerShell](#).

Manage Hyper-V Virtual Switch

Article • 07/29/2021 • 2 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

You can use this topic to access Hyper-V Virtual Switch management content.

This section contains the following topics.

- [Configure VLANs on Hyper-V Virtual Switch Ports](#)
- [Create Security Policies with Extended Port Access Control Lists](#)

Create Security Policies with Extended Port Access Control Lists

Article • 07/29/2021 • 9 minutes to read

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

This topic provides information about extended port Access Control Lists (ACLs) in Windows Server 2016. You can configure extended ACLs on the Hyper-V Virtual Switch to allow and block network traffic to and from the virtual machines (VMs) that are connected to the switch via virtual network adapters.

This topic contains the following sections.

- [Detailed ACL rules](#)
- [Stateful ACL rules](#)

Detailed ACL rules

Hyper-V Virtual Switch extended ACLs allow you to create detailed rules that you can apply to individual VM network adapters that are connected to the Hyper-V Virtual Switch. The ability to create detailed rules allows Enterprises and Cloud Service Providers (CSPs) to address network-based security threats in a multitenant shared server environment.

With extended ACLs, rather than having to create broad rules that block or allow all traffic from all protocols to or from a VM, you can now block or allow the network traffic of individual protocols that are running on VMs. You can create extended ACL rules in Windows Server 2016 that include the following 5-tuple set of parameters: source IP address, destination IP address, protocol, source port, and destination port. In addition, each rule can specify network traffic direction (in or out), and the action the rule supports (block or allow traffic).

For example, you can configure port ACLs for a VM to allow all incoming and outgoing HTTP and HTTPS traffic on port 80, while blocking the network traffic of all other protocols on all ports.

This ability to designate the protocol traffic that can or cannot be received by tenant VMs provides flexibility when you configure security policies.

Configuring ACL rules with Windows PowerShell

To configure an extended ACL, you must use the Windows PowerShell command **Add-VMNetworkAdapterExtendedAcl**. This command has four different syntaxes, with a distinct use for each syntax:

1. Add an extended ACL to all of the network adapters of a named VM - which is specified by the first parameter, **-VMName**. Syntax:

ⓘ Note

If you want to add an extended ACL to one network adapter rather than all, you can specify the network adapter with the parameter **-VMNetworkAdapterName**.

```
Add-VMNetworkAdapterExtendedAcl [-VMName] <string[]> [-Action]
<VMNetworkAdapterExtendedAclAction> {Allow | Deny}
[-Direction] <VMNetworkAdapterExtendedAclDirection> {Inbound |
Outbound} [[-LocalIPAddress] <string>]
[[[-RemoteIPAddress] <string>] [[-LocalPort] <string>] [[-
RemotePort] <string>] [[-Protocol] <string>] [-Weight]
<int> [-Stateful <bool>] [-IdleSessionTimeout <int>] [-IsolationID
<int>] [-Passthru] [-VMNetworkAdapterName
<string>] [-ComputerName <string[]>] [-WhatIf] [-Confirm]
[<CommonParameters>]
```

2. Add an extended ACL to a specific virtual network adapter on a specific VM.

Syntax:

```
Add-VMNetworkAdapterExtendedAcl [-VMNetworkAdapter]
<VMNetworkAdapterBase[]> [-Action]
<VMNetworkAdapterExtendedAclAction> {Allow | Deny} [-Direction]
<VMNetworkAdapterExtendedAclDirection> {Inbound |
Outbound} [[-LocalIPAddress] <string>] [[-RemoteIPAddress]
<string>] [[-LocalPort] <string>] [[-RemotePort]
<string>] [[-Protocol] <string>] [-Weight] <int> [-Stateful <bool>]
[-IdleSessionTimeout <int>] [-IsolationID
<int>] [-Passthru] [-WhatIf] [-Confirm] [<CommonParameters>]
```

3. Add an extended ACL to all virtual network adapters that are reserved for use by the Hyper-V host management operating system.

(!) Note

If you want to add an extended ACL to one network adapter rather than all, you can specify the network adapter with the parameter -VMNetworkAdapterName.

```
Add-VMNetworkAdapterExtendedAcl [-Action]
<VMNetworkAdapterExtendedAclAction> {Allow | Deny} [-Direction]
    <VMNetworkAdapterExtendedAclDirection> {Inbound | Outbound} [[-LocalIPAddress] <string>] [[-RemoteIPAddress]
        <string>] [[-LocalPort] <string>] [[-RemotePort] <string>] [[-Protocol] <string>] [-Weight] <int> -ManagementOS
        [-Stateful <bool>] [-IdleSessionTimeout <int>] [-IsolationID <int>]
    [-Passthru] [-VMNetworkAdapterName <string>]
        [-ComputerName <string[]>] [-WhatIf] [-Confirm]
    [<CommonParameters>]
```

4. Add an extended ACL to a VM object that you have created in Windows PowerShell, such as \$vm = get-vm "my_vm". In the next line of code you can run this command to create an extended ACL with the following syntax:

```
Add-VMNetworkAdapterExtendedAcl [-VM] <VirtualMachine[]> [-Action]
<VMNetworkAdapterExtendedAclAction> {Allow |
    Deny} [-Direction] <VMNetworkAdapterExtendedAclDirection> {Inbound |
    Outbound} [[-LocalIPAddress] <string>]
    [[-RemoteIPAddress] <string>] [[-LocalPort] <string>] [[-RemotePort]
        <string>] [[-Protocol] <string>] [-Weight]
        <int> [-Stateful <bool>] [-IdleSessionTimeout <int>] [-IsolationID
    <int>] [-Passthru] [-VMNetworkAdapterName
        <string>] [-WhatIf] [-Confirm] [<CommonParameters>]
```

Detailed ACL rule examples

Following are several examples of how you can use the **Add-VMNetworkAdapterExtendedAcl** command to configure extended port ACLs and to create security policies for VMs.

- Enforce application-level security
- Enforce both user-level and application-level security
- Provide security support to a non-TCP/UDP application

Note

The values for the rule parameter **Direction** in the tables below are based on traffic flow to or from the VM for which you are creating the rule. If the VM is receiving traffic, the traffic is inbound; if the VM is sending traffic, the traffic is outbound. For example, if you apply a rule to a VM that blocks inbound traffic, the direction of inbound traffic is from external resources to the VM. If you apply a rule that blocks outbound traffic, the direction of outbound traffic is from the local VM to external resources.

Enforce application-level security

Because many application servers use standardized TCP/UDP ports to communicate with client computers, it is easy to create rules that block or allow access to an application server by filtering traffic going to and coming from the port designated to the application.

For example, you might want to allow a user to login to an application server in your datacenter by using Remote Desktop Connection (RDP). Because RDP uses TCP port 3389, you can quickly set up the following rule:

Source IP	Destination IP	Protocol	Source Port	Destination Port	Direction	Action
*	*	TCP	*	3389	In	Allow

Following are two examples of how you can create rules with Windows PowerShell commands. The first example rule blocks all traffic to the VM named "ApplicationServer." The second example rule, which is applied to the network adapter of the VM named "ApplicationServer," allows only inbound RDP traffic to the VM.

Note

When you create rules, you can use the **-Weight** parameter to determine the order in which the Hyper-V Virtual Switch processes the rules. Values for **-Weight** are expressed as integers; rules with a higher integer are processed before rules with lower integers. For example, if you have applied two rules to a VM network adapter, one with a weight of 1 and one with a weight of 10, the rule with the weight of 10 is applied first.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Deny" -  
Direction "Inbound" -Weight 1  
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Allow"  
-Direction "Inbound" -LocalPort 3389 -Protocol "TCP" -Weight 10
```

Enforce both user-level and application-level security

Because a rule can match a 5-tuple IP packet (Source IP, Destination IP, Protocol, Source Port, and Destination Port), the rule can enforce a more detailed security policy than a Port ACL.

For example, if you want to provide DHCP service to a limited number of client computers using a specific set of DHCP servers, you can configure the following rules on the Windows Server 2016 computer that is running Hyper-V, where the user VMs are hosted:

Source IP	Destination IP	Protocol	Source Port	Destination Port	Direction	Action
*	255.255.255.255	UDP	*	67	Out	Allow
*	10.175.124.0/25	UDP	*	67	Out	Allow
10.175.124.0/25	*	UDP	*	68	In	Allow

Following are examples of how you can create these rules with Windows PowerShell commands.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Deny" -  
Direction "Outbound" -Weight 1  
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -  
Direction "Outbound" -RemoteIPAddress 255.255.255.255 -RemotePort 67 -  
Protocol "UDP"-Weight 10  
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -  
Direction "Outbound" -RemoteIPAddress 10.175.124.0/25 -RemotePort 67 -  
Protocol "UDP"-Weight 20  
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -  
Direction "Inbound" -RemoteIPAddress 10.175.124.0/25 -RemotePort 68 -  
Protocol "UDP"-Weight 20
```

Provide security support to a non-TCP/UDP application

While most network traffic in a datacenter is TCP and UDP, there is still some traffic that utilizes other protocols. For example, if you want to permit a group of servers to run an

IP-multicast application that relies on Internet Group Management Protocol (IGMP), you can create the following rule.

 **Note**

IGMP has a designated IP protocol number of 0x02.

Source IP	Destination IP	Protocol	Source Port	Destination Port	Direction	Action
*	*	0x02	*	*	In	Allow
*	*	0x02	*	*	Out	Allow

Following is an example of how you can create these rules with Windows PowerShell commands.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -  
Direction "Inbound" -Protocol 2 -Weight 20  
Add-VMNetworkAdapterExtendedAcl -VMName "ServerName" -Action "Allow" -  
Direction "Outbound" -Protocol 2 -Weight 20
```

Stateful ACL rules

Another new capability of extended ACLs allows you to configure stateful rules. A stateful rule filters packets based on five attributes in a packet - Source IP, Destination IP, Protocol, Source Port, and Destination Port.

Stateful rules have the following capabilities:

- They always allow traffic and are not used to block traffic.
- If you specify that the value for the parameter **Direction** is inbound and traffic matches the rule, Hyper-V Virtual Switch dynamically creates a matching rule that allows the VM to send outbound traffic in response to the external resource.
- If you specify that the value for the parameter **Direction** is outbound and traffic matches the rule, Hyper-V Virtual Switch dynamically creates a matching rule that allows the external resource inbound traffic to be received by the VM.
- They include a timeout attribute that is measured in seconds. When a network packet arrives at the switch and the packet matches a stateful rule, Hyper-V Virtual

Switch creates a state so that all subsequent packets in both directions of the same flow are allowed. The state expires if there is no traffic in either direction in the period of time that is specified by the timeout value.

Following is an example of how you can use stateful rules.

Allow inbound remote server traffic only after it is contacted by the local server

In some cases, a stateful rule must be employed because only a stateful rule can keep track of a known, established connection, and distinguish the connection from other connections.

For example, if you want to allow a VM application server to initiate connections on port 80 to web services on the Internet, and you want the remote Web servers to be able to respond to the VM traffic, you can configure a stateful rule that allows initial outbound traffic from the VM to the Web services; because the rule is stateful, return traffic to the VM from the Web servers is also allowed. For security reasons, you can block all other inbound network traffic to the VM.

To achieve this rule configuration, you can use the settings in the table below.

ⓘ Note

Due to formatting restrictions and the amount of information in the table below, the information is displayed differently than in previous tables in this document.

Parameter	Rule 1	Rule 2	Rule 3
Source IP	*	*	*
Destination IP	*	*	*
Protocol	*	*	TCP
Source Port	*	*	*
Destination Port	*	*	80
Direction	In	Out	Out
Action	Deny	Deny	Allow
Stateful	No	No	Yes

Parameter	Rule 1	Rule 2	Rule 3
Timeout (in seconds)	N/A	N/A	3600

The stateful rule allows the VM application server to connect to a remote Web server. When the first packet is sent out, Hyper-V Virtual switch dynamically creates two flow states to allow all packets sent to and all returning packets from the remote Web server. When the flow of packets between the servers stops, the flow states time out in the designated timeout value of 3600 seconds, or one hour.

Following is an example of how you can create these rules with Windows PowerShell commands.

```
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Deny" -Direction "Inbound" -Weight 1
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Deny" -Direction "Outbound" -Weight 1
Add-VMNetworkAdapterExtendedAcl -VMName "ApplicationServer" -Action "Allow" -Direction "Outbound" 80 "TCP" -Weight 100 -Stateful -Timeout 3600
```